



CENTENVEILINGSYSTEEM VOOR ZWOOZI

Afstudeerverslag



3 OKTOBER 2016

HAAGSE HOGESCHOOL ZOETERMEER
OPLEIDING: INFORMATICA

Locatie: Folkert Elsingastraat 3-7
Auteur: Daveron van der Ham
E-mailadres: daveron.van.der.ham@gmail.com
Studentnummer: 12053635

Referaat

Ham, D.A.J. van der Afstudeerverslag, Zoetermeer, Haagse Hogeschool, 2016

Afstuderen van de afstudeerder vond plaats op de Haagse Hogeschool(HHS), afdeling Acedemie voor ICT&Media, opleiding Informatica.

Descriptoren:

- Afstuderen
- Procesverslag
- Plan van Aanpak
- Systeemontwikkelmethode
 - o RUP
 - o Waterval
- Proof of concept
- Technieken
 - o Requirementsanalyse
 - o UML
 - o SCSS
 - o MoSCoW
 - o JetBrains PhpStorm
 - o Compass
 - o PHP
 - o HTML
 - o JQuery
 - o Handlebars
 - o CakePHP V2.8
 - o XAMPP

Voorwoord

Het procesverslag is een verslag dat het afstudeerproces van de stagiair, D.A.J van der Ham bij Didacticum op de afdeling Zwoozi. Dit verslag heeft betrekking op het afstudeertraject bij Zwoozi. De opdracht vindt u bij de opdrachtoomschrijving in het procesverslag.

Hierbij wil ik de opdrachtgever en producteigenaar, Nick Verheij bedanken voor de input die hij voor de opdracht heeft geleverd en voor het geven van de afstudeeropdracht. Verder wil ik mijn stagebegeleider, Florian de Groot en Lead Developer Koen Brouwer bedanken voor de ondersteuning die zij hebben geleverd tijdens het project.

Als laatst wil ik mijn familie bedanken voor het geven van steun en doorzettingsvermogen.

Samenvatting

Het project dat uitgevoerd wordt heeft betrekking op het realiseren van een centenveilingsysteem voor het veilingplatform van Zwoози. Het centenveilingsysteem is voortgekomen uit het idee die de opdrachtgever heeft gekregen na het bezoeken van 123bied.nl. Het centenveilingsysteem zal het veilingplatform Zwoози uitbreiden met een nieuw verdienmodel en een extra soort veiling.

Bij centenveiling moet je betalen om te kunnen bieden waarbij de prijs van een product met één cent zal stijgen per bod. Het bieden wordt gedaan aan de hand van biedcredits. De biedcredits moeten de gebruiker aanschaffen.

Het systeem dat ontworpen en gerealiseerd wordt is een proof of concept van centenveilingen omdat de opdrachtgever nog niet wist hoe hij dit moest realiseren.

Bij het ontwerpen zijn eerst de eisen en wensen van het systeem in kaart gebracht. De eisen en wensen hebben betrekking op de processen die moeten worden uitgevoerd door de gebruikers. De opdrachtgever wil met het systeem meer premiumaccounts gaan verkopen en ook meer credits. Verder wil hij de omzet van het veilingplatform vergroten.

De applicatie is ontworpen aan de hand van de RUP-methodiek (Rational Unified Process). Het proof of concept is gebouwd in het framework CakePHP en bevindt zich binnen de ontwikkelomgeving van Zwoози op een aparte branche binnen git.

Uiteindelijk zijn er voor de oplevering unittesten en een functionele acceptatietest uitgevoerd om de werking en de staat van het systeem in kaart te kunnen stellen. De applicatie is aan het einde van het proces werkend opgeleverd aan de opdrachtgever.

Tijdens het project heb ik de competenties aangetoond voor het analyseren door middel van requirements, het opstellen van een gegevensmodel voor de database en het bouwen van de applicatie.

Inhoudsopgave

REFERAAT	0
VOORWOORD	1
SAMENVATTING	2
INLEIDING.....	5
1. HET AFSTUDEERBEDRIJF.....	6
2. SITUATIEOMSCHRIJVING	7
3. GEBRUIKTE TECHNIEKEN EN METHODEN	8
3.1. Ontwikkelmethodiek keuze	8
3.2. Techniekkeuze.....	10
4. PLAN VAN AANPAK.....	12
4.1. Doelstelling.....	12
4.2. Op te leveren resultaat/resultaten	12
4.3. Afbakening	12
4.4. Programma van Eisen	12
4.5. Risicoanalyse	14
4.6. Fasering en planning van tussenresultaten.....	14
4.7. Projectbeheersing.....	14
4.8. Iteratieplanning.....	16
5. ANALYSEFASE.....	17
6. OVERSTAP VAN ONTWIKKELMETHODIEK	23
7. ELABORATIEFASE	24
7.1. Iteratie 1.....	24
7.2. Ontwerpen van sequentiediagrammen	26
7.3. Iteratie 2.....	28

8.	CONSTRUCTIEFASE : BOUWEN VAN DE APPLICATIE.....	34
8.1.	Iteratie 1 Doel:.....	34
8.2.	Iteratie 2.....	37
8.3.	Iteratie 3.....	40
9.	EVALUATIE	43
9.1.	Procesevaluatie	43
9.2.	Productevaluatie	44
9.3.	Competentieevaluatie	44
10.	BIBLIOGRAFIE	46
11.	BIJLAGEN.....	47
11.1.	Bijlage 1: Mailcontact met kansspelautoriteit.....	47
11.2.	Bijlage 2: Visie- en scoperequirementsdocument	47
11.3.	Bijlage 3: Softwarerequirementsspecificatie document	47
11.4.	Bijlage 4: Analysedisciplinedocument	47
11.5.	Bijlage 5: Designdisciplinedocument.....	47
11.6.	Bijlage 6: Testrapport.....	47

Inleiding

Dit document betreft de verslaglegging van het proces tijdens de afstudeerperiode van afstudeerder Daveron van der Ham. De afstudeerperiode heeft zich plaatsgevonden bij Didacticum onder de afdeling Zwoози. De periode heeft plaatsgevonden van 25 mei t/m 3 oktober. De procesbeschrijving geeft aan welke activiteiten zijn uitgevoerd tijdens het afstuderen welke beslissingen er zijn genomen gedurende de activiteit.

Dit document geeft weer wat er tijdens het project is uitgevoerd, hoe dit verlopen is en wat het resultaat ervan was. In het verslag zijn de beslismomenten opgenomen en worden alternatieven besproken.

Het eerste deel van het verslag gaat over de oriëntatie van het project. In de oriëntatie wordt de organisatie van het afstudeerbedrijf aangegeven en wordt de afstudeeropdracht behandeld. Na het behandelen van de afstudeeropdracht worden de methodieken besproken die ik bij het project heb toegepast om het project in goede banen te leiden.

Het tweede deel bevat de fasering van het project aan de hand van de fasen die aanwezig zijn in de gekozen projectontwikkelmethodieken.

Het derde deel bestaat uit de evaluatie van het proces, de opgeleverde producten en de competenties. In de evaluatie van het proces wordt weergegeven hoe het proces in zijn geheel is verlopen. De evaluatie van de opgeleverde producten evalueert over de totstandkoming van de producten. De competentie evaluatie evalueert of er aan de competenties is voldaan.

Aan het einde van het document bevindt zich de literatuurlijst waar de gebruikte bronnen vermeld staan. Als laatste zijn de belagen inbegrepen.

1. Het afstudeerbedrijf

Didacticum is een opleider en ICT dienstverlener gericht op het migreren, beheren en optimaliseren van de ICT omgeving sinds 1999. Zij combineren kennisimplementatie en kennisoverdracht op dusdanige wijze dat het synergievoordelen oplevert. Maar naast kennis van ICT, kennen ze ook hun klanten, hun bedrijven en de branches waarin hun klanten werkzaam zijn. Die combinatie leidt snel, effectief en meetbaar tot betere en concretere oplossingen die bijdragen aan de langetermijndoelstellingen van hun klanten.

Voor Didacticum is ICT-kennis geen doel op zich maar een middel waarmee zij als ICT-partner bijdragen aan de langetermijndoelstellingen en winstcijfers van hun klanten. Met 70 hoog opgeleide ICT'ers van HBO+-niveau van diverse achtergronden is Didacticum groot genoeg om voldoende mensen in te zetten op omvangrijke ICT-trajecten en klein genoeg om de wendbaarheid te hebben die men van een middelgrote organisatie mag verwachten.

Hun missie is het verschil maken voor hun opdrachtgevers. Zij zijn ervan overtuigd dat door een optimale inzet van ICT (mens, kennis én structuur) een organisatie beter, sneller, efficiënter én meer flexibel kan werken aan het behalen van bedrijfsdoelstellingen.

Hun consultants zorgen voor synergievoordelen door implementatie en overdracht van kennis naadloos op elkaar te laten aansluiten.

Bij hun Company Classes – de ICT-traineeship – komen de synergievoordelen optimaal tot hun recht: Met maatwerkopleidingen en praktijkervaring stomen zij jonge high potentials klaar voor een carrière bij onze klanten. zij overbruggen daarmee nadrukkelijk de kwalitatieve en kwantitatieve mismatch op de arbeidsmarkt van ICT-professionals.

Didacticum biedt haar diensten aan rondom 'de werkplek'. Als partner voor werken en samenwerken omarmen zij de trend 'cloud first/mobile first', en definiëren vier kennisgebieden waarbinnen zij zich specialiseren: **Cloud, Mobile, Social en Security**. Zij bieden snelle, effectieve en praktische oplossingen die daadwerkelijk bijdragen aan de klantdoelstellingen. Ze maken daarbij gebruik van Microsoft producten, zoals Enterprise Mobility Suite, Office 365 en Azure. Ze floreren in organisaties waar structurele werkplekautomatisering een must is. Diensten die ze aanbieden zijn detachering, opleidingen, company class en solutions.

Zwoozi is een startup die is opgezet vanuit Didacticum. Het is een project dat na het realiseren van de startup als zelfstandig bedrijf zal opereren. Het team dat werkt aan het project bevindt zich in een zelfstandig kantoor aan de Folkert Elsingastraat in Rotterdam. Zwoozi moet een marktplaatsachtige veilingsite worden waarmee zij een community van kopers en verkopers willen creëren.

2. Situatieomschrijving

De afdeling Development van Didacticum is bezig met het opzetten van een veilingplatform genaamd Zwooz. Er is een klein team gevormd dat werkt aan het opzetten van het platform, van het idee tot aan de realisatie.

Zwooz wordt een veilingplatform waar men oude producten kwijt kan raken door ze aan andere gebruikers te veilen. Dit kunnen zowel nieuwe als tweedehands producten zijn. Het betalen van een veiling gebeurt niet met geld, maar aan de hand van credits. Gebruikers verdienen credits voor het veilen van een product. De credits die zij verdienen kunnen zij in het gevolg weer gebruiken om nieuwe producten aan te schaffen. Zo wil het Zwooz-team een community creëren waarbij men continu elkaar de producten kan verkopen zonder daar veel geld voor uit te geven. Het idee is gebaseerd op het Listia, een Amerikaanse veilingssite die een grote community heeft weten op te bouwen.

Het Zwooz-platform zal bestaan uit verschillende onderdelen. Veilingen, een shop en een gebruikerssysteem. Eén van de nog te ontwikkelen onderdelen is een veilingssysteem waar Zwooz gesponsorde producten van leveranciers kan veilen. Het betreft een centenveilingssysteem.

Nu volgt een korte uitleg wat een centenveiling is. *“Een centenveiling is een veilingconcept waarbij elk bod een vast bedrag is – een cent. Om te mogen bieden, moet je betalen, meestal in de orde van 1 à 2 euro. Na elk bod wordt de eindtijd van de veiling verlengd zodat andere deelnemers ook weer kunnen bieden. Voor de bidders is echter niet inzichtelijk hoeveel bidders er zijn en hoeveel die bidders al hebben geboden. Het is moeilijk te voorspellen wanneer andere bidders niet meer zullen bieden. Verder is het niet mogelijk op enige wijze behendigheid of kennis toe te passen bij het concept van goudkoortsveiling.”*

- [HTTPS://gamerecht.nl/kansspelen/wanneer-is-een-centenveiling-legaal/](https://gamerecht.nl/kansspelen/wanneer-is-een-centenveiling-legaal/)

Het centenveilingssysteem zal het huidige platform uitbreiden en zorgt ervoor dat het systeem als een extra verdienmodel voor de organisatie dient. Het systeem sluit uiteindelijk aan op de andere systemen.

Het platform Zwooz bevindt zich in een bèta stadium. Dit betekent dat het platform nog volop in de ontwikkelingsfase is. Ook de andere onderdelen van Zwooz zijn volop in ontwikkeling. Centenveilingen is pas later in het ontwikkelproces bedacht door de opdrachtgever. De toevoeging van centenveiling zal een grote impact hebben op Zwooz wanneer het gelanceerd wordt. Centenveilingen zijn interessant voor het platform, omdat het zal zorgen voor een extra inkomstenbron en een extra type veiling aanbied voor de gebruikers. De opdrachtgever heeft echter nog niet goed nagedacht over hoe zij het systeem moeten opzetten, ontwerpen en realiseren. Voor het ontwerpen en bouwen van het systeem hebben zij het project aan mij uitbesteed. Het centenveilingssysteem zal nog niet gelanceerd worden totdat Zwooz een groot genoeg userbase heeft, om verlies te voorkomen. Zij hebben aan mij gevraagd om onderzoek te doen naar het concept van centenveilingen en een proof of concept te ontwikkelen.

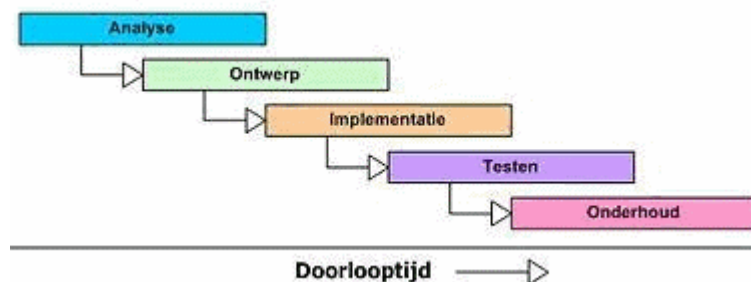
Het onderzoek naar het concept vindt plaats in de requirementsanalysefase voor het proof of concept. Deze requirements dienen als input om het ontwerp van het proof of concept te maken. Na het ontwerpen wordt het systeem gebouwd en getest. Na afloop wordt het proof of concept opgeleverd.

3. Gebruikte technieken en methoden

3.1. Ontwikkelmethodiek keuze

Het ontwikkelen van systemen kan worden versneld wanneer men zich houdt aan een ontwikkelmethodiek. Ontwikkelmethodieken dragen bij aan de kwaliteit van een systeem. Om de kwaliteit van dit project in het team van Zwoozi van voldoende kwaliteit te voorzien kies ik in eerste instantie voor de ontwikkelmethode Waterval. De selectiecriteria van die methodiek is de volwassenheid van het projectteam van Zwoozi. Dit team bestaat nog niet lang en is daardoor als organisatie onvolwassen. De waterval methodiek is geschikt voor nieuwe en onvolwassen organisaties (W-mdr90, 2012).

Waterval

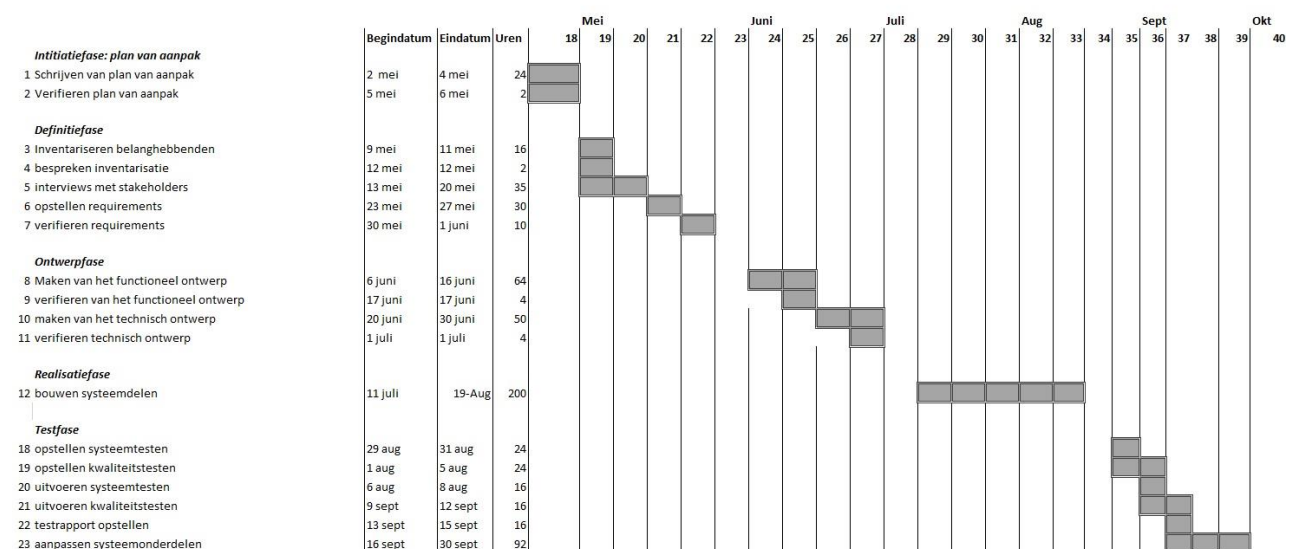


Figuur 1. Weergave van de watervalfasering.

Waterval is een lineaire methode die werkt aan de hand van aflopende fasen. Fasen worden eerst volledig afgerond voordat het ontwikkelteam doorgaat naar de volgende. In de bovenstaande figuur worden de volgende 5 fasen genoemd: Analyse, Ontwerp, Implementatie, Testen en Onderhoud. Voor de analyse wordt een plan van aanpak en de requirementsdocumentatie opgesteld. Voor het ontwerp wordt een functioneel en een technisch ontwerp gemaakt. Bij de implementatie wordt het systeem gebouwd en in het gevolg getest.

Deze methodiek kent enkele gevaren. Fasen worden volledig afgerond voordat men naar de volgende fasen gaat en kan er niet meer op de vorige fase worden teruggevallen. De methodiek is daardoor vatbaar voor tussentijdse veranderingen. Wanneer deze veranderingen tijdens het project optreden, verhoogd dit de kans dat er een verkeerde applicatie wordt gebouwd die niet meer aan de eisen en wensen van de opdrachtgever voldoen. Een eis van deze methodiek is dat de opdrachtgever de requirements van het project goed kent. In figuur 2 wordt de initiële planning van het project weergegeven volgens de waterval methodiek. Voor meer informatie, zie (W-mdr90, 2012).

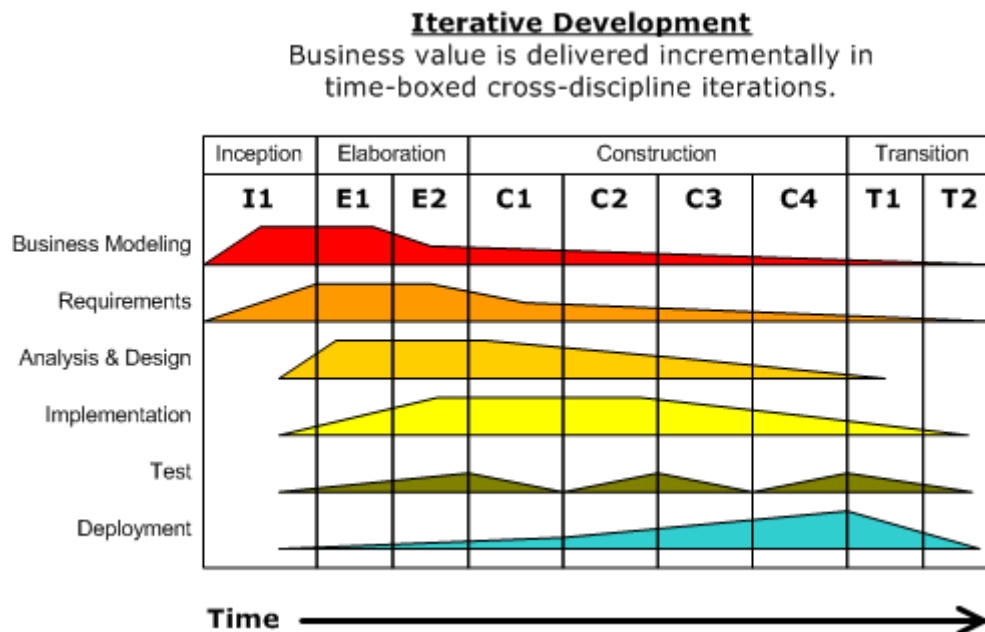
Balkenplanning centenveilingsysteem Zwoozi



Figuur 2. Balkenplanning voor de waterval methodiek

Tijdens het project stap ik echter over naar de methodiek RUP (Rational Unified Proces). Dit is het gevolg van het uitbreiden van het team met een Lead Developer. Door zijn komst treden er veranderingen van requirements op tijdens het project. Om de requirements beter te kunnen beheren en risico's in het project te verkleinen heb ik de overstap moeten maken.

Rational Unified Process (RUP)



Figuur 3. Fasering van de RUP-methode.

RUP is een ontwikkelmethodiek die werkt met iteraties. RUP bevindt zich tussen de watervalmethode en iteratieve methoden en maakt gebruik van vele best practices uit andere methodieken. Een iteratie is een vaste cyclus van 2 weken waarin het team aan de verschillende producten van het project kan werken. Zoals in afbeelding 2 hierboven is aangegeven lopen de producten (links) en iteraties (boven) parallel. Zo begin je bij RUP met het definiëren van globale requirements en werk je de producten over de verschillende iteraties uit tot een gedetailleerde versies.

RUP biedt het voordeel dat de producten in iedere iteratie van het proces kunnen worden gewijzigd door nieuwe inzichten. Het project is op deze wijze niet vatbaar voor tussentijdse veranderingen. De veranderingen hebben een grote impact op de requirements, ontwerp en realisatie van het systeem en zorgen dat deze overeen blijven komen met de huidige eisen en wensen van de opdrachtgever. Het nadeel is dat het onderhouden van de documenten veel moeite en tijd kost om de consistentie te behouden, vooral als je alleen aan het project werkt. Hoewel RUP beter werkt in een groter projectteam, heb ik alleen de wijzigingen door te voeren in documenten die ik zelf heb gemaakt. Een verandering in requirements kan immers veranderingen met zich meebrengen in het ontwerp en een gerealiseerd stuk software (als deze al is gebouwd). Zie (Ambler, 2001-2012) voor meer informatie over RUP.

Een andere mogelijkheid is de agile ontwikkelmethodiek SCRUM. SCRUM wil snel werkende stukken software opleveren om snel de applicatie te kunnen bouwen. Zij leggen de nadruk op user stories en willen zo min mogelijk documentatie schrijven. Voor dit project is documentatie voor het project echter belangrijk omdat er een gebrek is aan documentatie. Zonder de documentatie kunnen toekomstige nieuwe werknemers niet direct aan de slag. Daarom heb ik de ontwikkelmethodiek RUP boven SCRUM verkozen.

3.2. Techniekkeuze

Gedurende het project maak ik gebruik van verschillende technieken. In deze paragraaf licht ik deze toe.

Requirementstechnieken:

Voor de technieken die betrekking hebben tot het analyseren van requirements, maak ik gebruik van enkele methoden uit het (Swart, 2010). Het boek beschrijft vele technieken om requirements te achterhalen, te beheren en beter te specificeren. De technieken die ik uit het boek heb gebruikt zijn:

- Identificeren van belanghebbenden
- Eliciteren door middel van het houden van interviews
- Analyseren door middel van prioriteren en categoriseren
- Specificeren door middel van tekst structureren
- Valideren door middel van reviewen en doorspreken

Deze technieken heb ik gedurende mijn opleiding tijdens projecten moeten toepassen.

UML: (OMG, 2005)

UML (Unified Modeling Language) is een modelering taal die helpt bij het weergeven van een simpele duidelijke weergave van situaties en/of systemen. UML biedt voordelen rondom het vormgeven van informatie in de vorm van modellen en wordt veel gebruikt bij ICT-bedrijven. UML wordt gebruikt om vele verschillende diagrammen vorm te geven. Deze diagrammen zijn: Use Case modellen, klassendiagrammen, sequentiediagrammen, staaldiagrammen, activiteidiagrammen enz. Deze manier van modeleren gebruik ik bij het ontwerpen van klassendiagrammen, staaldiagrammen, Use Cases modellen en Use Cases flows. UML geeft een globaal overzicht weer door middel van modellen die door verschillende belanghebbenden kunnen worden begrepen. UML wordt binnen het bedrijf gebruikt om verschillende diagrammen te maken.

SCSS: : (hampton Catlin, 2006-2015)

SCSS (Sass Cascading Style Sheet) is een laag bovenop CSS die meer flexibiliteit biedt bij het schrijven van stylesheets. De extra laag biedt ondersteuning aan voor het overerven van CSS attributen, het gebruiken van variabelen om CSS dynamischer te maken en het nesten van CSS elementen. SCSS wordt gebruikt voor het stylen van het platform Zwoози. Voor een goede integratie met het platform en de andere systemen zal bij het realiseren van centenevelingen gebruik worden gemaakt van SCSS. De extra laag biedt de mogelijkheid om CSS bestanden beter te kunnen beheren en SCSS bestanden moet echter eerst worden gecompileerd naar CSS voordat zij kunnen worden gebruikt. Voor het converteren van SCSS bestanden naar CSS heb ik gebruik gemaakt van Compass.

Compass: (Compass, 2016)

Compass is een command-line tool dat gebruikt wordt om SCSS bestanden te compilen naar CSS bestanden. Compass heeft een volledige integratie met JetBrains PhpStorm waardoor het compilen automatisch gebeurt na elke verandering.

XAMPP: (Apache, 2016)

XAMPP is een virtueel hosting programma van Apache. Het programma zet een lokale webserver op. XAMPP wordt gebruikt om lokaal PHP bestanden te kunnen ontwikkelen en testen. XAMPP is uitgerust met een MySQL database voor dataopslag. Ik gebruik deze virtuele hosting programma om een lokale ontwikkel- en testomgeving op te zetten.

JetBrains PhpStorm: (Jetbrains, 2016)

PhpStorm is een IDE (Integrated Development Environment) dat voornamelijk wordt gebruikt om PHP, HTML, JavaScript en CSS bestanden te kunnen schrijven. De IDE ondersteunt de applicatiestructuur van webapplicaties en zorgt dat je binnen de applicatie kan navigeren naar andere bestanden. PhpStorm bevat vele macro's die automatisch templates voor bestanden als jQuery kan opzetten, de code te structureren enz. binnen PhpStorm maak ik gebruik van .git om de bestanden onder versiebeheer te behouden, Compass om de SCSS bestanden te compilen, JQuery

Navicat Premium: (Navicat, 2016)

Navicat is een databaseadministratietool die de mogelijkheid biedt om verbinding te leggen met zes verschillende soorten databases binnen één applicatie. Navicat ondersteunt MySQL; MariaDB, SQL Server, Oracle en PostgreSQL. Navicat biedt een snelle en makkelijke toegang tot alle databses in één omslag.

PHP: (PHP, 2016)

PHP is een server-side HTML embedded scripting language dat wordt gebruikt om dynamische websites te bouwen. PHP is de meest gebruikte programmeertaal om webapplicaties mee te bouwen. PHP scripts worden altijd server-sided uitgevoerd en biedt vele mogelijkheden om je applicatie te verbinden met verschillende databases. Het bevat vele tools zoals een API voor het leggen van verbindingen met Apache.

jQuery: (jQuery, 2016)

jQuery is een snelle, kleine functioneelrijke Javascript library. jQuery maakt het makkelijker om HTML-bestanden te doorzoeken en te manipuleren, events af te handelen, animaties weer te geven en AJAX (Asynchronous JavaScript and XML) te hanteren. jQuery werkt op een grote hoeveelheid webbrowsers. Ik maak gebruik van jQuery om javascriptcode te schrijven die werkt over de verschillende browsers heen. Verder maak ik er ook gebruik van om makkelijker javascript te schrijven.

Handlebars: (Handlebars, 2016)

Handlebars is een javascript library die semantische templates kan opbouwen. De HTML code wordt hierbij gescheiden in een aparte HTML-bestand. Handlesbars haalt de template op en compileert deze om waarden aan toe te kunnen voegen. De locatie waar hij de waarden toevoegt wordt aangegeven door {{waarde}}. Ik gebruik handlebars om door middel van javascript HTML-elementen toe te kunnen voegen aan de webpagina in realtime.

4. Plan van aanpak

4.1. Doelstelling

Het algemene doel voor organisatie is het zo snel mogelijk realiseren van Zwoozi zodat deze aan het einde van het jaar in een live omgeving kan gaan draaien. De doelstelling die de opdrachtgever wil halen met dit project is om het veiling platform Zwoozi uit te breiden met centenveilingen, zodat het systeem dicht bij een 1.0 versie komt. Deze type veilingen moeten een extra verdienmodel worden voor het systeem en moet het gebruik van de premium accounts promoten zodat er meer kredieten worden aangeschaft.

4.2. Op te leveren resultaat/resultaten

De resultaat van het project is een systeem dat centenveilingen kan regelen. Dit systeem zal een deel gaan uitmaken van het platform van Zwoozi. Het systeem zal de beschikking moeten hebben op het aanschaffen van kredieten. Doordat het bieden zelf ook kredieten kost, moet dit leiden tot een toename in het aanschaffen van kredieten voor premium accounts. Het systeem moet verder te gebruiken zijn door gebruikers van Zwoozi. Zwoozi moet de mogelijkheid hebben om zelf producten te veilen via een centenveiling. Het systeem moet voldoen aan het verdienmodel die door de opdrachtgevers tijdens dit project zal worden vastgesteld. Om directe verlies bij de implementatie tegen te gaan, zal het systeem pas gaan draaien op het moment dat er een x aantal gebruikers aanwezig zijn (het aantal gebruikers wordt later bepaald).

Hieronder staan de resultaten die de opdrachtgevers uit het project willen zien:

- Requirementsdocumentatie:
Hierin zijn o.a. de eisen, wensen wetten en regelgeving terug te zien die van pas zullen zijn bij centenveilingen. Verder worden hier ook de eisen en wensen van de belanghebbenden van Zwoozi geïdentificeerd.
- Ontwerp documentatie:
Onder ontwerpdocumentatie vallen het analyse discipline en design discipline document. Het analyse discipline document bevat de globale weergave van het systeem. De analyse van het systeem wordt aangegeven door middel van analyse pakket diagrammen, analyseklassendiagrammen en Use Case-realisaties.
Het design discipline document werkt de Analyse discipline uit naar een gedetailleerde versie, waar wordt bepaald welke onderdelen tot de architecturale baseline toebehoren. De grafische user interface wordt hier ook weergegeven.
- Testrapportage:
De testrapportage geeft de resultaten weer van de testen die zich bij het realiseren van het systeem hebben plaatsgevonden en geeft ook de staat van het systeem aan.

4.3. Afbakening

Dit project is gericht op het systeem van centenveilingen, zoals het ontwerpen realiseren en testen ervan. Hoewel er voor het platform van Zwoozi nog vele functionaliteiten en processen op tafel liggen die van belang zijn voor het realiseren van Zwoozi, zullen deze buiten de scope van dit project vallen.

4.4. Programma van Eisen

Bij het project zijn er een aantal eisen die aan te pas komen, of in dit geval de eisen waaraan het systeem en zijn documentatie aan moeten voldoen.

4.4.1. Randvoorwaarden

- Het systeem moet komen te draaien op de webserver van Zwoozi. De webserver van Zwoozi draait op een service host van Neostrada (www.neostrada.nl)
- Het systeem moet worden gebouwd met het PHP framework genaamd CakePHP. Het platform van Zwoozi maakt ook gebruik van dit framework.
- Het systeem moet via het platform van Zwoozi te bereiken zijn door cliënten die over een webbrowser beschikken (o.a. de bekende webbrowsers als Google Chrome, Safari, Internet Explorer, Mozilla Firefox etc.).
- Bij het ontwerpen van het systeem moet er rekening worden gehouden met het hergebruik, testbaarheid en wijzigbaarheid ervan. De systeemdelen moeten op de juiste wijze worden opgebouwd en werkend worden opgeleverd door middel van het gebruik van programmeerstandaarden.
- Bij het ontwikkelen en testen van het systeem moet er gebruik worden gemaakt van een ontwikkel- en testomgeving samen met een versiebeheertool om zowel het ontwikkelen en het testen te bevorderen.
- Het systeem moet zich houden aan de wetten en regels die bij kansspelen aan te pas komen, om te zorgen dat het systeem legaal is. Deze moeten terugkomen in de requirementsdocumentatie.
- Het systeem moet dezelfde huisstijl aanhouden als het platform van Zwoozi voor een vloeiende overloop wanneer het systeem wordt geïmplementeerd in Zwoozi.
- De requirements in de requirementsdocumentatie moeten volledig, dekkend, ondubbelzinnig en duidelijk worden opgesteld, zodat deze makkelijk te begrijpen zijn door de andere belanghebbenden.
- De gegevensmodel van de database moet worden opgesteld, waarbij de juiste objecten uit het domein worden gekozen. De beschrijvingen van deze objecten moeten volledig, duidelijk en consistent het beeld weergeven van het domein en de onderliggende samenhang tussen de objecten moet juist worden vastgelegd. Het gegevensmodel moet toetsbaar zijn op duidelijkheid, volledigheid en consistentie.

4.4.2. Functionele eisen

- Het systeem moet de gebruiker voldoende inzicht bieden in de centenveilingen om het gevoel van fraude te voorkomen.
- Om het aanschaffen van kredieten te bevorderen moet het systeem de deelnemers belonen door middel van korting bij een participatie van een centenveiling.
- Bij het ontwikkelen van het systeem moet er rekening worden gehouden dat de centenveilingen actueel te volgen moeten zijn en dat de status ervan actueel moet worden geüpdatet.

4.4.3. Niet functionele eisen

- De requirements documentatie moet goed leesbaar zijn voor de belanghebbenden van het project, zodat zij makkelijk inzicht kunnen krijgen in de eisen, wensen, wetten en regelgeving die bij het systeem van belang zijn. De documentatie moet potentiële nieuwe ontwikkelaars bij Didacticum en Zwoozi wegwijs maken voordat zij aan het systeem kunnen werken.

4.5. Risicoanalyse

Deze paragraaf geeft de risico's weer die bij het project aanwezig zijn.

Risico	Maatregel	Verantwoordelijk
Gebrek aan eindgebruikers: Voor het project is de gebruikersgroep wel geïdentificeerd, maar is er nog geen mogelijkheid om contact met enkele gebruikers binnen deze gebruikersgroep. Hierdoor kan er een gebrek aan gebruiker requirements komen doordat er niet goed in kaart kan worden gebracht wat de eindgebruikers nou vragen van het systeem.	Samen met de opdrachtgevers kunnen we contact leggen met enkele mogelijke eindgebruikers die vaker gebruik maken van online veilingen. Zij kunnen dan het gebrek aan requirements verhelpen door hun eisen en wensen aan ons door te geven.	Nick Verheij, Daveron van der Ham.
Regulering centenveilingen: Doordat een centenveiling wordt beschouwt als een kansspel, gelden er andere regels voor dan voor ene normale veiling. Zo is een centenveiling in eerste instantie verboden. Maar een centenveiling wordt wel uitgevoerd door instanties als www.123bied.nl .	Voor de regeling moeten wij een gesprek houden met de instelling kansspelautoriteit. Wij moeten in overleg met hun tot een overeenkomst komen om centenveiling toch voor Zwooz te legaliseren, al moeten er een aantal aanpassingen aan het systeemontwerp worden gedaan.	Nick Verheij, Daveron van der Ham

4.6. Fasering en planning van tussenresultaten

Fase	(Tussen)resultaten/ producten	Startdatum	Einddatum	Datum akkoord opdrachtgever
1. Initiatiefase	Plan van aanpak	25-04-2016	04-05-2016	
2. Inceptiefase	Requirementsdocumentatie	09-05-2016	27-05-2016	
3. Elaboratiefase	Analyse- en design discipline document.	06-06-2016	01-07-2016	
4. constructiefase	Centenveilingen systeem	11-07-2016	19-08-2016	
5 Testfase	testrapport	29-08-2016	09-09-2016	
5. Implementatie/ nazorg/evaluatie		19-09-2016	30-09-2016	

4.7. Projectbeheersing

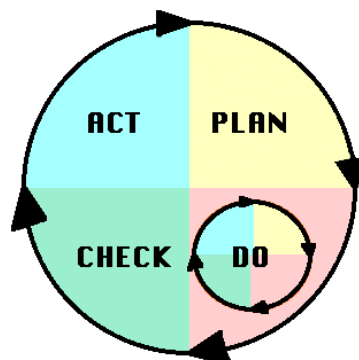
4.7.1. Tijd

Er wordt 2 keer per week een gesprek gehouden met de begeleider van dit project om de voortgang van het project te controleren. Op basis van deze gesprekken kan er worden achterhaald of de planning goed wordt gevolgd. Deze activiteitenplanning wijst een aantal dagen aan een activiteit toe. Dit betekent niet dat deze activiteiten niet altijd in volgorde worden uitgevoerd, maar dat de activiteiten overloopt met elkaar kunnen hebben. Door middel van de uitloophases kunnen er extra dagen aan de activiteiten worden gewerkt.

Het project verloopt verder door middel van iteraties. Deze iteraties hebben ene lengte van 2 weken. Aan het einde van de iteraties wordt er aan de hand van een iteratierapport vastgelegd hoe de iteratie is verlopen en wat er in de volgend iteratie moet worden gehanteerd.

4.7.2. Kwaliteit

Om de kwaliteit van het project te beheersen zal er gebruik worden gemaakt van de Demingwheelmethode, doordat deze techniek in de praktijk veel wordt gebruikt. De methode maakt gebruik voor een iteratieve aanpak. Voor elke activiteit in de activiteitenplanning zal er een korte planning worden gemaakt (Plan). Daarna worden de werkzaamheden van de activiteit uitgevoerd (Do). Vervolgens kunnen we nagaan of de kwaliteit van de resultaten van de activiteit voldoende is (Check). Vervolgens kan er daarop actie worden ondernomen om de kwaliteit te verhogen (Act).



Figuur 4. Demingwheerl

Verder zullen er korte tussentijdse presentaties aan de organisatie worden gegeven zodat de organisatie samen de producten kunnen evalueren. De informatie die in de documentatie komt te staan zoals requirements en de ontwerpen, zullen SMART moeten worden geformuleerd. Dit maakt de producten meetbaar voor kwaliteit bij de Checkfase uit de Demingwheelmethode.

4.7.3. Organisatie

De organisatie die bij het project van toepassing komen zijn in het schema hieronder opgenomen

Wie	Functie	Verantwoordelijkheid	Bevoegdheid
Nick Verheij	Opdrachtgever Producteigenaar	Tijdige communicatie, tijdig akkoord op Tussenresultaten Tijdig akkoord op tussenresultaten	goedkeuren van tussenresultaten
Koen Brouwer (werkzaam vanaf 11-07-2016)	Opdrachtgever Producteigenaar, Projectbegeleider Lead Developer	Tijdige communicatie, tijdig akkoord op tussenresultaten	goedkeuren van tussenresultaten
Florian de Groot	Projectbegeleider Developer	Tijdig begeleiding project Tijdige communicatie Tijdig akkoord op tussenresultaten	X
Daveron van der Ham (Stagiair)	Ontwikkelaar	Tijdige oplevering tussenresultaten Tijdig communicatie	X

4.7.4. Informatie

Met de projectbegeleider en andere ontwikkelaars communiceren we via de app 'Telegram' of via de mail. Verder houden we elke week een projectoverleg om de voortgang van het project te bespreken. Het systeem van centenveilingen komt onder de versiebeheertool Git te staan. De documenten van het project zullen bewaard worden in Dropbox met elk een eigen versie nummer waarbij ook de vorige versies worden bewaard. Tijdig zullen deze documenten naar de projectbegeleider worden gestuurd voor controle.

Wie	Akkoord geven	Advies geven	Op de hoogte zijn
Florian de Groot	X	X	X
Nick Verheij	X	X	X
Koen Brouwer	X	X	X
Daveron van der Ham		X	X

4.8. Iteratieplanning

De iteratieplanning worden aan het begin van de iteraties in de volgende hoofdstukken besproken. Hierbij wordt tin het kort aangegeven wat de doelen zijn van deze iteratie. Aan het einde van elke iteratie wordt aangegeven welke doelen behaald zijn.

5. Analysefase

De analysefase brengt de eisen van het te ontwikkelen systeem in kaart. Projectdoelen worden gespecificeerd, stakeholders worden geanalyseerd, requirements worden achterhaald en opgesteld. Bij deze fase heb ik twee documenten gemaakt:

- Visie en scope
- Requirements-specificatie

De visie en scope document bevat de projectdoelen, de specificatie van de stakeholders en een globale set aan requirements. De requirements-specificatie document bevat de specifieke requirements aan de hand van Use Cases.

5.1.1. Achterhalen van projectdoelen

Met zijn bedrijfsvisie wil de opdrachtgever verbeteringen zien in de business van het Zwoozzi platform. De voornaamste reden waarom centenveilingen ontwikkeld moet worden, is om het platform uit te breiden met een extra veilingmogelijkheid. De extra mogelijkheid is direct ook een nieuw verdienmodel. Er bestaan al enkele online centenveilingen, maar Zwoozzi wil de klanten van het systeem meer voordelen aanbieden tegenover de andere online centenveilingen.

De business requirements die de opdrachtgever heeft aangegeven zijn:

- de opdrachtgever wil de omzet verhogen met 25%.
- de opdrachtgever wil het aanschaffen van kredieten verhogen met 15%
- de opdrachtgever wil het aantal premium accounts verhogen met 10%.
- de opdrachtgever wil de verkoopwaarde van producten die hij wil gaan veilen verhogen tegenover de inkoopprijs met een marge van 30%.
- de opdrachtgever wil verschillende type veilingen aanbieden.
- de opdrachtgever wil de gebruiker meer inzicht geven in het veiling proces tegenover andere online centenveilingen.
- de gebruikers willen producten tegen een lage prijs kunnen winnen.
- de opdrachtgever wil de gebruikers het gevoel geven dat er geen verliezers zijn.

De businessrequirements heb ik achterhaald door middel van technieken die beschreven staan in (Swart, 2010).

5.1.2. Identificeren van belanghebbenden

De belanghebbenden zijn geïdentificeerd door middel van de checklist-belanghebbenden uit (Swart, 2010). Het systeem heeft de volgende belanghebbenden:

- Consumenten (via internet)
- Medewerkers
- Kansspelautoriteit
- Directie
- Lead Developer
- Concurrent 123bied.nl

De consumenten zullen via het internet in contact komen met het systeem. Zij zijn de eindegebruikers van het systeem en zitten vooral aan de front-end. Medewerkers zijn de interne medewerkers van Zwoози. Ook zij zijn eindgebruikers bij het systeem maar zullen vooral via de backend gebruik maken van het systeem. Kansspelautoriteit is de instelling die verantwoordelijk is voor de vergunningen van centenveilingen. Zij houden illegale kansspelen in de gaten en halen ze neer als er teveel negatieve meldingen over zijn. De directie zijn de investeerders van het project. Zij willen het resultaat terugzien die de opdrachtgever bij de doelen (paragraaf 3.2) zijn benoemd. De Lead Developer onderhoudt het platform waar centenveilingen zich in vestigt. Deze zal voorlopig nog niet bij het team aanwezig zijn en kunnen daardoor nog geen requirements van hem vandaan worden gehaald. Concurrent 123bied.nl is één van de actieve centenveilingen rondom Nederland en België. De opdrachtgever heeft aangegeven dat ik veel requirements bij hun systeem vandaan kan halen. De geïdentificeerde belanghebbenden zijn geverifieerd met de opdrachtgever.

5.1.3. Gesprekken houden met belanghebbenden

De belanghebbenden medewerkers, directie en applicatiebeheerder bevinden zich intern in de organisatie. Consumenten, kansspelautoriteit en 123bied.nl bevinden zich extern. Kansspelautoriteit is een instelling die zich bezighoudt met het autoriseren van kansspelen. Bij het uitzoeken naar het centenveiling concept, werd aangegeven dat deze instelling zich bezig houdt met kansspelen in Nederland. Ik heb deze instelling via de mail benaderd omdat zij meer weten over centenveilingen. Kansspelautoriteit vindt dat centenveilingen kansspelen is doordat men geen vat en inzicht heeft over het winnen van een centenveiling zie (bijlage 1). Zij verbieden kansspelen op het internet door gebrek aan fysieke deelname en controle. Aan de hand van deze verbodening wordt het systeem ingericht om de gebruiker meer inzicht te geven in het winnen, met als doel het systeem te legaliseren. Kansspelautoriteit weigerde mijn verzoek om een persoonlijk interview af te nemen. Zij wilden verder geen medewerking verlenen. In overleg met de directie is er besloten om ons gedurende het project niet druk te maken over kansspelautoriteit, omdat ik immers enkele belangrijke requirements uit het mailcontact heb kunnen achterhalen.

De interne stakeholders heb ik interviews mee gehouden om de requirements uit hun oogpunt te achterhalen. De medewerkers gaven aan dat zij hun processen graag geautomatiseerd zien. Zij willen niet elke centenveiling hoeven te onderhouden, maar willen wel inzicht hebben in hoe de centenveilingen verlopen.

De directie heeft aangegeven dat de werking van een centenveiling hetzelfde verloopt als bij 123bied.nl. Deze requirements heb ik achterhaald door deel te nemen aan 123bied.nl met de credits die ik gratis kreeg bij het registreren. Elk onderdeel op hun website heb ik geanalyseerd en omgezet naar requirements.

Zwoози heeft op dit moment nog geen applicatiebeheerder. Ik heb van deze belanghebbende in deze fase geen requirements kunnen achterhalen. Consumenten zijn externe belanghebbenden, maar de vertegenwoordiger is de marketingexpert Nick Verheij. Nick is ook onderdeel van de directie en heeft hetzelfde antwoord gegeven als de directie.

5.1.4. Requirements definiëren

De resultaten uit de gesprekken heb ik vertaald naar requirements. Het ontleden van de resultaten zorgde voor een kleine set aan requirements. Hierin bevonden zich een aantal globale requirements. De globale requirements staan beschreven in de visie en scope [zie bijlage 2].

De requirements zijn op te delen in softwarerequirements en gebruikerrequirements en bedrijfsregels. De gebruikerrequirements gevende eisen en wensen weer die zij in het systeem terug willen zien. De softwarerequirements geeft de eisen waaraan het systeem moet voldoen weer. Een gebruikerrequirement heeft een softwarerequirement als antwoord. De overige requirements zijn toegevoegd aan de softwarerequirementsspecificatie [zie bijlage 3].

De bedrijfsregels geven de regels weer die binnen de organisatie van toepassing zijn.

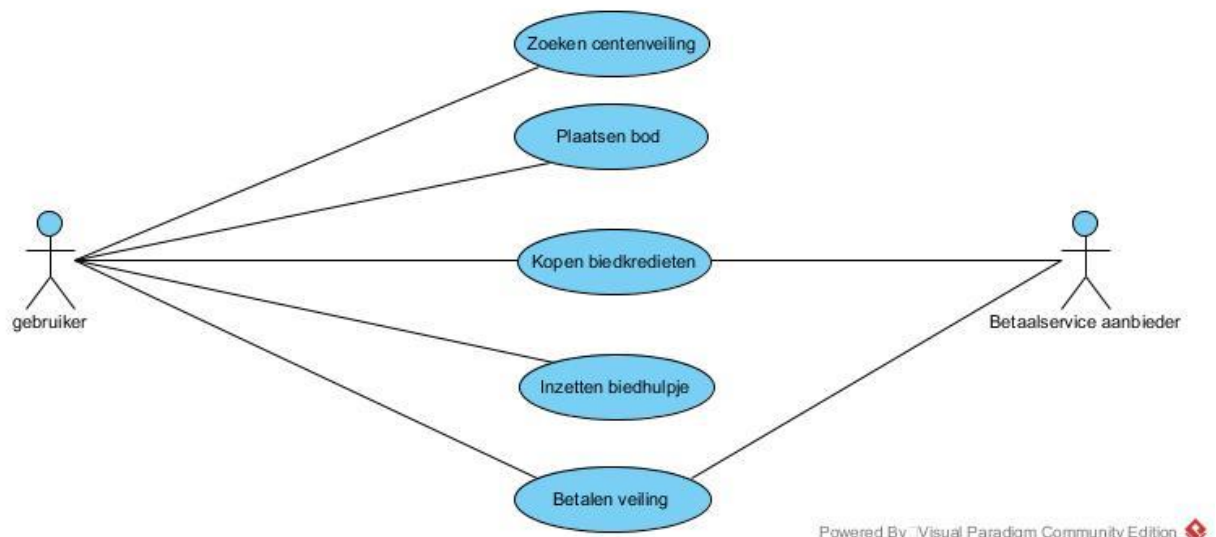
Voor de requirements gebruik ik de notatiewijze uit (Swart, 2010). Waarbij de requirements zijn opgebouwd als volgt:

- De gebruiker requirements staan genoteerd als “De [rol] wil [...]”.
- De softwarerequirements staan genoteerd als “Het systeem moet [...] doen”.

Naast deze requirements zijn als laatste de kwaliteitsattributen opgenomen in het document.

5.1.5. Use Cases opstellen

Aan de hand van de requirements heb ik een aantal Use Cases kunnen opstellen. Een Use Case is een groepering van requirements dat leidt tot een procesbeschrijving.



Figuur 5. Use Cases model met gebruiker

Use Cases geven processen aan die op elk moment door de bijbehorende actor kan worden uitgevoerd. Niet iedereen binnen de organisatie heeft een expertise met ICT. Om te zorgen dat alle leden van de organisatie de Use Cases begrijpen, maak ik gebruik van de notatiewijze uit (Swart, 2010). Deze notatiewijze is gericht om aan iedere lezer duidelijk te maken wat er met de beschrijving wordt bedoeld. In de requirements heb ik geen implementatie details opgenomen. De notatiewijze geeft de interactie tussen de actor en het systeem weer.

In figuur 6 is een voorbeeld van de notatiewijze binnen de Use Case beschrijving “Zoeken centenveiling” samen met een Use Case flow. De Use Case en Use Case flow modellen zijn gemaakt in UML om de lezers een duidelijk overzicht van de processen en hun stroming weer te geven. De UML modellen zijn in Visual Paradigm gemaakt.

Use Case naam	Zoeken centenveiling
Omschrijving	De gebruiker wil een centenveiling zoeken
Primaire actor	gebruiker
Secundaire actor	
Pre condities	De gebruiker is ingelogd.
Basisflow	1. De gebruiker geeft aan dat hij een centenveiling wil zoeken. 2. Het systeem vraagt naar de gewenste zoekcriteria. De gebruiker voert deze data in. 3. Het systeem zoekt de actuele centenveilingen op waarvan de titel of productomschrijving voldoen aan de zoekcriteria en toont deze aan de gebruiker.
Na condities	Er is naar een centenveiling gezocht.
Alternatieve flows	<u>A1: Geen resultaat:</u> In stap 3, als er geen centenveilingen gevonden worden die voldoen aan de zoekcriteria dan.... geeft het systeem weer dat er geen resultaten zijn gevonden. De gebruiker kiest of hij opnieuw wil zoeken. De Use Case stopt of gaat terug gaat naar stap 2.

Figuur 6. Use Case Zoeken centenveiling

5.1.6. Ontbrekende requirements achterhalen

Requirements waren nog niet duidelijk genoeg. Om aan deze requirements te komen, heb ik de interne belanghebbenden bij elkaar geroepen voor een brainstormsessie.

Met het samenkomen van deze belanghebbenden heb ik samen met de belanghebbenden kunnen overleggen welke requirements zij precies in het systeem wouden hebben. De requirements zijn beschreven op plakbriefjes en opgehangen op een whiteboard. Het opschrijven van de requirements op briefjes en deze vervolgens ophangen is afkomstig uit Scrum Op dit whiteboard hebben de requirements geprioriteerd op belang en hebben we een MoSCoW analyse erop uitgeoefend. De requirements hebben we geordend op 3 niveaus.

- Niveau 1: Bevat de kernfunctionaliteiten die het systeem op gang zullen brengen. Dit zijn kritische functionaliteiten voor een eerste versie.
- Niveau 2: Bevat de belangrijke functionaliteiten die niet direct in een eerste versie aanwezig moet zijn.
- Niveau 3: bevat extra functionaliteiten die in een later versie pas hoeft te worden gerealiseerd.

De MoSCoW analyse toont aan welke functionaliteiten belangrijk zijn voor het systeem. De afkorting van MoSCoW staat voor:

- Must have
- Should have
- Could have
- Won't have

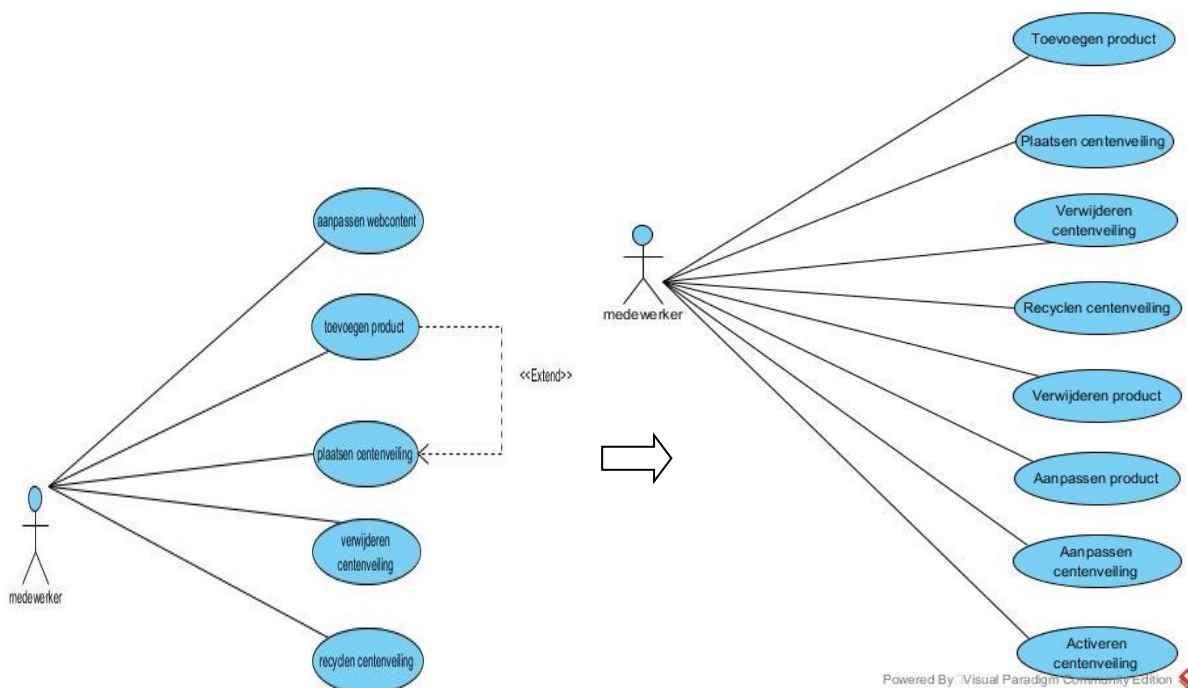
Op basis van de prioritering en de MoSCoW analyse heb ik verschillende sets van requirements kunnen definiëren. De functionaliteiten met een prioriteit van 1 zullen worden gerealiseerd voor het proof of concept.

5.1.7. Herdefiniëren van requirements

Aan de hand van de set requirements uit de brainstormsessie kunnen nieuwe requirements worden gedefinieerd. De oude requirements waren te globaal en konden niet duidelijk de werking van het systeem weergeven. De requirements ben ik volgens (Swart, 2010) opnieuw gaan beschrijven. Tussendoor heb ik de requirements voorgelegd aan de belanghebbenden voor feedback op de duidelijkheid van de requirements. De requirements die in de brainstormsessie waren echter kort opgeschreven. Deze hadden ten eerste verduidelijking nodig. Deze verduidelijking heb ik bij de desbetreffende belanghebbenden gevraagd en genoteerd. De verduidelijking heb ik vervolgens omgezet naar requirements en getoond aan de belanghebbenden.

Nadat de requirements voor de belanghebbenden duidelijk genoeg waren en ze een goed beeld gaven over wat het systeem moet gaan doen, heb ik met de opdrachtgever een aantal kwaliteitsattributen geselecteerd. Bij centeneilingen en het bieden erop is snelheid voor het bieden en actualiteit van de gegevens zeer belangrijk. Een gebruiker moet ten alle tijden snel en bod kunnen uitbrengen. De kwaliteitsattributen hebben betrekking op de snelheid dat een bod moet kunnen worden geplaatst en hoe actueel de gegevens moeten zijn. Immers moeten de tijd en nieuwe biedingen bij elke gebruiker worden geüpdatet. De snelheid van een systeem ligt echter lager als er een grote belasting aanwezig is door de vele hoeveelheid van actieve gebruikers. Samen hebben we bepaald hoe de snelheid en actualiteit van de gegevens moet zijn onder een piekbelasting van 1000 gebruikers.

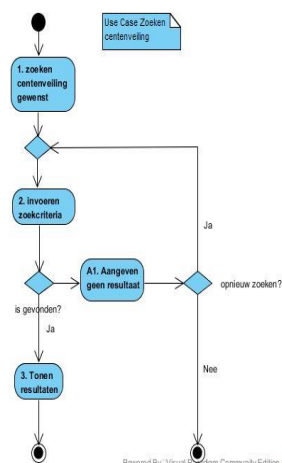
5.1.8. Use Cases verduidelijken



Figuur 7. Transactie van oude naar nieuwe Use Cases.

Op basis van de nieuwe lijst aan requirements is duidelijk geworden wat het systeem moet doen. Uit deze requirements heb ik de requirements geselecteerd die samen kunnen worden gebundeld tot een proces. Ik heb de oude Use Cases kunnen aanpassen en nieuwe Use Cases kunnen toevoegen. Per

Use Case ben ik Use Case flows gaan maken om de duidelijkheid van de Use Cases beter aan te kunnen geven. In figuur 8 is weergegeven hoe de Use Case flow van “Zoeken centenvelling” eruit ziet.



Figuur 8. Use Case Flow van zoeken centenvelling

De flows gaven de belanghebbenden duidelijkheid waar de beslismomenten in de processen aanwezig zijn. In combinatie met de geüpdatete Use Case omschrijvingen zijn de Use Cases duidelijker geworden voor de belanghebbenden. Ik heb de Use Cases samen met de requirementsdocumentatie gevalideerd met de belanghebbenden.

6. Overstap van ontwikkelmethodiek

Bij het afronden van de analysefase volgens de watervalmethode, heb ik de overstap gemaakt naar een andere ontwikkelmethodiek. De methodiek in kwestie is de RUP methodiek. De overstap is gemaakt op basis van het resultaat bij opstellen van de requirements in de analysefase. Tijdens de requirementsfase zijn nog niet alle requirements gedefinieerd en met de komst van een Lead Developer als nieuw teamlid, komen er nieuwe requirements aan te pas bij het project. Dit heeft ervoor gezorgd dat de requirements moeten worden beheerd gedurende het gehele traject van het afstudeerproject. De methodiek RUP biedt mij de opties om de requirements gedurende het gehele traject te beheren.

7. Elaboratiefase

7.1. Iteratie 1

Doel:

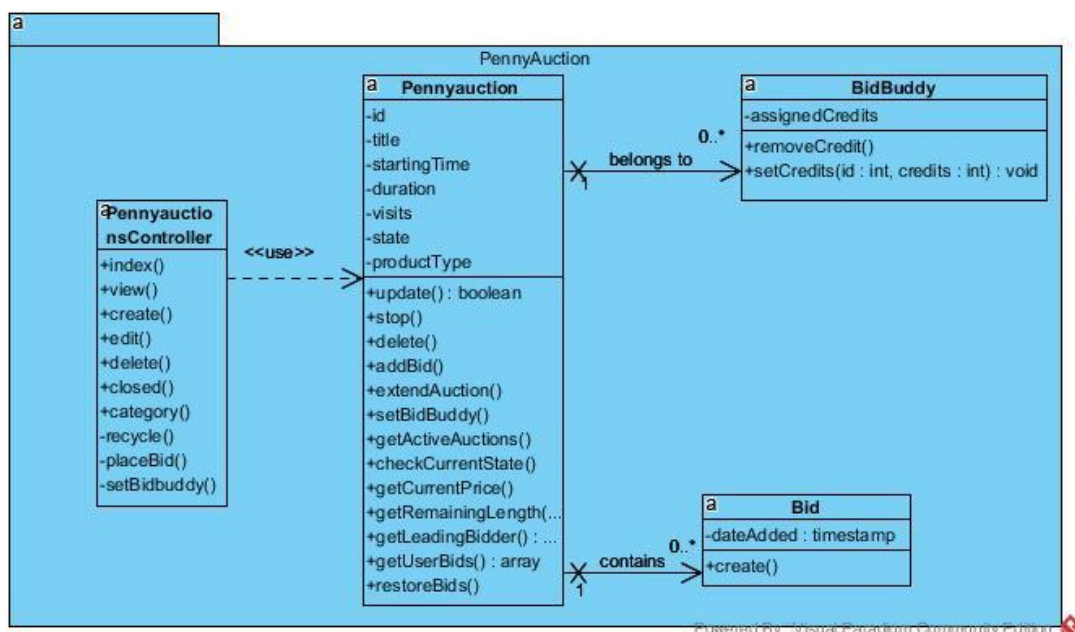
- Definieren van analysepakketten en klassen
- Ontwerpen van het analyseklassendiagram
- Ontwerpen van sequentiediagrammen
- Ontwerpen van schermontwerpen

7.1.1. Definieren van analyse pakketten en klassen

Het definiëren van de analyse pakken en klassen heeft plaatsgevonden aan de hand van het proces van (Ambler, 2001-2012). Het proces is gericht op het definiëren van klassen die een object in de echte wereld moet reflecteren. De objecten die ik heb kunnen definiëren zijn:

- Centenveiling
- Product
- Bod
- Aankoop
- Gebruiker
- Medewerker
- Categorie
- Creditpakket
- Leverancier

De objecten heb ik omgezet in klassen. Deze klassen zijn weergegeven in figuur 10. Klassen worden na het definiëren opgedeeld in pakketten. Een pakket bevat de relevante klassen van een bepaalde type. Een pakket wordt later uitgewerkt tot een component (een los stukje software dat op zichzelf kan werken). Het centenveilingssysteem behoort zich toe tot één component. De eerste versie van een analysepakket wordt weergegeven in figuur 9.



Figuur 9. Analyse pakket van PennyAuction.

In het klassendiagram hebben de klassen ieder hun eigen verantwoordelijkheid. Zij hebben attributen en functies, maar ook relaties met andere klassen. Een relatie indiceert dat een klasse verbonden is met een andere klasse. Via deze verbinding kan een klasse gebruik maken van de functies van de andere klassen.

Bij het opzetten van de analyseklassen door middel van analyse pakketten heb ik de structuur van CakePHP meegenomen. Dit is zichtbaar in figuur 10, waarbij de klassen zijn opgedeeld in twee typen klassen, modellen en controllers.

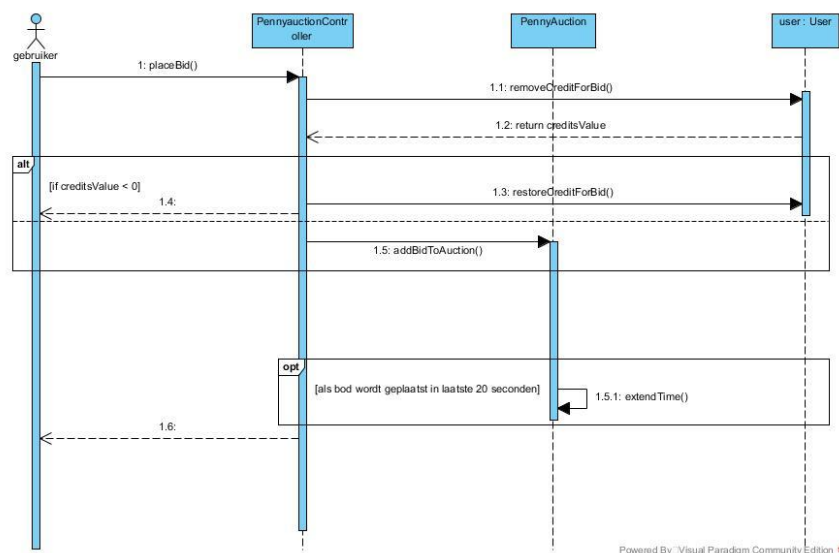


25

Het toekennen van de verantwoordelijkheden kan lastig zijn. De verantwoordelijkheden heb ik toegekend door middel van object georiënteerd programmeren (OOP). Echter is PHP geen object georiënteerde taal. Het framework CakePHP zet PHP voor een gedeelte om naar een object georiënteerde structuur, maar is vooral gebaseerd op de tabellen die aanwezig zijn in de database.

7.2. Ontwerpen van sequentiediagrammen

De sequentiediagrammen zijn uitwerkingen van de Use Cases. Aan de hand van het klassendiagram zet ik sequenties op. In figuur 11 wordt een sequentiediagram weergegeven uit het analysedisciplinedocument.



Figuur 11. Eerste versie sequentiediagram plaatsen bod.

Een sequentiediagram verloopt de stappen van de Use Case. De sequentiediagrammen gebruiken de structuur en de functies die zijn gedefinieerd in het analyseklassendiagram.

7.2.1. Ontwerpen van schermontwerpen

De schermontwerpen zijn gemaakt aan de hand van de website van 123bied.nl. Dit was een eis die door de opdrachtgever is gesteld. Voor het maken van de schermontwerpen maakt ik gebruik van wireframes. Wireframes geven een simpele weergave weer van wat waar op het scherm moet staan. In figuur 12 is een wireframe te zien van een beknopte weergave van een centenveiling.



Figuur 12. Wireframe beknopte weergave van een centenveiling.

De wireframes geven samen met het klassendiagram weer hoe de structuur van het systeem eruit moet komen te zien. Deze heb ik voorgelegd aan de opdrachtgever en de interne belanghebbenden.

Iteratievoortgang:

In deze iteratie is het analysediagram onduidelijk opgesteld. Iemand die dit diagram ziet zou niet weten wat hij er mee aan moet. Dit heeft ook een gevolg op de sequentiediagrammen. Bij een onduidelijk klassendiagram leiden de sequentiediagrammen ook tot onduidelijkheid.

Plan:

In de volgende iteratie moeten de analysemodellen geherdefinieerd worden zodat de structuur van de applicatie verbeterd wordt en de verantwoordelijkheden van de klassen duidelijker worden van het analyse definitieproces in (Ambler, 2001-2012). Na het herdefinieren moet het design discipline worden opgezet, samen met de schermontwerpen en een databasemodel. De schermontwerpen kunnen los van de modellen worden opgezet. Het ontwerp van het databasemodel zal moeten worden gebaseerd op de modellen van het klassendiagram.

Do:

Het herdefiniëren van de modellen zal moeten gebeuren aan de hand van het analyseproces dat beschreven staat in (Ambler, 2001-2012). Bij het opnieuw definiëren moet er rekening mee gehouden worden dat CakePHP gedeeltelijk object georiënteerd is en het proces gericht is op volledige OOP-talen. Vervolgens moeten de designmodellen worden opgezet aan de hand van het designproces uit hetzelfde boek.

Check:

Voor het controleren van de structuur en de verantwoordelijkheden moeten de analysemodellen worden voorgelegd aan mijn begeleider. Zijn inzicht kan zorgen voor goede feedback over de structuur en de verantwoordelijkheden die mogelijk gewijzigd moeten worden.

Act:

De wijzigingen die ik door middel van feedback van mijn begeleider terugkrijg zal ik moeten verwerken in de modellen. Daarna kan de eerste stap worden gezet in het opzetten van een designklassendiagram. Deze stap moet zijn afgerond voordat er wordt begonnen aan de designmodellen.

7.3. Iteratie 2

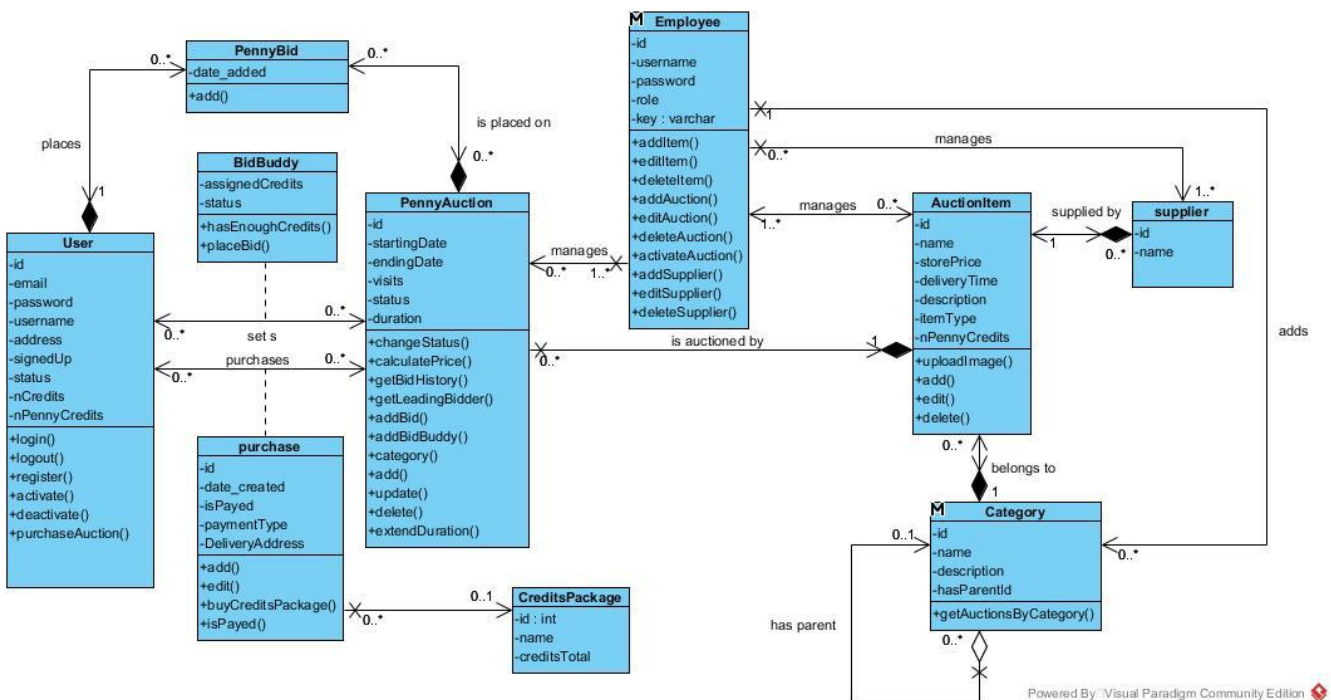
Planning (week12-14)

Doel:

- Opzetten nieuw analyseklassendiagram
- Opzetten nieuwe sequentiediagrammen
- Opzetten van een designklassendiagram
- Opzetten van design sequentiediagrammen
- Opzetten van een databasemodel

7.3.1. Opzetten nieuw klassendiagram

In de eerste opzet van het analyseklassendiagram was de structuur van het framework CakePHP meegenomen in het definiëren van de analysemodellen. Om een analyseklassendiagram die overzichtelijker is voor de belanghebbenden, is het belangrijk om de framework van CakePHP uit het klassendiagram te halen. Het resultaat is de volgende klassendiagram die aanwezig is in figuur 13. Dit model is opgebouwd puur op de werkwijze van OOP.

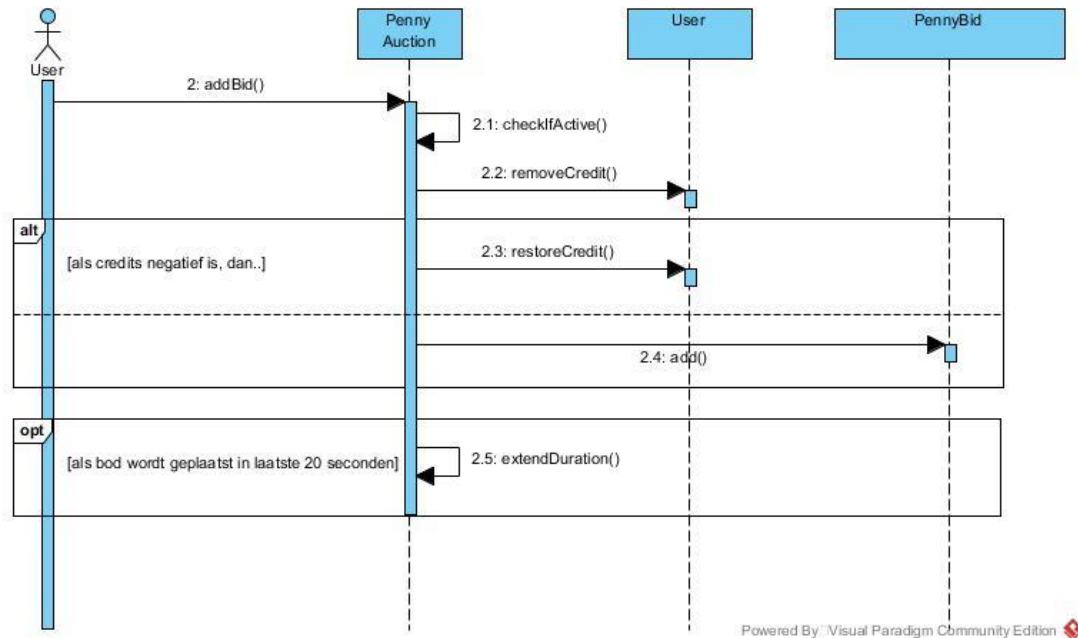


Figuur 13. Versie 3.0 analyseklassendiagram

Het nieuwe klassendiagram is opnieuw opgebouwd waarbij de analyseklassen opnieuw zijn vastgelegd. De verschillende pakketten die aanwezig waren in de eerste versie zijn eruit gehaald. De klassen behoren allen toe aan het pakket centveniling. Het nieuwe klassendiagram heb ik voorgelegd aan de Lead Developer Koen Brouwer. Het diagram was zeer duidelijk geworden in tegenstelling tot de eerste versie. De klassen hebben nu elk een duidelijke verantwoordelijkheid.

7.3.2. Opzetten nieuwe sequentiediagrammen

Op basis van het nieuwe klassendiagram heb ik nieuwe sequentiediagrammen opgezet. De nieuwe diagrammen gebruiken de functies uit het nieuwe klassendiagram. In deze sequentiediagrammen zijn er geen controllers aanwezig, enkel klassen volgens de OOP manier. De sequentiediagrammen zijn versimpeld tegenover de eerste versie, maar geven een betere duidelijkheid over hoe de processen van de Use Cases plaatvinden. In figuur 14 wordt het nieuwe sequentiediagram weergegeven.



Figuur 14. Nieuwe versie sequentiediagram.

Bij de sequentiediagrammen worden enkel de functies van het klassendiagram aangeroepen. Er worden nog geen parameters aangeduid of de waarden die men ontvangt na gebruiken van een functie. De parameters en de return waarde worden pas in het design sequentie vastgelegd. Ik heb de nieuwe analyse sequentiediagrammen gevalideerd bij de Lead Developer.

Het designklassendiagram is een uitgewerkte analyseklassendiagram waar de details van de functies zijn meegenomen. Het designklassendiagram implementeert extra klassen die aanwezig zijn door middel van het CakePHP framework. In figuur 15 zie is de transactie van analyse naar design weergegeven.



Figuur 15. Transactie van enkele klassen van analyse naar het designklassendiagram.

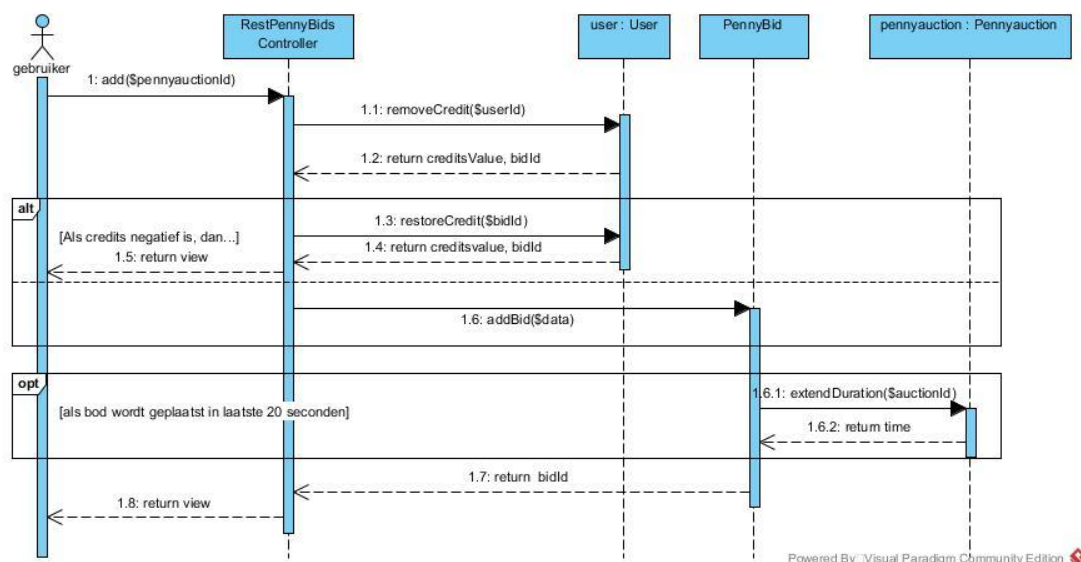
Het designklassendiagram kent 2 soorten klassen: modelklassen en controllerklassen. De controllerklassen nemen verschillende functies over van bepaalde klassen die in het analyseklassendiagram staan beschreven. Controllers nemen functies over de betrekking hebben op het afhandelen van functionaliteiten die de gebruiker wil aanroepen.

Een voorbeeld hiervan is dat de Employee-klasse zijn toevoegingsfunctionaliteit verliest, doordat deze over de verschillende controllerklassen zijn verspreid. De controller klasse van centenveilingen (PennyAuctionsController) bevat nu de functionaliteit voor het aanmaken beheren en verwijderen van centenveilingen. Hetzelfde geldt voor veilingproducten. Deze functies kunnen direct door de gebruiker worden aangeroepen waardoor ze worden uitgevoerd. De uitvoering van die functies in de controller roept op zijn beurt gedefinieerde functies of standaard functies aan van de modellen. De data waarmee de controllers en de modellen hun functionaliteiten uitvoeren is vaak vanuit een view. De views zijn niet opgenomen in het diagram doordat zij geen klasse of functie representeren. Voor een betere weergave van het designklassendiagram, zie bijlage 5.

In het designklassendiagram zijn twee componenten weergegeven. Dit zijn de componenten van Shop en Accounts. In het Accounts component bestaat al een User-klasse met de bijbehorende controller. De User-klasse bevat echter teveel details om in een diagram weer te geven. Voor het designklassendiagram heb ik alleen de nieuwe functies en attributen weergegeven bij de klasse. Hetzelfde geldt voor de ShopController in het Shop Component. Het designklassendiagram heb ik gevalideerd bij de Lead Developer.

7.3.4. Opzetten van design sequentiediagrammen

De design sequentiediagrammen zijn opgezet op basis van de analyse sequentiediagrammen en het designklassendiagram. De design sequentiediagrammen zijn een uitwerking op de analyse sequenties. De design sequenties maken gebruik van de model- en controllerklassen die in het designmodel staan weergegeven. In figuur 16 wordt het design sequentiediagram van plaatsen bod weergegeven.



Figuur 16. Design sequentiediagram plaatsen bod.

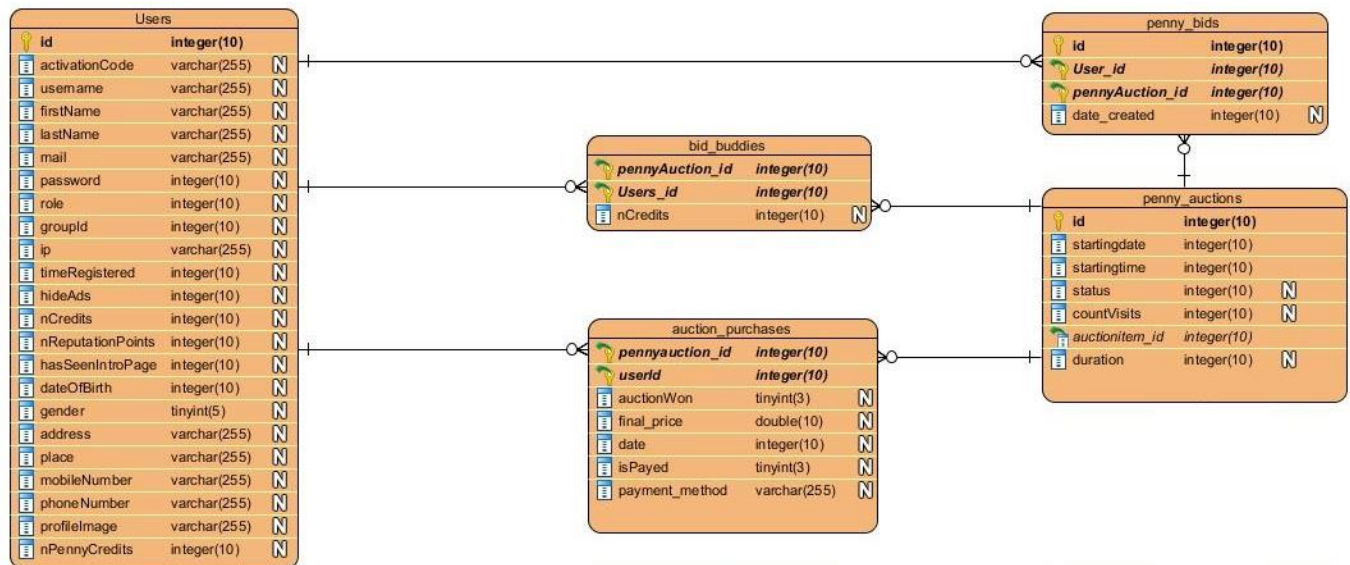
In tegenstelling tot de analyse sequenties, worden in de design sequenties de return waarden van de functie weergegeven (de pijl met een gebroken lijn). De functies bevatten parameters die in het designklassendiagram staan aangegeven.

De design sequentiediagrammen heb ik gevalideerd met de Lead Developer.

7.3.5. Opzetten van een databasemodel

Op basis van het Designklassendiagram heb ik het databasemodel ontworpen. Het creëren van het databasemodel is gedaan volgens het Entiteit Relatie Diagram (ERD).

De entiteiten in het diagram behoren toe aan de modellen uit het designklassendiagram. De modellen uit het klassendiagram hebben een één op één relatie met hun entiteit in het databasemodel. In figuur 17 wordt het databasemodel weergegeven.



Figuur 17. Enkele database tabellen uit het ERD databasemodel.

In tegenstelling tot het designklassendiagram, is de gehele entiteit van gebruik opgenomen in het model. Dit heb ik gedaan doordat de Users entiteit een directe relatie heeft met de andere entiteiten binnen de database.

In het model worden bij vele attributen die een datum weergeven een integer opgeslagen. De data in het platform van Zwoozie worden opgeslagen aan de hand van een Unix-timestamp. De timestamp is een seconden teller die vanaf 1970 actief is en doorloopt. De reden waarom er voor de timestamp is gekozen is, is dat de timestamp onafhankelijk is van de tijdzone waar een gebruiker zich bevindt. De timestamp kan echter per tijdzone anders worden weergegeven. Voor centenveilingen is het belangrijk dat de datum in seconden wordt opgeslagen. Bij centenveilingen moet er veel met data worden gerekend, zoals het verlengen van de eindtijd met 15 seconden als er in de laatste 20 seconden op de centenveiling wordt geboden. De timestamp is een makkelijke oplossing voor het probleem. Het databasemodel is gevalideerd met de Lead Developer.

Iteratievoortgang:

De verantwoordelijkheden van het analyseklassendiagram geven meer duidelijkheid door de afwezigheid van het CakePHP framework. Dit zorgt ervoor dat de sequentiediagrammen op zijn beurt ook duidelijker werden. Het designklassendiagram geeft een goede weergave van de klassen binnen het systeem. De design sequentiediagrammen konden duidelijk worden opgebouwd doordat de analyse sequenties duidelijk zijn opgebouwd. Het databasemodel is redelijk gelijk aan de modellen uit het design diagram.

Plan:

Het ontwerp bevindt zich in de stabiele staat waarbij er kan worden gestart met het realiseren van het ontwerp. Het realiseren zal ongeacht voor nieuwe inzichten zorgen waarbij het ontwerp moet worden aangepast. Het ontwerp moet in komende iteraties worden bijgewerkt gelijkzijdig aan de realisatie.

Do:

Beginnen in de volgende iteratie met het opzetten van de ontwikkelomgeving. Ik zal hierbij een aparte ontwikkel- en testomgeving moeten opzetten. Er moet begonnen worden met het ontwikkelen van de kern van het systeem, namelijk centenveilingen en veilingproducten.

Check: De ontwikkelomgeving moet gelijk zijn aan de ontwikkelomgeving waar de overige onderdelen van Zwoozie zich bevinden. De kern functionaliteiten zal in de komende iteratie moeten voldoen aan: Create, Read, Update en Delete (CRUD).

Act: Nieuwe inzichten tijdens het realiseren moet worden bijgewerkt in de documenten.

Veranderingen in requirements en het ontwerp moeten tijdig worden bijgewerkt om de consistentie te bewaken.

8. Constructiefase : bouwen van de applicatie

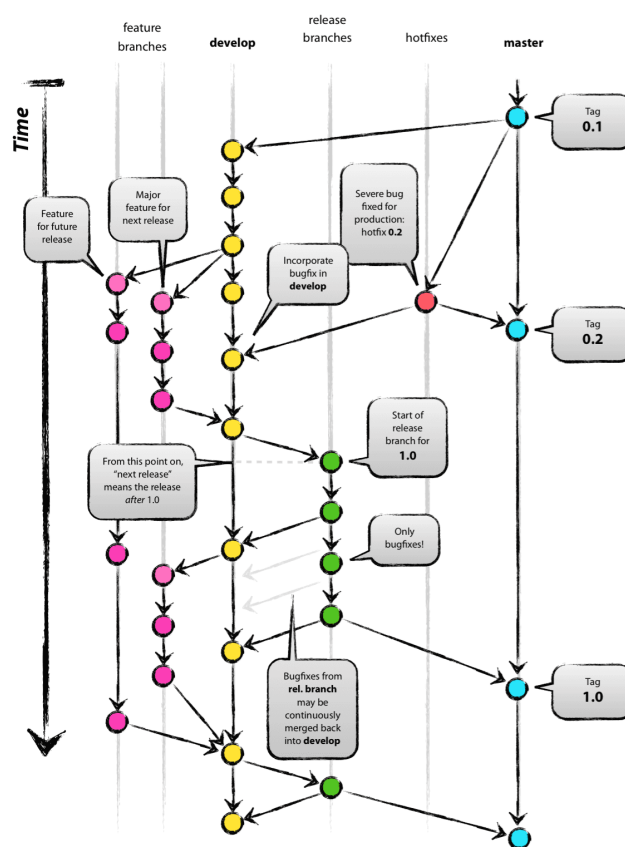
De constructiefase wordt aan de hand van de requirements en het ontwerp gebouwd. Voordat het bouwen kan beginnen, moet eerst de ontwikkelomgeving worden opgezet.

8.1. Iteratie 1

Doel:

- Opzetten van ontwikkelomgeving
- Realiseren van CRUD functionaliteit centenveilingen
- Bijwerken documentatie

8.1.1. Opzetten van ontwikkelomgeving



Figuur 18. Gitbranchingmodel

De ontwikkelomgeving van Zwoози wordt beheerd via Git. Git is een versiebeheertool die werkt door middel van branches (vertakkingen). In git zijn er verschillende branches aanwezig:

- Master-branch
- Development-branch
- Release-branch
- Features-branches
- Hotfixes-branch

De master branche geeft de live-omgeving van het platform weer. Op de development-branch wordt het systeem gerealiseerd. De release-branch bevat de versies die in de toekomst naar de master branches toegaan en dient als testomgeving. Features zijn vertakkingen van de development-branch waarin specifieke onderdelen van het systeem wordt ontwikkeld. In mijn geval wordt in de

pennyauction feature centenveilingen gerealiseerd. Wanneer een feature klaar is, wordt deze samengevoegd met de development branche. Hotfixes zijn bugs die snel moeten worden opgelost en vinden dicht plaats bij de master branche.

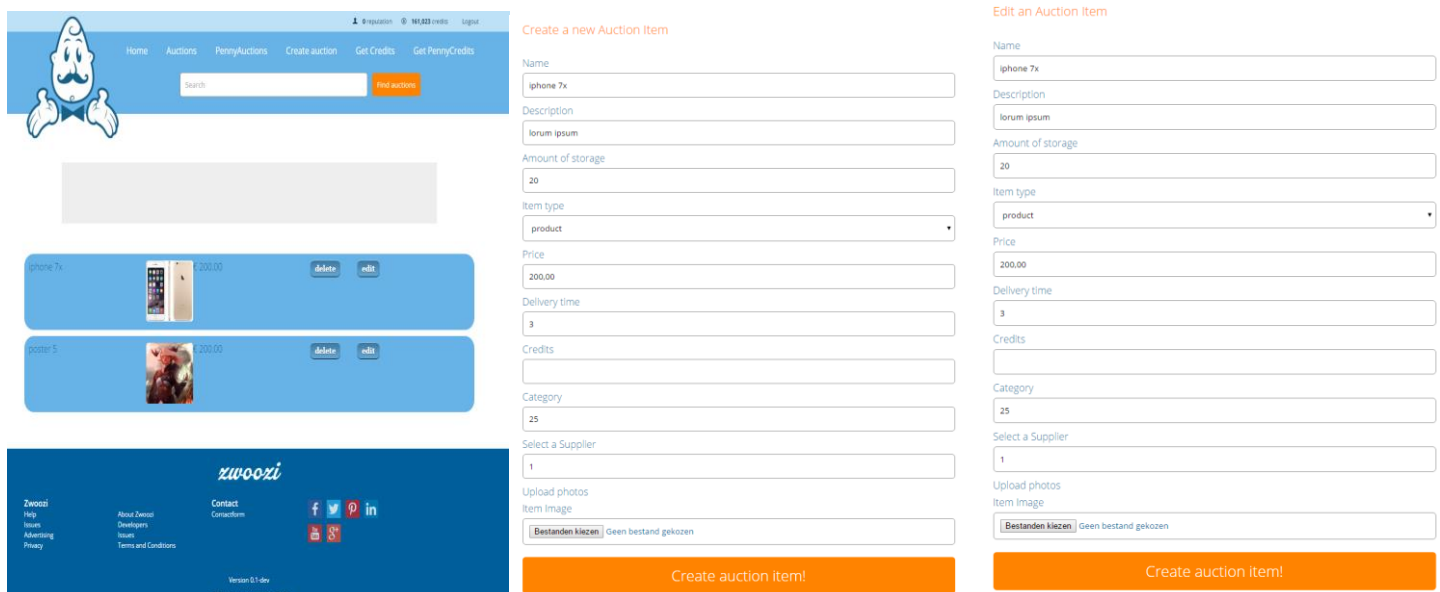
Aan de hand van features kunnen meerdere ontwikkelaars delen van het systeem opbouwen zonder dat ze elkaar in de weg zitten. Het systeem kan simpel en snel worden opgebouwd.

Het component van centenveilingen ontwikkel ik in de feature centenveiling, een vertakking van de development-branche. Ik gebruik het programma SmartGit als een UI bovenop de command line-applicatie.

Voor het ontwikkelen van het systeem maak ik gebruik van PhpStorm om code sneller op te kunnen bouwen. PhpStorm kan door middel van Compass SCSS bestanden compileren naar stylesheets. Voor het maken van de verbinding met de database van Zwoozii maak ik gebruik van Navicat Premium. Met deze middelen kon ik perfect aansluiten op de bestaande data-flow van Zwoozii.

8.1.2. Realiseren van CRUD functionaliteit centenveilingen

Met de start van het ontwikkelen ben ik met het realiseren van de CRUD (Create, Read, Update, Delete) functionaliteit van centenveilingen bezig geweest. De kern van centenveilingen bevindt zich rond het product dat wordt geveild. Ik ben daarmee begonnen met de CRUD van producten. Er zijn twee type producten die bij ene centenveiling kunnen worden geveild. Men kan credits veilen en men kan producten veilen. Deze 2 producten waren in het ontwerp weergegeven als twee aparte klassen, zie figuur 16, door middel van overerving. Het framework CakePHP is geen fan van overerving en raadt het ten strengste af. In overleg met de applicatiebeheerder was de beste keus om de twee klassen samen te voegen.



Figuur 19. Webpagina's van veilingproducten.

Het resultaat van het samenvoegen is één formulier die beide producten kan aanmaken. In figuur 19 zijn 3 pagina's zichtbaar. De eerste geeft het overzicht aan van de producten. De tweede geeft het aanmaken van een product weer en de derde geeft het aanpassen van een product weer. Hoewel deze functionaliteit alleen voor de medewerkers is bedoeld, is in overleg met de opdrachtgever en de applicatiebeheerder de keuze gemaakt om voor het proof of concept de functionaliteit beschikbaar te maken voor alle gebruikers.

Na het realiseren van de CRUD van veilingproducten ben ik begonnen met het maken van de CRUD van een centenveiling. Bij het realiseren van de CRUD kwam een probleem naar voren. Centenveilingen moeten wanneer ze actief zijn continu worden geüpdatet door de browser van de gebruiker zonder dat de pagina herlaad. Dit is niet mogelijk in PHP.

Om te zorgen dat dit wel mogelijk is voor het systeem, ben ik een klein onderzoek gestart naar realtime updaten. Ik stuitte hierbij op het fenomeen van Restcontrollers. Restcontrollers halen data op door middel van een HTTP-request. Deze requests zijn uit te voeren door middel van javascript. Restcontrollers kunnen net net als gewone controllers gebruik maken van de modellen om de data op te halen uit de database. Zij sturen de data naar het javascript echter toe in JSON (JavaScript Object Notation). De Restcontrollers heb ik opgenomen in de planning van de volgende iteratie.

8.1.3. Bijwerken van documentatie

Met de komst van REST-controllers moet het ontwerp worden bijgewerkt, met name het designklassendiagram en de design sequentiediagrammen. Naast REST-controllers moest de documentatie ook worden bijgewerkt door middel van het samenvoegen van de twee product modellen. De product modellen bestonden voorheen uit een leverancierproduct en een creditsproduct. Bij goed gebruik van OOP kan er bij deze modellen gebruik worden gemaakt van overerving. Helaas laat CakePHP dit niet toe en raden overerving ten strengste af. Een van de oplossingen was de twee modellen apart opnemen, ook al hebben ze dezelfde attributen, of door de klassen samen te voegen tot één klasse met vele null waarden. In overleg met de Lead Developer hebben we ervoor gekozen om er een klasse van te maken omdat die zorgt voor minder data verkeer. Deze verandering moest worden doorgevoerd in het databasemodel. In het requirements model, met name de Use Cases, zijn er nieuwe Use Cases bijgekomen. De nieuwe Use Cases zijn:

- Aanpassen product
- Verwijderen product
- Aanpassen centenveiling
- Activeren centenveiling

Deze Use Cases vullen de bestaande Use Cases aan. De nieuwe Use Cases zijn gevalideerd met de Lead Developer.

Iteratievoortgang:

De ontwerpdocumentatie is bijgewerkt voor de REST-controllers samen met de samenvoeging van producten naar 1 tabel. Het realiseren van de CRUD van centenveilingen liep vertraging op door middel van onderzoek naar de REST-controllers.

Plan:

In de volgende iteratie moeten de REST-controllers worden gebouwd voor centenveilingen en de CRUD moet verder worden afgemaakt. Er

Do:

De REST-controllers vereisen wijzigingen in de configuratie van CakePHP. Deze configuratie moet worden aangepast anders kunnen de REST-controllers geen HTTP-requests binnen krijgen. Voor de REST-controllers moet er javascript worden geschreven om de centenveilingen te kunnen updaten.

Check:

De IDE van PhpStorm bevat een tool waarmee HTTP-requests kunnen worden verzonden. Deze tool zal gebruikt moeten worden om de JSON data die de REST-controllers die wordt ontvangen te testen

Act:

Als de REST-controllers in de volgende iteratie goed functioneren, dan dragen ze bij aan de CRUD van centenveilingen. De CRUD van centenveilingen zou hiermee afgerond kunnen worden.

8.2. Iteratie 2

Doel:

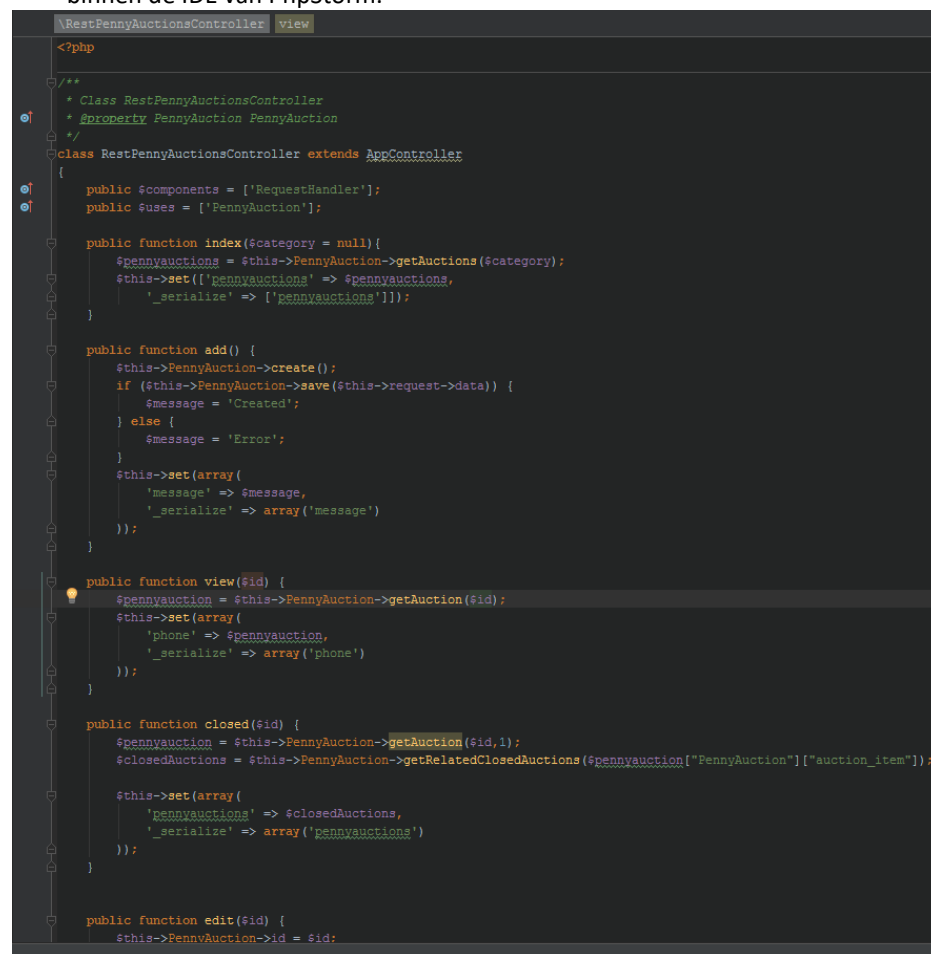
- Realiseren van REST-controllers
- Realiseren CRUD centenveiling
- Documentatie aanpassen.

8.2.1. Realiseren van REST-controllers

Na het ontwerpen moeten de REST-controllers worden gerealiseerd. Ik heb twee REST-controllers gerealiseerd. Een Restcontroller voor centenveilingen en een Restcontroller voor biedingen. De Restcontroller voor centenveilingen bevat de functionaliteiten:

- Index
- Add
- View
- Closed
- Edit

De Restcontroller voor biedingen bevat alleen de add functionaliteit. Om de Restcontroller te realiseren moet er enkele configuratie van CakePHP worden aangepast. CakePHP blokkeert automatisch HTTP-requests naar de controller toe. In de configuratie heb ik aangegeven dat hij deze moet toelaten tot de Restcontroller. Na het opzetten heb ik de REST-controllers kunnen aanspreken via Javascript HTTP-requests. Aan de hand van deze requests krijg ik JSON terug die ik kan gebruiken om een deel van de pagina up te daten met nieuwe informatie. Het testen van de Restcontroller heb ik gedaan aan de hand de tool RESTfull Web Service tester dat zich bevindt binnen de IDE van PhpStorm.



```
<?php
/**
 * Class RestPennyAuctionsController
 * @property PennyAuction PennyAuction
 */
class RestPennyAuctionsController extends AppController
{
    public $components = ['RequestHandler'];
    public $uses = ['PennyAuction'];

    public function index($category = null) {
        $pennyauctions = $this->PennyAuction->getAuctions($category);
        $this->set(['pennyauctions' => $pennyauctions,
            '_serialize' => ['pennyauctions']]);
    }

    public function add() {
        $this->PennyAuction->create();
        if ($this->PennyAuction->save($this->request->data)) {
            $message = 'Created';
        } else {
            $message = 'Error';
        }
        $this->set(array(
            'message' => $message,
            '_serialize' => array('message')
        ));
    }

    public function view($id) {
        $pennyauction = $this->PennyAuction->getAuction($id);
        $this->set(array(
            'phone' => $pennyauction,
            '_serialize' => array('phone')
        ));
    }

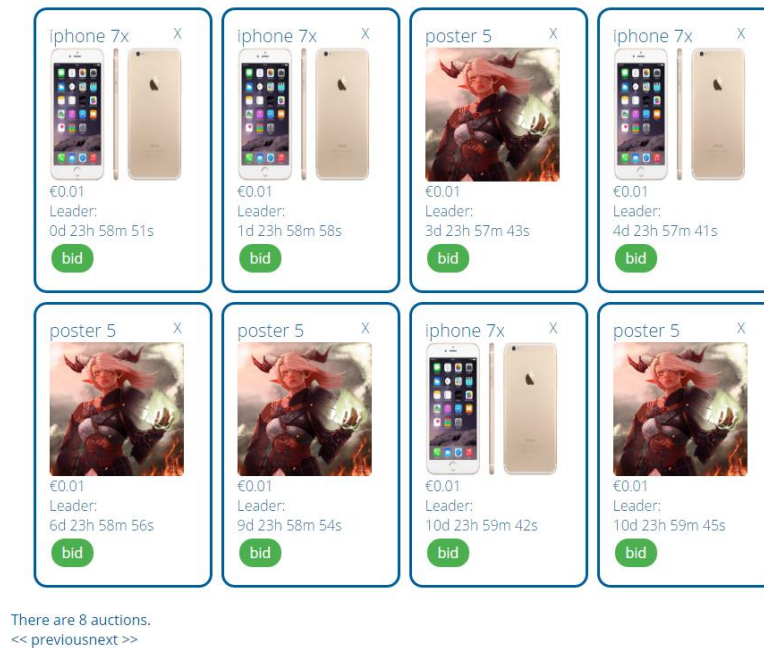
    public function closed($id) {
        $pennyauction = $this->PennyAuction->getAuction($id,1);
        $closedAuctions = $this->PennyAuction->getRelatedClosedAuctions($pennyauction['PennyAuction']['auction_item']);

        $this->set(array(
            'pennyauctions' => $closedAuctions,
            '_serialize' => array('pennyauctions')
        ));
    }

    public function edit($id) {
        $this->PennyAuction->id = $id;
    }
}
```

Figuur 20. Code snippet van een Restcontroller in CakePHP

[View Pennyauctions](#)



Figuur 21. Actieve centenveilingen met testdata

In figuur 21 wordt een overzicht van centenveiling weergegeven. De data van de veilingen worden opgehaald via Javascript, die de data weer haalt bij de REST-controllers. De drie velden die onder de productafbeelding wordt weergegeven zijn de velden die continue moeten worden bijgewerkt. De reden voor het bijwerken is om de gebruikers actueel gebruik te laten maken van centenveilingen.

8.2.2. Realiseren CRUD centenveiling

De REST-controllers zijn de UPDATE van CRUD van centenveilingen. Aangezien het creëren in de vorige iteratie al is gerealiseerd, hoeft alleen het aanpassen van een veiling en het verwijderen van een veiling worden gerealiseerd. Na het realiseren van deze functionaliteiten ben ik het aankopen van biedcredits gaan realiseren. Zonder biedcredits kan een centenveiling niet functioneren. Hoewel er op die moment op een centenveiling kon worden geboden in realtime, werd het bieden nog niet verrekend met de biedcredits die aan de account van de gebruiker toebehoren. Het aanschaffen van biedcredits gebeurt aan de hand van creditspakketten. Deze pakketten kunnen credits en biedcredits bevatten. Bij het kopen van credits in het prototype, wordt het credits aantal direct bij het account toegevoegd. De data voor het betalen van een credits pakket is gemokt en zal later pas worden geïmplementeerd.

Iteratievoortgang:

Plan: De centenveilingen bevinden zich in een staat waarin men gebruik kan maken van de actieve centenveiling. Ze kunnen nu bieden en de data wordt iedere halve seconde bijgewerkt. De onderdelen van het systeem moeten nog worden getest er moet een testplan worden opgesteld en het kopen van een centenveiling moet worden ontwikkeld. In overleg met de Lead Developer zal het testen van centenveilingen gelijk gebeuren met Zwoози.

Do: Voor het testen moeten de testsoorten en de testcriteria worden aangeduid. Het systeem moet worden voorbereid op het uitvoeren van de testen in de volgende iteratie.

Check: De testen geven resultaat weer over de staat van het systeem. Aan de testen kunnen we zien of het prototype klaar is voor oplevering of dat er nog onderdelen gewijzigd moeten worden.

Act: Als de testen aangeven dat vele functionaliteiten niet werken, dan moeten deze direct na de test worden aangepast. Functionaliteiten die niet zo kritisch zijn kunnen na de oplevering aangepast worden.

8.3. Iteratie 3

Doel:

- Schrijven van het testplan
- Unittesten
- Functionele acceptatietest
- Opleveren van het prototype

8.3.1. Schrijven van een testplan

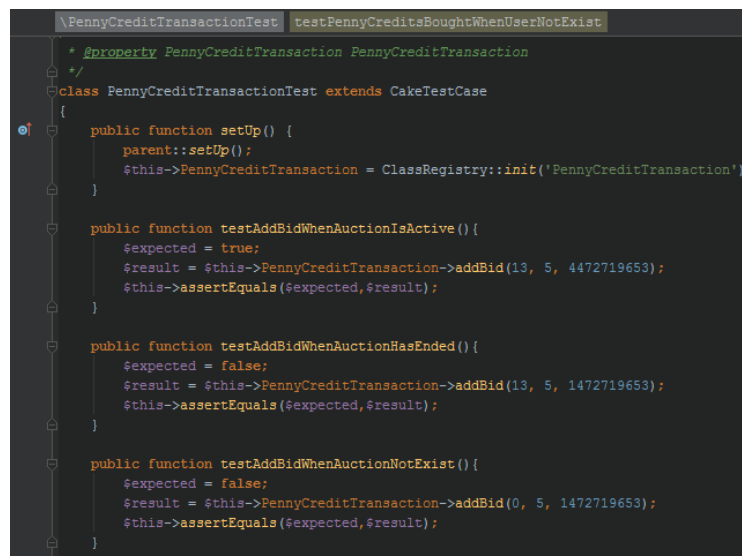
Voor deze iteratie is in de planning opgenomen dat het testplan moet worden geschreven. Het schrijven van het testplan en het uitvoeren van de testen zouden gelijk gebeuren met het platform van Zwooz. Door gebrek aan documentatie en afspraken met betrekking tot het testen van Zwooz, was het testen voor onbepaalde tijd uitgesteld. Dit hield in dat de testen voor centenveiling niet gelijktijdig zou verlopen met de testen van Zwooz. De tijd voor het opleveren begint immers te dringen. Bij het schrijven van het testplan was de acceptatiecriteria niet bekend en was het lastig om een testplan te schrijven. Ik heb in de laatste iteratie geen testplan meer geschreven doordat het schrijven ervan teveel tijd in beslag zou nemen. Het prototype moest echter nog getest worden om aan te tonen dat het systeem wel of niet naar behoren werkt. Zonder het testplan ben ik direct de testen gaan schrijven.

De testsoorten die ik heb gekozen zijn de unittesten en de functionele acceptatietest. De acceptatie test is gekozen door middel van dat het prototype de juiste functionaliteiten moet aantonen en of deze kloppen met wat de opdrachtgever verwacht van het prototype. Unittesten heb ik uit eigen belang gekozen omdat ik de direct de code wou testen die ik had geschreven.

8.3.2. Unittesten

Bij unittesten test je direct de functionaliteiten door middel van direct de geschreven code te testen. Bij de testen wordt er een input verwacht en een output. Aan de hand van de inputs komen er verschillende outputs. Voor de unittesten test ik op de gewenste output op basis van de verschillende scenario's waarin de functie zich kan bevinden. De functie `givePennyCredits` heeft 2 parameters nodig om de functie uit te voeren. De id van de gebruiker en het aantal biedcredits. Door middel van het opstellen van verschillende testcases met goede en foute parameters kunnen vele scenario's worden gedekt en getest.

CakePHP heeft zelf een module waarmee unittesten kunnen worden uitgevoerd. De unittesten zijn alleen effectief op de modellen. De controllers functionaliteiten kan ik door middel van unittesten niet testen.



```
PennyCreditTransactionTest testPennyCreditsBoughtWhenUserNotExist

* @property PennyCreditTransaction PennyCreditTransaction
*/
class PennyCreditTransactionTest extends CakeTestCase
{
    public function setUp() {
        parent::setUp();
        $this->PennyCreditTransaction = ClassRegistry::init('PennyCreditTransaction');
    }

    public function testAddBidWhenAuctionIsActive() {
        $expected = true;
        $result = $this->PennyCreditTransaction->addBid(13, 5, 4472719653);
        $this->assertEquals($expected, $result);
    }

    public function testAddBidWhenAuctionHasEnded() {
        $expected = false;
        $result = $this->PennyCreditTransaction->addBid(13, 5, 1472719653);
        $this->assertEquals($expected, $result);
    }

    public function testAddBidWhenAuctionNotExist() {
        $expected = false;
        $result = $this->PennyCreditTransaction->addBid(0, 5, 1472719653);
        $this->assertEquals($expected, $result);
    }
}
```

Figuur 22. Statische unittesten voor transactie van biedcredits.

De unittesten zijn in eerste instantie statisch opgezet. Dit houdt in dat vaste parameters worden meegegeven ongeacht de situatie. Enkele testen veranderen de data die tot de parameters toebehoort. Bij het herhaaldelijk uitvoeren van de testen ga je merken dat de testen niet meer kloppen. Als voorbeeld probeer je de centenveiling aankoop te betalen omdat hij nog niet betaald is. Na de eerste uitvoering is de aankoop al betaald. Bij het nogmaals uitvoeren van de functie krijg je andere resultaten. De testen zijn vervolgens dynamischer gemaakt. In plaats van vaste gegevens mee te geven laat ik de test vooraf op zoek gaan naar parameters waaraan de test moet voldoen. De gegeven parameter zal per test verschillen, maar zal wel dezelfde situatie schetsen en testen.

```
public function testGetLatestPennyBid()
{
    //we want to test if we can get the latest bid of a pennyauktion
    $pennyauktionWithBids = $this->PennyBid->find("first", [
        "group" => "PennyBid.pennyauktionId",
        "fields" => [
            "PennyAuction.id",
            "PennyAuction.endingdate",
            "PennyAuction.status",
            "PennyBid.*",
            "COUNT(*) as amount"
        ],
        "having" => [
            "amount >" => 0
        ]
    ]);
    //fetching the latest pennybid from a pennyauktion that has bids
    $expected = true;
    $result = !empty($this->PennyAuction->getLatestPennyBid($pennyauktionWithBids["PennyAuction"]["id"]));
    $this->assertEquals($result, $expected);
}
```

Figuur 23. Dynamische unittest van model PennyAuction

Uit de testresultaten kwam interessante data naar voren. Zo bleek dat de biedcredits niet van het account werden gehaald bij het bieden op een centenveiling. Bij het kopen van ene pakket met biedcredits werd deze ook niet aangevuld bij het biedcredits totaal van de gebruikers. Dit werd veroorzaakt doordat de wijzigingen bij het toekennen van de biedcredits niet werden vastgelegd door middel van een afwezige commit. Zonder de commit voert de database een rollback uit en maakt alle wijzigingen ongedaan. Na het toevoegen van de commit-functionaliteit heb ik de test nogmaals uitgevoerd. Ditmaal werden de biedcredits wel toegekend/afgenomen bij het plaatsen van biedingen of het aanschaffen van biedcredits. Hieronder worden unittesten weergegeven van de model PennyCreditTransaction. In figuur 23 is een dynamische unittest weergegeven.

8.3.3. Acceptatietest

Unittesten alleen is niet voldoende om een systeem te testen. Bij het belang van het prototype in de staat af waarin het wordt opgeleverd belangrijk. Om te testen of het prototype voldoet aan de requirements en de juiste functionaliteiten levert, wordt door middel van een functionele acceptatietest de test met de producteigenaar uitgevoerd. De producteigenaar heeft immers het grootste belang bij het systeem en de beslismomenten van het systeem ligt bij hem. Hij beslist of het prototype voldoet aan zijn verwachtingen en geeft een Go/NoGo voordat het prototype wordt opgeleverd.

De test heb ik uitgevoerd door de producteigenaar het prototype vanuit de functionele kant te testen. Door middel van de test is hij de applicatie doorlopen en heeft bij elke functionaliteit kunnen aangeven hoe hij de test ervaren heeft en of het prototype voldoet aan zijn eisen. Bij elke functionaliteit heb ik de relevante requirements opgenomen zodat er bij elke geteste functionaliteit kan worden teruggevallen op de requirements die gedurende het project zijn vastgelegd.

Na de resultaten van het testen bleek daar er een paar functionaliteiten aanwezig waren die nog niet helemaal klopte. De functionaliteiten in kwestie was het aanpassen van een veilingproduct waarbij de aanpassing alleen werd doorgevoerd als je een nieuwe afbeelding van het product uploadde. Bij het

bieden op een veiling mag de laatste bidder niet nog een keer bieden. Deze functionaliteit was afwezig doordat er geen requirements voor was opgesteld. De requirement heb ik vast moeten leggen in de requirements voor het verder realiseren van het prototype in de toekomst.

De opdrachtgever had verder nog wat opmerking over hoe de stroming van de functionaliteiten aanwezig of dat het uiterlijk nog moest worden aangepast. De rest van de applicatie gaf hem een goed beeld over hoe het uiteindelijke systeem moet functioneren en was tevreden met het prototype. Voor de oplevering van het prototype heeft de producteigenaar een Go gegeven.

8.3.4. Opleveren van het prototype

Na het verkrijgen van de Go en de documentatie up-to-date te hebben gemaakt kan het prototype worden opgeleverd. Het prototype bevindt zich al in .git, wat zorgt voor een makkelijke transactie om het prototype te bewaren in de ontwikkelomgeving van Zwoozi. Het prototype zal na de stageperiode verder worden ontwikkeld, waarbij ze mij hebben gevraagd om het systeem verder te ontwikkelen als vaste werknemer bij Zwoozi.

Iteratie evaluatie

Het afstudeertraject van het project bevindt zich aan het einde. Het prototype is ontwikkeld en geaccepteerd door de producteigenaar. De volgende stap is het uitwerken van het prototype naar een volledig systeem als nieuwe werknemer. Voor het uitwerken van het prototype zal een nieuw project worden gestart.

9. Evaluatie

In dit hoofdstuk evalueer ik het afstudeertraject op het dorlopen proces, de totstandkoming van de producten en de behaalde competenties.

9.1. *Procesevaluatie*

Gedurende het proces heb ik gebruik gemaakt van twee verschillende ontwikkeltechnieken. Na de eerste fase heb ik de overstap moeten maken naar een andere methodiek doordat de watervalmethode niet de juiste ondersteuning bood om de requirements en het ontwerp het gehele traject te beheersen. Mijn criteria voor het kiezen van de watervalmethode was omdat het makkelijker uit te voeren is in een nieuw en onvolwassen team. Ik was er zeker van dat ik de requirements gedurende de eerste fase van het project volledig kon definiëren. Dit heb ik echter verkeerd ingeschat waarbij het verder volgen van de methode voor grote risico's zal zorgen. Bij het schrijven van het afstudeerplan had ik de risico in acht moeten nemen dat de producteigenaar nog geen duidelijk beeld had van hoe het systeem eruit moet komen te zien. Het systeem was echter pas voortgekomen uit het idee. Voor volgende projecten moet ik beter gaan letten op de ontwikkelomgeving en de eisen die aan het project worden gesteld voordat ik een methodiek kies die ik denk dat er geschikt voor is. Dit zal als gevolg hebben dat ik onnodige risico's kan vermijden en een betere beheersing heb op het project.

Voor een groot deel van het afstudeertraject heb ik weinig tot geen begeleiding gekregen van mijn begeleider bij het afstudeerbedrijf. Dit heeft zich vooral de eerste helft van het traject voorgedaan. Mijn begeleider Florian de Groot is echter een maand voor mij gestart als vaste werknemer van het bedrijf en had zelf nog niet veel ervaring binnen het bedrijf. In het plan van aanpak heb ik opgenomen om per week voortgangsgesprekken met hem te houden, maar de gesprekken hebben zich weinig voorgedaan. Ik nam niet de initiatief of hem keer op keer te vragen om een voortgang gesprek. Door het gebrek aan voortgang gesprekken heb ik in de eerste fase moeilijk de tijd kunnen beheersen. Met het definiëren van requirements kwam ik echter krap te zitten met tijdnoed om alle requirements te kunnen achterhalen en duidelijk kunnen opstellen. De begeleiding verbeterde door de komst van de Lead Developer, Koen Brouwer. In de resterende periode heeft Koen mij begeleid bij het ontwikkelen van prototype. Ik kreeg meer vat op de tijdbeheersing bij het project,

Na het overstappen wordt er verwacht dat er volledig volgens de RUP methode wordt gewerkt. Dit houdt in dat kleine delen van het systeem stukje bij stukje worden gerealiseerd tot een stuk software en dat de producten (requirements en ontwerp) van globaal naar specifiek gaan worden uitgewerkt. Zelf ben ik gewend om een fase compleet af te ronden voordat ik naar de volgende fase ga zoals die bij de watervalmethode aanwezig is. Zelfs in RUP probeerde ik in één iteratie of in één fase het gehele product te ontwikkelen. Ik had dit beter kunnen aanpassen door te beginnen met het globaal specificeren van het project en per iteratie stuk voor stuk het ontwerp van de applicatie specifieker te maken. In de watervalmethode had ik al een groot deel van de requirements gespecificeerd, waardoor ik de inceptiefase van de RUP methodiek heb overgeslagen. De werkwijze van de watervalmethodiek werd nog gedeeltelijk doorgevoerd in de RUP methodiek. De vrijheid om de documenten aan te passen in iedere iteratie heeft ervoor gezorgd dat ik nieuwe inzichten heb kunnen verwerken en de het prototype naar de wens van de opdrachtgever heb kunnen realiseren.

Als ik terug kijk op het gehele traject dat ik heb afgelegd, zijn er momenten waar het wat minder ging. Het werkproces veranderde waarmee ik me niet goed heb kunnen aanpassen. Maar voor de meeste problemen heb ik samen met de Lead Developer en mijn begeleider een oplossing voor kunnen bedenken. Requirements en het ontwerp zijn door de belanghebbenden goed te begrijpen. Voor volgende projecten moet ik me meer bezig gaan houden met de voorbereiding van het project.

9.2. *Productevaluatie*

Het opzetten van de requirementsdocumentatie is vrij makkelijk doorlopen, al koste het redelijk wat tijd voordat ik überhaupt de juiste sets aan requirements heb weten te analyseren. Zoals in de risicoanalyse is opgenomen in het plan van aanpak, heeft het gebrek aan eindgebruikers ervoor gezorgd dat de requirements van de eindgebruikers aan het schommelen waren. Zelfs de marketingexpert Nick Verheij wist niet helemaal duidelijk wat de eindgebruikers van het systeem verwachtte. De eisen voor de eindgebruiker zijn naar boven te gekomen door te kijken naar 123bied.nl. ik heb gedurende het requirements proces veel tijd doorgebracht op de website van 123bied.nl. Vele requirements over de eindgebruikers heb ik bij hen ontdekt en voorgelegd aan de opdrachtgever. De requirements zijn gedurende het afstudeertraject goed beheerd en de documenten is bij elke iteratie snel bijgewerkt. Het document heeft elke interne stakeholder laten zien waaraan het systeem moet voldoen. De Use Cases zijn duidelijk evenals de business rules en de overige requirements.

Het ontwerpen van het analyse en design discipline document heeft echter veel problemen ondervonden. Dit komt vooral doordat het ontwerpen van de analysemodellen gedurende mijn opleiding gedaan werden aan de hand van OOP. De structuur van CakePHP volgt dit principe echter niet en daardoor was het lastig om verantwoordelijkheden binnen het systeem toe te kunnen kennen aan klassen. Het continu aanpassen van de analysemodellen en het vastleggen van de verantwoordelijkheden heeft mij veel tijd gekost. Dit was een probleem waar ik echter niet zelf uitkwam. De problemen die ik heb ondervonden heb ik uiteindelijk maar voorgelegd aan mijn begeleider, maar de feedback die ik kreeg was niet geheel bruikbaar. Ik kon voorlopig wel verder met het ontwerpen, maar moest later terugkomen om het ontwerp een stuk beter aan te scherpen. Uiteindelijk laat het ontwerp wel zien hoe het systeem in elkaar komt te steken. Het database ontwerp was duidelijk te definiëren voor de interne stakeholders.

Het bouwen van het systeem ging niet geheel soepel omdat het ontwerp op dat moment nog niet heel stevig in zijn schoenen stond. De onderdelen van het systeem die wel al stevig stonden heb ik al kunnen bouwen. De toevoeging van REST-controllers maakt een deel het prototype een RESTful webservice. Het bouwen van het systeem hield gedurende de realisatie de standaard aan van CakePHP, namelijk magere controllers en dikke modellen. Hoewel het prototype niet helemaal werkte volgens de eisen van de opdrachtgever, was het prototype een geslaagd stukje software wat kan worden uitgebouwd naar een volledig systeem.

Aan het testen is heel weinig aandacht besteed. Zoals in vele gevallen blijft testen altijd een achtergedachte bij het realiseren. Het project heeft zich te ver gevorderd voordat er aan testen werd gedacht. Zo had de criteria van het testplan direct worden opgezet bij het opzetten van de requirements. Aan de hand van die criteria zullen er vast andere testen zijn geweest die veel geschikter waren om het systeem te testen. Voor het volgende project zal ik een testgedreven aanpak moeten kiezen, zodat testen in een vroeg stadium al wordt meegenomen.

9.3. *Competentieevaluatie*

Competentie 1.4 Uitvoeren analyse door definitie van requirements

De requirements heb ik volledig en dekkend opgesteld, want ik heb gebruik gemaakt van de methoden die beschreven staan in (Swart, 2010). Aan de hand van de methoden en de wijze waarop ze zijn geformuleerd laat duidelijk het verschil zien tussen de verschillende requirements. De stakeholders hebben de requirements gevalideerd en geprioriteerd, waardoor zelf stakeholders die niet veel met ICT te maken hebben de requirements kunnen begrijpen. De requirements zijn geprioriteerd op basis van een MoSCoW analyse en zijn traceerbaar met hun stakeholder in het requirementsmatrix. De requirements zijn beheerd door middel van meerdere versies van de requirementmodellen zoals Use Cases te bewaren in dropbox, samen met verschillende versies van de requirementsdocumentatie. De requirements zijn consistent met elkaar voor zover dit mogelijk is bij het proof of concept. De requirements resulteerde in duidelijke Use Case realisaties in de analyse en design documenten van het ontwerp.

Competentie 2.1 Opstellen gegevensmodel voor de database

Bij het opstellen van het gegevensmodel van de database zijn de juiste objecten uit het domein gekozen, want de objecten geven een reële weergave van de objecten in de werkelijkheid. De tabel zijn genormaliseerd voor zover het kan en enkele delen zijn gedenormaliseerd vanwege de structuur van het CakePHP framework die overerving ten strengste afraad. De beschrijvingen van de objecten geven een duidelijk consistent beeld over het domein door het gebruik van duidelijke namen voor de attributen en de tabelnamen. De onderlinge samenhang is juist vastgelegd doordat replica van data tot het minimum is behouden.

Het gegevensmodel is getoetst op de juiste wijze op volledigheid, duidelijkheid, consistentie en andere normen, want de tabellen in de database leggen de structuur vast van de objecten in de modellen van de applicatie. De tabellen bevatten geen dubbele data en er is geen gebrek van attributen aanwezig in het gegevensmodel. De tabellen in de database hebben een duidelijk omschrijving en geven weer waarom ze welke relaties hebben met de andere tabellen.

Competentie 3.3 Bouwen applicatie

Het ontwerp is verfijnd omdat er rekening is gehouden met het hergebruiken van de modellen en functies binnen de applicatie. De verantwoordelijkheid is per model gedefinieerd en er is rekening gehouden met de testbaarheid door middel van unittesten. De applicatie is goed te wijzigen in zowel het ontwerp als het proof of concept.

Het systeemdeel is op de juiste wijze gebouwd en werkend is opgeleverd doordat de opdrachtgever een functionele acceptatietest heeft uitgevoerd en daarbij de werking van de applicatie een Go heeft gegeven. Bij het bouwen is er goed gebruikt gemaakt van de structuur en standaard binnen CakePHP. Verder is er gebruik gemaakt van de standaard door met een Restfull web service te werken die enkel communiceert door middel van javascript. Er is op de juiste wijze gebruik gemaakt van het framework CakePHP waarbij ze magere controllers en dikke modellen willen zien. De applicatie is samengesteld door middel van controllers die gebruik maken van de modellen en data zal door middel van javascript niet lekken doordat er alles of niks wordt getoond bij het manipuleren van de webpagina.

10. Bibliografie

- Ambler, S. W. (2001-2012). *Agile Modeling and the Rational Unified Proces (RUP)*. Opgehaald van Agile Modeling: [HTTPwww.agilemodeling.com/essays/agileModelingRUP.htm](http://www.agilemodeling.com/essays/agileModelingRUP.htm)
- Apache. (2016). *index*. Opgehaald van apachefriends: [HTTPS://www.apachefriends.org/index.html](https://www.apachefriends.org/index.html)
- compass. (2016). *compass-style*. Opgehaald van compass: [HTTPcompass-style.org/](http://compass-style.org/)
- hampton Catlin, N. W. (2006-2015). *sass-lang.com*. Opgehaald van sass-lang.com: [HTTPSass-lang.com](https://sass-lang.com)
- Handlebars. (2016). *index*. Opgehaald van handelbarsjs: [HTTPhandlebarsjs.com/](http://handlebarsjs.com/)
- HTTPcompass-style.org/. (2016). *HTTPcompass-style.org/*. Opgehaald van compass-style: [HTTPcompass-style.org/](http://compass-style.org/)
- JetBrains. (2016). *phpstorm*. Opgehaald van www.jetbrains.com: [HTTPS://www.jetbrains.com/phpstorm/](https://www.jetbrains.com/phpstorm/)
- JQuery. (2016). *JQuery*. Opgehaald van JQuery: [HTTPS://jquery.com/](https://jquery.com/)
- Navicat. (2016). *navicat premium*. Opgehaald van www.navicat.com: [HTTPS://www.navicat.com/products/navicat-premium](https://www.navicat.com/products/navicat-premium)
- OMG. (2005). *what is uml*. Opgehaald van www.uml.org: [HTTPwww.uml.org/what-is-uml.htm](http://www.uml.org/what-is-uml.htm)
- PHP. (2016). *secure php*. Opgehaald van HTTPS://secure.php.net: [HTTPS://secure.php.net](https://secure.php.net)
- Swart, N. d. (2010). *Handboek Requirements*. Delft: Eburon Business.
- W-mdr90. (2012, 05 23). *de watervalmethode*. Opgehaald van pc-en-internet.infonu: [HTTPpc-en-internet.infonu.nl/diversen/97898-de-watervalmethode.html](http://pc-en-internet.infonu.nl/diversen/97898-de-watervalmethode.html)

11. Bijlagen

11.1. *Bijlage 1: Mailcontact met kansspelautoriteit*

11.2. *Bijlage 2: Visie- en scoperequirementsdocument*

11.3. *Bijlage 3: Softwarerequirementsspecificatie document*

11.4. *Bijlage 4: Analysedisciplinedocument*

11.5. *Bijlage 5: Designdisciplinedocument*

11.6. *Bijlage 6: Testrapport*

Mailcontact met kansspelautoriteit:

- 12053635@student.hhs.nl:

Geachte kansspelautoriteit,

Voor een afstudeerproject doe ik onderzoek naar centenveilingen ter voorbereiding van een implementatie ervan in een systeem. Nu viel het bij op bij enkele publicaties die jullie hebben gepubliceerd dat centenveilingen verboden zijn omdat de gebruiker ervan onvoldoende inzicht in de veiling heeft. Wat wij willen doen is om de gebruikers echter wel deze inzicht te geven. Wat ik me hierbij afvraag is waaraan een centenveiling dan allemaal moet voldoen om legaal te zijn voor gebruik. Hierbij vraag ik u of wij hiervoor een gesprek kunnen houden. Het zou mij heel erg helpen bij het afstuderen. Ik hoop snel van u te horen.

- Kansspelautoriteit

Geachte heer Van der Ham,

Wij hebben uw vragenformulier in goede orde ontvangen. U heeft een vraag over centveilingen en vraagt of dit is toegestaan. Hieronder ga ik op uw vraag in.

Relevant is of hier sprake is van een kansspel. De definitie van een kansspel staat in artikel 1 van de Wet op de kansspelen, en luidt als volgt. *Gelegenheid geven om mede te dingen naar prijzen of premies, indien de aanwijzing der winnaars geschiedt door enige kansbepaling waarop de deelnemers in het algemeen geen overwegende invloed kunnen uitoefenen.* Belangrijke elementen zijn het feit dat er sprake moet zijn van enige kansbepaling en het feit dat deelnemers over het algemeen geen overwegende invloed hebben op de kansbepaling. In artikel 1 van de Wet op de kansspelen (Wok) is dit wettelijk vastgelegd en is het verboden een kansspel zonder vergunning aan te bieden.

Klassieke veilingen, waarbij het hoogste bod altijd wint en de tijd waar binnen te bieden is van te voren duidelijk vast staat, vallen over het algemeen niet onder de Wet op de kansspelen. Voor een zuivere veiling is dan ook geen vergunning vanuit de Wet op de kansspelen nodig. Dit komt omdat er geen sprake is van een kansbepaling en de bidders overwegende invloed hebben op hun kans om te winnen. Daarnaast is het ook inzichtelijk hoeveel en hoe hoog er door anderen geboden is. Dit maakt dat veilingen als in een veilinghuis of op bijvoorbeeld eBay niet onder de Wet op de kansspelen vallen. Voor een zuivere veiling is dan ook geen vergunning vanuit de Wet op de kansspelen nodig.

Echter, er zijn ook veilingen die wel een kansspel zijn (zoals centveilingen). Hier is het kanselement vaak gelegen in het feit dat nadat er een bod is gedaan, de biedingstijd niet met een vooraf vastgestelde tijd wordt verlengd, waardoor het einde van een bepaalde veiling niet te bepalen is. Er lijkt bovendien geen sprake te zijn van overwegende invloed op de uitkomst van het veilingproces; men weet nooit of een bod het laatste is en hoeveel mensen geïnteresseerd zijn en ook hebben geboden. Ook speelt hierin mee dat het uitbrengen van een bod vaak geld kost. De bieder betaalt dus om een bod te doen en is hier feitelijk sprake van een inleg. De uitvoering van dit soort veilingen bevat kenmerken van een kansspel en valt hiermee onder de bepalingen van de Wok.

De centenveilingwebsites die de Kansspelautoriteit is tegengekomen, hebben vaak de volgende kenmerken:

- De centveiling wordt via het internet of de telefoon;
- Men doet een bod op aangeboden producten, waarna het reeds geboden bedrag met één cent wordt verhoogd;
- Na elk gedane bod begint de klok terug te tellen vanaf een tevoren vastgestelde tijd (enkele minuten/seconden). Wanneer in die periode niemand anders biedt is het product door de laatste bieder gewonnen. Wanneer iemand anders biedt springt de klok terug naar de starttijd en wordt het product op naam van de nieuwe bieder gesteld;
- Elke keer dat je een bod uitbrengt, kost het je een biedpunt (of je de veiling wint of niet). Voor een biedpunt/biedbundel moet meestal € 1,- betaald worden;
- Deelnemers hebben vaak geen inzicht in het aantal deelnemers dat meedoet en weten ook niet hoe vaak de (andere) deelnemers de prijs van het product verhoogd hebben.

Essentieel is dat elk bod 1 cent toevoegt aan het laatste bod op een product, maar om te kunnen bieden moeten er eerst biedpunten of een biedbundel gekocht worden.

Tot slot kunnen wij u niet uitnodigen voor een gesprek maar hopelijk geeft bovenstaande tekst u genoeg informatie voor uw afstudeeropdracht. Veel succes ermee,

met vriendelijke groet,

A.P. (Annette) Sluiter

Communicatiemedewerker/publieksvoorlichter

.....
Afdeling Juridische Zaken en Communicatie

Kansspelautoriteit

Rijnstraat 50 | 2515 XP | Den Haag

Postbus 298 | 2501 CG | Den Haag

T 070 302 1300

F 070 302 1399

E asluijter@kansspelautoriteit.nl

W <http://www.kansspelautoriteit.nl>

- 12053635@student.hhs.nl

Bedankt voor de reactie die u had gestuurd. Uit de reactie zijn er echter nog wat vragen naar boven gekomen, waarbij we hopen dat u deze antwoorden kan geven.

Aangezien het verboden is om kansspelen uit te voeren zonder een vergunning, kunnen er voor centenveilingen een vergunning worden bemachtigd?

Zo ja, aan welke eisen en regels moet een centenveiling zich houden om een vergunning te kunnen bemachtigen?

Wat zijn de bestaande klachten rondom centenveilingen, zodat wij deze klachten als input kunnen gebruiken om vertrouwd een centenveiling te houden.

Wij streven ernaar om de gebruiker zoveel mogelijk inzicht te geven in het veiling proces. Hiervoor willen wij het concept wat aanpassen om het kans element te verminderen.

Op jullie site geven jullie aan dat mensen weleens mee mogen doen met een kansspel mits dit in een vertrouwde gebeurd. Ik hoop met de antwoorden die u geeft om het centenveilingconcept aan te passen zodat deze juist in een vertrouwde omgeving kan afspelen.

Het centenveilingsysteem zal gaan draaien naast een normaal veilingsysteem wanneer het systeem zal worden gelanceerd. Dus de klant zal meerdere mogelijkheden hebben m.b.t. veilingen.

Als online centenveilingen verboden zijn omdat alle online kansspelen illegaal zijn, Hoe komt het dat bekende centenveilingen als 123bied.nl nog actief zijn na 6 jaar. In eerder contact met uw instantie kwam naar voren dat zij niet over een vergunning bezitten om centenveilingen aan te bieden maar dit toch doen. Kunt u mij hiervoor inlichten?

Ik hoop dat u deze vragen kan beantwoorden.

- Kansspelautoriteit

Geachte heer Van der Ham,

Voor de aanpak van online kansspelen, zoals ook bij centenveilingen, hanteert de Kansspelautoriteit prioriteringscriteria.

Dit omdat er veel illegale online kansspelen worden aangeboden en de Kansspelautoriteit niet meteen tegen alle aanbieders daarvan kan optreden. De Kansspelautoriteit zal daarom keuzes moeten maken welke ze eerst aanpakt.

Mede op grond van deze prioriteringscriteria bepaalt de Kansspelautoriteit tegen welke illegale online kansspelen ze eerst gaat optreden. Dit betekent echter niet dat wanneer een aanbieder nog niet geprioriteerd wordt voor verdere aanpak deze dan legaal is. Indien zo'n aanbieder kansspelen aanbiedt in Nederland en daarvoor geen vergunning heeft, dan is en blijft hij illegaal.

Het kan dus voorkomen dat aanbieders, zoals door u genoemd nog gewoon aanbieden. Dat betekent nogmaals dus niet dat zij legaal bezig zijn.

Met vriendelijke groet,

Publieksvoorlichting

.....
Kansspelautoriteit

Wijnhaven 24 | 2511 GA | Den Haag

Postbus 298 | 2501 CG | Den Haag

T 070 302 1300

E info@kansspelautoriteit.nl

W www.kansspelautoriteit.nl

Didacticum

Centenveilingen Zwoozi

Visie en Scope

Versie 1.0

Centenveilingen Zwoozi	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

Documenthistorie

Datum	Versie	Beschrijving	Auteur
17-05-2016	1.0	Eerste versie	Daveron van der Ham

Centenveilingen Zwoozi	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

Inhoudsopgave

1.	Inleiding	4
2.	Bedrijfsvisie	4
2.1	Huidige situatie	4
2.2	Gewenste situatie	4
2.3	Businessrequirements	4
3.	Belanghebbenden	4
3.1	Eindgebruikers	4
3.2	Gebruikersomgeving	5
3.3	Overige belanghebbenden	5
4.	Systeemscope	6
4.1	Omschrijving systeem	6
4.2	Positionering systeem	6

Centenveilingen Zwoozi	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

1. Inleiding

Dit document bevat de visie van het veilingplatform 'Zwoozi' en het te ontwikkelen centenveilingen systeem. Het heeft tot doel om alle betrokkenen bij het centenveilingen systeem duidelijk te maken wat voor systeem er moet worden ontwikkeld en wat de redenen er daarvoor zijn.

2. Bedrijfsvisie

2.1 Huidige situatie

Het huidige platform van Zwoozi bevindt zich in de opstartfase. Zo heeft het platform een huisstijl en enkele basissystemen/basisfunctionaliteiten als een account systeem en het aanmaken van veilingen. Bij dit platform ontbreken momenteel veel systemen en functionaliteiten. Dit leidt tot het probleem toe dat het platform incompleet is en dat het platform nog niet kan worden gelanceerd in de productieomgeving.

2.2 Gewenste situatie

Om het ontwikkelproces van Zwoozi te verbeteren wordt één van de systemen, het systeem van centenveilingen, uitbesteed als een apart project. Hiermee willen zij bereiken dat het ontwikkelen van het platform bevorderd wordt en het systeem wordt uitgebreid met extra verdienmodellen. Het systeem moet ervoor zorgen dat klanten in een vertrouwde omgeving mee kunnen doen aan centenveilingen. Om te zorgen dat klanten tevreden zijn met dit systeem streven zij erna om de klanten meerdere voordelen aan te bieden tegenover andere online centenveilingen.

2.3 Business Requirements

De opdrachtgever wil het systeem van centenveilingen laten ontwikkelen want:

- de opdrachtgever wil de omzet verhogen met 25%.
- de opdrachtgever wil het aanschaffen van kredieten verhogen met 15%
- de opdrachtgever wil het aantal premium accounts verhogen met 10%.
- de opdrachtgever wil de verkoopwaarde van producten die hij wil gaan veilen verhogen tegenover de inkoopprijs met een marge van 30%.
- de opdrachtgever wil verschillende type veilingen aanbieden.
- de opdrachtgever wil de gebruiker meer inzicht geven in het veiling proces tegenover andere online centenveilingen.
- de gebruikers willen producten tegen een lage prijs kunnen winnen.
- de opdrachtgever wil de gebruikers het gevoel geven dat er geen verliezers zijn.

3. Belanghebbenden

3.1 Eindgebruikers

Consumenten (via internet)

Consumenten zijn internetgebruikers die via het platform Zwoozi gebruik kunnen maken van de centenveilingen. Zij kunnen participeren en bieden op de centenveilingen wanneer zij een account bij Zwoozi hebben aangemaakt. De consumenten worden vertegenwoordigd door marketingmedewerker en producteigenaar Nick Verheij.

Zwoozi medewerkers

De Zwoozi medewerkers bemensen de controlepalen van het centenveilingen systeem. Zij zijn verantwoordelijk voor het aanmaken, starten en beheren van centenveilingen. Zij zorgen dat de aanwezige informatie op de site up-to-date blijven. Zij moeten een levering na kunnen trekken wanneer een product verkeerd wordt afgeleverd bij de klant. Voor hun is het belangrijk dat de bovengenoemde processen zo veel mogelijk worden geautomatiseerd zodat het beheer simpel in gebruik kan worden genomen. De medewerkers van Zwoozi zullen vertegenwoordigd worden door ontwikkelaar en projectleider Koen Brouwer.

Centenveilingen Zwoози	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

3.2 Gebruikersomgeving

Het veilingplatform Zwoози gaat gebruik maken van dit systeem. Het platform zal online beschikbaar zijn voor gebruikers en bezoekers. De Zwoози medewerkers moeten onafhankelijk van locatie met de beheertools kunnen werken. De medewerkers zijn bekend met de werking van webapplicaties.

3.3 Overige belanghebbenden

Directie(opdrachtgever)

De directie van het Zwoози platform bestaat uit de producteigenaar en de investeerders. Zij zijn verantwoordelijk voor het maken van beslissingen binnen en rondom dit project en/of systeem en zijn ook de verantwoordelijke van het systeem. Zij hebben de mogelijkheid om extra mankracht bij het project toe te kennen. Zij worden vertegenwoordigd door de marketingmedewerker en producteigenaar Nick Verheij.

Applicatiebeheerder

De applicatiebeheerder is verantwoordelijk voor het technisch beheeraspecten voor het platform Zwoози en zal ook verantwoordelijk zijn voor de technische beheeraspecten van het systeem voor centenveilingen. Hij wordt vertegenwoordigd door de projectleider Koen Brouwer

Kansspelautoriteit(overheidsinstelling)

Kansspelautoriteit geeft autorisatie voor verschillende kansspelen in vertrouwde omgevingen. Zij zijn bekend met de klachten die rondom centenveilingen plaatsvinden en verlenen vergunningen voor kansspelen. Hoewel zij niet iemand hebben die als vertegenwoordiger kan dienen, zal alle contact verlopen met info@kansspelautoriteit.nl

123bied.nl(concurrent)

123bied is een centenveiling site dat al minimaal 6 jaar draait. Zij tonen aan hoe centenveilingen zich moeten gedragen. Zij zullen echter geen vertegenwoordiger hebben, doordat wij de informatie over centenveilingen kunnen verkrijgen door het bezoeken van hun systeem.

Centenveilingen Zwoozl	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

4. Systeemscope

4.1 Omschrijving systeem

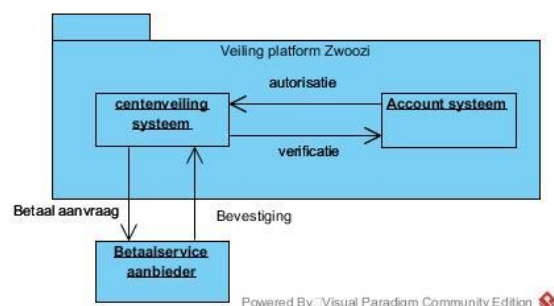
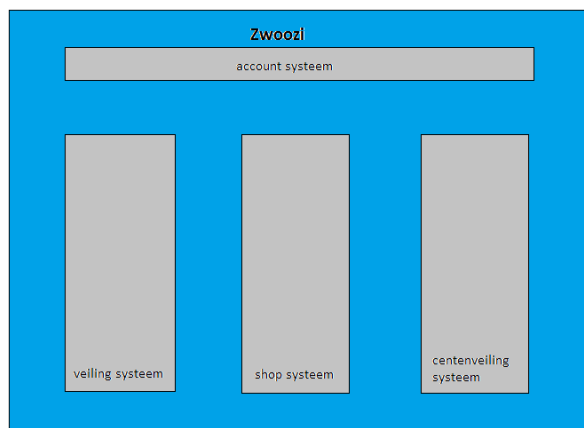
Centenveilingen zijn een type veiling waar men betaalt voor biedkredieten die zij kunnen besteden om te kunnen bieden. De prijs van een product begint bij 1 cent. Per keer dat er op het product wordt geboden wordt de prijs verhoogd met 1 cent. Dit kost ze echter 1 biedkrediet. Een bod dat op het laatste moment wordt geplaatst zorgt ervoor dat er extra tijd bij de veiling wordt opgeteld, waardoor het einde van een veiling niet altijd in zicht is. De winnende bidder moet na het plaatsen van zijn laatste bod nog wel het product betalen. De aankoop prijs van een product ligt stukken lager dan de winkelprijs. Centenveilingen zijn echter een vorm van kansspelen en de geboden biedkredieten zijn niet meer terug te krijgen. Voor de gebruikers is de betrouwbaarheid van het systeem een kritische factor, doordat gebruikers hun vertrouwen verliezen door het meedoen met andere online centenveilingen.

4.2 Afbakening systeem

Het Centenveilingensysteem zal voorlopig nog niet live gaan binnen Zwoozl. Daardoor wordt er tijdens dit project een 1.0 versie ontwikkeld. De 1.0 versie van dit systeem zal zich focussen op de belangrijkste functionaliteiten die zorgen dat het systeem functioneel wordt. Dit houdt in dat koppelingen met externe betaalaanbieders en leveranciers geen deel zullen uitmaken van het project. Binnen Zwoozl moet het systeem gebruik gaan maken van het account systeem, maar doordat deze nog niet ontwikkeld is, wordt dit systeem in eerste instantie los gebouwd.

4.3 Positionering systeem

Het systeem van centenveilingen zal een onderdeel van Zwoozl uit gaan maken en zal draaien op dezelfde server. Didacticum heeft een off-shore serverpark die zij zelf volledig in beheer hebben. Binnen het platform communiceert het systeem met enkele andere systemen binnen Zwoozl. Het systeem heeft een interface met het account systeem. Het systeem zal verder met externe betaalaanbieders moeten communiceren. Dit zal doormiddel van een API worden gedaan.. Hieronder wordt het contextdiagram weergegeven van het systeem. Voor de technische beperking geldt dat het systeem gebruik moet maken van een MySQL database.



Centenveilingen Zwoozi	Datum: 17-05-2016
Visie en Scope	Versie: 1.0

4.4 Globale gebruiker requirements

Bij de volgende werkzaamheden is de ondersteuning van het veilingssysteem vereist:

- De consument wil via het internet deelnemen aan een centenveiling;
- De consument wil een bod plaatsen op een centenveiling;
- De consument kan zijn bieding niet intrekken;
- De consument wil kredieten kunnen toevoegen aan zijn account;
- De consument wil een biedhulpje kunnen inzetten om voor hem te bieden bij voor een centenveiling;
- De consument wil een product goedkoper kunnen bemachtigen;
- De medewerker wil een centenveiling kunnen wijzigen;
- De medewerker wil een centenveiling plaatsen;
- De medewerker wil producten kunnen toevoegen om te veilen.

4.5 Belangrijkste niet-functionele requirements

Het centenveilingssysteem moet zich minimaal aan de volgende kwaliteitseisen voldoen:

- Een onervaren internetgebruiker moeten binnen 5 minuten kunnen bieden na registratie. Een ervaren internetgebruiker moet binnen 2 minuten kunnen bieden nadat zij zich hebben geregistreerd.
- De duur van het plaatsen van een bod moet in minder dan een halve seconden gebeuren. Bij drukte mag het maximaal 1 seconde duren.

Hiernaast zijn de volgende kwaliteitseigenschappen uit de ISO 9126 standaard van belang: leerbaarheid, begrijpelijkheid, snelheid en wijzigbaarheid. De eisen waaraan het systeem hierbij moet voldoen worden op een later moment gespecificeerd.

Centenveilingen Zwoози

Software Requirements Specificatie:

Versie 2.3

Auteur:

Daveron van der Ham

Documenthistorie

Datum	Versie	Beschrijving	Auteur
16-05-2016	0.1	Initiële versie	Daveron van der Ham
18-05-2016	0.2	UC model toegevoegd	Daveron van der Ham
20-05-2016	0.3	UC specificaties toegevoegd	Daveron van der Ham
25-05-2016	0.4	UC specificaties aangepast	Daveron van der Ham
27-05-2016	0.5	UC workflows toegevoegd	Daveron van der Ham
30-05-2016	0.6	Aanvullende requirements toegevoegd.	Daveron van der Ham
03-06-2016	1.0	Eerste versie voor requirements voor het basissysteem	Daveron van der Ham
18-07-2016	2.0	Nieuwe requirements door nieuw teamlid.	Daveron van der Ham
15-08-2016	2.1	Use cases aangepast door verandering processen	Daveron van der Ham
29-08-2016	2.2	Use case flows aangepast	Daveron van der Ham
26-09-2016	2.3	Requirements bijwerken aan de hand van de laatste constructie iteratie.	Daveron van der Ham

Inhoudsopgave

1.	Inleiding	4
2.	Systeem	5
3.1	Context	5
3.2	Eigenschappen	7
3.3	Aannames en afhankelijkheden	7
3.	Requirements	8
3.1	Use Case model	8
3.2	Use case specificaties	9
3.2.1	Use case Zoeken centenveiling	9
3.2.2	Use case Plaatsen bod	10
3.2.3	Use case Inzetten biedhulpje	11
3.2.4	Use case Kopen biedkredieten	12
3.2.5	Use case Betalen centenveiling	13
3.2.6	Use case Plaatsen centenveiling	14
3.2.7	Use case Toevoegen product	15
3.2.8	Use case Aanpassen product	16
3.2.9	Use case Verwijderen product	17
3.2.10	Use case Verwijderen centenveiling	18
3.2.11	Use case Recyclen centenveiling	19
3.2.12	Use case Aanpassen centenveiling	20
3.2.13	Use case activeren centenveiling	20
3.3	Aanvullende requirements	21
3.3.1	Business rules	21
3.3.2	Gebruiker requirements	22
3.3.3	Software requirements:	23
3.3.4	Kwaliteitseigenschappen	24
4.	Aanvullend informatie	25

1. Inleiding

Dit document is bedoeld voor de belanghebbenden van het project in te lichten over de eisen en wensen die worden gesteld aan het centenveilingsysteem. Hierbij wordt er onderscheid gemaakt tussen gebruikers requirements, software requirements en kwaliteitseigenschappen. Verder zal dit document over de aannames en afhankelijkheden spreken die bij dit project van belang zijn.

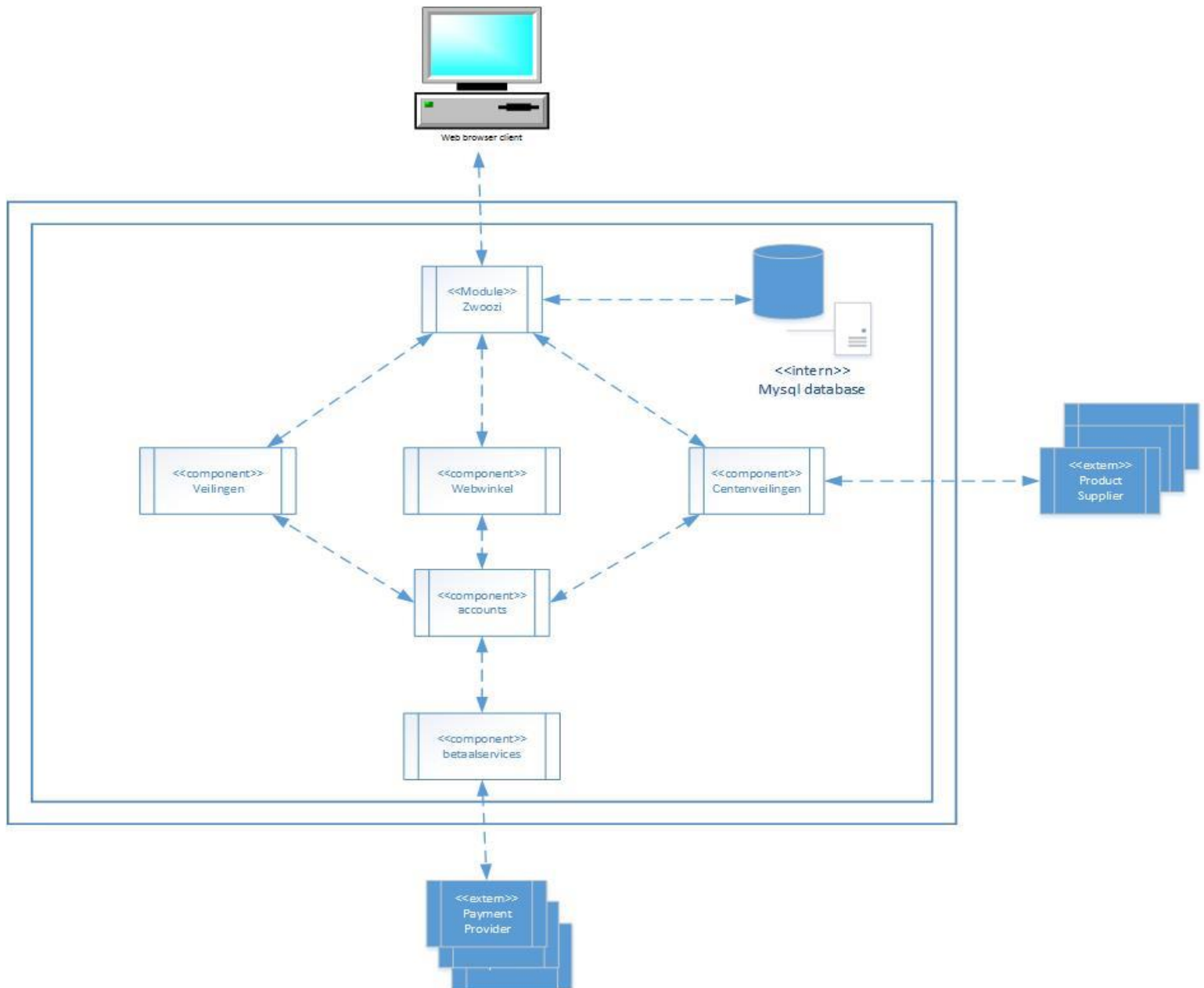
2. Systeem

3.1 Context

Centenveilingen zijn een type veiling waar men op kunnen bieden aan de hand van biedcredits. Het kost één biedcredit om een bod op de centenveiling te plaatsen, waarbij de prijs van het geveilde product met één cent toeneemt. De prijs van een product begint bij één cent aan de start van een centenveiling. De gebruikers betalen om te kunnen bieden. Een bod dat op het laatste moment wordt geplaatst zorgt ervoor dat er extra tijd bij de veiling wordt opgeteld, waardoor het einde van een veiling niet altijd in zicht is. De winnende bidder moet na het plaatsen van zijn laatste bod, wanneer de centenveiling is geëindigd, nog wel het product betalen. De aankoop prijs van een product ligt stukken lager dan de winkelprijs, vaak maar enkele euro's. Centenveilingen zijn echter een vorm van kansspelen en de geboden biedkredieten zijn niet meer terug te krijgen.

Om centenveilingen binnen het platform van Zwoozie te introduceren, wordt er een systeem ontwikkeld die de centenveilingen een onderdeel van Zwoozie maken. Het systeem zal als onderdeel van Zwoozie gaan draaien op de zelf beheerde servers van Didacticum en zal het platform uitrusten met een extra verdienmodel. Doordat het onderdeel is van Zwoozie zal het gebruik maken van hetzelfde account systeem en webwinkel dat zwoozie gebruikt.

Het systeem zal verder met externe betaalservice aanbieders communiceren en in de toekomst ook met leverancier systemen. Hun systemen zullen gekoppeld worden doormiddel van een API. Hieronder is een contextdiagram te zien van het systeem. Voor de technische beperking geldt dat het systeem gebruik moet maken van een MySQL database.



Hoewel de communicatie met de andere interne systemen hierboven worden weergegeven, moeten deze systemen nog worden gebouwd. Het doel van dit project is om een concept versie te ontwikkelen. In het concept worden alleen de belangrijkste functionaliteiten van het systeem opgenomen en gerealiseerd. Dit houdt in de functionaliteiten die uit de eisen en wensen komen met de hoogste prioriteit.

3.2 Eigenschappen

Het systeem zal centenveilingen aanbieden via het web. De gebruikers van het systeem kunnen aan deze veilingen deelnemen. Een centenveiling heeft een bijbehorend product dat wordt geveild. Bij de veiling wordt bijgehouden wie er heeft geboden en wie er voor het laatst heeft geboden, een biedhistorie en de leidende bieder. Elke centenveiling die actief is heeft een aflooptijd die aangeeft hoeveel tijd er nog aanwezig is voordat de centenveiling afloopt. Wanneer een gebruiker in de laatste 20 seconden bidt dan wordt er 15 seconden bij de aflooptijd opgeteld. Het product bij een veiling bestaat uit een omschrijving en heeft één of meerdere afbeeldingen die getoond moeten worden. Gebruikers hebben de mogelijkheid om direct het product te kunnen kopen, mits er meerdere aantallen van het product aanwezig zijn. Hebben zij voor het direct kopen geboden op de centenveiling, dan krijgen ze korting aan de hand van hoe vaak zij hebben geboden. Het systeem moet medewerkers de mogelijkheid stellen om centenveilingen te kunnen aanmaken en deze te kunnen plaatsen. Verder moeten de medewerkers de veilingen kunnen wijzigen en zelfs verwijderen. Dit geldt ook voor de producten die worden geveild. Hoewel Zwoози en centenveilingen beide met credits werken, zal centenveilingen een eigen valuta krijgen, namelijk biedcredits.

3.3 Aannames en afhankelijkheden

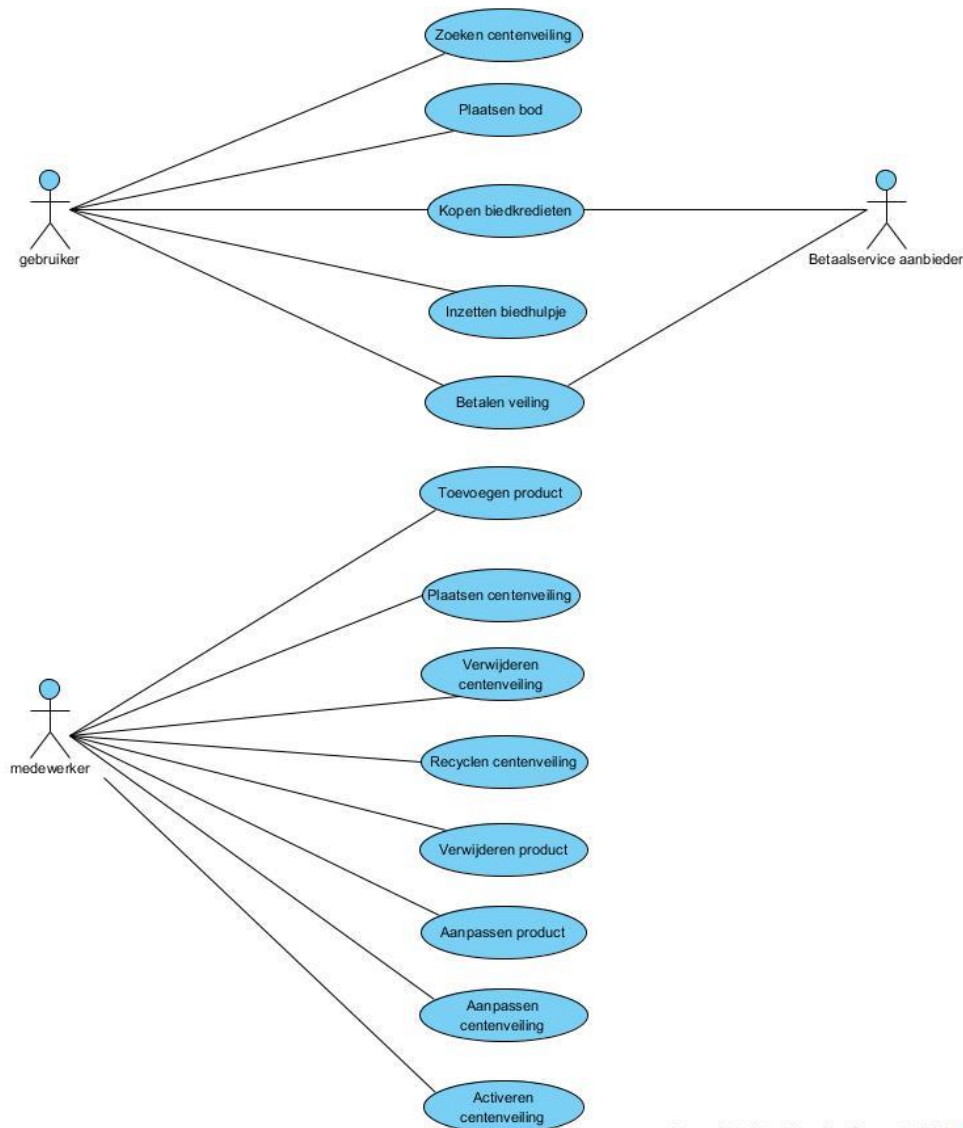
Bij het opstellen van de requirements is aangenomen dat het systeem gebruik gaat maken van het account systeem en de webwinkel binnen Zwoози en daar ook afhankelijk van zou zijn. Dit systeem wordt echter gelijktijdig naast centenveilingen opgebouwd en is het moeilijk om rekening met de andere systemen te houden bij het ontwerpen en realiseren van dit systeem. Het systeem wordt daarom afzonderlijk gebouwd. Eisen en wensen voor de accounts van de gebruiker zullen niet in dit project worden opgenomen, doordat ze elders bij het account systeem zijn opgenomen. Bij het realiseren zal het systeem echter gebruik gaan maken van het account systeem.

Het systeem is voor de betalingen afhankelijk van de betaalservice aanbieders. De services van de betaalaanbieders kunnen worden gebruikt doormiddel van een API. De koppeling tussen deze systeem en het eigen systeem zullen nog niet worden vastgelegd. Het systeem moet echter wel rekening houden met de toekomstige implementaties van de services.

3. Requirements

In dit hoofdstuk bespreken we de eisen en wensen waaraan het centenveilingsysteem moet voldoen. De eisen en wensen worden geplaatst op verschillende niveaus. Use Cases, business rules, overige requirements en kwaliteitsattributen. De Use cases zijn een aantal eisen en wensen die zijn gebundeld tot een proces. De processen in de Use Case diagrammen moet op elk moment door de bijbehorende actor kunnen worden uitgevoerd.. In paragraaf 3.1 wordt het use case model weergegeven waar de belangrijkste processen van het systeem worden weergegeven. Elke Use Case heeft een specificatie waarin de stappen van het proces wordt weergegeven. De eisen en wensen die niet in de use case specificaties zijn opgenomen worden onder het kopje overige requirements opgenomen. De Use Cases en de requirements hebben alleen betrekking op de functionaliteit die in een 1.0 versie tot stand moet komen. Voor alle aangegeven requirements, zie de requirements matrix in de Bijlage.

3.1 Use Case model



3.2 Use case specificaties

3.2.1 Use case Zoeken centenveiling

Use Case naam	Zoeken centenveiling	Use case flow
Omschrijving	De gebruiker wil een centenveiling zoeken	<pre> graph TD Start(()) --> U1(1. zoeken centenveiling gewenst) U1 --> D1{ } D1 --> U2(2. invoeren zoekcriteria) U2 --> D2{ } D2 -- "is gevonden?" --> J1((Ja)) J1 --> U3(3. Tonen resultaten) U3 --> End1((())) D2 --> A1(A1. Aangeven geen resultaat) A1 --> D3{opnieuw zoeken?} D3 -- Ja --> D1 D3 -- Nee --> End2((())) </pre> The flowchart illustrates the process of searching for a cent auction. It begins with the user selecting a desired cent auction (U1). A decision point follows to determine if search criteria are entered (D1). If criteria are entered (U2), another decision point checks if results are found (D2). If results are found (Ja), the system displays the results (U3) and the process ends. If no results are found (A1), a decision is made on whether to search again (D3). If the user chooses to search again (Ja), the process loops back to the criteria entry step (D1). If not (Nee), the process ends.
Primaire actor	gebruiker	
Pre condities	De gebruiker is ingelogd.	
Basisflow	- De gebruiker geeft aan dat hij een centenveiling wil zoeken. - Het systeem vraagt naar de gewenste zoekcriteria. De gebruiker voert deze data in. - Het systeem zoekt de actuele centenveilingen op die voldoen aan de zoekcriteria en toont deze aan de gebruiker.	
Na condities	Er is naar een centenveiling gezocht.	
Alternatieve flows	<u>A1: Geen resultaat:</u> In stap 3, als er geen centenveilingen gevonden worden die voldoen aan de zoekcriteria dan.... geeft het systeem weer dat er geen resultaten zijn gevonden. De gebruiker kiest of hij opnieuw wil zoeken. De Use Case stopt of gaat terug gaat naar stap 2.	

3.2.2 Use case Plaatsen bod

Use Case naam	Plaatsen bod	Use case flow
Omschrijving	De gebruiker wil een bod plaatsen op een centenvveiling	<pre> graph TD Start(()) --> U1(1. plaatsen bod gewenst) U1 --> D1{Genoeg kredieten?} D1 -- Nee --> A2_1(A2. Aangeven niet genoeg kredieten) A2_1 --> End1((())) D1 -- Ja --> U2(2+3. Plaatsen bod) U2 --> U3(4. bijwerken veiling) U3 --> D2{geboden in laatste 20 seconden?} D2 -- Ja --> A2_2(A2. Terugzetten aflooptijd) A2_2 --> End2((())) D2 -- Nee --> End3((())) </pre> The diagram illustrates the 'Plaatsen bod' (Place bid) use case. It begins with the actor '1. plaatsen bod gewenst' (1. place bid desired). A decision point 'Genoeg kredieten?' (Enough credits?) follows. If 'Nee' (No), the flow goes to 'A2. Aangeven niet genoeg kredieten' (A2. Indicate not enough credits) and ends. If 'Ja' (Yes), the flow proceeds to '2+3. Plaatsen bod' (2+3. Place bid), then to '4. bijwerken veiling' (4. update auction). Another decision point 'geboden in laatste 20 seconden?' (bid in last 20 seconds?) follows. If 'Ja' (Yes), the flow goes to 'A2. Terugzetten aflooptijd' (A2. Reset timeout) and ends. If 'Nee' (No), the flow ends directly.
Primaire actor	Gebruiker	
Pre condities	De gebruiker is ingelogd (als gebruiker de primaire actor is). De gewenste veiling is actueel. De gebruiker is niet de laatste beider	
Basisflow	1. De gebruiker geeft aan dat hij een bod wil plaatsen op de gewenste veiling. 2. Het systeem controleert of de gebruiker kan bieden. 3. Het systeem plaatst het bod en haalt de biedkosten van de gebruiker zijn account af. 4. Het systeem werkt de veiling bij en verhoogd de biedprijs. Het systeem werkt de bidhistorie bij en zet de gebruiker als leidende bidder.	
Na condities	Er is een bot geplaatst op een actuele centenvveiling.	
Alternatieve flow	<u>A1: Niet genoeg biedkredieten</u> In stap 2, als de gebruiker niet genoeg biedkredieten bezit, dan.... geeft het systeem aan dat de gebruiker niet genoeg biedkredieten bezit. Het bod kan niet worden geplaatst. De use case stopt. <u>A2: Bieden laatste 20 seconden</u> In stap 3, als het systeem de gebruiker als leidende bidder in de laatste 20 seconden aanstelt van de veiling, dan... zal het systeem de aflooptijd van de veiling met 15 seconden verhogen. Use Case eindigt. <u>A2: Bieden op een gesloten centenvveiling</u> In stap 2,als de gebruiker een bod probeert te plaatsen op een gesloten veiling, dan... al het systeem een melding weergeven dat het plaatsen niet mogelijk is. Use Case eindigt. <u>A2: Bieden als bidleider</u> In stap 2,als de gebruiker een bod probeert te plaatsen op een gesloten veiling waar hij de leidende bidder, dan... geeft het systeem een melding weer dat het plaatsen niet mogelijk is. Use Case eindigt.	

3.2.3 Use case Inzetten bidhulpje

Use Case naam	Inzetten bidhulpje	Use case flow
Omschrijving	De gebruiker wil een bidhulpje instellen voor een centenveiling	<pre> graph TD Start(()) --> U1(1. Inzetten bidhulpje gewenst) U1 --> J1{ } J1 --> U2(2 + 3. Opgeven kredieten) U2 --> J2{Niet genoeg kredieten?} J2 --> U3(4. activeren bidhulpje) J2 --> U4(A1. Aangeven niet genoeg kredieten) U3 --> End1((())) U4 --> J1 U4 --> End2((())) </pre> The flowchart illustrates the process of setting a bid helper. It starts with a start node leading to use case '1. Inzetten bidhulpje gewenst'. This leads to a junction point, then to '2 + 3. Opgeven kredieten'. A decision diamond 'Niet genoeg kredieten?' follows. If 'Yes', it leads to 'A1. Aangeven niet genoeg kredieten', which loops back to the junction point before '2 + 3. Opgeven kredieten'. If 'No', it leads to '4. activeren bidhulpje', which ends at a final node. A note 'Use case inzetten bidhulpje' is attached to the start.
Primaire actor	Gebruiker	
Pre condities	De gebruiker is ingelogd. De gewenste veiling is actief. De gebruiker moet minimaal 1 biedkrediet bezitten. De gebruiker heeft nog geen bidhulpje ingezet bij de gewenste veiling.	
Basisflow	- De gebruiker geeft aan dat hij een bidhulpje wil inzetten bij een centenveiling. - Het systeem vraagt hoeveel kredieten het bidhulpje mag besteden. - De gebruiker geeft het aantal biedkredieten op en geeft aan data hij het bidhulpje wil activeren. Het systeem controleert of de gebruiker in bezit is van genoeg kredieten. - Het systeem activeert het bidhulpje en toont een bevestiging aan de gebruiker. De gebruiker accepteert de bevestiging.	
Na condities	De gebruiker heeft een bidhulpje ingesteld voor een centenveiling.	
Alternatieve flow	<u>A1: onvoldoende biedkredieten</u> In stap 3 als de gebruiker niet genoeg kredieten tot zijn beschikking heeft, dan... geeft het systeem aan dat de gebruiker niet genoeg biedkredieten bezit en het bidhulpje niet kan worden geactiveerd. Het systeem vraagt de gebruiker om een nieuw aantal in te voeren. De gebruiker kiest of de use case verder gaat bij stap 3 of stopt.	

3.2.4 Use case Kopen biedkredieten

Use Case naam	Kopen biedkredieten	Use case flow
Omschrijving	De gebruiker wil biedkredieten aanschaffen voor zijn account	<pre> graph TD Start(()) --> U1[1. kopen biedkredieten gewenst] U1 --> J1{ } J1 --> U2[2+3. tonen en selecteren biedpakketten] U2 --> J2{ } J2 -- "Heeft premium account?" --> N2[Nee] N2 --> A2[A2. tonen geen premium account] A2 --> J1 J2 -- "Ja" --> J3{ } J3 -- "Premium pakket geselecteerd?" --> N3[Nee] N3 --> U4[4+5. Tonen en selecteren betaalmethoden] J3 -- "Ja" --> U4 U4 --> J4{ } J4 -- "Betaling gelukt?" --> N4[Nee] N4 --> A1[A1. tonen foutmelding betaalproces] A1 --> End1(()) J4 -- "Ja" --> U6[6. ontvangst bevestiging en toevoegen kredieten] U6 --> End2(()) </pre> The flowchart illustrates the process of purchasing bid credits. It starts with the user wanting to buy credits, followed by showing and selecting packages. A decision is made on whether the user has a premium account. If not, they are shown no premium packages. If yes, another decision is made on whether a premium package is selected. If not, they proceed to show and select payment methods. If yes, they proceed to show and select payment methods. A decision is then made on whether the payment was successful. If not, an error message is shown. If yes, the system receives confirmation and adds the credits to the account.
Primaire actor	Gebruiker	
Secundaire actor	Betaalservice aanbieder	
Pre condities	De gebruiker is ingelogd.	
Basisflow	- De gebruiker geeft aan dat hij biedcredits wil kopen. - Het systeem haalt de creditpakketten op en toont deze aan de gebruiker. - De gebruiker geeft aan dat hij een pakket met biedcredits wil kopen. De gebruiker legt deze aanvraag vast. - Het systeem haalt de beschikbare betaalmethoden op en toont deze aan de gebruiker. - De gebruiker kiest een betaalmethode. Het systeem stuurt de gebruiker naar de desbetreffende betaalservice aanbieder van de gekozen betaalmethode. - Het systeem ontvangt een bevestiging van desbetreffende betaalservice aanbieder dat het betalen is gelukt en legt dit vast. - Het systeem toont de bevestiging aan de gebruiker en voegt de gekochte biedcredits toe aan zijn account.	
Na condities	Er zijn biedcredits gekocht en toegevoegd aan de account van de gebruiker.	
Alternatieve flow	<u>A1: foutmelding betaalproces</u> In stap 5, als er bij het betaalproces iets verkeerd gaat, dan... krijgt het systeem de melding dat er iets fout is gegaan tijdens het betalen. Het systeem toont aan de gebruiker dat er iets fout is gegaan. Bij de betaling. Use Case stopt. <u>A2: Geen premium account</u> In stap 3 als de gebruiker een premium pakket kiest die alleen voor premium gebruikers is bedoeld zonder een premium account te bezitten, dan... geeft het systeem aan dat het gekozen pakket alleen voor premium gebruikers is bedoeld. Het systeem vraagt de gebruiker om een ander pakket te kiezen. Use case gaat verder bij stap 3.	

3.2.5 Use case Betalen centenveiling

Use Case naam	Betalen centenveiling	Use case flow
Omschrijving	De gebruiker wil een centenveiling betalen	<pre> graph TD Start(()) --> U1[1. Product kopen gewenst] U1 --> D1{ } D1 --> U2[2. Tonen betaal- en adresgegevens] U2 --> D2{adresgegevens bekend?} D2 -- Nee --> A1A2[A1+A2. invullen adresgegevens] A1A2 --> D1 D2 -- Ja --> D3{adresgegevens juist?} D3 -- Nee --> A1A2 D3 -- Ja --> U3[3+4. tonen en selecteren betaalmethode] U3 --> U4[5+6. ontvangen betaalbevestiging en afhandeling] U4 --> End((())) </pre> Powered By: Visual Paradigm Community Edition
Primaire actor	Gebruiker	
Secundaire actor	Betaalservice aanbieder	
Pre condities	De gebruiker moet zijn ingelogd. De gebruiker heeft aangegeven dat hij een product van een veiling wil kopen.	
Basisflow	- De gebruiker geeft aan dat hij een product van een veiling wil kopen. - Het systeem haalt de aankoop- en adresgegevens op en toont deze aan de gebruiker. Het systeem vraagt of deze gegevens kloppen. De gebruiker bevestigt dit. - Het systeem haalt de beschikbare betaalmethoden op en toont deze aan de gebruiker. - De gebruiker kiest een betaalmethode. Het systeem stuurt de gebruiker naar de desbetreffende betaalservice aanbieder van de gekozen betaalmethode. - Het systeem ontvangt een bevestiging van de desbetreffende betaalservice aanbieder dat het betalen is gelukt en legt dit vast. - Het systeem geeft aan dat het product binnen de aangegeven levertijd wordt bezorgt aan het aangegeven adres en geeft een bevestiging. De gebruiker accepteert de bevestiging.	
Na condities	De gebruiker heeft een betaling afgehandeld.	
Alternatieve flows	<u>A1: ontbrekende adresgegevens</u> In stap 2 als het systeem ziet dat er bij de gebruiker nog geen adresgegevens aanwezig zijn, dan... vraagt het systeem of de gebruiker zijn adresgegevens wil invullen. De gebruiker vult deze gegevens in. Use Case gaat verder bij stap 2. <u>A2: onjuiste adresgegevens</u> In stap 2 als de gebruiker aangeeft dat zijn adresgegevens onjuist zijn, dan... geeft het systeem de gebruiker de mogelijkheid om de adresgegevens te wijzigen. De gebruiker wijzigt deze data en geeft aan dat hij de data wil bijwerken. Het systeem werkt de adresgegevens bij. Use case gaat verder bij stap 2.	

3.2.6 Use case Plaatsen centenveiling

Use Case naam	Plaatsen centenveiling	Use case flow
Omschrijving	De medewerker wil een centenveiling plaatsen	<pre> graph TD Start(()) --> U1[1. Plaatsen centenveiling gewenst] U1 --> U2[2. invoeren veilinggegevens] U2 --> J1{ } J1 --> U3[3. Tonen beschikbare producten] U3 --> J2{nieuw product toevoegen?} J2 -- Ja --> A1[A1. Use Case Toevoegen nieuw product] A1 --> J1 J2 -- Nee --> U4[4. product selecteren] U4 --> U5[5. veiling bevestigen] U5 --> End((())) </pre> The flowchart illustrates the process of placing a cent auction. It begins with the user wanting to place an auction, followed by entering auction details. A decision point follows: if a new product is to be added, the user is directed to a sub-use case 'A1. Use Case Toevoegen nieuw product' before returning to the product selection step. The process concludes with product selection and confirmation of the auction.
Primaire actor	Medewerker	
Pre condities	De medewerker is ingelogd.	
Basisflow	1. De medewerker geeft aan dat hij een centenveiling wil plaatsen. 2. Het systeem vraagt naar de gewenste duur van de centenveiling. De medewerker voert deze data in. 3. Het systeem haalt de beschikbare producten op en toont deze aan de medewerker. 4. De gebruiker selecteert het product dat hij wil toevoegen aan de veiling. Het systeem legt deze aanvraag vast. 5. Het systeem toont de veilinggegevens samen met de productgegevens en vraagt om een bevestiging. De medewerker bevestigt de veilinggegevens.	
Na condities	Er is een centenveiling toegevoegd	
Alternatieve flows		

3.2.7 Use case Toevoegen product

Use Case naam	Toevoegen product	Use case flow
Omschrijving	De medewerker wil een product toevoegen aan het systeem.	<pre> graph TD Start(()) --> U1(1. Toevoegen product gewenst) U1 --> J1{ } J1 --> U2(2. Productgegevens invoeren) U2 --> J2{gegevens onjuist?} J2 -- Ja --> A1(A1. Tonen foutmelding onjuiste gegevens) A1 --> J1 J2 -- Nee --> U3(3. Afbeeldingen toevoegen) U3 --> U4(4. Bevestigen product) U4 --> End((())) </pre> Use Case: Toevoegen product Powered By: Visual Paradigm Community Edition
Primaire actor	Medewerker	
Pre condities	De medewerker is ingelogd.	
Basisflow	1. De medewerker geeft aan dat hij een product wil toevoegen. 2. Het systeem vraagt naar de productgegevens. De medewerker voert deze data in. 3. Het systeem vraagt naar afbeeldingen voor het product. De medewerker voegt deze afbeeldingen toe. 4. Het systeem toont het product met bijbehorende afbeeldingen en vraagt om een bevestiging. De medewerker bevestigt het product.	
Na condities	Er is een product toegevoegd aan het systeem	
Alternatieve flows	<u>A1. onjuiste gegevens:</u> in stap 2, als de gebruiker verkeerde data invoert, dan... geeft het systeem een foutmelding aan en vraagt de gebruiker om een om de juiste data in te voeren. in te voeren. De use case gaat verder bij stap 2.	

3.2.8 Use case Aanpassen product

Use Case naam	Aanpassen product	Use case flow
Omschrijving	De medewerker wil een product aanpassen aan het systeem.	<pre> graph TD Start(()) --> 1[1. Toevoegen product gewenst] 1 --> 2{ } 2 --> 2[2. Productgegevens invoeren] 2 --> 3{ } 3 -- "gegevens onjuist?" -- "Ja" --> A1[A1. Tonen foutmelding onjuiste gegevens] A1 --> 2 3 --> 4[3. Afbeeldingen toevoegen] 4 --> 5[4. Bevestigen product] 5 --> End((())) </pre> Use Case: Toevoegen product Powered By: Visual Paradigm Community Edition
Primaire actor	Medewerker	
Pre condities	De medewerker is ingelogd.	
Basisflow	1. De medewerker geeft aan dat hij een product wil aanpassen. 2. Het systeem vraagt naar het gewenste product. De medewerker geeft zijn keuze op. 3. Het systeem haalt het gekozen product op en toont deze aan de gebruiker. 4. De medewerker past de informatie aan die hij bij het product wil veranderen en geeft vervolgens aan dat hij de informatie wil opslaan. 5. Het systeem slaat de wijzigingen op.	
Na condities	Er is een product aangepast in het systeem	
Alternatieve flows	<u>A1. onjuiste gegevens:</u> in stap 2, als de gebruiker onjuiste data invoert, dan... geeft het systeem een foutmelding aan en vraagt de gebruiker om juiste data in te voeren. De use case gaat verder bij stap 4.	

3.2.9 Use case Verwijderen product

Use Case naam	Verwijderen product	Use case flow
Omschrijving	De medewerker wil een veilingproduct verwijderen.	<pre> graph TD Start(()) --> U1(1. verwijdering product gewenst) U1 --> U2(2. bevestigen verwijdering) U2 --> U3(verwijderen product) U3 --> End((())) </pre> Powered By: Visual Paradigm Community Edition
Primaire actor	Medewerker	
Secundaire actor		
Pre condities	De medewerker is ingelogd Er moet een veilingproduct aanwezig zijn.	
Basisflow	1. De medewerker geeft aan dat hij het gewenste veilingproduct wil verwijderen. 2. Het systeem controleert of het product zich niet in een actuele veiling bevindt. 3. Het systeem geeft aan dat de centenveiling wordt verwijderd.	
Na condities	De centenveiling wordt verwijderd.	
Alternatieve flow	<u>A1. Product wordt nog gebruikt:</u> in stap 2, als het product zich nog in een actuele veiling bevindt, dan... Geeft het systeem aan dat het product nog wordt gebruikt en niet kan worden verwijderd. Use case stopt.	

3.2.10 Use case Verwijderen centenveiling

Use Case naam	Verwijderen centenveiling	Use case flow
Omschrijving	De medewerker wil een centenveiling verwijderen.	<pre> graph TD Start(()) --> U1(1. Verwijderen velling gewenst) U1 --> U2(2. stopzetten centenveiling) U2 --> U3(3. Verwijderen centenveiling) U3 --> End((())) </pre> Use Case Verwijderen centenveiling Powered By Visual Paradigm Community Edition
Primaire actor	Medewerker	
Secundaire actor		
Pre condities	De medewerker is ingelogd Er moet een centenveiling aanwezig zijn.	
Basisflow	- De medewerker geeft aan dat hij de gewenste centenveiling wil verwijderen. - Het systeem vraagt de medewerker of hij de centenveiling wil stoppen. De gebruiker accordeert. - Het systeem geeft aan dat de centenveiling wordt verwijderd.	
Na condities	De centenveiling wordt verwijderd.	
Alternatieve flow	<u>A1. Centenveiling loopt nog:</u> in stap 3,als het systeem het product probeert te verwijderen wanneer die nog actief is, dan... moet het systeem de biedingen retourneren aan de gebruikers en de gebruikers informeren dat de centenveiling wordt verwijderd. Use case stopt.	

3.2.11 Use case Recyclen centenveiling

Use Case naam	Recyclen centenveiling	Use case flow
Omschrijving	De gebruiker wil de gewenste centenveiling recyclen en opnieuw aanbieden.	<pre> graph TD Start(()) --> Step1(1. Recyclen centenveiling gewenst) Step1 --> Step2(Invoeren startdatum en duur) Step2 --> Step3(3. plaatsen recyclede centenveiling) Step3 --> End((())) </pre> Use Case Recyclen centenveiling Powered By Visual Paradigm Community Edition
Primaire actor	Medewerker	
Secundaire actor		
Pre condities	De medewerker moet zijn ingelogd. De gewenste centenveiling moet zijn geplaatst.	
Basisflow	- De medewerker geeft aan dat hij de gewenste veiling wil recyclen. - Het systeem vraagt de medewerker om de nieuwe startdatum, starttijd en duur in te voeren. De medewerker voert deze data in. - Het systeem plaatst een nieuwe instantie van de centenveiling met de status inactief.	
Na condities	De gewenste centenveiling is gerecycled. Er is een nieuwe centenveiling geplaatst.	

3.2.12 Use case Aanpassen centenveiling

Use Case naam	Aanpassen centenveiling	Use case flow
Omschrijving	De medewerker wil de gewenste centenveiling recylen en opnieuw aanbieden.	<pre> graph TD Start(()) --> Step1([1. Aanpassen centenveiling]) Step1 --> Step2([2. Invoeren nieuwe duur]) Step2 --> Step3([3. selecteren product]) Step3 --> Step4([4. Aanpassingen vastleggen]) Step4 --> End((())) </pre>
Primaire actor	Medewerker	
Pre condities	De medewerker moet zijn ingelogd. De gewenste centenveiling moet zijn geplaatst. De gewenste centenveiling is inactief.	
Basisflow	- De medewerker geeft aan dat hij de gewenste veiling wil aanpassen. - Het systeem vraagt de medewerker om de nieuwe duur in te voeren. De medewerker voert deze data in. - Het systeem vraagt hem om een product te selecteren. De medewerker vult deze data in. - Het systeem past de gewenste centenveiling aan met de nieuwe data.	
Na condities	De gewenste centenveiling is aangepast	

3.2.13 Use case activeren centenveiling

Use Case naam	Activeren centenveiling	Use case flow
Omschrijving	De medewerker wil de gewenste centenveiling recylen en opnieuw aanbieden.	<pre> graph TD Start(()) --> Step1([1. Activeren centenveiling gewenst]) Step1 --> Step2([2. bevestiging vragen activeren]) Step2 --> Step3([3+4. veranderen status centenveiling en starten]) Step3 --> End((())) </pre>
Primaire actor	Medewerker	
Pre condities	De medewerker moet zijn ingelogd. De gewenste centenveiling staat op inactief	
Basisflow	- De medewerker geeft aan dat hij de gewenste veiling wil activeren. - Het systeem vraagt om een bevestiging voor het activeren. De medewerker bevestigt dit. - Het systeem Het systeem verandert de status naar actief en laat de veiling starten.	
Na condities	De gewenste centenveiling is geactiveerd. De gewenste veiling is gestart.	

3.3 Aanvullende requirements

Aanvullende requirements zijn requirements van het systeem die niet in de Use Cases aan bod komen.

3.3.1 Business rules

- Een product moet tot 24 uur na het einde van een centenveiling beschikbaar zijn voor aankoop, mits er meerdere exemplaren aanwezig zijn.
- Als een gebruiker op een centenveiling heeft geboden, moet hij het product goedkoper (met korting) kunnen kopen aan de hand van hoe vaak hij heeft geboden.
- De korting op een product moet berekend worden door 40 cent keer het aantal biedkredieten besteed.
- Een gewonnen centenveiling moet aan de winnende bidder het product voor de biedprijs verkopen.
- Als een gebruiker het product van een gewonnen veiling niet binnen 24 uur koopt, dan vervalt zijn recht om het product voor de biedprijs te kopen.
- Als een beheerder een veiling forcerend gestopt heeft, dan moet de veiling na 10 minuten worden verwijderd. Alle gebruikers die hebben geboden op de veiling moeten hun biedcredits terugkrijgen.
- Als er in de laatste 20 seconden geboden wordt. Dan moet de aflooptijd van de centenveiling met 15 seconden worden verlengd.
- De aflooptijd van een veiling kan niet worden verlengd boven de laatste 30 seconden.
- De totaalprijs voor biedcredits moet berekend worden door 75 cent per biedkrediet te rekenen.
- Een gebruiker mag alleen een biedhulpje inzetten als hij genoeg biedcredits bezit.
- Bij het inzetten van een biedhulpje moet het aantal biedcredits die hij mag besteden in aantallen worden weergegeven.
- Het biedhulpje mag alleen bieden zolang hij nog genoeg biedcredits tot zijn beschikking.
- Een centenveiling moet worden afgesloten zodra de aflooptijd de 0 heeft bereikt.
- De levertijd moet altijd in dagen worden weergegeven.
- Bedragen moeten in euro's worden weergegeven.

3.3.2 *Gebruiker requirements*

De requirements die hier worden weergegeven zijn de eisen die gebruikers stellen die niet terugkomen in de Use Cases.

- De gebruiker wil dat een biedhulpje in zijn plaats kan bieden op een veiling.
- De gebruiker wil in een centenveiling inzien.
- De gebruiker wil het product zien bij een centenveiling.
- De gebruiker wil een overzicht van zijn biedhulpjes kunnen zien.
- De gebruiker wil een overzicht zien van actieve centenveilingen.
- De gebruiker wil de details van een centenveiling inzien.
- De gebruiker wil een overzicht zien van gesloten centenveilingen.
- De gebruiker wil de details van een gesloten centenveiling inzien.
- De gebruiker wil bij een centenveiling de huidige aflooptijd zien.
- De gebruiker wil bij een centenveiling de leidende bieder zien.
- De gebruiker wil bij een centenveiling de biedhistorie zien.
- De gebruiker wil zijn eigen biedcredits aankopen zien.
- De gebruiker wil zijn eigen biedcredits bestedingen zien.
- De gebruiker wil de gewonnen centenveilingen bezichtigen.
- De gebruiker wil het product van een centenveiling direct kunnen kopen.
- De gebruiker wil de aankoop van een veiling betalen.
- De gebruiker wil zijn biedcredits totaal zien.
- De medewerker wil het aantal biedcredits bij een creditpakket kunnen aanpassen.
- De medewerker wil de waarde van een biedcredit kunnen aanpassen.
- De medewerker wil de prijs van een biedpakket kunnen aanpassen.

3.3.3 *Software requirements:*

- Het systeem moet een overzicht kunnen tonen van de biedhulpjes van de gebruiker.
- Het systeem moet een overzicht van actieve centenveilingen kunnen tonen, gesorteerd op aflooptijd.
- Het systeem moet een overzicht van gesloten centenveilingen kunnen tonen, gesorteerd op de sluitdatum.
- Het systeem moet het biedhulpje laten bieden op de bijbehorende centenveiling zolang hij biedcredits tot zijn beschikking heeft.
- Het systeem moet de gebruiker de mogelijkheid geven om het aantal biedcredits bij het biedhulpje op te geven.
- Het systeem moet controleren dat het aantal biedcredits niet negatief kan worden ingevoerd door de gebruiker.
- Het systeem moet aangeven dat een premiumpakket alleen voor premium gebruikers is bedoeld.
- Het systeem moet per bieder in de biedhistorie de schermnaam, biedprijs en tonen.
- Het systeem moet bij de biedhistorie de laatste 10 bidders tonen.
- Het systeem moet bij de aankoopgegevens de productnaam, aankoopprijs, verzendkosten en het totaalbedrag tonen.
- Het systeem moet bij de adresgegevens de straatnaam, huisnummer, postcode, land en telefoonnummer weergeven.
- Het systeem moet de tijdelijke eindtijd van een centenveiling vaststellen wanneer deze wordt geactiveerd.
- Het systeem moet de actuele aflooptijd van een centenveiling tonen.
- Het systeem moet aangeven dat een centenveiling is gesloten.
- Het systeem moet het biedrecht bij een centenveiling ontnemen wanneer een centenveiling is gesloten.
- Het systeem moet de mogelijkheid bieden om meerdere afbeeldingen bij een product toe te voegen.
- Het systeem moet bij het verwijderen van een centenveiling de gebruikers informeren over de verwijdering voordat de veiling wordt verwijderd.
- Het systeem moet bij het verwijderen de bestede biedcredits retourneren aan de bidders.
- Het systeem moet controleren dat de duur van een centenveiling bij het aanmaken niet negatief mag zijn.
- Het systeem moet in het overzicht per centenveiling de productnaam, productafbeelding, aflooptijd, huidige biedprijs en de leidende bieder van de centenveiling tonen.
- Het systeem moet bij een product de naam, productomschrijving, winkelprijs, levertijd en afbeeldingen weergeven.
- Het systeem moet bij de inzage van de details van een centenveiling de productgegevens en de biedlijst erbij tonen.
- Het systeem moet de gebruiker een product direct kunnen laten kopen.
- Het systeem moet een korting berekenen bij het direct kopen als de gebruiker op de centenveiling heeft geboden.
- Het systeem moet de winnende gebruiker informeren dat hij de veiling heeft gewonnen.
- Het systeem moet bij het aanmaken van de aankoop de huidige biedprijs als koopprijs neerzetten.
- Het systeem moet controleren dat aankopen na 24 uur vervallen als ze nog niet betaald zijn.
- Het systeem moet verliezende gebruikers informeren dat zij een veiling niet hebben gewonnen.

- Het systeem moet een centenveiling die is gestart zichtbaar maken.
- Het systeem moet de prijs van een biedpakket aanpasbaar maken voor de medewerkers.
- Het systeem moet de waarde van biedkredieten aanpasbaar maken voor medewerkers.
- Het systeem moet het aantal biedkredieten bij een biedpakket aanpasbaar maken voor medewerkers.
- Het systeem moet bij de aankoop van de winnende gebruiker de biedprijs als aankoopprijs stellen.

3.3.4 *Kwaliteitseigenschappen*

- Snelheid: Het verschil tussen de aflooptijd van een centenveiling onder alle participanten moet minder dan 0,5 seconde zijn. Dit mag bij drukte maximaal 1 seconde duren.
- Snelheid: De duur van het plaatsen van een bod moet in minder dan een halve seconde gebeuren. Bij drukte mag het maximaal 1 seconde duren.
- Juistheid: Van de 10.000 gebruikers mag het plaatsen van een bod maximaal bij 50 gebruikers niet worden uitgevoerd.
- Begrijpelijkheid: Een ervaren internetgebruiker moet binnen 2 minuten kunnen bieden nadat zij zich hebben geregistreerd. Een onervaren internetgebruiker moeten binnen 5 minuten kunnen bieden na registratie.
- Leerbaarheid: 80% van de medewerkers moet na 10 minuten uitleg zelfstandig het systeem kunnen bedienen.
- Bij het opleveren van het proof of concept moet het systeem aan 75% van de requirements voldoen.

4. Aanvullend informatie

Begrip	Extra informatie
Actieve veilingen	Een veiling die nog loopt is.
Actuele aflooptijd	De huidige tijd dat het duurt voordat de veiling is afgelopen.
Belanghebbenden	Personen, groepen of bedrijven die baat bij het systeem hebben.
Betaalmethode	Manieren om te betalen zoals IDEal, Paypal en creditcard.
Betaalservice aanbieder	Aanbieder van een betaalservice zoals IDEal, Paypal creditcard.
Biedhistorie	Een lijst van bidders die op de veiling hebben geboden op volgorde van welke bidder het laatst heeft geboden.
Biedhulpje	Een functionaliteit die de gebruiker kan inzetten om voor hem te bieden, ook al is de gebruiker afwezig.
Centenveiling	Een veiling waar je kleine bedragen betaalt om te kunnen bieden. De prijs van het bieden gaat met één cent per bod omhoog.
Productgegevens	Bestaan uit productnaam, productomschrijving, winkelprijs en levertijd van een product.
Requirement	Een wens of eis die een belanghebbende bij het systeem heeft.
Veilinggegevens	Bestaan uit product, veilingnaam, biedhistorie, leidende bidder en het huidige bedrag.

5. Bijlage: requirements matrix

Gebruikerrequirements

REQUIREMENT NR	REQUIREMENT OMSCHRIJVING	STAKEHOLDER	AFKOMST	PRIORITEIT	MOSCOW
	De gebruiker wil biedkredieten kunnen aanschaffen.	Marketing/directive	brainstormsessie/123bied.nl		
	De gebruiker wil het product van een centenveiling zien.	Marketing/directive	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil de actuele aflooptijd van een centenveiling zien.	Marketing/directive	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil de leidende bidder bij een centenveiling zien	Marketing/directie	brainstormsessie/123bied.nl		
	De gebruiker wil de huidige biedprijs van een veiling kunnen zien.	Marketing/directive	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil de biedhistorie bij een veiling kunnen zien.	Marketing/directie	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil een bod kunnen plaatsen op een actuele centenveiling.	Marketing/directie	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil een product direct kunnen kopen.	Marketing/directie	brainstormsessie	1	Must have
	De gebruiker wil direct een product kopen dat geveild wordt.	Marketing/directie	brainstormsessie	1	Must have
	De gebruiker wil bij een actieve veiling de aflooptijd zien.	Marketing/directie	brainstormsessie/123bied.nl	1	Must have
	De gebruiker wil bij een product de winkelprijs, levertijd, productomschrijving en foto's zien.	Marketing/directie	123bied.nl	1	Must have
	De gebruiker wil continu kunnen zien wat de status is van de centenveiling.	Marketing	123bied.nl	1	Must have
	De gebruiker wil een overzicht zien van zijn biedhulpjes.	Marketing	123bied.nl	1	Must have
	De medewerker wil een centenveiling kunnen plaatsen op de website.	Functioneelbeheerder	brainstormsessie	1	Must have
	De medewerker wil bij het plaatsen van een centenveiling een nieuw product kunnen toevoegen.	Marketing	brainstormsessie	1	Must have
	De medewerker wil bij het plaatsen van een centenveiling een bestaand product kunnen selecteren.	Functioneelbeheerder	brainstormsessie	1	Must have
	De medewerker wil bij een product afbeeldingen kunnen	Functioneelbeheerder	brainstormsessie	1	Must have

	uploaden.				
	De medewerker wil een centenveiling kunnen verwijderen.	Functioneelbeheerder	brainstormsessie	1	Must have
	De medewerker wil een centenveiling opnieuw kunnen plaatsen met hetzelfde product nadat deze is afgerond.	Functioneelbeheerder	brainstormsessie	1	Must have
	De medewerker wil een centenveiling in een wachtrij zetten die automatisch centenveilingen start.	Functioneelbeheerder	brainstormsessie	1	Must have
	De gebruiker wil zijn account kunnen opwaarderen naar een premium account.	Marketing/directie	brainstormsessie	1	Should have
	De gebruiker wil een overzicht zien van actieve centenveilingen.	Marketing/directie	123bied.nl	1	Must have
	De gebruiker wil simpel kunnen deelnemen aan centenveilingen.	Marketing/directie	brainstormsessie	1	Should have
	De gebruiker wil de biedhistorie bij een veiling kunnen inzien.	Marketing/directie	123bied.nl	1	Should have
	De medewerker wil promoties kunnen plaatsen op de website.	Functioneelbeheerder	brainstormsessie	1	Should have
	De gebruiker wil een biedhulpje automatisch laten bieden op een centenveiling.	Marketing/directie	brainstormsessie	2	should have
	De gebruiker wil biedkredieten verdienen voor het uitnodigen van vrienden.	Marketing/directie	brainstormsessie	2	Must have
	De gebruiker wil bij het aanschaffen van kredieten de mogelijkheid hebben om te kunnen betalen met creditcard, Ideal en paypal.	Marketing/directie	brainstormsessie	2	Must have
	De gebruiker wil vrienden kunnen uitnodigen via social media.	Marketing/directie	brainstormsessie	3	should have
	De gebruiker wil een veiling toevoegen aan de lijst met gevolgde veilingen.	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil de biedkredieten die hij besteed heeft aan het gewenste product kunnen inruilen voor korting op het product.	Marketing/directie	123bied.nl	2	Should have
	De gebruiker wil zien hoeveel biedkredieten hij heeft besteed bij een centenveiling.	Marketing/directie	brainstormsessie	2	Should have

	De gebruiker wil zien hoe vaak een gebruiker op de veiling heeft geboden.	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil een gewonnen veiling kunnen betalen via IDEal, paypal en creditcard.	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil Zwoozi kredieten kunnen omwisselen voor biedkredieten.	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil biedkredieten kunnen inwisselen voor Zwoozi kredieten.	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil centenveilingen laten zoeken op basis van de ingevoerde selectiecriteria	Marketing/directie	brainstormsessie	2	Should have
	De gebruiker wil zijn eigen biedhistorie kunnen zien	Marketing	123bied.nl	2	Should have
	De gebruiker wil zijn betaalgeschiedenis kunnen zien.	Marketing	123bied.nl	2	Should have
	De gebruiker wil een foto uploaden van zichzelf met het gewonnen product.	Marketing	123bied.nl	2	should have
	De medewerker wil statistieken kunnen genereren.	Functioneelbeheerder	brainstormsessie	2	Should have
	De gebruiker wil een biedhulpje inzetten bij een centenveiling door het gewenst aantal biedkredieten op te geven.	Marketing/directie	123bied.nl	2	Could have
	De gebruiker wil ingelicht worden wanneer een product uit zijn wishlist wordt geveild.	Marketing/directie	brainstormsessie	2	Could have
	De gebruiker wil het uiterlijk van de website personaliseren.	Marketing/directie	brainstormsessie	2	Could have
	De gebruiker wil een kennisbank kunnen raadplegen.	Marketing/directie	brainstormsessie	3	Should have
	De gebruiker wil een FAQ kunnen raadplegen.	Marketing/directie	brainstormsessie	3	Should have
	De gebruiker wil de handleiding kunnen bezichtigen waarin het gebruik van het systeem wordt toegelicht.	Marketing/directie	brainstormsessie	3	Should have
	De gebruiker wil tips en trucs over centenveilingen bekijken.	Marketing/directie	123bied.nl	3	Should have

	De gebruiker wil gesloten centenveilingen kunnen inzien.	Marketing/directie	123bied.nl	3	Could have
--	--	--------------------	------------	---	------------

Softwarerequirements

REQUIREMENT NR	REQUIREMENT OMSCHRIJVING	STAKEHOLDER	AFKOMST	PRIORITEIT
	Het systeem moet een overzicht kunnen weergeven van de bidhulpjes van de gebruiker.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet het ingezette bidhulpje laten bieden op 2 seconden voor de aflooptijd van de veiling.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet zorgen dat het bidhulpje op de toegewezen veiling biedt zolang hij kredieten bezit.	directie	brainstormsessie	2
	Het systeem moet controleren dat het aantal bidcredits niet negatief kan worden bij de gebruiker.	directie	brainstormsessie	1
	Het systeem moet aanduiden of een bidpakket alleen voor premium gebruikers is bedoeld.	directie	brainstormsessie	2
	Het systeem moet per bieder in de bidhistorie de schermnaam, biedprijs en het aantal keer dat hij heeft geboden tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij de bidhistorie de laatste 10 bidders tonen.	directie	brainstormsessie	1
	Het systeem moet bij de aankoopgegevens de productgegevens, aankoopprijs, verzendkosten en het totaalbedrag tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij de adresgegevens de straatnaam, huisnummer, postcode, land en telefoonnummer weergeven.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet de centenveiling starten als de starten wanneer deze door een medewerker wordt geactiveerd.	directie	brainstormsessie	1
	Het systeem moet de actuele aflooptijd van actuele centenveiling tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet aangeven dat een centenveiling is gesloten.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet controleren dat een gebruiker alleen kan bieden als hij genoeg bidcredits heeft.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet ondersteuning bieden om meerdere afbeeldingen bij een product te kunnen toevoegen.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet controleren dat een bod alleen kan worden toegevoegd als de centenveiling actueel is.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet de centenveiling na 10 minuten verwijderen na het stopzetten.	directie	brainstormsessie	2
	Het systeem moet bij het stopzetten van een centenveiling de gebruikers informeren.	directie	brainstormsessie	2
	Het systeem moet bij het stopzetten van een centenveiling de bidkredieten retourneren die die gebruikers hebben besteed.	directie	brainstormsessie	2
	Het systeem moet bij het activeren van een centenveiling de eindtijd bepalen op basis	directie	brainstormsessie	1

	van de duur.			
	Het systeem moet een overzicht van centenveilingen kunnen weergeven op basis van de opgegeven categorie	123bied.nl	brainstormsessie/123bied.nl	2
	Het systeem moet controleren dat het aantal dagen van de levertijd niet negatief mag zijn.	directie	brainstormsessie	1
	Het systeem moet controleren dat een bedrag niet negatief mag zijn.	directie	brainstormsessie	1
	Het systeem moet een overzicht van actuele centenveilingen kunnen tonen, gesorteerd op aflooptijd.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet een overzicht van gesloten veilingen kunnen tonen, gesorteerd op de sluitdatum.	123bied.nl	brainstormsessie/123bied.nl	2
	Het systeem moet per centenveiling in het overzicht de productnaam, productafbeelding, aflooptijd, huidige biedprijs en de leidende bieder tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een product de naam, productomschrijving, winkelprijs, levertijd en afbeeldingen weergeven.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet de levertijd van een product in dagen aangegeven.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een volledige inzage van een centenveiling de productgegevens en de biedlijst erbij tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	het systeem moet een kennisbank ondersteunen.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet tips en trucs weergeven op de webpagina.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet de handleiding aan gebruikers weergeven.	directie	brainstormsessie	3
	Het systeem moet een FAQ(frequently Asked Questions) ondersteunen.	123bied.nl	brainstormsessie/123bied.nl	3
	Het systeem moet het plaatsen van promoties ondersteunen.	123bied.nl	brainstormsessie/123bied.nl	2
	Het systeem moet de gewonnen veilingen van een gebruiker tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet de gekochte producten van een gebruiker tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet een centenveiling kunnen markeren als 'gevolgd'.	directie	brainstormsessie	3
	Het systeem moet de gebruiker inlichten als een product uit zijn wishlist wordt geveild.	directie	brainstormsessie	3
	Het systeem moet gebruikers de mogelijkheid geven om vrienden te kunnen uitnodigen via social media.	directie	brainstormsessie	2
	Het systeem moet bij een product de productnaam, winkelprijs, levertijd, productomschrijving en afbeeldingen tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een veiling het product samen met de aflooptijd, biedhistorie, huidige biedprijs en de naam van de laatste bieder tonen.	123bied.nl	brainstormsessie/123bied.nl	1

	Het systeem moet bij het plaatsen van een bod de gebruiker als nieuwe bidleider aanstellen en bovenaan de bidhistorie neerzetten	directie	brainstormsessie/123bied.nl	1
	Het systeem moet de aflooptijd van een centenveiling met 15 seconden verhogen wanneer er in de laatste 20 seconden een nieuw bod is uitgebracht.	directie	brainstormsessie/123bied.nl	1
	Het systeem moet het aanschafproces ondersteunen wanneer een gebruiker besluit om een product direct te kopen bij een veiling.	123bied.nl	brainstormsessie	1
	Het systeem moet de gevraagde centenveilingen zoeken aan de hand van de opgegeven zoekcriteria.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet 24 uur na het einde van een centenveiling de mogelijkheid om het veilingproduct direct te kunnen kopen weghalen	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij het aanschaffen van een veilingproduct korting toepassen als de gebruiker geboden heeft op de bijbehorende centenveiling.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een gesloten centenveiling tonen dat de veiling gesloten is in plaats van de aflooptijd.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een gesloten centenveiling de mogelijkheid om te kunnen bieden weghalen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij het uitnodigen van vrienden via social media de gebruiker belonen met biedkredieten.	directie	brainstormsessie	2
	Het systeem moet bij aanschafprocessen ondersteuning bieden om met creditcard, paypal en iDeal te kunnen betalen	directie	brainstormsessie	2
	Het systeem moet gebruikers doorsturen naar het opwaardeerproces van het accountsysteem, als zij een premium account willen.	directie	brainstormsessie	2
	Het systeem moet aantonen hoeveel biedkredieten een gebruiker heeft besteed aan een centenveiling.	directie	brainstormsessie	2
	Het systeem moet een overzicht tonen met de actuele bidhulpjes van een gebruiker.	123bied.nl	brainstormsessie/123bied.nl	2
	Het systeem moet de eigen bidhistorie van een gebruiker kunnen tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet de betaalgeschiedenis van de gebruiker kunnen tonen.	123bied.nl	brainstormsessie/123bied.nl	1
	Het systeem moet bij een gewonnen veiling de mogelijkheid geven om een foto te uploaden.	123bied.nl	brainstormsessie/123bied.nl	2
	Het systeem moet de ondersteuning bieden om statistieken te generen over centenveilingdata.	Lead developer	brainstormsessie	3

	Het systeem moet de gebruiker de mogelijkheid geven om tot 24 uur nadat de veiling is gesloten het product te kopen.	123bied.nl	brainstormsessie/123bied.nl	1
--	--	------------	-----------------------------	---

Analyse

Discipline Document

Centenveiling systeem

Project: Centenveilingsysteem voor Zwoozi
Date: 22-06-2016
Owner: Daveron van der Ham
Version: 3.0

Document Historie

Revisie Historie

Version	Revisie datum	Samenvatting van veranderingen
0.1	20-06-2016	Initiële versie
	21-06-2016	Packages toegevoegd en het eerste opzet analyse klassendiagram.
	22-06-2016	Klassendiagram verder uitgewerkt op basis van use cases en requirements.
1.0	23-06-2016	Eerste sequentie diagrammen toegevoegd.
2.0	01-07-2016	Klassendiagram aangepast
2.2	01-07-2016	Sequentie diagrammen aangepast
2.3	03-09-2016	Controllers uit de diagrammen gehaald
3.0	05-09-2019	Context view en functioneel view toegevoegd aan omgeving

Distributie

Dit document is bedoelt voor:

Name	Functie	Date of issue	Version
Nick Verheij	Producteigenaar		
Koen Brouwer	Projectleider/programmeur		
Florian de Groot	Begeleider/programmeur		

Inhoudsopgave

1.	INTRODUCTIE	1
2.	OMGEVING	2
3.	ANALYSE PAKKETTEN.....	4
4.	GLOBALE ANALYSE KLASSENDIAGRAM	5
5.	ANALYSE USE CASE REALISATIE	6
5.1.	SD plaatsen bod.....	6
5.2.	SD inzetten biedhulpje	6
5.3.	Zoeken centenveiling.....	7
5.4.	Betalen veiling.....	7
5.5.	Kopen biedpakket.....	8
5.6.	Toevoegen product.....	8
5.7.	Plaatsen centenveiling.....	9
5.8.	Verwijderen centenveiling.....	9
5.9.	Aanpassen product.....	10
5.10.	Verwijderen product.....	10
5.11.	Recyclen centenveiling.....	10
5.12.	Aanpassen centenveiling	11
5.13.	Activeren centenveiling	11

1. Introductie

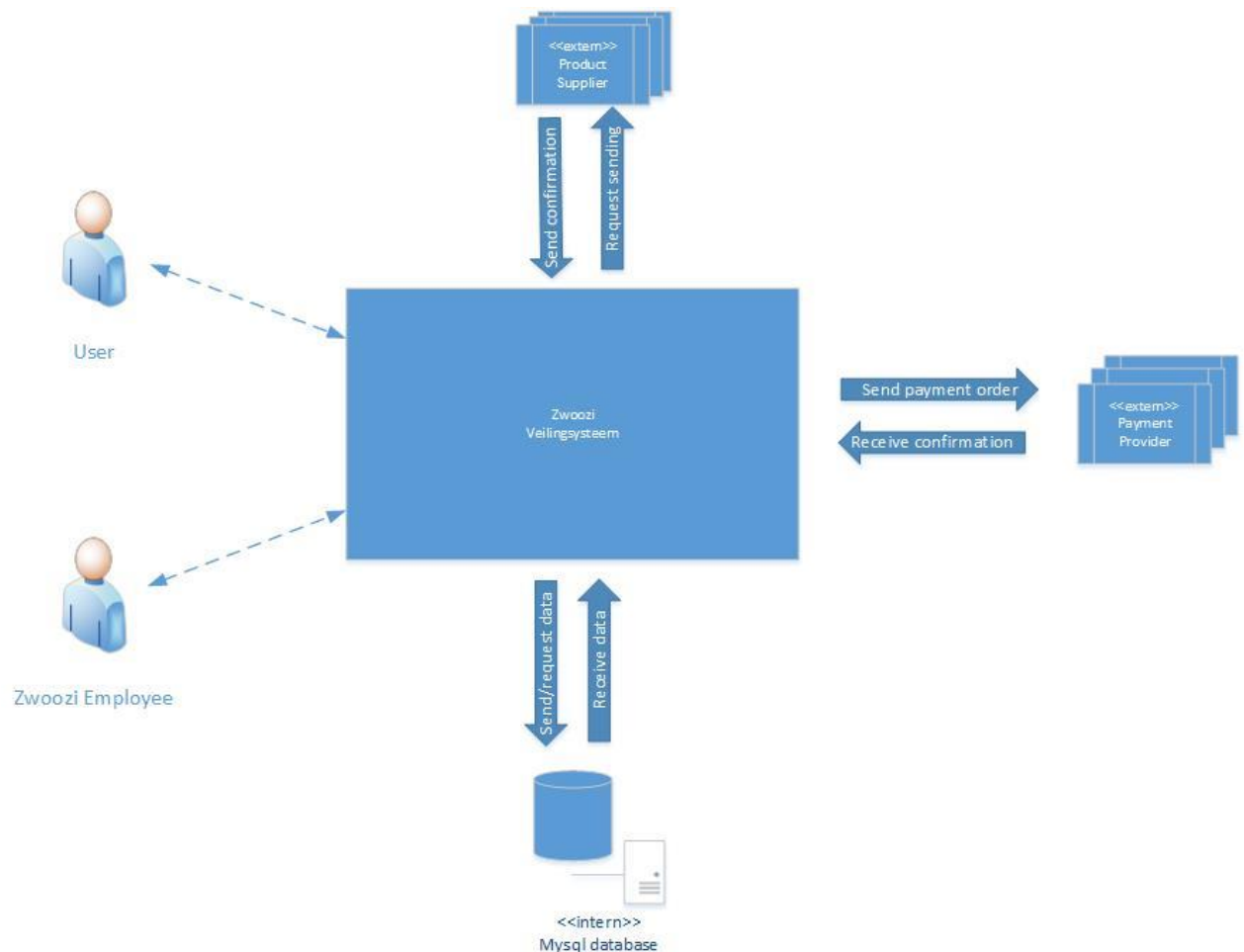
Dit document is gericht op het overbrengen van de functionaliteiten en de applicatiestructuur voor de belanghebbenden, afkomstig uit de eisen en wensen van het software-requirements-specificatie. Het document geeft de globale structuur van het systeem aan samen met de belangrijkste gedefinieerde functionaliteiten. De omgeving geeft aan waar het systeem zich zal gaan bevinden. De sequentiediagrammen geven de uitwerking van een Use Case weer doormiddel van de functionaliteiten die in de globale structuur staan aangegeven. Voor het verduidelijken van de weergave van de structuur, wordt er gebruik gemaakt van UML diagrammen. De analyse diagrammen geven een globale weergave van de belangrijkste klassen weer zonder te diep in details te treden.

2. Omgeving

Het centenveilingsysteem is een systeem dat zich zal bevinden binnen het platform Zwoози. Dit houdt in dat centenveilingen in dezelfde omgeving als Zwoози komt te draaien. Zwoози bevindt zich in een omgeving met twee verschillende gebruikers. Deze eindgebruikers zijn:

- Consument(User)
- Medewerker(Zwoози Employee)

Zwoози maakt gebruik van een interne database om data op te halen en op te slaan. Het platform gaat gebruik maken van leveranciers die de producten voor de veilingen zullen aanleveren. Voor het afhandelen van betalingen gaat Zwoози gebruik maken van externe betaalservices als iDeal en Paypal. In het onderstaande diagram wordt een context view weergegeven van het platform Zwoози.



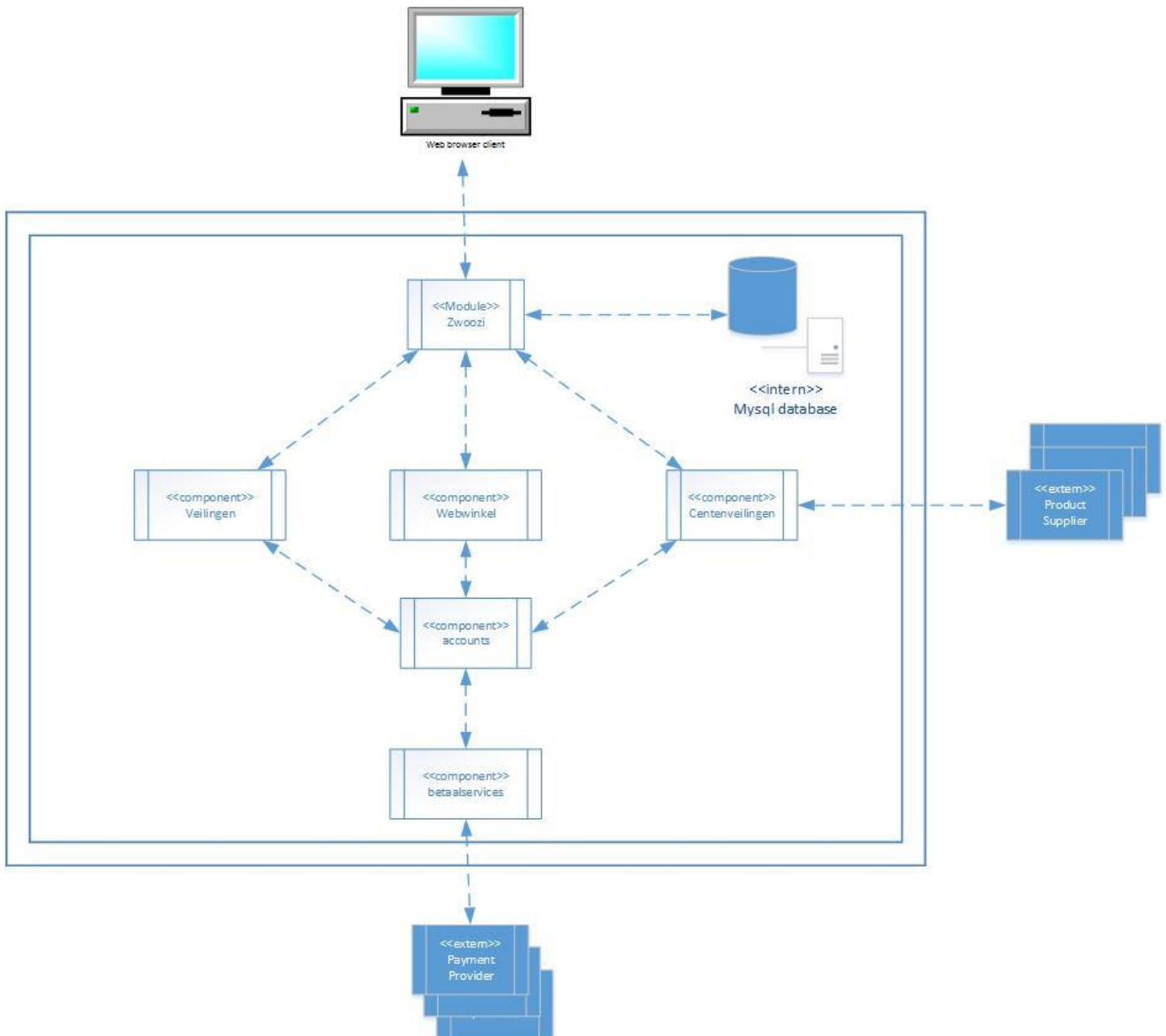
Figuur 1. Context view van Zwoози.

In het bovenstaande diagram wordt Zwoози als een blackbox weergegeven. Het diagram geeft doormiddel van een globale weergave de gebruikers weer en de externe aanbieders waarmee het systeem moet communiceren. Het diagram geeft niet weer wat er binnen Zwoози bevindt. In het volgende diagram (figuur 2) bevindt zich een aantal componenten. De gebruiker maakt gebruik van een web browser om aanvragen te sturen naar Zwoози. Aan de hand van deze aanvragen kan het systeem bepalen welk component hij moet gebruiken voor het afhandelen van de aanvragen. De module Zwoози maakt gebruik van de verschillende aanwezige componenten.

Analysis discipline

Deze componenten zijn:

- Veilingen
- Centenveilingen
- Webwinkel
- accounts



Uit het figuur is te zien dat centenveilingen het component is dat gebruik maakt van externe leveranciers. Centenveilingen heeft toegang tot de database door de verbinding met de module Zwoozi. Verder heeft het een koppeling met accounts voor de eindgebruikers. Doormiddel van accounts kunnen de eindgebruikers worden geïdentificeerd.

Het verdere ontwerp is gericht om het component centenveilingen. De overige componenten worden een ander project ontwikkeld.

3. Analyse pakketten

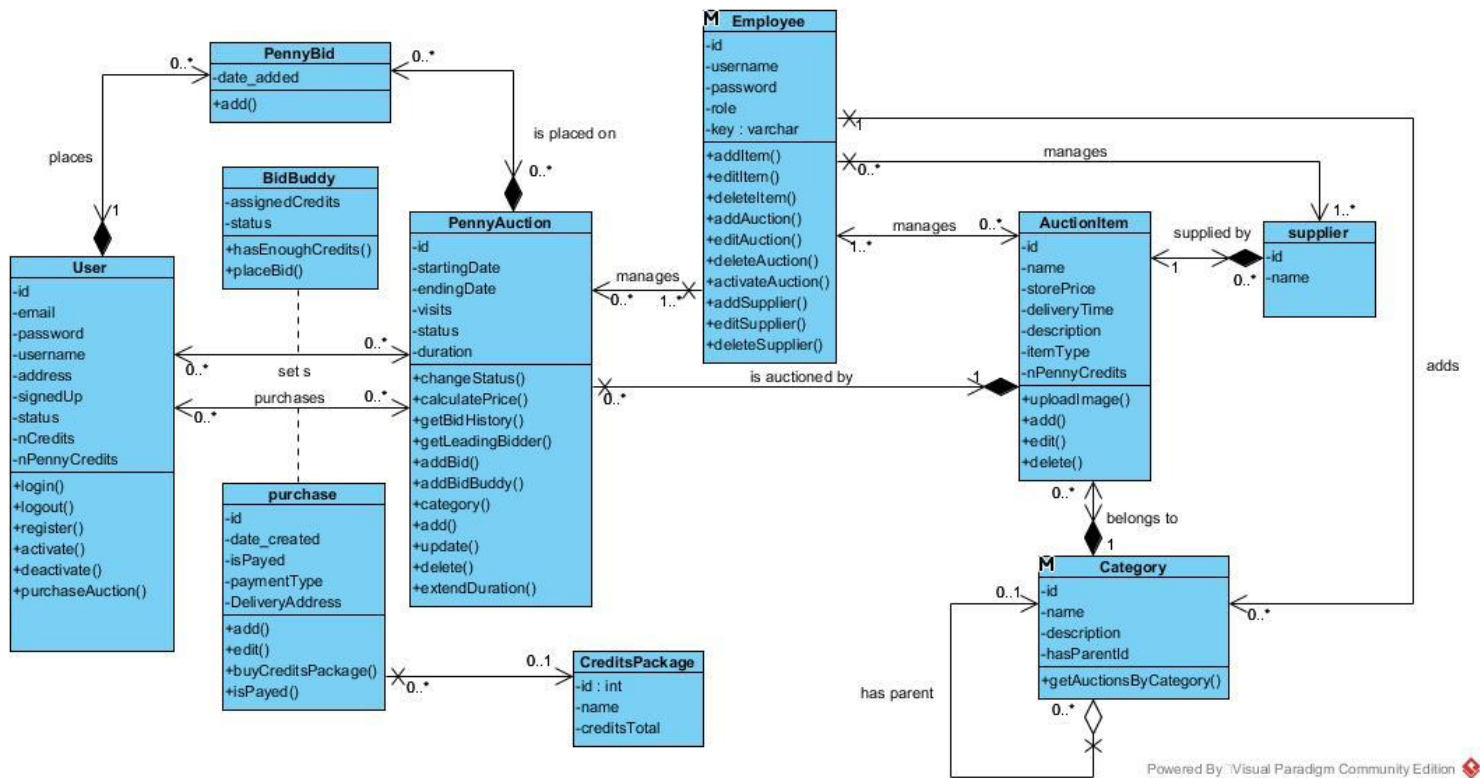
De Analyse pakketten groeperen de klassen die van belang zijn bij het bijbehorende pakket. Een pakket wordt in het design document uitgewerkt naar een component. Het pakket dat de klassen van centenveilingen bevat wordt een component. Hieronder wordt een omschrijving van het pakket weergegeven.

Pakket pennyauktion:

Het pakket bevat de klassen die verantwoordelijk zijn voor centenveilingen binnen het platform van Zwooz. Dit pakket moet op zichzelf kunnen opereren. De enige klasse die niet in dit pakket zal zitten is de user en Employee klassen.

4. Globale Analyse klassendiagram

Het klassendiagram beschrijft de structuur en de afhankelijkheden binnen het te bouwen systeem. De klassen binnen het systeem worden geïdentificeerd en hun verantwoordelijkheden worden bepaald. De klassen hebben verder hun eigen attributen en functienamen. Hieronder volgt een conceptversie van het klassendiagram.



Figuur 1, conceptversie klassendiagram

Binnen het systeem kan een gebruiker zich registreren om vervolgens te kunnen inloggen. Na het inloggen kan hij biedingen plaatsen op een centenveiling. Op de centenveiling kan de gebruiker een bidhulpje inzetten die in zijn plaats op de veiling kan bieden. Een gebruiker kan een veilingproduct kopen. De aankoop kan zich in drie situaties bevinden: wanneer de gebruiker de veiling heeft gewonnen; wanneer hij op de veiling heeft gewonnen en alsnog het product direct wil kopen; wanneer hij het product van een veiling wil kopen zonder dat hij erop heeft geboden.

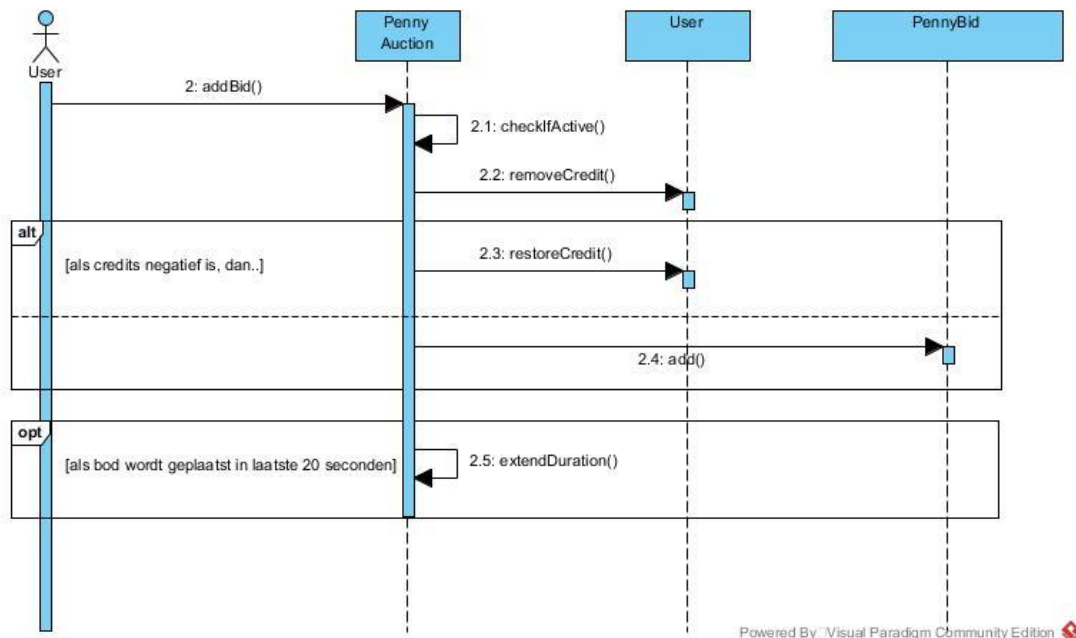
Een centenveiling heeft een startdatum, einddatum en een duur om te bepalen wanneer de veiling start en eindigt. De status van een centenveiling bepaald of de veiling nog geactiveerd moet worden, als actief is of dat hij al is afgerond. Bij de veiling willen we weten wie de laatste vijf bidders zijn doormiddel van een bidhistorie. Een centenveiling heeft een product dat wordt geveild. Het product geeft de details van het product weer. Er zijn twee soorten producten: het veilen van credits en het veiling van een product van een leverancier. Dit wordt aangegeven door het producttype. Een product behoort toe aan een categorie. De categorie kan onderverdeeld worden in subcategorieën. De medewerker kan een product toevoegen, onderhouden of verwijderen. Hij kan ook een centenveiling opzetten, onderhouden of verwijderen.

Het concept model geeft de werking van het systeem weer zonder enkele details op te noemen. In het analyse diagram wordt dit klassendiagram uitgebreid.

5. Analyse use case realisatie

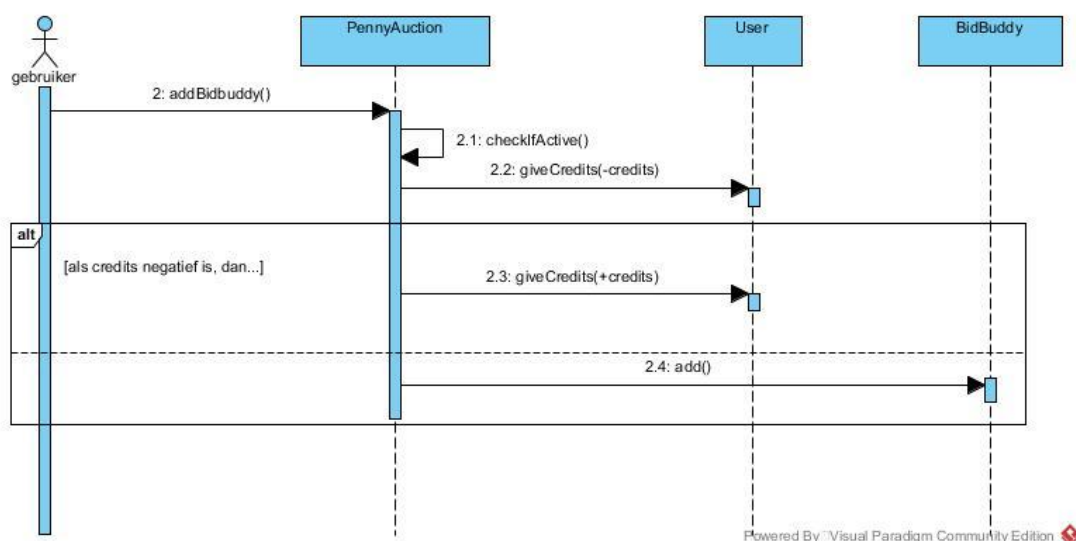
Hieronder vinden de realisaties van de use cases plaats. Deze worden aan de hand van sequentie diagrammen getoond. Sequentie diagrammen laten aan de hand van de functies die in het analyse klassendiagram zijn beschreven zien hoe de use cases in zijn werk gaan.

5.1. SD plaatsen bod



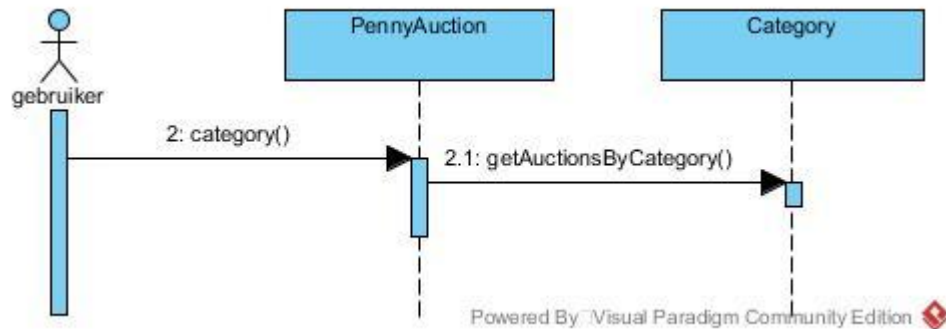
Een gebruiker (User) geeft bij een centenveiling (PennyAuction) aan dat hij een bod wil toevoegen. De centenveiling gaat na of de veiling actief is, anders kan het bod (PennyBid) niet worden geplaatst. Vervolgens kijkt de centenveiling of de gebruiker genoeg credits heeft om een bod te kunnen plaatsen. Heeft hij genoeg credits, dan wordt het bod geplaatst. Wordt het bod in de laatste 20 seconden geplaatst, dan moet de duur van de veiling worden verlengd.

5.2. SD inzetten biedhulpje



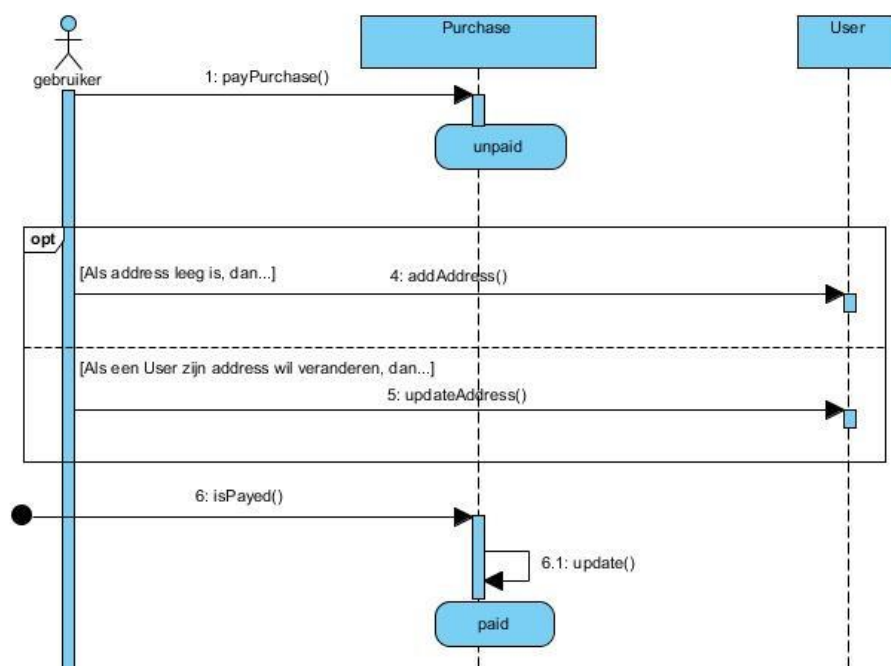
De gebruiker geeft bij een centenveiling aan dat hij een biedhulpje (BidBuddy) wil toevoegen. De centenveiling gaat na of de centenveiling actief is om een biedhulpje toe te voegen. De veiling haalt het aantal biedcredits van de gebruiker af om deze toe te wijzen aan het biedhulpje. Heeft de gebruiker nog genoeg biedcredits, dan wordt het afgehaalde biedcredits teruggestort. Heeft hij wel genoeg biedcredits, dan wordt het biedhulpje toegevoegd.

5.3. Zoeken centenveiling



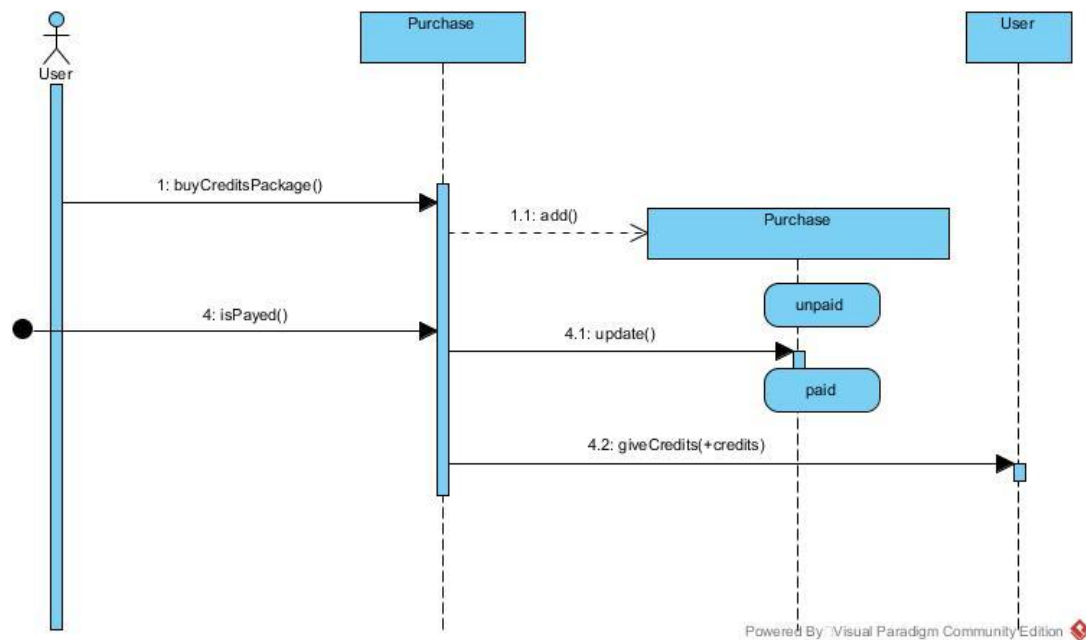
De gebruiker vraagt aan de centenveiling om de veiling op te halen die aan de categorie voldoen. De centenveiling vraagt aan categorie (Category) op alle veiling van het desbetreffende categorie op te halen.

5.4. Betalen veiling



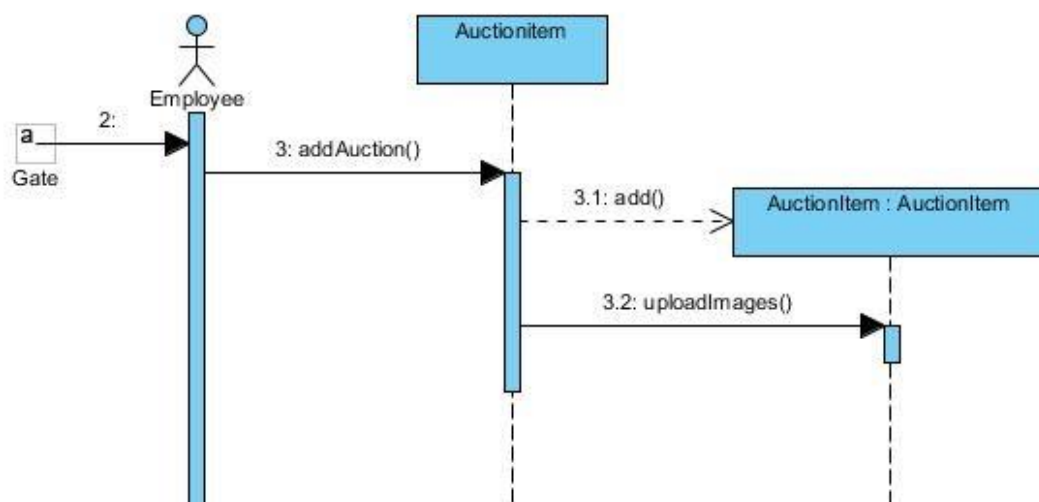
De gebruiker geeft aan dat hij een betaling van een veiling wil afhandelen. De aankoop (Purchase) controleert of een adres bekend is bij de gebruiker. Is dit niet het geval dan moet hij het adres toevoegen. De gebruiker kan hier ook zijn adres veranderen. Wanneer de aankoop van een externe betaalservice te horen krijgt dat de veiling is betaald, dan moet de betaling worden bijgewerkt.

5.5. Kopen biedpakket



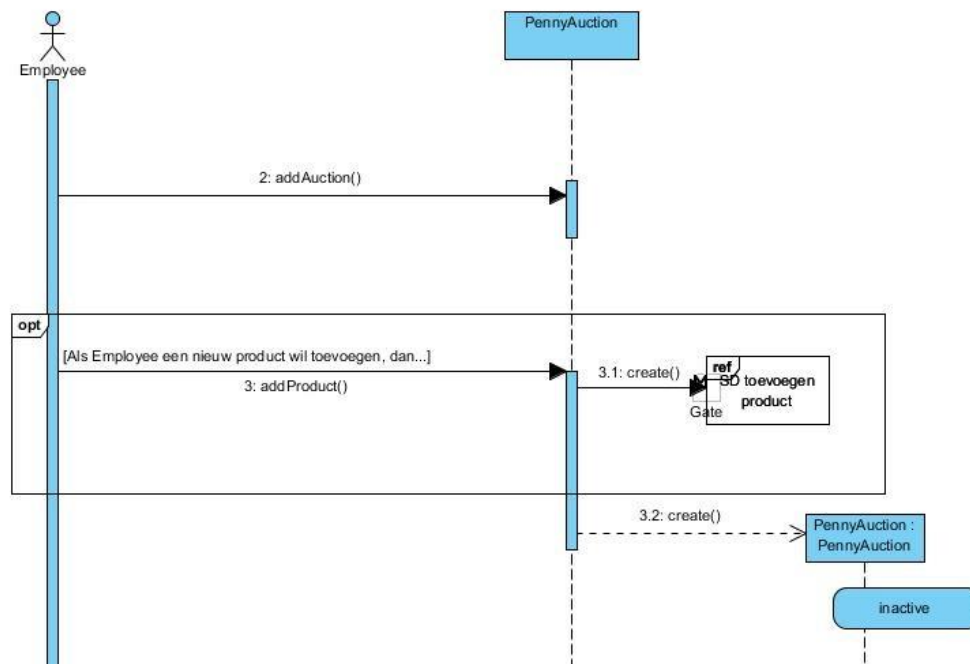
De gebruiker geeft bij de aankoop aan dat hij een biedpakket (CreditsPackage) wil aanschaffen. De aankoop maakt een nieuwe aankoop aan met de status onbetaald. Wanneer de aankoop van een externe betaalservice te horen krijgt dat de aankoop is betaald, wordt de aankoop geüpdatet. Vervolgens worden de biedcredits aan de gebruiker toegevoegd.

5.6. Toevoegen product



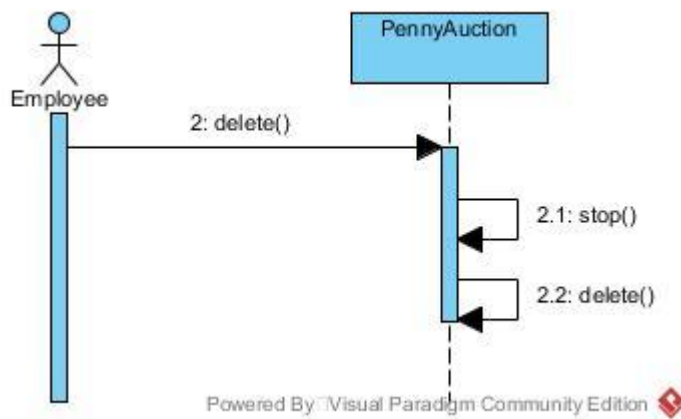
De werknemer (Employee) geeft bij het veilingproduct (AuctionItem) aan dat hij een product wil toevoegen. Het veilingproduct maakt een nieuw veilingproduct aan. Vervolgens wordt er bij het veilingproduct de afbeeldingen toegevoegd.

5.7. Plaatsen centenveiling



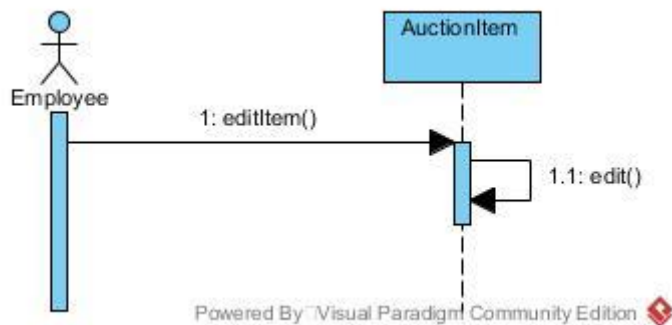
De medewerker geeft bij centenveiling aan dat hij een centenveiling wil toevoegen. De medewerker kan ervoor kiezen om een nieuw product toe te voegen (zie sequentie diagram 5.6). Daarna maakt de centenveiling een nieuwe centenveiling aan.

5.8. Verwijderen centenveiling



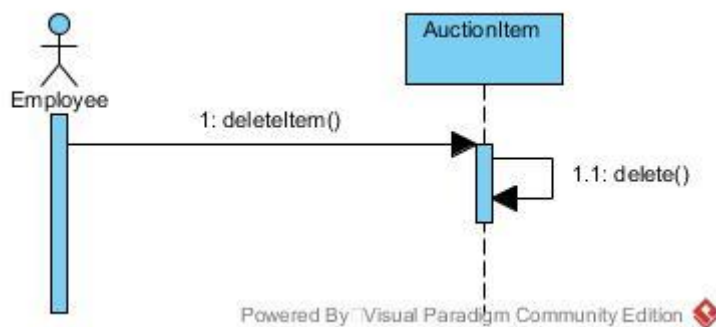
De medewerker geeft bij centenveiling aan dat hij een centenveiling wil verwijderen. De centenveiling zet zichzelf eerst op stop. Even later zal de centenveiling zichzelf verwijderen.

5.9. Aanpassen product



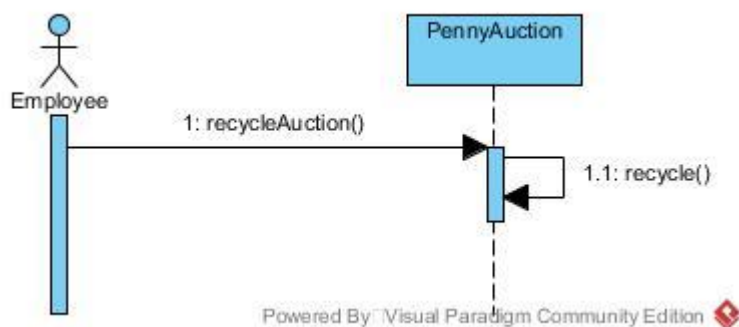
De Employee geeft bij veilingproduct aan dat hij een veilingproduct wil aanpassen. Het veilingproduct past zichzelf aan.

5.10. Verwijderen product



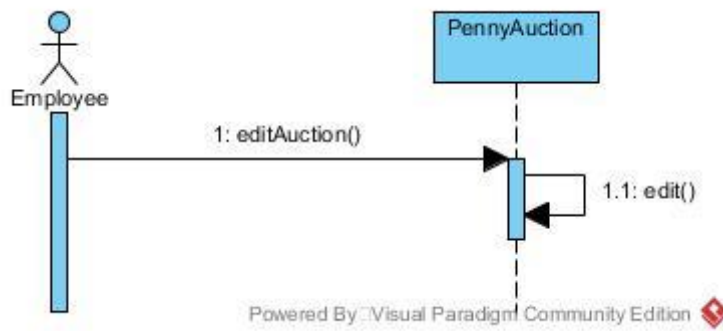
De Employee geeft bij veilingproduct aan dat hij een veilingproduct wil verwijderen. Het veilingproduct verwijdert zichzelf.

5.11. Recyclen centenveiling



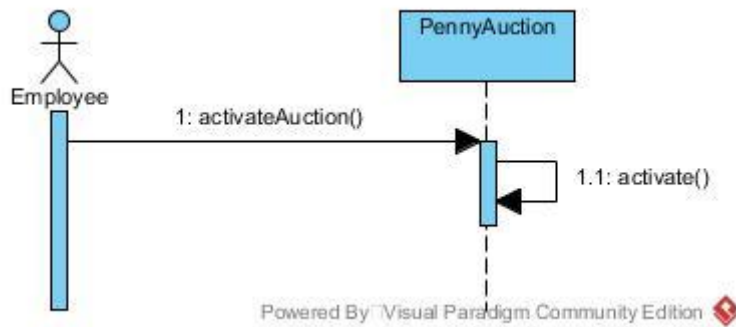
De Employee geeft bij centenveiling aan dat hij een centenveiling wil recyclen. Het veilingproduct recyclet zichzelf.

5.12. Aanpassen centenveiling



De Employee geeft bij centenveiling aan dat hij een centenveiling wil aanpassen. Het veilingproduct past zichzelf aan.

5.13. Activeren centenveiling



De Employee geeft bij centenveiling aan dat hij een centenveiling wil activeren. De centenveiling activeert zichzelf.

Design Discipline Document

Centenveilingsysteem

Project: Centenveilingsysteem voor Zwoozi

Date: 29-6-2016

Owner: *Daveron van der Ham*

Version:3.0

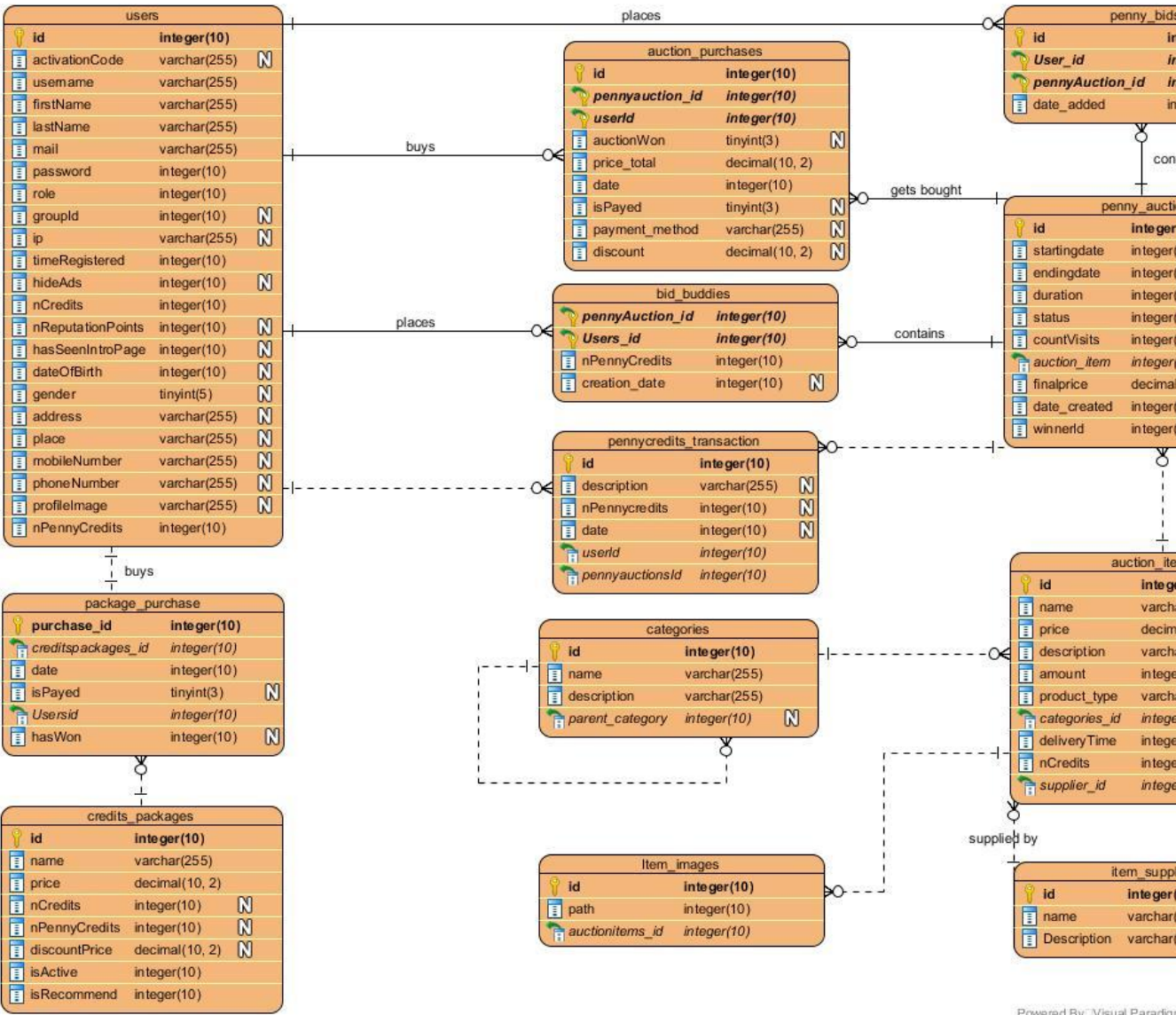
Document History

Revision History

Version	Revision date	Summary of changes
1.0	04-07-2016	Eerste versie
1.1	11-07-2016	Toevoegen van wireframes
1.2	10-08-2016	Database model aangepast.
2.0	06-09-2016	Nieuwe klassendiagram met bijbehorende sequentiediagrammen en een bijgewerkt database model.
3.0	26-09-2016	Klassendiagram en bijbehorende sequentie diagram bijgewerkt op basis van de realisatie in de constructie fase.

Table of Contents

1.	INTRODUCTIE	1
2.	ARCHITECTURALE STRUCTUUR	2
2.1.	Controllers	2
2.2.	Modellen	2
3.	ARCHITECTURALE BASELINE	3
4.	COMPONENT CENTENVEILINGEN	4
	GEDETAILLEERDE DESIGN MODEL	1
4.1.	Design klassendiagram	1
4.2.	Design sequentie diagrammen	1
4.3.	Staat diagrammen	7
4.4.	Grafische user interface design	9
4.5.	Design Patterns	21
5.	DATABASE MODEL	1



Powered By: Visual Paradigm

1

5.1. Tabellen beschrijving

2

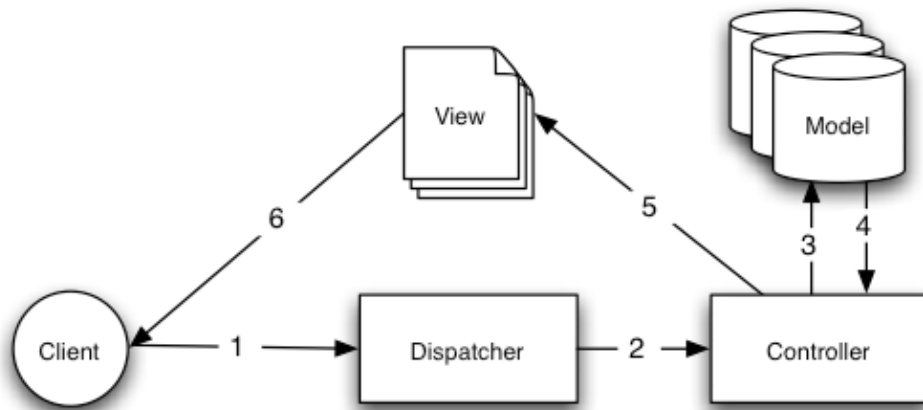
1. Introductie

Dit document is bedoeld voor de huidige en toekomstige werknemers van het platform van Zwoozi. Het document betreft een uitwerking van het analyse document en geeft meer details over het ontwerp. Het design ontwerp dient als input voor de realisatie van het systeem. In het ontwerp worden het klassendiagram, sequentie diagrammen, staat diagrammen, Grafische User Interface(GUI) en het database model weergegeven.

2. Architecturale structuur

De componenten binnen het platform Zwoози worden gebouwd in het raamwerk van Cakephp 2.8. Aangezien centenevelingen een component is van Zwoози, zal ook deze in het raamwerk worden gebouwd.

CakePHP is een PHP raamwerk dat gebruikt maakt van het MVC principe (Model-View-Controller). Het raamwerk focust zich op de objecten uit de database. Het raamwerk is voor een gedeelte object georiënteerd. Het raamwerk bestaat uit drie lagen: de model laag; de view laag en de controller laag.



Een cliënt stuurt vanaf zijn webbrowser een aanvraag naar het raamwerk. De dispatcher in het raamwerk ontleed de aanvraag en zal het naar de juiste controller sturen. De controller maakt gebruik van modellen om de aanvraag af te handelen en de response terug te sturen in een view.

2.1. Controllers

Een controller ontvangt een aanvraag van de dispatcher. De controller roept vervolgens de juiste modellen aan om de business logica af te kunnen handelen. Binnen het systeem worden er voornamelijk gebruik gemaakt van twee soorten controllers, Viewcontrollers en Restcontrollers.

Viewcontroller:

Een Viewcontroller retourneert in zijn functies de data die het systeem aan de cliënt wil tonen. Het aanroepen van een controller gebeurt aan de hand van de ingevoerde URL op de webpagina, gevolgd door de naam van een functie in de controller. Dit resulteert in `http://(webhost)/(controller)/(functie)`. Viewcontrollers retourneren dat doormiddel van views. De view is een pagina of deel van een pagina die gevuld wordt met data voordat deze aan de cliënt wordt getoond.

Restcontroller:

Een Restcontroller is een controller die via een http-aanvraag aanvragen kan ontvangen. De controller retourneert aan de hand van de gevraagde data JSON (JavaScript Object Notation) terug via de http aanvraag. Rest controllers worden gebruikt doordat de webpagina's aan de Client-side hun data continu moet updaten zonder de pagina te verlaten, iets wat een view controller op zichzelf niet kan.

2.2. Modellen

Modellen zijn klassen die de business logica bevatten van het systeem. Deze klassen communiceren met de database om de data van de modellen op te slaan, op te halen of te bewerken.

Zij handelen de data aanvraag die door de controllers worden aangeroepen. Modellen volgen voor een deel het object georiënteerd programmeren (OOP) principe. De data wordt opgehaald uit de database aan de hand van het model dat de data opvraagt. De attributen van een model zijn gelijk aan hun object in de database. Modellen kunnen onderling relaties met elkaar hebben, aan de hand

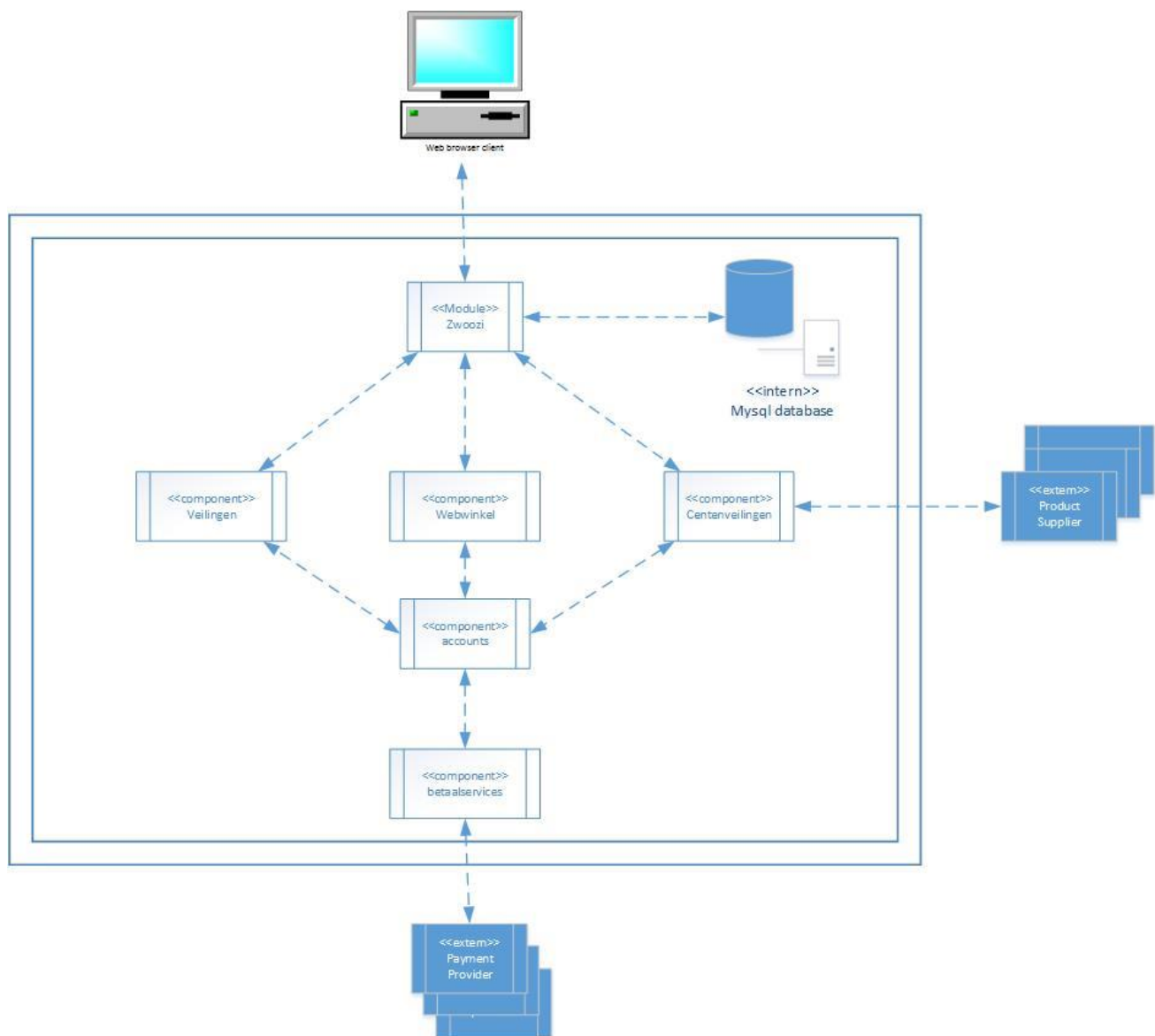
van vreemde sleutels in de database. Dit zorgt dat een model de functies van een ander model kan gebruiken.

3. Architecturale baseline

In de onderstaande afbeelding wordt de interne structuur van het platform Zwoozii weergegeven. Binnen het platform zijn er verschillende componenten en modules aanwezig. De module Zwoozii staat centraal in het platform en zal met de componenten moeten communiceren. De module initieert ook de verbinding met de database. Door gebruik te maken van de verschillende componenten kunnen de componenten gebruik maken van de verbinding van de module met de database. De aanwezige componenten zijn:

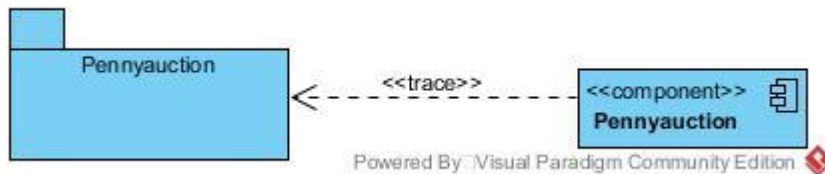
- Veilingen
- Webwinkel (Shop)
- Accounts
- centenveilingen

De componenten worden in verschillende projecten ontwikkeld. Dit project focust zich op het realiseren van het component centenveilingen.



4. Component Centenveilingen

In het analyse model is voor het component een pakket gedefinieerd waar de klassen die nodig zijn voor centenveilingen zich bevinden. Dit component wordt als losse bouwsteen van de rest van het platform ontwikkeld. Er zijn 2 klassen uit de andere componenten nodig om dit component te realiseren. Het betreft hier de User en de Employee klasse. Deze klassen zijn in hun desbetreffende componenten al opgebouwd en worden binnen het centenveiling component gebruikt. Er zal nog geen interface aanwezig zijn om te kunnen communiceren met de externe betaalservices en leveranciers.



Bij het centenveiling component hoort natuurlijk de achterliggende database erbij. De verbinding van de database wordt geregeld door de module Zwoози. Omdat er bij dit project wordt over een proof of concept, wordt er aan de hand van dit ontwerp een prototype gebouwd.

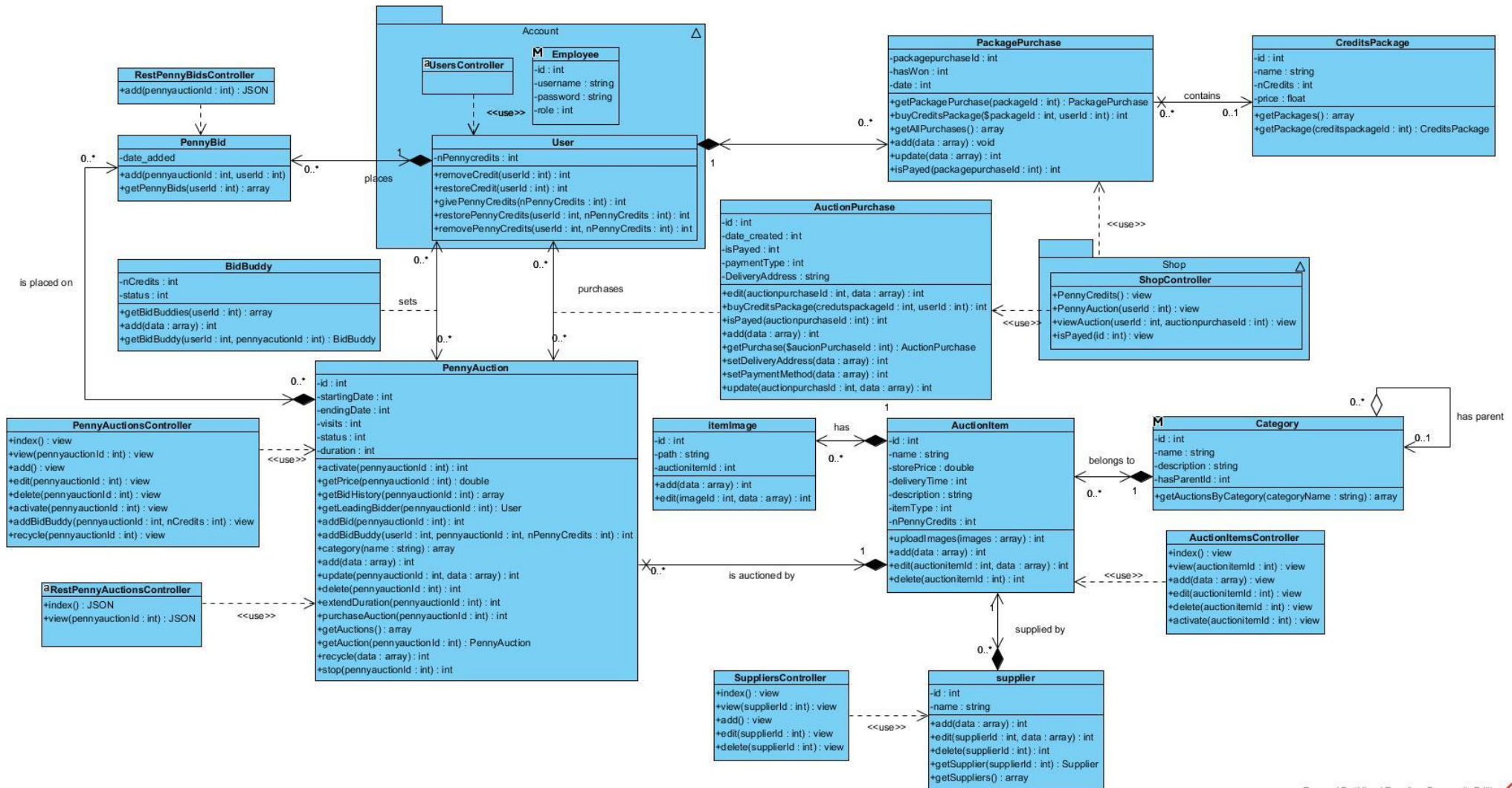
Door de huidige structuur van de ontwikkelomgeving zijn bepaalde onderdelen nog niet te realiseren in het prototype. Deze onderdelen moeten aan de backend van het systeem real-time worden beheert. De laag om deze beheer uit te voeren ontbreekt. Gedurende het ontwerp en realisatie van het prototype zal de laag nog niet worden opgezet.

De onderdelen die nog niet gerealiseerd kunnen worden zijn:

- Het real-time behouden van veranderingen die optreden bij een centenveiling
- Het real-time afronden van een veiling wanneer de tijd is verlopen.
- Het stopzetten van een centenveiling waarbij de centenveiling na enige tijd moet worden verwijderd.

Gedetailleerde design model

4.1. Design klassendiagram



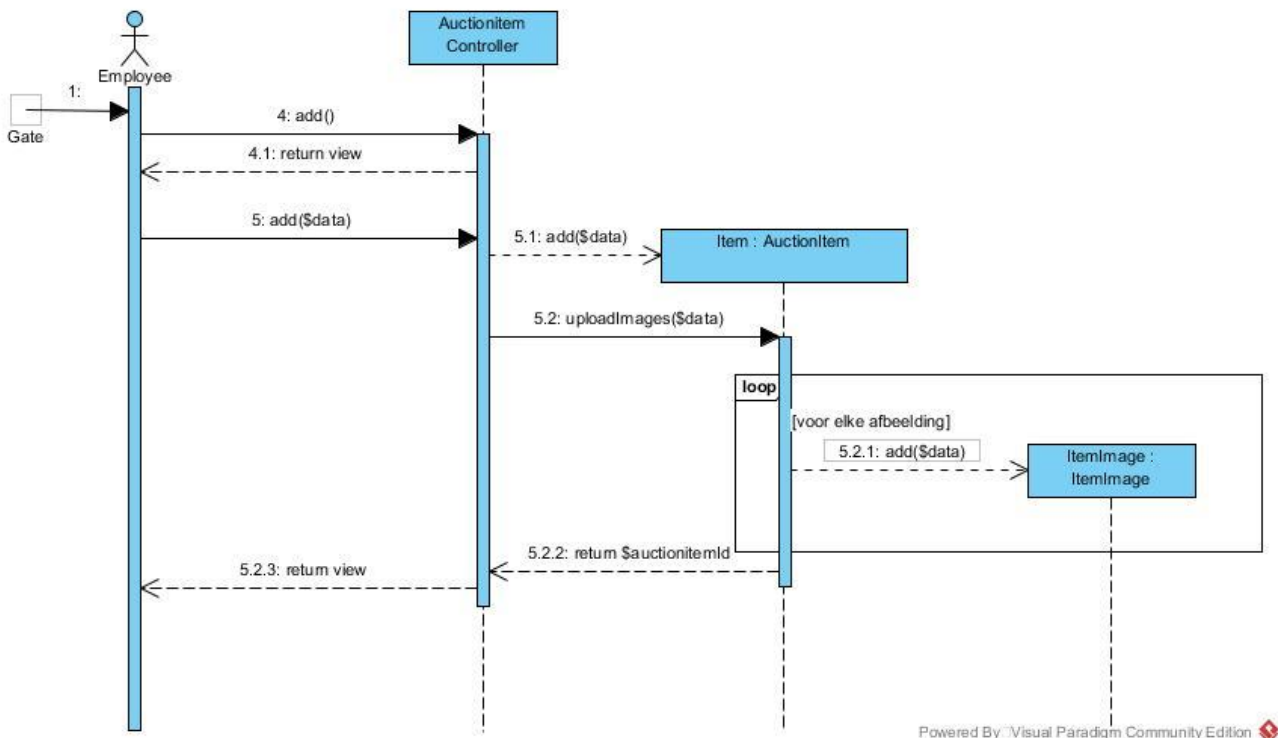
Centenveilingen is een component van het platform Zwoozi. In het bovenstaande diagram maakt dit component gebruik van twee andere componenten, namelijk accounts- en shopcomponent. In de transactie van analyse diagram naar klassendiagram verschuiven een aantal functionaliteiten en verantwoordelijkheden naar de controllers toe. Zo heeft de Employee klasse bijvoorbeeld geen functies om veiling, producten en aanbieders toe te voegen. De controllers retourneren views naar de gebruiker toe. Views zijn bestanden die de opmaak van een pagina weergeven, maar zijn geen klassen op zichzelf. Deze worden voor de leesbaarheid van het diagram niet meegenomen.

In de componenten Accounts en shop worden alleen de functionaliteiten neergezet die nieuw zijn doormiddel van het centenveiling component. De andere attributen en functies van de klassen zouden het diagram te groot maken en zijn niet relevant van voor het component.

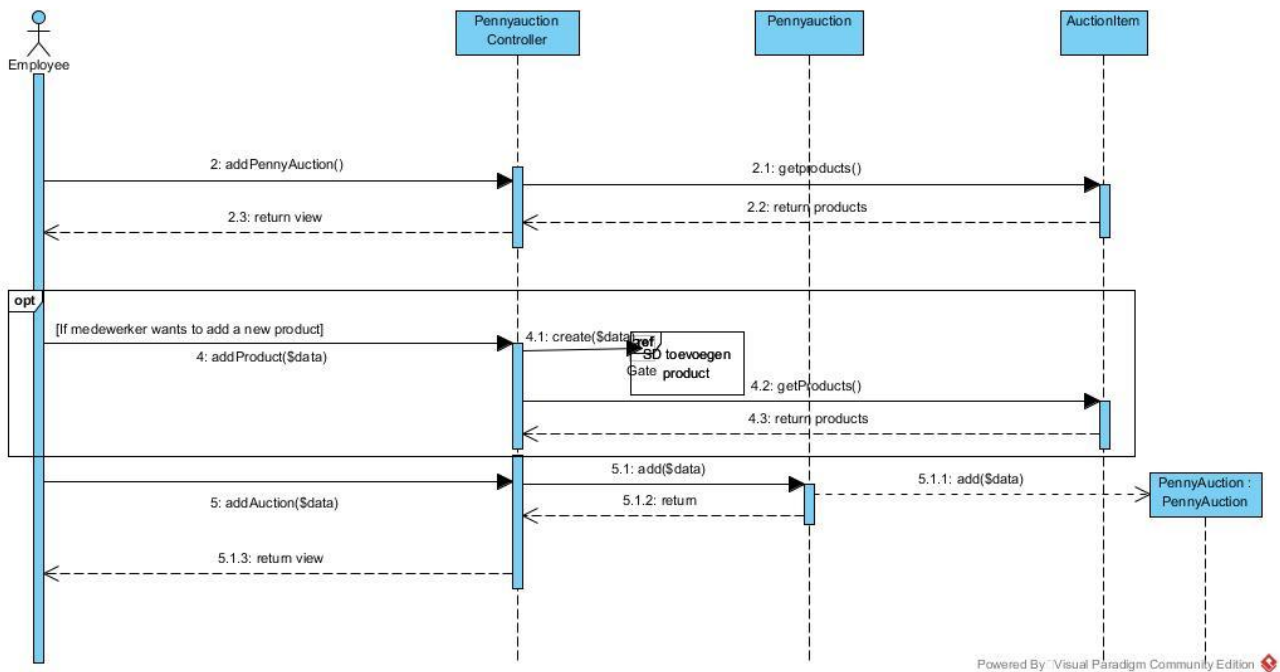
4.2. Design sequentie diagrammen

De design sequentie diagrammen komen voort uit de analyse sequentie diagrammen, die op hun beurt weer afkomstig zijn van de use cases uit het software specificatiedocument. Deze diagrammen bevatten meer details ten overstaande van de analyse diagrammen waarbij de weergegeven functies uit het design klassendiagram komen.

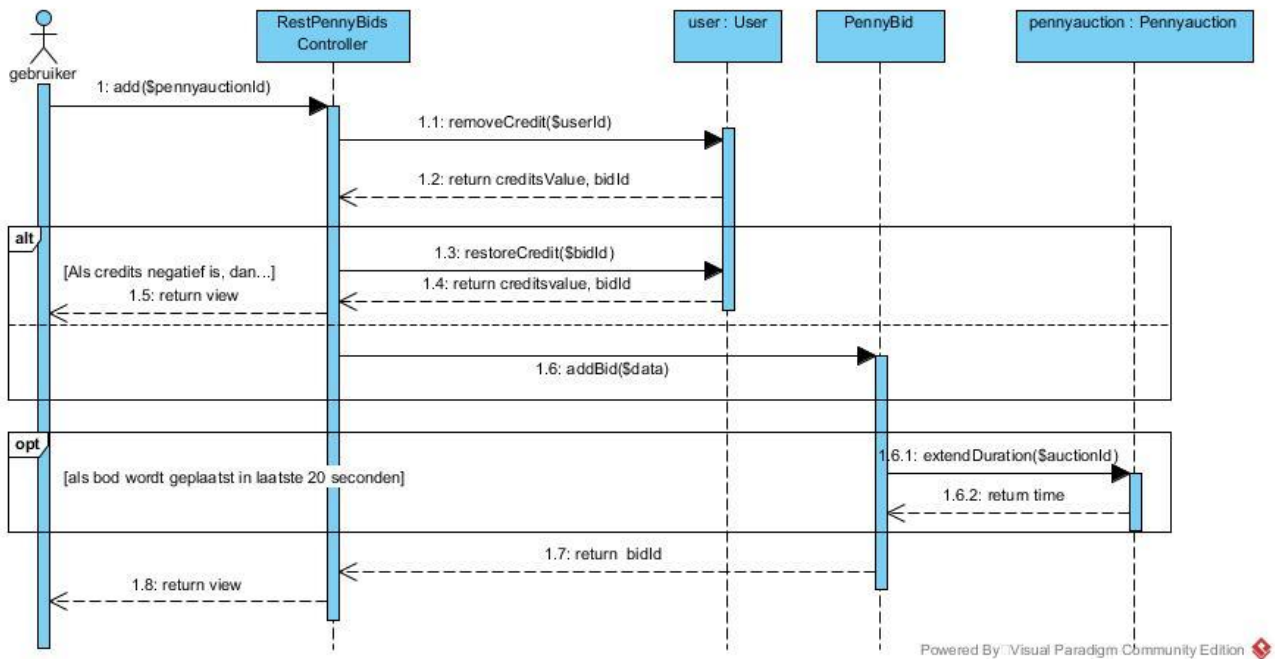
4.2.1. Diagram voor toevoegen product.



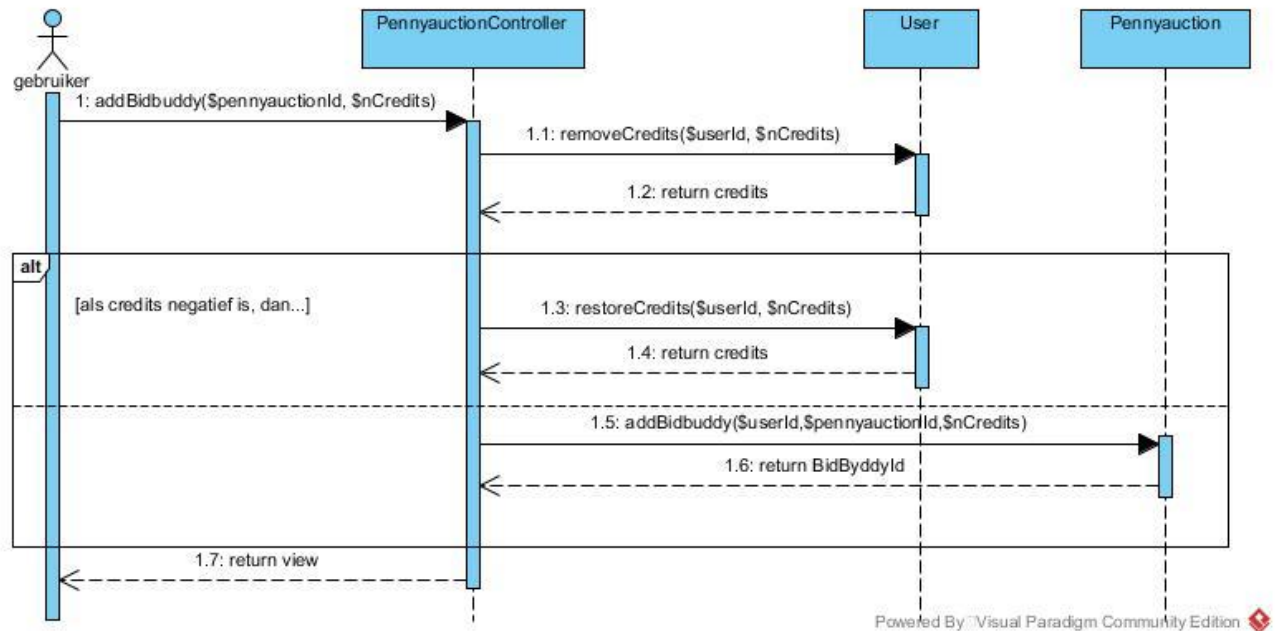
4.2.2. Diagram voor plaatsen centenveiling



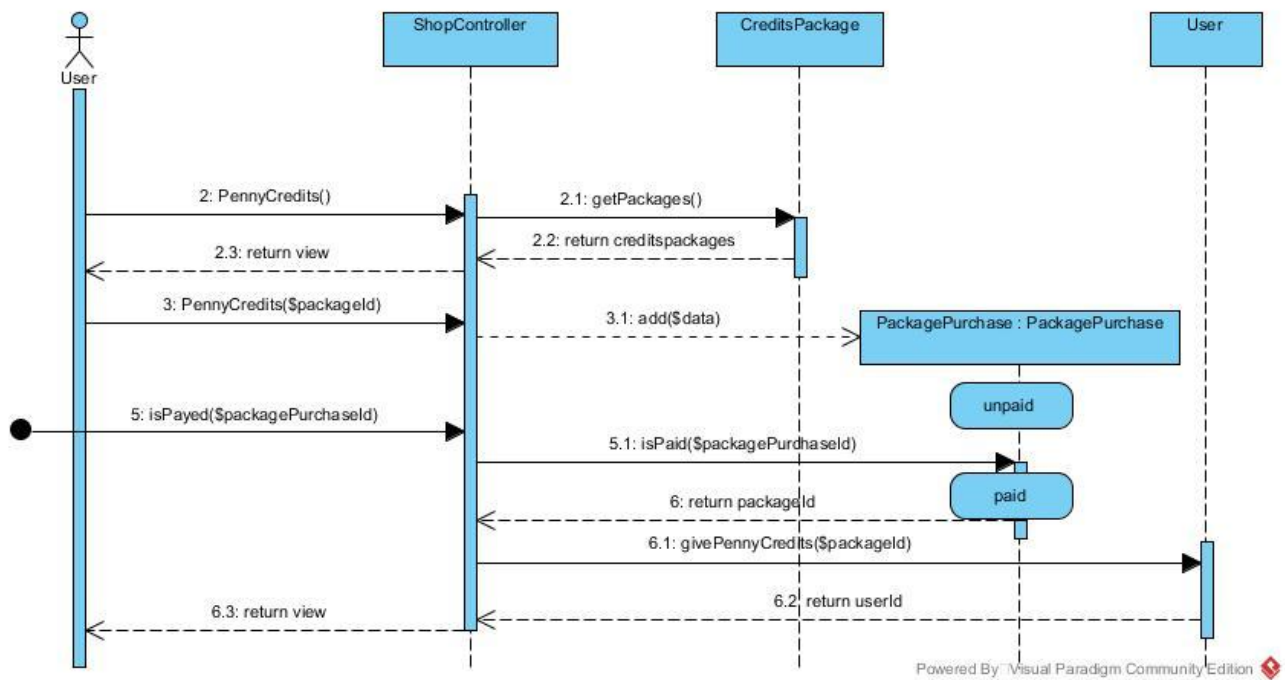
4.2.3. Diagram voor plaatsen bod



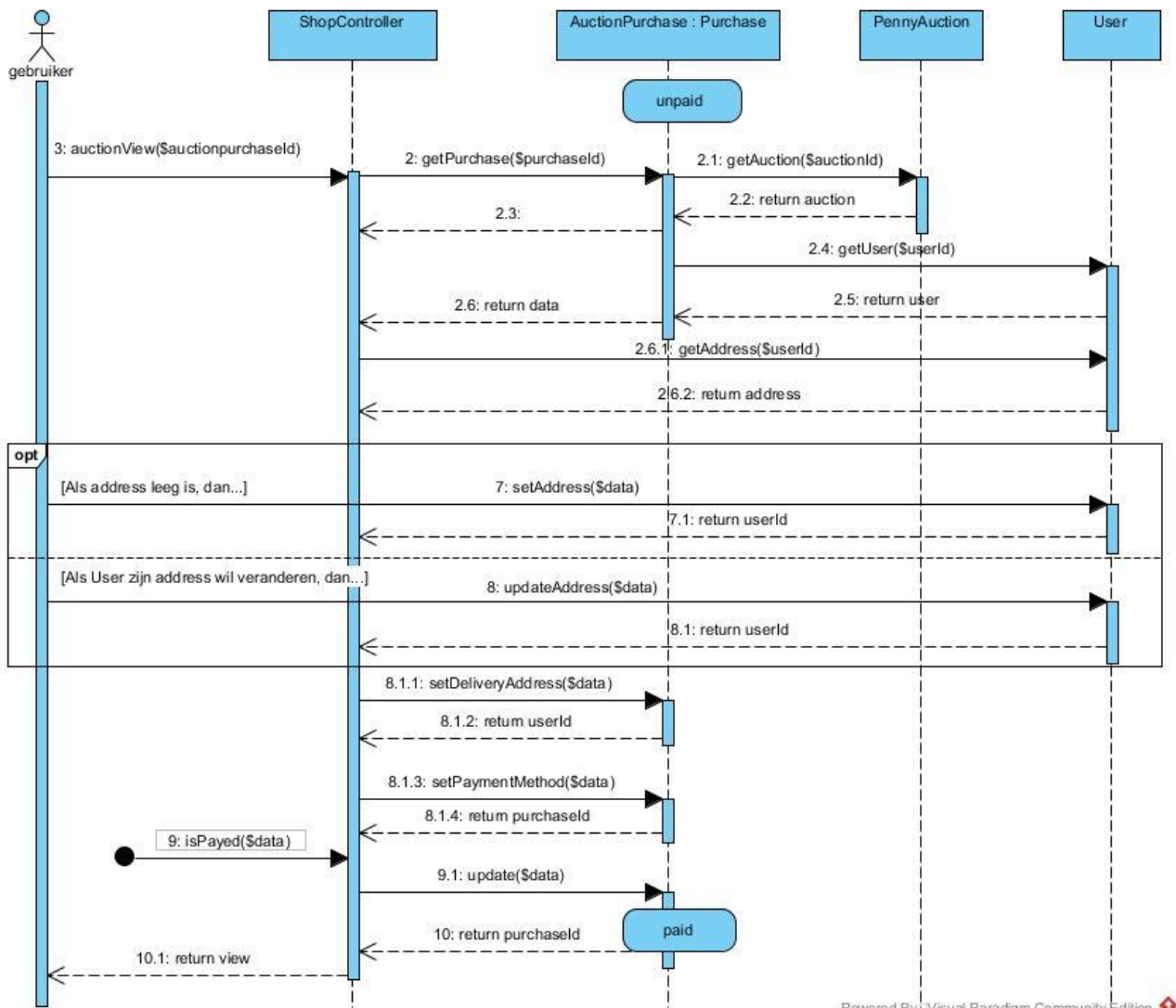
4.2.4. Diagram voor inzetten biedhulpje



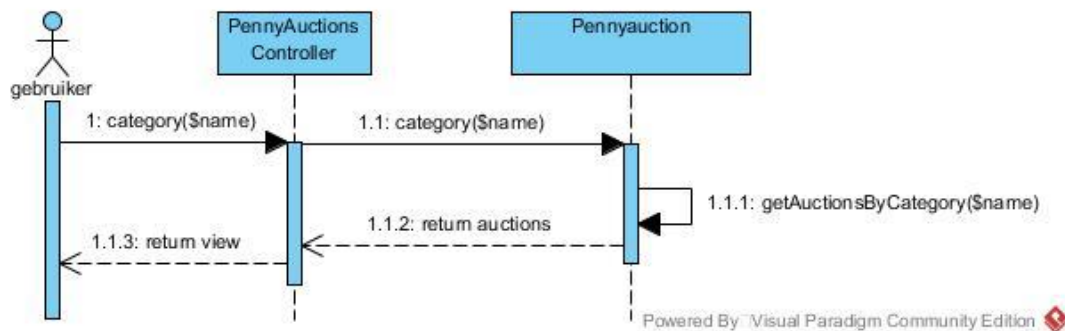
4.2.5. Diagram voor kopen biedkredieten



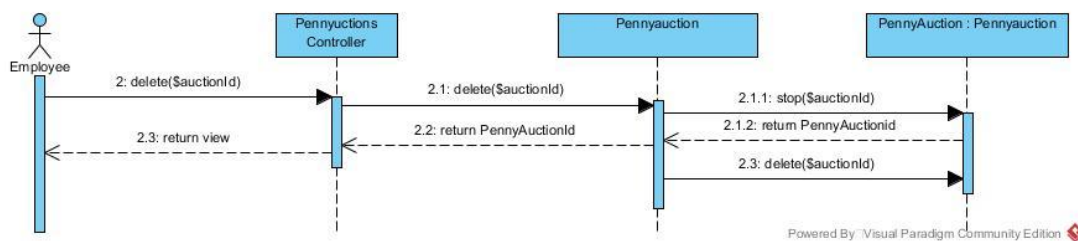
4.2.6. Diagram voor betalen veiling



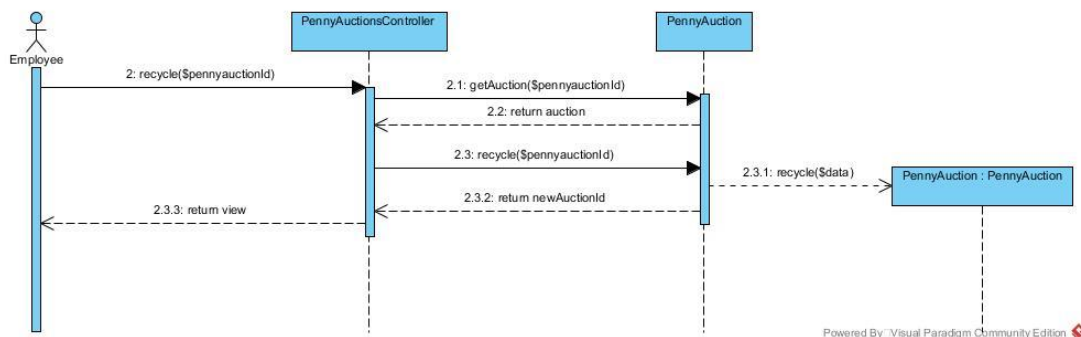
4.2.7. Diagram voor zoeken centenveilingen



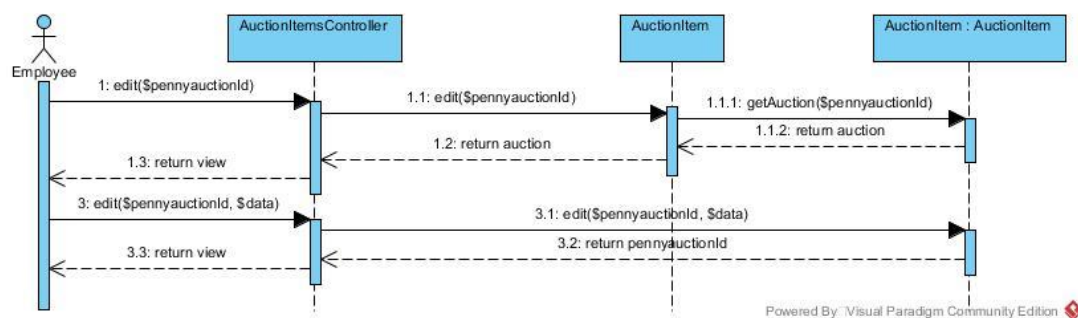
4.2.8. Diagram voor verwijderen centenveiling



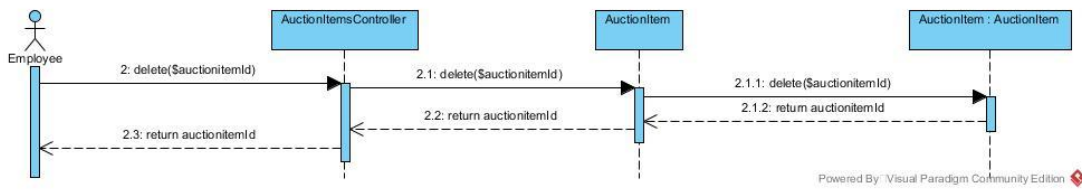
4.2.9. Diagram voor recyclen centenveiling



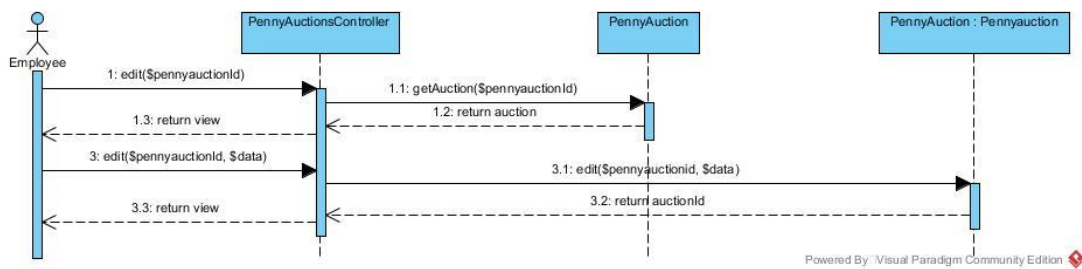
4.2.10. Aanpassen product



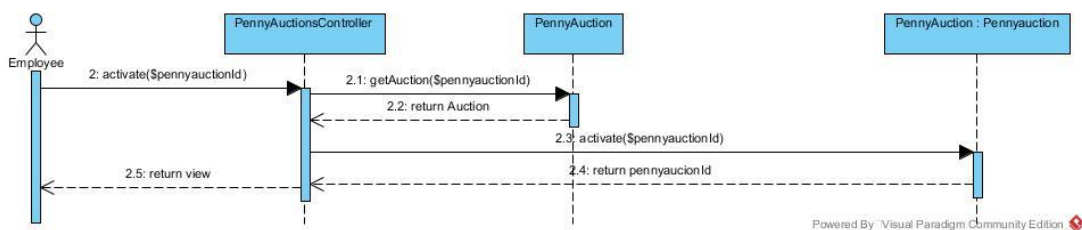
4.2.11. Product verwijderen



4.2.12. Aanpassen centenveiling



4.2.13. Activeren centenveiling



4.3. Staat diagrammen

Diagram voor de status van een centenveiling. Een veiling heeft 3 statussen: inactive; active en closed. Wanneer een veiling gesloten is, kan hij worden gerecycled worden.

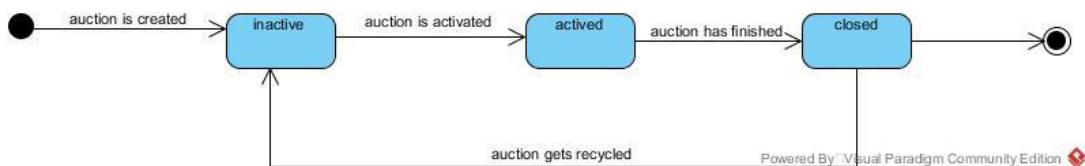
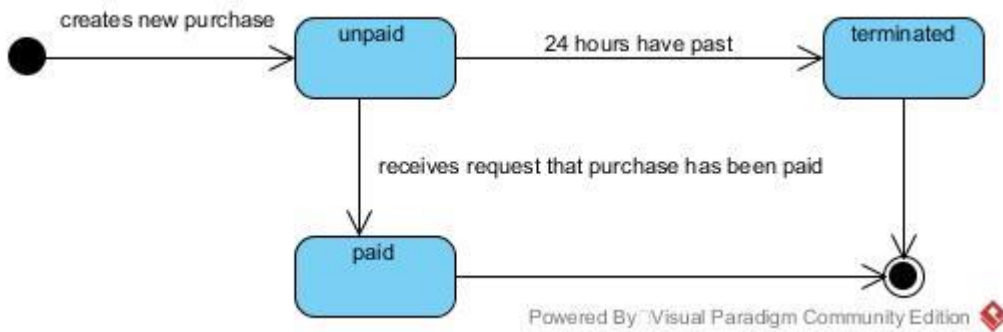
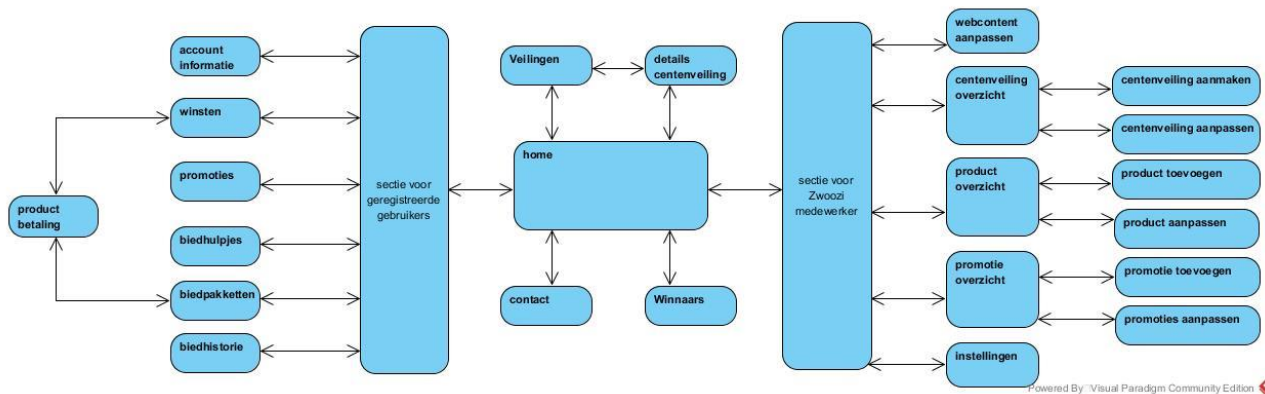


Diagram voor de status van een aankoop. Wanneer er een aankoop wordt gecreëerd, begint die met de status onbetaald. Na het afhandelen van de betaling binnen 24 uur verandert de status naar paid. Als de gebruiker de aankoop niet binnen 24 uur wordt afgehandeld, dan heeft de gebruiker geen recht meer om het product te kopen.

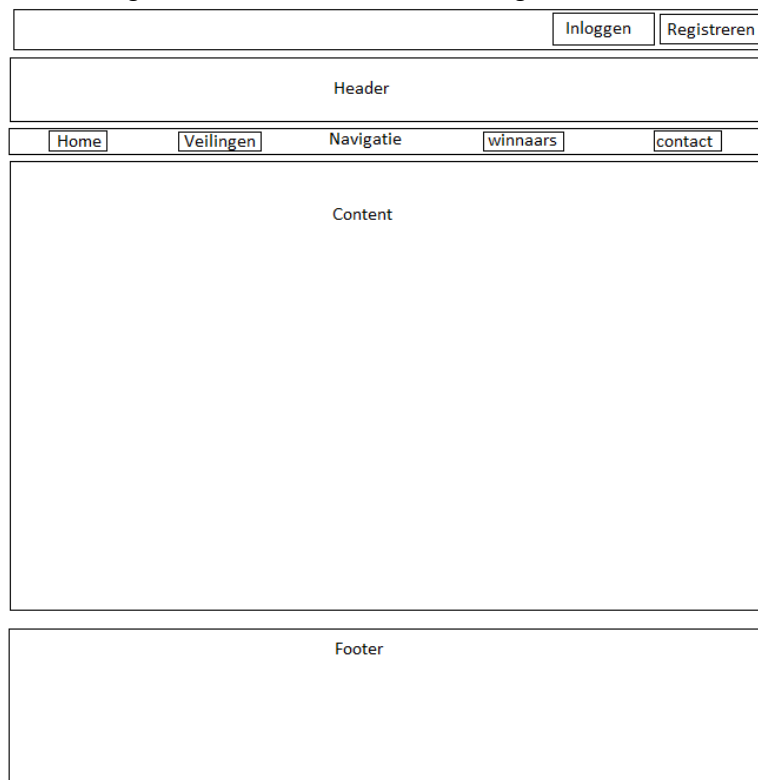


4.4. Grafische user interface design

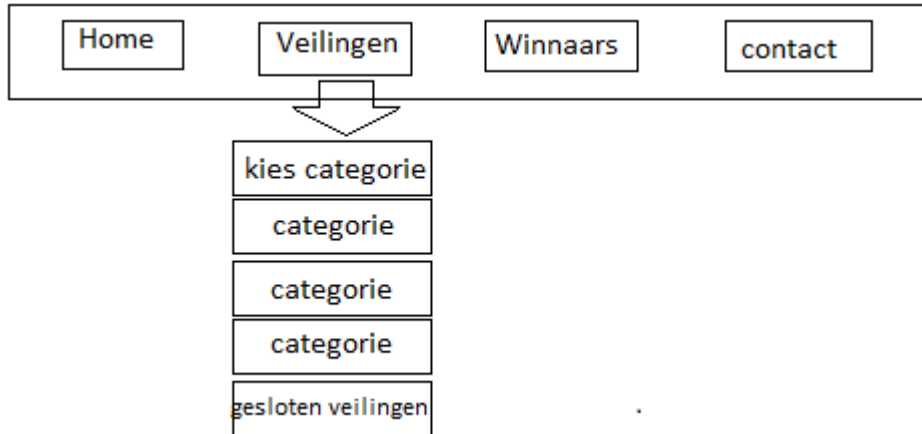
Bij het klassendiagram en de sequentie diagrammen hoort natuurlijk ook de visuele weergave van het systeem. In deze paragraaf worden de ontwerpen in wireframes weergegeven. Hieronder wordt de navigatie van het systeem weergegeven. De navigatie is opgesplitst in 3 secties. Een algemenen sectie voor de bezoekers en de gebruikers, een sectie voor de gebruikers en een sectie voor de medewerkers.



Het systeem maakt gebruik van een globale web interface die de indeling van de website weergeeft.



Bij de navigatie bevinden zich onder het navigatie-element van veilingen een extra set aan navigatie-elementen.



De webpagina's maken allen gebruik van de bovenstaande structuur voor het systeem. De gebruiker/bezoekers kunnen zich registreren en inloggen. De wireframes die volgen bevinden zich in de content sectie van de bovengenoemde wireframe.



Over de verschillende pagina's worden beknopte centenveilingen weergegeven zoals bovenstaande afbeelding. Deze weergaven hebben dezelfde opmaak ongeacht van welke soort veilingen er worden opgevraagd.

4.4.1. Home pagina

De home pagina geeft een aantal beknopte veilingen weer die het snelst zullen aflopen, oftewel met de minste aflooptijd.

Actuele veilingen

Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling
Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling
Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling

4.4.2. Actuele centenveiling weergave

Productfoto	Koop nu!	Direct kopen bedrag	Veiling info
	Huidige biedprijs	Biedprijs	
	Actuele aflooptijd	Aflooptijd	
	Leidende bidder	Biednaam	
Biedhulpje +		BIEDEN	
Productomschrijving			
foto carousel		biedhistorie	
[] [] [] []		Bieder1	
		Bieder2	
		Bieder1	
		Bieder2	
		Bieder3	
Vorige veilingen van het product			
gesloten veiling gesloten veiling gesloten veiling gesloten veiling			

4.4.3. Veilingen pagina

Meldingen		
Actuele veilingen		
Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling
Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling
Beknopte weergave centenveiling	Beknopte weergave centenveiling	Beknopte weergave centenveiling

4.4.4. Winnaars pagina

Winnaars

Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving
Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving
Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving	Productnaam eindprijs winkelprijs besparing gebruikersnaam Korte omschrijving

4.4.5. Account pagina

Je persoonlijke informatie

Voornaam	Achternaam
Geboortedatum	Emailadres
	Geslacht
Schermnaam	
	Opslaan

Aflever adres

Land	
Postcode	Huisnummer
Adres	
Stad	Telefoonnummer
Opslaan	

Wachtwoord wijzigen

Oud wachtwoord	Nieuw wachtwoord	Herhaal	Opslaan
Oud emailadres	Nieuw emailadres	Herhaal	Opslaan

4.4.6. Gewonnen veilingen pagina

gewonnen veilingen

Veiling	Datum	Eindprijs	Status	Optie	Betalingstijd
productafbeelding productnaam	DD/MM/JJJJ	\$---,--	wel of niet betaald	Betaal	Timer
productafbeelding productnaam	DD/MM/JJJJ	\$---,--	wel of niet betaald	Betaal	Timer
productafbeelding productnaam	DD/MM/JJJJ	\$---,--	wel of niet betaald	Betaal	Timer

4.4.7. betalingspagina

Samenvatting betaal informatie

Productnaam

Prijs product

verzendkosten

Totaalbedrag

Aflever informatie

Aantal werkdagen

Adres

straat

postcode

land

telefoon nummer

Betaalmethode 1

Betaalmethode 2

Betaalmethode 3

Betaalmethode 4

4.4.8. biedhistorie pagina

Bied historie

Datum	Beschrijving	Veiling	Bij	Af
DD/MM/JJJJ	Bod geplaatst of kredieten toegevoegd	Veiling naam	Aantal	Aantal
DD/MM/JJJJ	Bod geplaatst of kredieten toegevoegd	Veiling naam	Aantal	Aantal
DD/MM/JJJJ	Bod geplaatst of kredieten toegevoegd	Veiling naam	Aantal	Aantal

4.4.9. Biedpakketten pagina

Aanschaffen Bied-pakketten

pakket	Biedkredieten	Prijs
Pakket 1	Aantal biedkredieten	Prijs aantal
Pakket 2	Aantal biedkredieten	Prijs aantal
Pakket 3	Aantal biedkredieten	Prijs aantal
Pakket 4	Aantal biedkredieten	Prijs aantal

4.4.10. Control paneel centenveiling aanmaken pagina

control panel

centenveilingen

webcontent

promoties

producten

instellingen

centenveiling aanmaken

startdatum

starttijd

duur

product

selecteer product...

product 1

product 2

product 3

product 4

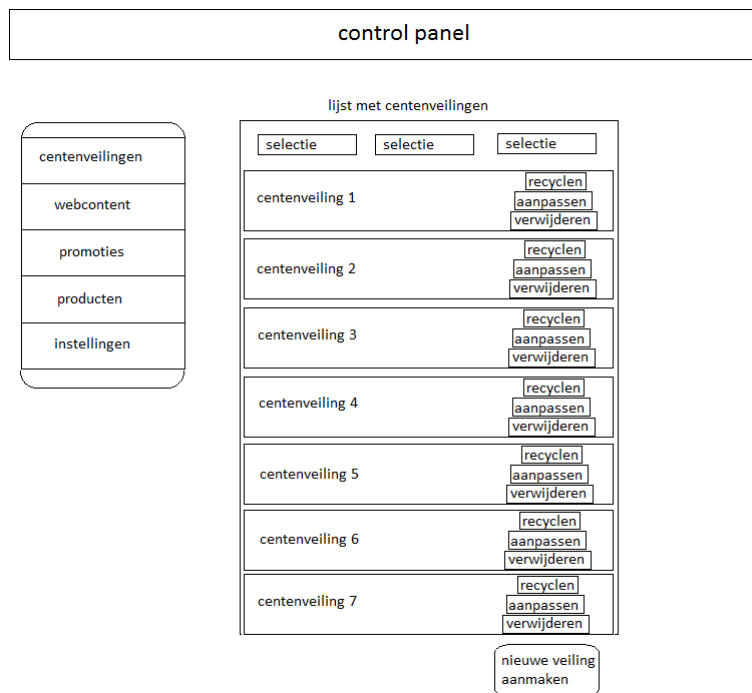
voeg nieuw product toe...

Afbeelding
(weergeven na
selecteren product)

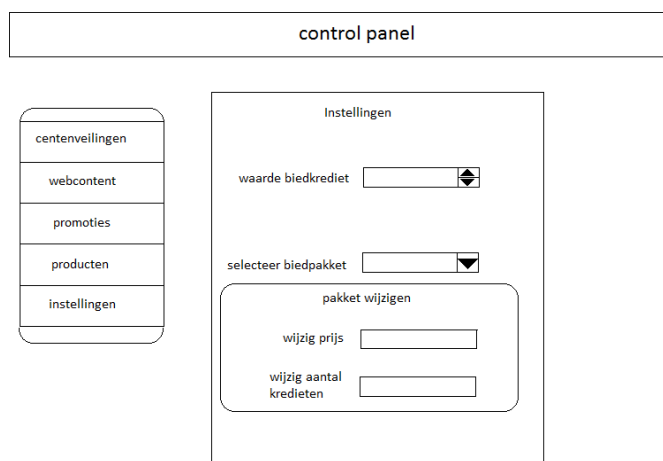
Afbeelding
(weergeven na
selecteren product)

centenveiling aanmaken

4.4.11. Control panel centenveilingenlijst pagina



4.4.12. Control panel instellingen



4.4.13.

control paneel productenlijst pagina

Control paneel productenlijst

control panel

centenveilingen

webcontent

promoties

producten

instellingen

product toevoegen

productnaam

levertijd

winkelprijs

product omschrijving

afbeelding toevoegen

product toevoegen

4.4.14. Control paneel productenlijst

control panel

centenveilingen

webcontent

promoties

producten

instellingen

lijst van producten

selectie

selectie

selectie

product 1

aantal

aanpassen

verwijderen

product 2

aantal

aanpassen

verwijderen

product 3

aantal

aanpassen

verwijderen

product 4

aantal

aanpassen

verwijderen

product 5

aantal

aanpassen

verwijderen

product 6

aantal

aanpassen

verwijderen

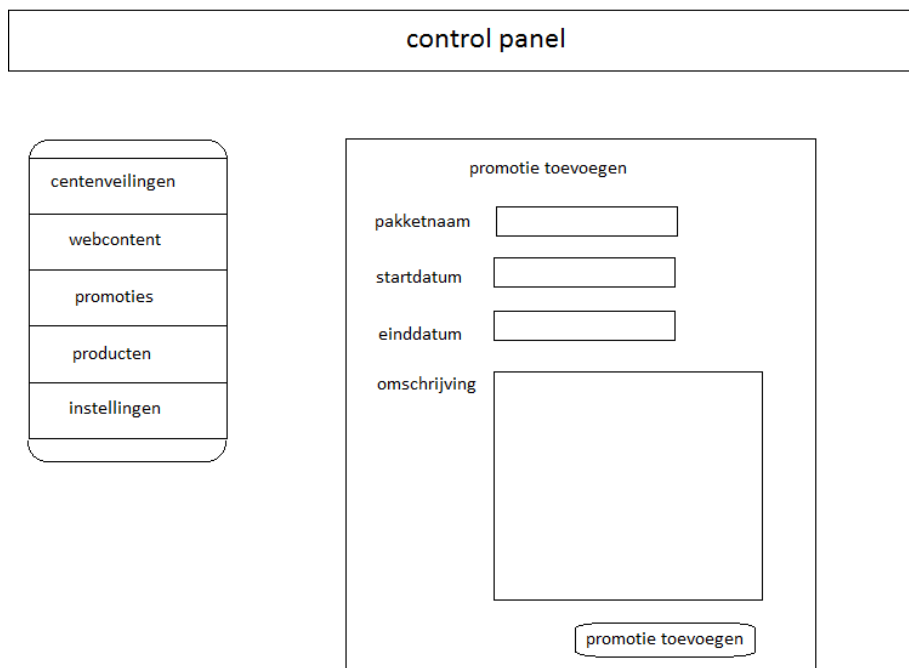
product 7

aantal

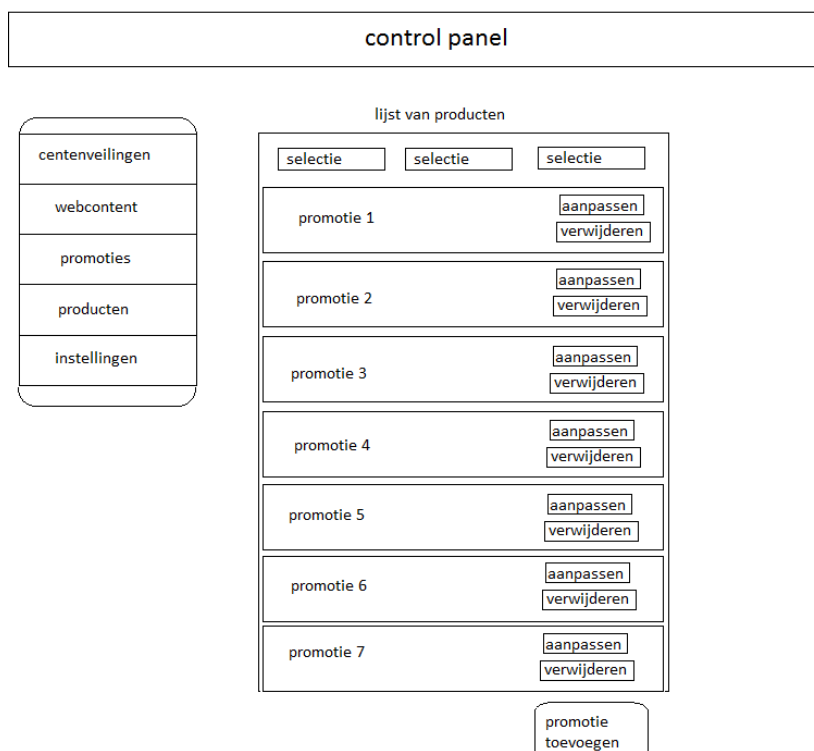
aanpassen

verwijderen

nieuw product toevoegen



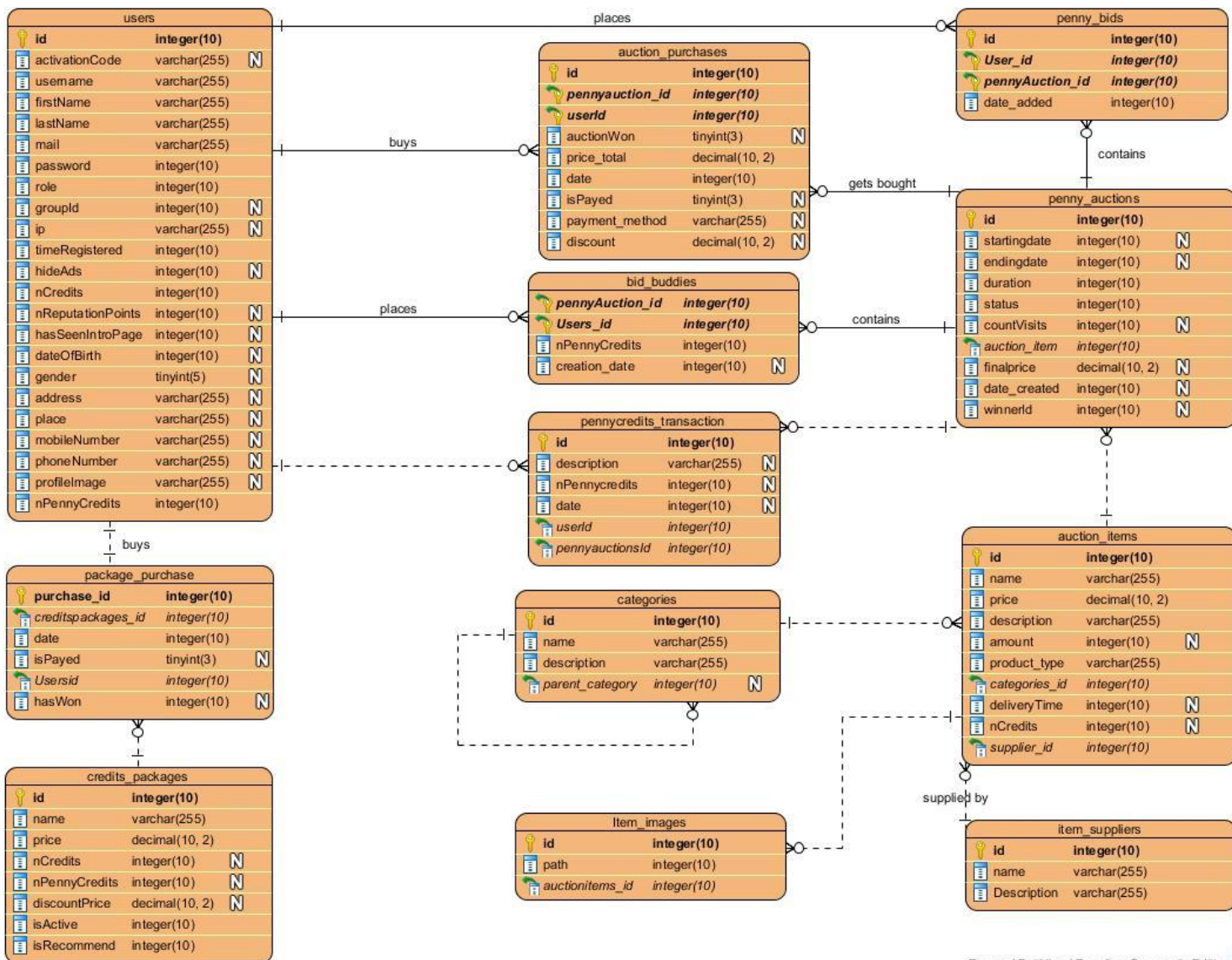
4.4.15. Promotie beheerlijst pagina



4.5. Design Patterns

In het bovenstaande diagram zijn er geen design patterns toegepast. Het Cakephp framework maakt al gebruik van een groot aantal verschillende design patterns en het toevoegen van een extra design pattern zal de structuur van het systeem te complex maken. Het toevoegen van een extra design pattern kan de structuur te complex maken en verliest men het overzicht.

5. Database model



De modellen uit het databasemodel zijn gebaseerd op objecten uit het domein van het systeem. Doormiddel van normalisering zijn de objecten opgesplitst in verschillende tabellen om de herbruikbaarheid per object te vergroten.

Het employee model uit het design klassendiagram heeft in het datamodel geen entiteit. Dit model valt buiten de tabellen van centenveilingen doordat employees geen directe verbinding heeft met de centenveilings tabellen.

In de tabellen wordt er voor de datum in plaats van de date notatie een integer gebruikt. De integers zijn bedoeld om een UNIX timestamp op te slaan. Een UNIX timestamp is een seconde teller die gestart is in 1970. De timestamp is voor iedere gebruiker hetzelfde en kan worden geconverteerd naar de juiste notatie. Omdat een timestamp een integer is, valt er makkelijker om mee te rekenen. Het rekenen bij het afstemmen van de data heeft een groot effect op de werking van centenveilingen. Bij auction_item kan bijvoorbeeld doormiddel van de duur(duration) de start en einddatum van een centenveilings worden bepaald wanneer een centenveilings wordt geactiveerd. [Einddatum = startdatum (huidige timestamp) + duration]

5.1. Tabellen beschrijving

Tabel naam	Domein object	Omschrijving
Users	(eind)gebruikers	Dient om de identificatie van een gebruiker vast te leggen zodat hij geïdentificeerd kan worden bij het gebruiken van de applicatie. Verder wordt er specifieke data opgeslagen zoals credits, adres en profielgegevens. De id van deze tabel dient als identificatie in gerelateerde tabellen om te zien welke records er bij de gebruiker toebehoren. Een User kan bieden op een centenveiling en hij kan credits en veilingproducten kopen.
Penny_auctions	centenveilingen	Bevat de data waarmee een centenveiling kan wordt gestart en beëindigd. Penny_auctions heeft een veel-op-veel relatie met Users voor zowel het plaatsen van een bod/biedhulpje en het kopen van een veilingproduct. Een penny_auction behoort altijd maar tot één auction_item toe, zodat een auction_item meerdere keren kan worden geveild in verschillende centenveilingen.
Penny_bids	Biedingen met biedcredits	Het resultaat van de veel-op-veel relatie tussen Users en penny_auctions. Naast de id's van de gerelateerde tabellen bevat deze tabel ook nog een eigen id zodat een User meerdere biedingen kan doen op een penny_auction.
Bid_buddies	biedhulpje	Net als penny_bids is dit het resultaat van een veel-op-veel relatie tussen Users en penny_auctions. In tegenstelling tot penny_bids, heeft bid_buddies geen eigen id. Een User kan maar één biedhulpje bij een veiling gebruiken.
Auction_items	veilingproducten	Het product dat doormiddel van een centenveiling wordt geveild. Een auction_item heeft meerdere centenveiling omdat één product meerdere keren kan worden geveild. De tabel auction_items is het resultaat van twee objecten, namelijk creditsproducten en leverancierproducten die kunnen worden geveild. Doordat het raamwerk Cakephp overerving ten strengste afraadt, hebben we de twee objecten moeten denormaliseren naar één tabel.
Item_suppliers	productleveranciers	De tabel die aangeeft waar een veilingproduct vandaan komt. Deze tabel moet later worden uitgebreid als de leveranciers worden toegevoegd aan de scope van het systeem.
categories	Categorieën	De tabel die aangeeft in welke categorie een product zich bevindt. Categories heeft een unaire relatie met zichzelf doordat een categorie onder te verdelen is in subcategorieën. Voorlopig geven we alleen de naam van de leverancier item_suppliers om als dummy data te dienen.
Item_images	productafbeeldingen	De tabel slaat de locatie op van de afbeeldingen die worden geüpload en legt vast bij wel auction_items ze toebehoren. Een auction_items kan meerdere afbeeldingen van hetzelfde product bevatten.
Auction_purchases	Centenveiling aankoop	De tabel is het resultaat tussen een veel-op-veel relatie tussen Users en penny_auctions. De tabel heeft naast de id's van de gerelateerde tabellen ook een eigen id. Een user kan meerdere keren het veilingproduct van een centenveiling kopen. Auction_purchases houdt bij welke centenveilingen de gebruiker wil kopen. Verder houdt het bij wanneer de aankoop verloopt, of de aankoop al betaald is, of de aankoop bij de winnende bidder hoort.
Transaction_pennycredits	Biedcredits transactie	Deze tabel houdt bij hoeveel biedcredits er bij de gebruiker af wordt gehaald of wordt toegevoegd. De omschrijving geeft aan of de af/toename doormiddel van het kopen van een biedpakket of het plaatsen van een bod plaats heeft gevonden. Deze tabel dient tegelijkertijd als een logging.
Credits_packages	Credits pakketten	Dit tabel bevat de credits pakketten die gebruikers kunnen aanschaffen. Credits_packages bevatten zowel credits als biedcredits.



TESTRAPPORT

Centenveilingsysteem Zwoozi



21 SEPTEMBER 2016

DIDACTICUM
Folkert Elsingastraat 3-7

Inhoud

1.	Unittesten	2
2.	User model.....	2
3.	PennyCreditsTransaction model	4
4.	PennyAuction model.....	8
5.	PennyAuctionPurchase model	11
6.	Performance acceptatietest.....	14
7.	Log in op Zwoozi:	14
8.	Bieden op een veiling zonder biedcredits:.....	15
9.	Kopen van biedcredits	16
10.	Zoeken naar een veiling op basis van categorieën:	17
11.	Bieden op een veiling met biedcredits:	18
12.	Bieden op de details pagina van een centenveiling:.....	19
13.	Direct kopen van een veilingproduct:.....	20
14.	Betalen van een productaankoop:.....	21
15.	Toevoegen van een veilingproduct:.....	23
16.	Wijzigen van een veilingproduct:.....	25
17.	Verwijderen van een veilingproduct:.....	26
18.	Aanmaken van een centenveiling:.....	27
19.	verwijderen van een centenveiling:.....	29
20.	activeren van een centenveiling:	30

1. Unittesten

Dit hoofdstuk beschrijft de unittesten die zullen worden uitgevoerd tijdens het testen van het systeem. De unittesten zullen een pseudo code worden beschreven en hebben alleen betrekking op de functionaliteiten die aanwezig zijn in de modellen. Vele functionaliteiten bevinden zich ook in de controllers en zullen doormiddel van de performance acceptatie test worden getest. Hieronder volgen de cases van de verschillende unittesten. De unittesten worden ingedeeld op basis van de modellen waar de functionaliteiten zich in bevinden. De testen worden binnen cakePHP uitgevoerd doormiddel van PHPUnit, een PHP tool die het unittesten mogelijk maakt.

2. User model

functienaam	givePennyCredits
parameters	• UserId • credits
	1. De functie checkt of de opgegeven gebruiker bestaat in het systeem en dat het aantal credits dat moet worden verwijderd niet 0 is, anders retourneert de functie "false". 2. De functie legt de data vast van het nieuwe credits totaal. 3. De functie kijkt of het nieuwe credittotaal niet negatief is, anders mag het nieuwe creditaantal niet worden opgeslagen en retourneert de functie "false". 4. De functie slaat het nieuwe creditaantal bij de gebruiker op. Als het opslaan niet lukt zal de functie "false" retourneren. Bij succes zal de functie "true" retourneren.
Unittest ID	U1
Testomschrijving	We testen het verwijderen van een penny credit bij een gebruiker. Deze functie wordt gebruikt bij het bieden op een veiling.
Testdata	• We geven het ID mee van een gebruiker waar zijn credittotaal hoger is dan nul. • We geven mee dat de credits die moeten worden toegevoegd -1 is.
Verwacht resultaat	De functie zal "true" moeten retourneren en in de database moet een PennyCredit bij de gebruiker zijn verwijderd.
Resultaat 1 ^{ste} uitvoering	De test resulteerde in "true", maar de PennyCredit werd niet bij het account van de User verwijderd. Het probleem kan liggen aan de recente veranderingen die zijn toegevoegd om te controleren dat de PennyCredits van een gebruiker niet negatief mag zijn.
Resultaat 2 ^{de} uitvoering	De test resulteerde in "true", maar de PennyCredit werd niet bij het account van de User verwijderd. Het probleem wordt veroorzaakt doordat de functie een rollback uitvoert nadat de functie de waarde "true" teruggeeft
Resultaat 3 ^{de} uitvoering	De 3 ^{de} test resulteert in "true" en een PennyBid wordt bij de User verwijderd.
Unittest ID	U2
Testomschrijving	We testen het toevoegen van penny credits die afkomstig zijn bij het aanschaffen van een biedpakket bij een gebruiker.
Testdata	• We geven het ID mee van een gebruiker waar zijn credittotaal hoger is dan nul. • We geven mee dat de credits die moeten worden toegevoegd +250 is.

Verwacht resultaat	De functie zal “true” moeten retourneren en in de database moet er 250 PennyCredits bij het account worden toegevoegd.
Resultaat 1^{ste} uitvoering	De test resulteerde in “true”, maar de PennyCredits werd niet bij het account van de User toegevoegd. Het probleem kan liggen aan de recente veranderingen die zijn toegevoegd om te controleren dat de PennyCredits van een gebruiker niet negatief mag zijn.
Resultaat 2^{de} uitvoering	De test resulteerde in “true”, maar de PennyCredits werd niet bij het account van de User toegevoegd. Het probleem wordt veroorzaakt doordat de functie een rollback uitvoert nadat de functie de waarde “true” teruggeeft
Resultaat 3^{de} uitvoering	De 3 ^{de} test resulteert in “true” en een PennyBid wordt bij de User toegevoegd.

Unittest ID	U3
Testomschrijving	We testen het toevoegen/verwijderen van penny credits bij een gebruiker die niet bestaat.
Testdata	• We geven het ID mee van een gebruiker die niet bestaat • We geven mee dat de credits die moeten worden toegevoegd -1 is.
Verwacht resultaat	De functie zal "false" moeten retourneren
Unittest ID	U4
Testomschrijving	We testen het verwijderen van een penny credit bij een gebruiker als hij geen credits heeft.
Testdata	• We geven het ID mee van een gebruiker waar zijn credittotaal 0 is. • We geven mee dat de credits die moet worden toegevoegd -1 is.
Verwacht resultaat	De functie zal "false" moeten retourneren

3. PennyCreditsTransaction model

functienaam	PennyCreditsTransaction
parameters	• PennyAuctionId • UserId • Unix timestamp
	1. De functie checkt of de unix timestamp niet kleiner is dan de timestamp van de huidige tijd. Is deze wel kleiner, dan retourneert de functie "false" 2. De functie houdt zet de data op die bij een tranactie moet wordne opgeslagen. 3. De functie haalt de datasource op van het model om een database transactie te starten. 4. De functie slaat de data op bij PennyCreditsTransaction. Als deze niet kan worden opgeslagen, doe dan een rollback op de transactie en retourneer "false". 5. Bij succes wordt de functie givePennyCredits aangeroepen om een credit eraf te halen. Lukt dit niet, doe dan een rollback en retourneer "false". Anders retourneer "true".
Unittest ID	U1
Testomschrijving	We testen het toevoegen van een transactie bij het plaatsen van een bod bij een actieve centenvailing.
Testdata	• We geven het ID mee van een actieve veiling . • We geven de Id mee van een gebruiker die genoeg PennyCredits heeft. • We geven een unix timestamp mee die groter is dan de timestamp van de huidige tijd.
Verwacht resultaat	De functie zal "true" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "true", maar de PennyCredit werd niet bij het account van de User verwijderd. Het probleem kan liggen aan de recente veranderingen die zijn toegevoegd om te controleren dat de PennyCredits van een gebruiker niet negatief mag zijn.
Resultaat 2^{de} uitvoering	De test resulteerde in "true", maar de PennyCredit werd niet bij het account van de User verwijderd. Het probleem wordt veroorzaakt doordat de functie een rollback uitvoert nadat de functie de waarde "true" teruggeeft

**Resultaat 3^{de}
uitvoering**

De 3^{de} test resulteert in “true” en een PennyBid wordt bij de User verwijderd.

Unittest ID	U2
Testomschrijving	We testen het toevoegen van een transactie bij het plaatsen van een bod bij een centenveiling die al is afgelopen.
Testdata	• We geven het ID mee van een centenveiling die al is afgelopen. • We geven de Id mee van een gebruiker die genoeg PennyCredits heeft. • We geven een unix timestamp mee die kleiner is dan de timestamp van de huidige tijd.
Verwacht resultaat	De functie zal "false" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"
Unittest ID	U3
Testomschrijving	We testen het toevoegen van een transactie bij het plaatsen van een bod bij een centenveiling die niet bestaat.
Testdata	• We geven het ID mee van een centenveiling die niet bestaat. • We geven de Id mee van een gebruiker die genoeg PennyCredits heeft. • We geven een unix timestamp mee die kleiner is dan de timestamp van de huidige tijd.
Verwacht resultaat	De functie zal "false" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"
Unittest ID	U4
Testomschrijving	We testen het toevoegen van een transactie bij het plaatsen van een bod bij een centenveiling waarbij de User niet bestaat.
Testdata	• We geven het ID mee van een centenveiling die actief is. • We geven de Id mee van een gebruiker die niet bestaat. We geven een unix timestamp mee die groter is dan de timestamp van de huidige tijd.
Verwacht resultaat	De functie zal "false" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"
Unittest ID	U5
Testomschrijving	We testen het toevoegen van een transactie bij het plaatsen van een bod bij een centenveiling wanneer de User niet genoeg PennyCredits heeft.
Testdata	• We geven het ID mee van een centenveiling die actief is. • We geven de Id mee van een gebruiker die geen PennyCredits heeft. • We geven een unix timestamp mee die groter is dan de timestamp van de huidige tijd.
Verwacht resultaat	De functie zal "false" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"

functienaam	PennyCreditsBought
parameters	• UserId • Unix timestamp
	1. De functie houdt zet de data op die bij een tranactie moet wordne opgeslagen. 2. De functie haalt de datasource op van het model om een database transactie te starten. 3. De functie slaat de data op bij PennyCreditsTransaction. Als deze niet kan worden opgeslagen, doe dan een rollback op de transactie en retourneer "false". 4. Bij succes wordt de functie givePennyCredits aangeroepen om een credit eraf te halen. Lukt dit niet, doe dan een rollback en retourneer "false". Anders retourneer "true".
Unittest ID	U6
Testomschrijving	We testen het toevoegen van een transactie bij het kopen van PennyCredits.
Testdata	• We geven de Id mee van een bestaande gebruiker mee. • We geven een aantal pennycredits op die groter is dan 0.
Verwacht resultaat	De functie zal "true" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "true", maar de PennyCredit werdt niet bij het account van de User verwijderd. Het probleem kan liggen aan de recente veranderingen die zijn toegevoegd om te controleren dat de PennyCredits van een gebruiker niet negatief mag zijn.
Resultaat 2^{de} uitvoering	De test resulteerde in "true", maar de PennyCredit werd niet bij het account van de User verwijderd. Het probleem wordt veroorzaakt doordat de functie een rollback uitvoert nadat de functie de waarde "true" teruggeeft
Resultaat 3^{de} uitvoering	De 3 ^{de} test resulteert in "true" en een PennyBid wordt bij de User verwijderd.
Unittest ID	U7
Testomschrijving	We testen het toevoegen van een transactie bij het kopen van PennyCredits.
Testdata	• We geven de Id mee van een gebruiker mee die niet bestaat. • We geven een aantal pennycredits op die groter is dan 0.
Verwacht resultaat	De functie zal "false" moeten retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"

4. PennyAuction model

functienaam	getAuctions
parameters	• categoryName
	1. De functie checkt of er AuctionItems bestaan die behoren onder de categoryName. Als dit waar is, dan worden de actieve PennyAuctions opgehaald van de desbetreffende categoryName. Als de categoryName "closed" is, worden de PennyAuctions opgehaald die zijn gesloten. Als de categoryName niks bevat, worden alle actieve PennyAuctions opgehaald. 2. Voor elke PennyAuction worden de biedingen geteld om de CurrentBid op te slaan en om een BidHistory aan te maken van de laatste 5 bidders. 3. De functie retourneert een array van PennyAuctions
Unittest ID	PA1
Testomschrijving	We testen of er per PennyAuction in de array de objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] aanwezig zijn, waarbij alle PennyAuctions toebehoren aan de juiste Category
Testdata	• We geven de categoryName mee van een Category waar PennyAuctions aanwezig zijn. • We geven de array objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] om te checken of deze in het resultaat aanwezig is.
Verwacht resultaat	De functie zal aan moeten geven dat de objecten aanwezig zijn bij de PennyAuctions in het resultaat.
Resultaat 1^{ste} uitvoering	De test resulteerde in "true"
Unittest ID	PA2
Testomschrijving	We testen of er geen PennyAuction worden teruggegeven.
Testdata	• We geven de categoryName mee van een Category waar geen PennyAuctions aanwezig zijn. • We geven een lege array mee omdat we geen resultaat verwachten
Verwacht resultaat	De lege array moet gelijk zijn aan het resultaat.
Resultaat 1^{ste} uitvoering	De test resulteerde in "true"
Unittest ID	PA3
Testomschrijving	We testen of er per PennyAuction in de array de objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] aanwezig zijn, bij alle PennyAuctions die zijn gesloten.
Testdata	• We geven de categoryName "closed" mee. • We geven de array objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] om te checken of deze in het resultaat aanwezig is.
Verwacht resultaat	De functie zal aan moeten geven dat de objecten aanwezig zijn bij de PennyAuctions in het resultaat.
Resultaat 1^{ste} uitvoering	De test resulteerde in "true"

functienaam	getAuction
parameters	PennyAuctionId
	- De functie checkt of een PennyAuction bestaat met het opgevraagde PennyAuctionId. Bij geen vondst retourneert de functie "Mr. Zwoози cannot find the auction you are looking for". - De functie halt de PennyAuction op die bij de PennyAuctionId toebehoort. - Voor elke PennyAuction worden de biedingen geteld om de CurrentBid op te slaan en om een BidHistory aan te maken van de laatste 5 bidders. - De functie retourneert een array van de PennyAuction
Unittest ID	PA4
Testomschrijving	We testen het ophalen van een bestaande PennyAuction en of de PennyAuction array de objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] bevat.
Testdata	- We geven een PennyAuctionId mee van een bestaande centenveiling. - We geven de array objecten ['PennyAuction', 'AuctionItem', 'currentBid', 'BidHistory'] om te checken of deze in het resultaat aanwezig is.
Verwacht resultaat	De functie zal aan moeten geven dat de objecten aanwezig zijn bij de PennyAuction array van het resultaat.
Resultaat 1 ^{ste} uitvoering	De test resulteerde in "true"
Unittest ID	PA5
Testomschrijving	We testen of het ophalen van een niet bestaande PennyAuction een foutmelding teruggeeft.
Testdata	We geven een PennyAuctionId mee van een bestaande centenveiling.
Verwacht resultaat	De functie moet "Mr. Zwoози cannot find the auction you are looking for" retourneren.
Resultaat 1 ^{ste} uitvoering	De test resulteerde in "false"

functienaam	getLatestPennyBid
parameters	PennyAuctionId
	- De functie checkt of hij de laatste PennyBid van een PennyAuction kan ophalen. De lijst met gevonden PennyBids worden aflopend gesorteerd op de datum wanneer ze toegevoegd zijn. Wanneer dit niet mogelijk is, retourneert hij "false" - De functie retourneert de eerste Pennybid uit de gevonden Pennybids van een PennyAuction.
Unittest ID	PA6
Testomschrijving	We testen of de functie de laatst geplaatste PennyBid bij een PennyAuction kan ophalen.
Testdata	We geven een PennyAuctionId mee van een bestaande centenveiling die PennyBids heeft.
Verwacht resultaat	De functie moet een array van de laatste PennyBid retourneren.

Resultaat 1^{ste} uitvoering	De test resulteerde in "true"
---	-------------------------------

Unittest ID	PA7
Testomschrijving	We testen of de functie de laatst geplaatste PennyBid bij een PennyAuction kan ophalen die geen PennyBids heeft.
Testdata	We geven een PennyAuctionId mee van een bestaande centenveiling die geen PennyBids heeft.
Verwacht resultaat	De functie moet een "false" retourneren.
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"

Unittest ID	PA8
Testomschrijving	We testen of de functie de laatst geplaatste PennyBid bij een niet bestaande PennyAuction kan ophalen.
Testdata	We geven een PennyAuctionId mee van een niet bestaande PennyAuction.
Verwacht resultaat	De functie moet een "false" retourneren.
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"

5. PennyAuctionPurchase model

functienaam	addPurchase
parameters	• PennyAuctionId • UserId
	3. De functie haalt de PennyAuction op aan de hand van de opgegeven PennyAuctionId. 4. De functie controleert of de PennyAuction de status 1 of 2 en dat de huidige tijd kleiner is 24 uur na de afloop van een PennyAuction. Als dit niet zo is, dan retourneert de functie "false" 5. De functie haalt alle PennyBids die de User heeft geplaatst op de PennyAuction. Aan de hand van het aantal PennyBids stelt de functie een discount op. 6. De functie zet de opslagdata vast van de PennyAuctionPurchase. Als de PennyAuction is afgerond en de huidige User is de winnaar, dan wordt de prijs geupdated naar de huidige biedprijs van de PennyAuction. 7. De functie slaat de data op doormiddel van een databasetransactie. Bij succes retourneert die "true", bij falen retourneert hij "false"
Unittest ID	PAP1
Testomschrijving	We testen of de functie een PennyAuctionPurchase voor de verloopdatum kan toevoegen.
Testdata	• We geven een PennyAuctionId waarbij de verloopdatum nog niet is bereikt. • We geven de UserId van een bestaande gebruiker mee.
Verwacht resultaat	De functie moet "true" retourneren
Resultaat 1 ^{ste} uitvoering	De test resulteerde in "true"
Unittest ID	PAP2
Testomschrijving	We testen of de functie een PennyAuctionPurchase na de verloopdatum kan toevoegen.
Testdata	• We geven een PennyAuctionId waarbij de verloopdatum al is bereikt. • We geven de UserId van een bestaande gebruiker mee.
Verwacht resultaat	De functie moet "false" retourneren
Resultaat 1 ^{ste} uitvoering	De test resulteerde in "false"

Unittest ID	PAP3
Testomschrijving	We testen of de functie een PennyAuctionPurchase kan toevoegen bij de winnende gebruiker.
Testdata	• We geven een PennyAuctionId op waarbij de verloopdatum al is bereikt. • We geven de UserId van een bestaande gebruiker mee.
Verwacht resultaat	De functie moet "true" retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "true"

functienaam	finalizePurchase
parameters	• PennyAuctionId • UserId
	1. De functie checkt of er een PennyAuctionPurchase bestaat met de PennyAuctionId. Als er geen PennyAuctionPurchase kan worden gevonden, dan retourneert de functie "false" 2. De functie update de PennyAuctionpurchase dat hij is betaald en retourneerd "succes". Bij het falen van het updaten retourneert de functie "failure".

Unittest ID	PAP4
Testomschrijving	We testen de functie om een PennyAuctionPurchase af te ronden die nog niet betaald is.
Testdata	• We geven een PennyAuctionPurchaseId waarbij de verloopdatum nog niet is bereikt. • We geven de confirmatie "true" mee • We geven de UserId van de eigenaar van de aankoop mee.
Verwacht resultaat	De functie moet "success" retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "succes"

Unittest ID	PAP5
Testomschrijving	We testen de functie om een PennyAuctionPurchase af te ronden die niet bestaat.
Testdata	• We geven een PennyAuctionPurchaseId die niet bestaat. • We geven de confirmatie "true" mee. • We geven de UserId van de eigenaar van de aankoop mee.
Verwacht resultaat	De functie moet "Not found" retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "Not found"

Unittest ID	PAP6
Testomschrijving	We testen de functie om een PennyAuctionPurchase af te ronden die al betaald is.
Testdata	• We geven een PennyAuctionPurchaseId mee die al is betaald • We geven de cofnirmatie "true" mee • We geven de UserId van de eigenaar van de aankoop mee.
Verwacht resultaat	De functie moet "Not found" retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "Not found"

Unittest ID	PAP7
Testomschrijving	We testen de functie om een PennyAuctionPurchase af te ronden als de betaling niet is bevestigd.
Testdata	• We geven een PennyAuctionPurchaseId mee die nog niet betaald is • We geven de confirmatie "false" mee • We geven de UserId van de eigenaar van de aankoop mee.
Verwacht resultaat	De functie moet "false" retourneren
Resultaat 1^{ste} uitvoering	De test resulteerde in "false"

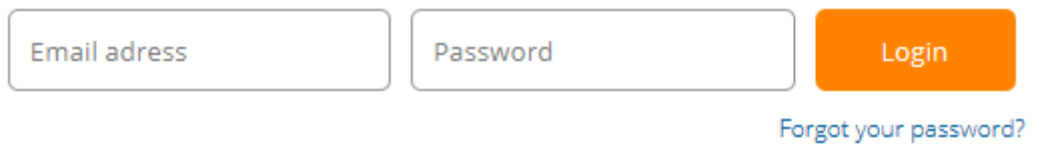
6. Performance acceptatietest

7. Log in op Zwoozi:

Bij het bezoeken komt de gebruiker op de inlog-pagina. Gebruikers die nog geen account hebben kunnen zich hier registreren. Voor het testen kan de testgebruiker inloggen met:

Emailadres: testuser@zwoozi.com

Wachtwoord: test1234



The login form consists of two input fields and a button. The first field is labeled 'Email adress' and the second is labeled 'Password'. To the right of the password field is an orange button labeled 'Login'. Below the password field is a link that says 'Forgot your password?'.

Email adress	Password	Login
--------------	----------	-------

[Forgot your password?](#)

8. Bieden op een veiling zonder biedcredits:

Via de URL zwoozzi.dev/pennyauctions kan de testgebruiker het overzicht van de actieve centenveilingen zien. Bij deze veilingen is de aflooptijd, huidige bidleider, de huidige prijs en de foto van het product aanwezig per actieve veiling.

Per veiling is er een knop aanwezig waarmee je op de veiling kan bieden. Het testaccount heeft echter nog geen biedcredits om op een veiling te kunnen bieden.

[View Pennyauctions](#)



Bied op een veiling zonder dat je biedcredits heb. Je krijgt een melding te zien dat je niet genoeg biedcredits bezit en deze moet aanschaffen. Het bod wordt niet aan een veiling toegevoegd en de prijs zal niet veranderen.

De requirements bij deze uitvoering zijn:

- [Use Case plaatsen bod] Als de gebruiker niet genoeg biedcredits bezit, dan.... geeft het systeem aan dat de gebruiker niet genoeg biedcredits bezit. Het bod kan niet worden geplaatst.

Wat werkt er niet bij deze uitvoering:

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Andere weergave bieden niet mogelijk

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

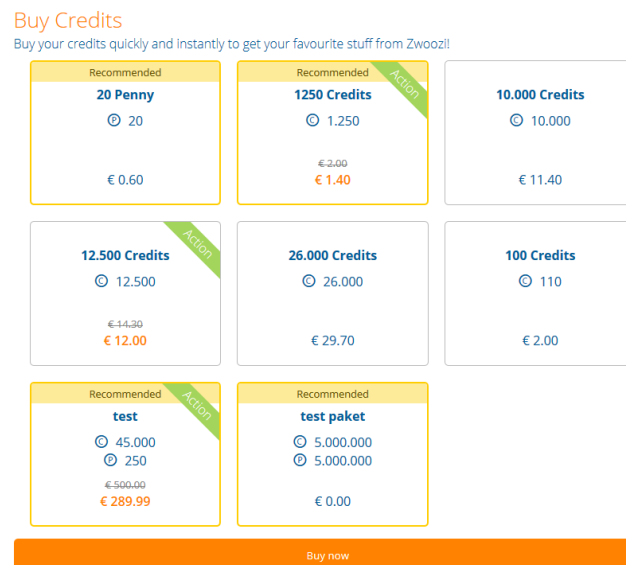
Ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

Ja

9. Kopen van biedcredits

Via de “get credits” wordt je doorgestuurd naar de pagina waar je credits kan kopen. Er worden een aantal pakketten weergegeven die per pakket een ander aantal credits bevat. Er wordt onderscheid gemaakt tussen 2 soorten credits. Credits en biedcredits. In het pakket wordt credits aangegeven met een C en biedcredits wordt aangegeven met een P.



Selecteer een biedpakket dat biedcredits bevat en druk op de knop “buy now”. De pagina zal een bericht geven dat je het pakket heb gekocht. In de balk bovenaan de pagina moeten de biedcredits toegevoegd worden aan het totaal van de testaccount.

Selecteer vervolgens geen pakket druk op de knop “buy now”. Je krijgt een melding met “No package selected”.

De requirements bij deze uitvoering zijn:

- [Use case kopen biedcredits] De gebruiker geeft aan dat hij een pakket met biedcredits wil kopen.
- [Use case kopen biedcredits] Het systeem toont de bevestiging aan de gebruiker en voegt de gekochte biedcredits toe aan zijn account.

Wat werkt er niet bij deze uitvoering:

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Meldingen duidelijker en pakketten biedcredits en credits duidelijker qua onderscheid.

“Buy now” knop locatie moet worden veranderd om sneller door te kunnen klikken. Zo weinig mogelijk klikken om functioneel te zijn.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

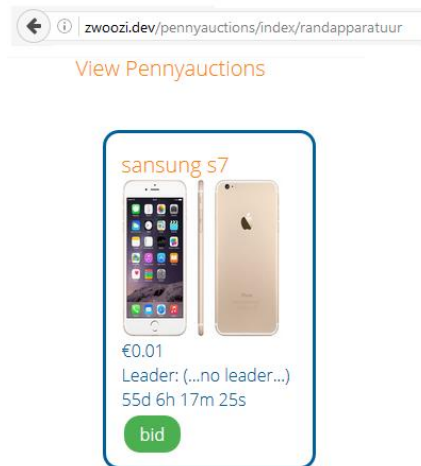
Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

10. Zoeken naar een veiling op basis van categorieën:

Via de URL zwoozi.dev/pennyauctions kan de testgebruiker het overzicht van de actieve centenveilingen zien. Bij deze veilingen is de aflooptijd, huidige bidleider, de huidige prijs en de foto van het product aanwezig per actieve veiling.

Op het moment krijg je alle centenveilingen te zien die actief zijn. Ga naar de URL zwoozi.dev/pennyauctions/index/randapparatuur om alle centenveilingen in de categorie opslagmedia weer te geven.



Ga naar de URL zwoozi.dev/pennyauctions/index/antiek om alle centenveilingen in de categorie antiek weer te geven. Er zal niks worden weergegeven doordat er geen centenveilingen in deze categorie aanwezig zijn.

Ga naar de URL zwoozi.dev/pennyauctions/index/closed om alle gesloten centenveilingen weer te geven. De getoonde centenveilingen hebben als aflooptijd dat ze “Finished” zijn en dat de centenveiling is gesloten.

De requirements bij deze uitvoering zijn:

- [Use Case zoeken centenveiling] Het systeem vraagt naar de gewenste zoekcriteria.
- [Use Case zoeken centenveiling] Het systeem zoekt de actuele centenveilingen op die voldoen aan de zoekcriteria en toont deze aan de gebruiker
- [Use Case zoeken centenveiling] Als er geen centenveilingen gevonden worden die voldoen aan de zoekcriteria dan geeft het systeem weer dat er geen resultaten zijn gevonden.

Wat werkt er niet bij deze uitvoering:

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Makkelijker bij categorieën komen (lijst categorieën) .

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

11. Bieden op een veiling met biedcredits:

Via de URL zwooz.dev/pennyauctions kan de testgebruiker het overzicht van de actieve centenveilingen zien. Bij deze veilingen is de aflooptijd, huidige bidleider, de huidige prijs en de foto van het product aanwezig per actieve veiling.

Per veiling is er een knop aanwezig waarmee je op de veiling kan bieden. Het testaccount heeft nu biedcredits sinds je die net heb gekocht.

Bied op een veiling. De prijs van centenveiling zal met één cent toenemen. De leidende bieder zal verandert worden naar testuser, behalve als testuser als de leidende bieder is.

De requirements bij deze uitvoering zijn:

- [Use Case plaatsen bod] De gebruiker geeft aan dat hij een bod wil plaatsen op de gewenste veiling.
- [Use Case plaatsen bod] Het systeem controleert of de gebruiker kan bieden.
- [Use Case plaatsen bod] Het systeem plaatst het bod en haalt de biedkosten van de gebruiker zijn account af.
- [Use Case plaatsen bod] Het systeem werkt de veiling bij en verhoogd de biedprijs. Het systeem werkt de bidhistorie bij en zet de gebruiker als leidende bieder.

Wat werkt er niet bij deze uitvoering:

Uitvoering werkt alleen gebruiker mag niet twee keer bieden als hij of zij boven aan staat.

Welke verbeteringen moeten er worden aangebracht:

Biedcredits moeten wel afnemen als je bid niet eerst de page refreshen om te zien hoeveel je over hebt.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

Ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

12. Bieden op de details pagina van een centenveiling:

Via de URL zwoozu.dev/pennyauctions kan de testgebruiker het overzicht van de actieve centenveilingen zien. Druk bij een centenveiling op de titel. De pagina zal je nu herleiden naar de details pagina van de centenveiling.

Op de details pagina is direct bieden aanwezig, evenals een knop voor een bidhulpje, gegevens over het product, de bidhistorie en de gerelateerde gesloten veilingen.

testitem 1



direct buy: [Buy now!](#)

€0.22

Leader: testuserLeet

Time left: 0d 1h 14m 59s

Bid [Activate bidbuddy](#)

Product details

store price: 200.00

delivery time: 3 days

product stock: left

description

orum ipsum

BidHistory

testuserLeet
pinoswine
testuserLeet
pinoswine
testuserLeet
pinoswine
testuserLeet
pinoswine
testuserLeet
pinoswine

Previous pennyauctions

iphone 7



\$0.23
Closed

iphone 7



\$0.07
Closed

iphone 7



\$0.01
Closed

iphone 7



\$0.01
Closed

iphone 7



\$0.28
Closed

Bied nu nog een keer op de veiling. Zoals in de vorige test zal de biedprijs verhoogd worden. De bidhistorie wordt bijgewerkt en testuser1234 zal nu bovenaan de bidhistorie staan.

De requirements bij deze uitvoering zijn:

- [Use Case plaatsen bod] De gebruiker geeft aan dat hij een bod wil plaatsen op de gewenste veiling.
- [Use Case plaatsen bod] Het systeem controleert of de gebruiker kan bieden.
- [Use Case plaatsen bod] Het systeem plaatst het bod en haalt de biedkosten van de gebruiker zijn account af.
- [Use Case plaatsen bod] Het systeem werkt de veiling bij en verhoogd de biedprijs. Het systeem werkt de bidhistorie bij en zet de gebruiker als leidende bieder.

Wat werkt er niet bij deze uitvoering: werkt

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Ook hier sneller laten zien hoeveel credits je over hebt en niet mogelijk boven op eigen bod bieden.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

13. Direct kopen van een veilingproduct:

Op de details pagina is een “direct buy” knop aanwezig. Via deze knop kan een gebruiker het product kopen zonder hij tot aan het einde van de centenveiling moet deelnemen. De veiling op de details pagina is nog actief.

Druk op de “direct buy” knop en leg een aankoop vast. De pagina leidt je door naar een aankooppagina waar je aankopen worden weergegeven. De aankoop die zojuist is toegevoegd zich bevinden tussen de lijst met de status “Not paid”. Daarnaast zie je de tijd aanwezig die aangeeft hoelang het duurt voordat de mogelijk om de aankoop te betalen zal vervallen.

De requirements bij deze uitvoering zijn:

- [User requirement] De gebruiker wil het product van een centenveiling direct kunnen kopen.
- [Software requirement] Het systeem moet de gebruiker een product direct kunnen laten kopen.
- [software requirement] Het systeem moet een korting berekenen bij het direct kopen als de gebruiker op de centenveiling heeft geboden.
- [software requirement] Het systeem moet bij de aankoop van de winnende gebruiker de biedprijs als aankoopprijs stellen.

Wat werkt er niet bij deze uitvoering: werkt

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Uiterlijk moet verbeterd worden en daadwerkelijk functie om te kunnen betalen via IDEAL, Paypal enz. erin zetten.

Overzicht verbeteren van “pay now” niet teveel producten laten zien als je op “pay now” heb gedrukt.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

14. Betalen van een productaankoop:

Op de aankoop pagina zie je de aankoop die je zojuist heb toegevoegd. Druk op de knop “pay” om naar de gegevensbevestigingspagina te gaan. Op de bevestigingspagina worden de gegevens van de het product en de aankoop weergegeven. Bij het afleveradres kan de gebruiker zijn informatie invullen.

Product	date	final price	status	payment	time left
 iPhone 7	17-09-16	€ 498.80	Paid	<button>Pay</button>	Finished
 iPhone 7	23-09-16	€ 491.60	Paid	<button>Pay</button>	Finished
 iPhone 7	23-09-16	€ 491.60	Paid	<button>Pay</button>	Finished
 iPhone 7	23-09-16	€ 491.60	Not paid	<button>Pay</button>	Finished

Druk op “edit” om de velden invulbaar te maken. Daarna kan je het adres invullen. Na het invullen selecteer je een betaalmethode. De pagina zal je terugsturen naar de aankooppagina en geeft als bericht aan dat de aankoop is betaald. De status van de aankoop is gewijzigd naar “Paid”.

product title:

Price amount:

Shipping fee:

Total price:

Delivery time:

Street name:

Zipcode:

country:

Phone number:

De requirements bij deze uitvoering zijn:

- [Use Case betalen centenveiling] Het systeem haalt de aankoop- en adresgegevens op en toont deze aan de gebruiker. Het systeem vraagt of deze gegevens kloppen. De gebruiker bevestigt dit.
- [use Case betalen centenveiling] Het systeem haalt de beschikbare betaalmethoden op en toont deze aan de gebruiker.
- [Use case betalen centenveiling] als het systeem ziet dat er bij de gebruiker nog geen adresgegevens aanwezig zijn, dan vraagt het systeem of de gebruiker zijn adresgegevens wil invullen. De gebruiker vult deze gegevens in.
- [Use Case betalen centenveiling] als de gebruiker aangeeft dat zijn adresgegevens onjuist zijn,

dan geeft het systeem de gebruiker de mogelijkheid om de adresgegevens te wijzigen. De gebruiker wijzigt deze data en geeft aan dat hij de data wil bijwerken. Het systeem werkt de adresgegevens bij.

- [Use Case betalen centenveiling] Het systeem ontvangt een bevestiging van de desbetreffende betaalservice aanbieder dat het betalen is gelukt en legt dit vast.

Wat werkt er niet bij deze uitvoering: het werkt

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Layout en knoppen moeten mooier en strakker worden gemaakt.

Het betalen moet simpeler gemaakt worden.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

15. Toevoegen van een veilingproduct:

Nu ga je de functionaliteiten testen die bedoeld zijn voor de beheerder. Ga naar de URL zwoozi.dev/AuctionItems om het beheerpaneel van de veilingproducten te bezichtigen. Om een product toe te voegen.

Manage Auction Items

[Add new item](#)

iphone 7		€ 500.00	delete	edit
samsung s7		€ 500.00	delete	edit
testitem 1		€ 200.00	delete	edit
testitem 26		€ 200.00	delete	edit
testitem25		€ 200.00	delete	edit

Create a new Auction Item

Name*

Description*

Amount of storage

Item type*

product

Price*

Delivery time

Credits

Category*

-- Select a category --

Select a Supplier

-- Select a supplier --

Upload photos

Item Image

[Browse...](#) No files selected.

Create auction item!

Op deze pagina worden een aantal velden weergegeven die moeten worden ingevuld. Niet alle velden hoeven te worden ingevuld. Er kunnen 3 soorten veilingproducten worden aangemaakt. Leverancierproducten, creditproducten en een combinatie van de 2.

We maken alleen een leverancierproduct aan. Vul de velden in die aangegeven staan met een *, Vervolgens vul je de velden in van "amount of storage", selecte

Auctions

[Add new auction](#)

Auctions that you might find interesting

testitem 1	inactive		Edit	Delete	Activate
samsung s7	active		Edit	Delete	
iphone 7	active		Edit	Delete	
iphone 7	completed		Edit	Delete	Recycle

er product bij “product type” en geef de “delivery time” op. Na het invullen kan je een afbeelding kiezen om bij het product weer te geven.

Druk vervolgens op de knop “Create Auction Item” om het product toe te voegen. Na het drukken op de knop wordt je naar de overzichtspagina van veilingproducten geleid. Het product wat je zojuist heb toegevoegd zal in de lijst aanwezig moeten zijn.

De requirements bij deze uitvoering zijn:

- [Use case toevoegen veilingproduct] Het systeem vraagt naar de productgegevens. De medewerker voert deze data in.
- [Use case toevoegen veilingproduct] Het systeem vraagt naar afbeeldingen voor het product. De medewerker voegt deze afbeeldingen toe.
- [Use case toevoegen veilingproduct] Het systeem toont het product met bijbehorende afbeeldingen en vraagt om een bevestiging. De medewerker bevestigt het product.
- [Use Case betalen veilingproduct] als de gebruiker verkeerde data invoert, dan geeft het systeem een foutmelding aan en vraagt de gebruiker om een om de juiste data in te voeren.

Wat werkt er niet bij deze uitvoering:

Geen opmerkingen

Welke verbeteringen moeten er worden aangebracht:

Het werkt aantal velden moeten nog aangepast worden voor de rest werkt het.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

16. Wijzigen van een veilingproduct:

Ga naar de URL zwoozl.dev/AuctionItems om de veilingproducten te bezichtigen. Bij de veilingproducten is een “edit” en een “delete” knop aanwezig.

Druk op de “edit” knop om een veilingproduct aan te passen. Je wordt naar de edit pagina geleid die er hetzelfde uitziet als bij het aanmaken van een veilingproduct. Integendeel tot de toevoegingspagina, zijn de velden hier ingevuld.

Pas de velden aan die zijn ingevuld. Na het invullen moet je een nieuwe afbeelding uploaden. Upload een nieuwe afbeelding en druk op de knop om de wijzigingen door te voeren. Nadat de wijzigingen zijn doorgevoerd wordt je teruggeleid naar de index pagina van veilingproducten.

De requirements bij deze uitvoering zijn:

- [Use case aanpassen veilingproduct] Het systeem haalt het gekozen product op en toont deze aan de gebruiker.
- [Use case aanpassen veilingproduct] De medewerker past de informatie aan die hij bij het product wil veranderen en geeft vervolgens aan dat hij de informatie wil opslaan.
- [Use case aanpassen veilingproduct] Het systeem slaat de wijzigingen op.

Wat werkt er niet bij deze uitvoering:

Wijzigen van product werkt niet als er geen afbeelding word bijgevoegd dan komt er een error.

Met afbeelding werkt het wel

Delete knop werkt ook.

Welke verbeteringen moeten er worden aangebracht:

Uiterlijk, overzicht en het editen van een veiling.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

17. Verwijderen van een veilingproduct:

Ga naar de URL zwoozl.dev/AuctionItems om de veilingproducten te bezichtigen. Bij de veilingproducten is een “edit” en een “delete” knop aanwezig.

Druk op de “delete” knop van het veilingproduct dat je net hebt aangemaakt om het te verwijderen. Na het indrukken wordt de pagina herladen is moet het product uit de lijst verdwenen zijn.

Druk op de “delete” knop van het veilingproduct “iphone 7” om deze te verwijderen. Dit product is echter nog in gebruik bij een actieve centenveiling en kan niet worden verwijderd. De pagina geeft een foutmelding dat product niet kan worden verwijderd.

De requirements bij deze uitvoering zijn:

- [Use case verwijderen veilingproduct] De medewerker geeft aan dat hij het gewenste veilingproduct wil verwijderen.
- [Use case verwijderen veilingproduct] Het systeem controleert of het product zich niet in een actuele veiling bevindt.
- [Use case verwijderen veilingproduct] Het systeem geeft aan dat de centenveiling wordt verwijderd.
- [Use case verwijderen veilingproduct] als het product zich nog in een actuele veiling bevindt, dan geeft het systeem aan dat het product nog wordt gebruikt en niet kan worden verwijderd

Wat werkt er niet bij deze uitvoering:

Geen opmerkingen

Welke verbeteringen moeten er worden aangebracht:

Overzicht en uiterlijk

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

Ja het werkt goed

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

jazeker

18. Aanmaken van een centenveiling:

Ga naar de URL zwoozzi.dev/PennyAuctions/management om op de pagina te komen waar centenveilingen kunnen worden aangemaakt en beheerd worden. Om een centenveiling aan te maken, druk op de “Add new auction” knop.

Auctions

[Add new auction](#)

Auctions that you might find interesting

testitem 1	inactive		Edit	Delete	Activate
samsung s7	active		Edit	Delete	
iphone 7	active		Edit	Delete	
iphone 7	completed		Edit	Delete	Recycle

Bij het aanmaken van de veiling wordt om 2 dingen gevraagd. De duur van de veiling en welk product er moet worden geveild. Kies als product het veilingproduct wat je hebt aangemaakt. Vul deze data in en druk op de knop om de centenveiling aan te maken. Je wordt doorgestuurd naar de activatiepagina waar de zojuist aangemaakte centenveiling wordt weergegeven.

Create new auction

Want to get rid of your old stuff? Put it on Zwooz!

Duration(in days)

Item to auction

-- Select an item --

Create Pennyauction!

De requirements bij deze uitvoering zijn:

- [Use case aanmaken centenveiling] Het systeem vraagt naar de gewenste duur van de centenveiling. De medewerker voert deze data in.
- [Use case aanmaken centenveiling] Het systeem haalt de beschikbare producten op en toont deze aan de medewerker.
- [Use case aanmaken centenveiling] De gebruiker selecteert het product dat hij wil toevoegen aan de veiling. Het systeem legt deze aanvraag vast.
- Het systeem toont de veilinggegevens samen met de productgegevens en vraagt om een bevestiging. De medewerker bevestigt de veilinggegevens.

Wat werkt er niet bij deze uitvoering:

Het werkt goed

Welke verbeteringen moeten er worden aangebracht:

Niet echt een opmerking wel het uiterlijk verfraaien.

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

19. verwijderen van een centenveiling:

Ga naar de op de management pagina waarbij een lijst van centenveilingen wordt getoond. Een veiling heeft verschillende knoppen op basis van hun status. Verwijder een centenveiling dat nog actief is. Je krijgt een melding dat de veiling niet kan worden verwijderd.

Ga nu terug en verwijder een veiling die compleet is. Deze veiling kan wel worden verwijderd.

De requirements bij deze uitvoering zijn:

- [Use case verwijderen centenveiling] Het systeem vraagt de medewerker of hij de centenveiling wil stoppen. De gebruiker accordeert.
- [Use case verwijderen centenveiling] Het systeem geeft aan dat de centenveiling wordt verwijderd.
- [Use case verwijderen centenveiling] in stap 3, als het systeem het product probeert te verwijderen wanneer die nog actief is, dan moet het systeem de biedingen retourneren aan de gebruikers en de gebruikers informeren dat de centenveiling wordt verwijderd.
- [software requirement] Het systeem moet bij het verwijderen van een centenveiling de gebruikers informeren over de verwijdering voordat de veiling wordt verwijderd.
- [software requirement] Het systeem moet bij het verwijderen de bestede biedcredits retourneren aan de bidders.

Wat werkt er niet bij deze uitvoering:

Geen opmerking

Welke verbeteringen moeten er worden aangebracht:

Geen opmerking

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

20. activeren van een centenveiling:

Ga naar de management van centenveiling om een lijst met centenveilingen te zien. De centenveiling die je eerder hebt aangemaakt bevat de knop om een veiling te activeren. Activeer de veiling. De pagina zal worden herladen en de veiling die je hebt geactiveerd zal veranderen van status.

Ga naar de URL `zwoozi.dev/PennyAuctions/index/category` waar categorie de categorie is van het veilingproduct die wordt geveild aan de hand van je geactiveerde veiling. De centenveiling die je zojuist heb aangemaakt zal zich tussen de centenveilingen bevinden.

De requirements bij deze uitvoering zijn:

- [Use case activeren centenveiling] Het systeem vraagt om een bevestiging voor het activeren. De medewerker bevestigt dit.
- [Use case activeren centenveiling] Het systeem Het systeem verandert de status naar actief en laat de veiling starten.

Wat werkt er niet bij deze uitvoering:

Het werkt goed

Welke verbeteringen moeten er worden aangebracht:

Geen opmerking

Geeft het prototype de juiste werking weer van de functionaliteit zoals u dat had verwacht?

ja

Voldoet de functionaliteiten aan de eisen die u stelt aan het prototype?

ja

Heeft het prototype u een goede indicatie gegeven over hoe het systeem eruit moet komen te zien?

Het geeft al een duidelijke weergave hoe het eruit moet komen te zien.

Zo niet, waar voldoet het prototype niet aan?

Voldoet het prototype aan de eisen die u gedurende het project heb aangekaart?

Ja de functies werken daar gaat het om.

Zijn er nog andere opmerkingen?

Ik heb beoordeeld op functionaliteit en het werkt. Het enige wat ik heb aan te merken is het versimpelen van processen en ze overzichtelijker maken meer samenhang met elkaar en daarnaast het mooi maken van alles.

Krijgt het prototype een Go of een NoGo?

Go/NoGo

GO

Naam testpersoon:
Nick Verheij

Functie binnen Zwoozi:
Producteigenaar

Datum uitgevoerd:
28-09-2016

Handtekening: