



# Afstudeerverslag

*Ontwikkeling van beheersoftware voor online casino systeem*

Afstudeerverslag HBO Informatica aan de Academie van ICT & Media Zoetermeer

In de periode 8 februari 2010 tot en met 4 juni 2010

Door Gertjan Al, studentnummer 20069275

Bij bedrijf MC Advies & Management Ltd.

Voor bedrijf Gaming & Gambling, Mejdí NV



---

Document: Afstudeerverslag  
Onderdeel van: Afstuderen Gertjan AI, 20069275  
Versie: 2-6-2010 2:15

# Afstudeerverslag

*Ontwikkeling van beheerssoftware voor online casino systeem*

*Afstudeerverslag HBO Informatica aan de Academie van ICT & Media Zoetermeer*

*In de periode 8 februari 2010 tot en met 4 juni 2010*

*Door Gertjan Al, studentnummer 20069275*

*Bij bedrijf MC Advies & Management Ltd.*

*Voor bedrijf Gaming & Gambling, Mejdí NV*





---

Document: Afstudeerverslag  
Onderdeel van: Afstuderen Gertjan AI, 20069275  
Versie: 2-6-2010 2:15



## Referaat

Dit afstudeerverslag beschrijft het afstudeerproces van Gertjan Al. Het afstudeerproject is gehouden bij het bedrijf MC Advies & Management, in opdracht van het bedrijf Gaming & Gambling. De auteur volgt de HBO opleiding Informatica aan de Academie van ICT & Media in Zoetermeer, onderdeel van de Haagse Hogeschool.

Gaming & Gambling heeft enkele online casino's, met zowel casinospellen als pokerspellen. Met het oog op fikse uitbreidingen van servers en medewerkers dient voor deze singleplayer en multiplayer softwarepakketten een nieuw overkoepelend beheersysteem gemaakt te worden. De auteur zal dit systeem specificeren, ontwerpen en implementeren.

## *Descriptoren*

Iterative Application Development

Usage Centered Design

Java JBoss Seam

Java Hibernate

Java Richfaces



## Voorwoord

Het bedrijf Gaming & Gambling beschikt over twee grote online softwarepakketen waarmee dagelijks vele mensen ter wereld hun geluk beproeven op gokspelen zoals fruitmachines, bingo en videogames. De uitdaging van deze opdracht is het samenvoegen van de beheerdersomgeving van deze twee pakketten.

De opdracht is ontstaan vanwege een uitbreiding van het aantal casino's en medewerkers bij Gaming & Gambling. Het bedrijf MC Advies & Management zal in opdracht van G&G de nieuwe beheerderssoftware ontwikkelen. De opdrachtgever zal dhr. M. de la Rie zijn, de bedrijfsmentor dhr. B. Schlärmann. Ik wil beide heren hartelijk bedanken voor hun hulp, uitleg en kritiek bij het uitvoeren van dit project. Daarnaast wil ik dhr. Albert en dhr. Astley bedanken voor hun mentale steun bij het uitvoeren van dit project.

Het afstudeerproject is gevolgd door twee docenten van de Academie van ICT & Media. Ik wil begeleider-examinator dhr. A.A. Nederend hartelijk bedanken voor zijn steun, kritieken en tips voor het voltooien van dit project en expert-examinator dhr. dr. T. Cocx voor de geleverde kritieken op het verslag en documentatie. Met behulp van deze opmerkingen is het verslag geworden wat het nu is.

Leidschendam, 31 mei 2010

Gertjan Al



# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>10</b>
<b>2</b>	<b>Over het bedrijf</b>	<b>11</b>
2.1	Positie van het afstudeerproject binnen het bedrijf	11
2.2	Projectomgeving	11
<b>3</b>	<b>Afstudeeropdracht</b>	<b>12</b>
3.1	Casino- en pokersoftware	12
3.2	Overeenkomsten	12
3.3	Beheer	13
3.4	Ontstaan van de opdracht	13
3.5	Mogelijke knelpunten	14
<b>4</b>	<b>Tot stand komen van het Plan van Aanpak</b>	<b>15</b>
4.1	Kiezen van ontwikkelmethode	15
4.2	Keuze voor Iterative Application Development	15
4.2.1	Gebruik van pilots	17
4.3	Opstellen van systeemeisen	17
4.3.1	Mogelijkheden van de huidige systemen	17
4.3.2	Systeemeisen aan de hand van vergadering eindgebruikers	20
4.3.3	Functionele eisen voor de Game Management Service	21
4.3.4	Niet functionele eisen voor de Game Management Service	22
4.3.5	Afbakening	23
4.3.6	Gebruik van Usage Centered Design	23
<b>5</b>	<b>Testen</b>	<b>24</b>
5.1	Uit te voeren tests	24
5.1.1	White-box test: unit tests	24
5.1.2	Black-box test: functionaliteit	24
5.1.3	Black-box test: integratie	25
5.1.4	Kwaliteitstests	25
5.1.5	White-box test: load testing	26
5.1.6	Black-box test: uiteindelijke acceptatie test	26
5.2	Testproces	27
5.3	Testplanning	27
5.4	Opbouw Game Management Systeem	28
5.4.1	Gewenste opbouw Game Management Systeem	28
5.4.2	Onderzoek werking Casinobackend	29
5.5	Projectplanning	31



5.5.1	Pilot prioriteiten .....	31
5.5.2	Planning.....	32
<b>6</b>	<b>1<sup>e</sup> Pilot: proof of concept .....</b>	<b>33</b>
6.1	<i>Definitiestudie 1<sup>e</sup> pilot .....</i>	33
6.1.1	Systeemeisen 1 <sup>e</sup> pilot .....	33
6.1.2	Randvoorwaarden .....	34
6.1.3	Beperkingen .....	34
6.1.4	Systeemconcept .....	34
6.1.5	Pilotplan .....	34
6.2	<i>Plan van Aanpak 1<sup>e</sup> pilot.....</i>	35
6.2.1	Ontwerp backendadapter .....	35
6.2.2	Gebruik van Seam en Richfaces .....	37
6.2.3	Web User Interface mockup .....	38
6.3	<i>Implementeren ontwerp 1<sup>e</sup> pilot.....</i>	40
6.4	<i>Pilotdocument 1<sup>e</sup> pilot .....</i>	42
6.4.1	Backendadapter implementaties .....	42
6.4.2	Beveiliging .....	42
6.4.3	Spelerweergave.....	42
6.4.4	Datamodel.....	43
6.4.5	Problemen tussen datamodel en Seam .....	45
6.5	<i>Conclusie 1<sup>e</sup> pilot.....</i>	45
<b>7</b>	<b>2<sup>e</sup> Pilot: Spelers, affiliates &amp; managers.....</b>	<b>46</b>
7.1	<i>Definitiestudie 2<sup>e</sup> pilot .....</i>	46
7.1.1	Doelen .....	46
7.1.2	Systeemeisen van de pilot.....	47
7.1.3	Randvoorwaarden .....	47
7.1.4	Testrapport.....	48
7.1.5	Pilotplan .....	48
7.2	<i>Plan van Aanpak 2<sup>e</sup> pilot.....</i>	49
7.2.1	Bevestiging reikwijdte .....	49
7.2.2	Op te leveren onderdelen .....	49
7.2.3	Kwaliteitszorg .....	49
7.2.4	Tijdschatting .....	49
7.2.5	Faseplanning .....	50
7.3	<i>Opzetten testomgeving 2<sup>e</sup> pilot .....</i>	51
7.3.1	Opstellen validatietests .....	51
7.3.2	Testopzet unit tests.....	52
7.3.3	Zelf-controlerende tests.....	54
7.4	<i>Pilotdocument 2<sup>e</sup> pilot .....</i>	55
7.4.1	Implementatie GMS backendadapters .....	55
7.4.2	Implementatie gebruikersomgeving .....	58





7.5	<i>Testrapport 2<sup>e</sup> pilot</i> .....	64
7.5.1	Resultaten validatietests .....	64
7.5.2	Resultaten unit tests Casino backend .....	65
7.5.3	Resultaten unit tests Poker backend.....	66
<b>8</b>	<b>3<sup>e</sup> Pilot: Uitbreiden spelerdetails</b> .....	<b>67</b>
8.1	<i>Definitiestudie en Plan van Aanpak 3e pilot</i> .....	67
8.1.1	Doelen .....	68
8.1.2	Systeemeisen van de pilot.....	68
8.1.3	Tijdschatting .....	69
8.1.4	Faseplanning .....	69
8.2	<i>Implementeren ontwerp 3e pilot</i> .....	70
8.2.1	Implementatie GMS backendadapters .....	70
8.2.2	Aanpassing van winnaars en verliezers .....	71
8.2.3	Implementatie gebruikersomgeving .....	72
8.2.4	Manager rollen.....	74
8.3	<i>Testrapport 3e pilot</i> .....	75
8.3.1	Resultaten validatietests .....	75
8.3.2	Resultaten unit tests Casino backend .....	76
8.3.3	Resultaten unit tests Poker backend.....	76
<b>9</b>	<b>Afronding project</b> .....	<b>77</b>
<b>10</b>	<b>Evaluatie</b> .....	<b>78</b>
10.1	<i>Procesevaluatie</i> .....	78
10.2	<i>Productevaluatie</i> .....	79
10.3	<i>Evaluatie opgeleverde software</i> .....	80
10.3.1	Backendadapters .....	80
10.3.2	Gebruikersinterface .....	80
10.4	<i>Behalen van de competenties</i> .....	81
10.5	<i>Advies voor vervolg van het project</i> .....	82
10.6	<i>Persoonlijke evaluatie</i> .....	82
<b>11</b>	<b>Verklarende woordenlijst</b> .....	<b>83</b>
<b>12</b>	<b>Boeken en rapporten</b> .....	<b>83</b>
<b>13</b>	<b>Index</b> .....	<b>84</b>



# 1 Inleiding

In verband met de groei van het aantal online casino's bij Gaming & Gambling heeft bedrijf MC Advies & Management de opdracht gekregen om een beheersysteem te ontwikkelen waaruit de verschillende softwarepakketten kunnen worden onderhouden. Een manager moet kunnen inloggen bij deze webomgeving, waarna deze automatisch wordt ingelogd bij de beschikbare pakketten.

Dit afstudeerverslag beschrijft het proces van de ontwikkeling van deze Game Management Service. Het doel is om een duidelijk inzicht te geven in de genomen stappen en overwogen keuzes. Dit document is geschreven om de examinatoren te overtuigen dat de aanpak van dit project HBO waardig is en tot een goede beoordeling van de afstudeerstage te komen.

Dit verslag is opgedeeld in een aantal hoofdstukken; hoofdstuk 2 geeft achterliggende informatie over het bedrijf waar de stage wordt gelopen en hoe dit in betrekking staat tot het bedrijf waar de software voor wordt ontwikkeld. De omschrijving van de afstudeeropdracht is te vinden in hoofdstuk 3, waarna hoofdstuk 4 het tot stand komen van het plan van aanpak van dit project beschrijft. Hoofdstuk 5 beschrijft wat en hoe de iteraties (pilots) getest zullen worden.

De hoofdstukken 6, 7 en 8 beschrijven de ondernomen stappen en opgeleverde documenten die tijdens de pilots zijn ontwikkeld. Hoofdstuk 9 beschrijft de afronding van het project, hoofdstuk 10 is een evaluatie van het project en bevat het eindadvies wat betreft de opgeleverde software.



## 2 Over het bedrijf

De afstudeerstage zal worden gelopen bij het bedrijf MC Advies & Management Ltd. (vanaf hier ook wel afgekort tot MC). Dit bedrijf is opgericht in 2005. MC doet management, adviseert en ontwikkelt, voornamelijk online casino- en betaalsoftware. Het bedrijf bestaat uit een manager (dhr M. de la Rie) en vier vaste medewerkers. Daarnaast wordt er gebruik gemaakt van verschillende afstudeerders en stagiairs. Dit levert een dagelijkse bezetting op van zeven tot acht personen.

### 2.1 Positie van het afstudeerproject binnen het bedrijf

Het afstudeerproject zal gemaakt worden in opdracht van het bedrijf *“Gaming & Gambling”*, onderdeel van Mejdí NV, gevestigd op Curaçao. Een gedeelte van de softwareontwikkeling is uitbesteed aan MC. De ontwikkeling van de online casino software is gestart in 2002 en sinds anderhalf jaar zijn diverse casino gerelateerde sites online te bezoeken. Er zijn inmiddels een aantal casino's opgezet, namelijk casino-nederland.com, casino-belgie.com, casino-deutschland.com, netcashcasino.com en sphinxcasino.com. Naar schatting maken per week duizend spelers actief gebruik van de verschillende casinogames.

In 2005 heeft dhr. de la Rie het bedrijf Payment House BV opgericht. Dit bedrijf ontwikkelt online betaalsystemen. De ontwikkeling hiervan verloopt in samenwerking met MC. Bekende online merknamen hiervan zijn MasterPay (een Payment Service Provider) en Tikkies (een e-wallet). Beide betaaloplossingen zijn geïmplementeerd in de casino's van Gaming & Gambling.

In 2007 is in opdracht van MC Advies pokersoftware gemaakt door een extern softwarebedrijf in Nepal. De ontwikkeling hiervan is echter voortijdig afgebroken; deze software is onvolledig opgeleverd, waardoor deze nog steeds in bètastadium is.

### 2.2 Projectomgeving

Voor het afstudeerproject is een bedrijfsmentor aangesteld, dhr. B. Schlärman. Deze is werkzaam bij MC als (hoofd) softwareontwikkelaar en softwarebeheerder. Terwijl de la Rie alle financiële en bedrijfskundige taken heeft, neemt Schlärman de softwarematige en technische verantwoordelijkheden op zich. Tijdens de afstudeerperiode er nauw samengewerkt worden met de bedrijfsmentor. Hij zal het project op de voet volgen en waar nodig aanwijzingen of tips geven.

Bij MC wordt een eigen ontwikkelmethode gebruikt. Na het opstellen van de systeemeisen wordt er een domeinmodel gemaakt, waarna de code geïmplementeerd wordt. Ter afsluiting wordt er een checklist met eisen doorlopen om de functionaliteit te testen. Gebruik van watervalmethode en iteratief ontwikkelen worden naar eigen keuze toegepast, evenals unittesten en loadtesten.



## 3 Afstudeeropdracht

### 3.1 Casino- en pokersoftware

Het bedrijf Gaming & Gambling biedt online casino- en pokersoftware aan via verschillende websites. De casinospellen bestaan uit een groot aantal Adobe Flash games, waarbij bezoekers kunnen kiezen om gratis of voor echt geld te spelen. In alle aangeboden games speelt de bezoeker tegen de computer. Deze website is in gebruik genomen in augustus 2008.



Figuur 3-1: Webpagina Netcash Casino



Figuur 3-2: Desktop applicatie van 123PokerPlayer

bedrijfszeker is, is momenteel alleen via een bètaserver te spelen. In de toekomst zal deze software echter wel in gebruik worden genomen.

### 3.2 Overeenkomsten

De twee softwarepakketten hebben een veel overeenkomsten. Zo heeft bijvoorbeeld iedere casino- of pokerserver een groep met spelers, die onder te verdelen is in een groep actieve (betalende) spelers, bezoekers die gratis spelen, een groep winnende spelers en een groep verliezende spelers. Daarnaast hebben beide pakketten affiliates. Dit zijn bedrijven of personen die reclame maken voor het casino op hun eigen website. Voor iedere betalende aangemelde speler krijgen deze partners een vergoeding, bijvoorbeeld een bepaald percentage van de winst die gemaakt is op deze persoon.

Een andere overeenkomst is de toepassing van bonussen. Zo kan er bijvoorbeeld een bonus worden ingesteld voor een bepaalde (feest)dag, waarmee spelers bijvoorbeeld een bepaald percentage gratis speelgeld krijgen bij een nieuwe storting. In zekere mate zijn de pakketten dus gelijk aan elkaar, wat betekent dat ze beiden vanuit een centraal onderhoudssysteem zouden kunnen worden beheerd.





### 3.3 Beheer

Het casino- en het pokerpakket hebben ieder een eigen management systeem. Met een management systeem kan de beheerder controleren of de spellen nog naar behoren functioneren, spellen herstarten, bonusprogramma's aanmaken en beheren, affiliates beheren, winnende of verliezende spelers inzien en valsspelers detecteren. De belangrijkste taak van een beheerder is het beantwoorden van vragen en tevreden houden van de klanten. Ontevreden klanten zullen immers geen speelgeld storten. Deze taak omvat onder andere het behandelen van veelgestelde vragen (Frequently Asked Questions), hulp bij aanmelden en het afhandelen van foutmeldingen. Een beheerder kan meerdere casino's in beheer hebben.

### 3.4 Ontstaan van de opdracht

Op dit moment zijn er 4 casinosites online te bezoeken. Bij MC is besloten om dit aantal gamedomeinen uit te breiden over de wereld, er zullen online casino's worden geopend in nieuwe landen. Voor ieder domein (land) is een eigen managementsysteem beschikbaar. Er zijn echter wat problemen bij het uitbreiden van het aantal domeinen;

- De domeinen zullen worden beheerd door meerdere werknemers, die per domein moeten inloggen. Met de groei van de domeinen zal ook het aantal spelers groeien. Een grotere spelersgroep betekent dat de controle op valsspelen ook tijdrovender zal zijn; er zijn meer spelers om te controleren, verdeeld over die verschillende gameservers. Stel dat er om het half uur een controle zal plaatsvinden voor iedere server, betekent dit dat een medewerker zo'n 40 keer per uur moet inloggen (uitgaande van twintig domeinen). Dit is erg tijdrovend en vooral vervelend werk. Om het herhaaldelijke inloggen te beperken wil MC graag één centrale inlogpagina, vanwaar alle domeinen beheerd kunnen worden. Een opsomming van de cashout requests en een overzicht van alle winnende (vals)spelers zal op deze pagina's te zien moeten zijn.
- Het managementsysteem voor de casinosoftware is opgezet met verouderde ontwikkeltechnieken, waardoor uitbreiding veel tijd en uitzoekwerk kost. De ontwikkelaars van deze casinosoftware zijn niet meer in dienst bij MC. Er zal een oplossing gevonden moeten worden om deze beter onderhoudbaar te maken.
- Het managementsysteem voor de pokersoftware is niet volledig. Belangrijke functies om valsspelen te voorkomen zijn niet aanwezig.

Het doel van de afstudeeropdracht is het ontwikkelen van een overkoepelende Game Management Service (GMS). Het plan is dat werknemers kunnen inloggen bij het GMS, waarna ze het beheer kunnen doen van zowel de casino als de pokersoftware. Een webinterface (Game Management Service User Interface, afgekort tot GMS UI of webUI) zal de taken van de onderliggende managementsystemen vervangen.



### 3.5 Mogelijke knelpunten

Integratie tussen beide softwareonderdelen zal lastig zijn aangezien ze verschillen van programmeertaal. De casinosoftware is gemaakt in Java, gebruikmakend van een Sybase Database; de pokersoftware is geschreven in C#, gebruikmakend van een Microsoft SQL Server voor de opslag van de gegevens. Het GMS zal waarschijnlijk worden ontwikkeld in Java, waardoor er dus een oplossing gevonden moeten worden om de pokersoftware te beheren. Het is niet mogelijk om vanuit Java direct objecten op te vragen uit een C# programma. Mogelijk kan dit worden opgelost met een RMI of COM verbinding. Daarnaast moet er worden gezorgd dat bij de implementatie van de poker en casinosoftware de uitbreiding met andere games en backends niet wordt uitgesloten.



## 4 Tot stand komen van het Plan van Aanpak

De eerste weken zijn besteed aan het uitwerken van de systeemeisen en het maken van een Plan van Aanpak. In dit plan is onder andere beschreven welke methodieken en technieken gebruikt gaan worden. MC heeft gekozen om het GMS te laten bouwen in Java, zodat ontwikkeling en onderhoud gedaan kan worden door Linux en Windows gebruikende medewerkers in het bedrijf. Java is immers platform onafhankelijk, wat betekent dat het onder verschillende besturingssystemen kan worden gedraaid. Daarnaast is de ontwikkelsoftware voor Java gratis beschikbaar en worden alle andere projecten ook in Java ontwikkeld. De kennis van deze taal is dus bij het bedrijf aanwezig, zodat overname en onderhoud van de software zonder problemen kan verlopen.

### 4.1 Kiezen van ontwikkelmethode

Voor de ontwikkeling is keuze uit een waterval methode, of een variant van een iteratieve ontwikkelmethode. Bij de watervalmethode worden eerst de systeemeisen gespecificeerd, waarna de code wordt geïmplementeerd en opgeleverd. De perfecte indeling van het GMS (vooral de UI) zal moeten worden gemaakt aan de hand van input van de toekomstige gebruikers. Om deze zo goed mogelijk af te stemmen op de wensen, wordt er gekozen om op een iteratieve manier te ontwikkelen. Na iedere iteratie zal de opdrachtgever kunnen beoordelen of het gemaakte onderdeel is geworden zoals verwacht. Bij deze beoordeling kunnen ook nieuwe systeemeisen worden toegevoegd of gewijzigd. Er zijn vele iteratieve ontwikkelmethoden. In voorgaande projecten heeft de auteur gebruik gemaakt van RUP, DSDM, Scrum (Agile), Extreme Programming.

### 4.2 Keuze voor Iterative Application Development

Voor dit project leek het interessant om een nieuwe methode uit te proberen. Er is gekozen om gebruik te maken van de ontwikkeltechniek IAD, Iterative Application Development. In deze methode wordt het project opgedeeld in pilots, de iteratieve stappen. Om het ontwikkelen systematisch te laten verlopen worden per pilot een aantal documenten opgeleverd;

Naam	Omschrijving
<b>Definitiestudie</b>	Beschrijft de eisen van de pilot(s)
<b>Plan van Aanpak</b>	Beschrijft de aanpak per pilot
<b>Pilotdocument</b>	Beschrijft de ontwikkelde pilot; ontwerp en werking
<b>Testplan</b>	Beschrijft wat en hoe er getest gaat worden
<b>Testrapport</b>	Beschrijft de tests en testresultaten

Allereerst wordt er een definitiestudie gedaan naar de aanpak van de pilot en de plaats daarvan in het gehele project. Om te beginnen is er gekeken welke IAD methode geschikt zal zijn voor dit project. Bij IAD is er keuze uit een aantal vormen. Bedenk daarbij dat IAD volgens het volgende principe pilots afhandelt:



Naam	Kenmerken
<b>Evolutionair ontwikkelen</b>	Alle drie de fasen worden doorlopen tijdens iedere iteratie van de ontwikkelcyclus
<b>Incrementeel opleveren</b>	Deze variant schrijft voor dat alle systeemeisen eerst volledig worden gespecificeerd. Daarna wordt het systeem in stappen, volgens de iteratieve benadering ontwikkeld en ingevoerd in de organisatie.
<b>Big-bang invoeren</b>	In deze variant komen zowel systeemeisen en het systeemconcept alsook het informatiesysteem op een iteratieve wijze tot stand. Met invoering wordt echter gewacht totdat de laatste pilot is ontwikkeld.
<b>Incrementeel ontwikkelen</b>	Ook wel RAD (Rapid Application Development). De toepassing van incrementeel ontwikkelen gaat er van uit dat de systeemeisen en het systeemconcept van tevoren volledig zijn gespecificeerd. Vervolgens komt het systeem tot stand in een aantal iteraties.

**Tabel 4-1: Verschillende werkvormen van iteratief ontwikkelen, vrij naar 'IAD, Het evolutionair ontwikkelen van informatiesystemen' (Tolido, 1995, pp. 12-13)**

Na een iteratie kan het GMS al functionaliteiten bieden die momenteel niet in de huidige software worden ondersteund. Door gebruik te maken van incrementeel opleveren kunnen deze nieuwe mogelijkheden direct gebruikt worden door de backendbeheerders.



### 4.2.1 Gebruik van pilots

Een project bestaat bij IAD uit een verzameling pilots; een projectperiode die eindigt met een mijlpaal. Voor aanvang van deze pilot wordt er een definitiestudie gedaan, waarin wordt besloten wat er in één of meerder pilots zal worden gedaan. Na deze definitiestudie zal er per pilot een Plan van Aanpak worden gemaakt. Indien zinvol sluit een pilot af met één of meerdere tests om de kwaliteit te garanderen.

## 4.3 Opstellen van systeemeisen

Een (software)project wordt altijd gestart met een bepaald doel. Voorafgaand aan een project worden dit doel samen met de systeemeisen gespecificeerd, zodat de softwareontwikkelaar weet wat er gerealiseerd moet worden en de opdrachtgever weet wat er aan het einde van het project opgeleverd zal worden. De opdrachtgever heeft echter niet altijd een concreet plan over de eisen van het eindproduct. De softwareontwikkelaar moet in dit geval in samenwerking met de opdrachtgever specificeren wat deze systeemeisen zijn.

De gebruikersomgeving van de Game Management Service moet dezelfde mogelijkheden bieden als het casino en poker managementsysteem. Deze webpagina's bevatten alle belangrijke functionaliteiten om de backends te beheren. De gebruikersomgeving voor het GMS moet een aggregatie van deze twee managementsystemen worden, zodat er geen functionaliteit verloren gaat.

Veel functionaliteiten in de twee managementsystemen komen overeen. Alle mogelijkheden moeten via de GMS gebruikersomgeving te benaderen zijn. Naast deze eisen is het bekend dat de huidige gebruikers ideeën hebben over verbeteringen en nieuwe functionaliteiten. Deze zullen ook meegenomen moeten worden in het nieuwe ontwerp.

### 4.3.1 Mogelijkheden van de huidige systemen

Op de volgende pagina's is een overzicht van de menu's uit het casino managementsysteem en het poker managementsysteem te vinden. Hoewel de omschrijvingen niet altijd overeenkomen zijn de volgende belangrijke onderdelen in beide systemen aanwezig:

- Een systeem bevat spelers
- Een systeem bevat spellen
- Een systeem bevat affiliates
- Een systeem bevat geldtransacties, zowel stortingen als aanvragen voor uitbetalingen
- Verschillende manieren om de klanten te bereiken of te woord te staan
- Mogelijkheid om bonusprogramma's aan te maken
- Verschillende manieren om valsspelen te detecteren of tegen te gaan
- Mogelijkheid om verschillende soorten rapporten (reports) te genereren.



### 4.3.1.1 Functionaliteiten casino managementsysteem

De gebruikersomgeving van het casino bevat een groot aantal mogelijkheden. Dit zijn de onderdelen die vanuit de menu's in de beheeromgeving beschikbaar zijn;

#### *Settings*

- End game
- Add manager
- Reporting period
- Manager information
- Game options
  - View Game history
  - Set game parameters
- Jackpot control center
  - Jackpot games summary
  - Activity simulator settings

#### *Transactions*

- Transaction rates
  - Set payment rates
  - Set bonus rates
- View transactions
  - View suspended transactions
  - View completed transactions
- Fax confirmation
- Set fax confirmation limit

#### *Users*

- Winning users
- View visitor stats
- View suspicious accounts
- Add users
- Losing users
- Users information
- Online logs
- Visit logs

#### *Affiliates*

- View payments
- Affiliate information
- Affiliate summary
- Commision setting
  - Default values
  - For current affiliates

#### *Cashier*

- Deposits
- Cashout
- Balance adjustment
- Bonuses

#### *Financial report*

- General report
- Report on casino
- Report on users
- Report on games
- Report on affiliates

#### *Newsletter*

- Create mailing list
- Message templates
  - View message templates
  - Create message template
- Set default messages
  - Signup message
  - Deposit message
- Email preferences

#### *Pit boss*

- Restrictions
  - View country restriction
  - Add country restriction
  - View credit card restriction
  - Add credit card restriction
  - View player restriction
  - Add player restriction
  - View domain restriction
  - Add domain restriction
- Fraud control
  - Black list
    - Black list summary
    - Black list settings
  - Suspicious values
  - View suspicious players
  - Fraud transactions

#### *Marketing*

- Create campaign
- View campaign info
- Summary campaign info
- List bonus codes
- Create bonus code



### 4.3.1.2 Functionaliteiten poker managementsysteem

De gebruikersomgeving van de pokersoftware lijkt minder mogelijkheden te bevatten. Dit zijn de onderdelen die vanuit de menu's in de beheeromgeving beschikbaar zijn;

#### *User Management*

- Register Tournament
- Unique Avatar
- UnRegister Tournament
- View Players

#### *Admin Management*

- Account Override
- Deposit Point Rule
- Email Marketing
- Email Templates
- Foul Language Filter
- Generate Random Number
- Manage Groups
- Manage Operator
- Manage Role
- Real Hand Point Rule
- Scrolling Message
- Send Message
- View Game Servers

#### *Account Management*

- Transfer Money

#### *Games Management*

- Add Tournaments
- Assign Table Logo
- Create Table
- Detect Collusions
- Game Icons

#### *Reports*

- Affiliate Signups
- Bonus Report
- Deposits
- Payouts
- Players Point
- Players Signups
- Rake Report
- Reconciliations
- Tournaments Report
- Users Online



### 4.3.2 Systeemeisen aan de hand van vergadering eindgebruikers

Voor een aanvulling op de systeemeisen is er een vergadering geweest met twee eindgebruikers. Momenteel wordt alleen het Casino Management Systeem gebruikt, waarin de eindgebruikers graag nog enkele verbeteringen in zouden zien. De functionaliteiten uit de casino gebruikersinterface zullen dus samen met deze verbeteringen beschikbaar moeten zijn in de gebruikersinterface van de Game Management Service.

De vergadering is gehouden om de nieuwe ideeën te bespreken en de voortgang en toekomstvisie van het casino. Niet alle informatie was relevant voor het GMS project, maar de volgende ideeën over het GMS zijn tijdens de vergadering geopperd:

- **Geavanceerde bonus berekening;** het is nu wel mogelijk om bonussen toe te passen, maar bij een aanvraag tot uitbetaling is het niet direct mogelijk om de toegepaste bonussen te controleren. Er moet dus bij iedere uitbetaling gekeken worden van welke bonusprogramma's gebruik is gemaakt tijdens de speelperiode, waarna kan worden gecontroleerd of het aangevraagde bedrag rechtmatig is. Een overzicht van de toegepaste bonussen per aanvraag of per speler zou tijdrovend werk schelen.
- **Ticketing support;** een onderdeel waardoor de e-mailconversaties tussen gebruikers en beheerders vervangen zal worden door een intern berichten systeem, waaraan de meldingen "in behandeling", "afgehandeld" etc. kunnen worden gegeven. Dit is nodig omdat het aantal beheerders ook zal meegroeien met het aantal domeinen. Op deze manier kunnen de tickets worden toegewezen en is het voor een klant duidelijk of hun ticket in behandeling is.
- **Webapplicatie geschikt voor weergave op mobiele platforms;** in het speciaal de Google G1 Android phone. Deze telefoon heeft een resolutie van 320 x 480 pixels. De huidige casino backoffice is lastig te bedienen via deze telefoon, aangezien de menustructuur bestaat uit vele mouse-overs. Hier zal rekening mee moeten worden gehouden bij het ontwerpen van de GUI. Als de schaalbare (mobiele) resolutie de mogelijkheden van de webbrowser-versie teveel aantast, zal in een volgend project een gebruikersomgeving moeten worden gemaakt voor mobiele telefoons. De onderliggende software kan hierbij ondersteuning bieden, door bijvoorbeeld gebruik te maken van een Model View Controller model. In dit geval hoeft er dus enkel een nieuwe View gemaakt te worden.





### 4.3.3 Functionele eisen voor de Game Management Service

In samenwerking met de opdrachtgever zijn de volgende eisen opgesteld;

#### 4.3.3.1 Binnen de scope

- Een Game Management Service
  - Heeft een backend verzameling
  - Krijgt later functionaliteiten zoals beschreven in hoofdstuk 4.3.3.2
- Backendadapters voor Casino en Poker software
  - Zijn een universele wrapper tussen de Game Management Service en de achterliggende gok-software
- Uit beide backendadapters kan de volgende data worden opgehaald:
  - Lijst met spelers met kolommen NAW, balans, inloggegevens
  - Lijst met winnende spelers binnen een bepaald datumbereik
  - Lijst met verliezende spelers binnen een bepaald datumbereik
  - Lijst met spellen
  - Lijst met affiliates
  - Lijst met bonusprogramma's
  - Lijst met beheerders en hun rollen
- In de backendadapters kunnen de volgende acties worden uitgevoerd:
  - Speler verwijderen
  - Speler toevoegen
  - Speler bewerken
  - Speler zoeken
    - Zoekvak email en inlognaam
    - Uitgebreide zoekfilter, op te slaan voor hergebruik
  - Blokkeren van een speler
  - Spel herstarten
  - Spel uitschakelen
  - Affiliate toevoegen
  - Affiliate bewerken
  - Affiliate verwijderen
  - Bonusprogramma toevoegen
    - Bonusregel toevoegen
  - Beheerder toevoegen
  - Beheerder rol toewijzen
  - Beheerder verwijderen
- Een webinterface voor de Game Management Service
  - Aangeboden functionaliteiten afhankelijk van de rol van de ingelogde beheerder
  - De backenddata kan hierin getoond worden
  - De backendacties zijn hierin aan te roepen



#### 4.3.3.2 Buiten de scope

De volgende systeemeisen behoren wel bij het eisenpakket van de Game Management Service, maar zullen niet tijdens dit project worden geïmplementeerd. Tijdens het project dient echter wel rekening gehouden te worden met deze toekomstige functionaliteiten. Aan deze eisen dient voldaan te worden in een opvolgend project.

- Implementatie van een betaalsysteem in Game Management Service, welke kan worden aangeroepen vanuit de backends.
- Implementatie van de geavanceerde bonus berekeningen (zie hoofdstuk 0)
- Een ticket / issue systeem in de Game Management Service voor helpdesk doeleinden, waarmee:
  - Berichten kunnen worden geschreven en ontvangen van en naar gebruikers en andere medewerkers
  - Probleemmeldingen (uit berichten) kunnen worden toegewezen aan medewerkers
  - Medewerkers een lijst met taken kunnen opvragen
  - Medewerkers de status van een taak kunnen aanpassen in “afgehandeld”, “in behandeling”, “afgewezen” e.d.
- Gebruikersinterface voor gebruik in webbrowsers op mobiele telefoons, getoond in een resolutie van 320 x 480 pixels (Google G1 Android Phone)

#### 4.3.4 Niet functionele eisen voor de Game Management Service

- Een document met de systeemeisen
- Systeemontwerp volgens de methode Usage Centered Design:
  - Use cases
  - Task cases
  - Task map
  - Abstracte prototypes
  - Mockups gebruikersinterface
- Klassendiagrammen van Game Management Systeem en backendadapter
- Software wordt ontwikkeld in het Seam framework, draaiend op een Glassfish server.
- Een testplan voor het testen van de geproduceerde broncode
- Testrapport(en) van de ontwikkelde software
- Overdrachtsdocument waarin de ontwikkelde softwareonderdelen worden beschreven



### 4.3.5 Afbakening

Bij het opstellen van het Plan van Aanpak bleek al snel dat er erg veel eisen waren voor het Game Management Systeem; er moest veel ontworpen en ontwikkeld worden. Het is dus maar bezien of alle functionaliteiten kunnen worden geïmplementeerd tijdens de stage. In de afbakening is dan ook opgenomen dat niet alle functionaliteiten binnen de scope van het project vallen. Er zal getracht worden zoveel mogelijk te ontwerpen en te implementeren.

Het systeemontwerp en de testdocumentatie dienen zo te worden opgesteld dat een volgende medewerker een vervolgproject kan starten.

### 4.3.6 Gebruik van Usage Centered Design

Voor het ontwerpen van het Game Management Systeem is inzicht nodig in de behoeften van gebruikers en hoe deze te werk gaan. Veelal wordt er bij softwareontwerpen gebruik gemaakt van de methode *user-centered design*, waarbij de eisen van de gebruikers worden omgezet in use cases. Deze use cases worden dan gebruikt om een visueel ontwerp (mockups) te maken van het eindproduct.

Een groot nadeel van gebruik van user-centered design is dat de normale eisen en workflows worden overschreeuwd door nieuwe wensen van de gebruikers. Zo krijgt de softwareontwikkelaar een vervormd beeld van wat het systeem werkelijk moet kunnen, omdat de focus dan ligt op het tevreden stellen van de gebruikers en implementeren van de nieuwe wensen. Vaak moet de software later meerdere malen aangepast worden omdat de gebruikers functionaliteiten missen, waarvan ze hadden verwacht dat die wel zouden worden geïmplementeerd. De ontwerpers op hun beurt komen er in deze latere stadia achter dat hun ontwerp niet meer is aan te passen aan deze (voor gebruikers) normale functionaliteiten. Vaak worden de functionaliteiten toegevoegd, zonder dat over het ontwerp is nagedacht, of het is simpelweg niet mogelijk om de functies toe te voegen. Dit resulteert in onvolledige software, of software met lagere kwaliteit (snelheid, robuustheid, onderhoudbaarheid).

Er is gekozen om voor dit project gebruik te maken van de ontwerpmethode *Usage Centered Design*. Bij deze methode staat niet zozeer de gebruiker, maar de workflows centraal. Het Game Management Systeem zal gebruikt worden door verschillende soorten medewerkers, ieder met zijn of haar eigen verantwoordelijkheden en taken. Een medewerker kan meerdere rollen hebben in een systeem, zoals bijvoorbeeld een financiële en marketing gerelateerde rol. Een medewerker heeft dus één of meerdere rollen, een rol heeft één of meerdere taken, een taak heeft één of meerdere workflows om te taak te volbrengen.

Met behulp van de Usage Centered Design methode wordt per rol beschreven wat de omschrijving, context, kenmerken en criteria zijn. Nadat van alle rollen de use cases beschreven zijn, worden de task cases gemaakt. Deze task cases (gebaseerd op de use cases) beschrijven de interactie tussen de gebruiker en het systeem voor een bepaalde taak. Met de task cases kan een task map worden gemaakt, waar uiteindelijk de gebruikersinterface op kan worden gebaseerd (zie hoofdstuk 6.2.3).

*Zie de bijlage voor het volledige rapport over het systeemontwerp*

## 5 Testen

Als testmethode is er gekozen voor TMap (Test Management Approach), ontwikkeld door het Nederlandse ICT-bedrijf Sogeti. De auteur heeft deze methode bij voorgaande projecten gebruikt en deze is goed bevallen voor gebruik bij iteratief ontwikkelen.

TMap bestaat uit vier onderdelen:

- Fasering van testactiviteiten (wat er getest gaat worden en wanneer)
- Technieken bruikbaar voor testactiviteiten (hoe er getest gaat worden)
- Infrastructuur (waarmee er getest gaat worden)
- Organisatie (wie er wat gaan testen)

TMap past prima in de IAD methode; er kan getest worden na voltooiing van een pilot en ter afsluiting van het project. Bij TMap wordt gesproken van white-box tests en black-box tests. Een white-box test wordt uitsluitend uitgevoerd door de ontwikkelaar. De tests zullen worden afgestemd op de broncode van de ontwikkelde software. Bij een black-box test wordt vooral getest of een functionaliteit naar behoren werkt. Deze test kan door iedereen worden uitgevoerd.

### 5.1 Uit te voeren tests

Om de opdrachtgever te overtuigen dat de software naar behoren werkt, wordt het systeem getest op integratie en functionaliteit. De kans is groot dat de Game Management Service later zal worden uitgebreid. Om de werking van de andere onderdelen te garanderen worden er unit tests gemaakt, waarmee gecontroleerd kan worden of een aanpassing van invloed is op de functionaliteit van andere onderdelen. Verder is er gekozen voor de kwaliteitstests uitbreidbaarheid, validiteit, gebruiksvriendelijkheid, omdat deze van belang zijn bij latere uitbreidingen en om de juiste toepassing van het usage centered ontwerp van de webinterface te bewijzen. Aangezien het GMS met meerdere backends gaat werken waarin duizenden personen (spelers, affiliates) voorkomen, wordt er ook een load-test gedaan om de capaciteit te meten.

#### 5.1.1 White-box test: unit tests

Bij een unit test worden de verschillende stukken broncode (units) getest op de correcte werking. Het doel is om deze units zo onafhankelijk mogelijk van elkaar te testen, gebruikmakend van gegenereerde data en / of objecten. Deze test zal na iedere pilot worden gedaan, om te bewijzen dat de opgeleverde code naar behoren werkt. Dit is een white-box test die door de programmeur wordt uitgevoerd; inhoudelijke kennis van de broncode wordt gebruikt om de tests op te stellen.

#### 5.1.2 Black-box test: functionaliteit

Het doel van deze test is het aantonen van de functionele eisen, zoals opgesteld in het plan van aanpak van iedere pilot. In dit stadium worden ook de validatietests gedaan, waar de validiteit van de invoervelden in de webinterface wordt onderzocht. Zie hiervoor hoofdstuk 5.1.4.2.



### 5.1.3 Black-box test: integratie

Er wordt low-level getest op functionaliteit. Hiervoor worden de dataflows doorlopen zoals beschreven in het systeemontwerp. Deze dataflows beschrijven de verschillende rollen en hun respectievelijke taken.

### 5.1.4 Kwaliteitstests

De Game Management Service is de laag tussen de gebruikersinterface (webui) en de verzameling casino- en pokerservers. Op deze servers draait de software en is de data opgeslagen van de games, spelers enz. Per manager kunnen backends worden toegewezen die deze kan beheren.

#### 5.1.4.1 White-box test: uitbreidbaarheid

Deze test zal worden uitgevoerd tegen het einde van het project. Met behulp van de documentatie zal een andere programmeur moeten proberen het systeem uit te breiden. Deze zal moeten proberen de software uit te breiden. Denk hierbij aan het aanmaken en toevoegen van een nieuw backend, of het toevoegen van een nieuwe webpagina. Er zal documentatie worden opgeleverd die beschrijft wat voor stappen er moeten worden doorlopen.

De meetresultaten uit deze test zijn niet kwantitatief, er kan geen percentage “uitbreidbaarheid” berekend worden. In plaats daarvan kan de tweede programmeur aangeven welke passages uit het document moeten worden verbeterd of uitgebreid. Op deze manier zou een derde programmeur genoeg informatie hebben om de software uit te breiden in een later stadium.

#### 5.1.4.2 Black-box test: validiteit

De Game Management Service zal naar alle waarschijnlijkheid veel invoervelden bevatten, zoals gebruikt bij het bewerken van de personalia van een persoon of het afhandelen van een probleemmelding. Met de validiteitstest kan gecontroleerd worden of de invoer in deze velden juist wordt verwerkt en weer op de juiste manier wordt getoond bij opnieuw bewerken van het invoerveld. Een voorbeeld hiervan is het opslaan van een datum. De invoervelden zijn *dag*(numeriek), *maand*(tekst selectiebox) en *jaar*(numeriek). De opslag van deze waarden in de database is een varchar(8), namelijk {jaar}{maand}{dag}. De backends voor poker en casino slaan hun gegevens op in een externe database. De databases (Microsoft SQL Server en Sybase) zijn direct gekoppeld aan de backends, of via methoden in de bronsoftware (software specifieke backoffices / adminpanels). Met de validiteitstest kan dan worden gecontroleerd of bij opnieuw bewerken van het jaar de waarden goed geparsed zijn.



#### **5.1.4.3 Black-box test: gebruiksvriendelijkheid**

In het systeemontwerp wordt beschreven welke soorten managers er zijn (financieel, games, marketing). Met behulp van de uit te voeren taken van deze personen moet worden gecontroleerd of de dataflows van de GMS webinterface gebruiksvriendelijk zijn. Eventueel kan een testpersoon de taken proberen te vervullen. Het doel is dat de gebruiker zijn taken (snel) kan volbrengen zonder hulp van de ontwerper. Denk hierbij aan het vinden van gerelateerde taken in de sidebar. Het streven is om dit met zo min mogelijk muisclicks hoeven doen.

#### **5.1.5 White-box test: load testing**

Het GMS wordt gebouwd om te voldoen aan de groter wordende hoeveelheid gameservers. Er zullen in een groot aantal landen online casino's geopend worden, die allemaal via het GMS worden beheerd. Met deze test zal onderzocht worden wat het effect is van een grote datahoeveelheid op het systeem, door bijvoorbeeld het verband tussen de databasegrootte en de laadtijd van een webpagina te meten. Deze test zal worden uitgevoerd tegen het einde van het project, om de instellingen te fine-tunen. De test kan gedaan worden met behulp van een profiler, welke de verstreken tijd kan berekenen. Het is bij deze test vereist om kennis te hebben van de onderliggende code, zodat profiler kan tonen welke functieaanroepen relatief veel tijd in beslag nemen.

#### **5.1.6 Black-box test: uiteindelijke acceptatie test**

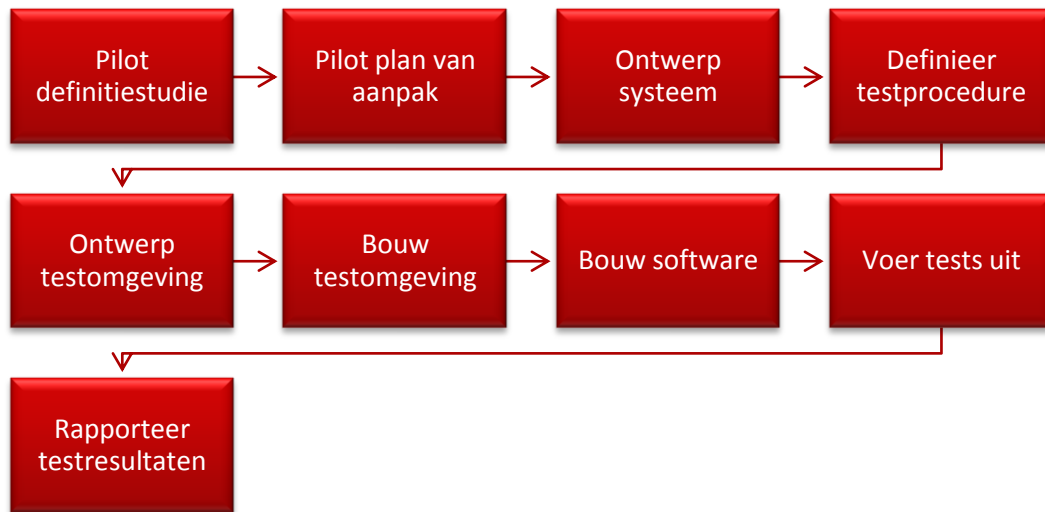
Ter afsluiting van een pilot of het gehele project kan een acceptatietest gedaan worden. Deze test bestaat uit een checklist waarmee de aanwezigheid van functionele eisen kan worden vastgesteld. Deze test zal worden uitgevoerd door het team wat de software gaat installeren en door personen die het systeem daadwerkelijk gaan gebruiken. Deze laatste gebruikers kunnen hun eigen testcriteria opstellen en zelf deze tests uitvoeren.





## 5.2 Testproces

Per pilot wordt een proces doorlopen. Per pilot worden de tests uitgevoerd en beschreven in een testrapport. Hieronder een weergave van de procescyclus.



## 5.3 Testplanning

Aan het einde van iedere pilot worden tests uitgevoerd. Met deze tests kan worden aangetoond dat de geïmplementeerde code voldoet aan de eisen van de pilotdefinitie. Eventuele negatieve testresultaten worden opgenomen in de evaluatie van de daaropvolgende pilot, waarin de problemen of oneffenheden worden onderzocht en behandeld.

Hieronder een overzicht van de pilots en de bijbehorende testmethoden. Dit zijn de tests die worden uitgevoerd tijdens een pilot.

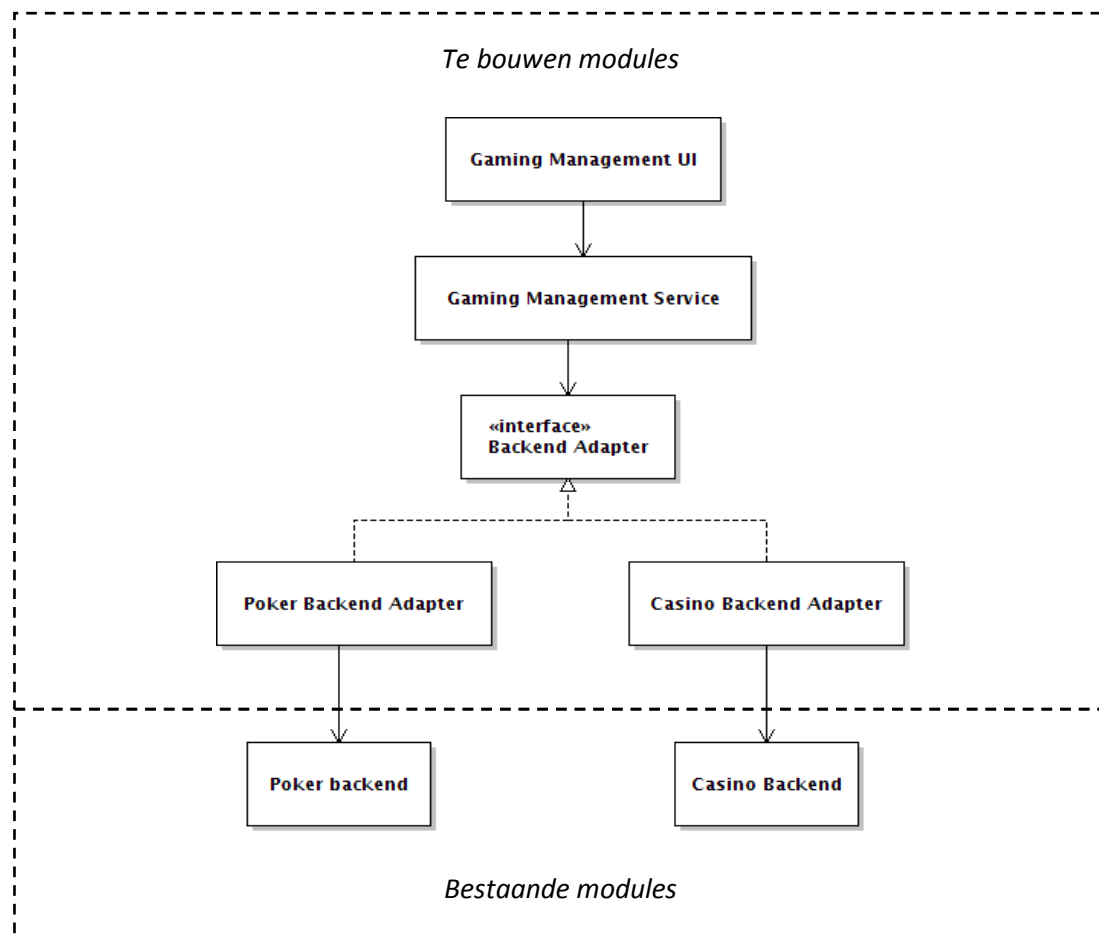
Pilot	Omschrijving	Testmethoden
Iteratie	Uitvoeren na een implementatie	Validatietest
Eindpilot	Afronding project	Unit tests
		Test gebruikersvriendelijkheid
		Load test
		Uitbreidbaarheid test
		Functionele test
		Integratietest

## 5.4 Opbouw Game Management Systeem

### 5.4.1 Gewenste opbouw Game Management Systeem

De huidige casinosoftware is geschreven in Java en sinds 2005 in gebruik. Deze software zal als voorbeeld dienen voor het GMS. Het GMS is immers een verzameling van backends, die allemaal de functionaliteiten hebben zoals het casinobackend. Een backend is de laag die onder een management systeem zit, waarin de functies en methoden zitten die een beheerder kan aanroepen om het systeem te onderhouden. Voorbeelden van backendmethodes zijn het weergeven van alle spelers of verwijderen van een gebruiker.

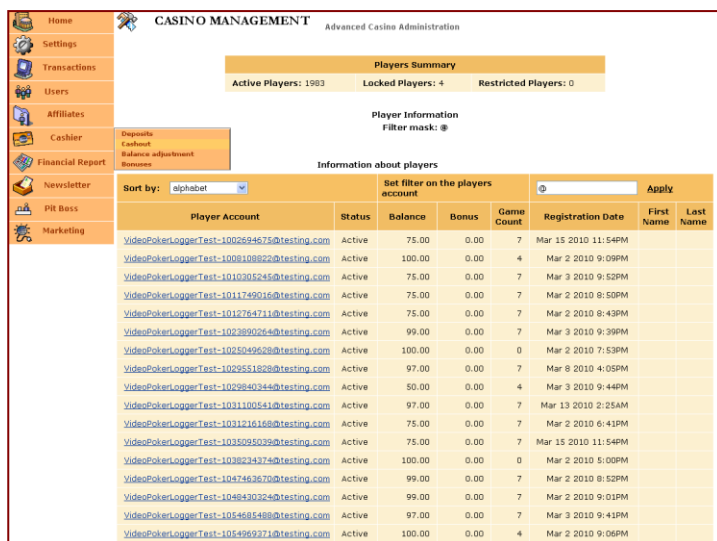
Het GMS zal een overkoepelende service zijn, die de beschikbare backends beheert. Zo kunnen er bijvoorbeeld 21 casinobackends en 3 pokerbackends zijn aangesloten op een GMS. Via de webinterface (webbrowser) kan een ingelogde beheerder deze backends onderhouden (zie Figuur 5-1). Er is sprake van 3 verschillende lagen; de objecten en functies uit de huidige backends kunnen worden aangesproken met behulp van een backendadapter. Dit is een wrapper (adapter design pattern) die de backends aan elkaar gelijk maakt; de objecten tussen beide adapters zijn onderling uitwisselbaar, ondanks de verschillende implementaties van de onderliggende 'echte' backends. Het GMS zal de collectie van deze backendadapters beheren, die een medewerker via een WebUI kan gebruiken. In een later stadium zullen de functionaliteiten als een issuetracker worden toegevoegd aan dit GMS.



Figuur 5-1: Schema van Game Management Service, zoals beschreven in het Plan van Aanpak

## 5.4.2 Onderzoek werking Casinobackend

Het managementsysteem van het casino bevat veel mogelijkheden voor beheer. Het is de bedoeling dat al deze mogelijkheden ook beschikbaar zijn in het te ontwerpen Game Management Systeem. In de komende periode zal er een ontwerp moeten worden gemaakt van dit GMS. Om inzicht en duidelijkheid te krijgen over de opbouw van een backend is besloten om met een testproject te onderzoeken hoe deze backend aangeroepen moet worden.



The screenshot shows the 'CASINO MANAGEMENT' interface with a sidebar menu on the left containing options like Home, Settings, Transactions, Users, Affiliates, Cashier, Financial Report, Newsletter, PR Boss, and Marketing. The main content area displays a 'Players Summary' with statistics: Active Players: 1983, Locked Players: 4, and Restricted Players: 0. Below this is a 'Player Information' section with a filter mask. The 'Information about players' section features a table with columns for Player Account, Status, Balance, Bonus, Game Count, Registration Date, First Name, and Last Name. The table lists 15 test accounts, all with a status of 'Active' and a balance of 75.00 or 100.00.

Player Account	Status	Balance	Bonus	Game Count	Registration Date	First Name	Last Name
VideoPokerLoggerTest-1002694675@testing.com	Active	75.00	0.00	7	Mar 15 2010 11:54PM		
VideoPokerLoggerTest-1008108822@testing.com	Active	100.00	0.00	4	Mar 2 2010 9:09PM		
VideoPokerLoggerTest-1010305245@testing.com	Active	75.00	0.00	7	Mar 3 2010 9:52PM		
VideoPokerLoggerTest-1011749016@testing.com	Active	75.00	0.00	7	Mar 2 2010 8:50PM		
VideoPokerLoggerTest-1012764711@testing.com	Active	75.00	0.00	7	Mar 2 2010 8:43PM		
VideoPokerLoggerTest-1023890264@testing.com	Active	99.00	0.00	7	Mar 3 2010 9:39PM		
VideoPokerLoggerTest-1025049528@testing.com	Active	100.00	0.00	0	Mar 2 2010 7:53PM		
VideoPokerLoggerTest-1029551828@testing.com	Active	97.00	0.00	7	Mar 8 2010 4:05PM		
VideoPokerLoggerTest-1029840344@testing.com	Active	50.00	0.00	4	Mar 3 2010 9:44PM		
VideoPokerLoggerTest-1031100541@testing.com	Active	97.00	0.00	7	Mar 13 2010 2:25AM		
VideoPokerLoggerTest-1031216168@testing.com	Active	75.00	0.00	7	Mar 2 2010 6:41PM		
VideoPokerLoggerTest-1035095039@testing.com	Active	75.00	0.00	7	Mar 15 2010 11:54PM		
VideoPokerLoggerTest-1038234379@testing.com	Active	100.00	0.00	0	Mar 2 2010 5:00PM		
VideoPokerLoggerTest-1047463670@testing.com	Active	99.00	0.00	7	Mar 2 2010 8:52PM		
VideoPokerLoggerTest-1048430324@testing.com	Active	99.00	0.00	7	Mar 2 2010 9:01PM		
VideoPokerLoggerTest-1054685488@testing.com	Active	97.00	0.00	7	Mar 3 2010 9:41PM		
VideoPokerLoggerTest-1054969371@testing.com	Active	100.00	0.00	4	Mar 2 2010 9:06PM		

Figuur 5-2: Speleroverzicht in het casino management systeem

Zoals eerder beschreven, is het casino gebouwd met verouderde ontwikkeltechnieken; het is niet mogelijk een (java) instantie van het casino aan te maken. Er zijn enkel static methoden en variabelen beschikbaar, wat betekent dat een server maar één casino tegelijk kan draaien. Ieder land (met een eigen casino) zal dus een eigen server moeten hebben, wat de kosten aanzienlijk zal verhogen. Wellicht is het mogelijk om gebruik te maken van de java klasse ThreadLocal, die per instantie een aparte thread start met een eigen scope. Verder bleken de resultaten van het casinobackend untyped Vektoren te zijn. Hierdoor is het niet duidelijk is wat voor objecten een methode retourneert, en of deze altijd van hetzelfde type zijn. Dit zal betekenen dat in de code bij iedere vector gecontroleerd moet worden of een object wel van het juiste type is, wat zal resulteren in verlies van snelheid ten opzichte van typed lijsten. Uitleg van het backend zal volgen aan de hand van Codevoorbeeld 5-1.



```

1 public static void main(String[] args) throws BaseException {
2     CasinoDB.JNDI_GOLD_ALIAS = "jndigold";
3     CasinoDB.JNDI_GOLD_STAT_ALIAS = "jndigoldstat";
4
5     Vector<Vector<String>> output = new Vector();
6     CustomerInformation.getCustomerList(null, 1, 1, "%", 1, output);
7
8     Vector<String> columns = output.elementAt(0);
9     String columnnames = "";
10
11     for (int i = 0; i < columns.size(); i++) {
12         columnnames += columns.elementAt(i) + ", ";
13     }
14     System.out.println("Columns: " + columnnames);
15 }

```

**Codevoorbeeld 5-1: Print de kolommen van de spelerinformatie**

In het codevoorbeeld is te zien hoe een verbinding tot stand gebracht kan worden met het casinobackend. Allereerst wordt in regel 2 & 3 opgegeven welke database aliases er moeten worden gebruikt bij de verbinding. Deze zijn gelinkt aan een databaseconfiguratiebestand ("*poolman.xml*"), wat in de *<default package>* van het javaproject geplaatst is.

Daarna wordt in regel 6 de in regel 5 aangemaakte vector gevuld met customers (klanten). De eerste parameter van de methode is de zoekterm. Aangezien er geen zoekterm (*null*) is opgegeven, worden alle klanten geretourneerd.

De uitkomst van de zoekactie wordt als tabel teruggegeven, namelijk een *Vector<Vector<String>>*. De eerste voorkomende vector (*elementAt(0)*) zijn de labels van de kolommen. De daaropvolgende vectoren bevatten de data (niet in dit codevoorbeeld opgenomen). Opvallend is dat alle datatypes worden teruggegeven als string; booleans als "0" of "1", integers en doubles als "34" en "45.86". Aan de hand van het label zal dus moeten worden bepaald of het een integer, double, boolean of string betreft.

Dit testproject heeft bewezen dat het mogelijk is om de informatie uit het casinobackend te gebruiken in een ander project. De casino backendadapter zal hier gebruik van maken.



## 5.5 Projectplanning

### 5.5.1 Pilot prioriteiten

Zoals in hoofdstuk 4.3 beschreven bestaat het Game Management Systeem uit een aantal onderdelen, die als volgt te groeperen zijn:

- Overzicht met spelers en details van een speler
- Instellingen Game Management Systeem
- Onderhoud spellen van een backend, met mogelijkheid om bijvoorbeeld spellen uit te schakelen.
- Financiële functies, zoals overzicht van uitbetalingaanvragen en stortingenoverzicht
- Marketing gerelateerde functies, zoals overzicht van campagnes en bonusprogramma's
- Overzicht en contacten met affiliates

Bij een casino zijn de inkomsten en tevredenheid van de spelers het belangrijkste. Zonder casinospelers kan er immers geen geld verdiend worden. Dit onderdeel zal de hoogste prioriteit krijgen. Om te zorgen dat het Game Management Systeem niet voor iedere bezoeker beschikbaar is zal er een inlogsysteem gemaakt moeten worden. Dit zal in de tweede pilot worden behandeld.

Het tevreden houden van de klanten bestaat ook uit het bieden van goede ondersteuning (support) bij eventuele stortingsproblemen. Dit zal de derde prioriteit krijgen. Hoewel een casino zonder spellen niet bestaan kan, wordt de mogelijkheid tot het afsluiten van spellen niet gezien als een kritiek onderdeel van het GMS. Deze zal een lage prioriteit krijgen. De gewenste functionaliteiten rondom de affiliates, campagnes en bonusprogramma's zijn nog niet geheel duidelijk. In een volgende planning zullen deze worden toegevoegd.

Deze onderdelen zijn verdeeld over verschillende pilots, die in de projectplanning zijn ingedeeld. Bij de volgorde van de pilots is de prioriteit van de onderdelen gebruikt.



## 5.5.2 Planning

Onderstaande planning is opgesteld na het samenstellen van het plan van aanpak, versie 17 februari 2010. De voorgaande (eerste) planning was niet meer van toepassing, omdat daarin het gebruik van pilots uit de IAD methode onjuist was verwerkt. Alle planningen zijn te vinden in de bijlage.

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
1	8-2-2010	Planning maken, onderzoek naar user interface design methoden uitvoeren, te gebruiken java web framework vaststellen, werkplek en computer (software) klaar maken voor ontwikkeling. Logboek starten.	Plan van Aanpak	
2	15-2-2010	Huidige backoffice implementaties onderzoeken, 1 <sup>e</sup> definitiestudie, start ontwerp & implementatie 1 <sup>e</sup> pilot	Definitiestudie 1 Pilotontwerp 1	1
3	22-2-2010	Vervolg vorige week		1
4	1-3-2010	Testplan maken, Mock server opzetten en tests uitvoeren	Testplan Testrapport	1
5	8-3-2010	Pilot 2; settings & newsletters		2
6	15-3-2010	Vervolg vorige week		2
7	22-3-2010	Testen pilot 2	Testrapport	2
8	29-3-2010	Pilot 3; Transactions, Cashier & Financial reports		3
9	5-4-2010	Vervolg vorige week		3
10	12-4-2010	Testen pilot 3	Testrapport	3
11	19-4-2010	Pilot 4: Pitt boss & dashboard		4
12	26-4-2010	Vervolg vorige week		4
13	3-5-2010	Testen pilot 4	Testrapport	4
14	10-5-2010	Uitloop / extra functionaliteiten	Testrapport	5
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



## 6 1<sup>e</sup> Pilot: proof of concept

In deze pilot zal een begin worden gemaakt met het ontwerpen en implementeren van het Game Management Systeem. Deze pilot zal gebruikt worden om een proof of concept te maken; een overzicht van data uit de twee backends, getoond in één webapplicatie.

### 6.1 Definitiestudie 1<sup>e</sup> pilot

#### 6.1.1 Systeemeisen 1<sup>e</sup> pilot

In het plan van aanpak is besloten om het project deel voor deel op te bouwen en uit te breiden. Voor de definitiestudie is een lijst gemaakt met de functionaliteiten die beide backends bieden (zie bijlage, definitiestudie 1<sup>e</sup> pilot). Hieruit zijn de acties voor spelerweergave gekozen, omdat dit de eenvoudigste functionaliteit lijkt. Er kan waarschijnlijk net als in het Codevoorbeeld 5-1 een datatabel gevormd worden die de spelers weergeeft. Met dit relatief eenvoudige voorbeeld kan er een begin gemaakt worden aan alle onderdelen van het systeem. Hoewel in de eerste planning nog werd gesproken van een projectindeling waarbij in iedere pilot een module zal worden gemaakt, lijkt het nu beter om de modules per pilot te laten groeien. De systeemeisen die voor deze pilot zijn gespecificeerd, zijn als volgt beschreven:

*In de eerste pilot zal het gedeelte van de players / users worden ontworpen en geïmplementeerd. Hier zullen de volgende systeemeisen worden gehanteerd;*

- *Spelers tonen en bladeren*
- *Spelers toevoegen*
- *Spelers verwijderen*
- *Toon winnende spelers*
- *Toon verliezende spelers*

*Momenteel worden deze functies nog niet ondersteund in de pokerbackend, maar de backend moet dit (later) wel kunnen leveren. Met deze (redelijk) eenvoudige functies kan een begin gemaakt worden aan het GMS en de backendadapter (interface). Zo kan met een simpele functionaliteit inzicht verkregen worden in de meest bruikbare opbouw van de modules, bijvoorbeeld met behulp van een klassendiagram.*

**Citaat 6-1, uit: Definitiestudie 1<sup>e</sup> pilot**

Na deze pilot zal er dus een lijst met spelers, een lijst met winnende spelers en een lijst met verliezende spelers te tonen zijn.





### 6.1.2 Randvoorwaarden

De software van de pokerbackend is nog niet beschikbaar. Dhr. Schlärmann is bezig om deze server op te zetten en te configureren. Voor deze pilot zal de casino backend gebruikt worden voor het testen van de software. Er is inmiddels een testversie van de casinosoftware beschikbaar.

Een belangrijke voorwaarde bij het maken van een GMS is dat de gewenste functies ook beschikbaar zijn in de backend. Er zal in deze pilot echter geen functionaliteit worden toegevoegd aan de bestaande casino en pokersoftware.

### 6.1.3 Beperkingen

Zoals eerder beschreven is de backend van de pokersoftware nog niet beschikbaar. Deze zal waarschijnlijk bij de start van deze pilot beschikbaar komen. Het ophalen van informatie uit deze (op C# gebaseerde) software zal niet in deze pilot worden geïmplementeerd.

Deze pilot draait vooral om het ontwerpen van de backend interface, ontwerpen van het GMS en het implementeren van de backend adapter voor de casino software. De webinterface zal in deze pilot in mindere mate aan bod komen.

### 6.1.4 Systeemconcept

De gebruiker moet in de webinterface een overzicht kunnen krijgen van de huidige spelers, die te bekijken, te verwijderen of toe te voegen zijn. Daarnaast moet de beheerder inzicht kunnen krijgen een lijst met winnende en verliezende spelers. Aan de hand van de winnende spelers kunnen ook onjuiste toekenningen (valsspelen) gedetecteerd worden.

### 6.1.5 Pilotplan

De komende pilot is een onderdeel van een verzameling pilots die uiteindelijk tezamen het volledige project vormen. Voor de toekomst staan de volgende pilots in de planning:

- Settings en newsletter (beheerdersopties)
- Transactions, Cashier & Financial reports (geldzaken)
- Pit boss en dashboard (extra beheerdersopties & afronding)



## 6.2 Plan van Aanpak 1<sup>e</sup> pilot

Nadat er in de definitiestudie was bepaald wat het doel van de eerste pilot zal zijn, wordt er in een plan van aanpak beschreven hoe aan de piloteisen kan worden voldaan. Dit plan beschrijft de op te leveren resultaten, kwaliteitszorg en tijdschatting. De op te leveren resultaten zijn hieronder te lezen; zie voor meer informatie het “Plan van Aanpak 1<sup>e</sup> pilot” in de bijlage.

1. *Softwareontwerp van backend, GMS en GUI; met functionaliteiten voor players. Het softwareontwerp zal later worden uitgebreid met de andere functionaliteiten.*
2. *Implementatie van backend interface, GMS en GUI; voor players.*
3. *Implementatie van spelers in casino backendadapter, zoals beschreven in definitiestudie*
4. *Testrapport van de ontwikkelde software.*

Citaat 6-2, uit: Plan van aanpak 1<sup>e</sup> pilot

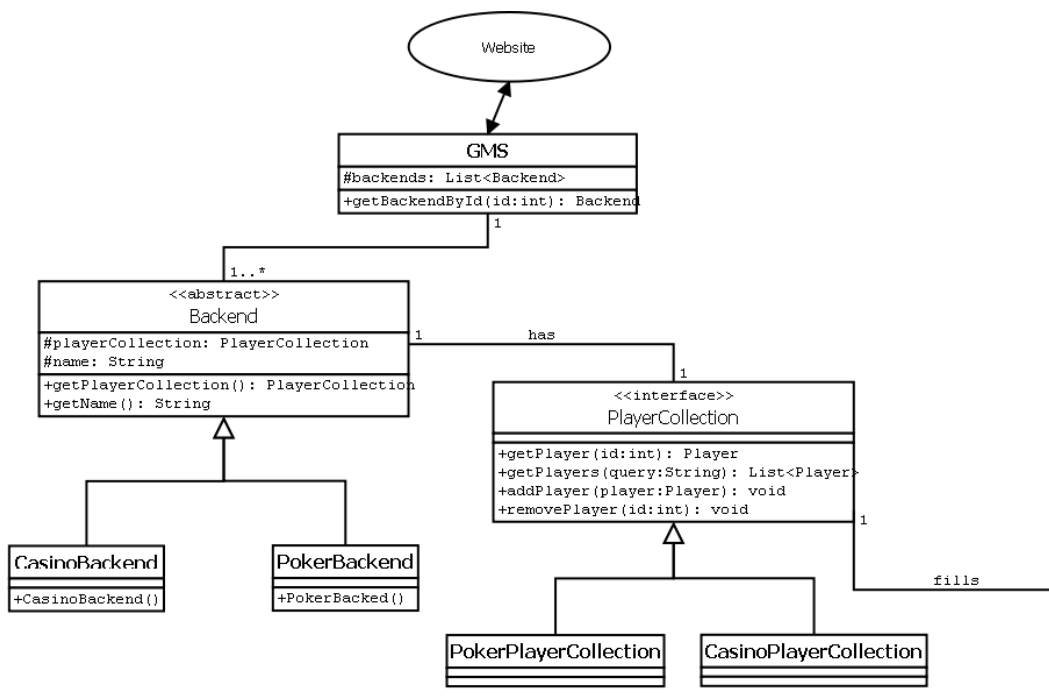
### 6.2.1 Ontwerp backendadapter

Het doel is om de huidige backends aan te kunnen roepen via een backendadapter, zodat bestanden en informatie tussen de backends overdraagbaar zijn, ondanks de verschillende onderliggende implementatie. Bij de start van de pilot is de testserver van de pokersoftware nog niet beschikbaar, maar die van het casino wel. Er is daarom eerst begonnen met het ontwerp en implementatie van de backendadapter van het casino.

Voor aanvang van de pilot is er onderzoek gedaan naar de werking van het casino backend. Aangezien de backends opgedeeld kunnen worden in een aantal taakgebieden, is gekozen om deze af te handelen in verschillende Java classes. Zo zouden bijvoorbeeld alle acties omtrent spelers in een aparte klasse kunnen worden afgehandeld. Zoals te zien in Figuur 6-1, is er gekozen om zo’n verzameling een ‘Collection’ te noemen. De PlayerCollection heeft bijvoorbeeld methoden om spelers op te halen, te verwijderen of toe te voegen. De website bepaalt de weergave en gebruik van deze functies. Zolang alle backends overerven van de interface of abstracte classes (*extends* of *implements*), zijn de objecten overdraagbaar tussen de softwaresystemen.

Zoals te zien in Figuur 6-1, heeft iedere backend een unieke naam. In de constructor van de overervende class worden de parameters meegegeven voor het aanmaken van de verbinding met de onderliggende software. Dit kan een verbinding zijn met een database, of een verbinding met een gameserver. De rol van het backend is het onderhouden van de sessie, en het in stand houden van de verschillende collections. Voor deze pilot wordt alleen de PlayerCollection ontworpen en geïmplementeerd.





**Figuur 6-1: Gedeelte van het klassendiagram voor het Game Management Systeem, met voornamelijk de focus op het ontwerp van de backendadapter. Zie voor de volledige versie het pilotdocument.**

De PlayerCollection bevat de methoden voor het onderhoud van de spelers. Het is mogelijk om spelers (*List<Player>*) op te halen met behulp van een SQL query, of een enkele speler aan de hand van zijn unieke ID. Verder is het mogelijk om een speler toe te voegen en een speler te verwijderen aan de hand van zijn ID. Een *Player* object is een class met getters en setters voor naam, adres, woonplaats, email, inlognaam, wachtwoord en dergelijke.

## 6.2.2 Gebruik van Seam en Richfaces

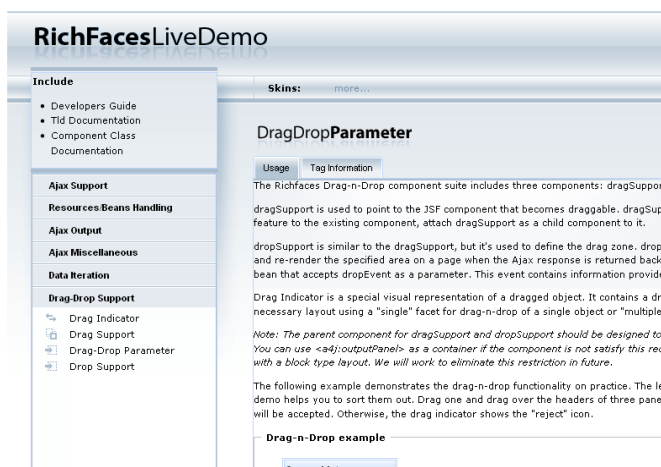
Zoals beschreven in de opdrachtsomschrijving moet de software beter onderhoudbaar zijn. Door gebruik te maken van een officieel web framework zal er meer zekerheid zijn over de efficiëntie van de code en is er meer informatie te vinden op internet over de toepassing van het MVC (Model View Controller) design pattern in dit web framework. Dit verhoogt de aanpasbaarheid en overdraagbaarheid. Door gebruik te maken van een framework is de software beter onderhoudbaar, aangezien er dan gebruik kan worden gemaakt van bestaande configuraties en indeling van bestanden. Door een andere student is tijdens een voorgaande stage onderzocht welk web framework het beste geschikt zou zijn.



Enkele voorbeelden van frameworks zijn *JavaServer Faces*, *Apache Wicket*, *JBoss Seam*, *Spring MVC* en *Struts*. Er is uiteindelijk gekozen voor Seam, wat is goedgekeurd door de bedrijfsmentor van die student. Seam is ondertussen al in enkele projecten toegepast, wat goed bevallen is. MC heeft daarom bepaald dat ook voor dit project Seam zal worden gebruikt.

Seam is ontwikkeld voor JBoss Application Server, onder leiding van Gavin King. King is onder andere bekend van zijn bijdrage aan het object-gerelateerde mapping framework Hibernate. Seam kan worden gebruikt onder Java Enterprise Edition 5, een Java platform voor servers. Seam combineert de twee frameworks Enterprise JavaBeans (EJB3) en JavaServer Faces (JSF). Een EJB back-end component kan worden aangeroepen door de front-end door gebruik te maken van de Seam componentnaam. Met de komst van Seam wordt ook de ondersteuning van bijection geïntroduceerd zoals toegepast in de Spring dependency injection feature. Hiermee worden injection en outjection van objecten mogelijk, aangegeven met `@In` en `@Out` annotaties. Zie voor meer uitleg Codevoorbeeld 6-2.

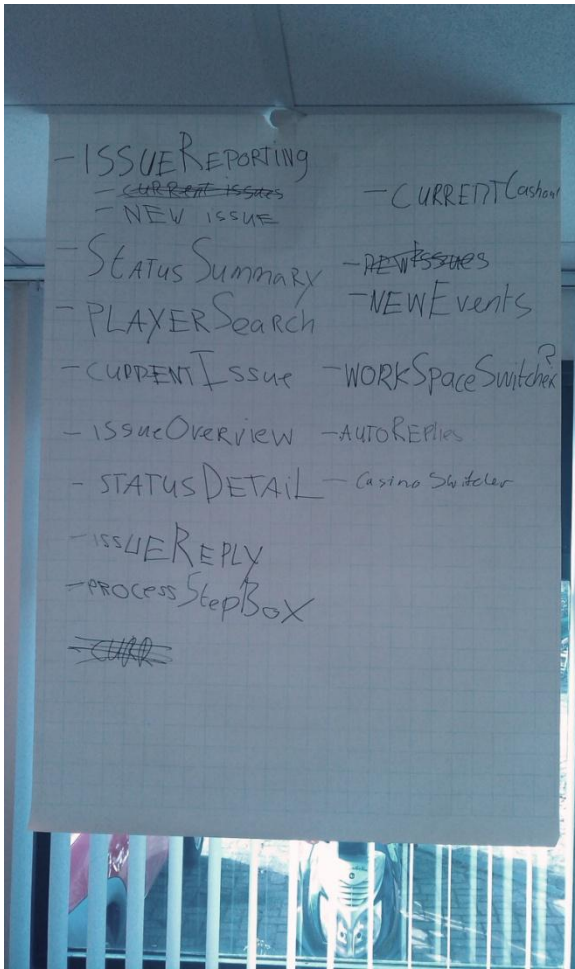
Naast Seam zal er gebruik worden gemaakt van Richfaces, een Rich Component Library die is gebaseerd op het ajax4jsf framework. Met behulp van Richfaces kunnen er eenvoudig Ajax elementen worden toegevoegd aan een webpagina. Zo worden pagina's dynamisch geüpdate, zonder de webpagina te herladen. Dit bevordert de gebruiksvriendelijkheid omdat er op deze manier multi-tasking kan worden toegepast.



Een online demo van de componenten van Richfaces kan gevonden worden op: <http://liferay.exadel.com/web/guest>



### 6.2.3 Web User Interface mockup



**Figuur 6-2: Eisen en wensen Web GUI**

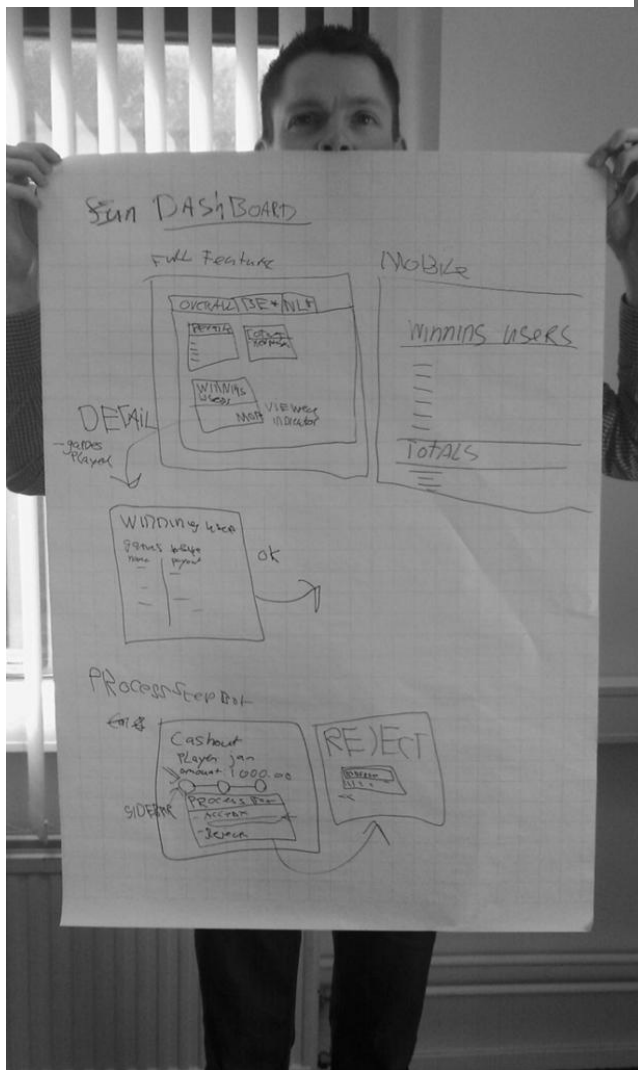
webpagina kan gebruik maken van de sidebar **(3)**. Deze zal pagina-relevante snelkoppelingen of widgets bevatten. De titel van de pagina is overigens te zien bij **(4)**. De beheerders kunnen meerdere backends vanuit deze webUI beheren. Deze backends kunnen worden geselecteerd in de select-box **(5)**. In het veld bij **(6)** worden systeemmeldingen en issues getoond. Hier kan een beheerder bijvoorbeeld info of cashout requests zien, of weergeven wat de (te voltooien) taken voor die dag zijn. Dit venster gaat veelvuldig gebruikt worden als er meerdere beheerders voor één backend zijn; de taakverdeling zal hierin zichtbaar zijn. Er zijn in totaal 18 mockup schermen of onderdelen ontworpen en uitgewerkt.

In de tweede week van het project is er een vergadering gehouden waarin de systeemeisen werden bepaald (zie hoofdstuk 0). In de derde week is er een vergadering geweest met de bedrijfsmentor om te bepalen hoe de systeemeisen verwerkt kunnen worden in een nieuw User Interface ontwerp. In de vergadering zijn er verschillende mockups getekend. Een mockup is een voorstadium van een prototype, dus een voorstel voor een te ontwerpen onderdeel. Er is een lijst met functies gemaakt die de webUI moet bieden, en een groepering van deze items. Deze zijn terug te vinden in het document over het toepassen van Usage Centered Design.

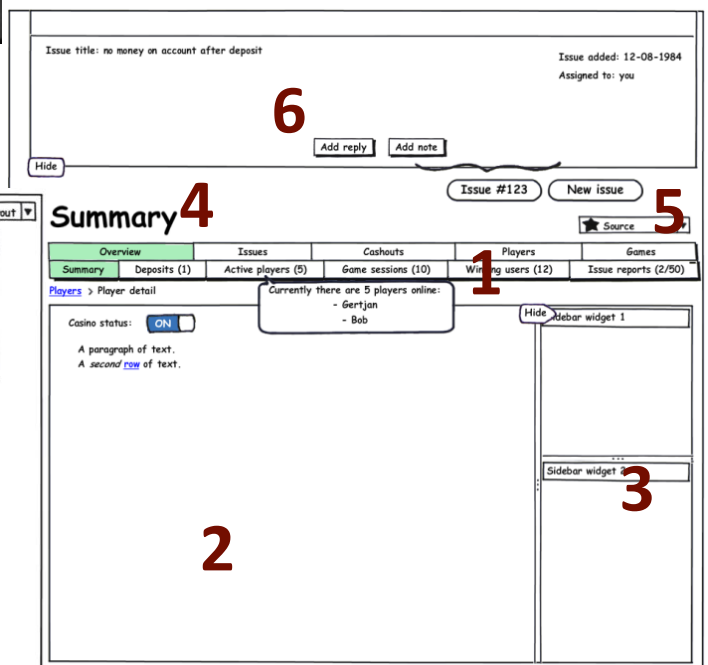
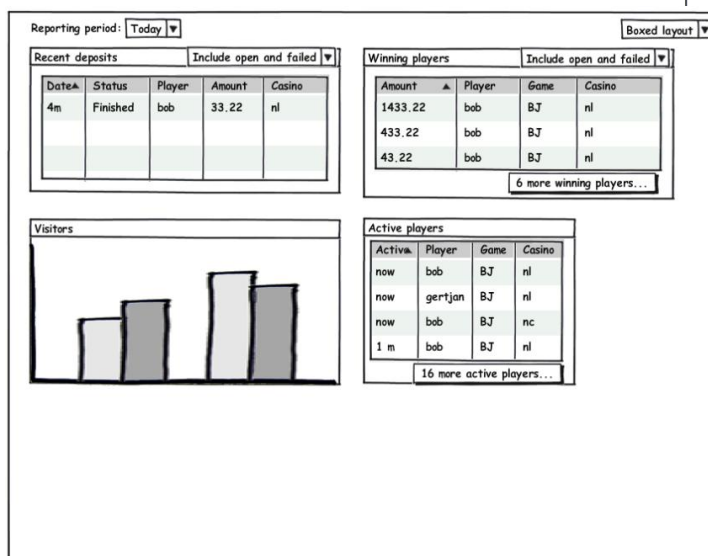
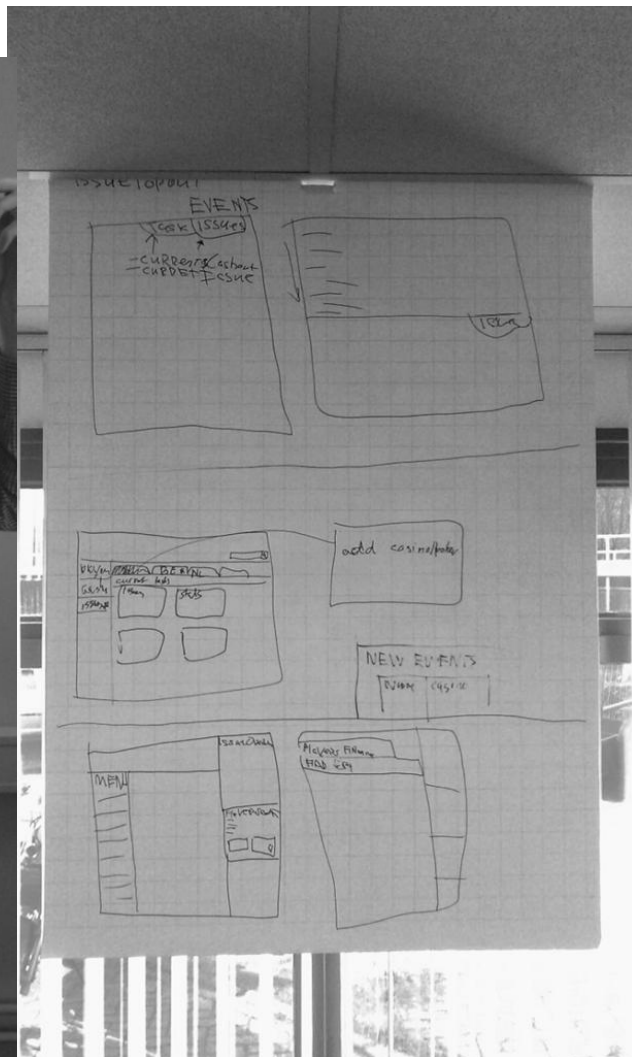
Hierna zijn er verschillende ontwerpschetsen gemaakt voor het uiteindelijke uiterlijk van de webpagina's, zie Figuur 6-3. Na de vergadering zijn de schetsen gedigitaliseerd en uitgewerkt in Balsamiq Mockup Studio.

In de mockups (Figuur 6-4) is te zien dat de pagina bestaat uit een aantal onderdelen. Met de navigatiebalk **(1)** kunnen de verschillende webpagina's bereikt worden. De webpagina's tonen de inhoud in het grote veld **(2)**. Iedere





Figuur 6-3: Mockupschetsen van de Web GUI



Figuur 6-4: Mockups van de Web GUI, gemaakt in Balsamiq Mockup Studio

### 6.3 Implementeren ontwerp 1<sup>e</sup> pilot

Het ontwerp is grotendeels bekend, het uiterlijk is bekend en de piloteisen zijn bekend; er kan begonnen worden aan de implementatie van de code. In deze pilot staat het weergeven van de spelers centraal. Met Richfaces is het mogelijk om een datatabel te genereren, waarin de lijst met spelers getoond kan worden. Hieronder een codevoorbeeld van zo'n pagina;

```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml"
5       xmlns:ui="http://java.sun.com/jsf/facelets"
6       xmlns:h="http://java.sun.com/jsf/html"
7       xmlns:c="http://java.sun.com/jsp/jstl/core"
8       xmlns:s="http://jboss.com/products/seam/taglib"
9       xmlns:a4j="http://richfaces.org/a4j"
10      xmlns:rich="http://richfaces.org/rich"
11      xmlns:f="http://java.sun.com/jsf/core">
12 <body>
13   <ui:composition template="../template.xhtml">
14     <ui:define name="uibody">
15       <a4j:form ajaxSubmit="true" ajaxSingle="true">
16         <rich:extendedDataTable id="playerTable"
17           value="#{playerData}" var="player" width="700px" height="402px"
18           sortMode="multi" selectionMode="single">
19           <rich:column sortable="true" sortBy="#{player.id}"
20             label="#{messages['person.id']}">
21             <f:facet name="header">
22               <h:outputText value="#{messages['person.id']}" />
23             </f:facet>
24             <h:outputText value="#{player.id}" />
25           </rich:column>
26           <rich:column sortable="true" sortBy="#{player.firstname}"
27             label="#{messages['person.firstname']}"
28             filterBy="#{player.firstname}" filterEvent="onkeyup"
29             filterValue="#{players.filter.firstname}">
30             <f:facet name="header">
31               <h:outputText value="#{messages['person.firstname']}" />
32             </f:facet>
33             <h:outputText value="#{player.firstname}" />
34           </rich:column>
35   ...
```

Codevoorbeeld 6-1: Ingekort gedeelte van de broncode in het bestand WEB-INF\players\Players.xhtml

In regel 4 t/m 11 zijn de xml namespace prefixes opgegeven, welke te vergelijken zijn met de imports in een Java bestand. Met de namespace prefix `xmlns:f="http://java.sun.com/jsf/core"` kan bijvoorbeeld gebruik worden gemaakt van de tag `<f:facet...>`.

Zoals te zien in regel 13 wordt bij deze pagina gebruik gemaakt van een template, waarin de pagina-indeling (sidebar, menubar, body) zijn gedefinieerd. Op regel 14 wordt de body in de template gevuld met de daaropvolgende code.

De datatabel (regel 16) is in dit voorbeeld gevuld met twee kolommen (start op regel 19 & regel 26). Van iedere kolom is de kolomkop (header) opgegeven, en de waarde uit het itererende object in dit geval `#{player}`.

```
1 /**
2  * @author Gertjan AI, Gaming & Gambling
3  */
4 @Name("Players")
```





```

5 public class Players implements Serializable
6 {
7
8     private String searchText;
9     @In(scope = ScopeType.SESSION, required = true)
10    private String selectedBackend;
11    @org.jboss.seam.annotations.Logger
12    private Log log;
13    @DataModel
14    private List<Player> playerData;
15    @In(required = true)
16    private GameManagementService gms;
17    ...
18
19    @Factory("playerData")
20    public void searchPlayer()
21    {
22        try
23        {
24            selection = null;
25            playerData = null;
26            playerData = gms.getSelectedBackend().getPlayerCollection().
27                getPlayers(searchText);
28        } catch (Exception e)
29        {
30            String error = "Couldn't get players from '" + selectedBackend
31                + "' " + e.getLocalizedMessage();
32            FacesMessages.instance().add(new FacesMessage(error));
33            log.error(error);
34        }
35    }
36    ...

```

**Codevoorbeeld 6-2: Achterliggende Javacode, behorende tot WEB-INF\players\Players.xhtml**

In Codevoorbeeld 6-2 is een deel van de javacode te zien die bij Codevoorbeeld 6-1 hoort. De `@In` annotaties geven aan dat dit object gevuld moet worden met het object uit de sessie, zoals het object `GameManagementService` (regel 16). De annotatie `@Factory("playerData")` geeft aan dat de daaropvolgende methode gebruikt moet worden om het object met de naam `playerData` te vullen. Tijdens het laden van de pagina zal deze methode dus worden uitgevoerd. In dit geval wordt er direct gezocht naar een speler. De variabele `searchText` is nog niet ingevuld, dus de methode retourneert alle spelers uit de backend.

De uitkomst van de vorige codevoorbeelden geeft onderstaand resultaat in de webbrowser. De templateonderdelen zijn voor de eenvoud weggelaten.

id	E-mail	Firstname	Lastname	Registration date
1252669	bobpokertest@test.lan			

**Figuur 6-5: Voorbeeld van een datamodel in Richfaces**

## 6.4 Pilotdocument 1<sup>e</sup> pilot

Tijdens het uitvoeren van de pilot is er onderzocht hoe Seam en Richfaces gebruikt konden worden voor dit project. Het is gelukt om een spelerslijst te tonen van beide backends. Desondanks is er besloten de pilot te stoppen; het systeemontwerp bleek niet goed aan te sluiten bij de mogelijkheden van Seam. Met deze kennis kan in de volgende pilot het ontwerp worden aangepast.

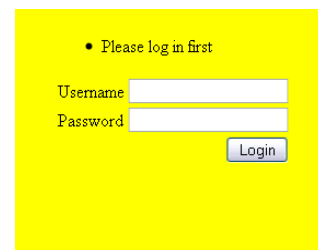
In deze pilot is een proof of concept gemaakt van het GMS, de user interface en de backends. Het is mogelijk om spelers te zoeken en weer te geven van de Poker en Casino software.

### 6.4.1 Backendadapter implementaties

De backendadapter van de casinosoftware maakt verbinding met de Java casino backoffice, waarin verschillende methoden de resultaten teruggegeven worden. Bij de pokersoftware (C#) is besloten om niet via rmi/remoting verbinding te maken met de backoffice, maar direct met de SQL database. Dit levert een hogere performance op dan als de data via de C# backend zou worden opgevraagd.

### 6.4.2 Beveiliging

De webinterface is gemaakt om de spelerlijsten te tonen, inclusief het formulier waarin de beheerder de zoekopdracht kan invoeren. De GUI is beveiligd met een wachtwoord, momenteel met de bekende *admin/admin* username/password combinatie. Later zal hier een database met gebruikers aan moeten worden gekoppeld. De GUI is momenteel opgezet in eenvoudige tabellen met rare kleuren; deze onderdelen hebben een vreemde kleur gekregen omdat ze nog moeten worden geconfigureerd in het css bestand. Op deze manier is direct te zien welke objecten nog moeten worden toegevoegd aan het stijlbestand.



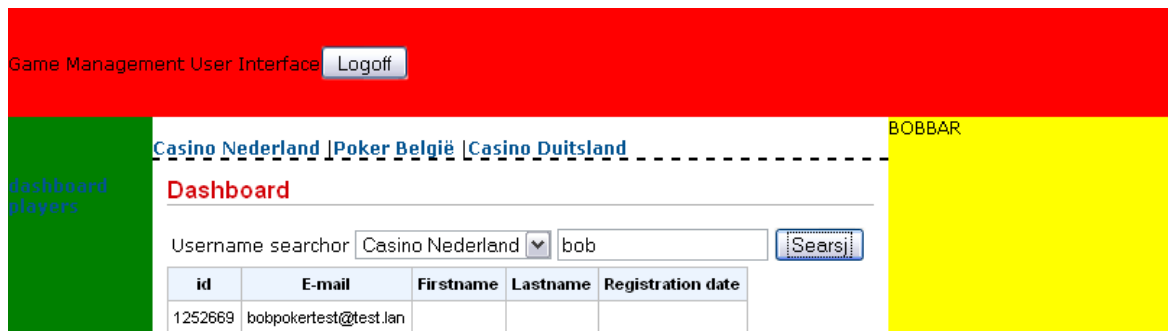
**Figuur 6-6: Inlogformulier van de GMS webinterface**

### 6.4.3 Spelerweergave

Na het inloggen kan de beheerder een van de twee backends selecteren in het formulier en een zoekopdracht opgeven. De twee backends zijn geconfigureerd in de JAVA code en niet dynamisch aan te passen. In een volgende pilot moeten deze worden aangemaakt naar de rol van de gebruiker (welke backends toegankelijk zijn voor de beheerder). De backends worden geladen als ze voor het eerst worden gebruikt, bijvoorbeeld bij een zoekopdracht. De verbinding wordt dus niet direct tot stand gebracht. Op deze manier worden er geen onnodige aanvragen naar de server(s) gedaan of instanties van de backends aangemaakt.

Merk op dat ook in deze layout afwijkende kleuren en expres foutieve spelling is gebruikt bij de teksten en knoppen. Deze teksten zullen met behulp van localization moeten worden omgezet, door bijvoorbeeld per beheerder een voorkeurstaal op te geven.





**Figuur 6-7: Screenshot van de gebruikersinterface met overzicht van spelers, mogelijkheid tot selectie van backend en zoeken van personen**

## 6.4.4 Datamodel

Voor het proof of concept is een datamodel gemaakt (klassendiagram). De achterliggende gedachte is dat iedere backend een *PlayerCollection* heeft, die alle spelerfuncties biedt. Bij een zoekopdracht geeft de *PlayerCollection* een *List* terug met *Player* objecten, die gevuld zijn met de data uit de backoffice van de respectievelijke backend. In de *PlayerCollection* worden dus de *Player* objecten gevuld en worden de key/value pairs (database kolommen en rijen) gemapt naar een gelijke data key en klasse.

Bijvoorbeeld:

1. De casinosoftware bestaat uit players, met o.a. een *String "Firstname"* en een *String "Registration Date"*
2. De pokersoftware bestaat uit players (join users & profiles), met o.a. een *String "Fullname"* en een *String "SignUpDate"*

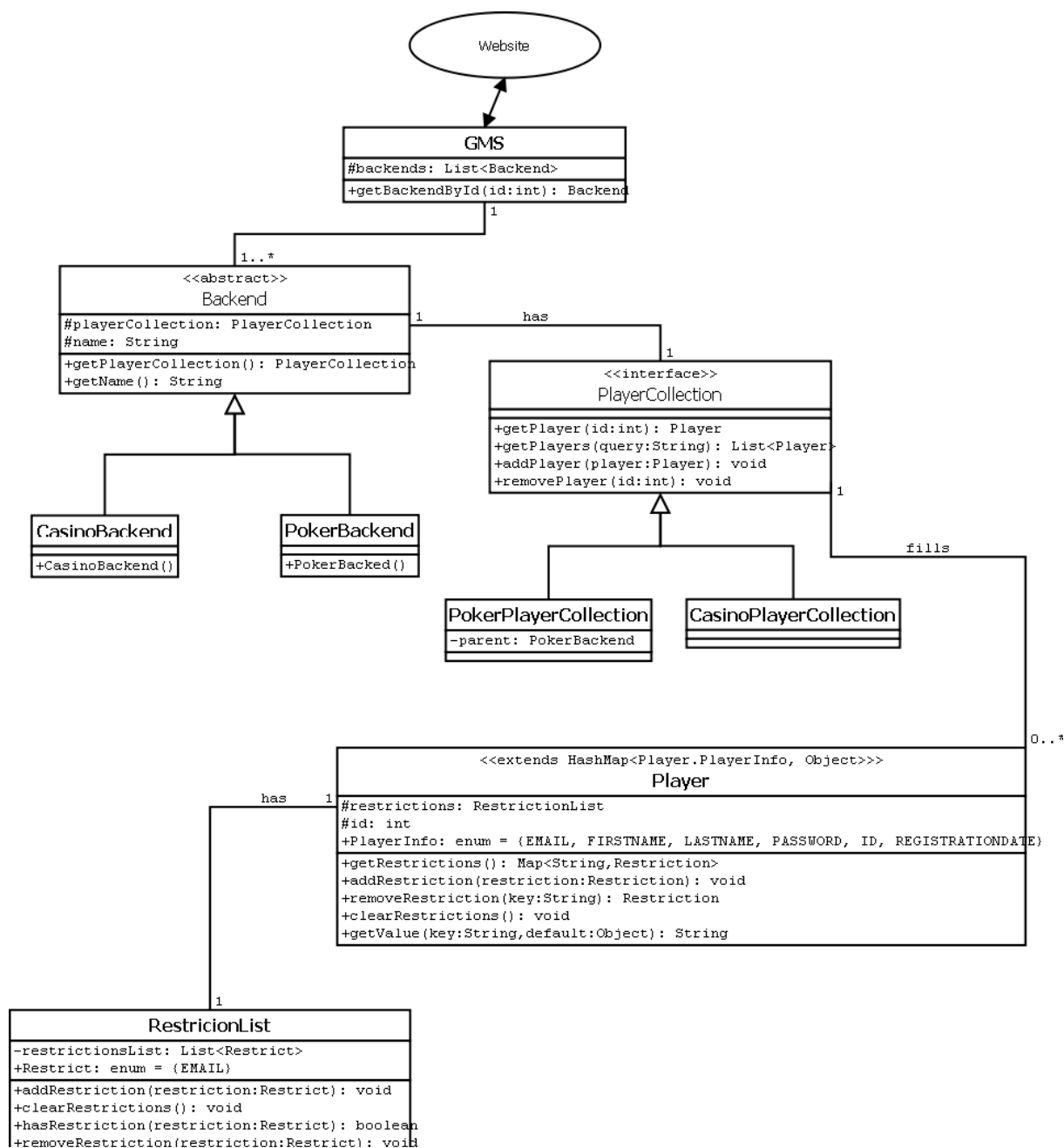
De voornamen worden gemapt naar *Player.PlayerInfo.FIRSTNAME (String)*, de registratiedatums worden gemapt naar *Player.PlayerInfo.REGISTRATION\_DATE (java.util.Date)*.

Ondanks de verschillen in opslag van de beide databases, is de opslag in de *Player* objecten dus wel gelijk. Op deze manier zijn de *player* objecten dus uitwisselbaar tussen de backends.

Players kunnen een restrictie opgelegd krijgen, bijvoorbeeld het ontzeggen van de toegang. Deze restricties worden ook aangemaakt door de *PlayerCollection*. In de *CasinoPlayerCollection* is een voorbeeld van deze restriction gemaakt voor de *EMAILRESTRICTION*.

Het domeinmodel op de volgende pagina is het ontwerp. Merk op dat in het proof of concept geen gebruik meer wordt gemaakt van een *List<Backend>* in *GMS*, maar een *Map<String<sub>(name)</sub>, Backend>*.





Figuur 6-8: Klassendiagram van GMS en de backends

### 6.4.5 Problemen tussen datamodel en Seam

Het gebruik van deze manier van opslag bleek echter niet goed te werken in combinatie met Seam. In het ontwerp is gekozen om de data op te slaan in een HashMap, zodat er geïtereerd kon worden door de variabelen. Het ophalen van een variabele in Seam zou dan echter verlopen zoals te zien is in Codevoorbeeld 6-3, regel 2. In het framework van Seam is echter de mogelijkheid om variabelen van objecten aan te roepen via de manier zoals in regel 3, waarbij automatisch de get-methode van de variabele wordt aangeroepen. Het doel van het gebruik van een framework was dat er gebruik zou worden gemaakt van standaarden. Het ontwerp voldoet niet aan deze standaard en zal moeten worden aangepast.

```
1  ...  
2  <h:outputText value="#{player.getValue(Player.PlayerInfo.FIRSTNAME)}"/>  
3  <h:outputText value="#{player.firstname}"/>  
4  ...
```

**Codevoorbeeld 6-3: Ophalen van de voornaam van een speler; regel 2 geeft aan hoe het met dit systeemontwerp moet worden gedaan, regel 3 geeft aan hoe het in Seam kan met behulp van automatisch gebruik van `player.getFirstname()`**

### 6.5 Conclusie 1<sup>e</sup> pilot

Het ontwerp voldoet niet aan de gestelde standaarden. Hierdoor is de implementatiefase van de pilot eerder gestopt dan gepland. Nadat het bij beide backends mogelijk was om spelers te zoeken en weer te geven leek het een goed plan om het softwareontwerp nogmaals na te lopen en te herzien. Er is nu immers een duidelijker beeld op de achterliggende software, waardoor het ontwerp kan worden verbeterd. Dit betekent dat een aantal systeemeisen van de eerste pilot worden verschoven naar de tweede pilot.

De volgende onderdelen uit de vorige pilot worden verschoven naar een volgende pilot:

- Weergeven winnende spelers
- Weergeven verliezende spelers
- Verwijderen spelers
- Toevoegen spelers
- Testen van de geïmplementeerde software. Omdat de opbouw totaal gaat veranderen in de tweede pilot is besloten niet te testen

De volgende onderdelen zijn in de vorige pilot toegevoegd:

- Security (single username/password logon)
- Implementatie van de pokerbackend adapter
- Verbinding met de poker database



## 7 2<sup>e</sup> Pilot: Spelers, affiliates & managers

### 7.1 Definitiestudie 2<sup>e</sup> pilot

Na het afsluiten van de 1<sup>e</sup> pilot is er een tweede definitiestudie gedaan om de eisen van de volgende pilot te specificeren. Daar de 1<sup>e</sup> pilot een proof of concept betrof, zal in deze pilot begonnen worden aan het implementeren van de uiteindelijke GMS.

In deze pilot zal gebruik worden gemaakt van de bevindingen uit de 1<sup>e</sup> pilot; er zal worden doorgewerkt met de functionaliteiten die in het proof of concept zijn geïmplementeerd. Momenteel is het mogelijk om players weer te geven en om players te zoeken. In deze pilot zal dit worden uitgebreid, zodat alle persoons-gerelateerde objecten (players, affiliates, managers) worden ingevuld.

#### 7.1.1 Doelen

##### 7.1.1.1 Player klasse naar Person klasse

Bij de backends zijn meerdere soorten personen betrokken; spelers, managers en affiliates. Deze personen hebben overeenkomstige persoonskenmerken zoals naam, adres, woonplaats, gebruikersnaam, wachtwoord. Hier kan abstractie worden toegepast door player, manager en affiliate te laten overerven van een nieuwe persoonklasse. Naast deze persoonsinformatie heeft bijvoorbeeld een speler een lijst van restricties (mogelijke blokkades) en een manager één of meerdere beheerdersrollen. Een affiliate heeft naast deze informatie ook referenties naar affiliate codes, bankrekeninggegevens, uitbetaalpercentages en ondergeschikte affiliates.

##### 7.1.1.2 Playerrestrictions implementeren van poker en casino

Momenteel worden de restrictions uit de casino en pokersoftware nog niet gekoppeld aan de restrictionlist in het Player object. Als proof of concept is de email-restriction uit casino geïmplementeerd. Nu zullen de overige restrictions moeten worden geïmplementeerd. Bij dit soort restricties kan gedacht worden aan het blokkeren van inloggen, blokkeren van toegang tot (bepaalde) spellen enz.

##### 7.1.1.3 Inloggen

Er kan nu worden ingelogd door middel van één enkele gebruikersnaam en wachtwoord combinatie. Dit moet veranderd worden naar een manager database, die op basis van de rol van de manager de benodigde beheeropties toegankelijk maakt. Ook moeten er managers kunnen worden aangemaakt, die op basis van hun gebruikersnaam hun eigen backend(s) toegewezen krijgen.



#### **7.1.1.4 Details van een persoon**

Van een speler moet de onlinestatus, de persoonsdetails en de balans getoond kunnen worden.

#### **7.1.1.5 Winnende en verliezende spelers**

Er moet een lijstje weergegeven kunnen worden per backend, waarin te zien is welke personen er onlangs gewonnen hebben en welke personen er onlangs verloren hebben (bijvoorbeeld een top 10 of winnaars binnen een bepaald datumbereik).

#### **7.1.1.6 Settings**

Managerspecifieke instellingen, waarin voorkeursinstellingen kunnen worden opgeslagen en waarin de toegankelijke rollen zijn opgeslagen.

### **7.1.2 Systeemeisen van de pilot**

Voorgaande doelen geven de volgende systeemeisen voor deze pilot:

- Mogelijkheid tot inzien van spelers, met bijbehorende details
- Mogelijkheid tot inzien van managers, met bijbehorende details
- Mogelijkheid tot inzien van affiliates, met bijbehorende details
- Mogelijkheid tot inloggen met accounts en toewijzing van backends a.d.h.v. gebruikersnaam
- Weergave van winnende spelers
- Weergave van verliezende spelers
- Verbeterd softwareontwerp, in de vorm van een klassendiagram
- Ondersteuning van restricties voor spelers
- Testrapport met unit test en integratietest

#### **7.1.3 Randvoorwaarden**

Een belangrijke voorwaarde bij het maken van de gebruikersomgeving is dat de gewenste functies ook beschikbaar zijn in de backend. Er zal in deze pilot geen functionaliteit worden toegevoegd aan de bestaande software.





### 7.1.4 Testrapport

De correcte werking van de backends is cruciaal bij het werken met de Game Management Service. De webinterface is enkel een gebruiksvriendelijke manier om de functionaliteiten in de backends aan te roepen. Om de consistentie van de backends te waarborgen zullen er unit tests worden gemaakt voor deze backendmethoden. In de unit test zal bijvoorbeeld een speler worden toegevoegd, waarna gecontroleerd wordt of deze correct is opgeslagen in de backend.

Naast de unit test zal er ook een integratietest worden gedaan om te controleren of de gebruikersomgeving naar behoren functioneert. Deze test zal bestaan uit een checklist met alle mogelijkheden voor de gebruiker.

### 7.1.5 Pilotplan

De komende pilot is een onderdeel van een verzameling pilots die uiteindelijk tezamen het volledige project vormen. Voor de toekomst staan de volgende pilots in de planning:

Pilot	Te ontwikkelen onderdeel
3 <sup>e</sup> pilot	Games
4 <sup>e</sup> pilot	Transactions, Cashier & Financial
5 <sup>e</sup> pilot	Reports & fraud control



## 7.2 Plan van Aanpak 2<sup>e</sup> pilot

### 7.2.1 Bevestiging reikwijdte

Van de spelers, affiliates en managers moet alle informatie getoond kunnen worden, zolang dit geen overlap heeft met andere pilots. Een mogelijke overlap zou bijvoorbeeld tussen affiliate en de financial pilot kunnen zijn. Het doel van deze pilot is vooral het implementeren van de GUI, de weergave van de data en het uitbreiden van de implementatie van de personen.

### 7.2.2 Op te leveren onderdelen

In deze fase dienen de volgende resultaten te worden opgeleverd;

- Aangepaste versie van softwareontwerp van backend, GMS en GUI; met functionaliteiten zoals beschreven in de systeemeisen van de 2<sup>e</sup> pilot. Het softwareontwerp zal later worden uitgebreid met de andere functionaliteiten.
- Implementatie van backend interface, GMS en GUI; zoals beschreven in de systeemeisen voor pilot 2.
- Testrapport van de ontwikkelde software.

### 7.2.3 Kwaliteitszorg

De kwaliteit in deze pilot zal bewaakt worden door rekening te houden met een drietal kwaliteitskenmerken. Tijdens het ontwerpen en implementeren dient rekening gehouden te worden met deze punten, die in de testfase gecontroleerd zullen worden.

- Bruikbaarheid
- Dekkingsgraad (ondersteuning van functies uit beide backends)
- Aanpasbaarheid (mogelijkheid voor toevoegen van andere backends of functies)

### 7.2.4 Tijdschatting

Totale tijdschatting voor deze pilot is 3 weken;

- Definitiestudie en plan van aanpak (2 dagen)
- Softwareontwerp van de te nieuwe functionaliteiten (2 dagen)
- Implementatie van nieuwe functionaliteiten (6 dagen)
- Testplan en testrapport van de ontwikkelde software (5 dagen)



### 7.2.5 Faseplanning

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
5	8-3-2010	Pilot 2; Persons & login DB		2
6	15-3-2010	Testen pilot 2	Testrapport	2
7	22-3-2010	Pilot 3; Games & Tournaments		3
8	29-3-2010	Testen pilot 3	Testrapport	3
9	5-4-2010	Pilot 4: Financial		4
10	12-4-2010	Vervolg vorige week		4
11	19-4-2010	Vervolg en testen pilot 5	Testrapport	4
12	26-4-2010	Pilot 5: Reports & Fraud control		5
13	3-5-2010	Vervolg vorige week & testen pilot 5	Testrapport	5
14	10-5-2010	Uitloop / extra functionaliteiten	Testrapport	6
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	

*Uit: Projectplanning (versie 3)*



## 7.3 Opzetten testomgeving 2<sup>e</sup> pilot

Zoals beschreven in het testplan zullen er ter afsluiting van deze pilot validatietests en unittests worden uitgevoerd. Dit hoofdstuk beschrijft de opgestelde tests van de 2<sup>e</sup> pilot, maar dit proces is bij iedere pilot uitgevoerd.

### 7.3.1 Opstellen validatietests

Met een validatietest wordt getest of een geïmplementeerd onderdeel ook naar behoren werkt. Deze tests zijn dan ook indirect afkomstig van de eisen van de pilot en de functionele eisen. Met behulp van een validatietest wordt aangetoond of invoer en uitvoer van bepaalde objecten identiek zijn, en of het systeem foutieve invoer op een gewenste manier afvangt. Met deze tests zal getracht worden alle mogelijke functionaliteiten te behandelen die in deze pilot zijn gemaakt.

Voorbeelden van enkele validatietests<sup>1</sup>:

#### Inlogscherf

- Inloggen zonder gebruikersnaam en wachtwoord
- Inloggen met enkel bestaande gebruikersnaam
- Inloggen met enkel bestaand wachtwoord
- Inloggen met onjuiste gebruikersnaam en wachtwoord
- Inloggen met juiste gebruikersnaam, onjuist wachtwoord
- Inloggen met onjuiste gebruikersnaam, juist wachtwoord
- Inloggen met juiste gebruikersnaam en wachtwoord

#### Spelersoverzicht - Selecteren speler

- Selecteer de eerste speler, vergelijk datatabel gegevens met detailgegevens
- Selecteer de laatste speler, vergelijk datatabel gegevens met detailgegevens
- Selecteer een speler in het midden van de lijst, vergelijk datatabel gegevens met detailgegevens
- Filter spelers met behulp van alle kolomfilters

#### Spelersoverzicht - Bewerken van een speler

1. Klik knop voor bewerken van speler
2. Vul nieuwe spelersdata in, of verwijder spelersdata
3. Sla speler op
4. Verifieer nieuwe data in speleroverzicht

#### Winnersoverzicht - Selecteren speler

- Selecteer de eerste speler, vergelijk datatabel gegevens met detailgegevens
- Selecteer de laatste speler, vergelijk datatabel gegevens met detailgegevens
- Selecteer een speler in het midden van de lijst, vergelijk datatabel gegevens met detailgegevens

---

<sup>1</sup> Zie het pilot testrapport voor het volledige overzicht



### 7.3.2 Testopzet unit tests

Unittesten is een methode om stukken code (units) afzonderlijk te testen op een correcte werking. Bij unittesten zal voor iedere unit een test ontwikkeld worden. Hierbij worden dan verschillende testcases doorlopen<sup>2</sup>.

#### 7.3.2.1 Tests

Het Game Management Systeem is momenteel een soort MVC (Model View Controller), waarbij de backends voor de belangrijke translatie zorgen tussen de goksoftware en de webUI. Er is dan ook gekozen om de backends te Unittesten, en de webUI buiten beschouwing te laten. De webUI zorgt enkel voor de weergave van de resultaten uit de backends.

```
1 public abstract class BackendPlayerTest
2 {
3
4     protected GameManagementService gms;
5     protected Backend backend;
6     protected static int staticId;
7
8     public void setUp() throws Exception
9     {
10         gms = new GameManagementService();
11     }
12
13     @Test
14     public void createPlayer() throws Exception
15     {
16         ...
17     }
18
19     @Test
20     public void playerRestrictionList() throws Exception
21     {
22         ...
23     }
24
25     @Test
26     public void deletePlayer() throws Exception
27     {
28         ...
29     }
30
31     @After
32     public void tearDown() throws Exception
33     {
34         ...
35     }
36
37     public Player createRandomPlayer() throws IllegalAccessException,
38         InvocationTargetException, NoSuchMethodException
39     {
40         ...
41     }
42
43     Code continues...
```

<sup>2</sup> Vrij naar: Wikipedia, Unit Testing



```

44     private void checkMethodExistence(String classType, String method)
45     {
46         try
47         {
48             // Suppose method is in lowercase, create camelcase
49             // for testmethod (e.g. 'classtypeMethod')
50             method = classType + method.substring(0, 1).toUpperCase()
51                 + method.substring(1);
52
53             // Check if this file has a testmethod for the object's
54             // property
55             this.getClass().getMethod(method);
56
57         } catch (NoSuchMethodException ex)
58         {
59             // No testmethod found
60             System.err.println("WARNING:\tTestmethod '" + method
61                 + "' doesn't exist. Please create testmethod.");
62         }
63     }
64 }

```

**Codevoorbeeld 7-1: Ingekorte Unit test van de Backend PlayerCollection**

Er zijn 3 testmethoden; *createPlayer()*, *playerRestrictionList()* en *removePlayer()* (regels 13, 19 & 25). De namen en functies van deze tests spreken voor zich. Met de functie *createRandomPlayer()* kan een spelerobject worden aangemaakt, die gevuld is met willekeurige data. Aangezien iedere test een eigen scope heeft, worden na iedere test het playerobject weggegooid. Door de identificatiecode van de player op te slaan in een static variabele (*protected static int staticId*), kan deze player weer worden opgevraagd uit de onderliggende backend. Zo kan voor de hele testset dezelfde speler worden gebruikt. Deze test wordt voor de casinobackend en het pokerbackend uitgevoerd (zie Codevoorbeeld 7-2).

```

1  public class CasinoBackendPlayerTest extends BackendPlayerTest
2  {
3
4      @Before
5      @Override
6      public void setUp() throws Exception
7      {
8          super.setUp();
9
10         String jndiGoldAlias = "jndigold";
11         String jndiGoldStatAlias = "jndigoldstat";
12
13         backend = new CasinoBackend(jndiGoldAlias, jndiGoldStatAlias);
14         gms.addBackend(backend);
15         assertEquals("Adding PokerBackend", 1,
16             gms.getNumberOfBackends());
17     }
18 }

```

**Codevoorbeeld 7-2: Unit test van de PlayerCollection voor de Casinosoftware. Deze erft functies over van de BackendPlayerTest (Codevoorbeeld 7-1).**

### 7.3.3 Zelf-controlerende tests

De tests zijn zelf-controlerend gemaakt; van het `gag.gms.backend.collections.persons.Player` object wordt opgevraagd welke getters en setters er beschikbaar zijn, welke vervolgens één voor één worden getest op consistentie. Dit betekent dus ook dat als er aan het `Player` object een nieuwe getter of setter wordt toegevoegd (variabele), er automatisch een test voor wordt gemaakt. Op deze manier is de Unit test ook in de toekomst te gebruiken en hoeft deze niet aangepast te worden als de software is bijgewerkt.

In het geval van Strings, Integers, Booleans, Shorts of Doubles is het mogelijk om waarden te vergelijken door de objecten om te zetten naar `toString().equals(...)`. In het geval van objecten (zoals de `restrictionList`) is dit moeilijker te testen. Als een onbekende klasse is gedetecteerd wordt er gekeken of er een testmethode is aangemaakt voor deze variabele, met de methode `checkMethodExistence()`, zoals te zien in Codevoorbeeld 7-1; regel 37-56. Op regel 43 en regel 48 van deze methode wordt gecontroleerd of de testmethode bestaat voor bijvoorbeeld `registrationDate`, (methodenaam `playerRegistrationdate`).





## 7.4 Pilotdocument 2<sup>e</sup> pilot

Dit hoofdstuk beschrijft welke onderdelen er zijn geïmplementeerd in deze pilot<sup>3</sup>.

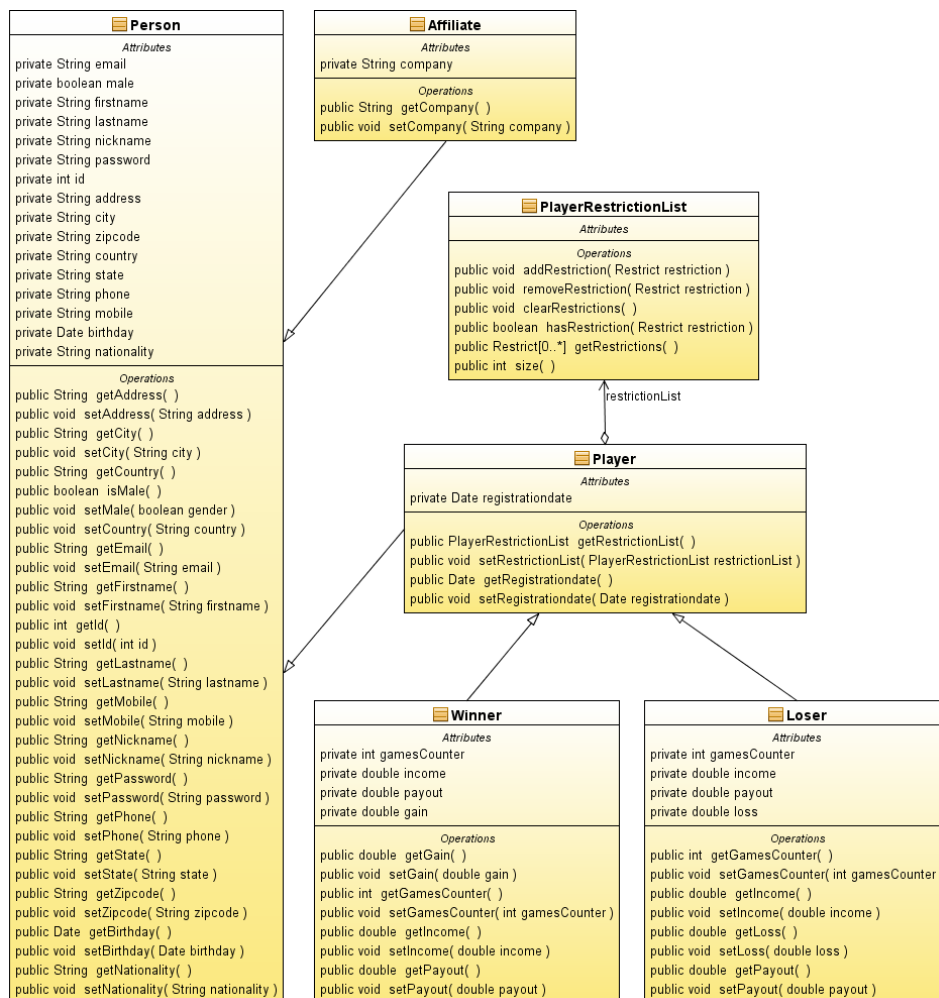
### 7.4.1 Implementatie GMS backendadapters

Iedere backendadapter heeft een verzameling spelers (PlayerCollection) en een verzameling affiliates (AffiliateCollection). Deze collecties zijn aan te roepen vanuit de respectievelijke backend. De in- en uitkomende objecten zijn onderling uitwisselbaar, de implementatie is uiteraard verschillend.

#### 7.4.1.1 Abstractie van spelers en affiliates

De speler, winnende speler, verliezende speler en affiliate erven allemaal de persoonsvariabelen van de klasse Persoon. Deze objecten worden gevuld in de backendadapters en worden in de webinterface gebruikt voor de weergave van persoonsgegevens.

*Zie de digitale versie voor een duidelijkere weergave*



Figuur 7-1: Soorten personen in de Game Management Service

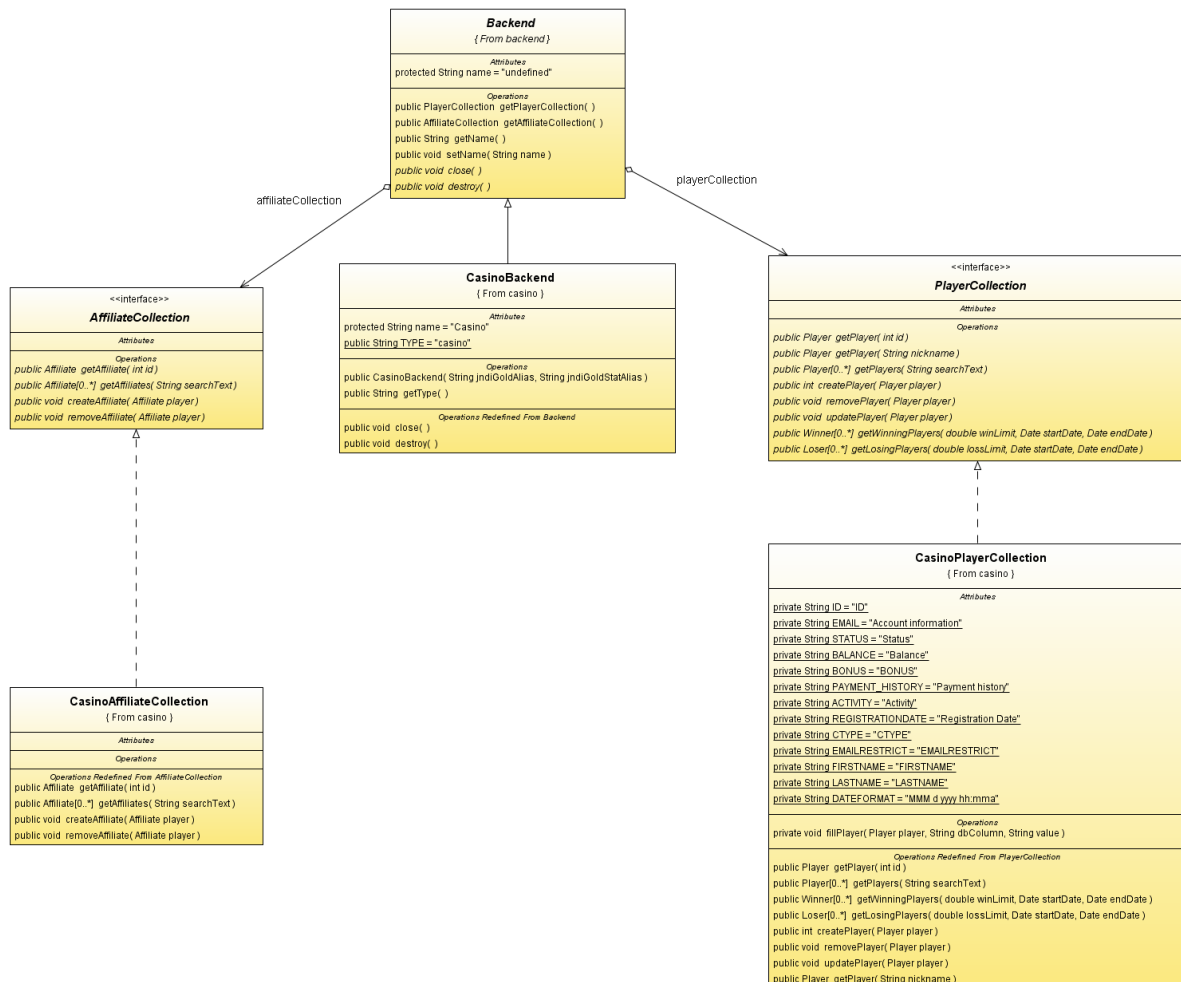
<sup>3</sup> Zie het pilotdocument voor meer informatie



### 7.4.1.2 Implementatie Casino backendadapter

De backendadapter van het casino maakt voor het ophalen van data gebruik van de Casinosoftware, zoals in Codevoorbeeld 5-1 is voorgedaan. In Figuur 7-2 is het klassendiagram te zien van deze adapter.

*Zie de digitale versie voor een duidelijkere weergave*



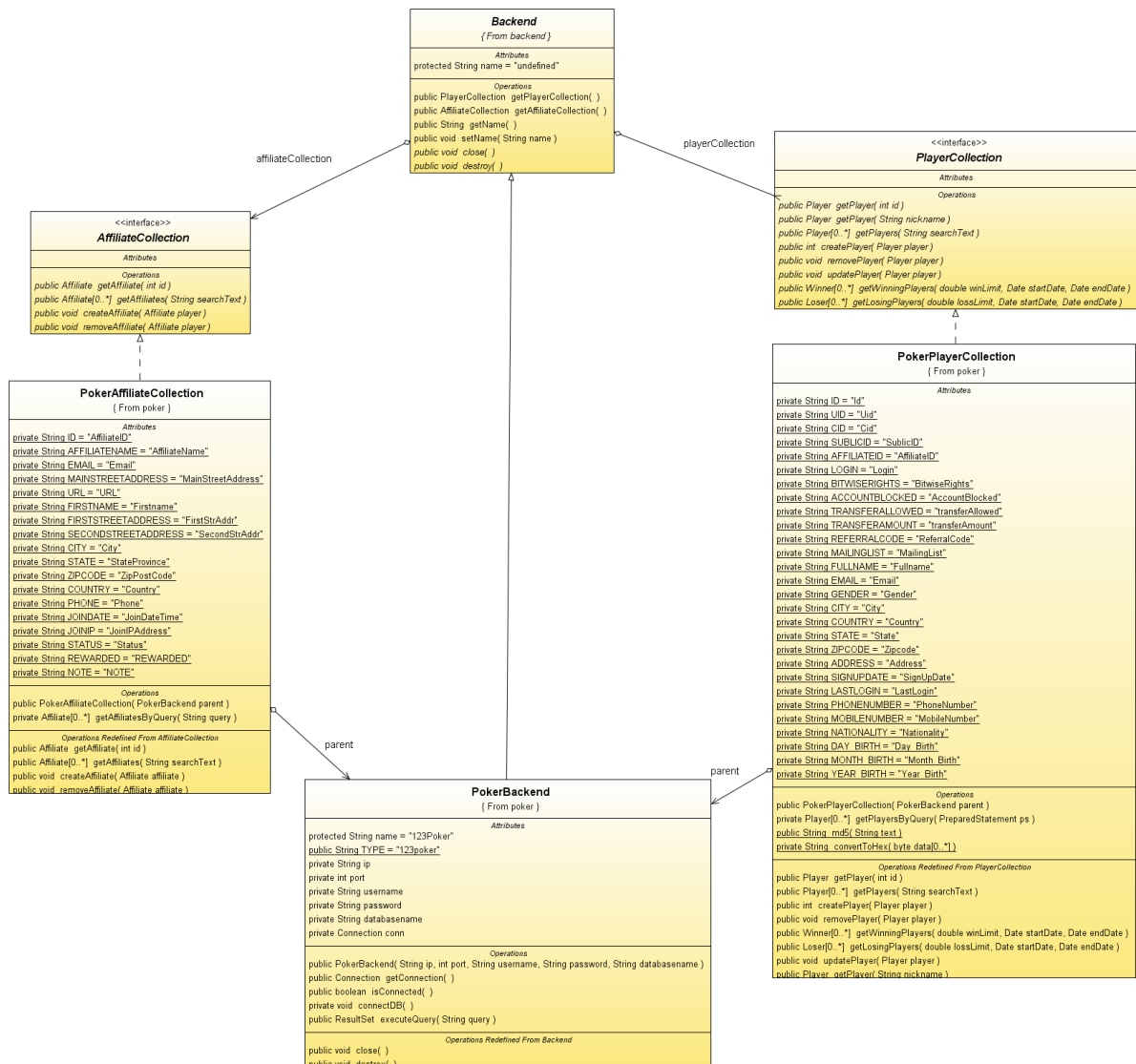
**Figuur 7-2: Klassendiagram backendadapter voor de casinosoftware**



### 7.4.1.3 Implementatie Poker backendadapter

De pokersoftware is gemaakt in C#. Om snelheid van gegevensoverdracht te behouden is besloten de data niet op te halen uit het pokerbackend, maar direct uit de Microsoft SQL Server. In dit geval onderhoudt de PokerBackend(adapter) de verbinding met deze database en maken de PlayerCollection en AffiliateCollection hier gebruik van.

*Zie de digitale versie voor een duidelijkere weergave*



Figuur 7-3: Klassendiagram backendadapter voor de pokersoftware



## 7.4.2 Implementatie gebruikersomgeving

### 7.4.2.1 Gebruik van pagina template

Er is een template gemaakt, welke gebruikt kan worden voor de pagina's in de GMS webinterface. Deze template bevat momenteel de locaties van de titel, body en sidebar. Daarnaast is de menubalk en eerste versie van de issuetracker aangemaakt in deze template, zodat deze vanaf iedere pagina bereikbaar is.

In Codevoorbeeld 7-3 is een deel te zien uit het bestand template.xhtml, met daarin aangegeven waar de verschillende contentblokken moeten worden ingevoegd. Codevoorbeeld 7-4 toont de players pagina, welke gebruik maakt van deze template. Voor de eenvoud is de code in de codevoorbeelden ingekort, zie regelnummers voor de locaties.

```
...
113 <tr>
114     <td class="bodycontent">
115         <s:div id="bodycontent">
116             <ui:insert name="uibody">No body defined</ui:insert>
117         </s:div>
118     </td>
119     <td class="sidebarcontent">
120         <s:div id="sidebarcontent">
121             <ui:insert name="uisidebar">No sidebar defined</ui:insert>
122         </s:div>
123     </td>
124 </tr>
...
```

Codevoorbeeld 7-3: Placeholders van de body en sidebar content in template.xhtml

```
...
10 <body>
11     <ui:composition template="../template.xhtml">
12         <ui:define name="uititle">
13             <h:outputText value="#{messages['players.title']}" />
14         </ui:define>
15
16         <ui:define name="uibody">
17             -- Page content --
130        </ui:define>
131        <ui:define name="uisidebar">
174            -- Widgets code --
175        </ui:define>
176    </ui:composition>
...
176 </body>
...
```

Codevoorbeeld 7-4: Voorbeelddefinities in players.xhtml, gebruikmakend van de template uit Codevoorbeeld 7-3



Issues

R3fr 35h 109in

Speleroverzicht

Logoff

Ingelogd als admin

Overzicht

Problemen

Uitbetalingen

Spellen

Spelers

Instellingen

Samenvatting

Deposits

Actieve spelers

Spelsessies

Winnende speler

Probleemrapport

123PokerPlayer Nederland

ID	Voornaam	Achternaam	E-mail	Ikcknaam	Registratiedatum
140	bob		bob@masterpay.co	sihaya	2008-03-31 16:25
141	bob		sihaya@stredis.nl	ayahis	2008-03-31 16:34
142	bob		blackhole@stredis.nl	ayahis	2008-03-31 16:35
143	Philip		philip@gamingandg	poker123	2008-04-01 06:10
144	Lori		lori@masterpay.co	Lori	2008-04-01 06:42
145	chengde		chengde@masterp	chengdeyang	2008-04-01 08:42
146	Niels		spam.flip@gmail.cc	Flip182	2008-04-01 11:32
147	Aanmelden		lori@masterpay.co	test1	2008-04-01 16:15
148	Marco		marco@masterpay	marco123	2008-04-01 16:16
149	marco		marco@tikies.com	marco123456	2008-04-01 17:01
150	Bob		sihaya@stredis.nl	muaddib	2008-04-01 18:40
151	Gertjan		gertjan@gamingan	gertjan	2010-02-15 15:53
155	firstname junittest_		email junittest_3147	nickname junittest_	2010-03-21 23:47
156	firstname junittest_		email junittest_325c	nickname junittest_	2010-03-21 23:48

1

2

3

Speler zoeken

Zoeken

2

Speler details

Adres

Steenplaatsstraat 6

??person.birthday???

Stad

Rijswijk

Land

Nederland

E-mail

lori@masterpay.com

Voornaam

Lori

ID

144

Achternaam

??person.male???

true

Mobiel

0031648332072

??person.nationality???

Ikcknaam

Lori

Telefoon

003170388699

Registratiedatum

2008-04-01 08:42

Restricties

transfer

Staat Provincie

Zuid Holland

Postcode

2288AA

Bewerten

Gaming and Gambling 2010

Figuur 7-4: Speleroverzicht van de pokerbackend

### 7.4.2.2 Sidebar voor widgets

Tijdens het samenstellen van de task map en abstracte prototypes (zie document Systeemontwerp) bleek dat de acties op een pagina te verdelen zijn in een hoofdactie en verschillende kleine subacties. Een voorbeeld van een hoofdactie is bijvoorbeeld het tonen van een spelerlijst (zie Figuur 7-4:1), een subactie is bijvoorbeeld een zoekbalk of een detailoverzicht van en geselecteerde speler. Er is bedacht om deze sub-acties te plaatsen in een zijbalk (zie Figuur 7-4:2), waarin iedere sub-actie een eigen venstertje krijgt. Deze sub-acties worden *widgets* genoemd. Pagina's kunnen globale widgets of pagina-specifieke widgets hebben. De widgets worden op dit moment gespecificeerd door gebruik te maken van de invoegmogelijkheden van de template.

### 7.4.2.3 Toolbar

Er is een menubalk (zie Figuur 7-4:3) gemaakt waarmee de verschillende pagina's kunnen worden geopend. De menubalk is verdeeld in vijf hoofdgroepen; Overview, Issues, Cashouts, Players, Games en Settings. De pagina's uit deze hoofdgroepen zijn verdeeld over de onderliggende toolbar.

Met behulp van ajax-requests wordt deze onderste toolbar geladen, afhankelijk van de aangeklikte hoofdgroep. Het is voor de gebruiker ook mogelijk om deze selectie automatisch te laten verlopen door middel van mouse-over (zie 7.4.2.5)



#### 7.4.2.4 Manager login

Het is mogelijk om beheerderaccounts aan te maken, waaraan backends kunnen worden toegewezen. De beheerder kan inloggen bij de Game Management Service en zijn eigen backends onderhouden. In Figuur 7-5:1 is de lijst te zien met beheerders en de toegankelijke backends voor deze personen.

The screenshot displays the 'Manager Overview' page. At the top, there's a header with 'Issues' and a notification 'New manager 'gertjan' added'. Below this is a green bar with the title '???managers.title???' and a 'Logoff' button. A navigation bar contains tabs: 'Overzicht', 'Problemen', 'Uitbetalingen', 'Spellen', 'Spelers', and 'Instellingen'. Below this is a sub-navigation bar with 'Algemeen' and 'Managers'. The main content area shows a table with columns 'Gebruikersnaam' and 'Toegankelijke backends'. The table lists 'admin' and 'gertjan' as users, with '123PokerPlayer Nederland, Casino Nederland' as the accessible backend for 'gertjan'. A sidebar on the right contains a 'Nieuwe manager' form (labeled 2) and a 'Manager bewerken' form (labeled 3). The sidebar also shows '123PokerPlayer Nederland' and 'Gaming and Gambling 2010'.

Figuur 7-5: Manageroverzicht; aanmaken van beheerders en toekennen van backends

#### 7.4.2.5 Instellingen

Een beheerder heeft verschillende configuratieopties. Er zijn momenteel zes configureerbare onderdelen. De instellingen worden alleen opgeslagen in de database als ze afwijken van de aanbevolen voorinstelling. Dit bespaart database opslag. De voorinstellingen zijn opgeslagen in een properties file.

The screenshot shows the 'Manager Settings' page. The 'Managers' tab is selected. The settings are as follows:

Setting	Value
Thema	emeraldTown
Taal	Nederlands (Nederland)
Datumformaat	yyyy-MM-dd HH:mm
Automatische toolbarselectie	Nee
Eerste dag van de week (0 = zondag)	1
???settings.general.winningCalendarRange???	21

A 'Toepassen' button is located at the bottom right of the settings area.

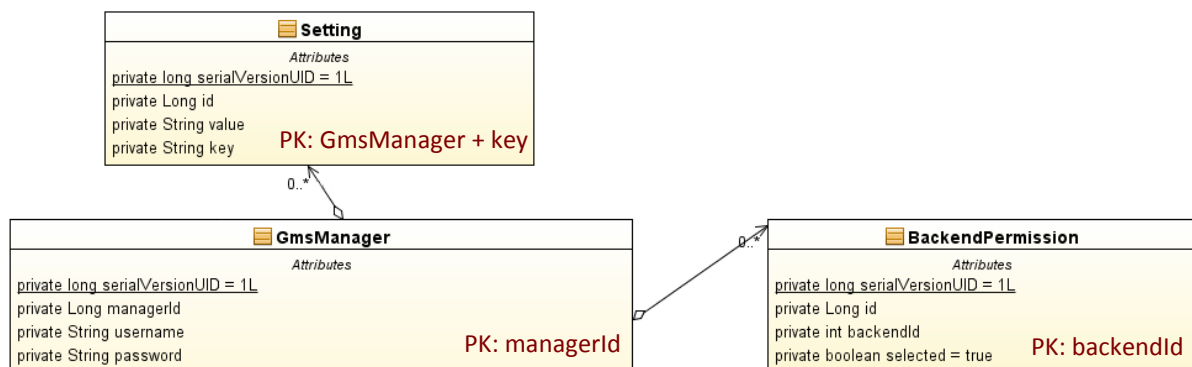
Figuur 7-6: Instellingen per beheerder



### 7.4.2.6 Hibernate

Voor de opslag van de beheerders, hun rollen en instellingen is een MySQL database aangemaakt. Een manager kan toegang hebben tot één of meerdere backends en één of meerdere instellingen hebben opgeslagen. Met behulp van de ORM-oplossing (Object/Relational Mapping) Hibernate kunnen deze tabellen aangeroepen worden als gelinkte java objecten, wat de code voor het toevoegen van instellingen en ophalen van backendpermissies schoner (korter) houdt dan als de data opgehaald zou moeten worden met behulp van sql queries.

Het datamodel van deze database is te zien in Figuur 7-7. Er is gekozen voor MySQL omdat dit een vrij beschikbare lichtgewicht database is. Deze database draait momenteel op de ontwikkelpc, in de toekomst kan deze op een server worden gedraaid. Omdat Hibernate de queries maakt en afhandelt, is het ook mogelijk om een andere database te gebruiken zoals Derby, Postgres of Sybase.



**Figuur 7-7: Datamodel van de MySQL database waarin de inloggegevens, instellingen en toegankelijke backends van de beheerders worden opgeslagen**

Hibernate creëert de databasetabellen automatisch aan de hand van de object-gerelateerde klassen. In Codevoorbeeld 7-5 is te zien hoe in de klasse GmsManager de relaties met de instellingen en backendpermissies wordt gedefinieerd. Er wordt hierbij gebruik gemaakt van annotaties, welke aangeven waar een variabele voor gebruikt moet worden.

```
...
28 @Column(nullable = false)
29 private String password;
30 @OneToMany(mappedBy = "gmsManager", cascade = CascadeType.ALL)
31 private Collection<Setting> settings;
32 @OneToMany(mappedBy = "gmsManager", cascade = CascadeType.ALL)
33 private Collection<BackendPermission> backendPermissions;
...
```

**Codevoorbeeld 7-5: Definieren van de Hibernate klasse GmsManager**

De backendPermission van een manager verwijst naar een backendId. Met behulp van deze id kan in een configuratiebestand opgezocht worden wat voor backend het betreft en wat de verbindingsparameters zijn. Dit wordt door een BackendFactory (factory design pattern) afgehandeld. Via hibernate wordt de toegestane backendId opgehaald, waarmee de backendFactory verbinding maakt met de achterliggende software.



### 7.4.2.7 Aanmaken van managers

Met behulp van een widget (zie Figuur 7-5:2) kunnen managers aangemaakt worden. Door op *bewerken* te klikken in de details-widget zullen in een later stadium de backendpermissies worden toegewezen.

### 7.4.2.8 Localization

Het de gehele gebruikersinterface is opgebouwd met localized teksten. Dit betekent dat per taal de teksten staan opgeslagen in één enkel bestand. Er is op dit moment een bestand *messages\_en\_US.properties* en een *messages\_nl\_NL.properties* geconfigureerd. Zoals in sommige screenshots te zien is zijn niet alle teksten vertaald. Dit is echter een eenvoudige klus die gedaan kan worden als de indeling van de gebruikersomgeving goedgekeurd is door de opdrachtgever.

```
...
70 affiliates.title.details=Affiliate details
71 winners.title=Winnende spelers
72 losers.title=Verliezende spelers
73 players.title=Speleroverzicht
74 affiliates.title=Affiliates
75 loading=Laden...
...
```

Codevoorbeeld 7-6: Deel van het vertalingsbestand *messages\_nl\_NL.properties*

```
1 <h:outputText value="#{messages['players.title']}"/>
```

Codevoorbeeld 7-7: Aanroepen van een tekst uit het geselecteerde vertaalbestand



#### 7.4.2.9 Speleroverzicht

Van beide backends is de lijst op te vragen met aangemelde spelers. In Figuur 7-4 is de lijst met spelers uit de pokersoftware te zien. Deze lijst is opgedeeld in meerdere pagina's, zodat er bij een grote spelerdatabase geen lange tabel gemaakt hoeft te worden. Dit bevordert de laadtijd van de pagina.

Op deze pagina staat een zoekwidget, waarmee gezocht kan worden op email en naam van een speler. De widget eronder bevat de details van een geselecteerde speler. Deze widget zal later ook de balans en eventuele conversaties met de speler tonen. De gebruiker kan ook in de kolommen een zoekwoord typen, zodat de spelerlijst gefilterd wordt.

#### 7.4.2.10 Winnende en verliezende spelers

De pagina's voor winnende en verliezende spelers hebben verschillende informatie, maar de indeling is identiek. Op deze pagina's kunnen de winnende / verliezende spelers getoond worden. Met behulp van een kalenderwidget kan een datumbereik worden opgegeven waarvan de spelers getoond moeten worden. Het aantal dagen dat teruggekeken moet worden is ook configureerbaar in de instellingen van de beheerder. Een beheerder kan op deze manier bijvoorbeeld de spelers van de afgelopen drie weken bekijken of die van de eerste x dagen van maand y.

Boven de kalenderwidget is een widget met een schuifbalk om de hoeveelheid geldwinst (/verlies) te bepalen. Dit is het bedrag dat spelers minimaal gewonnen moeten hebben om in de lijst voor te komen.

#### 7.4.2.11 Affiliateoverzicht

Op deze pagina staat een overzicht van de affiliates. Het is mogelijk om van deze affiliates de details te bekijken. In een opvolgende pilot zal deze pagina worden uitgebreid met de mogelijkheid tot starten van conversaties en toewijzen van bonusprogramma's.



## 7.5 Testrapport 2<sup>e</sup> pilot

### 7.5.1 Resultaten validatietests

Niet alle onderdelen van deze pilot zijn afgerond, omdat ze afhankelijk zijn van een andere pilot. Deze functionaliteiten zullen daarom negatief uit de validatietest komen. Er is besloten om deze onderdelen te testen, zodat er een duidelijk overzicht kan worden gemaakt van de onvoltooide en dus nog te implementeren onderdelen.

De validatietest bestond uit 44 punten waarop het systeem te testen is. Een voorbeeld hiervan is of een speler gezocht kan worden of dat een beheerder kan inloggen. Van de 38% van deze punten kwamen negatief uit de test; deze onderdelen werken (nog) niet. In tabel 7-1 is een opsomming van deze punten te zien. Deze zullen worden meegenomen bij het definiëren van de eisen van de 3<sup>e</sup> pilot.

Spelersoverzicht	Opmerkingen	Geslaagd
Zoeken op willekeurige persoon	Case sensitive bij Casino, error bij Poker ( <i>Incorrect syntax near '@P1'</i> )	NEE
Selecteer de eerste speler, vergelijk datatabel gegevens met detailgegevens	Bij Casino is er één uur tijdsverschil tussen registratiedatum in de tabel en in de details, bij Poker is er twee uur tijdsverschil	NEE
Winnersoverzicht	Opmerkingen	Geslaagd
Selecteer de eerste speler, vergelijk datatabel gegevens met detailgegevens	Registratiedatum mist, waarschijnlijk extend van Person ipv Player; onvolledige details.	NEE
Filter winlimiet van spelers (min, max en random waarde) en controleer uitvoer	<i>Winlimiet blijkt een 'hoger dan' te zijn ipv een 'hoger of gelijk dan' (backend implementatie)</i>	NEE
Verliezersoverzicht	Opmerkingen	Geslaagd
Selecteer de eerste speler, vergelijk datatabel gegevens met detailgegevens	Registratiedatum mist, waarschijnlijk extend van Person ipv Player; onvolledige details.	NEE
Filter verlieslimiet van spelers (min, max en random waarde) en controleer uitvoer	<i>Losslimiet blijkt een 'hoger dan' te zijn ipv een 'hoger of gelijk dan' (backend implementatie)</i>	NEE
Maak nieuwe manager aan	Werkt niet als de username al bezet is...	NEE
Bewerk manager	Opslaan niet mogelijk (known feature)	NEE

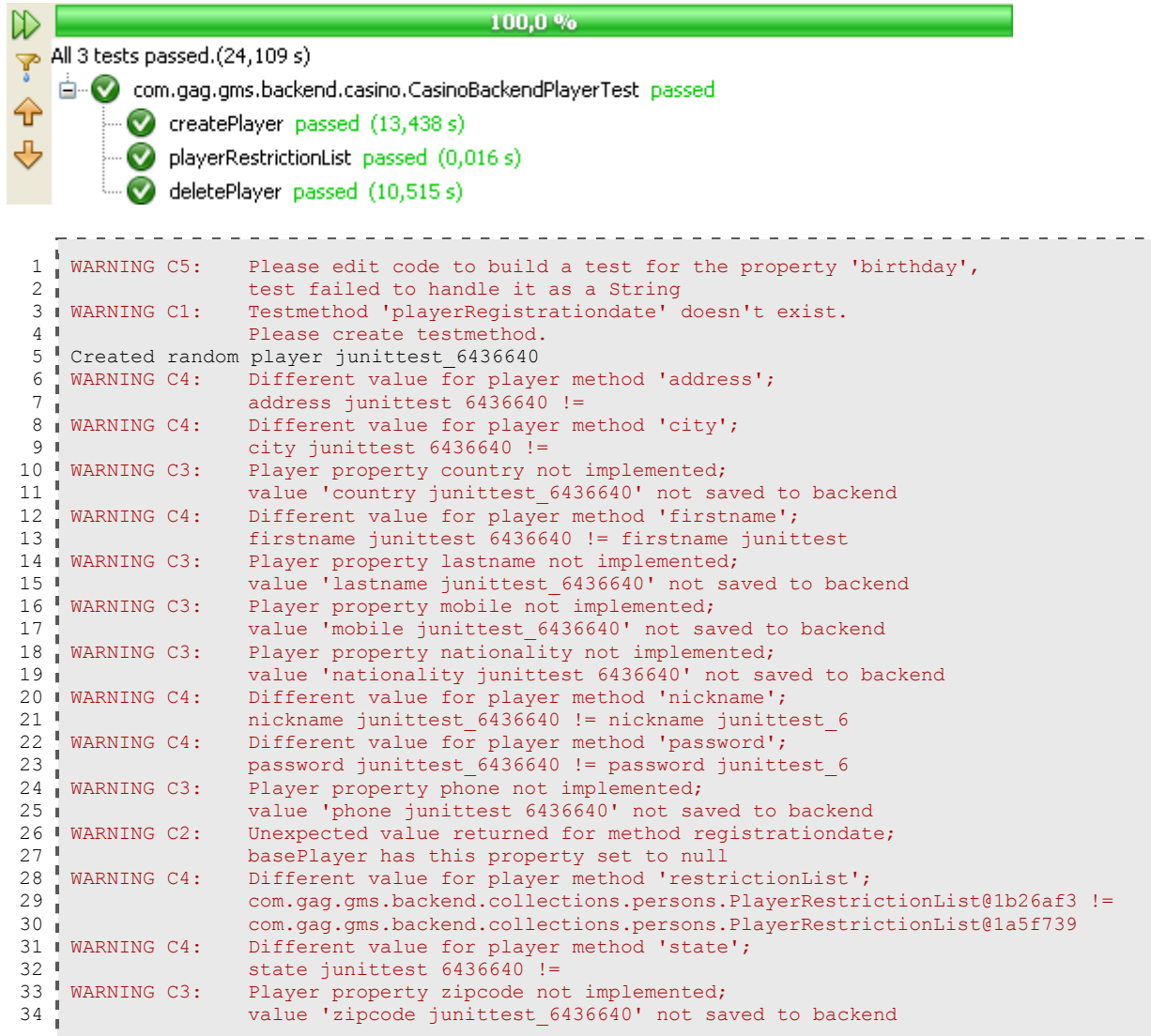
Tabel 7-1: Belangrijkste resultaten van de validatietests met negatieve uitkomst. Zie voor volledige resultatenlijst het testrapport



## 7.5.2 Resultaten unit tests Casino backend

### 7.5.2.1 Uitvoeren tests

De unit tests voor het casinobackend zijn succesvol doorlopen, ondanks dat er een aantal implementatiefouten in zitten. Dit betekent dat sommige waarden niet goed worden opgeslagen of worden opgehaald. Er is gekozen om de Unit tests een warning te laten geven, in plaats van een error, zodat de deletePlayer() test kon worden uitgevoerd. Ook kan er op deze manier een volledige lijst gegeven worden van welke variabelen nog moet worden geïmplementeerd<sup>4</sup>.



```
1 WARNING C5: Please edit code to build a test for the property 'birthday',
2 test failed to handle it as a String
3 WARNING C1: Testmethod 'playerRegistrationdate' doesn't exist.
4 Please create testmethod.
5 Created random player junittest_6436640
6 WARNING C4: Different value for player method 'address';
7 address junittest_6436640 !=
8 WARNING C4: Different value for player method 'city';
9 city junittest_6436640 !=
10 WARNING C3: Player property country not implemented;
11 value 'country junittest_6436640' not saved to backend
12 WARNING C4: Different value for player method 'firstname';
13 firstname junittest_6436640 != firstname junittest
14 WARNING C3: Player property lastname not implemented;
15 value 'lastname junittest_6436640' not saved to backend
16 WARNING C3: Player property mobile not implemented;
17 value 'mobile junittest_6436640' not saved to backend
18 WARNING C3: Player property nationality not implemented;
19 value 'nationality junittest_6436640' not saved to backend
20 WARNING C4: Different value for player method 'nickname';
21 nickname junittest_6436640 != nickname junittest_6
22 WARNING C4: Different value for player method 'password';
23 password junittest_6436640 != password junittest_6
24 WARNING C3: Player property phone not implemented;
25 value 'phone junittest_6436640' not saved to backend
26 WARNING C2: Unexpected value returned for method registrationdate;
27 basePlayer has this property set to null
28 WARNING C4: Different value for player method 'restrictionList';
29 com.gag.gms.backend.collections.persons.PlayerRestrictionList@1b26af3 !=
30 com.gag.gms.backend.collections.persons.PlayerRestrictionList@1a5f739
31 WARNING C4: Different value for player method 'state';
32 state junittest_6436640 !=
33 WARNING C3: Player property zipcode not implemented;
34 value 'zipcode junittest_6436640' not saved to backend
```

System output 7-1: Waarschuwingen bij de Unit tests van het Casino backend (ingekort, enkel Unit warnings en infos)

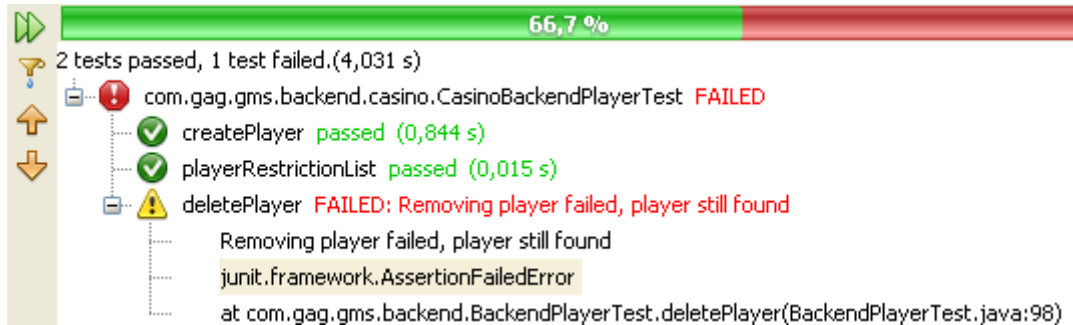
<sup>4</sup> Zie het testrapport voor de conclusie van de casino unittest



## 7.5.3 Resultaten unit tests Poker backend

### 7.5.3.1 Uitvoeren tests

De unit test voor het poker backend gaf een foutmelding bij het verwijderen van de zojuist aangemaakte speler, zoals te zien in onderstaande screenshot.



Er bleek in de backendcode een klein foutje te zitten; de functie `getPlayer(int id)` gaf altijd een Player object terug. Als er geen player was gevonden in de onderliggende pokersoftware (in dit geval de Microsoft SQL database), werd dit object niet gevuld, maar wel gereturned. De code is aangepast; geen speler met de opgegeven ID gevonden == return null.



Er zijn wel enkele waarschuwingen<sup>5</sup>:

```
1  WARNING C5:    Please edit code to build a test for the property 'birthday',
2                  test failed to handle it as a String
3  Created random player junittest_7295500
4  WARNING C1:    Testmethod 'playerRegistrationdate' doesn't exist.
5                  Please create testmethod.
6  WARNING C4:    Different value for player method 'firstname';
7                  firstname junittest_7295500 !=
8                  firstname junittest_7295500 lastname junittest_7295500
9  WARNING C3:    Player property lastname not implemented;
10                 value 'lastname junittest_7295500' not saved to backend
11  WARNING C3:    Player property nationality not implemented;
12                 value 'nationality junittest_7295500' not saved to backend
13  WARNING C3:    Player property password not implemented;
14                 value 'password junittest_7295500' not saved to backend
15  WARNING C2:    Unexpected value returned for method registrationdate;
16                 basePlayer has this property set to null
17  WARNING C4:    Different value for player method 'restrictionList';
18                 com.gag.gms.backend.collections.persons.PlayerRestrictionList@199f91c !=
19                 com.gag.gms.backend.collections.persons.PlayerRestrictionList@1df8b99
20
```

System output 7-2: Waarschuwingen bij de Unit tests van het Poker backend (ingekort, enkel Unit warnings en infos)

<sup>5</sup> Zie het testrapport voor de conclusie van de poker unittest



## 8 3<sup>e</sup> Pilot: Uitbreiden spelerdetails

### 8.1 Definitiestudie en Plan van Aanpak 3<sup>e</sup> pilot

De pilot start met het opstellen van de doelen van de pilot en het plannen van de werkzaamheden. Dit hoofdstuk beschrijft de belangrijkste onderdelen van de definitiestudie en het plan van aanpak<sup>6</sup>.

Met dank aan de werkzaamheden in de vorige pilots is het mogelijk om spelerdata uit de casino en pokersoftware op te halen en weer te geven. Met het oog op de ondersteuning en het monitoren van de klanten zal in deze pilot een uitbreiding moeten komen op het reeds ontwikkelde speleroverzicht. Het is mogelijk om de spelers weer te geven, te zoeken en de details zoals NAW weer te geven. Voor verschillende beheerdersrollen is het nodig om meer informatie van deze personen in te zien, zoals de laatste gespeelde casinospellen, laatste aanmelding en een overzicht van financiële zaken zoals uitbetalingen en stortingen.

Met deze selectiemogelijkheden kan de validiteit van gewonnen geld worden onderzocht en kunnen eventuele betalingsproblemen worden onderzocht. In het systeemontwerp is bepaald dat deze opties te benaderen zijn door een speler te selecteren in het speleroverzicht. Het belangrijkste doel van deze pilot zal daarom het ontwerpen en implementeren zijn van deze functionaliteiten.

The figure displays four mockups of a player details page, arranged in a 2x2 grid. Each mockup shows a navigation bar at the top with tabs: Overview, Issues, Cashouts, Players, and Games. Below the navigation bar is a search bar and a list of tabs: Summary, Overview, Deposits, Active players, and Winning players. The main content area is divided into several sections:

- Overview:** Displays player information (Username: bob@bob.com, Email: bob@bob.com, Balance: 3432.33, Real/Bonus: 3299.00/33.33) and a list of recent deposits and activity.
- Issues:** Displays a list of issues (e.g., "I have no balance received on monday my last deposit") and a table of past issues.
- Financial accounting:** Displays a table of financial transactions (e.g., "Loss/winn: \$333.33 (ratio: 0.44)").
- Game sessions:** Displays a table of game sessions (e.g., "22-22: 22-23: B7").
- Player details:** Displays a table of player details (e.g., "22-22: 22-23: B7").

Each mockup also includes a sidebar with links (Block player, Add bonus, Send message) and a search widget. The bottom right mockup includes a yellow sticky note that says "Summary shows total per game type".

Figuur 8-1: Mockups spelerdetails, zoals gedefinieerd in het systeemontwerp

<sup>6</sup> Zie Definitiestudie 3<sup>e</sup> pilot en Plan van Aanpak 3<sup>e</sup> pilot voor meer informatie



### 8.1.1 Doelen

Naast het uitbreiden van de spelerdetails zijn er ook nog wat onopgeloste zaken uit de vorige pilot(s) die behandeld moeten worden. Hieronder een specificatie van de doelen voor deze derde pilot;

1. De problemen en onvoltooide onderdelen uit de vorige pilot oplossen, of bepalen hoe deze opgelost dienen te worden.
2. Mogelijkheid bieden om een geavanceerde spelerfilter aan te maken, die de gebruiker op kan slaan voor hergebruik.
3. Speleroverzicht overeen laten komen met de ontworpen mockups<sup>7</sup>.
4. Toekennen van rollen aan de managers, zodat de gebruikersomgeving taakrelevante resultaten kan tonen (zie Figuur 8-1).
5. Administrator moet rollen en backends aan de managers kunnen toewijzen.

### 8.1.2 Systeemeisen van de pilot

Voorgaande doelen geven een aantal systeemeisen voor deze pilot. Dit zijn de eisen waaraan het systeem dient te voldoen na afronding van de pilot. Als het systeem voldoet aan deze eisen zal ook voldaan worden aan de eerder opgestelde doelen.

- Correcte registratiedatum bij speleroverzichten
- Mogelijkheid tot bewerken (en opslaan) van speler
- Mogelijkheid tot filteren van spelerlijst aan de hand van eigen filters.
- Spelerlijstfilters dienen opgeslagen te kunnen worden voor hergebruik
- Spelerfilters dienen per gebruikersrol beschikbaar te kunnen zijn, zodat medewerkers met eenzelfde rol deze filter kunnen opvragen.
- Spelerdetails
  - Mogelijkheid tot inzien van deelgenomen gamesessies, met filtermogelijkheid op start en einddatum
  - Mogelijkheid tot inzien van uitbetalingen
  - Mogelijkheid tot inzien stortingen en openstaande stortingen
  - Chronologisch overzicht uitbetalingen
- Limieten bij winners en verliezers veranderen van *“kleiner dan”* naar *“kleiner of gelijk aan”*, zoals geconstateerd in de validatietest van de tweede pilot
- Rollensysteem voor beheerders, een beheerder moet één of meerdere rollen kunnen hebben
- Toegankelijkheidssysteem voor beheerders, een beheerder moet toegang kunnen hebben tot één of meerdere backends.
- De medewerker met een administrator rol moet de beheerders kunnen voorzien van rollen en backends. Deze optie dient niet toegankelijk te zijn voor andere rollen.
- De medewerker met een administrator rol moet de beheerders kunnen verwijderen, en toegankelijkheid tot rollen en backends kunnen intrekken. Deze optie dient niet toegankelijk te zijn voor andere rollen.

---

<sup>7</sup> Zie Systeemontwerp



### 8.1.3 Tijdschatting

Voor deze pilot zijn twee weken beschikbaar. Gezien verloop van eerdere pilots is onderstaande tijdsschatting een reële verdeling.

1. Definitiestudie en plan van aanpak (2 dagen)
2. Softwareontwerp van de te nieuwe functionaliteiten (2 dagen)
3. Implementatie van nieuwe functionaliteiten (4 dagen)
4. Testplan en testrapport van de ontwikkelde software (2 dagen)

### 8.1.4 Faseplanning

Voor het totale project zijn nog vier weken beschikbaar. De planning voor deze periode is als volgt;

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
14	10-5-2010	Pilot 3; Game sessions, speler details	Definitiestudie, plan van aanpak	3
15	17-5-2010	Pilot 3: implementatie, testen	Testrapport	3
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	





## 8.2 Implementeren ontwerp 3e pilot

De 3<sup>e</sup> pilot is te verdelen in het ontwerpen en implementeren van de backendinterface (de casino- en pokeradapters) en de gebruikersomgeving. Dit hoofdstuk beschrijft kort wat er tijdens deze periode is ontwikkeld.

### 8.2.1 Implementatie GMS backendadapters

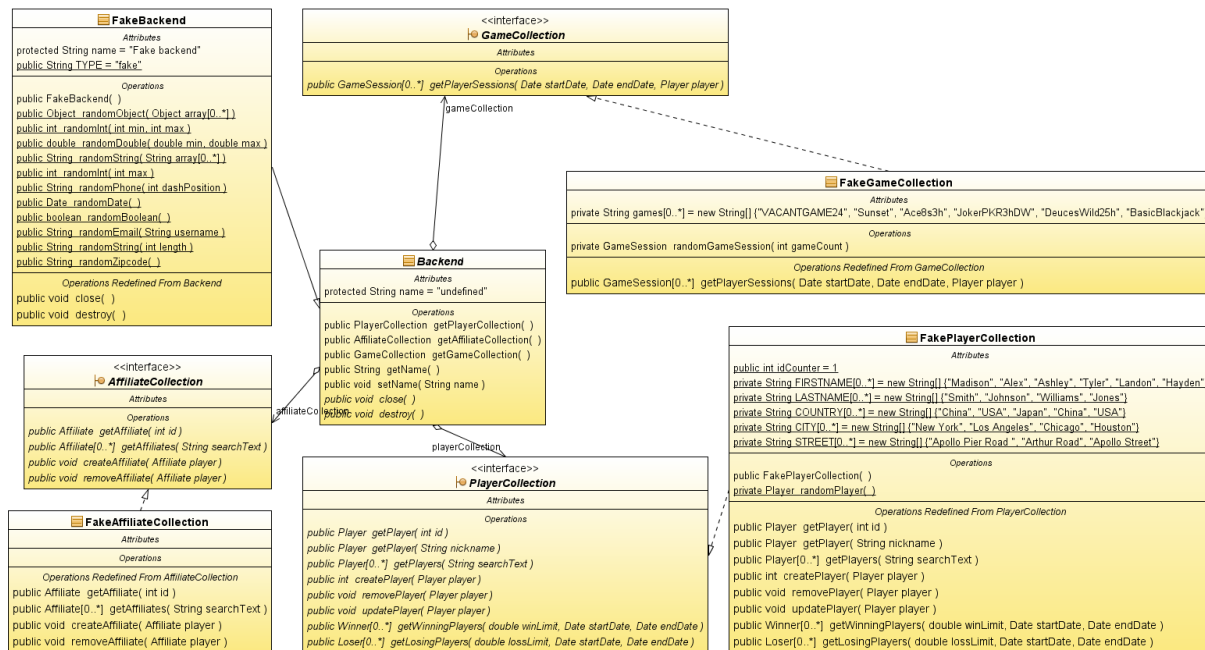
#### 8.2.1.1 Limieten bij winnaars en verliezers

Uit de validatietest bleek dat de filter op de win- en verlieslimieten enkel de resultaten retourneerde met een lagere waarde. Deze filter zit echter in de casino software zelf, waarvan is opgegeven in de randvoorwaarden deze niet aan te passen.

#### 8.2.1.2 Fake backend

Het ophalen van data uit de casinobackend kan lang duren. Om de ontwikkeling en het testen van functionaliteiten te versnellen is er bedacht een nep backend te maken; een backend met niet bestaande spelers, wat snel toegankelijk zou zijn. Er is onderzoek gedaan naar de mogelijkheden van open-source project Mockito, waarmee willekeurige objecten gemaakt en getest zouden kunnen worden. Omdat het genereren van deze objecten niet verliep zoals gewenst is er besloten om een nieuw backend te maken; het Fake Backend. Dit backend heeft geen externe verbindingen (zoals een database) en bevat mogelijkheden om een willekeurige lijst met spelers en spelsessies te genereren. Zie Figuur 8-2 voor een klassendiagram van dit backend.

*Zie de digitale versie voor een duidelijkere weergave*



**Figuur 8-2: Klassendiagram van het Fake Backend; een backend dat willekeurige data genereert voor het testen van de webUI**





## 8.2.2 Aanpassing van winnaars en verliezers

De bedrijfsmentor attendeerde erop dat de classes winner en loser niet geheel correct zijn geïmplementeerd. Deze classes zouden namelijk geen personen zijn, maar zouden moeten refereren naar een persoon. Een speler heeft immers niet de rol winnaar, maar de winst wordt toegekend aan een speler.

Dit resulteerde in het volgende klassendiagram:

*Zie de digitale versie voor een duidelijkere weergave*



Figuur 8-3: Klassendiagram van de personen Affiliate en Player, met hun gerelateerde klassen



## 8.2.3 Implementatie gebruikersomgeving

### 8.2.3.1 Registratiedatum bij speleroverzichten

Uit de validatietest van de 2<sup>e</sup> pilot bleek dat er een uur verschil zit in de weergegeven registratiedatum en de eigenlijke registratiedatum. In deze pilot is het probleem onderzocht. Voor beide backends is de registratiedatum opgeslagen in de database als tekst, bijvoorbeeld de string "2009-05-06 11:26AM". Voor het spelerobject wordt deze datum opgeslagen in het java.util.Date formaat. Dit betekent dat de datumstring geparsed moet worden met de US locale, zodat de notatie herkend wordt. In de GameManagementUI wordt de datum weer omgezet naar het formaat dat de gebruiker (manager) gespecificeerd heeft bij zijn instellingen. Waarschijnlijk verschuift de datumweergave in verband met de tijdzones. Een directe consistente oplossing is nog niet gevonden.

### 8.2.3.2 Speler bewerken

Het is vanaf nu mogelijk om de gegevens van een speler te bewerken. Deze functionaliteit is onderdeel geworden van de spelerdetails. De tab voor het bewerken van een speler is geïmplementeerd zoals ontworpen in het systeemontwerp.

The screenshot displays the 'Player overview' interface. At the top, there's a green header with 'Player overview' and a 'Logout' button. Below the header, a navigation bar contains tabs: Overview, Issues, Cashouts, Games, Players, and Settings. A dropdown menu shows 'Fake America'. Below this, another set of tabs includes Players, Winning Players, Losing Players, and Affiliates. The main content area is divided into three sections. On the left, a silhouette icon represents the player. The middle section, titled 'Player details of Alex Howard', shows fields for Username (Alex46166), E-mail (Alex46166@gmail.com), Balance (140.08), and Real / Bonus (228.4 / 424.91). It includes buttons for 'Block player', 'Add bonus', and 'Send message'. The right section, 'Find player', has a search bar and a 'Search' button. Below the search bar, 'Player details' are listed: Firstname (Alex), Lastname (Howard), Username (Alex46166), Last activity (2009-06-12 03:47), Balance (140.08), City (Overland Park), and Country (Venezuela), with a 'Back to data list' button. At the bottom, a tabbed interface shows 'Summary', 'Issues', 'Financial accounting', 'Game sessions', and 'Edit player'. The 'Edit player' tab is active, showing 'Edit player credentials' with input fields for E-mail, Firstname, Lastname, Username, Address, City, Zipcode, Country, State, Phone, and Mobile phone, followed by an 'Apply' button. The footer of the application reads 'Gaming and Gambling 2010'.

Figuur 8-4: Bewerken van de gegevens van een geselecteerde speler



### 8.2.3.3 Spelerfilter / geavanceerd zoeken

Naar deze functionaliteit is onderzoek gedaan. Er zijn twee proofs of concept gemaakt, waarmee de geavanceerde queries gemaakt moesten worden. Een eis van deze geavanceerde zoekfunctie was namelijk de mogelijkheid tot opslaan en hergebruik van de filter. Er is geprobeerd deze queries op te slaan met behulp van Hibernate classes en met behulp van java properties files. Beide concepten bleken niet succesvol; ze waren te groot of het lukte simpelweg niet om de geavanceerde condities op te slaan. Een voorbeeld van zo'n conditie is: de speler is ouder dan 40 jaar en heeft meer dan €20,- spelersgeld, of de speler heeft zich in de afgelopen 10 weken geregistreerd.

### 8.2.3.4 Deelgenomen gamesessies

Een van de nieuwe tabbladen van de spelersdetails is het weergeven van de spelsessies waar een speler aan heeft deelgenomen. Op deze tab is het mogelijk om te filteren op het speltype en een datumbereik om te geven van de getoonde spelsessies.

The screenshot shows a web application interface for 'Player overview'. At the top, there's a green header with 'Player overview' and a 'Logout' button. Below the header, there's a navigation bar with tabs: Overview, Issues, Cashouts, Games, Players, and Settings. A dropdown menu shows 'Fake America'. Below the navigation bar, there's a sub-navigation bar with tabs: Players, Winning Players, Losing Players, and Affiliates. The main content area is divided into two columns. The left column shows 'Player details of Alex Howard' with fields for Username (Alex46166), E-mail (Alex46166@gmail.com), Balance (140.08), and Real / Bonus (228.4 / 424.91). There are buttons for 'Block player', 'Add bonus', and 'Send message'. The right column shows a 'Find player' search bar and 'Player details' for Alex Howard, including Firstname, Lastname, Username, Last activity, Balance, City, and Country. Below the player details, there's a 'Back to data list' button. The bottom section shows a table of game sessions with columns: Game name, Game count, Bets, Payouts, and Gain/Loss. The table is filtered for the date range 'Aug 24, 2007' to 'May 20, 2010'. The table contains 14 rows of data. At the bottom right, there's a footer that says 'Gaming and Gambling 2010'.

Game name	Game count	Bets	Payouts	Gain/Loss
VACANTGAME24	100	390.45	5464.88	120.2
Sunset	93	190.95	4329.1	172.25
Sunset	85	-87.41	273.89	-73.05
Sunset	76	272.64	94.4	464.18
DeucesWild25h	69	-63.11	5606.35	480.98
Ace8s3h	64	371.32	1747.08	444.54
DeucesWild25h	62	235.18	3912.55	331.36
JokerPKR3hDW	54	376.82	3585.42	-22.83
JokerPKR3hDW	53	206.07	829.62	110.65
Ace8s3h	45	410.51	5108.61	153.2
VACANTGAME24	43	86.32	5486.59	305.57

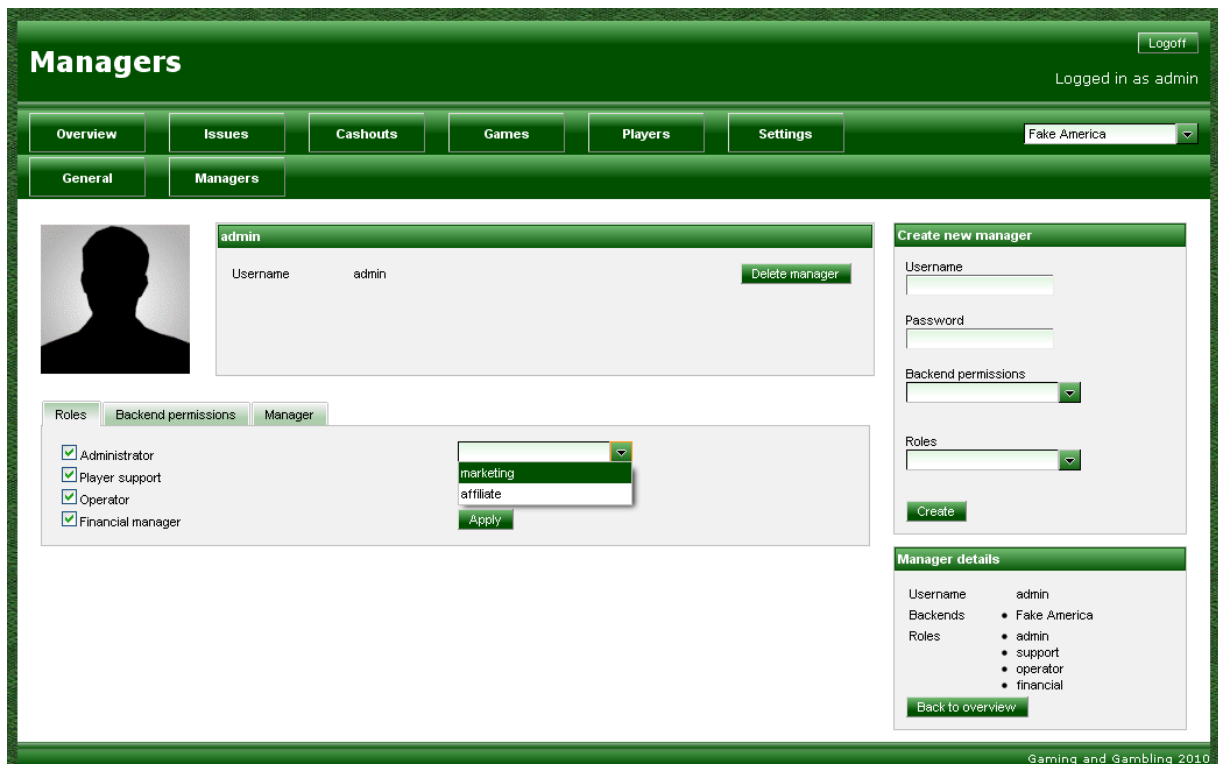
Figuur 8-5: Deelgenomen spelsessies van een geselecteerde speler



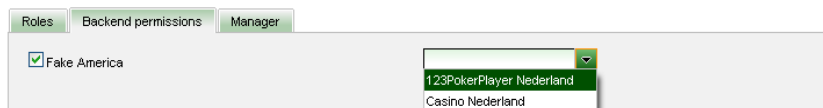
## 8.2.4 Manager rollen

De Game Management Service zal gebruikt worden door verschillende soorten beheerders, namelijk *administrator*, *operator (spelbeheerder)*, *support medewerker*, *marketing medewerker*, *financieel medewerker* en *affiliate beheerder*.

In deze pilot zijn deze rollen geïmplementeerd; aan de hand van de rol kan een functionaliteit getoond worden. De administrator heeft toegang tot alle functies. In het managers tabblad van de toolbar is de lijst met beheerders te vinden. Door een beheerder aan te klikken kan de administrator rollen toewijzen en verwijderen. Een persoon kan meerdere rollen tegelijk hebben. Daarnaast is het tabblad met de toewijzing van backends aangemaakt. Hier kan de administrator configureren welke backends een persoon mag benaderen.



Figuur 8-6: Toevoegen en verwijderen van managerrollen, afhankelijk van de taak van de werknemer



Figuur 8-7: Toestemmingen voor medewerkers om een backend te beheren

### 8.2.4.1 Overzicht uitbetalingen en stortingen

De financiële onderdelen van de spelerdetails zijn niet geïmplementeerd, omdat er geen ontwikkeltijd meer beschikbaar was. Deze onderdelen dienen in een vervolgproject te worden geïmplementeerd.



## 8.3 Testrapport 3e pilot

Hoewel niet alle pilotonderdelen zijn voltooid, is er volgens planning gestart met het testen van de geïmplementeerde onderdelen. De tests die worden uitgevoerd zijn de validatietest en de unittest.

### 8.3.1 Resultaten validatietests

Met de validatietest is op een gestructureerde manier onderzocht welke onderdelen niet goed werken, of nog gemaakt moeten worden. Uit de validatietest is dan ook eenvoudig een eisenlijst te halen die in de volgende pilot kan worden behandeld. Over het algemeen werkt de gebruikersinterface naar behoren.<sup>8</sup>

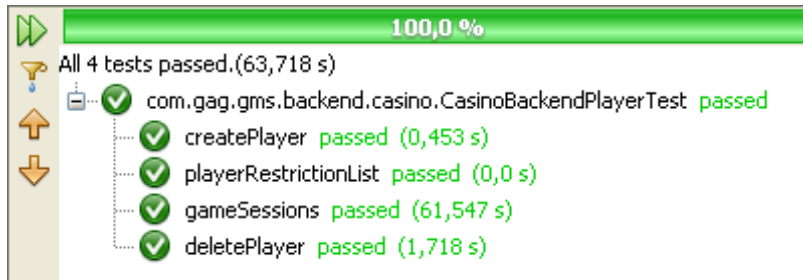
Spelersoverzicht	Opmerkingen	Geslaagd
Selecteer een speler, vergelijk datatabel gegevens met detailgegevens	Bij Casino is er één uur tijdsverschil tussen registratiedatum in de tabel en in de details, bij Poker is er twee uur tijdsverschil	NEE
Winnersoverzicht	Opmerkingen	Geslaagd
Filter winlimiet van spelers (min, max en random waarde) en controleer uitvoer	Winlimiet blijkt een 'hoger dan' te zijn ipv een 'hoger of gelijk dan' (backend implementatie)	NEE
Verliezersoverzicht	Opmerkingen	Geslaagd
Filter verlieslimiet van spelers (min, max en random waarde) en controleer uitvoer	Losslimiet blijkt een 'hoger dan' te zijn ipv een 'hoger of gelijk dan' (backend implementatie)	NEE
Spelsessies	Opmerkingen	Geslaagd
Selecteer andere startdatum en filter lijst	De datumfilter lijkt niet te werken; wellicht dat de pagina niet goed ververst wordt of de casinobackend niet de juiste elementen teruggeeft	NEE
Selecteer andere einddatum en filter lijst	<i>idem</i>	NEE
Wissel einddatum en startdatum en filter lijst		NEE

<sup>8</sup> Zie het testrapport voor de tests die wel geslaagd zijn



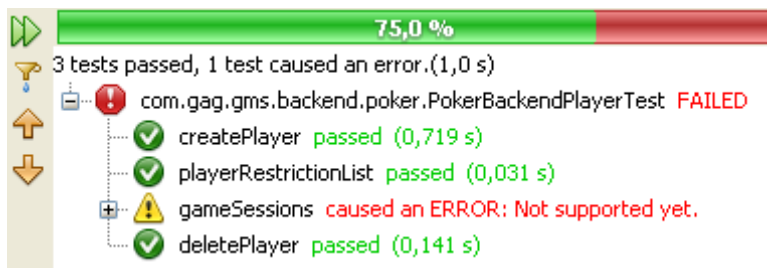
### 8.3.2 Resultaten unit tests Casino backend

Voor deze test zijn de testmethoden uit de vorige pilot aangevuld met de nieuwe functies rondom de gamesessions. Uit de test blijkt dat de createPlayer, playerResctrictionList en deletePlayer methoden nog steeds naar behoren werken. Het ophalen van de spelsessie informatie neemt veel tijd in beslag, namelijk 61 seconden. Dit word veroorzaakt door de vele objecttransformaties binnen de casinosoftware. Het zou stukken sneller zijn om de backend de data direct uit de database te laten halen.<sup>9</sup>



### 8.3.3 Resultaten unit tests Poker backend

Van de pokerbackend is het bekend dat het ophalen van de spelsessiedata nog niet mogelijk is. De tests createPlayer, playerRestrictionList en deletePlayer zijn wel succesvol verlopen.<sup>10</sup>



<sup>9</sup> Zie voor de volledige unittest van het casinobackend het testrapport

<sup>10</sup> Zie voor de volledige unittest van het pokerbackend het testrapport

## 9 Afronding project

Met de derde pilot wordt het project afgesloten. Er is geen ontwikkeltijd meer beschikbaar voor een volgende pilot. Ter afsluiting van het project is er een overdrachtsdocument gemaakt waarin de ontwikkelde onderdelen beschreven worden.

Met behulp van dit document is de laatste test uit het testplan uitgevoerd; het testen van de uitbreidbaarheid van het systeem. Een softwareontwikkelaar van MC Advies & Management gaat proberen het betalingssysteem te implementeren in de Game Management Service.

Voor deze test zal dhr. B. Schlärmann proberen het betalingsoverzicht van de spelers te implementeren. Deze informatie zal getoond worden bij een spelersselectie, op de tabbladen overzicht en financieel. Het betreft hier de geschiedenis van stortingen die een speler heeft gedaan. Aangezien de casinosoftware momenteel al in gebruik is, wordt deze uitbreiding gemaakt op het casinobackend. De code die tijdens deze test geproduceerd wordt kan dan direct in gebruik worden genomen.

Hoewel de auteur niet aanwezig is geweest bij het testproces, beschrijft dit rapport<sup>11</sup> de bevindingen van dhr. Schlärmann. Het is gelukt om het betalingsoverzicht te maken. Zie het rapport voor meer informatie. De screenshot hieronder toont het resultaat.

The screenshot displays a web application interface titled "Player overview". At the top right, there is a "Logoff" button and the text "Logged in as casino". Below the title bar is a navigation menu with tabs: Overview, Issues, Cashouts, Games, Players, and Settings. A dropdown menu shows "Casino Nederland". Below this is a secondary menu with tabs: Summary, Deposits, Active Players, Game Sessions, Winning Player, and Issue reports.

The main content area is divided into two columns. The left column shows "Player details of" for a player with a silhouette icon. The details include: Username, E-mail (affsubtest\_-1869349305@casino.lan), Balance (7.5), and Real / Bonus (0.0 / 7.5). There are buttons for "Block player", "Add bonus", and "Send message". Below this is a sub-menu with tabs: Summary, Issues, Financial accounting, Game sessions, and Edit player. The "Summary" tab is active, showing a table with the following data:

Date	Amount	Bonus	Method	Status
Feb 17, 2010 9:04 AM	\$200.00	\$215.00	Payment Gate1	FINISHED

The right column has a "Find player" search bar with the text "affsubtest\_-1869349305@e" and a "Search" button. Below this is a "Player details" section with fields for Firstname, Lastname, Username, Last activity, Balance (7.5), City, and Country. A "Back to data list" button is at the bottom.

At the bottom right of the interface, it says "Gaming and Gambling 2010".

**Figuur 9-1: Resultaat van de uitbreidbaarheidstest; de betalingsgeschiedenis van een geselecteerde speler**

<sup>11</sup> Zie Testrapport uitbreidbaarheid





## 10 Evaluatie

Dit hoofdstuk beschrijft de evaluatie van processen en producten van dit afstudeerproject.

### 10.1 Procesevaluatie

Tijdens het afstudeerproject is gebruik gemaakt van de ontwikkelmethode Iterative Application Development. Omdat er bij MC Advies & Management geen bedrijfsmethode is gespecificeerd, mag iedere softwareontwikkelaar zijn eigen ontwikkelingsmethode kiezen. Ik heb gekozen om gebruik te maken van de IAD vorm *incrementeel opleveren*. Dit betekent dat het project wordt opgedeeld in verschillende iteraties, waarin gegroepeerde delen van het uiteindelijke systeem ontwikkeld worden. Aan het einde van iedere iteratie kan de nieuwe functionaliteit worden gebruikt door de eindgebruiker(s).

Ieder iteratie wordt een pilot genoemd. Tijdens iedere pilot worden er een aantal documenten opgeleverd; definitiestudie (wat te doen), plan van aanpak (hoe te doen), testplan (wat te testen), testrapport (resultaten tests) en een pilotdocument (wat is er ontwikkeld).

Hoewel het documenteren aardig wat tijd in beslag nam, vond ik het een goed bruikbare ontwikkelmethode. Deze methode geeft een duidelijke structuur aan het project en biedt veel ruimte voor aanpassingen door voortschrijdend inzicht. Op deze manier blijft het proces flexibel; het doel van een pilot wordt afgestemd op de voorgaande pilot, met de meest recente inzichten. Hoewel een voorafgaande planning noodzakelijk is voor het overzicht, kan op deze manier iedere keer bekeken worden welke onderdelen prioriteit hebben of passen bij de voorgaande ontwikkelingen.

Hoewel de opdrachtgever M. de la Rie de taak heeft gegeven om dit systeem te ontwikkelen, is het meeste overleg geweest met de bedrijfsmentor B. Schlärman. Tijdens een vergadering heeft M. de la Rie samen met een andere medewerker aangegeven wat de doelen en eisen van het project zijn, zoals de nieuwe functionaliteiten beschreven in hoofdstuk 4.3.2. Na deze vergadering is samen met B. Schlärman onderzocht hoe deze eisen verwerkt konden worden.

Na afronding van iedere pilot is de software getoond aan deze bedrijfsmentor, waarna de voortgang en volgende pilotfocus werd bepaald. Ik vond het een prettige samenwerking. Ik had echter gehoopt veel verder te kunnen komen met dit project. Het geeft immers voldoening als een project volledig compleet wordt afgeleverd. Door de onderzoeken naar de juiste gebruiksvormen van Seam en Richfaces, de wachttijd bij deployen en natuurlijk het beschrijven van het proces in dit afstudeerverslag is er minder tijd beschikbaar geweest voor het implementeren van het ontwerp. Ik ben echter wel erg tevreden over dit opgeleverde systeemontwerp; als een andere softwareontwikkelaar verder gaat met dit project en zich houdt aan de beschreven workflows kan dit een prachtige webapplicatie worden.





## 10.2 Productevaluatie

Hoewel het project niet voltooid is, is wel aan alle vermeldde werkzaamheden uit het afstudeerplan<sup>12</sup> voldaan. Ook alle beloofde (tussen)producten zijn opgeleverd. Echter is er één afwijkend punt; in het afstudeerplan en in het plan van aanpak is aangegeven dat de data uit de pokersoftware zal worden benaderd met behulp van rmi, een koppeling waarmee tussen Java en C# objecten kunnen worden uitgewisseld. Alle benodigde data was echter beschikbaar vanuit de bijbehorende database, waarin stored procedures eenvoudige en lastige taken overnamen. Het bleek dan ook dat de koppeling met de C# software niet nodig was; alle data kon bereikt worden door middel van sql-statements op de database. Een bijkomend voordeel was dat deze oplossing ook veel sneller is dan een rmi verbinding. Zie het uiteindelijke advies voor de voortvloeiende conclusie wat betreft het casino backend.

Tijdens dit project zijn de volgende documenten opgeleverd:

- Plan van Aanpak (totale project)
- Systeemontwerp
- Testplan
- Pilot 1: Definitiestudie
- Pilot 1: Plan van aanpak
- Pilot 1: Pilotdocument
- Pilot 2: Definitiestudie
- Pilot 2: Plan van aanpak
- Pilot 2: Testrapport
- Pilot 2: Pilotdocument
- Pilot 3: Definitiestudie
- Pilot 3: Plan van aanpak
- Pilot 3: Testrapport
- Pilot 3: Pilotdocument
- Overdrachtsdocument (beschrijving onderdelen)
- Testrapport uitbreidbaarheid

En de volgende schoolgerelateerde documenten:

- Planningen
- Logboek
- Afstudeerplan
- Afstudeerverslag
- Voortgangsverslag

---

<sup>12</sup> Zie Afstudeerplan Gertjan AI



## 10.3 Evaluatie opgeleverde software

### 10.3.1 Backendadapters

Hier is het project mee begonnen; het onderzoeken van de mogelijkheid om data uit de bestaande software op te halen. Zodra dit mogelijk zou zijn zou de data getoond en bewerkt kunnen worden vanuit een andere gebruikersomgeving. Ik vind dit zelf altijd leuk om uit te zoeken en te ontwerpen; dit zijn de onderdelen van het systeem die geen enkele gebruiker zal zien. Hoewel deze codeonderdelen meerdere keren tijdens het gebruik van de webUI worden aangeroepen heeft dit deel niets te maken met de uiteindelijke weergave van de data. Dit is te vergelijken met het Model-View-Control model, waarin de spelers en affiliates de modellen zijn, de view de gebruikersomgeving en de control de manier om deze objecten te vullen en beschikbaar te maken.

Het project is gestart met het ontwerpen van de backendinterface, waarna de casinoadapter is geïmplementeerd. Deze maakt verbinding met de casinosoftware. Deze casinosoftware beschikt over de methoden om data op te halen. Echter zijn er onnodig veel translaties in dit proces. De data gaat namelijk van database tabel naar een untyped Vector, wordt dan geparsed naar een GMS model. Dit proces wordt voor alle queries uitgevoerd, dus bij een lange resultatenlijst wordt de data van iedere tupel op deze manier geconverteerd. In het advies in hoofdstuk 10.5 zal hier op teruggekomen worden.

De pokersoftware maakt –zoals eerder beschreven– gebruik van een snellere methode, namelijk het ophalen van de data direct uit de database. Met deze data worden de modelobjecten gevuld. Hoewel de pokerbackend prima te onderhouden is vanuit het ontwikkelde Game Management Systeem, is hier wel wat verbetering mogelijk. Daarover meer in het advies in hoofdstuk 10.5.

De fake backend is enkel ontwikkeld voor testdoeleinden. Eventueel kan deze ook gebruikt worden bij demo's waarbij geen internetverbinding beschikbaar is. Deze backend genereert een willekeurige spelersdatabase aan de hand van collecties NAW gegevens.

Achteraf gezien denk ik dat de architectuur van de backends degelijk in elkaar zit. Ze zijn makkelijk uitbreidbaar en overzichtelijk vanwege de gegroepeerde collecties. Er is nog wel snelheid te behalen door de implementaties te verbeteren.

### 10.3.2 Gebruikersinterface

Voor het tonen van de webinterface is gebruik gemaakt van het Seam framework, met hulp van de Richfaces library. Hoewel het deployen van een Seam webapplicatie redelijk wat tijd in beslag kan nemen, denk ik dat dit een goede keuze is geweest vanwege de vele bruikbare mogelijkheden zoals security options en page-redirection. De Richfaces library biedt mooi vormgegeven controls aan, zodat de standaard HTML formulierobjecten er wat aantrekkelijker uitzien. Ook de mogelijkheid voor het snel opbouwen en vullen van datatabellen is een welkome mogelijkheid. In andere projecten zal ik waarschijnlijk weer voor gebruik van Seam en Richfaces kiezen.

De medewerkers kunnen op dit moment een aantal taken uitvoeren met behulp van het nieuwe Game Management Systeem. Het is bijvoorbeeld mogelijk om spelers te zoeken, te bewerken en om



hiervan de spelsessies te bekijken. Met het tabblad *winning players* kan snel gefilterd worden welke personen eventueel valsspelen.

De rolverdeling van de managers vind ik ook mooi; de gebruikersomgeving past zich aan aan de rol (en dus de taak) van een medewerker. De administrator kan zo opgeven welke taken medewerkers wel en niet mogen uitvoeren. Ook kan hij de beschikbare backends toewijzen aan managers; dit zijn de backends die een manager mag bekijken en beheren.

## 10.4 Behalen van de competenties

In het afstudeerplan is aangegeven aan een vijftal competenties te gaan werken. Deze competenties geven aan of de auteur beschikt over voldoende kennis en kunnen om af te studeren voor de HBO opleiding Informatica. Hieronder het overzicht van deze competenties en een beschrijving hoe deze behaald is.

Naast deze vijf punten zouden de Informatica competenties voor database ontwerp en gebruik kunnen worden toegevoegd, daar deze zijn gebruikt voor het ophalen van de data, het ontwerp, implementeren en gebruik van de managers, managertoestemmingen, managerinstellingen en managerrollen.

### 1.4 *Uitvoeren analyse door definitie van requirements*

Onderzoek naar overeenkomsten tussen casino en pokerbackend, zoals beschreven in de eerste definitiestudie.

### 3.1 *Ontwerpen softwarearchitectuur*

Onderzoek naar taakgerelateerde workflows, gebruikmakend van de methode Usage Centered Design; systeemontwerp in de vorm van een klassendiagram, zoals beiden beschreven in het systeemontwerp.

### 3.2 *Ontwerpen systeemdeel*

Ontwerpen en implementeren van de pilots, ontwerpen en implementeren van de drie backendadapters.

### 3.3 *Bouwen applicatie*

Implementeren en bouwen van het Game Management Systeem, de web UI, de backend interface en de backend adapters van de casino- en pokersystemen.

### 3.5 *Uitvoeren van en rapporteren over het testproces*

Opstellen van het testplan, uitvoeren van de validatietests en unittests in pilots 2 en 3.



## 10.5 Advies voor vervolg van het project

Met dit project is een goede start gemaakt aan de Game Management Service. Dit project heeft een grote kans van slagen, omdat van zowel het casino- als de pokerbackendadapter bewezen is dat de data kan worden getoond en bewerkt met behulp van één webinterface. Echter zou ik willen adviseren om een aantal onderdelen aan te passen of toe te voegen;

Allereerst denk ik dat het een goede keuze zal zijn om de koppeling met de casinosoftware te vervangen door een directe verbinding met de database. De onnodige eerder beschreven objecttranslaties zijn dan niet meer van toepassing, waardoor meer snelheid kan worden gehaald uit het ophalen van data en uitvoeren van queries. Ik zou ook willen adviseren om deze verbinding niet te laten verlopen via jdbc sql queries, zoals momenteel geïmplementeerd in de pokerbackendadapter. Tijdens dit project ben ik erachter gekomen dat de verbindingen via Hibernate veel effectiever zijn en geoptimaliseerd zijn qua prestaties en snelheid. Het zou dus een goed plan zijn om de implementatie van beide backends te vervangen met een Hibernate verbinding. De eerder aangemaakte objecten Player, Affiliate, Winner en Loser kunnen hiervoor worden uitgebreid met de Hibernate annotaties.

Verder zou ik willen aanmoedigen op het gebruik van andere serversoftware, daar Glassfish v2 verouderd is. Het zou het beste zijn om een server of serveruitbreiding te vinden waarbij wijzigingen in Java code direct gedeployed kunnen worden. Dit zal veel herstarttijd van de server besparen en het ontwikkelen sneller laten verlopen.

## 10.6 Persoonlijke evaluatie

Ik vond het een erg leuke afstudeeropdracht; er kwam namelijk van bijna ieder schoolproject wel een onderdeel voor in dit project. Het ontwerpen van een webinterface (G-blok), het ontwerpen van een database met behulp van UML (I1-blok), het object-geïntendeerd programmeren van de personen klassen en toepassen van TMap (I2-blok), (het onderzoeken van) de bruikbaarheid van rmi (I3-blok), het gebruik van jdbc voor een verbinding met Microsoft SQL (I5-blok), het ontwerp van het systeem (I6-blok) en het onderzoeken van de mogelijkheden en wensen van de gebruikers voor het nieuwe systeem (I7-blok).

Verder vond ik de mogelijkheid tot overleg met de opdrachtgever en bedrijfsmentor erg prettig omdat er geen afstandelijke sfeer bij MC Advies & Management heerst. Hoewel veel projectonderdelen en technieken van school konden worden toegepast bij dit project heb ik het meeste geleerd van de nieuwe methode *Usage Centered Design*. Op school heb ik wel geleerd over het toepassen van *User Centered Design*, wat gefocust is op de eisen en wensen van de gebruiker. Bij Usage Centered Design staan de taken en workflows centraal, wat naar mijn mening een veel betere methode is om gebruiksvriendelijke software te ontwikkelen.

Al met al een leerzame, leuke en geslaagde afstudeeropdracht.



## 11 Verklarende woordenlijst

Woord	Verklaring
<b>Annotatie</b>	Vorm van toevoegen van meta-data aan een variabele, object of methode. De tag gaat vooraf aan een @-teken, en verwijst naar een klasse of property.
<b>Backend</b>	Onderdeel van goksoftware wat de beheerderstaken bevat in de vorm van aan te roepen methoden en variabelen.
<b>Game Management Service</b>	De service die de verbinding met verschillende backends tot stand brengt door adapters aan te maken en deze data te tonen in een gebruiksvriendelijke webomgeving.
<b>Pilot</b>	Iteratie in de softwareontwikkelmethodiek Iterative Application Development.
<b>Usage Centered Design</b>	Ontwerp gespecialiseerd op de taken en workflows van gebruikers.
<b>Web User Interface</b>	Verzameling webpagina's (visuele omgeving) vanwaar een gebruiker de onderliggende systeemfunctionaliteiten kan aanroepen.

## 12 Boeken en rapporten

Tijdens dit project is gebruik gemaakt van de volgende boeken en rapporten:

Larry Constantine, R. B. (sd). *http://citeseerx.ist.psu.edu*. Opgeroepen op 03 2010, van CiteceerX, Scientific Literature Digital Library:  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.4753&rep=rep1&type=pdf>

Luckwood, C. &. *Laboratory and Field testing of usability*.

Martin Pol, E. v. *An introduction to TMap*. Sdu.

Tolido, R. (1995). *IAD, Het evolutionair ontwikkelen van informatiesystemen*. Utrecht: Academic Service.

## 13 Index

adapter design pattern, 28  
affiliate, 12, 46  
Ajax, 37  
annotatie, 37, 41, 61, 84  
backend, 28, 84  
bijection, 37  
black-box test, 24  
database, 61  
datamodel, 43  
framework, 37  
Hibernate, 37, 61  
IAD. *Zie* Iterative Application Development  
injection, 37  
Iterative Application Development, 15, 16  
Java Enterprise Edition, 37  
JavaServer Faces, 37  
JBoss Application Server, 37  
klassendiagram, 36, 43, 44, 47, 56, 57  
kwaliteitstests, 25  
localization, 62  
Model View Controller, 37, 52  
MySQL, 61  
ontwikkelmethodiek, 15  
outjection, 37  
pilot, 84  
Richfaces, 37, 40, 42  
scope, 21  
Seam, 37, 42, 45  
systeemeisen, 17  
Test Management Approach. *Zie* TMap  
testen, 24  
testplanning, 27  
testproces, 27  
TMap, 24  
unit test, 52, 67  
Usage Centered Design, 22, 23, 24, 38, 82, 83, 84  
white-box test, 24



# Plan van Aanpak

*Ontwikkeling van beheerssoftware voor online casino systeem*

*Afstudeerstage MC Advies & Management*

*Gertjan Al, 20069275*

*8-2-2010 t/m 4-6-2010*

## Inleiding

Dit plan van aanpak is onderdeel van de documentatie rond de afstudeeropdracht bij MC Advies & Management. De afstudeeropdracht is het ontwikkelen van beheerssoftware voor een online casinosysteem.

In het eerste hoofdstuk wordt het probleem en de opdracht beschreven. Hoofdstuk 2 beschrijft hoe de opdracht zal worden aangepakt en welke methoden daarbij gebruikt zullen worden. In hoofdstuk drie worden de risico's en maatregelen besproken, in hoofdstuk vier de afbakening van het project. In hoofdstuk vijf en zes staan de concrete werkzaamheden en op te leveren producten. In het zevende hoofdstuk staat een planning van deze werkzaamheden.





## Inhoudsopgave

1. Opdrachtsomschrijving.....	4
2. Ontwikkeling.....	5
2.1. Methoden en technieken.....	5
2.1.1. Programmeertalen en ontwikkelomgevingen.....	5
2.1.2. Software ontwikkeltechniek.....	5
2.1.3. Testen .....	5
2.2. Architectuur.....	6
3. Risico's en maatregelen.....	7
4. Afbakening.....	7
5. Projectwerkzaamheden.....	8
6. Op te leveren (tussen)producten .....	8
7. Planning .....	9
8. Goedkeuring en ondertekening .....	10



## 1. Opdrachtsomschrijving

MC Advies & Management heeft bestaande software voor onder andere de online gaming markt. Er valt een onderscheid te maken tussen twee los van elkaar werkende systemen: de casino games en het pokersysteem. Deze systemen zijn met verschillende technieken opgebouwd en hebben beiden een aparte user interface om beheertaken uit te kunnen voeren. De casino backend is geschreven in Java en gebruikt een Sybase DBMS, de poker backend is geschreven in C# en gebruikt SQL Server als DBMS.

Hoewel de implementatie van beide systemen verschilt is er wel veel overeenkomende functionaliteit. Beide systemen hebben bijvoorbeeld spelers, games, rondes, bets en bonussen. Ook heeft de achterliggende backend overeenkomstige beheersfunctionaliteiten. Zo kan een beheerder bijvoorbeeld bij allebei de systemen de accounts (spelers) beheren, bekijken hoeveel geld een speler heeft, winsten berekenen, weddenschappen (bets) volgen, bonussen toekennen etc.

MC wil een nieuw systeem (webinterface) waarmee het mogelijk is om beide backends aan te spreken. Hiermee wordt het mogelijk om beheertaken op de casino games en de pokerrooms vanuit dezelfde user interface uit te voeren.

Ondanks de overeenkomstige functionaliteiten moet er wel worden opgemerkt dat dit lang niet voor alle functies geldt. Zo ondersteunt het poker backend multiplayer games, terwijl dat bij het casino niet het geval is.

De huidige *user interfaces* voor de backends ondersteunen niet alle functionaliteiten die het systeem biedt. De UI van het pokersysteem is momenteel verre van voltooid en niet of nauwelijks bruikbaar. De beheerders UI van het casino is opgezet met verouderde ontwikkeltechnieken, waardoor het toevoegen van functionaliteit veel tijd kost.

Beide UI's hebben een hoop ontwikkeling nodig om ze weer up to date te maken. Gezien de overeenkomsten is er daarom besloten om één overkoepelende Management UI te maken.



## 2. Ontwikkeling

### 2.1. Methoden en technieken

#### 2.1.1. Programmeertalen en ontwikkelomgevingen

Het nieuwe Game Management Systeem zal ontwikkeld worden in JAVA, met behulp van de NetBeans IDE. De webinterface zal gemaakt worden met behulp van JAVA framework SEAM. De pokersoftware is gemaakt in C#. Voor deze software moet ook een backend gemaakt worden. De broncode van deze software zal waarschijnlijk onderzocht worden met behulp van Visual Studio.

#### 2.1.2. Software ontwikkeltechniek

Er zal gebruik worden gemaakt van de ontwikkelmethodiek IAD (Iterative Application Development). Bij deze methode wordt er per functionaliteit een definitiestudie gedaan, waarin de te maken functionaliteit wordt beschreven en afgebakend. Na deze studie wordt er een pilot gestart; de ideeën worden geïmplementeerd. Als laatste wordt de nieuwe code getest en wordt de functionaliteit vrijgegeven aan de gebruikers. Naar verwachting zal iedere module een pilot zijn.

#### 2.1.3. Testen

Voor het testen van de pilots zal TMap gebruikt worden. Naar de regels van TMap zal er een testplan worden opgesteld, waarin beschreven wordt wat en hoe er getest zal worden. Per pilot zal dit testplan gebruikt worden om de code en werking van de opgeleverde producten te testen. Na iedere pilottest zal een testrapport gemaakt worden, waarin de bevindingen te lezen zijn.

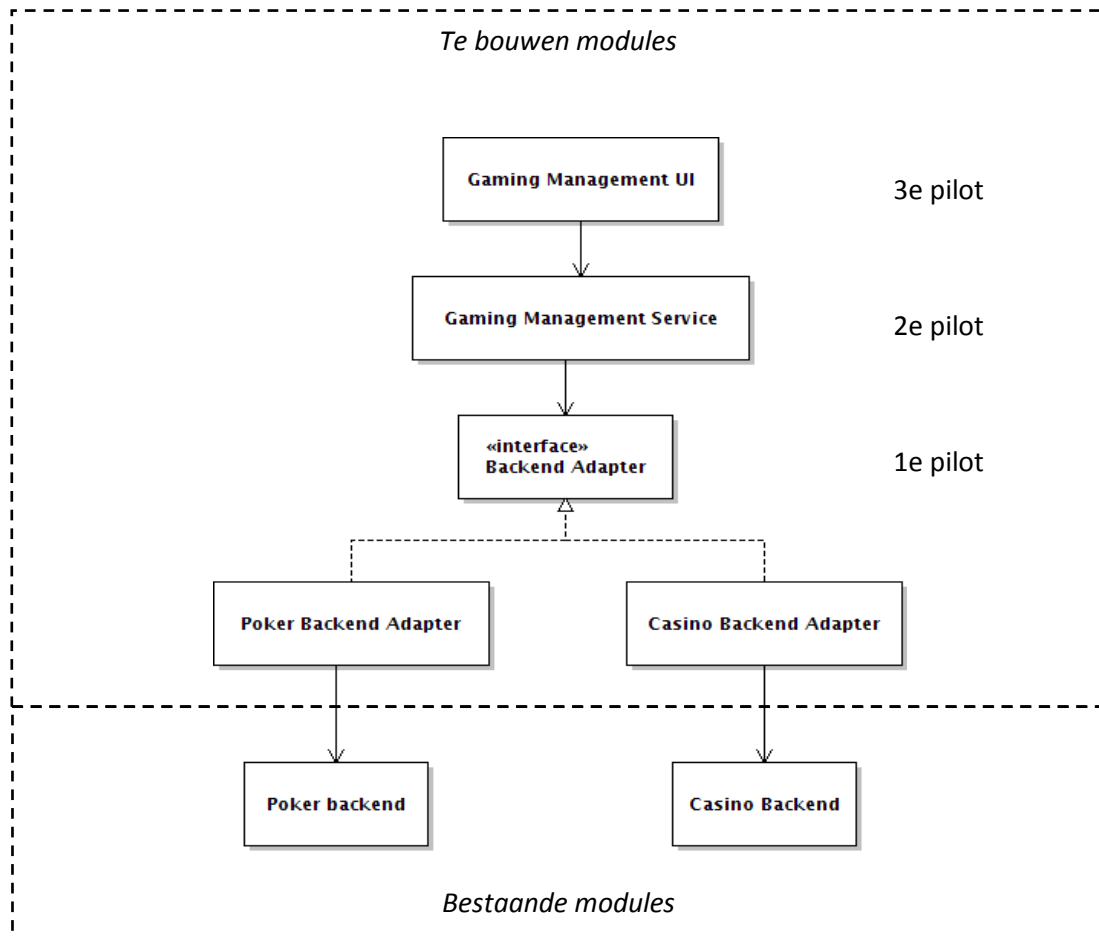
Het testplan bevat meer informatie over de tests die uitgevoerd gaan worden.



## 2.2. Architectuur

De bedrijfsmentor heeft bedacht een pull architectuur te gebruiken voor het Game Management Systeem. Hierbij wordt de data uit de aanwezige systemen 'getrokken'. Hiervoor dient een interface (API) gemaakt te worden die een datamodel heeft wat aansluit op de twee huidige systemen. Er komen dus twee implementaties van de interface die de functies vertalen naar het eigen systeem.

De beoogde architectuur is als volgt schematisch weer te geven:



### 3. Risico's en maatregelen

De pokersoftware is gemaakt in het buitenland en destijds onvolledig opgeleverd. Niet alle functionaliteiten werken volgens ontwerp. Het is goed mogelijk dat dit een probleem oplevert bij het aansturen via de backend. Daarnaast is er een kans dat de software niet (voldoende) gedocumenteerd is. Mocht dit zo zijn, zal er extra werk zijn; (alle) code zal moeten worden doorgelezen.

Een ander risico is dat de backends niet direct aangestuurd kunnen worden via het te bouwen Game Management Systeem, omdat bijvoorbeeld sommige functies niet bereikbaar zijn of dat de software niet geschikt (ontworpen) is om vanuit één backend aangestuurd te worden. Naast de backend adapter zal er dan een aparte klasse in het betreffende gokstelsel moeten worden gemaakt, die deze functionaliteiten wel kan bieden.

### 4. Afbakening

De opdracht zal zich beperken tot het ontwikkelen van de Game Management Service, de gebruikersinterface en de backendadapters. Voor het ondersteunen van GMS functionaliteiten kunnen aanpassingen nodig zijn in de poker of casino-software. Ingrijpende uitbreidingen van deze software zullen echter buiten de scope van dit project vallen, tenzij de auteur dat nodig acht voor het maken van het GMS.

Mocht het project eerder voltooid zijn dan gepland, kan de overige tijd besteed worden aan het implementeren van nieuwe functionaliteiten.



## 5. Projectwerkzaamheden

Tijdens de afstudeerstage moeten de volgende werkzaamheden worden verricht:

- Opstellen systeemeisen
- Bepalen van definitiestudie (IAD) (waarschijnlijk iedere module een pilot)
- Per pilot een Plan van Aanpak
- Uitzoeken van overeenkomsten en verschillen tussen beide backends met behulp van klasse diagrammen.
- Functies / methoden bepalen die in de interface aanwezig moeten zijn
- Ontwerpen van het Game Management Systeem
- Testplan opstellen (vaststellen van teststrategie, te gebruiken testmethode
- Poker en casino backend implementeren
- Game Management Systeem implementeren
- Gaming Management Service en Web User Interface maken (in JAVA / SEAM)
- Testen van de functionaliteit, met behulp van eerder geschreven testplan

## 6. Op te leveren (tussen)producten

Tijdens de afstudeerperiode moeten de volgende producten worden opgeleverd:

- Plan van Aanpak voor gehele project
- Definitiestudie
- Plan van Aanpak voor iedere pilot
- Klassendiagrammen van de bestaande software + overeenkomstige functies
- Lijst met overeenkomstige functies / methoden die geïmplementeerd gaan worden
- Ontwerp Game Management Systeem
- Backend interface (JAVA)
- Backend adapters voor C# pokersoftware & casinosoftware
- Game Management Systeem (JAVA)
- Game Management UI (JAVA / SEAM)
- Testrapport, met testresultaten van iedere pilotontwikkeling



## 7. Planning

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
1	8-2-2010	Planning maken, onderzoek naar user interface design methoden uitvoeren, te gebruiken java web framework vaststellen, werkplek en computer (software) klaar maken voor ontwikkeling. Logboek starten.		
2	15-2-2010	Huidige backoffice implementaties onderzoeken, Use Cases maken, definitiestudie doen	Onderzoeksrapport Plan van Aanpak	
3	22-2-2010	Pilot 1: Opstellen datamodel, Onderzoek C# <-> JAVA koppeling	Ontwerp	1
4	1-3-2010	Eerste versie van het datamodel + bijbehorende ui functionaliteit implementeren	Ontwerp	1
5	8-3-2010	Testplan maken, Mock server opzetten en tests uitvoeren	Testplan Testrapport	1
6	15-3-2010	Pilot 2: Use Cases / datamodel uitbreiden,		2
7	22-3-2010	Datamodel + GMSopzetten		2
8	29-3-2010	GMS functionaliteit uitbreiden		2
9	5-4-2010	GMS functionaliteit uitbreiden		2
10	12-4-2010	Testen geïmplementeerde code	Testrapport	2
11	19-4-2010	Pilot 3: Datamodel webinterface maken		3
12	26-4-2010	Implementeren van de webinterface		3
13	3-5-2010	Implementeren van de webinterface		3
14	10-5-2010	Testen geïmplementeerde code	Testrapport	3
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



## 8. Goedkeuring en ondertekening

De afstudeerder en de opdrachtgever gaan akkoord met de de beschreven aanpak en stemmen in zich in te zetten voor een succesvolle voltooiing van het project.

Naam afstudeerder:

Gertjan Al

Naam opdrachtgever:

Marco de la Rie





# Projectplanning

*Ontwikkeling van beheerssoftware voor online casino systeem*

*Afstudeerstage MC Advies & Management*

*Gertjan Al, 20069275*

*8-2-2010 t/m 4-6-2010*

## Planning versie 1 (8-2-2010)

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
1	8-2-2010	Planning maken, onderzoek naar user interface design methoden uitvoeren, te gebruiken java web framework vaststellen, werkplek en computer (software) klaar maken voor ontwikkeling. Logboek starten.		
2	15-2-2010	Huidige backoffice implementaties onderzoeken, Use Cases maken, definitiestudie doen	Onderzoeksrapport Plan van Aanpak	
3	22-2-2010	Pilot 1: Opstellen datamodel, Onderzoek C# <-> JAVA koppeling	Ontwerp	1
4	1-3-2010	Eerste versie van het datamodel + bijbehorende ui functionaliteit implementeren	Ontwerp	1
5	8-3-2010	Testplan maken, Mock server opzetten en tests uitvoeren	Testplan Testrapport	1
6	15-3-2010	Pilot 2: Use Cases / datamodel uitbreiden,		2
7	22-3-2010	Datamodel + GMS opzetten		2
8	29-3-2010	GMS functionaliteit uitbreiden		2
9	5-4-2010	GMS functionaliteit uitbreiden		2
10	12-4-2010	Testen geïmplementeerde code	Testrapport	2
11	19-4-2010	Pilot 3: Datamodel webinterface maken		3
12	26-4-2010	Implementeren van de webinterface		3
13	3-5-2010	Implementeren van de webinterface		3
14	10-5-2010	Testen geïmplementeerde code	Testrapport	3
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



## Planning versie 2 (17-2-2010)

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
1	8-2-2010	Planning maken, onderzoek naar user interface design methoden uitvoeren, te gebruiken java web framework vaststellen, werkplek en computer (software) klaar maken voor ontwikkeling. Logboek starten.	Plan van Aanpak	
2	15-2-2010	Huidige backoffice implementaties onderzoeken, 1 <sup>e</sup> definitiestudie, start ontwerp & implementatie 1 <sup>e</sup> pilot	Definitiestudie 1 Pilotontwerp 1	1
3	22-2-2010	Vervolg vorige week		1
4	1-3-2010	Testplan maken, Mock server opzetten en tests uitvoeren	Testplan Testrapport	1
5	8-3-2010	Pilot 2; settings & newsletters		2
6	15-3-2010	Vervolg vorige week		2
7	22-3-2010	Testen pilot 2	Testrapport	2
8	29-3-2010	Pilot 3; Transactions, Cashier & Financial reports		3
9	5-4-2010	Vervolg vorige week		3
10	12-4-2010	Testen pilot 3	Testrapport	3
11	19-4-2010	Pilot 4: Pitt boss & dashboard		4
12	26-4-2010	Vervolg vorige week		4
13	3-5-2010	Testen pilot 4	Testrapport	4
14	10-5-2010	Uitloop / extra functionaliteiten	Testrapport	5
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



## Planning versie 3 (8-3-2010)

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
1	8-2-2010	Planning maken, onderzoek naar user interface design methoden uitvoeren, te gebruiken java web framework vaststellen, werkplek en computer (software) klaar maken voor ontwikkeling. Logboek starten.	Plan van Aanpak	
2	15-2-2010	Huidige backoffice implementaties onderzoeken, 1 <sup>e</sup> definitiestudie, start ontwerp & implementatie 1 <sup>e</sup> pilot	Definitiestudie 1 Pilotontwerp 1	1
3	22-2-2010	Vervolg vorige week		1
4	1-3-2010	Testplan maken, Mock server opzetten en tests uitvoeren	Testplan Testrapport	1
5	8-3-2010	Pilot 2; Persons & login DB		2
6	15-3-2010	Testen pilot 2	Testrapport	2
7	22-3-2010	Pilot 3; Games & Tournaments		3
8	29-3-2010	Testen pilot 3	Testrapport	3
9	5-4-2010	Pilot 4: Financial		4
10	12-4-2010	Vervolg vorige week		4
11	19-4-2010	Vervolg en testen pilot 5	Testrapport	4
12	26-4-2010	Pilot 5: Reports & Fraud control		5
13	3-5-2010	Vervolg vorige week & testen pilot 5	Testrapport	5
14	10-5-2010	Uitloop / extra functionaliteiten	Testrapport	6
15	17-5-2010	Uitloop		
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



## Planning versie 4 (5-4-2010)

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
9	5-4-2010	Afstudeerverslag schrijven		
10	12-4-2010	Afstudeerverslag schrijven		
11	19-4-2010	Afstudeerverslag schrijven		
12	26-4-2010	Afstudeerverslag schrijven		
13	3-5-2010	Pilot 3; Game sessions, speler details	Definitiestudie, plan van aanpak	3
14	10-5-2010	Pilot 3: implementatie, testen	Testrapport	3
15	17-5-2010	Uitloop		3
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	

## Planning versie 5 (10-5-2010)

Week	Startdatum (ma)	Werkzaamheden	Producten	Iteratie
14	10-5-2010	Pilot 3; Game sessions, speler details	Definitiestudie, plan van aanpak	3
15	17-5-2010	Pilot 3: implementatie, testen	Testrapport	3
16	24-5-2010	Afstudeerverslag	Afstudeerverslag	
17	31-5-2010 4-6-2010	Afstudeerverslag <i>Inleverdatum afstudeerverslag</i>	Afstudeerverslag	



# Testplan

*Ontwikkeling van beheersoftware voor online casino systeem*

*Afstudeerstage MC Advies & Management*

*Gertjan Al, 20069275*

*8-2-2010 t/m 4-6-2010*

## Inhoudsopgave

1	Introductie .....	3
1.1	Overzicht nieuwe systeem .....	3
1.2	Doel van dit document .....	4
1.3	Doelen van de systeemtest .....	4
1.4	Invloed kwaliteitsbewaking .....	4
2	Scope en doelen .....	5
2.1	Testscope..... <b>Fout! Bladwijzer niet gedefinieerd.</b>	
2.1.1	Functionele tests .....	5
2.1.2	Integratietests .....	5
2.1.3	Kwaliteitstests .....	6
2.1.4	Eindacceptatie test .....	7
2.2	Testproces .....	7
3	Systeemtest planning .....	8
4	Personen en middelen.....	9
4.1	Personen.....	9
4.2	Middelen .....	9
5	Risico's en aannames .....	9
5.1	Risico's .....	9
5.2	Aannames .....	9



# 1 Introductie

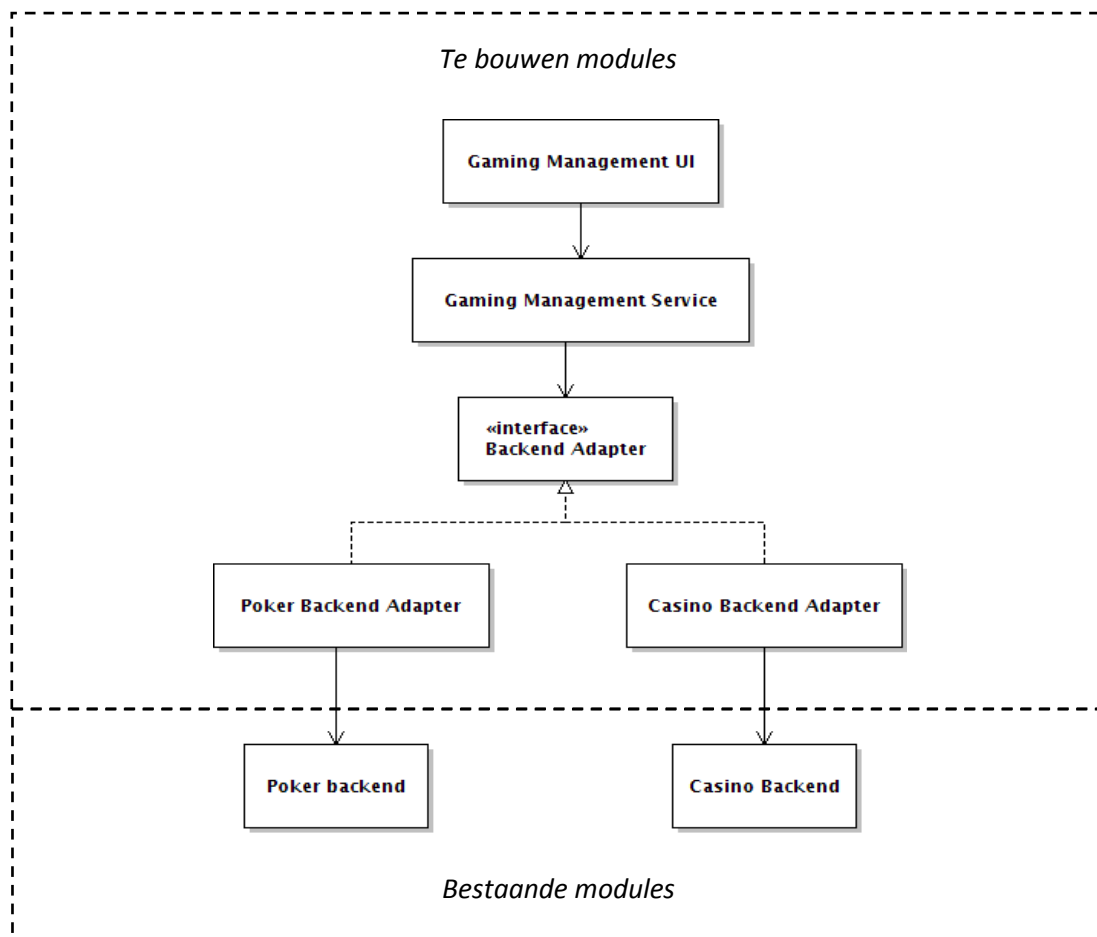
Dit testplan is geschreven voor het testen van het nieuwe Game Management Systeem voor het bedrijf Gaming & Gambling. Voor dit bedrijf wordt nieuwe software ontwikkeld waarmee de huidige online casino's en pokerspelen kunnen worden beheerd. Denk hierbij aan bijvoorbeeld het beheren en monitoren van de spelers van de verschillende casino- en pokerservers.

In dit testplan wordt gedefinieerd wat er getest gaat worden en hoe dit zal worden gedaan. Per pilot (mijlpaal) kan dit testplan gebruikt worden.

## 1.1 Overzicht nieuwe systeem

Met behulp van de Game Management Service (GMS) kan een collectie casino- en pokerservers worden beheerd. Iedere server heeft een eigen backoffice, waarin een manager (bijv. admin) de games, spelers e.d. kan beheren. Met behulp van een webinterface en de GMS moet de beheerder nu sneller meerdere casino- en pokerbackends kunnen beheren vanuit één webapplicatie.

De beoogde architectuur is als volgt schematisch weer te geven:



Figuur 1-1: Architectuur, uit: Plan van Aanpak





## 1.2 Doel van dit document

Met behulp van dit document kan er ter afsluiting van een pilot een systematische test worden uitgevoerd op de geïmplementeerde code. Per geteste pilot zal er dan ook een testrapport verschijnen met de doorlopen tests en de respectievelijke testresultaten.

## 1.3 Doelen van de systeemtest

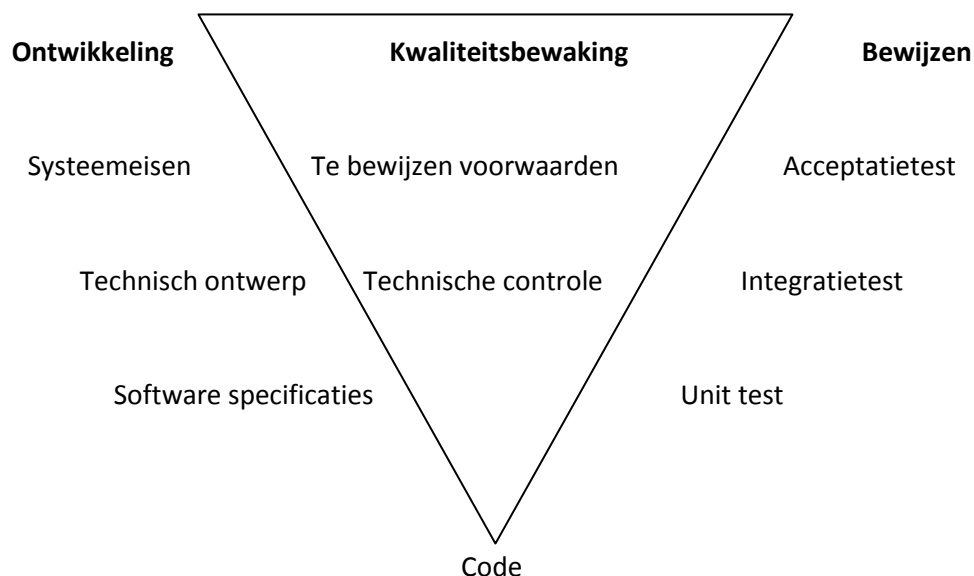
Met het uitvoeren van de tests wordt getracht te bewijzen dat de opgeleverde software:

- Functioneel is, zoals beschreven in de systeemeisen van de pilots.
- Van hoge kwaliteit is; de software is productiever dan de voorgaande backendsoftware en voldoet aan de software-eisen opgesteld door het bedrijf.

*Gedetailleerde doelen worden later in dit document beschreven*

## 1.4 Invloed kwaliteitsbewaking

In onderstaand model is te zien hoe de kwaliteitsbewaking bij veel softwarebedrijven is verdeeld. Dit GMS project wordt echter door één persoon gedaan, waardoor de scheidslijnen enigszins wegvallen. Wel is hierin te zien onder welke groep de verschillende tests horen.



## 2 Scope en doelen

Als testmethode is er gekozen voor TMap (Test Management Approach), ontwikkeld door het Nederlandse ICT-bedrijf Sogeti. De auteur heeft deze methode bij voorgaande projecten gebruikt en deze is goed bevallen voor gebruik bij iteratief ontwikkelen.

TMap bestaat uit 4 delen:

- Fasering van testactiviteiten (wat er getest gaat worden en wanneer)
- Technieken bruikbaar voor testactiviteiten (hoe er getest gaat worden)
- Infrastructuur (waarmee er getest gaat worden)
- Organisatie (wie er wat gaan testen)

TMap past prima in de IAD methode; er kan getest worden na voltooiing van een pilot en ter afsluiting van het project. Bij TMap wordt gesproken van white-box tests en black-box tests. Een white-box test wordt uitsluitend uitgevoerd door de ontwikkelaar. De tests zullen worden afgestemd op de broncode van de ontwikkelde software. Bij een black-box test wordt vooral getest of een functionaliteit naar behoren werkt. Deze test kan door iedereen worden uitgevoerd. De tests die in dit project zullen worden toegepast zijn hieronder toegelicht:

### 2.1 Uit te voeren tests

#### 2.1.1 White-box test: unit tests

Bij een unit test worden de verschillende stukken broncode (units) getest op de correcte werking. Het doel is om deze units zo onafhankelijk mogelijk van elkaar te testen, gebruikmakend van gegenereerde data en / of objecten. Deze test zal na iedere pilot worden gedaan, om te bewijzen dat de opgeleverde code naar behoren werkt. Dit is een white-box test die door de programmeur wordt uitgevoerd; inhoudelijke kennis van de broncode wordt gebruikt om de tests op te stellen.

#### 2.1.2 Black-box test: functionaliteit

Het doel van deze test is het aantonen van de functionele eisen, zoals opgesteld in het plan van aanpak van iedere pilot. In dit stadium worden ook de validatietests gedaan, waar de validiteit van de invoervelden in de webinterface wordt onderzocht. Zie hiervoor hoofdstuk 2.1.4.2.

#### 2.1.3 Black-box test: integratie

Er wordt low-level getest op functionaliteit. Hiervoor worden de dataflows doorlopen zoals beschreven in het document over usage-centered design. Deze dataflows beschrijven de verschillende rollen en hun respectievelijke taken.



## 2.1.4 Kwaliteitstests

De Game Management Service is de laag tussen de gebruikersinterface (webui) en de verzameling casino- en pokerservers. Op deze servers draait de software en is de data opgeslagen van de games, spelers enz. Per manager kunnen backends worden toegewezen die deze kan beheren.

### 2.1.4.1 White-box test: uitbreidbaarheid

Deze test zal worden uitgevoerd tegen het einde van het project. Met behulp van de documentatie zal een andere programmeur moeten proberen het systeem uit te breiden. Deze zal moeten proberen de software uit te breiden. Denk hierbij aan het aanmaken en toevoegen van een nieuw backend, of het toevoegen van een nieuwe webpagina. Er zal documentatie worden opgeleverd die beschrijft wat voor stappen er moeten worden doorlopen.

De meetresultaten uit deze test zijn niet kwantitatief, er kan geen percentage “uitbreidbaarheid” berekend worden. In plaats daarvan kan de tweede programmeur aangeven welke passages uit het document moeten worden verbeterd of uitgebreid. Op deze manier zou een derde programmeur genoeg informatie hebben om de software uit te breiden in een later stadium.

### 2.1.4.2 Black-box test: validiteit

De Game Management Service zal naar alle waarschijnlijkheid een hoop invoervelden bevatten, zoals gebruikt bij het bewerken van de personalia van een persoon of het afhandelen van een probleemmelding. Met de validiteitstest kan gecontroleerd worden of de invoer in deze velden juist wordt verwerkt en weer op de juiste manier wordt getoond bij opnieuw bewerken van het invoerveld. Een voorbeeld hiervan is het opslaan van een datum. De invoervelden zijn *dag*(numeriek), *maand*(tekst selectiebox) en *jaar*(numeriek). De opslag van deze waarden in de database is een varchar(8), namelijk {jaar}{maand}{dag}. De backends voor poker en casino slaan hun gegevens op in een externe database. De databases (Microsoft SQL Server en Sybase) zijn direct gekoppeld aan de backends, of via methoden in de bronsoftware (software specifieke backoffices / adminpanels). Met de validiteitstest kan dan worden gecontroleerd of bij opnieuw bewerken van het jaar de waarden goed geparsed zijn.

### 2.1.4.3 Black-box test: gebruiksvriendelijkheid

In het document over usage-centered design wordt beschreven welke soorten managers er zijn (financieel, games, marketing). Met behulp van de uit te voeren taken van deze personen moet worden gecontroleerd of de dataflows van de GMS webinterface gebruiksvriendelijk zijn. Eventueel kan een testpersoon de taken proberen te vervullen. Het doel is dat de gebruiker zijn taken (snel) kan volbrengen zonder hulp van de ontwerper. Denk hierbij aan het vinden van gerelateerde taken in de sidebar. Het streven is om dit met zo min mogelijk muisclicks hoeven doen.

## 2.1.5 White-box test: load testing



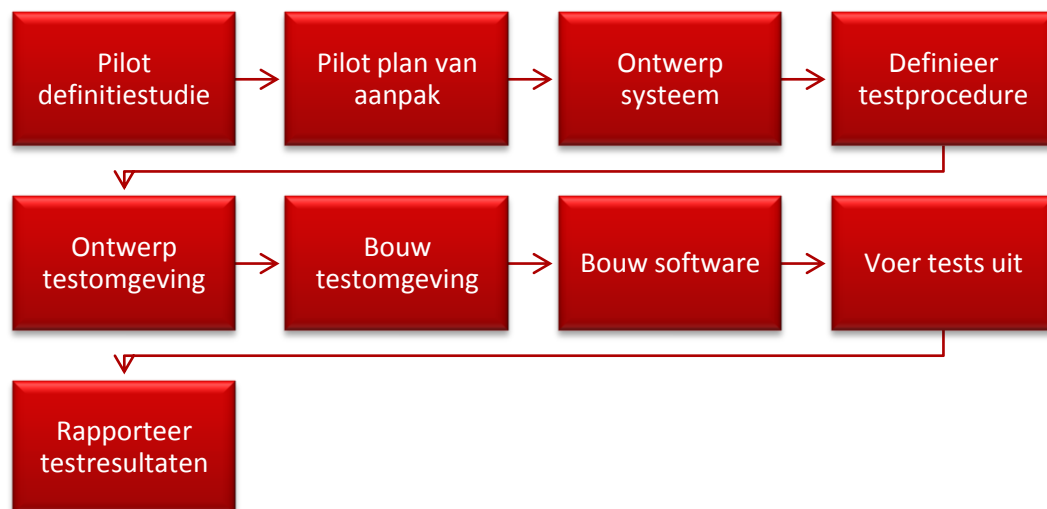
Het GMS wordt gebouwd om te voldoen aan de groter wordende hoeveelheid gameservers. Er zullen in een groot aantal landen online casino's geopend worden, die allemaal via het GMS worden beheerd. Met deze test zal onderzocht worden wat het effect is van een grote datahoeveelheid op het systeem, door bijvoorbeeld het verband tussen de databasegrootte en de laadtijd van een webpagina te meten. Deze test zal worden uitgevoerd tegen het einde van het project, om de instellingen te fine-tunen. De test kan gedaan worden met behulp van een profiler, welke de verstreken tijd kan berekenen. Het is bij deze test vereist om kennis te hebben van de onderliggende code, zodat profiler kan tonen welke functieaanroepen relatief veel tijd in beslag nemen.

### 2.1.6 Black-box test: uiteindelijke acceptatie test

Ter afsluiting van een pilot of het gehele project kan een acceptatietest gedaan worden. Deze test bestaat uit een checklist waarmee de aanwezigheid van functionele eisen kan worden vastgesteld. Deze test zal worden uitgevoerd door het team wat de software gaat installeren en door personen die het systeem daadwerkelijk gaan gebruiken. Deze laatste gebruikers kunnen hun eigen testcriteria opstellen en zelf deze tests uitvoeren.

## 2.2 Testproces

Per pilot wordt een proces doorlopen. Per pilot worden de tests uitgevoerd en beschreven in een testrapport. Hieronder een weergave van de procescyclus.



## 2.3 Systeemtest planning

Aan het einde van iedere pilot worden tests uitgevoerd. Met deze tests kan worden aangetoond dat de geïmplementeerde code voldoet aan de eisen van de pilotdefinitie. Eventuele negatieve testresultaten worden opgenomen in de evaluatie van de daaropvolgende pilot, waarin de problemen of oneffenheden worden onderzocht en behandeld.

Hieronder een overzicht van de pilots en de bijbehorende testmethoden. Dit zijn de tests die worden uitgevoerd tijdens een pilot.

Pilot	Omschrijving	Testmethoden
Eerste pilot	Proof of Concept	Geen test, code wordt herzien in 2 <sup>e</sup> pilot
Tweede pilot	Persoonsweergaven / UI	Validatietest
		Unit tests
Derde pilot	Spelerdetails, managers en gamesessies	Validatietest
		Unit test
Eindpilot	Afronding project	Test gebruikersvriendelijkheid
		Load test
		Uitbreidbaarheid test
		Functionele test
		Integratietest



## 3 Personen en middelen

### 3.1 Personen

Voor drie tests zijn externe medewerkers nodig. Deze personen zullen meewerken aan de tests rond uitbreidbaarheid, gebruiksvriendelijkheid. De opdrachtgever zal als afsluiting de functionele test doen.

### 3.2 Middelen

Voor de tests is de volgende hardware nodig:

- Ontwikkel PC
- Netbeans IDE 6.8 of hoger
- Glassfish voor het lokaal draaien van de GMS
- Eventuele testtools van Netbeans

## 4 Risico's en aannames

### 4.1 Risico's

De hoeveelheid werk voor dit project is groot. Hoogstwaarschijnlijk zullen niet alle taken kunnen worden volbracht binnen de geplande tijd. Mocht dit project niet tijdig afgerond kunnen worden, kan een volgende programmeur of volgend team dit project voortzetten en testen met behulp van dit testplan.

### 4.2 Aannames

De volgende aannames worden genomen bij aanvang van een test:

- Software is van goede kwaliteit
- Alle gevonden fouten zullen worden verholpen voordat de software opnieuw getest wordt; tests mogen niet worden beïnvloed door reeds eerder aangetoonde fouten.
- Benodigde broncode is beschikbaar voor de tests
- De software draait naar behoren op een ontwikkel pc, met minimaal Windows XP



# Systeemontwerp

*Ontwikkeling van beheersoftware voor online casino systeem*

*Afstudeerstage MC Advies & Management*

*Gertjan Al, 20069275*

*8-2-2010 t/m 4-6-2010*

## Inhoudsopgave

1	Inleiding .....	5
2	Systeemeisen.....	5
2.1	Ondersteuning en beheer van de huidige functionaliteiten .....	5
2.1.1	Functionaliteiten casino managementsysteem.....	6
2.1.2	Functionaliteiten poker managementsysteem .....	7
2.2	Systeemeisen aan de hand van vergadering eindgebruikers.....	8
2.3	Functionele eisen voor de Game Management Service .....	9
2.3.1	Binnen de scope .....	9
2.3.2	Buiten de scope .....	10
2.4	Niet functionele eisen voor de Game Management Service .....	10
3	Gebruik van usage-centered design .....	11
4	Use cases .....	12
5	Task cases .....	15
5.1	Monitortaken .....	15
5.1.1	Snel monitoren van meerdere backends, onder normale omstandigheden .....	16
5.1.2	Snel monitoren van meerdere backends, met stortingen .....	16
5.1.3	Snel monitoren van meerdere backends, met stortingpogingen .....	16
5.1.4	Nagaan van nieuw gerapporteerde problemen .....	17
5.1.5	Afsluiten spel .....	17
5.1.6	Controleren van stortingpogingen .....	17
5.2	Uitbetaling gerelateerde taken .....	18
5.2.1	Aanmaken van eerste cashout request (aanvraag uitbetaling) .....	18
5.2.2	Aanmaken van uitbetalingaanvraag, tijdens openstaande aanvraag .....	18
5.2.3	Inzien van uitbetalingstatus .....	18
5.2.4	Annuleren van een uitbetaling .....	19
5.2.5	Opvragen nieuwe uitbetalingaanvraag .....	19
5.2.6	Afwijzen aanvraag tot uitbetaling .....	19
5.2.7	Opzoeken uitbetaling .....	20
5.2.8	Accepteren van aanvraag tot uitbetaling .....	20
5.3	Game management.....	20





5.3.1	Monitoren van uitgaven per game.....	20
5.3.2	Controleren game sessies.....	20
5.3.3	Controleren uitkomst in slotmachine sessie .....	21
5.4	Speler management .....	21
5.4.1	Controleren uitkomst in slotmachine sessie .....	21
5.4.2	Speleroverzicht, gebruikmakend van opgeslagen filter .....	21
5.4.3	Controleren van nieuw aangemelde spelers, die nog niet ingelogd zijn .....	21
5.4.4	Controleer status van registratie-email.....	21
5.4.5	Toestaan van bankaccount van speler .....	22
5.4.6	Toevoegen van een bonus.....	22
5.5	Probleem (issue) management .....	22
5.5.1	Overzicht met problemen (issues) .....	22
5.5.2	Kiezen van een probleem .....	22
5.5.3	Oplossen van een betalingsprobleem .....	23
5.5.4	Afronden van een betalingsprobleem.....	23
5.6	Algemene taken.....	24
5.6.1	Probleem melden .....	24
5.6.2	Antwoorden op een probleemmelding.....	24
5.7	Marketinggerelateerde taken .....	25
5.7.1	Nieuw bonusprogramma aanmaken.....	25
5.7.2	Nieuwe bonusregel aanmaken.....	25
5.7.3	Overzicht van huidige bezoekers.....	25
6	Task map.....	26
7	Abstracte prototypes.....	28
7.1	Speleroverzicht.....	28
7.2	Affiliate overzicht .....	29
7.3	Speloverzicht .....	29
7.4	Stortingen .....	30
7.5	Verzoeken tot uitbetaling.....	30
7.6	Bonusprogramma's .....	31
7.7	Probleem / taakoverzicht .....	31



8	Gebruikersinterface.....	32
8.1	Te implementeren componenten .....	32
8.1.1	Probleemrapportage .....	32
8.1.2	Probleemmeldingen overzicht .....	32
8.1.3	Probleemmelding beantwoorden .....	33
8.1.4	Probleem in behandeling .....	33
8.1.5	Automatische antwoorden.....	34
8.1.6	Status overzicht .....	34
8.1.7	Status detail.....	34
8.1.8	Zoeken van speler(s).....	35
8.1.9	Proces en voortgangsoverzicht .....	35
8.1.10	Proces stappen afhandeling .....	36
8.1.11	Wisselen van backend .....	36
8.1.12	Huidige uitbetaling .....	37
8.1.13	Wijzigen van werkomgeving.....	37
8.1.14	Dashboard .....	37
8.1.15	Menu .....	37
8.1.16	Spelerslijst filter .....	38
8.1.17	Visualisatie van spelsessie.....	38
8.1.18	Relevante links.....	39
8.1.19	Opzoeklijst .....	39
8.1.20	Spelerdetail.....	40
8.1.21	Spelsessies .....	42
8.1.22	Sessie details.....	43
8.2	Prototypes .....	44
8.2.1	Navigatiemap.....	44
8.2.2	Paginaindeling .....	46



# 1 Inleiding

In dit document wordt beschreven wat de eisen voor de Game Management Service zijn, en hoe deze zullen worden ontworpen en geïmplementeerd.

## 2 Systeemeisen

Een (software)project wordt altijd gestart met een bepaald doel. Voorafgaand aan een project worden dit doel samen met de systeemeisen gespecificeerd, zodat de softwareontwikkelaar weet wat er gerealiseerd moet worden en de opdrachtgever weet wat er aan het einde van het project opgeleverd zal worden. De opdrachtgever heeft echter niet altijd een concreet plan over de eisen van het eindproduct. De softwareontwikkelaar moet in dit geval in samenwerking met de opdrachtgever specificeren wat deze systeemeisen zijn.

De gebruikersomgeving van de Game Management Service moet dezelfde mogelijkheden bieden als het casino en poker managementsysteem. Deze webpagina's bevatten alle belangrijke functionaliteiten om de backends te beheren. De gebruikersomgeving voor het GMS moet een aggregatie van deze twee managementsystemen worden, zodat er geen functionaliteit verloren gaat.

Veel functionaliteiten in de twee managementsystemen komen overeen. Alle mogelijkheden moeten via de GMS gebruikersomgeving te benaderen zijn. Naast deze eisen is het bekend dat de huidige gebruikers ideeën hebben over verbeteringen en nieuwe functionaliteiten. Deze zullen ook meegenomen moeten worden in het nieuwe ontwerp.

### 2.1 Ondersteuning en beheer van de huidige functionaliteiten

Op de volgende pagina's is een overzicht van de menu's uit het casino managementsysteem en het poker managementsysteem te vinden. Hoewel de omschrijvingen niet altijd overeenkomen zijn de volgende belangrijke onderdelen in beide systemen aanwezig:

- Een systeem bevat spelers
- Een systeem bevat spellen
- Een systeem bevat affiliates
- Een systeem bevat geldtransacties, zowel stortingen als aanvragen voor uitbetalingen
- Verschillende manieren om de klanten te bereiken of te woord te staan
- Mogelijkheid om bonusprogramma's aan te maken
- Verschillende manieren om valsspelen te detecteren of tegen te gaan
- Mogelijkheid om verschillende soorten rapporten (reports) te genereren.



## 2.1.1 Functionaliteiten casino managementsysteem

De gebruikersomgeving van het casino bevat de volgende mogelijkheden:

### *Settings*

- End game
- Add manager
- Reporting period
- Manager information
- Game options
  - View Game history
  - Set game parameters
- Jackpot control center
  - Jackpot games summary
  - Activity simulator settings

### *Transactions*

- Transaction rates
  - Set payment rates
  - Set bonus rates
- View transactions
  - View suspended transactions
  - View completed transactions
- Fax confirmation
- Set fax confirmation limit

### *Users*

- Winning users
- View visitor stats
- View suspicious accounts
- Add users
- Losing users
- Users information
- Online logs
- Visit logs

### *Affiliates*

- View payments
- Affiliate information
- Affiliate summary
- Commision setting
  - Default values
  - For current affiliates

### *Cashier*

- Deposits
- Cashout
- Balance adjustment
- Bonuses

### *Financial report*

- General report
- Report on casino
- Report on users
- Report on games
- Report on affiliates

### *Newsletter*

- Create mailing list
- Message templates
  - View message templates
  - Create message template
- Set default messages
  - Signup message
  - Deposit message
- Email preferences

### *Pit boss*

- Restrictions
  - View country restriction
  - Add country restriction
  - View credit card restriction
  - Add credit card restriction
  - View player restriction
  - Add player restriction
  - View domain restriction
  - Add domain restriction
- Fraud control
  - Black list
    - Black list summary
    - Black list settings
  - Suspicious values
  - View suspicious players
  - Fraud transactions

### *Marketing*

- Create campaign
- View campaign info
- Summary campaign info
- List bonus codes
- Create bonus code



## 2.1.2 Functionaliteiten poker managementsysteem

De gebruikersomgeving van de pokersoftware bevat de volgende mogelijkheden:

### *User Management*

- Register Tournament
- Unique Avatar
- UnRegister Tournament
- View Players

### *Admin Management*

- Account Override
- Deposit Point Rule
- Email Marketing
- Email Templates
- Foul Language Filter
- Generate Random Number
- Manage Groups
- Manage Operator
- Manage Role
- Real Hand Point Rule
- Scrolling Message
- Send Message
- View Game Servers

### *Account Management*

- Transfer Money

### *Games Management*

- Add Tournaments
- Assign Table Logo
- Create Table
- Detect Collusions
- Game Icons

### *Reports*

- Affiliate Signups
- Bonus Report
- Deposits
- Payouts
- Players Point
- Players Signups
- Rake Report
- Reconciliations
- Tournaments Report
- Users Online



## 2.2 Systeemeisen aan de hand van vergadering eindgebruikers

Voor een aanvulling op de systeemeisen is er een vergadering geweest met twee eindgebruikers. Momenteel wordt alleen het Casino Management Systeem gebruikt, waarin de eindgebruikers graag nog enkele verbeteringen in zouden zien. De functionaliteiten uit de casino gebruikersinterface zullen dus samen met deze verbeteringen beschikbaar moeten zijn in de gebruikersinterface van de Game Management Service.

De vergadering is gehouden om de nieuwe ideeën te bespreken en de voortgang en toekomstvisie van het casino. Niet alle informatie was relevant voor het GMS project, maar de volgende ideeën over het GMS zijn tijdens de vergadering geopperd:

- **Geavanceerde bonus berekening;** het is nu wel mogelijk om bonussen toe te passen, maar bij een aanvraag tot uitbetaling is het niet direct mogelijk om de toegepaste bonussen te controleren. Er moet dus bij iedere uitbetaling gekeken worden van welke bonusprogramma's gebruik is gemaakt tijdens de speelperiode, waarna kan worden gecontroleerd of het aangevraagde bedrag rechtmatig is. Een overzicht van de toegepaste bonussen per aanvraag of per speler zou een hoop tijdrovend werk schelen.
- **Ticketing support;** een onderdeel waardoor de e-mailconversaties tussen gebruikers en beheerders vervangen zal worden door een intern berichten systeem, waaraan de meldingen "in behandeling", "afgehandeld" etc. kunnen worden gegeven. Dit is nodig omdat het aantal beheerders ook zal meegroeien met het aantal domeinen. Op deze manier kunnen de tickets worden toegewezen en is het voor een klant duidelijk of hun ticket in behandeling is.
- **Webapplicatie geschikt voor weergave op mobiele platforms;** in het speciaal de Google G1 Android phone. Deze telefoon heeft een resolutie van 320 x 480 pixels. De huidige casino backoffice is lastig te bedienen via deze telefoon, aangezien de menustructuur bestaat uit vele mouse-overs. Hier zal rekening mee moeten worden gehouden bij het ontwerpen van de GUI. Als de schaalbare (mobiele) resolutie de mogelijkheden van de webbrowser-versie teveel aantast, zal in een volgend project een gebruikersomgeving moeten worden gemaakt voor mobiele telefoons. De onderliggende software kan hierbij ondersteuning bieden, door bijvoorbeeld gebruik te maken van een Model View Controller model. In dit geval hoeft er dus enkel een nieuwe View gemaakt te worden.



## 2.3 Functionele eisen voor de Game Management Service

In samenwerking met de opdrachtgever zijn de volgende eisen opgesteld;

### 2.3.1 Binnen de scope

- Een Game Management Service
  - Heeft een backend verzameling
  - Krijgt later functionaliteiten zoals beschreven in hoofdstuk 2.3.2
- Backendadapters voor Casino en Poker software
  - Zijn een universele wrapper tussen de Game Management Service en de achterliggende gok-software
- Uit beide backendadapters kan de volgende data worden opgehaald:
  - Lijst met spelers met kolommen NAW, balans, inloggegevens
  - Lijst met winnende spelers binnen een bepaald datumbereik
  - Lijst met verliezende spelers binnen een bepaald datumbereik
  - Lijst met spellen
  - Lijst met affiliates
  - Lijst met bonusprogramma's
  - Lijst met beheerders en hun rollen
- In de backendadapters kunnen de volgende acties worden uitgevoerd:
  - Speler verwijderen
  - Speler toevoegen
  - Speler bewerken
  - Speler zoeken
    - Zoekvak email en inlognaam
    - Uitgebreide zoekfilter, op te slaan voor hergebruik
  - Blokkeren van een speler
  - Spel herstarten
  - Spel uitschakelen
  - Affiliate toevoegen
  - Affiliate bewerken
  - Affiliate verwijderen
  - Bonusprogramma toevoegen
    - Bonusregel toevoegen
  - Beheerder toevoegen
  - Beheerder rol toewijzen
  - Beheerder verwijderen
- Een webinterface voor de Game Management Service
  - Aangeboden functionaliteiten afhankelijk van de rol van de ingelogde beheerder
  - De backenddata kan hierin getoond worden
  - De backendacties zijn hierin aan te roepen



### 2.3.2 Buiten de scope

De volgende systeemeisen behoren wel bij het eisenpakket van de Game Management Service, maar zullen niet tijdens dit project worden geïmplementeerd. Tijdens het project dient echter wel rekening gehouden te worden met deze toekomstige functionaliteiten. Aan deze eisen dient voldaan te worden in een opvolgend project.

- Implementatie van een betaalsysteem in Game Management Service, welke kan worden aangeroepen vanuit de backends.
- Implementatie van de geavanceerde bonus berekeningen (zie hoofdstuk 0)
- Een ticket / issue systeem in de Game Management Service voor helpdesk doeleinden, waarmee:
  - Berichten kunnen worden geschreven en ontvangen van en naar gebruikers en andere medewerkers
  - Probleemmeldingen (uit berichten) kunnen worden toegewezen aan medewerkers
  - Medewerkers een lijst met taken kunnen opvragen
  - Medewerkers de status van een taak kunnen aanpassen in “afgehandeld”, “in behandeling”, “afgewezen” e.d.
- Gebruikersinterface voor gebruik in webbrowsers op mobiele telefoons, getoond in een resolutie van 320 x 480 pixels (Google G1 Android Phone)

### 2.4 Niet functionele eisen voor de Game Management Service

- Een document met de systeemeisen
- Systeemontwerp volgens de methode usage-centered design:
  - Use cases
  - Task cases
  - Task map
  - Abstracte prototypes
  - Mockups gebruikersinterface
- Klassendiagrammen van Game Management Systeem en backendadapter
- Software wordt ontwikkeld in het Seam framework, draaiend op een Glassfish server.
- Een testplan voor het testen van de geproduceerde broncode
- Testrapport(en) van de ontwikkelde software
- Overdrachtsdocument waarin de ontwikkelde softwareonderdelen worden beschreven





### 3 Gebruik van usage-centered design

Voor het ontwerpen van het Game Management Systeem is inzicht nodig in de behoeften van gebruikers en hoe deze te werk gaan. Veelal wordt er bij softwareontwerpen gebruik gemaakt van de methode *user-centered design*, waarbij de eisen van de gebruikers worden omgezet in use cases. Deze use cases worden dan gebruikt om een visueel ontwerp (mockups) te maken van het eindproduct.

Een groot nadeel van gebruik van user-centered is dat de normale eisen en workflows worden overschreeuwd door nieuwe wensen van de gebruikers. Zo krijgt de softwareontwikkelaar een vervormd beeld van wat het systeem werkelijk moet kunnen, omdat de focus dan ligt op het tevreden stellen van de gebruikers. Vaak moet de software later meerdere malen aangepast worden omdat de gebruikers functionaliteiten missen, waarvan ze hadden verwacht dat die wel zouden worden geïmplementeerd. De ontwerpers op hun beurt komen er in deze latere stadia achter dat hun ontwerp niet meer is aan te passen aan deze (voor gebruikers) normale functionaliteiten. Vaak worden de functionaliteiten toegevoegd, zonder dat over het ontwerp is nagedacht, of het is simpelweg niet mogelijk om de functies toe te voegen. Dit resulteert in onvolledige software, of software met lagere kwaliteit (snelheid, robuustheid, onderhoudbaarheid).

Er is gekozen om voor dit project gebruik te maken van de ontwerpmethode *usage-centered design*. Bij deze methode staat niet zozeer de gebruiker, maar de workflows centraal. Het Game Management Systeem zal gebruikt worden door verschillende soorten medewerkers, ieder met zijn of haar eigen verantwoordelijkheden en taken. Een medewerker kan meerdere rollen hebben in een systeem, zoals bijvoorbeeld een financiële en marketing gerelateerde rol. Een medewerker heeft dus één of meerdere rollen, een rol heeft één of meerdere taken, een taak heeft één of meerdere workflows om te taak te volbrengen.

Met behulp van de usage-centered design methode wordt per rol beschreven wat de omschrijving, context, kenmerken en criteria zijn (zie hoofdstuk 4). Nadat van alle rollen de use cases beschreven zijn, worden de task cases gemaakt. Deze task cases (gebaseerd op de use cases) beschrijven de interactie tussen de gebruiker en het systeem voor een bepaalde taak.

Met de task cases kan een task map (hoofdstuk 6) worden gemaakt, waar uiteindelijk de abstracte prototypes op worden gebaseerd (hoofdstuk 7). Met behulp van die prototypes kan de uiteindelijke gebruikersinterface (hoofdstuk 8) worden ontworpen.



## 4 Use cases

### Normale beheerdersrol, tijdens kantooruren

<b>Beschrijving</b>	Verantwoordelijk voor het normaal functioneren van het casino, controleert gedrag van spelers om valsspelers tijdig te signaleren, zorgt dat alles naar behoren werkt.
<b>Context</b>	Op kantoor, gebruikmakend van desktop pc
<b>Kenmerken</b>	Veel voorkomende taak, deels ongestructureerd proces, identificeren van valsspelers is gebaseerd op ervaring.
<b>Criteria</b>	Snel overzicht hebben over backend status (games, winning players)

### Beheerdersrol buiten kantooruren

<b>Beschrijving</b>	Support medewerker, werkend buiten kantooruren, met dezelfde taken als de <i>normal office hours operator</i> . Veelal de taak van monitoren, niet zozeer het uitvoeren van taken. In geval van geconstateerde fouten (bijvoorbeeld vastgelopen games) wordt een stand-by medewerker ingeschakeld.
<b>Context</b>	Controle wordt buiten kantooruren uitgevoerd, gebruikmakend van een mobiele webbrowser op een PDA.
<b>Kenmerken</b>	Veel voorkomende taak, uitgevoerd buiten kantooruren. Mogelijk stressvolle omgeving.
<b>Criteria</b>	Snel overzicht hebben over backend status (games, winning players), ondersteuning voor mobiele webbrowsers nodig.

### Spelers ondersteunende rol

<b>Beschrijving</b>	Verzorgt de ondersteuning van spelers, bijvoorbeeld verwerken van meldingen over vastgelopen gamesessies en beantwoorden van vragen rondom casino promotieacties.
<b>Context</b>	Op kantoor, gebruikmakend van desktop pc waarop de emails van gebruikers worden ontvangen en afgehandeld.
<b>Kenmerken</b>	Deels ongestructureerd, mogelijk herhalende taken. Snelle afhandelingen of oplossingen zijn belangrijk voor het tevreden houden van de klanten.
<b>Criteria</b>	Behandelen van fouten en issue tracking moeten goed verlopen



<b>Casino speler</b>	
<b>Beschrijving</b>	Een speler van het casino
<b>Context</b>	Thuiscomputer, besturingssysteem en configuratie daarvan verschilt per speler
<b>Kenmerken</b>	Wil spelen om te winnen, kan gestrest raken bij verlies
<b>Criteria</b>	Gamesoftware zonder fouten of haperingen, effectieve hulp door <i>Player support</i> bij fouten of vragen

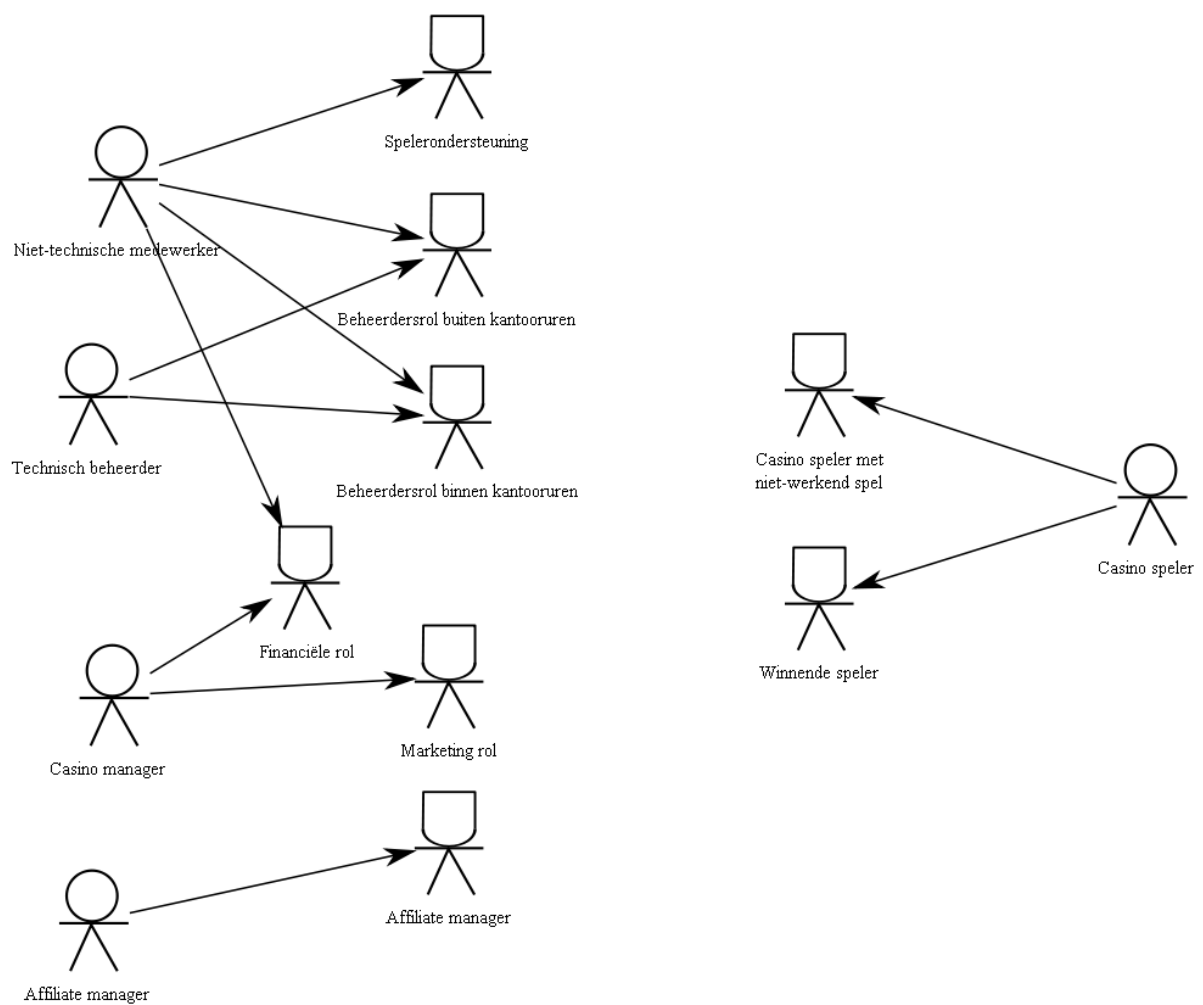
<b>Casino speler met niet-werkend spel</b>	
<b>Beschrijving</b>	Speler die een (ongegrond) probleem heeft ondervonden bij een spel
<b>Context</b>	Thuiscomputer, besturingssysteem en configuratie daarvan verschilt per speler
<b>Kenmerken</b>	Gefrustreerd, omdat er wellicht geld is verloren bij het optreden van de fout
<b>Criteria</b>	Mogelijkheid om snel geholpen te worden door <i>Player support</i>

<b>Financiële rol</b>	
<b>Beschrijving</b>	Vervult cashout requests (aanvraag tot uitbetaling) en verzorgt bonussen voor spelers die een fout hebben ontdekt of als promotie
<b>Context</b>	Op kantoor, gebruikmakend van desktop pc en kennis van de regels rondom uitbetaling
<b>Kenmerken</b>	Uitbetalingen moeten handmatig worden gecontroleerd op fraude, veel verantwoordelijkheid i.v.m. berekenen van de hoeveelheid geld een speler heeft en kan opvragen
<b>Criteria</b>	Uitgebreid overzicht van de geldtransacties en fluctuaties van een speler

<b>Marketing rol</b>	
<b>Beschrijving</b>	Onderzoekt de mate van effectiviteit van een marketingcampagne
<b>Context</b>	Op kantoor, als onderdeel van een groter marketingproces. Gebruikmakend van de data die met de campagne te maken heeft.
<b>Kenmerken</b>	Combineren van grote hoeveelheden data, op zoek naar trends in gamegebruik en spelgedrag
<b>Criteria</b>	Beheren van verschillende soorten data, mogelijkheid tot het exporteren van data

<b>Affiliate manager rol</b>	
<b>Beschrijving</b>	Behandelen van aanvragen van affiliates en contact houden met deze affiliates om ze te behouden voor de marketing van het casino
<b>Context</b>	Op kantoor, gebruikmakend van overzicht van affiliates.
<b>Kenmerken</b>	Veel contact met affiliates, controleert gedrag van affiliates
<b>Criteria</b>	Controleren van nieuwe affiliates, verdelen van taken tussen de verschillende affiliate managers, identificeren van affiliates die zich niet aan de regels houden.





**Figuur 4-1: Rolverdeling van de verschillende soorten medewerkers**

## 5 Task cases

Task cases beschrijven de interactie tussen de gebruiker en het systeem, bij het volbrengen van een bepaalde taak. Deze zijn gebaseerd op de eerder beschreven use cases.

### 5.1 Monitortaken

De task cases gaan over het monitoren of beheren van één of meerdere backends. In de cases wordt er van uit gegaan dat een backend data kan verschaffen over de volgende modules:

- Stortingen (deposits)
  - Inclusief nieuwe stortingen
  - Mislukte of openstaande stortingen (geselecteerd aan de hand van een overschrijding van een vooraf ingestelde tijds marge)
- Bezoekers die recentelijk een game gespeeld hebben
- Bezoeker logs
  - Zowel ongeregistreeerde als geregistreeerde bezoekers
- (Nieuwe) probleem meldingen (issue reports)
  - Van spelers
  - Van de backend, in geval van exceptions of warnings
- Overzicht van uitbetalingen van recentelijk gespeelde games (gesorteerd op uitbetaling)
- Winnende spelers

Om herhaling te voorkomen is de identificatie en verificatie van gebruikers uit de task cases gehouden. Voordat een gebruiker een sessie kan starten moet deze eerst inloggen bij het systeem, wat na acceptatie de backends toewijst waar toegang voor is verleend.



### 5.1.1 Snel monitoren van meerdere backends, onder normale omstandigheden

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer backend / startpagina	Toon samenvatting van geselecteerde backend
Selecteer een samenvatting van een backend	Samenvatting met de belangrijkste onderdelen van één enkel backend

### 5.1.2 Snel monitoren van meerdere backends, met stortingen

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer backend / startpagina	Toon samenvatting van geselecteerde backend
Controleert details van stortingen	Mogelijkheid tot opvragen van details rondom een stortingsaanvraag, inclusief datum, gebruiker en welke games er na de storting zijn gespeeld.
Controleert of stortingsaanvragers hebben gespeeld	Toon overzicht van spelers waarin te zien is of spelers met een storting ook daadwerkelijk games hebben gespeeld.
Ziet geen onregelmatigheden	Toon samenvatting van backend opnieuw
<b>Alternatief</b>	
Ziet deposits zonder gameplay, wil probleem melden	(zie probleem melden task case)

### 5.1.3 Snel monitoren van meerdere backends, met stortingpogingen

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer backend / startpagina	Toon samenvatting van geselecteerde backend
Gebruiker constateert vele open stortingen en wil deze onderzoeken	(Zie controleren van stortingspogingen)
Maak nieuwe issue aan	
<b>Alternatief</b>	
Ziet stortingen zonder gameplay, wil probleem melden	(Zie rapporteren van nieuwe issue task case)



#### 5.1.4 Nagaan van nieuw gerapporteerde problemen

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer issues overzicht	Toon samenvatting van nieuwe foutmeldingen en openstaande taken voor deze gebruiker
Selecteer een issue, gerelateerd aan een game	Toon overzicht met problemen, inclusief informatie over de speler die de fout gemeld heeft en eventueel eerder gemelde fouten.
Neem de issue op in de persoonlijke taaklijst	Markeer issue dat deze is toegewezen aan de betreffende gebruiker
Zoek uit of dit een gamespecifieke error is, sluit game af	(Zie Game afsluiten)
Ga terug naar issue	

#### 5.1.5 Afsluiten spel

Gebruikersacties	Systeemverantwoordelijkheden
Selecteert pagina met spellen	Toont spellen en staat opzoeken van game toe
Zoekt betreffend spel op	Toont spelinformatie, inclusief actieve spelers
Selecteert spel om af te sluiten	Toon overzicht om spel af te sluiten, vraagt om reden(en) voor afsluiting
Voer reden(en) in en bevestig	Sluit spel af, start nieuwe notificatie / issue thread, stuurt email naar game admins en verwijdert spel uit menu (Eventuele actieve spelers zullen geforceerd worden om het spel te verlaten)

#### 5.1.6 Controleren van stortingpogingen

Gebruikersacties	Systeemverantwoordelijkheden
Vraagt overzicht stortingen aan	Presenteer gedetailleerd overzicht van stortingen
Filtert resultaten door specifieke speler te selecteren of uit te sluiten	Filter (op) specifieke speler



## 5.2 Uitbetaling gerelateerde taken

### 5.2.1 Aanmaken van eerste cashout request (aanvraag uitbetaling)

<b>Gebruikersacties</b>	<b>Systeemverantwoordelijkheden</b>
<b>Kies uitbetalingsaanvraag</b>	Toon huidige balans
<b>Selecteer aanmaken van een cashout request</b>	Toon beschikbare balans en bedrag dat kan worden gevraagd (balance min bonus), (optioneel: geef link naar de regels omtrent cashout)
<b>Vul cashout hoeveelheid in</b>	Toon bevestiging met hoeveelheid cashout
<b>Bevestig</b>	Schrijf aangevraagde bedrag af van de huidige balans, maak nieuw cashout proces aan, maak notificatie thread aan. Toon resultaat van de operatie en een bericht dat het overgebleven bedrag toont.

### 5.2.2 Aanmaken van uitbetalingaanvraag, tijdens openstaande aanvraag

<b>Gebruikersacties</b>	<b>Systeemverantwoordelijkheden</b>
<b>Selecteer aanmaken aanvraag van uitbetaling</b>	Geef melding dat er maar één uitbetaling per keer mogelijk is.

### 5.2.3 Inzien van uitbetalingstatus

<b>Gebruikersacties</b>	<b>Systeemverantwoordelijkheden</b>
<b>Selecteer status van uitbetaling</b>	Toon huidige uitbetalingstatus, inclusief bijbehorende notificatie thread





## 5.2.4 Annuleren van een uitbetaling

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer uitbetalingstatus	Toon openstaande uitbetalingen
Annuleer uitbetaling	Controleer mogelijkheid tot annuleren, toon bevestigingsscherm
Bevestigen	Annuleer cashout, sluit berichten thread

## 5.2.5 Opvragen nieuwe uitbetalingaanvraag

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer cashout aanvraag overzicht	Toon overzicht van huidige uitbetalingen, gegroepeerd op status
Selecteer nieuwe cashout aanvraag	Markeer uitbetaling als “in behandeling” en vraag speler om bevestiging (notificatie via email?) Toon bedrag van cashout, eenvoudige info van de speler, berekening van de gespeelde spellen (gebaseerd op datum laatste uitbetaling)
Onderzoek aanvraag door middel van raw info	Genereer raw info document (Excel sheet or csv) van de cashout. Excel kan worden gebruikt om geavanceerde berekeningen te doen om de integriteit van de data te waarborgen.
Gaat offline voor verder onderzoek	

## 5.2.6 Afwijzen aanvraag tot uitbetaling

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer toegewezen verzoek tot uitbetaling	Toon uitbetaling van de toegewezen speler
Voer afgewezen bedrag in of selecteer uit lijst	Toon informatie van de uitbetaling
Selecteer optie om aanvraag af te wijzen	Toon bevestiging en mogelijkheid tot toevoegen van een bericht.
Kies standaard bericht uit de lijst	Geef standaard afwijzigingsbericht in tekstveld
Afwijzing bevestigen	Annuleer cashout request, herstel balans, stuur notificatie naar speler



## 5.2.7 Opzoeken uitbetaling

Gebruikersacties	Systeemverantwoordelijkheden
Geef ID van uitbetaling op	Toon uitbetaling details, behorende bij ID

## 5.2.8 Accepteren van aanvraag tot uitbetaling

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer toegewezen uitbetalingaanvraag	Toon uitbetalingaanvraag van de toegewezen speler
Accepteer uitbetaling	Toon bevestiging en sta invoer van optioneel bericht toe
Bevestig cashout	Stuur bericht naar gebruiker en rond uitbetalingproces af

## 5.3 Game management

### 5.3.1 Monitoren van uitgaven per game

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer game uitgaven overzicht	Toon overzicht van recentelijk gespeelde spellen, inclusief aantal sessies en uitgaven. Bied mogelijkheid om te filteren op uur, dag, week, 4 weken, alles. Standaard sorteren op meest recente uitgifte (toekennen van geld aan een speler).
Exporteer raw uitkomsten van een game	Genereer raw csv bestand, met per regel de datum van de game sessie, speler ID, spel naam, totale inzet en totaal gewonnen bedrag.

### 5.3.2 Controleren game sessies

Gebruikersacties	Systeemverantwoordelijkheden
Kies lijst met spellen	Toon lijst met beschikbare spellen
Selecteert game	Toon overzicht van meest recente spelsessies. Sta sortering van games toe; totale inzet, totaal gewonnen, totale inkomsten.
Selecteert overzicht van spel van vorige week, sorteert op uitbetaling	Toon overzicht, onthoudt resultaten
Selecteert een game sessie	Toon samenvatting van game sessie, totaal aantal spellen enz. Maak bladeren naar volgende sessie uit de resultatenlijst mogelijk.
Selecteert volgende game sessie	Toon volgende game sessie



### 5.3.3 Controleren uitkomst in slotmachine sessie

Gebruikersacties	Systeemverantwoordelijkheden
Selecteert sessie	Toont overzicht van slotmachinesessies
Selecteert een event in een sessie	Weergeven van de symbolen die op de rollen getoond zijn. In geval van een kaartspel worden details gegeven over de kaartverdeling.

## 5.4 Speler management

### 5.4.1 Controleren uitkomst in slotmachine sessie

Gebruikersacties	Systeemverantwoordelijkheden
Selecteert speler overzicht	Toon samenvatting van spelers, met mogelijkheid om spelers te bladeren en te doorzoeken.
Selecteert bladeren van spelers	Toon spelers, mogelijkheid om lijst te sorteren en te groeperen
Stelt filter samen en zoekt	Toon spelers die voldoen aan de zoek en filtercriteria
Slaat filter op	Vraag om naam
Geeft naam op en bevestigt	Sla filter op voor hergebruik

### 5.4.2 Speleroverzicht, gebruikmakend van opgeslagen filter

Gebruikersacties	Systeemverantwoordelijkheden
Selecteert filter uit filter lijst	Toon spelers in gefilterde lijst

### 5.4.3 Controleren van nieuw aangemelde spelers, die nog niet ingelogd zijn

Gebruikersacties	Systeemverantwoordelijkheden
Selecteert gebruikers overzicht	Toon nieuw geregistreerde spelers en tijdsberekening hoe lang geleden ze zijn aangemeld Groepeer op status
Filter spelers die geregistreerd zijn, maar nog nooit hebben ingelogd.	Toon enkel spelers die voldoen aan de filter

### 5.4.4 Controleer status van registratie-email

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer status van de email registratie van een speler	Toon status en resultaten uit een eventuele emailconversatie



### 5.4.5 Toestaan van bankaccount van speler

Gebruikersacties	Systeemverantwoordelijkheden
Zoek speler	Toon resultaten als meerdere spelers voldoen aan de zoekcriteria, of ga direct naar de speler details
Selecteer bankgegevens van speler	Toon bankgegevens en of deze ooit zijn goedgekeurd

### 5.4.6 Toevoegen van een bonus

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer speler	Toon resultaten als meerdere spelers voldoen aan de zoekcriteria, of ga direct naar de speler details
Selecteer bankgegevens van speler	Toon bankgegevens en of deze voorheen zijn goedgekeurd

## 5.5 Probleem (issue) management

### 5.5.1 Overzicht met problemen (issues)

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer issue overzicht	Toon overzicht met nieuw toegevoegde problemen, toon wanneer dit probleem als eerste voorkwam, problemen die al zijn toegewezen aan een gebruiker en recent veranderde problemen

### 5.5.2 Kiezen van een probleem

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer probleem uit overzicht	Toon probleemdetails en eventueel het bericht van de meldende persoon. Vraag om invoer van titel, omschrijving en probleemcategorie. Biedt mogelijkheid om probleem met een template bericht te beantwoorden.
Titel, omschrijving enz. opgeven	Sla probleemdetails op Toon probleem informatie en context informatie die relevant is aan de probleemcategorie.



### 5.5.3 Oplossen van een betalingsprobleem

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer probleem uit de betalingscategorie	Toon probleem informatie, speler data, recente stortingen en stortingspogingen.
Selecteer recente onvoltooide betaling, met door de gebruiker aangegeven bedrag	Onthoud probleem, toon stortingsinfo inclusief stappen om PSP te benaderen
Onderzoek samen met de externe betalingprovider (payment service provider, PSP)	
Pas stortingsbedrag aan en meld de fout	Sla melding op met stortingsticket en probleemticket, pas stortingsbedrag aan
Laat probleem staan, tot nader bericht van de PSP	

### 5.5.4 Afronden van een betalingsprobleem

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer probleem uit lijst	Toon probleeminfo
Controleer of de storting succesvol is verlopen en of deze toebehoort aan de storting waarvan het bedrag is aangepast.	Toon stortingsinfo
Afronden van probleem	Verander status van probleem in "opgelost"



## 5.6 Algemene taken

### 5.6.1 Probleem melden

Gebruikersacties	Systeemverantwoordelijkheden
Klikt op melden van een probleem	Vraag om titel, beschrijving en eventueel wie de taak krijgt dit probleem op te lossen
Vult waarden in en meldt probleem	Toont bevestiging van het te melden probleem
Bevestigen	Sla notificatie op en stuur email naar desbetreffende medewerker / rol.

### 5.6.2 Antwoorden op een probleemmelding

Gebruikersacties	Systeemverantwoordelijkheden
Klikt op beantwoorden	Systeem toont mogelijkheden tot beantwoorden (voorgaande conversatie, basis info van ontvanger, Aan-veld)
Voert nieuw bericht in en de bijbehorende status	Vraag om bevestiging
Bevestigen	Voegt bericht toe aan de thread, update thread status en stuurt bericht naar ontvanger(s)



## 5.7 Marketinggerelateerde taken

### 5.7.1 Nieuw bonusprogramma aanmaken

Gebruikersacties	Systeemverantwoordelijkheden
Selecteer aanmaken van nieuw bonusprogramma	Toont basisinformatie, zoals name en beschrijving
Vult info in en bevestig aanmaken	Maakt een nieuw bonusprogramma aan met de opgegeven naam. Bonus is niet direct van toepassing en is uitgeschakeld. Toon venster met details van de bonus (zie <i>Nieuwe bonusregel aanmaken</i> )

### 5.7.2 Nieuwe bonusregel aanmaken

Gebruikersacties	Systeemverantwoordelijkheden
Kies aanpassen van bonusprogramma	Toont venster met details van het bonusprogramma, waar de bonusregels aan kunnen worden toegevoegd. Bied mogelijkheid aan gebruiker om een van de drie types bonus regels toe te voegen.
Selecteert een type bonusregel om toe te voegen	Toon regel specifieke parametervelden, bied mogelijkheid om AND en OR bij bonusregels te gebruiken.
Vult parameters in en bevestig	Verifieer ingevoerde data en voeg nieuwe regel toe aan het bonusprogramma

### 5.7.3 Overzicht van huidige bezoekers

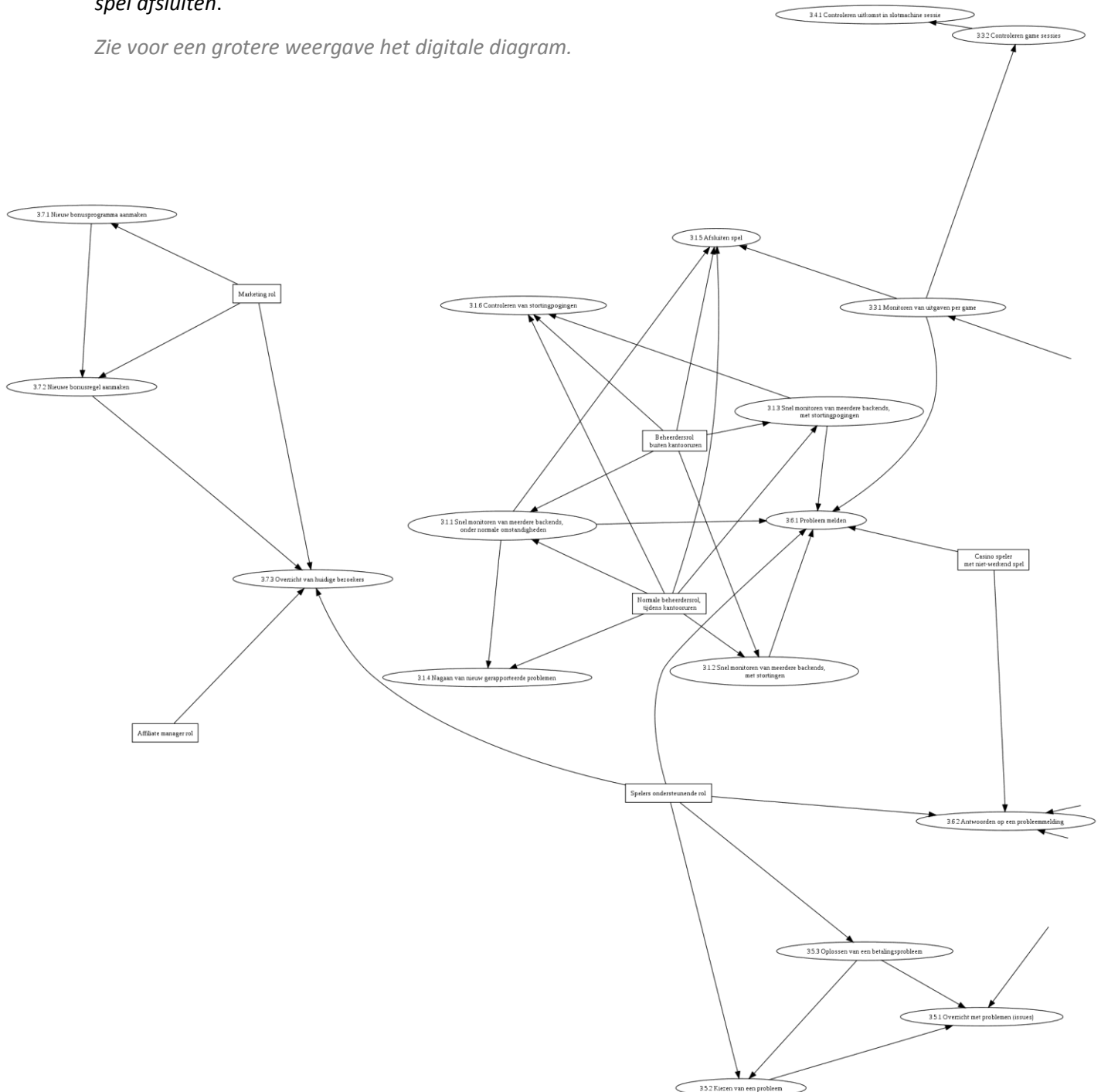
Gebruikersacties	Systeemverantwoordelijkheden
Selecteer bezoekersstatistieken	Systeem toont overzicht van bezoekers van vandaag, ingelogde bezoekers en spelende bezoekers in eenvoudige grafieken of lijsten



## 6 Task map

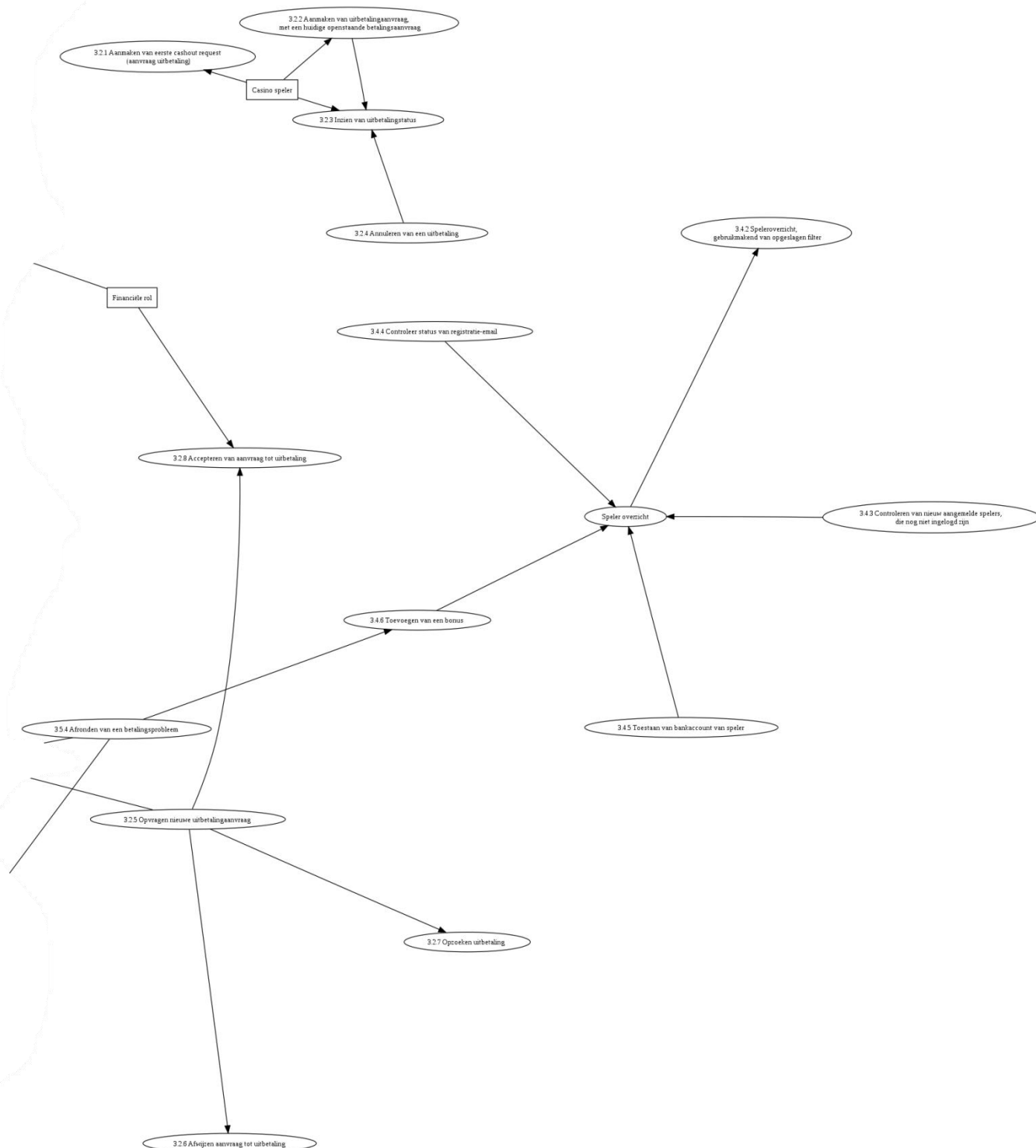
In Figuur 6-1 en Figuur 6-2 is een task map te zien, waarin de task cases zijn gecombineerd met de rollen. Met behulp van deze task map kan een venstervolgorde geoptimaliseerd worden om het aantal handelingen en mouseclicks op een pagina te verminderen. Zo zouden er bijvoorbeeld op de pagina voor speluitgaven (game returns monitoring) links kunnen komen naar *problemen melden* en *spel afsluiten*.

*Zie voor een grotere weergave het digitale diagram.*



Figuur 6-1: Task map (linker deel)





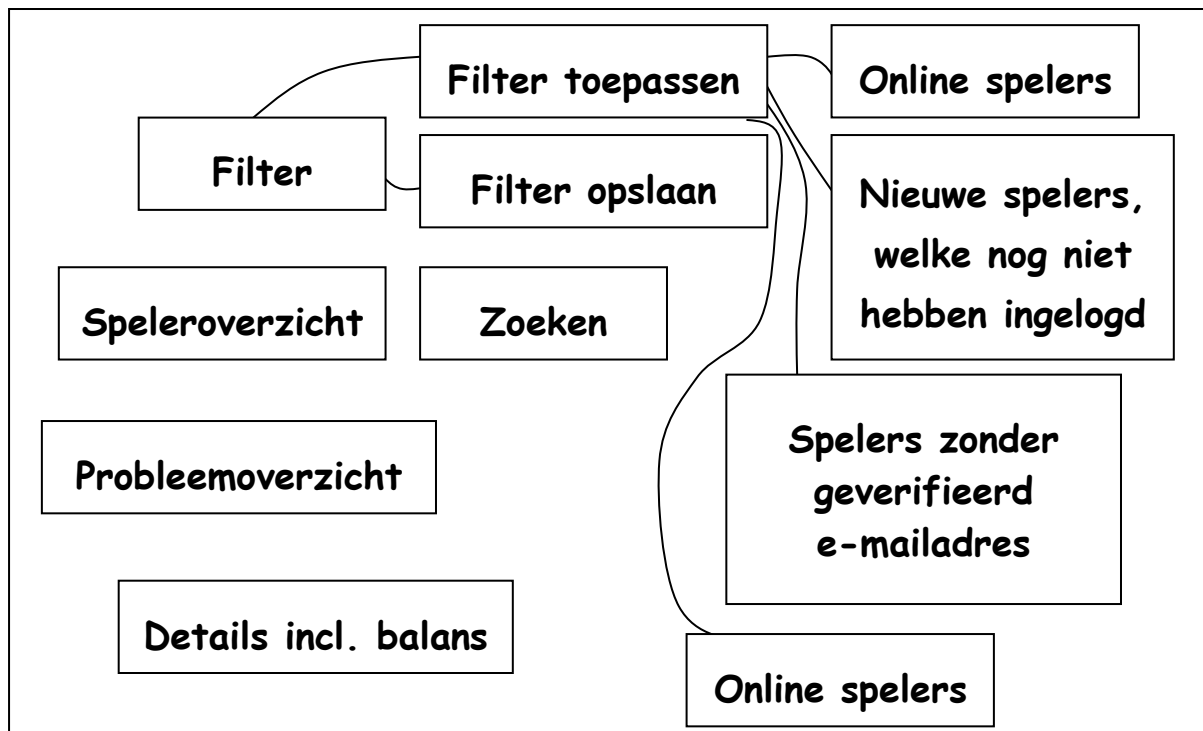
**Figuur 6-2: Task map (rechter deel)**



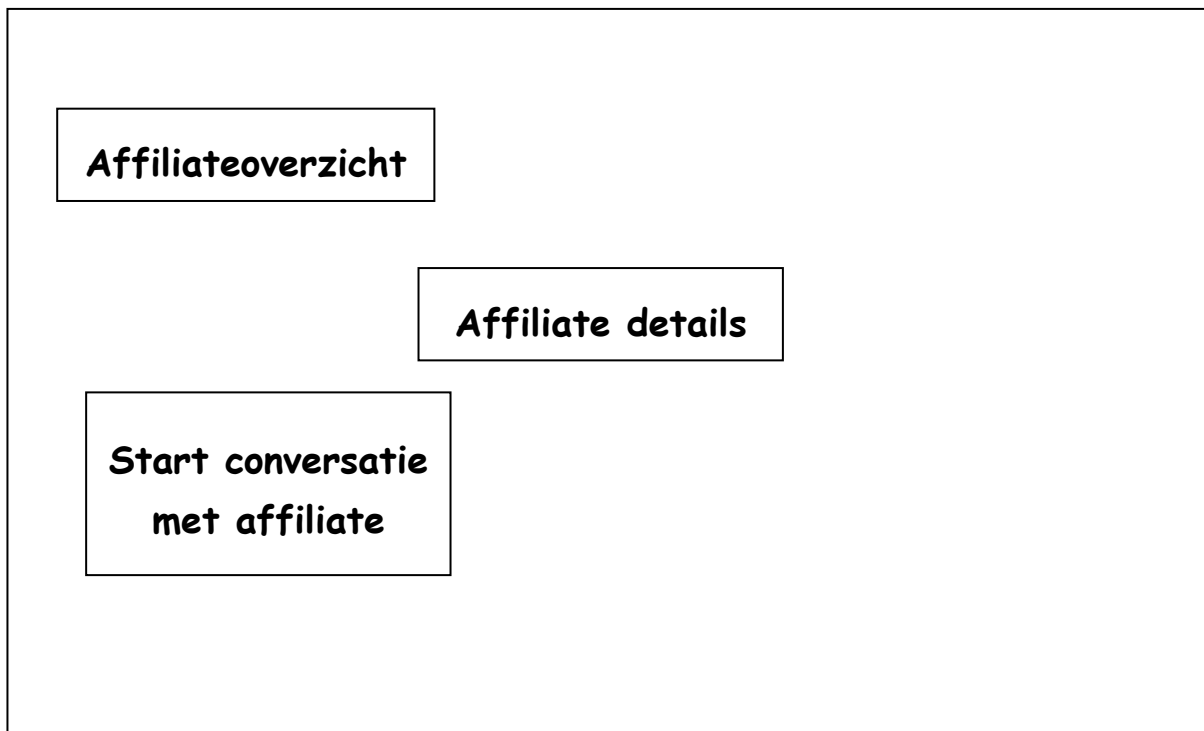
## 7 Abstracte prototypes

In de task cases is de gewenste functionaliteit gedefinieerd. Door aan elkaar verwante cases te groeperen kunnen abstracte prototypes gemaakt worden. Dit zijn de functionaliteiten die op een bepaalde webpagina aanwezig moeten zijn. Met behulp van de abstracte prototypes kan dan weer de gebruikersomgeving worden ontworpen.

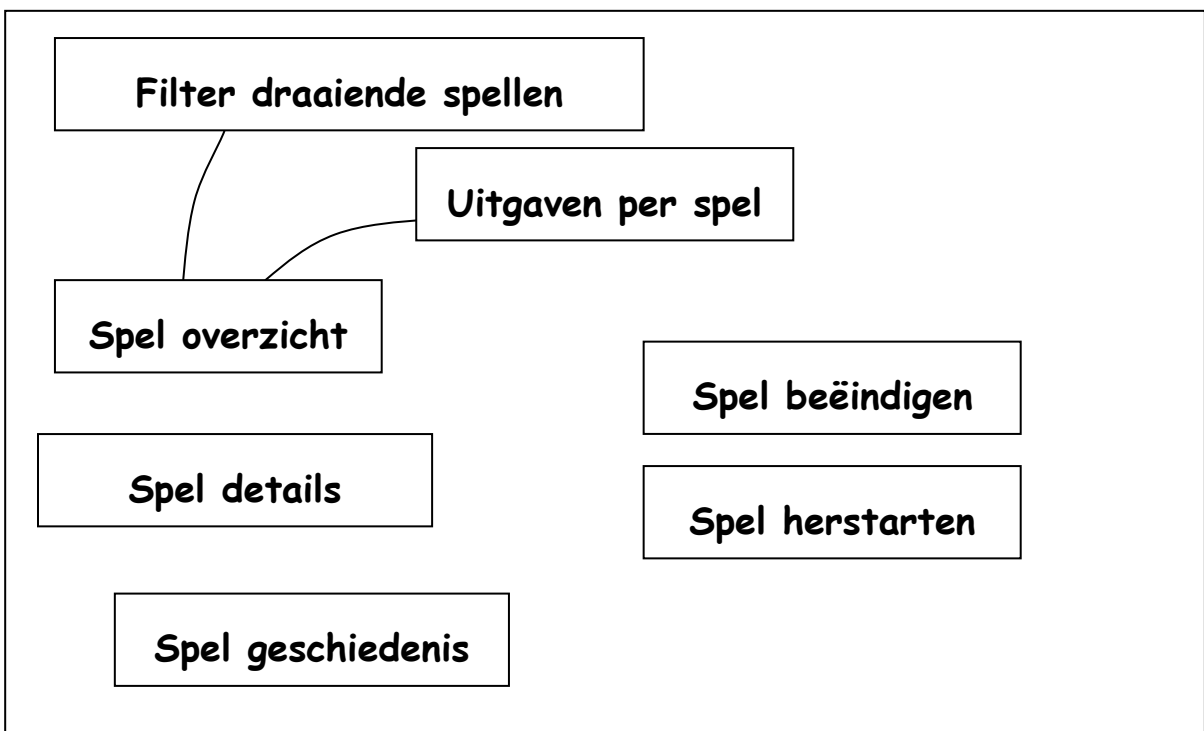
### 7.1 Speleroverzicht



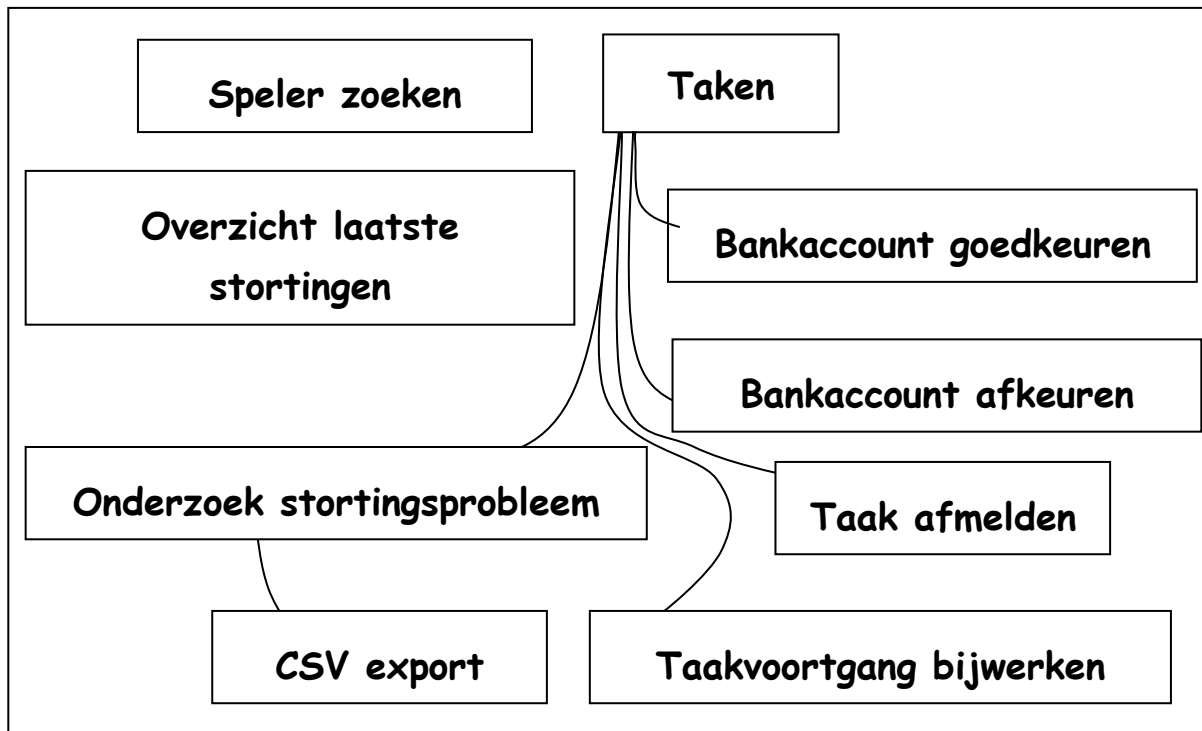
## 7.2 Affiliate overzicht



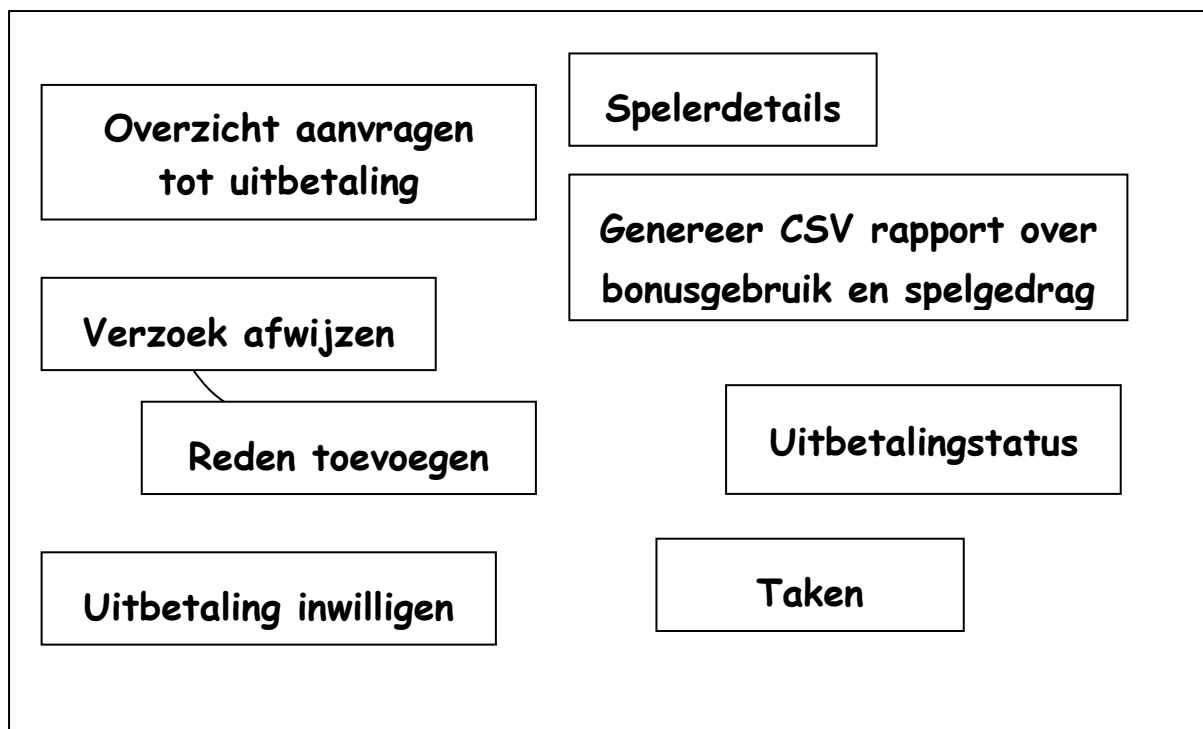
## 7.3 Speloverzicht



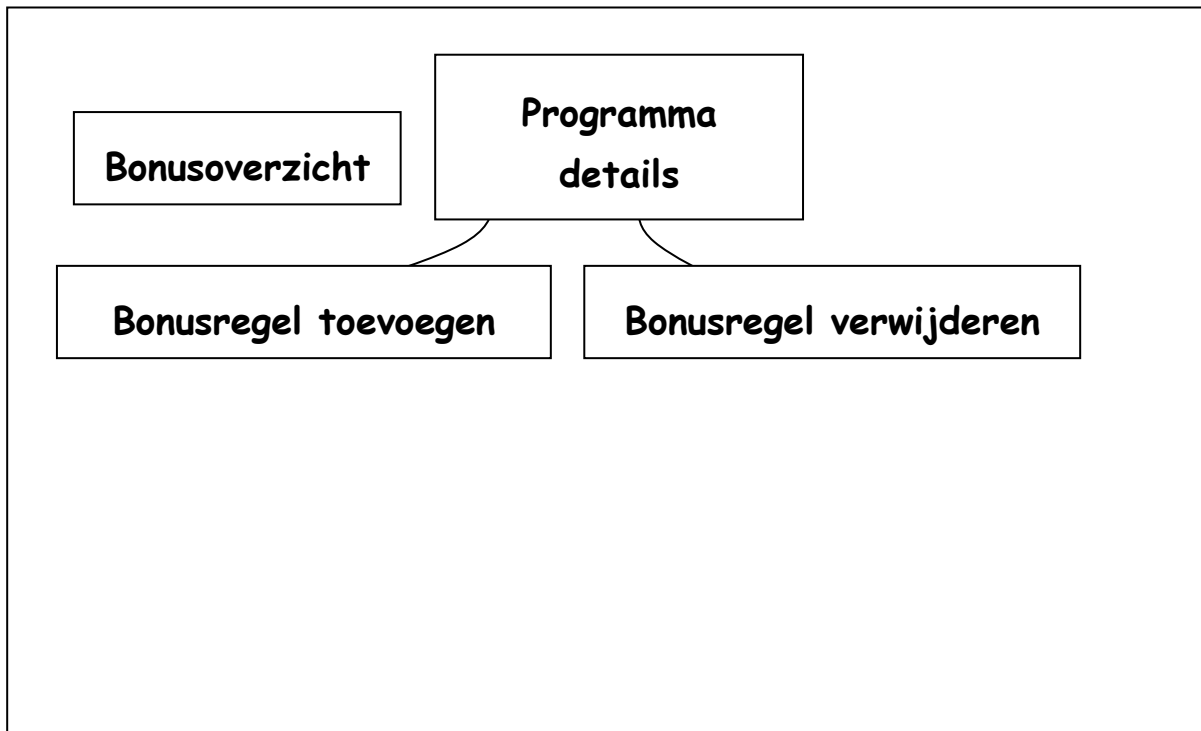
## 7.4 Stortingen



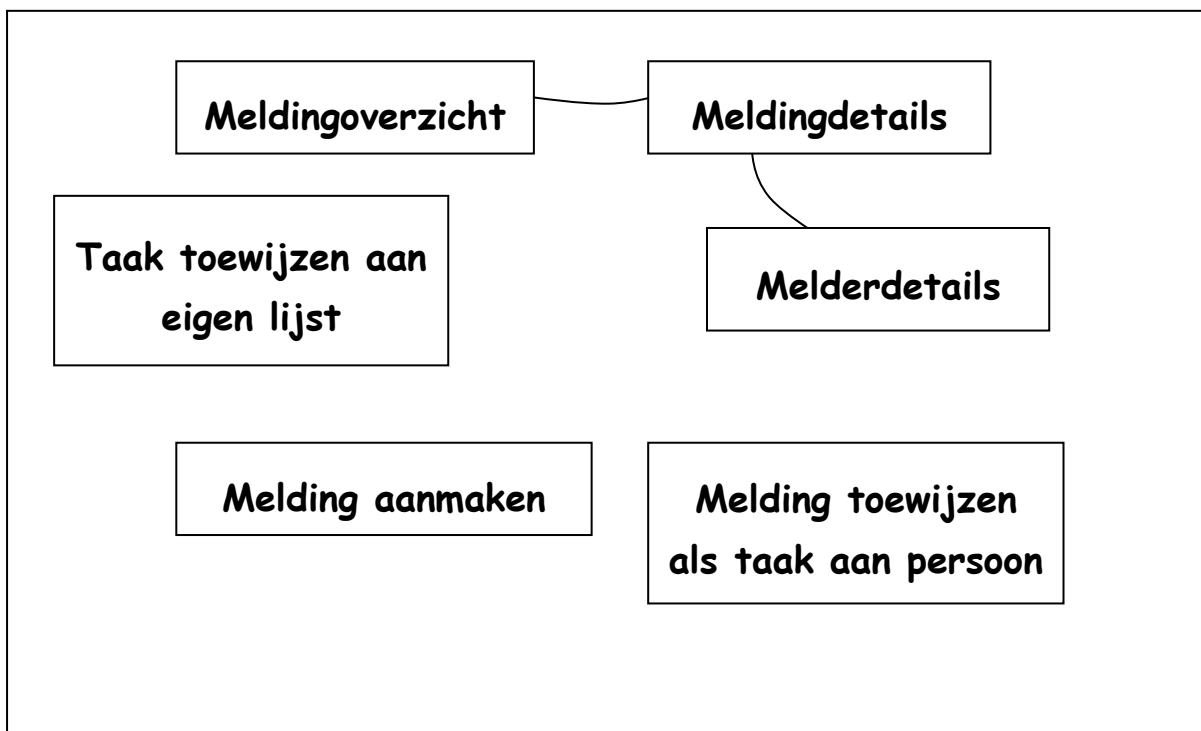
## 7.5 Verzoeken tot uitbetaling



## 7.6 Bonusprogramma's



## 7.7 Probleem / taakoverzicht



## 8 Gebruikersinterface

Met behulp van de abstracte prototypes kan nu worden nagedacht over de schermindeling van de gebruikersinterface. Het doel hierbij is om veel voorkomende taken (behorende bij een rol) eenvoudig te selecteren zijn. Als oplossing hiervoor is een sidebar bedacht, die snelkoppelingen en widgets bevat voor verschillende gerelateerde acties.

### 8.1 Te implementeren componenten

In dit hoofdstuk staan de componenten die er in de Game Management Service gebruikersomgeving moeten komen. Deze lijst is gebaseerd op de onderdelen uit de task map.

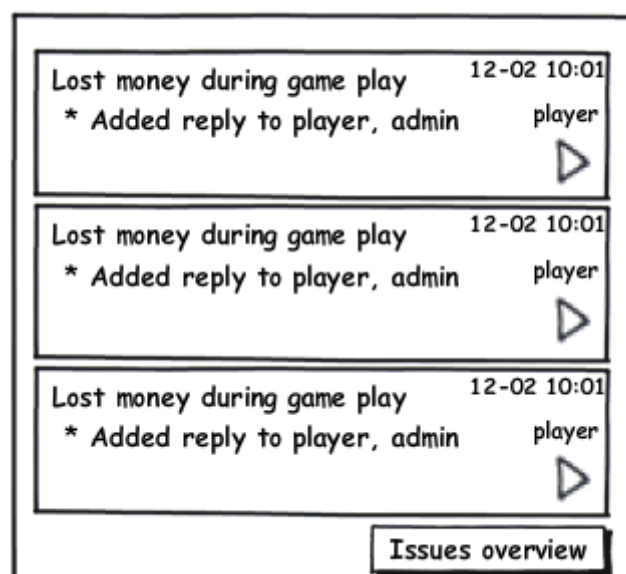
#### 8.1.1 Probleemrapportage

Een widget die de geschiedenis van een probleem weergeeft, hierin zijn de genomen stappen te herkennen. In dit (tekst)veld staan de status, eventuele notities, aanpassingen en de conversaties over het probleem met andere medewerkers of spelers.

```
22-02-2010 player@player.com: hi i've made a deposit but the money is not added please help!
[ - admin: added note (looked up players deposit history and a deposit was made at 22-01-2001:23:33:33 ]
  \< - admin: status updated to investigating \>
[ - admin: it seems the deposit is stuck in partial payment going to manually override this ]
admin: hi player we are investigating the issue
```

#### 8.1.2 Probleemmeldingen overzicht

Een overzicht van de openstaande of in behandeling zijnde problemen die zijn toegewezen aan de ingelogde medewerker. Mogelijkheid tot sluiten, updaten en beantwoorden van een probleem.



### 8.1.3 Probleemmelding beantwoorden

Widget voor het beantwoorden of toevoegen van een notitie aan een probleemmelding. In de widget kan geselecteerd worden welke personen melding krijgen van de mutatie.

The widget is a rectangular box containing a text editor and a recipient selection area. At the top, there is a toolbar with icons for bold (B), italic (I), underline (U), text color (abc), style (dropdown), list (bulleted and numbered), link, unlink, and smiley. To the right of the toolbar is an 'Auto-reply' dropdown menu and two radio buttons labeled 'Message' and 'Note'. Below the toolbar is a large text area containing the text 'Some text' and 'A second line of text'. At the bottom of the widget, there is a 'participants' field showing 'admin, [player](#)' and a 'Recipient' field with a dropdown menu showing '[ ] public [x] private'. A 'Send' button is located to the right of the recipient field.

### 8.1.4 Probleem in behandeling

Dit component toont de informatie van het probleem waar op dat moment aan gewerkt wordt. De medewerker kiest een probleem uit de lijst, waarna de status hiervan verandert in "in behandeling". Vanaf dat moment is het probleem te zien in de gebruikersomgeving, zodat tijdens het oplossen gebruik kan worden gemaakt van de gegevens die er van het probleem beschikbaar zijn.

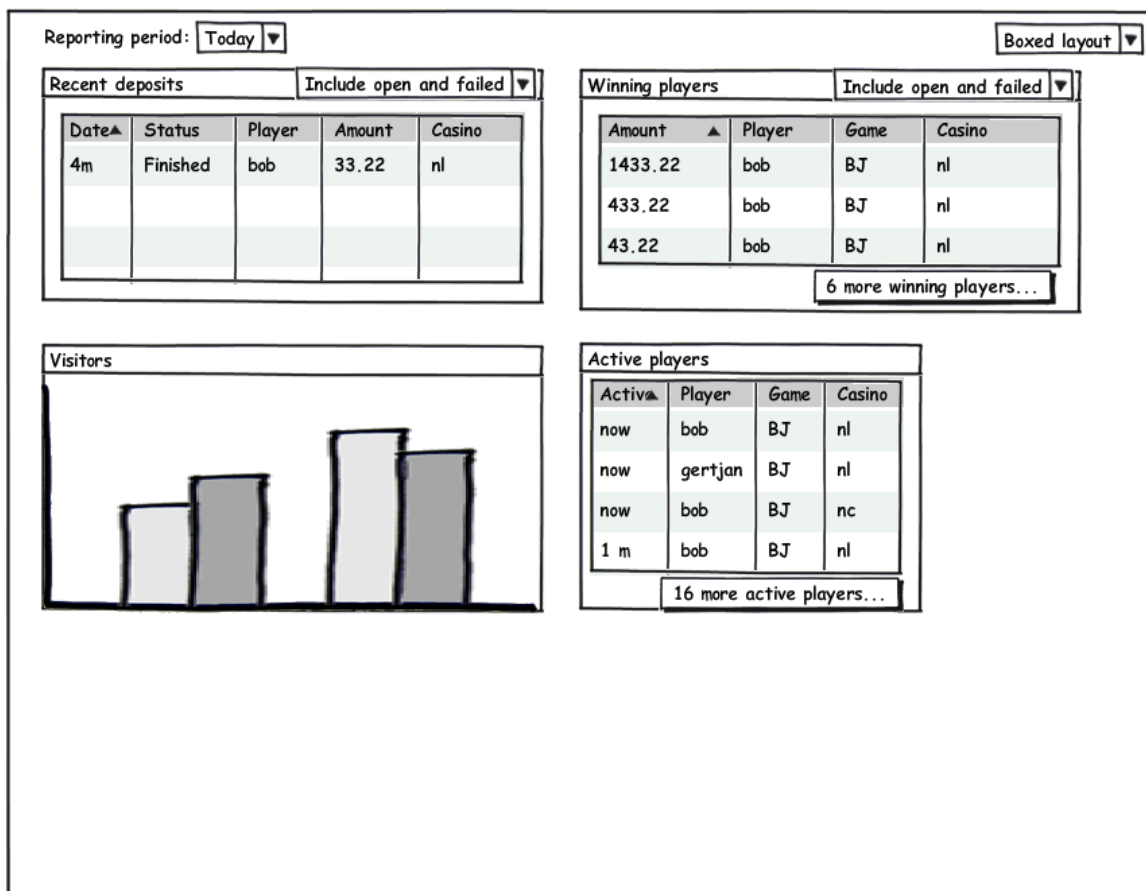
The widget is a rectangular box with a title bar that reads 'Currently working on 'lost money'' and an icon of three people. Below the title bar is a large rectangular area labeled 'issueReport'.

### 8.1.5 Automatische antwoorden

Het zal vaak voorkomen dat spelers dezelfde vragen stellen of overeenkomstige problemen hebben. Deze widget biedt de mogelijkheid om deze meldingen te beantwoorden met een automatisch of template bericht.

### 8.1.6 Status overzicht

Overzicht van de huidige backendstatus, met verschillende informatiewidgets over de winnende spelers, draaiende spellen, bezoekers, actieve spelers enz. Het is mogelijk om details van onderdelen op te vragen, bijvoorbeeld personalia.



### 8.1.7 Status detail

Geeft meer informatie over een bepaald onderdeel uit het status overzicht



### 8.1.8 Zoeken van speler(s)

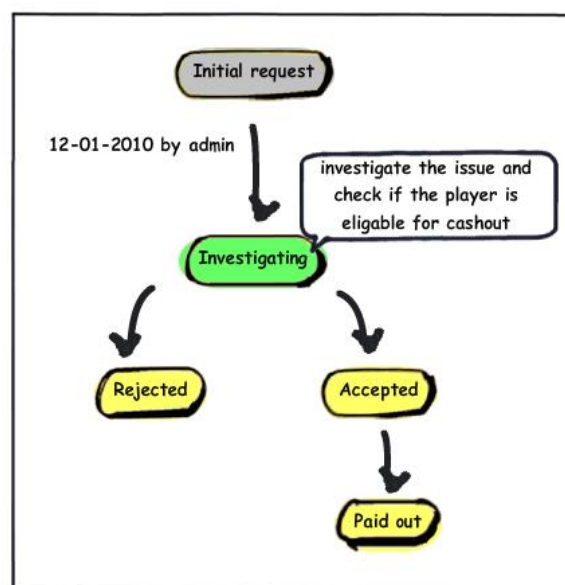
Zoekveld voor snelle selectie van spelers. Mogelijkheid om snel te zoeken (op email of id) en zoeken met geavanceerde opties. Na de zoekopdracht zal er een lijst(je) worden getoond met de resultaten, afhankelijk van de pagina vanwaar de zoekactie wordt aangeroepen. Deze kan namelijk als onderdeel van de spelerpagina dienen, maar ook als widget op verscheidene andere pagina's.

Search player	
John Smith \$ 11,22	11-02-2008
John Smith \$ 11,22	11-02-2008
John Smith \$ 11,22	11-02-2008

Pressing  
the arrow  
shows the  
advanced  
search field

### 8.1.9 Proces en voortgangsoverzicht

Veel taken die een medewerker uitvoert zijn onderdeel van een workflow. Met een procesoverzicht is een soort van broodkruimelpad te zien welke stappen er zijn genomen, en welke er nog genomen moeten worden.



### 8.1.10 Proces stappen afhandeling

Met dit component kiest de medewerker de volgende te nemen stap. Bij iedere keuze staat een beschrijving wat er verwacht kan worden van die actie.

issueReply

☐ Reject

- Reject the cashout and notify the player with the reason for rejection

☐ Accept

- Accept optionally provide a message

☐ Leave as is

- Return to the previous screen

Confirm

### 8.1.11 Wisselen van backend

Met het GMS kunnen meerdere backends tegelijk worden beheerd. Met deze optie kan de gebruiker een backend selecteren.

Select source

☐ Casino Belgie

☐ Casino Nederland

☐ Netcash Casino

☐ Poker Backend

---

☐ All sources



### 8.1.12 Huidige uitbetaling

In deze widget is de informatie van een uitbetaling te vinden die de medewerker op dat moment in behandeling heeft. Tijdens het controleren en verifiëren van de uitbetaling zal de medewerker verschillende pagina's gebruiken. De bedoeling van deze widget is dat de informatie over de aanvraag en de bijbehorende speler altijd opvraagbaar zijn.

Cashout	22-02@11:22
Playername: \$1000.00	
\$1044.22 (334.33)	
	<a href="#">Reject</a> <a href="#">Approve</a>

### 8.1.13 Wijzigen van werkomgeving

*Optioneel* – Een werkomgeving is een indeling van widgets en pagina's die is geoptimaliseerd voor een bepaalde rol die een medewerker heeft. Met deze optie kan gewisseld worden van werkomgeving, bijvoorbeeld van Marketing naar Spelerondersteuning.

### 8.1.14 Dashboard

Een dashboard is een overzichtspagina waarvanuit de belangrijkste componenten beschikbaar zijn. Dit dashboard bevat bijvoorbeeld het statusoverzicht zoals beschreven in 8.1.5. Het dashboard wordt over het algemeen als startpagina van de webUI gebruikt.

### 8.1.15 Menu

Het menu ondersteunt de navigatie tussen de hoofdcomponenten of componentverzamelingen. Aangezien de componenten hiërarchisch kunnen worden gerangschikt, zal het menu deze structuur ondersteunen.

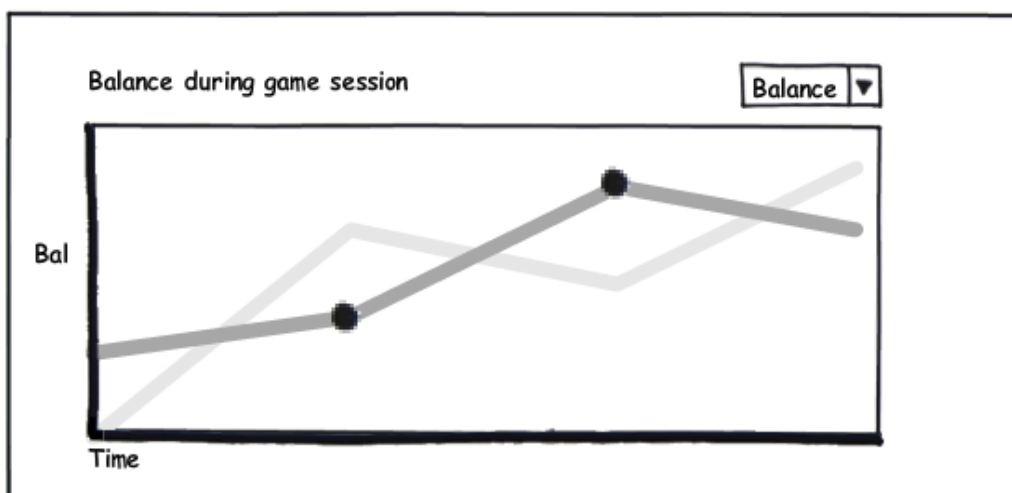


### 8.1.16 Spelerslijst filter

Widget waarmee de spelerslijst kan worden gefilterd. Deze wordt gebruikt bij het zoeken op personen, of het versturen van berichten aan spelercollecties. De zoekopties zijn: zoeken op voornaam, achternaam, email, nickname, laatst ingelogd (aantal dagen geleden).

### 8.1.17 Visualisatie van spelsessie

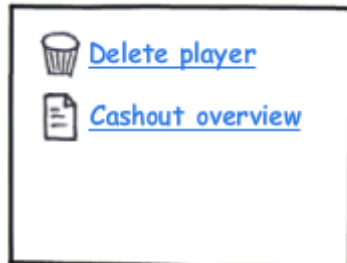
Toont grafiek van hoeveelheid verloren en gewonnen bedragen van één spelsessie. X-as is de tijd, y-as is de balans van de speler. De grafiek ondersteunt mouse-over informatie.



### 8.1.18 Relevante links

Klein venster met links naar andere pagina's, relevant aan de huidige pagina.

[Home](#) | [Products](#) | [Company](#) | [Blog](#)



### 8.1.19 Opzoeklijst


Inputveld boven kolommen in datatabellen. Hiermee kan de gebruiker een rij selecteren door een zoekwaarde in te typen (eenvoudige manier van de zoekfunctie). Het betreft hier zoeken op beginletter(s).

## 8.1.20 Spelerdetail

Details over een geselecteerde speler, inclusief de balans, laatste aanmelding, vip level enz.

Overview	Issues	Cashouts	Players	Games
Summary	Overview	Deposits	Active players	Winning players



[Players](#) > [Search](#) > Player detail (bob@bob.com - Schlarmann, Bob)



Username: bob@bob.com  
Email: bob@bob.com  
Balance: 3432.33  
Real/Bonus: 3299.00/33.33

[Block player](#)  
[Add bonus](#)  
[Send message](#)

Outstanding actions

-  [2 issues](#)
-  [1 cashout request](#)

Summary | Issues | Financial accounting | Game sessions | Player details

Registered: 22-12-2008 (2y 6m ago)  
VIP level: 3/5 (medium vip) [change vip level](#)  
Games played: 344  
Last activity: 29-03-2010 (1d ago)

Recent deposits


Date	Amount	Method	Status
<a href="#">yesterday</a>	33.22	iDEAL	success
<a href="#">monday</a>	22.33	iDEAL	

Recent activity

03-29 22:22:22: logged in  
22:25:22: played game BJ  
22:30:22: ended game

Overview	Issues	Cashouts	Players	Games
Summary	Overview	Deposits	Active players	Winning players

[Players](#) > [Search](#) > Player detail (bob@bob.com - Schlarmann, Bob)



Username: bob@bob.com  
Email: bob@bob.com  
Balance: 3432.33  
Real/Bonus: 3299.00/33.33

[Block player](#)  
[Add bonus](#)  
[Send message](#)

Summary | Issues | Financial accounting | Game sessions | Player details

Current issues

I has no balance (1d ago - new)

Player> i have no balance received on monday my last deposit  
Admin> {checked this issue it seems to be related to [deposit:3433](#)}

[Reply](#) [Pick issue](#)

I has no grammar (1w ago - working on it)

Past issues

Description^	Date	Status
No balance again & 2010-02-02	Solved	

[View all](#)

created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)



Overview	Issues	Cashouts	Players	Games
Summary	Overview	Deposits	Active players	Winning players

Players > Search > Player detail (bob@bob.com - Schlarmann, Bob)



Username: bob@bob.com  
Email: bob@bob.com  
Balance: 3432.33  
Real/Bonus: 3299.00/33.33

[Block player](#)  
[Add bonus](#)  
[Send message](#)

Outstanding actions  
2 issues  
1 cashout request

Summary Issues Financial accounting Game sessions Player details

[View balance statement](#)  
[View payment activity](#)

Unfinished deposits

Id	Type	Date	Amount	Bonus	Status	External Id
123	IDEAL	29-03-10	\$33.22	\$22.22		<a href="#">Adjust amount</a>

Pending cashouts

Id	Date	Amount	Status	
123	29-03-10	\$33.22	New	<a href="#">Approve</a> , <a href="#">Reject</a>

Other search results



Balance statement					Reporting period
Type	Id	Date	Add	Subtract	Today
Initial amount	-	-	\$333.00		
Game	155	Today		\$33.00	
Deposit	1244	Today	\$500.00		
End balance	-	-	-	\$800.00	

Overview	Issues	Cashouts	Players	Games
Summary	Overview	Deposits	Active players	Winning players

Players > Search > Player detail (bob@bob.com - Schlarmann, Bob)



Username: bob@bob.com  
Email: bob@bob.com  
Balance: 3432.33  
Real/Bonus: 3299.00/33.33

[Block player](#)  
[Add bonus](#)  
[Send message](#)

Outstanding actions  
2 issues  
1 cashout request

Summary Issues Financial accounting Game sessions Player details

From: / / To: / /

Details

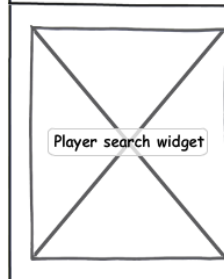
Loss/winn: \$333.33 (ratio: 0.44)

Start^	End	Game name	Win	Loss	Ratio	Duration
22:22	22:23	BJ		\$33.00	1.1	1m
20:00	21:00	Baccarat	\$333.00		0.3	1h
19:00	19:30	BJ		\$55.00	1.1	30m

Summary


Summary shows total per game type

Other search results



Overview	Issues	Cashouts	Players	Games
Summary	Overview	Deposits	Active players	Winning players

Players > Search > Player detail (bob@bob.com - Schlarmann. Bob)




Username: bob@bob.com  
 Email: bob@bob.com  
 Balance: 3432.33  
 Real/Bonus: 3299.00/33.33

[Block player](#)  
[Add bonus](#)  
[Send message](#)  
[Change password](#)

Outstanding actions

- 2 issues
- 1 cashout request

Other search results



Player search widget

Summary Issues Financial accounting Game sessions Player details Banking

Bank accounts

3646293xxx

Account number:

343244xxx

created with Balsamiq Mockups - www.balsamiq.com

## 8.1.21 Spelsessies

Overzicht van de sessies die een spel heeft gehad.

Sessions					
Game: Blackjack		sessionFilter ▼	playerFilter ▼		
Date ▲	Player	Duration	Rounds	Return	Total win
22-02-10	bob	active	333	88.4	-14
22-02-10	bob	3h	333	88.4	-12
total	overall				





### 8.1.22 Sessie details

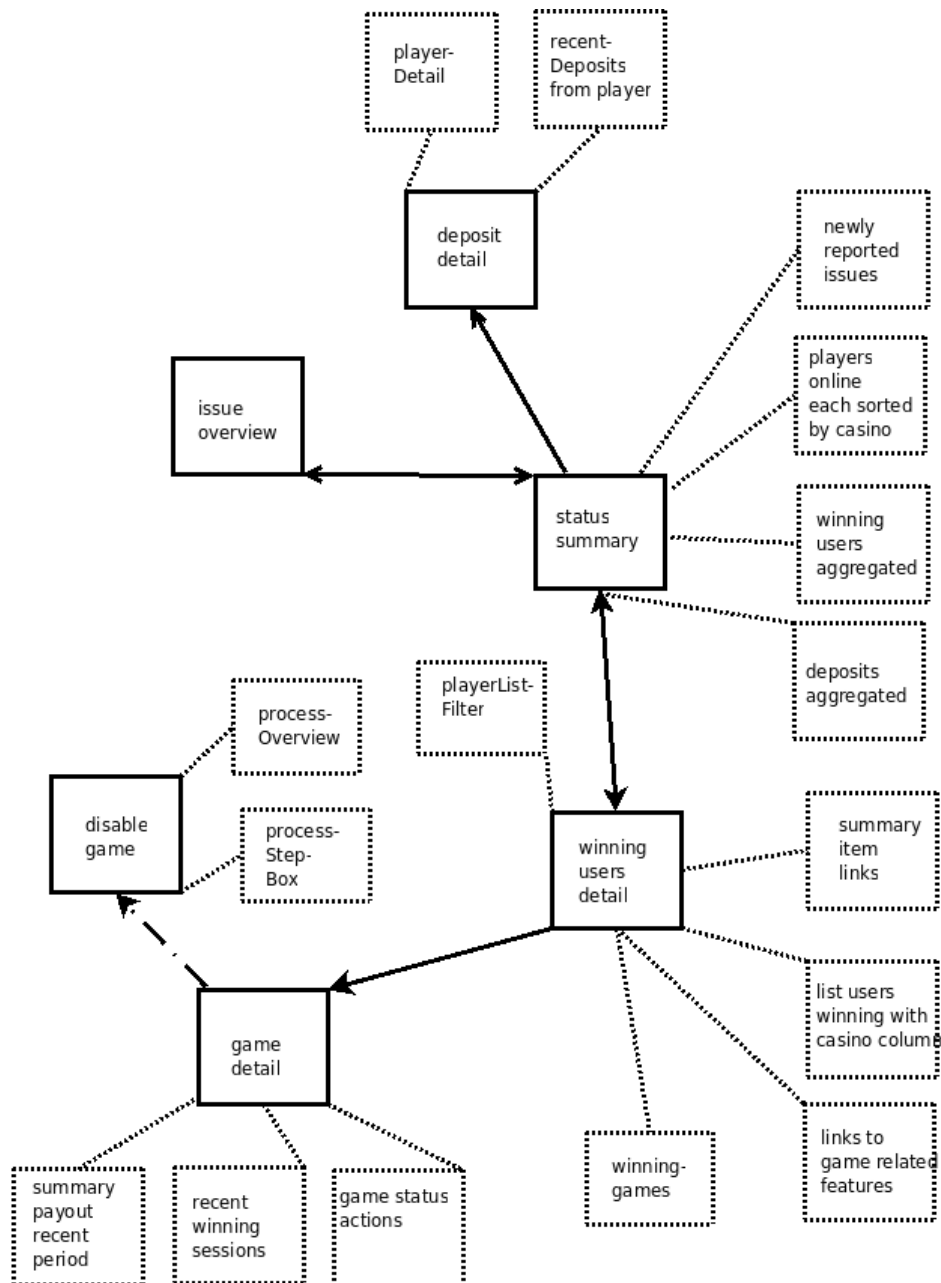
Geschiedenis hoe een sessie verlopen is

Session details				
Date ▲	Action	Bet	Win	
22:22	deal	33.22	22.00	details...
1,2,3, ...				
...				
gameSessionVisualizer				

## 8.2 Prototypes

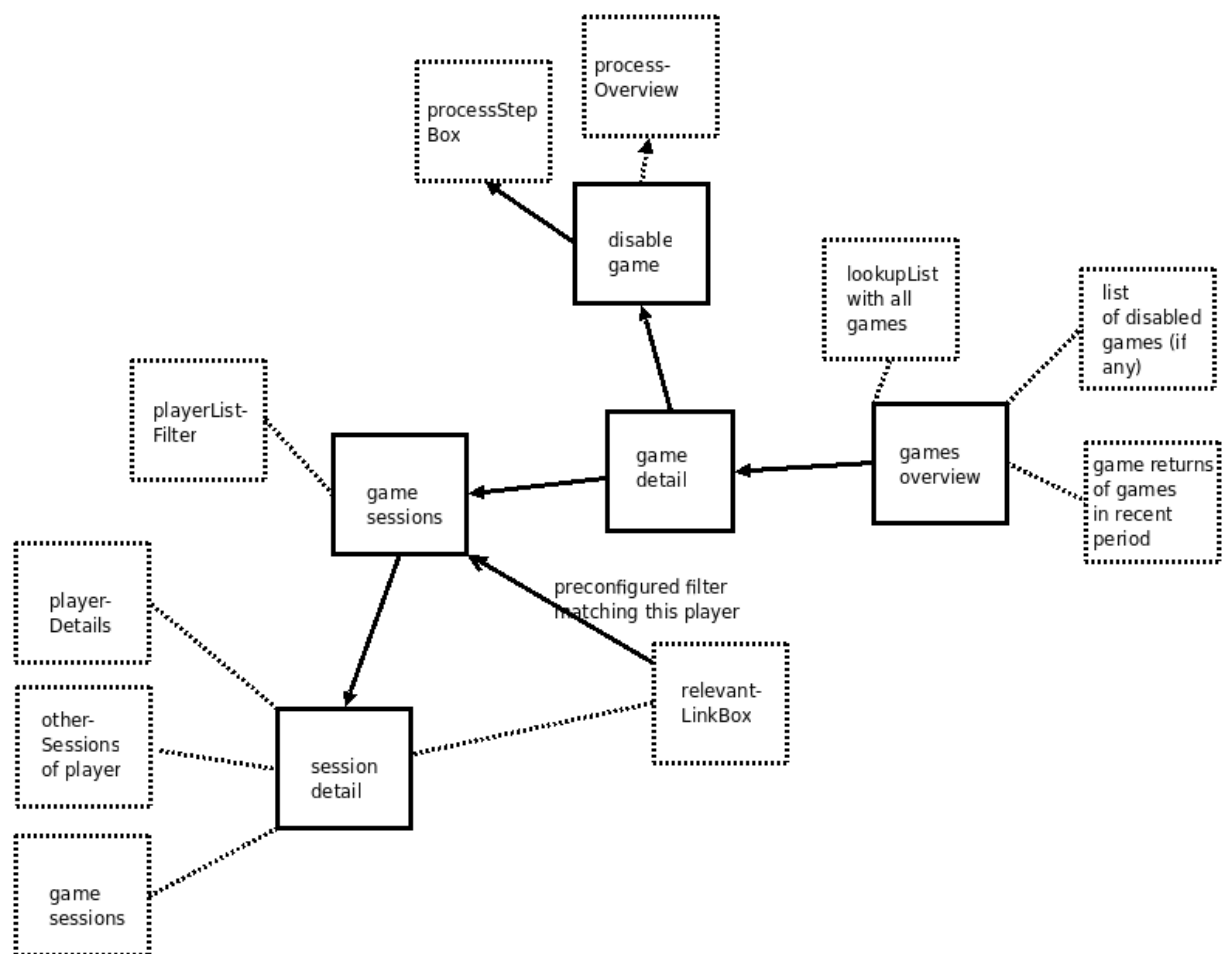
Met de componenten en de task map kan nu worden nagedacht over de layout van de pagina's.

### 8.2.1 Navigatiemap



Figuur 8-1: Navigatiemap voor spelerondersteuning

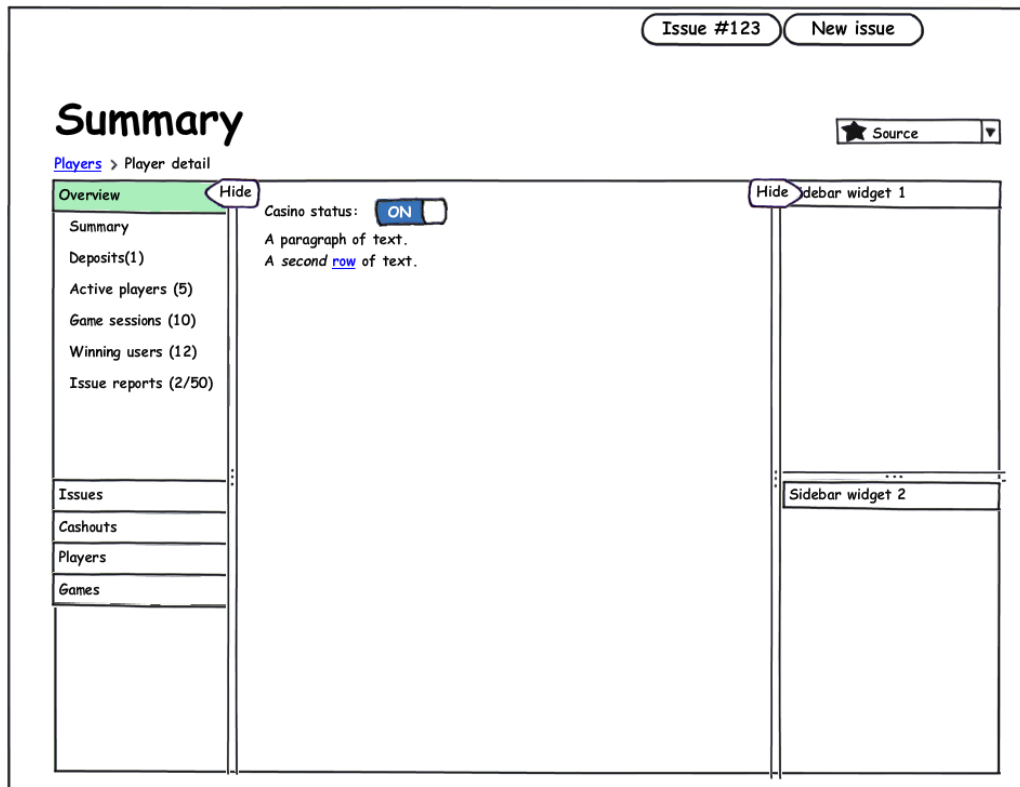




**Figuur 8-2: Navigatiemap voor het beheren van de spellen**

## 8.2.2 Paginaindeling

Voor de indeling van de pagina's is bedacht om een hoofdvenster, een menubalk en een zijbalk te maken. Deze indeling wordt vaak gebruikt bij websites, wat de gebruiksvriendelijkheid verhoogt; de gebruikers weten wat de bedoeling is van de indeling.



Issue #123

New issue

## Summary

★ Source

Overview	Issues	Cashouts	Players	Games
Summary	Deposits (1)	Active players (5)	Game sessions (10)	Winning users (12)
				Issue reports (2/50)

Players > Player detail

Casino status: ☒ ON

A paragraph of text.  
A second row of text.

Hide

debar widget 1

...

Sidebar widget 2

...

Issue title: no money on account after deposit

Issue added: 12-08-1984  
Assigned to: you

Hide

Add reply

Add note

Issue #123

New issue

## Summary

★ Source

Overview	Issues	Cashouts	Players	Games
Summary	Deposits (1)	Active players (5)	Game sessions (10)	Winning users (12)
				Issue reports (2/50)

Players > Player detail

Casino status: ☒ ON

A paragraph of text.  
A second row of text.

Hide

debar widget 1

...

Sidebar widget 2

...

Currently there are 5 players online:

- Gertjan
- Bob

