How can a mobile application contribute towards the commemoration of the Fireworks disaster and share historical and personal audio stories along a guided route in an intuitive way targeting adult users with an interest in history?

Final Graduation Report

Mobile application of location-based audio tours

April 2021
Bart-Jan Herweijer

Student CMGT Saxion
151679

First assessor: Tim Roosen
Second assessor: Mark Melenhorst

A graduation research of Creative Media and Game Technology at Saxion Universities of applied sciences

1. Abstract

This research focuses on examining, building, and testing a mobile application and its pipeline to production. The practical research is commissioned by the workgroup of Roombeek, which has focused on commemorating the fireworks disaster which happened in Enschede on the 13th of May 2000. Along with the company Pronksnor, a concept has been formed and worked out. The mobile application is production-ready and will be available during May 2021. Through an Audio Augmented Reality (AAR), the mobile application tells personal and historical stories of that disaster. Besides looking into the history and possibilities of location-based content and services, combined with storytelling, it also recognizes certain trends which will help with the adaptation and usage of such an application.

In the research a number of technicalities were investigated, and solutions were pointed out, implemented and tested. A number of iterations to the prototype have been described, with a focus on how story telling could best be implemented in an app. And also what the location-based functionalities would consist of, like helping users navigate between the waypoints.

Through competitive analysis, best practices are shared with competitors and similar mobile applications. Along with the functional and technical design, a scope is declared for the project.

Lastly, the project describes how a pipeline and workflow can be set up to create a mobile app from concept to production. Various tools, frameworks, and methods have been researched and choices have been substantiated to create a prototype for both iOS & Android devices.

Testing throughout the various stages of the project has made sure valuable feedback is incorporated into the final design of the mobile application. Assuring a pleasant and intuitive user interface and user experience.

This research and production of the concept, prototype, and final app have been conducted to be part of the graduation process of Creative Media and Game Technology at Saxion, during the academic year of 2020-2021.

While the company Pronksnor may be the product owner, the project has been made into a joint effort, consisting of multiple stakeholders and parties.

The student has been working from his own company, and was responsible for the development of the mobile application.

2. Preface

First of all, I would like to thank Saxion Universities of applied sciences, for giving me the opportunity to do a graduate research once more. For a period of over 10 years (sometimes on and off), I have been a student to the Academy of Creative Technology.
During these years I have developed myself as a creative professional. Although I have been working on a graduation thesis before, never have I been closer to graduation.
My previous graduation took place at the cee spot, in assignment of Cooperatie CeeCee Community UA, where I'm still employed. So I am grateful that in 2017 I took that opportunity both Niels Moshagen and Jasper Schutz gave me, because all of the things I have learned over there, being able to organize, and people I've met.

Years later, Jasper Schutz and I decided to hand in a concept to help commemorate the fireworks disaster, which led to this research. From within our companies, we have been working together to make this happen. So, I'm also expressing my thanks to all who were involved in the workgroup Roombeek2020. Especially Hadassa Meijer took a big part in this.

Needless to say, I am thankful for all of the teacher who have helped me come this far. In specific Tim Roosen, my graduate coach, has shown support for me through the process. But with many of the lecturers I've had a good connection. I still remember the early years, where René Heijnen was my coach, always strict and just.

But even those who I do not mention personally, thanks for all the support and guidance throughout the years.

Lastly, I would like to thank all of my family and friends, who had to put up with my during this rollercoaster ride. A special mention worth are my parents, who have always put in an effort and enable me to go to university.
And of course, my partner Brooke Wilton, deserves lots of praise. Even though we are separated 16.000 kms right now, she has always been a beacon of light, guiding me, supporting me and lifting my spirits. She has also been one of the reasons of enrolling Saxion once more, to settle with the unfinished business.
Her unconditional love and support are worth appraising.

After this chapter will be closed, the world will be at our feet.

- *Bart-Jan Herweijer, 6th of April 2021*

3.Table of Contents

4. Introduction

This paper will conclude the research done on the usage of and creation of a mobile application that can be used to learn more about the firework disaster of Enschede, which took place in May 2000, through personal and historical stories by linking them to the physical location of the disaster. Through a location-based audio tour, users will be guided along on various hotspots of Roombeek where they will be able to learn more about the unfortunate events that took place before, during, and after the catastrophe. But it will also feature stories of the resilience of the city, coping with disaster and rising above it, rebuilding the area to overcome the adversity that hit the Roombeek area.

In the history of Enschede, there have been several setbacks for the city, for instance, the great fire on the market (1862), bombardments in the Second World War, and the collapse of their textile industries (1960's). The most recent disaster took place on the 13th of May in the year 2000. In the neighborhood, Roombeek was a big explosion. Later was found out a shipping container full of fireworks blew up in the middle of the neighborhood, of which much has been rebuild by now. 23 people, among them inhabitants and firemen died in the disaster.

However, the city has been proven to be very resilient, being able to cope with these disasters and come stronger out of such a situation.

Current situation

Now that it has been over 20 years ago, most of the neighborhood has been reconstructed. Under the guidance of Pie de Bruijn, a famous city architect and urban planner, the area has been gentrified to a certain extend. Most of the (obsolete) textile factories have been replaced by trendy company buildings, of which many have a cultural or creative character, sometimes even by upgrading and renovating those old factories. Many of the social houses have disappeared, making room for a higher social class.

Roombeek has been reconstructed from the ground up, focusing on a mixture of living, work, and leisure. A kind of revolutionary and progressive concept in the early years of this millennium.

Since then, many tourists from all over the world have been visiting the neighborhood, where old and new meets. A unique mixture has been created of museums, like the *MuseumFabriek*, *Tetem*, and *Rijksmuseum Twenthe*, cultural heritage from the old textile industries in the early stage of industrialism, and contemporary architecture.

The clients

Each year there is still a memorial service to commemorate the fireworks disaster. This takes place at the main monument, where once was the fireworks factory of SE fireworks. Each 13th of May at 15:20 people come in and homage the fallen.

Because it will be 21 years after the date, some of the local initiatives have decided to give extra attention to this anniversary. Together they have formed a workgroup, investigating, bundling, and executing the different initiatives that have been coined in the neighborhood.

Roombeek 2020-workgroups

The initiatives have been collaborating in a workgroup. This consists of members from the *Gemeente Enschede*, *MuseumFabriek*, *AKI school for fine arts*, *Huis van Verhalen*, *Tetem*, *the creative campus CeeCee*, and the neighborhood association and community center. Each of them has researched ways to contributing to those commemorations. There has been decided that for the whole month of May there will be extra activities and expressions towards the memorial of the disaster.

There have been set up multiple tracks, from educational programs to cultural heritage, and one of those is an interactive tour. This will not take place with a tour guide, but people will be accompanied by an application. This interactive location-based (audio) tour will be the topic of this applied research.

Stadslegendes

Stadslegendes, in English best translated as *Urban Myths*, is a mobile application of the company Pronksnor. Jasper Schutz, the owner of *Pronksnor*, has come up with the idea of creating location-based audio tours through the usage of the application *Stadslegendes*. The app will be very suitable for the tour in Roombeek, because it will be not depending on tour guides, thus providing people an individual experience where they can find out more about the history of Roombeek.

The first version of *Stadslegendes* has been made in 2019 by a few ICT students from Saxion. However, maintenance has been proven difficult, and also the number of users is disappointing.

The project of Roombeek 2020 will be a good opportunity to refactor the code, make it available on iOS and Android devices, and *white label*, or rebrand it to the needs of the clients. Moreover, it may persuade the users to also use the regular app and grow its user base. This will attract more users to the platforms which will help to generate a critical mass for further development and maintenance of the platform and application, making it a viable project.

Summary

In May 2020 the neighborhood Roombeek would commemorate the fireworks disaster which took place 20 years ago. Due to the global outbreak of the Covid-19 virus and its restrictions early on in 2020, the plans have been postponed for one year. In May 2021 the project will take place. One of the components will be an application, with which people can get a tour through the area, being served more information on some of the so-called hotspots. The objective is to present some personal stories of survivors of the disaster, empathize with them, and bond with the city. And of course, to pay respect and homage to those who did not survive. To show the resilience of the city of Enschede, and how people can find comfort after a disaster.

5.Problem indication

The following chapter describes the initial problem indications and sets up indicators to success to determine the scope and achievements of the project.

Client:
Due to the ongoing Covid-19 pandemic, Huis van Verhalen is not able to serve big groups of 'tourists' to come over to Roombeek for a guided tour. Also, the planned commemorations of the fireworks disaster did not continue in their desired format, due to the discouragement of group gatherings. While there still is a need and importance to tell the stories around the fireworks disaster, new ways of storytelling have been researched.

Contractor:
Pronksnor has a working prototype of its app 'Stadslegendes' but it is lacking several things: there is currently only an Android version, support on the platform is hard and it is not getting much traction due to a limited userbase and content.
The application should be redesigned and refactored to be better maintainable and feature up-to-date content. There is also a need to investigate ways how to optimize the user experience.

Indicators of success

Various indicators could determine the success of the project. The fact of building a software application divides the indicators into functional & technical specifications.
The functional indicators specify mainly from a user's perspective, while technical aspects are more focused on the design & development process.

Functional

1. Users must be able to independently walk a tour, accompanied by the app.
2. Users should get extra information on the fireworks disaster and its impact through personal stories.
3. Stories should be available by physically visiting a number of hotspots along the route.

Technical

1. The application should be up and running in the Google Play store and Apple store on May 1st.
2. Because of numerous different devices, the application should be lightweight as possible.
3. Platform and code should be easy to develop & maintain.
4. Should have an intuitive user interface and interactions, due to the target group is supposedly elderly people (whom in general have a bigger interest in history and city tours).

After setting up the initial indicators of success, a functional and technical design was made to further dig into the exact specifications required for the intended requirements of the mobile application.

The conclusions of these design documents can be found in the research section of this paper.

The users

The users of the application will be very broad, as basically the application can be downloaded by anyone who is interested and has a smartphone. Therefore, user interactions and interface should be very clear intuitive.

The workgroup has decided that the main target group would be adults, with an interest in history ,or in special towards the neighborhood Roombeek. Users will be notified of the application through diverse channels, like the public and social media. The local and national newspapers will be informed of the project, which may cause a rise in visitors and users. Besides that, also posters will be used at the partners of the project. *Huis van Verhalen,* which provides the content for the app, also hosts tours themselves. During that month they will channel visitors towards usage of the application.

The market

This chapter will focus on the market. It will briefly describe all the current trends, along with the technological advancements needed to conduct the research and, more importantly, the viability of the project. Which opportunities and threats come into play in designing and developing the application?

Current trends

There are some trends within the current market visible. Some relevant trends will be briefly introduced below.

App development:

Although the high rise in the popularity of creating apps was bigger in the first two decades of this century, there is still a high demand for custom or white-labeled apps for all kinds of organizations and purposes. The different methods to create an app have been changed throughout the years, making it easier to create a mobile application. Several frameworks have caught the attention and are rising in popularity, the two most popular being React Native and Flutter. These frameworks will be discussed and compared later on in this paper.

Digital tools and media usage:

the global outbreak of the Covid-19 virus and the pandemic causing it, life is managed differently. Working from home has become the norm, even in between the national lockdowns. Of course, this situation can be translated directly into the usage of video conferencing tools, like Microsoft Teams or Zoom. Even events have to be held digitally, meaning a huge increase in the amount of screen time. Add up the curfew that is happening in various countries around the world, and it will be no surprise people spend more time behind a screen to connect with others.

But one other thing that really caught attention is the rise in audio-based media. This can be a product of the increase in screen time. Apps like Clubhouse became very popular in the first months of 2021, moreover to the number of celebrities using and nudging this. But also making and distributing podcasts is something many turned towards (with or without video stream).

LOCATION-BASED APPLICATIONS: There has been a rise in the usage of location-based applications. With apps like *Snapchat* and of course *Pokémon Go* depending heavily on location-based technology and interactions, you can see that from a market-pull perspective, location-based applications are getting more and more common nowadays. Apps like *Uber* disrupt the taxi industry due to the real-time location updates from both the users as well as the Uber drivers. This creates opportunities for app developers when they make smart use of location-based technology.

PERSONALIZED EXPERIENCES: As we are shifting more and more towards a sharing economy you see an increase in the need for personalized experiences. These custom-tailored experiences invoke a connection with the target group, mainly because they fulfill their needs to be original and have a sense of belonging. Of course, the increased state of individualism takes a big part in this too.

In the literature studies, which can be found at chapter 4, some of the trends will be described more in detail.

Competitive Analysis

Because the project has already been assigned there is a different look at the competition. Moreover, because of the local market and the short duration of the project, it is serving a niche market.

However, if we look at competitors, we can learn from their cases, making use of the best practices. The competitive analysis has been done in the design document in appendix x.7.1 and will be summarized in chapter 4.2 .

6. Research approach

The research approach is settled on the design thinking method. In these methods, there are five phases to go through in order to come up with an end result.
This design thinking process was described in the final implementation plan.

Besides using the guidance of this method, the research itself has been conducted to be an applied research. The process of prototyping and testing, and iteration over this, took a central place. However, during the empathize and inspiration phase also desk research has been carried out.
The chapters 1-6 will focus more on that initial information gathering, while the later chapters, starting from chapter 7, will have a more practical approach.

Accompanied by this written report, a design document has been made, which consists of several of the design choices and justifications needed to build a prototype of the app.

7. Literature studies

Most of the desk research, as described in this chapter, has been carried out to learn more about the available solutions and approaches. This will help to give an overview on the design choices to be made, by looking into current trends, technical feasibilities and competitive analysis. Lastly some theories were used to back up the choices made throughout the project.

7.1. Trends & technological advancements

7.1.1. Pokémon GO paving the road

Pokémon Go was launched in July 2016 and became an instant success. One of the reasons was nostalgia among its users. Young adults who have been raised with the cartoon show of Pokémon and playing the early games on Nintendo's Gameboy. The game is developed by *Ninantic*, who what prior experience with the Augmented Reality app Ingress. Some of the key elements of Pokémon Go have paved the road to the *Vuurwerkramp Verhalen*-app:

- Location-based (through GPS)
- Usage of hotspots, to stimulate users to walk and visit
- acceptance of usage of Augmented Reality

Still today it is one of the most popular mobile games, with over a billion downloads in total. (Iqbal, 2021). By combining a freemium game, featuring nostalgic elements and location-based content, it is clear there is a huge market for mobile games and location-based mobile applications.
Other examples of location-based applications are Foursquare and Layar.

7.1.2. The rise of audio content

Due to the Corona pandemic there has been a huge increase in the production and consumption of podcasts. According to Forbes "In 2020, an estimated 100 million people listened to a podcast each month and it's expected to reach 125 million in 2022" (Adgate, 2021). This is due to the fact people are spending more time at home, working long days behind their screens.
For media consumption, it is a nice change to be able to listen to content instead of viewing it.

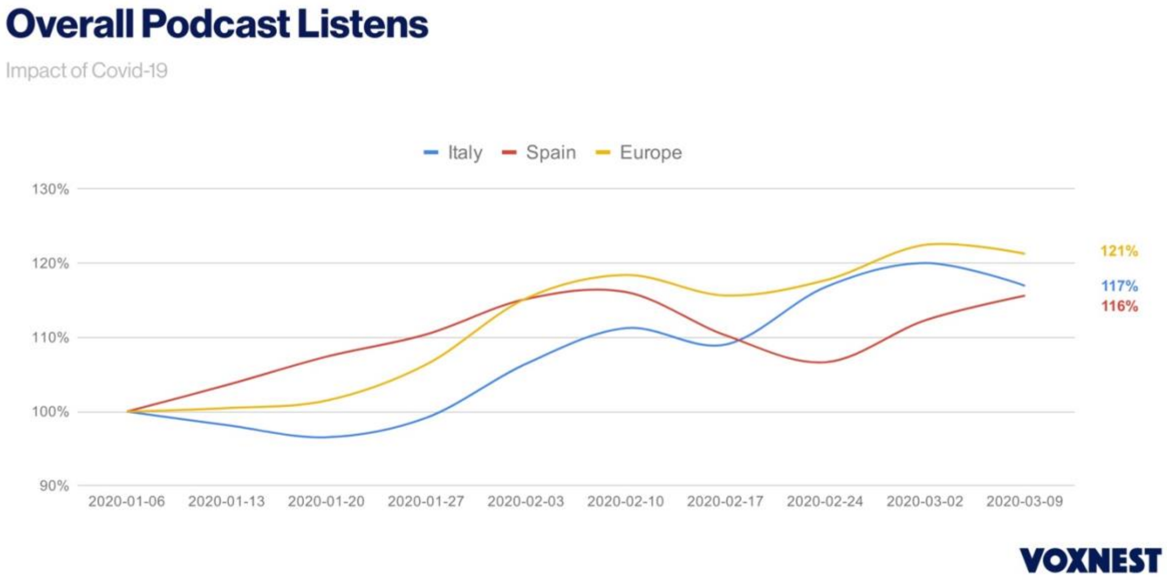


Figure 1: Overall increase in podcast listens in 2020 (Amburgey, Ive, 2020)

Besides the rise in podcast, a new contender showed up in the past year. Social media platform Clubhouse has gained a lot of popularity, mainly because it being used by high society and celebrities. In Clubhouse, there is the possibility to make audio rooms. These rooms can cover certain topics, and if set up publicly, all users can join as listeners. There is a 'stage' where speakers are allowed, which can broadcast through their microphone, as the app is audio only. The visitors can raise their hand in the app through a button, so they can get invited to take place on the stage. Everybody on stage can take a part in the discussion.

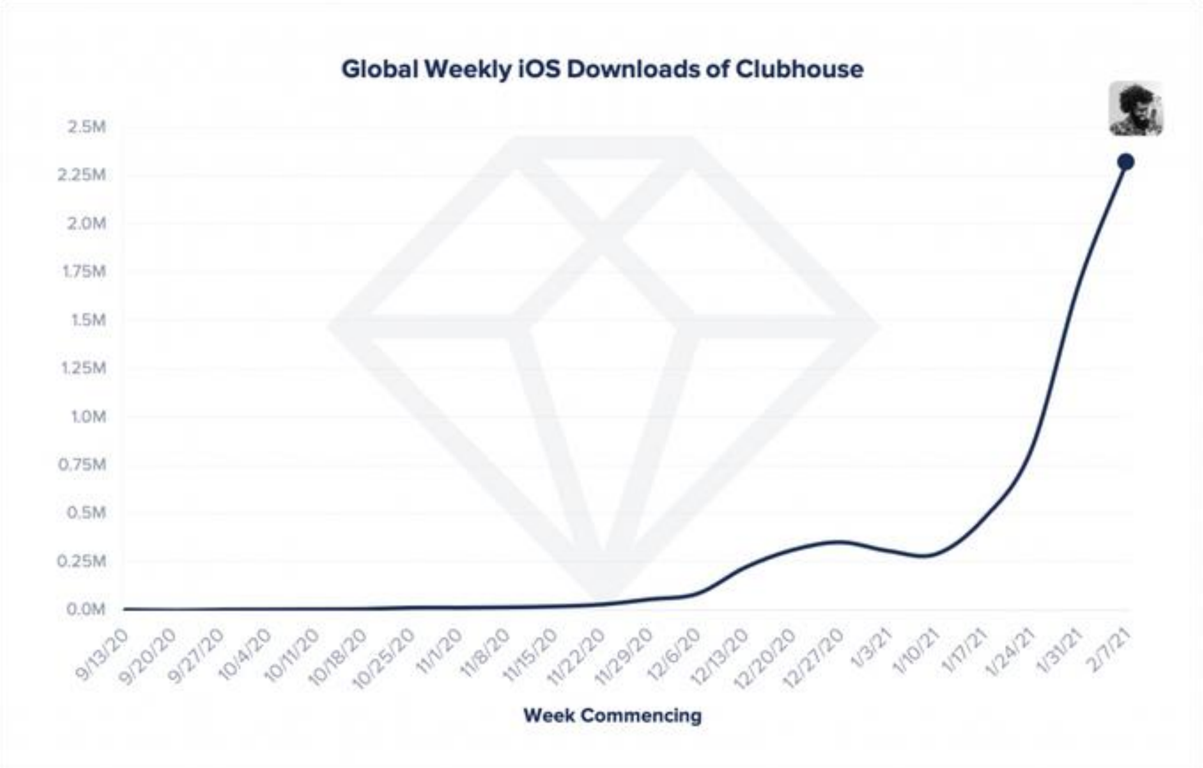


Figure 2: the sudden rise of the audio-only social media app Clubhouse (TechCrunch, 2021)

- https://audioar.org/
- https://www.bbc.co.uk/rd/blog/2019-03-audio-ar-geolocation-soundwalk
- https://www.theverge.com/2020/6/16/21293372/bose-augmented-reality-frames-glasses-over

-

7.2. Competitive analysis & best practices?

Frameworks & pipeline

There are different ways to create an app. The most common ways are either using native code or frameworks. Where native code compiles directly to the code run on the OS, the framework usually makes use of a higher level of abstractions, and through a pipeline of trans piling and compiling into an app. Frameworks have the benefit of writing code once, deploy it everywhere, whereas native code is written platform-specific. Usually, when writing native code, it is speedy and optimized for the platform, being able to make use of the functionalities of the OS directly, where frameworks rely more on translating code into instructions for the operating system.
Android uses native Java code, where iOS runs its own Swift language or Objective-C.
The most well-known frameworks are currently React Native and Flutter.
In the design document in appendices x.7.2, there is a more in-depth comparison and justification on the chosen approach of picking a frontend and framework for building this app.

A pipeline describes the different stages of producing an app and which tools are used at what stage. For example, the design thinking method can also be seen as a pipeline. It describes the process from start to finish, in this matter, from concept to production-ready application which can be downloaded in the store.

7.3. UI & UX theory

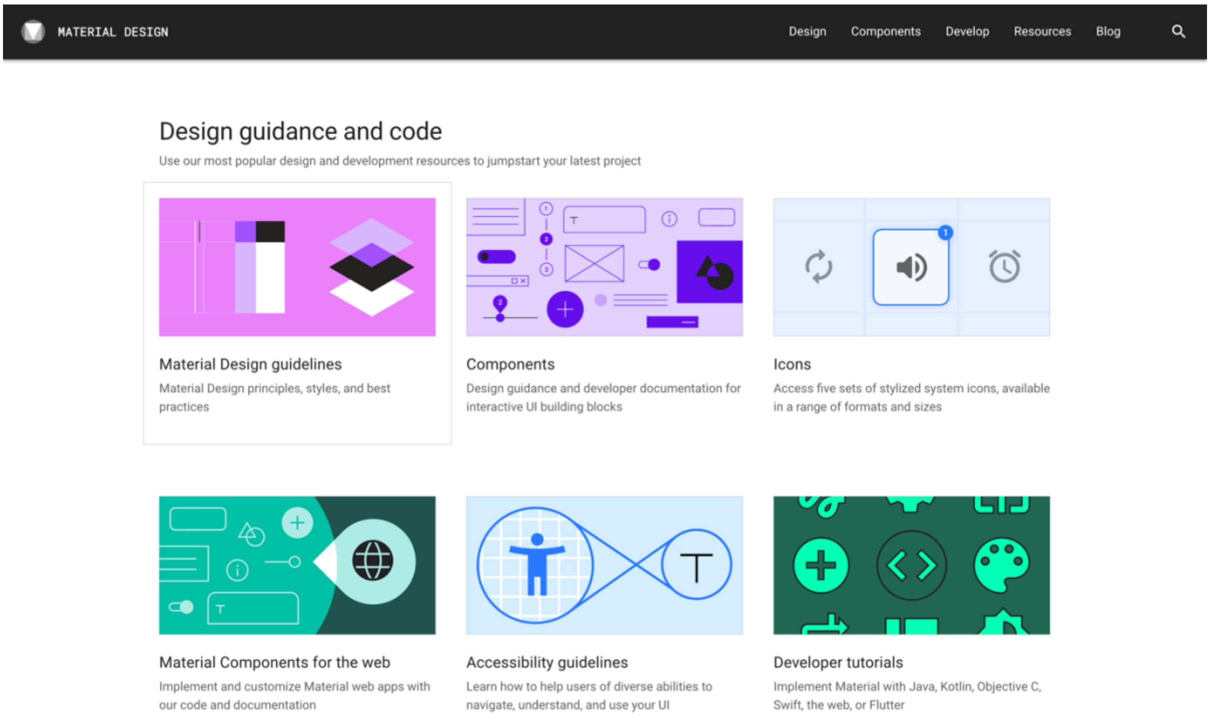
User interface (UI) focuses on building the interfaces upon which users can interact in a digital application. It mostly focuses on the looks and style of elements, while User Experience (UX) focuses more on the underlying actions and mechanisms provoked by interaction with the elements on the screen.
Jacob Nielsen’s Heuristics (Nielsen, 2020) is a very well-known tool for creating and rating UI systems, along with the Golden Rules of Interface Design by Ben Schneiderman. (Schneiderman, 2016)
These focus on general principles for interaction design.
The lessons and rules learned from Schneiderman and Nielsen can be captured in 4 simple design principles (Adobe, 2019):

1. Place users in control of the interface
2. Make it comfortable (and enjoyable!) to interact with a product

3. Reduce cognitive load
4. Make user interfaces consistent.

4.4. Material Design

Material Design is a design system to build digital experiences through using graphical interface elements, which are standardized. It is inspired by the physical world and its textures, including how they reflect light and cast shadows. (Google, 2020). Mimicking the feeling of materials like paper. It is often used stacked on top of each other to reimagine the psychical world and logic, creating emphasis on elements that are on top (on thus cast shadows). Besides that, it is the whole system consisting of elements like typography, (white) space, and colors. This creates hierarchy, meaning, and focus that immerse viewers in the experience. (Google, 2020)
By reusing a set of components, these building blocks can build recognizable user interfaces that help the user guide through an application and user actions.
It is highly customizable without losing the expected user feedback, like ripples when a user taps a button.



Design guidance and code

Use our most popular design and development resources to jumpstart your latest project

Material Design guidelines

Material Design principles, styles, and best practices

Components

Design guidance and developer documentation for interactive UI building blocks

Icons

Access five sets of stylized system icons, available in a range of formats and sizes

Material Components for the web

Implement and customize Material web apps with our code and documentation

Accessibility guidelines

Learn how to help users of diverse abilities to navigate, understand, and use your UI

Developer tutorials

Implement Material with Java, Kotlin, Objective C, Swift, the web, or Flutter

Figure 3: a screenshot from the website of Material Design, with the key features (Google, 2020)

8. Problem definition

This May, there should be an application launched that provides location-based audio tours to commemorate the fireworks disaster of Roombeek, Enschede. The goal of the tour will be to share some personal stories of those who were involved and affected by this misfortune. It should keep the memories alive, and help people find comfort and cope with their grief, but on the other hand a tribute to both the history and present of that neighborhood.

8.1. Main & sub-questions

How can a mobile application contribute towards the commemoration of the Fireworks disaster and share historical and personal audio stories along a guided route in an intuitive way targeting adult users with an interest in history?

1. What type of content and media works well in storytelling on location?
2. What is the best way to register a user's location and make use of the hotspots?
3. How can users be navigated along a route consisting of points of interest?
4. How can this app be valuable to the company Pronksnor?

9. Scope

The following chapter will describe the scope of the project and the research carried out. The scope described here will relate closely to the one written in the initial offer towards the client, only differing the following chapter will focus more on the research written with Saxion graduation criteria.

During this research, not only practical 'how to' questions have been asked and answered, but there is also a focus on the improvements implemented by the hand of the user's feedback. This feedback loop is an essential part of the design thinking process.

9.1. Scope of the product

The following chapter will narrow down the scope of the project. Of course, there are a number of restrains, requirements and edge cases in the project. As the student will be mostly focusing on the technical implementation of building a mobile application, a Moscow-list is inserted below.

Moscow-list

By making use of a Moscow-list t the priorities and requirements for the project can be set.

Must Have	Should have	Could have	Won't have
Working application for IOS & Android	Location awareness (through GPS /Bluetooth / 2/3/4G triangulation)	Real time updates on current location	A lifetime support for both platforms
A map with overview and pins on hotspots	A route between the hotspots	Navigational system between the hotspots	Augmented reality capabilities
A clear visual identity	Designed elements which align with corporate identity	Splash screens and animations	A multiplatform social media campaign or advertisement
Easy and intuitive control	User interface an UX-elements tested with target audience	Walkthroughs and aiding systems for impaired & technically challenged.	Extensive support through email / phone / help section website
Options to manage the content	A manageable backend	A content management system	Extensive options for users to control content

10. Research & outcomes

Throughout the whole project, research has been done. Part of the desk research took place in the initial phase. This is mostly described in previous chapters. The following chapter will describe more of the practical side of the research, which took place during the prototype and test phase of the design thinking methods.

10.1. What type of content and media works well in storytelling on location?

Choice of content

A big part of the application relies on the content: users will be guided through a story by visiting different physical locations. These so-called hotspots contain a chapter of the story. By walking the route, each part of the story will get unlocked and become listenable.
That was the initial idea. But with the stakeholders, we have also explored different options to deliver the content. Before digging deeper into the considerations on the form and substance of the content, it is good to set the goals.

The goal of the storytelling

The goal of the stories was to give a historical view of the history of the fireworks disaster. Through personal stories, users will get a grasp of the impact it has made on the lives of the survivors. That has to be approached with a certain dignity, as 23 lives were lost in the disaster. And not to forget the emotional and physical damage the explosion brought into the lives of those who experienced it firsthand.
Even today, the cause of the explosions has been up for discussion. As a workgroup, we have decided not to go into the cause of the explosions, as it still is a delicate matter.
Besides all the hurt and grief which came from the unfortunate event, we would also like to point out a message of hope. How resilient the citizens can be, how misfortune can bring people together, and how to deal with grief and disaster.

The type of content

The base of the content would be provided by 23 interviews with involved people. These interviews have been done by Marco Krijnsen in commission by Huis van Verhalen and de MuseumFabriek. This project, Oral History, would be a historical report for the archives, to preserve these stories. Initially, these stories should not be made public.
However, along with this project, these video interviews have been made accessible for usage within the application.
The quality of the videos was not that great, the images being not too crisp and clear, and the audio could be boosted to become clearer.
In the original concept and offer, the application should only use audio files, but during the stakeholder's meetings, we have discussed whether video files would be of additional value over audio.
Through ideation, comparison, and testing, the two options have been weighted in: a definite choice has been made to use audio stories instead of video stories.

Below you'll find some of the reasons which justify this choice. These choices were discussed within the workgroup, but eventually, the product owner did make the final decision.

Testing different types of media

As stated above, Jasper from Pronksnor made the final call to use audio content over video. During an expert interview with Jasper, both options discussed in-depth, resulting in a decisional balance sheet weighing the pros and cons.

	Pros	**Cons**
Audio	only listening, being able to look around	headphones preferred
	less data usage	no images
	Firebase storage can be used	users have to use imagination more
	more immersive? (assumption)	no subtitles
	Content already available	
	Editing audio is easy, no b-roll needed	
	option to autoplay	
Video	more data usage	if done on location, the connection for the user might be bigger
	linking to YouTube, need to be private videos	subtitles
	The video quality of the interviews not great	possibly more immersive
	redo videos take up more time if done right	usage of historical images
	The video takes the active attention of the users	The video takes active attention of the user
	more work because of filler shots and b-roll footage	

Besides listing the possibilities and the effects of using either video or audio, a small experiment was conducted.

In this experiment, the student did create two screens. One screen was containing a video player and the other an audio player. The video was loaded from YouTube, while the audio file made use of the Firebase storage.

In a field test, both the product owner and the student consumed the media files on location.

Both the test subjects felt a stronger connection to the place by only listening instead of also having to watch the screen of a mobile device.
To watch the video optimally, turning the mobile device to view it in landscape mode was preferred. The application is made to use in portrait mode, so this required additional action from the user.

Although not being a total valid test result, due to having to repeat the media instead of testing first impressions, it did give a good pointer to the style of storytelling and the differences. The self-approbation of the testers was not taken into account in the test. It did provide the confirmation of the hypothesis of audio being the correct media to deliver the story.
One of the clear differences in the test was loading times. While the audio files were loaded and played back instantly, the video took some time loading it from an external source. To watch the video optimally, turning the mobile device to view it in landscape mode was preferred. The application was made to use in portrait mode, so this required additional action from the user.

Additional content

Next to the main media files, containing the story files, additional content was placed in the app. The names of the point of interest and the geospatial data were obligatory because the app would be using them. To provide more data on the points of interest, the options for photos and text were added through a detailed screen.
Through a detail-screen, images and text could be added as additional content, enriching the app with more depth both in information as well as screens. With having detailed screens, the usability of the app has become better.
Although the option has been coined by the stakeholders to use more images, a single image was used per hotspot. These images contain a photo of nowadays point of interest, like a building or statue. This will help guide the users towards the correct location to follow along the tour.
The imagery coined by the stakeholders would contain pictures of before, during, and after the fireworks disaster. This may be included in a later version of the app but was beyond the scope of the initial launch.
The text is used to give a bit of history of the points of interest, and it can display temporary information, like the opening hours of the museums.
Recording new audio files, after the content has been finalized was not an option because of the extra time investment needed.

10.2. What is the best way to register a user's location and make use of the hotspots?

Goals and methods

Several options have been researched and reviewed to use location-based features as intended. The goal of this research is to look into the possibilities of checking whether a user

is nearby the point of interest. These points will act as hotspots. When near these hotspots, the users will be able to listen to the story at that exact location.
Various methods were selected to compare. These methods were all put into a SWOT analysis. Also, desk research has been conducted to see which of the methods is the most popular to use, and easiest to implement. By looking into tutorials online, check for packages compatible with Flutter, and other forms of literature studies, a selection was made.
Lastly, the ideas have been discussed with the owners of 100%FAT, to hear their professional opinion. 100%FAT is a company that creates interactive media, installations, and experiences. Mainly because they have a lot of practice with using public spaces, their expertise was valued.
The options have been explored in the implementation plan and copied over in the Design Document.
The usage of QR-codes or Bluetooth beacons would mean external resources are used. QR-codes have been discouraged in the meeting with 100%FAT, because they would rely on a physical location, and even more important, enough light to scan these codes. Therefore, using the GPS or network capabilities of a mobile device should be the most logical choice. By removing externalities, the risks of errors can be reduced. This would also be capable of comparing a user's location with the location of the hotspots, making it possible to provide navigational directions to the users.

Getting a user's location

The GPS data can be easily used from a mobile device through the usage of the packages available in Flutter. The two most popular packages are Location and Geolocator. Both have been tested and finally, the choice has been made for Geolocator, as it has extensive features, and is the most popular of the two.
The package will provide some functions. The location of the mobile device will be outputted as a stream of data. Each time the location gets an update, so does the stream. This stream of data can be used to indicate the user's position on the map screen with a blue dot. At the same time, these positional data can be used to calculate the distance and **bearing** between the coordinates of the hotspots.

Comparing user location to points of interest

After receiving a user location, which consists of latitude, longitude, and heading, it is time to use these values. The heading is the direction the user is currently moving in or pointing the mobile device towards (in the case of a mobile device having a gyroscope and compass).

To compare this user location with the geospatial location of the point of interest, different techniques can and will be used. Below the most common techniques will be explained.

The phone will emit its location while in usage, and these GPS coordinates can be compared to the position of the points of interest which are used to store the content.

Subtraction of coordinates

Gives a rough estimate of the distance between the phone's location and the location of the hotspot by comparing the sets of coordinates.

By having two sets of coordinates, it is possible to subtract the latitudes and longitudes from each other, leaving the difference in coordinates to get an estimate of distance. However, this method is pretty cumbersome, as the earth and its coordinate system do not represent a plain but a sphere, which means close to the poles, the longitude medians are closer together. And of course, the same goes for the latitude and the poles. Near the equator, the medians are the widest as compared to the poles.
Not to mention the two sets also need to take into account the Pythagorean theorem unless latitude or longitude is the same with the start and end coordinates.
All in all, this is a very premature option for calculating the distance between two points on a sphere, whereas the options below are more relevant and are more often used in navigation.

Geohashes

Encoding the coordinates in a string of numbers and letters, where each additional character provides further precision.

“Geohashing is a geocoding method used to encode geographic coordinates (latitude and longitude) into a short string of digits and letters delineating an area on a map, which is called a cell, with varying resolutions. The more characters in the string, the more precise the location.” (PubNub, 2021)

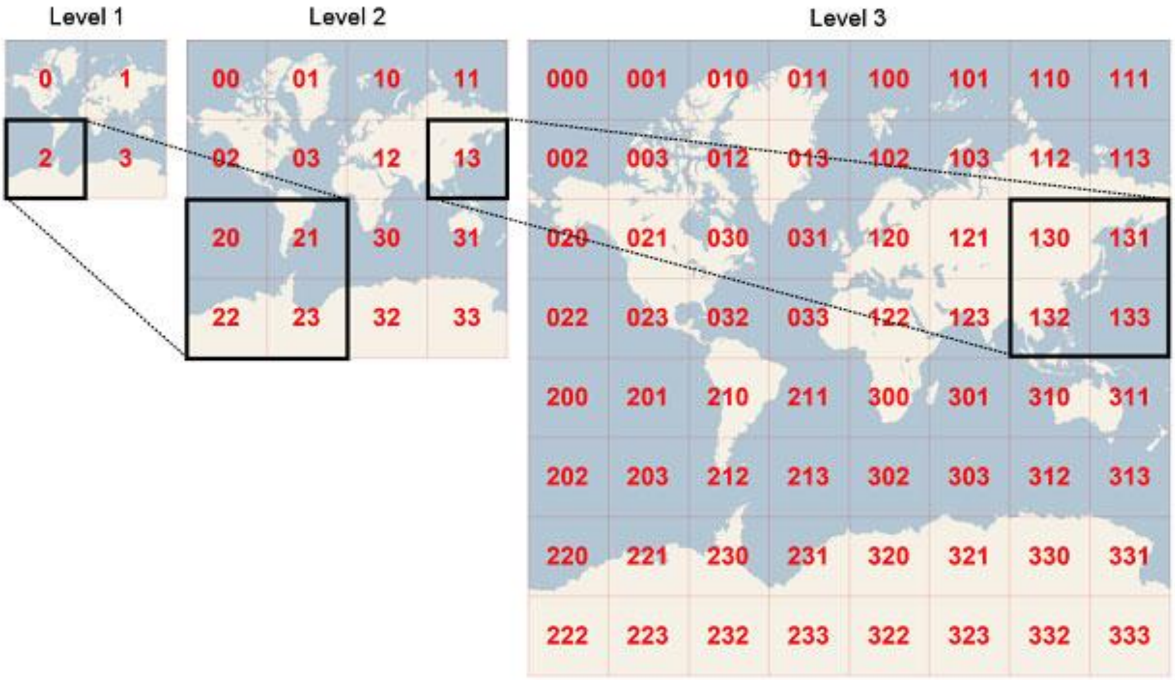


Figure 4: A graphical representation of the geohash system (PubNub, 2021)

The method of geohashing works a bit like postal codes in a city. It is an encoding system where each character added increases the accuracy. It does not specify a point but a plane on the sphere. It makes use of a Base 32 value (meaning each square gets divided into 32 equal squares, which recurs with each symbol added. A maximum of 9 characters is used to have a resolution of approximately 5x5 meters.
This method is commonly used to check the proximity of two sets of coordinates.

Haversine formula

The haversine formula, or great-circle distance, measures the distance between two points given their longitudes and latitudes.

To calculate the distance between two points on Earth, often the Haversine formula is used. It computes the great-circle distance, which is the shortest distance between those points. The line is measured over the surface of the sphere, taking into account the curvature of the Earth.

Going deep into the formula is beyond the scope of this research, but luckily the Geolocator package has the functionality built in.

Below is a graphical representation of how the formula works, and an explanation of the math behind the formula.

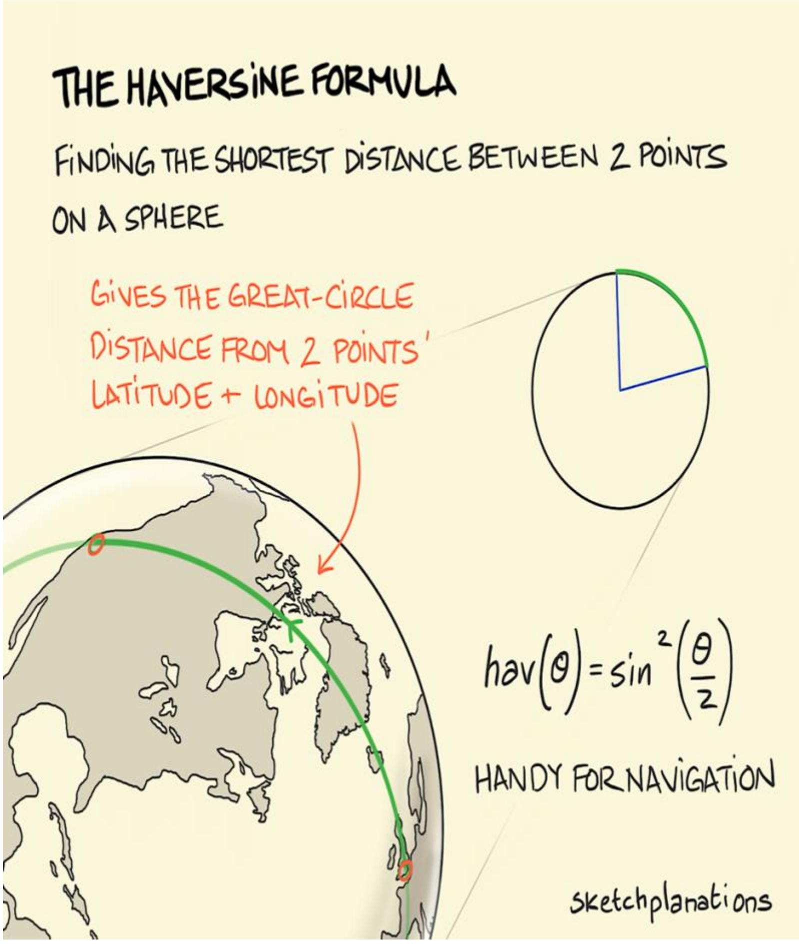
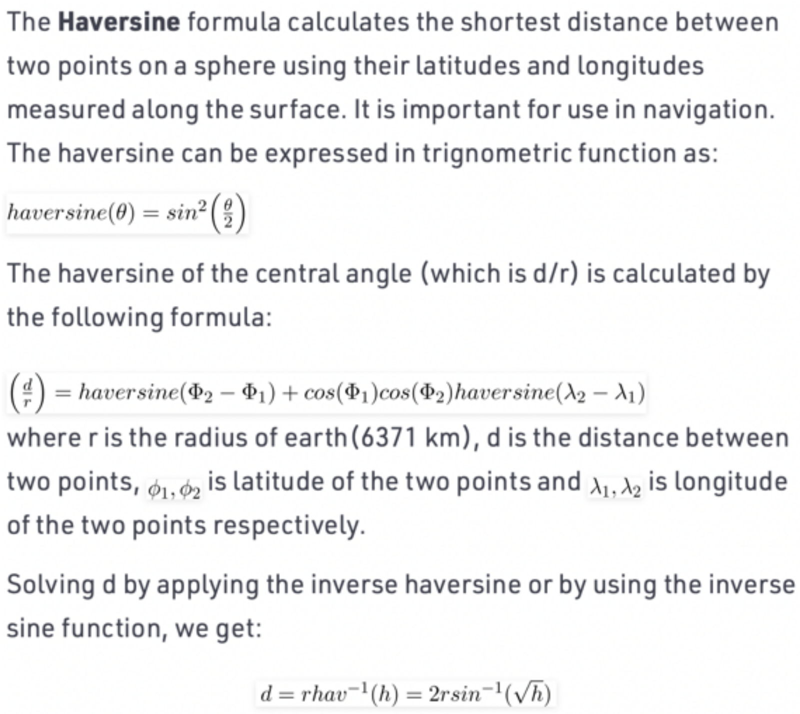

THE HAVERSINE FORMULA
FINDING THE SHORTEST DISTANCE BETWEEN 2 POINTS ON A SPHERE
GIVES THE GREAT-CIRCLE DISTANCE FROM 2 POINTS' LATITUDE + LONGITUDE
$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right)$
HANDY FOR NAVIGATION
sketchplanations

Figure 5: visual representation of the Haversine formula (Sketchplanations, 2021)



The **Haversine** formula calculates the shortest distance between two points on a sphere using their latitudes and longitudes measured along the surface. It is important for use in navigation. The haversine can be expressed in trignometric function as:

$$haversine(\theta) = sin^2\left(\frac{\theta}{2}\right)$$

The haversine of the central angle (which is d/r) is calculated by the following formula:

$$\left(\frac{d}{r}\right) = haversine(\Phi_2 - \Phi_1) + cos(\Phi_1)cos(\Phi_2)haversine(\lambda_2 - \lambda_1)$$

where r is the radius of earth (6371 km), d is the distance between two points, ϕ_1, ϕ_2 is latitude of the two points and λ_1, λ_2 is longitude of the two points respectively.

Solving d by applying the inverse haversine or by using the inverse sine function, we get:

$$d = rhav^{-1}(h) = 2rsin^{-1}(\sqrt{h})$$

Figure 6: The Haversine formula briefly explained (Haversine formula to find distance between to points on a sphere, 2021)

Conclusions of the desk research

Both Geohasing and the Haversine formula are commonly used to perform calculations on coordinates. While Geohasing is great for comparing the proximity of locations, the Haversine formula is mostly used to calculate a precise distance. While both methods have some edge cases, for the mobile application they will be sufficient, as locations will be close together. The Geolocator package can provide the calculation of distance through the great-circle distance method, while the package *geoflutterfire* makes use of the geohash to show points in proximity of the user.

10.3. How can users be navigated along a route consisting of points of interest?

Goals

The goal of this question is to find out how users can be encouraged and guided to follow the path along the hotspots in the right order. This is of importance because the storyline will be implemented over those hotspots, with a tension arc and narrative.
After initial brainstorming on possible solutions, and by looking at competitors' implementations, the different options were checked for effect and feasibility. Eventually, a choice was made to implement a combination of solutions to help guide users to navigate to the hotspots.

Comparing the options

During the competitive analysis, which can be found in chapter x of the appendices, one of the main targets was finding out suitable ways to show a route to the users. And being able

to follow the path along the hotspots. While some of the other apps provide both tours as well as a free-roam mode (where you can visit the points without following a tour), it strives that users will follow the points of interest in a specific order. This choice relates to the story being crafted in a way that the story relies on the previous chapters of the story, to give the best experience.
The following options for displaying the route on a map screen and adding navigational features were compared on clearness, feasibility, difficulty to implement, and documentation.

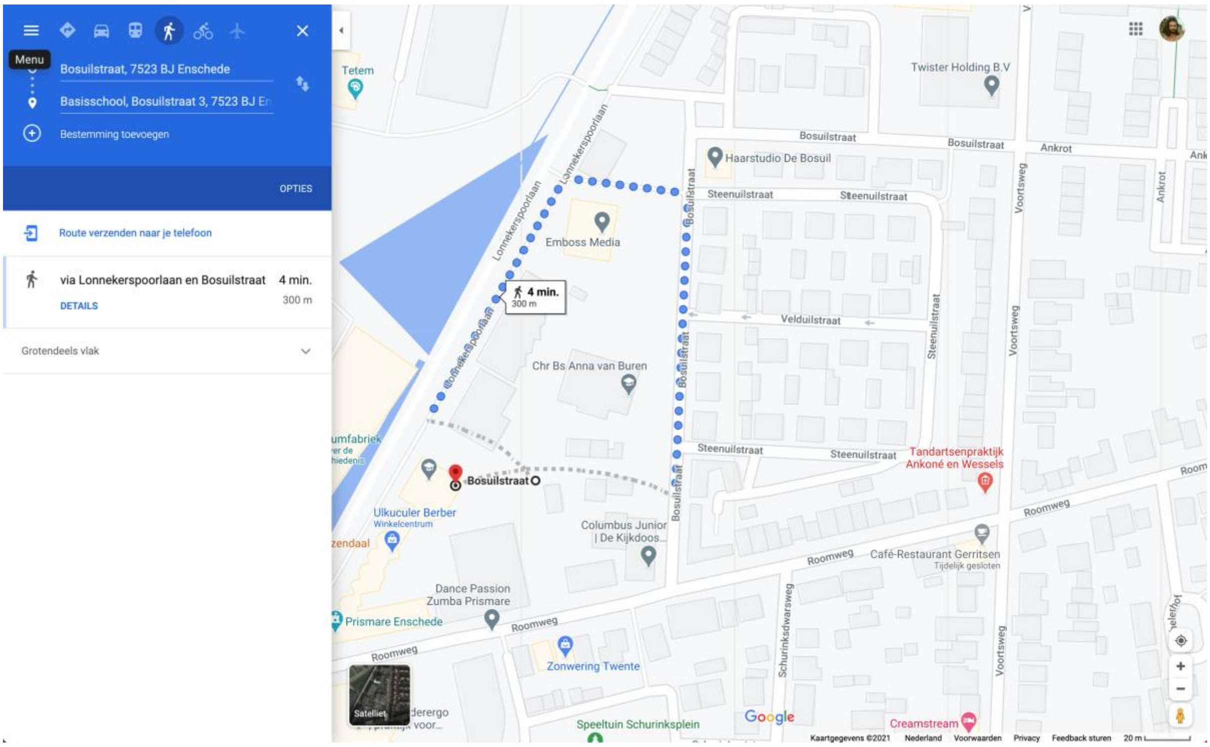
1. Drawing a route on the map between the markers
2. Turn-by-turn navigation
3. Numbering of the markers
4. Using a compass needle for direction and distance to the selected hotspot.
5. Guiding the user to a location through external navigation app like (Apple / Google) Maps

After the first feedback session on the user experience (Feedback Friday, 26th of November), it became clear users ideally would navigate in a way that's familiar to them. Plotting a route on a map and offering turn-by-turn navigation would be preferred. Most users will be able to navigate using an app like Google Maps, Apple Maps, or Waze.

However, these are not standard functionalities within Flutter or any of the packages chosen early in the process. Although it is possible to draw a line on the map, representing the route to be followed, there were some issues run into: The Google Maps API only can draw lines along official routes. And because the chosen route would go through pedestrian areas, like squares, or indoor locations, this did not give the desired result.

Polylines

These results were achieved by adding the Dart package Flutter Polyline Points. This option was somewhat unsatisfying, as some of the best parts are when people get off the official roads and visit the square at the school (hotspot number 2) of the area around de MuseumFabriek. Even adding extra coordinates between two points of the route would not make the wished outcome, so this option was insufficient.
It would be possible by drawing a line by hand, which could be stored as a vector (where the shape of the line is described) and linked to the zoom levels of the map, but some of the other options were more unique and appealing to the developer and product owner.



Bosuilstraat, 7523 BJ Enschede

Basisschool, Bosuilstraat 3, 7523 BJ En

Route verzenden naar je telefoon

via Lonnekerspoorlaan en Bosuilstraat 4 min. 300 m

Figure 7: Google Maps showing the directions API not being able to use pedestrian area. In this example it is a schoolyard, where it is unable to navigate over (Google, 2021)

Turn-by-turn navigation

During the desk research, no working examples of turn-by-turn navigation featuring Google Maps were found. Building an own solution to accompany this idea was out of scope, as it would take up too much time.

There was a promising package found, but this was relying on *Mapbox*, a counterpart of Google Maps. This package was found late in the process, so prior choices were already made on using Google Maps.

Numbering the markers

A different idea was to number the markers. This would give the users a certain level of freedom by navigating themselves to the next point. The markers need to have custom icons, as only standard markers are allowed. There is the option however to give them different colors or play with opacity. In the app Stadslegendes, the opacity was used in such a way where the opacity would be non-transparent for the next marker and gradually become more transparent for markers later on the route. Feedback from the users of Stadslegendes made it clear this was not sufficient enough.

The markers would need some states anyway, for being visited would require a graphical representation of being finished (by using a checkmark) or being within reach of a user (to trigger the audio file) so this seemed a good way of using visual cues.

Unfortunately, markers are a class within the Google Maps Flutter package and not a widget. A widget can hold a state easily, for using dynamic values in a marker some tinkering is needed. Luckily a helper class was found on the internet, which could be used to achieve

such a result. Another option would be to use an image of each marker that contained a number. And convert these to bitmaps so they can be used as a marker.
A final execution is yet to be implemented, but a proof of concept was already made. The benefit of choosing a dynamic option, meaning text will get converted and placed along with the marker, is that it offers more flexibility over static images.

Compass arrow

Another idea was to implement a compass rose on the screen, pointing towards the currently selected point. This would be a more playful solution, as users would be given more freedom to reach the destination instead of drawing a path.
For implementing such an arrow, several things are required:

- user location and heading
- the rotating of the map
- coordinates of the point of interest
- bearing between the user and the point of interest

Luckily, the Geolocator package offers two helper functions, to calculate the distance and bearing between two points on the map. Under the hood, it uses the Haversine formula to calculate the distance.
These can be used to give the needed information and show it on the screen.
The terms heading and bearing will be explained and shown how to put to use.
The heading gives a value that represents the direction a user is going in. This is measured from the North, so if a user is facing North, the heading will be zero, and when facing West, it will be -90 degrees, and East will be 90 degrees.
The bearing will be the number of degrees the map (or the camera viewing the map) is rotated from the North. It is measured clockwise from North, which means a bearing of 90 degrees will mean the camera, and also the viewport will point east.

The *bearingBetween* function from Geolocator takes two geocoordinates and determines the angle between those compared to the North. This will make an arrow point towards a selected location as long as the map is facing North.

By adding the result from the above function along with the user's heading, an icon consisting of an arrow can always be pointing towards a selected point, no matter the user's location or direction.
Lastly, the *distanceBetween* function provided by the Geolocator package returns a float value of the distance between the two points in meters.

So the idea is that the user can follow the compass arrow, and the distance to the point will be counting down until they are close enough to trigger the audio story.

Field test

During a field test, which took place on the 18th of March, two stakeholders were invited to test the application for its navigational features. During this test, two mobile devices were given to them. One phone would contain only the markers, the other the stories and route descriptions. The supervised and moderated usability test made clear several things that needed to be improved.

First of all, on the map screen, it should become visible in what order the points should be visited.
The map would center the user's position, and rotate towards the heading of the user, but this was not sufficient.
So, it has been decided the users would need more visual feedback.
This will be done by numbering the markers. It will be combined with the compass feature, which did not work at that time.
Furthermore, the blue dot representing the user's location did not provide enough information. The users were unable to zoom, toggle heading, or explore the map because the map was fixed on the current user's location.
These points of feedback were taken into account.

Solutions

With the field test regarding the navigational features, valuable feedback was gathered. This feedback was implemented in the next iteration. Toggle buttons were made to either explore the map, meaning move the camera independently from the user's location. Secondly, a toggle was added to rotate the map on the user's heading, or to have it static (pointing North). These features are also found on other mobile navigation apps.
Ideally, the points of interest would be selected automatically. By launching the app, the first point should be selected and give directions to it, by implementing the compass arrow solution. After visiting the first point, the marker should change to a different state (the finished state containing a checkmark), and the next point of interest should be automatically selected. This will enable the compass arrow to update accordingly, making navigation easier for the user, ensuring the user will be able to follow the stories in the right order.

Conclusions

Ideally, turn-by-turn navigation would be implemented in the app, as none of the competitors provide this in a good way. However, it is not a necessity. By numbering the markers, users are gently nudged to follow the predefined route. By not drawing a polyline and plot out the route, users tend more to explore the region and look more around the environment than on their screens. Besides that, plotting a line on a map with the usage of Google Directions API only allows to make use of official roads and doesn’t take into account navigating through a building or pedestrian area.

Markers should change visually to indicate they are within proximity and have a finished state which will be triggered after the visit
Markers should be numbered to guide users to follow the predefined route
Turn-by-turn navigation is out of scope due to complexity & time constraints
Optionally a (compass) arrow can point toward the next marker and show the distance to that point.

10.4. How can this mobile application be valuable to Pronksnor?

To make this application of added value for the product owner Pronksnor, some things were taken into consideration. The first thing mentioned was the maintainability of the project.

Secondly, modularity and reusability were taken into account when developing the application.

As mentioned earlier, the company Pronksnor has been working on the concept before. In 2019 a group of students made a prototype for the app Stadslegendes, where personal experiences and stories can be shared at a physical location. However, the maintainability of the code is difficult since the students left.
The goal is to provide Pronksnor with a solid basis, to which new features and functions can easily be added. Ideally, the product developed by the graduate student can be put into the market as a white-label application with minor adjustments. Along with the product owner, the student made a list together which would help them to create an app which can be used for different clients.

Version Control

With version control, all progress within the development will be stored. As a result, a working history can be reviewed. Working with Git has been the defacto standard for software for quite some time. When trying to keep all the different functions separated in different branches, it will become easier to use certain features in future projects. It also helps to hand over the project to another developer, as he or she can trace back the steps taken by the student, and make changes where needed.

Firebase: Backend as a Service

One of the ideas was to make it possible to swap or adjust content easily. By using Firebase's Firestore, content management can be done directly in the web environment of Google's Firebase (https://console.firebase.com), either by creating a new project or by using different collections. Next to Firestore, a NoSQL database solution providing near real-time updates and queries, Firebase's toolbox has extensive features. One of these is cloud storage, where all the media files can be stored and linked. Firebase also features hosting & deployment of web apps, extensive methods for authentication & SSO (single sign-on) options, along with tools for monitoring usage like analytics, A/B testing, and user segregation.
A final feature of Firebase worth mentioning is the possibility to make use of cloud messaging to send notifications and increase user engagement.

Flutter

Although the framework Flutter is relatively new, it is gaining more and more traction. By creating widgets for different elements and screens within the mobile application and separating business logic from the view layer, the app becomes much more modular and therefore reusable.

Code commenting

By commenting on the code and writing some documentation alongside this project it should become more clear how the functions within the code are working, making it easier to change functionality at a later stage.

All of this will help make the code easier to read, understand and maintain, thus providing extra value to the product owner.

11. Conclusions

Location-based social media, Audio Augmented Reality (AAR), or soundwalks are not new concepts. Especially in the high rise of the technology bubble (first decade of this millennium) a wide variety of applications has been using location-based content. However, there are a serious number of drawbacks indicated throughout history. With the rise of games like Ingress, and of course Pokémon Go, games have been heavily reliant on location-awareness and cleared the way for adaptation in Augmented Reality usage on a smartphone. For audio walks, the touristic sector seems most fitting, to enhance the experience in example city trips or a visit to a museum.

It is proven that from a learning experience, Augmented Reality (and thus being present at the location) in combination with audio enforces a better learning experience and higher levels of enjoyment.

On the other hand, current trends like the popularity of Bluetooth earbuds, enormous increase in podcasts, the usage of speech assistants, and of course the latest rise of social media apps like Clubhouse, it seems that audio is becoming a more de-facto standard to interface with a mobile device and has matured over the years.

Combined with the fact that gathering of larger groups is still discouraged (or sometimes even forbidden) in most parts of the world, both locations based as well as audio-featured mobile applications are gaining more traction.

To make a mobile application location-aware, one should make use of GPS data for geolocation. This is more fool-proof than other discussed options, like QR-markers.

Next to the location awareness content should also be linked to certain hotspots, which have a geographic coordinate. The measurements and calculations can be easily done through the technique of geo hashing. Flutter has many packages available which can help in making these functionalities. One of the bigger benefits of using flutter as a framework means it can be deployed on various OS, instead of building platform-specific in the native language. Flutter having a relatively shallow learning curve and an increase in popularity makes it a sound choice for developing mobile applications.

Besides that, Flutter uses material design out of the box, which is proven to provide a clear and recognizable visual representation of user actions. These (material) widgets can be further customized and styled to accustom the look and feel desired by a client.

Functionalities, or requirements, to make an app easy to use is to make sure actions are familiar to a user, like using a map with markers as a graphical representation. By numbering the markers, a subtle manner of guiding the user through the audio story was used. The competition has been using a variety of methods to work out this problem, with solutions like drawn routes or setting up geofences instead of hotspots. Implementing welcome navigation in the form of turn-by-turn navigation would be beyond the scope of this project, among other reasons. This is also what we hardly see with competitors.

The usage of audio for storytelling has been a proven solution. Combined with a location it is a form of augmented reality.

Through testing of the user interface and experience throughout various stages of the production of such an app, would assure some iterations on user feedback, assuring the quality and usefulness could be boosted along with the project.

By actually experiencing the route and zooming in on the stories, some adjustments were made to the content. Functional and technical design made the requirements and choices clear, but through testing user experience could be finalized.

12. Recommendations & Discussions

The following chapter will provide some recommendations and discussions regarding this.

Some of the recommendations to be taken into account. The duration of the project got extended. This is mostly to the fact decision making is hard with a large group of stakeholders.
It is recommended to have a true product owner, being responsible instead of the workgroup, and being able to make the call.
Furthermore, combining the results from the design document with the research paper have been proven hard. This is mainly because of the different scope the project has to Saxion's demands.
In hindsight, the design choices were maybe not all profound, as the researcher was self-biased.
For example, Flutter, which is a relatively new framework. This means a lot of changes in both the framework as well in its packages were experienced. Sometimes after updating, APIs would have changed and had to be refactored in order to work. This was also experienced by following available examples online. Maybe a more mature framework like React Native would be a more profound choice.

The bias also influenced the chosen solutions, which question the validation on some of the questions asked and answers given. Especially on the navigational solutions, time constraints played a part.

The research paper could contain more own images of each of the design steps. Some of the steps may have needed extra attention, like the initial draft of the UI. And more research could be put in the story telling and contents part.

Lastly, it is hard to showcase a product which features location-based functionality, as it is best experienced on location.
An showcase video should be present with this research.

13. Bibliography

PubNub. (2021). *What is Geohashing?* Retrieved from PubNub: https://www.pubnub.com/learn/glossary/what-is-geohashing/

Sketchplanations. (2021). *The Haversine formula*. Retrieved from Sketchplanations: https://sketchplanations.com/the-haversine-formula

Haversine formula to find distance between to points on a sphere. (2021). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/

Google. (2021). *Google Maps.* Retrieved from Google Maps: https://www.google.nl/maps/dir/52.2322525,6.896751/Basisschool,+Bosuilstraat+3,+7523+BJ+Enschede/@52.232676,6.8972138,18.05z/data=!4m9!4m8!1m0!1m5!1m1!1s0x47b81408d7683e1f:0x587bc3a6316400cf!2m2!1d6.8962388!2d52.2322354!3e2

Iqbal, M. (2021). *Business of Appes.* Retrieved from Pokémon Go Revenue and Usage Statistics (2020): https://www.businessofapps.com/data/pokemon-go-statistics/

Adgate, B. (2021). *As Podcasts Continue To Grow In Popularity, Ad Dollars Follow.* Retrieved from Forbes Magazine: es/bradadgate/2021/02/11/podcasting-has-become-a-big-business/

Amburgey, Ive. (2020). *COVID-19's Impact on Podcast Listening* . Retrieved from Voxnext : https://blog.voxnest.com/coronavirus-impact-on-podcast-listening/

TechCrunch. (2021). Retrieved from Report: Social audio app Clubhouse has topped 8 million global downloads: https://techcrunch.com/2021/02/18/report-social-audio-app-clubhouse-has-topped-8-million-global-downloads

Google. (2020). *Design guidance and code*. Retrieved from Material Design: https://material.io/

Schneiderman, B. (2016). *The Eight Golden Rules of Interface Design*. Retrieved from Ben Schneiderman: https://www.cs.umd.edu/users/ben/goldenrules.html

Nielsen, J. (2020). *10 Usability Heuristics for User Interface Design* . Retrieved from Nielsen Norman Group: https://www.nngroup.com/articles/ten-usability-heuristics/

Adobe. (2019). *The 4 Golden Rules of UI Design*. Retrieved from Adobe XD: https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/

Baglione, A. N., Girard, M. M., Price, M., Clawson, J., & Shih, P. C. (n.d.). Mobile Technologies for Grief Support: Prototyping an Application to Support the Bereaved. Retrieved 1 April 2020, from http://wish.gatech.edu/wp-content/uploads/baglione_anna.pdf

Harley, J. M. (2016, January 7). Comparing virtual and location-based augmented reality mobile learning: emotions and learning outcomes. Retrieved 1 April 2020, from https://link.springer.com/article/10.1007/s11423-015-9420-7?error=cookies_not_supported&code=2fff1e17-051e-4a05-930f-722c65067370

Harley, J. M., Lajoie, S. P., Tressel, T., & Jarrell, A. (2018, September 21). Fostering positive emotions and history knowledge with location-based augmented reality and tour-guide prompts. Retrieved 1 April 2020, from https://www.sciencedirect.com/science/article/abs/pii/S095947521730405X

IEEE Conference Publication. (2017, October 1). Mobile location-based augmented reality applications for urban tourism storytelling - IEEE Conference Publication. Retrieved 1 April 2020, from https://ieeexplore.ieee.org/abstract/document/8124314/

Indans, R. (2019, March 8). Towards an Audio-Locative Mobile Application for Immersive Storytelling. Retrieved 1 April 2020, from https://link.springer.com/article/10.1007/s42489-019-00007-1

Kaiser, G. B. (2014, May 30). US10244349B1 - Location based mobile device system and application for providing artifact tours - Google Patents. Retrieved from https://patents.google.com/patent/US10244349B1/en

Kalimullah, K., & Sushmitha, D. (2017, January 1). Influence of Design Elements in Mobile Applications on User Experience of Elderly People. Retrieved 1 April 2020, from https://www.sciencedirect.com/science/article/pii/S1877050917317544

Murdock, M. (2014, October 8). US10320924B2 - Location-based mobile storytelling using beacons - Google Patents. Retrieved 1 April 2020, from https://patents.google.com/patent/US10320924B2/en

Plaza, I., Martín, L., Martin, S., & Medrano, C. (2011, November 1). Mobile applications in an aging society: Status and trends. Retrieved 1 April 2020, from https://www.sciencedirect.com/science/article/pii/S016412121100135X

Sushmitha, D., & Kalimullah, K. (2017, January 1). Influence of Design Elements in Mobile Applications on User Experience of Elderly People. Retrieved 1 April 2020, from https://www.sciencedirect.com/science/article/pii/S1877050917317544

Yang, S., & Hsu, C. (2016, August 1). A location-based services and Google maps-based information master system for tour guiding. Retrieved 1 April 2020, from https://www.sciencedirect.com/science/article/pii/S0045790615004115

Delaney, J. (2019, February 13). Flutter Realtime Geolocation Firebase. Retrieved 1 April 2020, from https://fireship.io/lessons/flutter-realtime-geolocation-firebase/

14. Table of figures

15. Appendices

Aan: Commissie 'Roombeek 2020'

Een project van:

Pronksnor
G.J. van Heekstraat 408
7521 EN ENSCHEDE

Jasper Schütz

06 49 336 997
jasper@pronksnor.nl
http://www.stadslegendes.nl

Datum: 3 december 2019

Betreft: Verhalen app voor 'Roombeek 2020'

Beste Roombeek 2020 commissie,

Hartelijk dank voor de mogelijkheid om te vertellen wat Stadslegendes kan betekenen voor 'Roombeek 2020'!

Stadslegendes is op dit moment een app waarmee heel gemakkelijk verhalen vastgelegd kunnen worden. Deze verhalen kunnen worden gecombineerd tot een tour. Op elke locatie van de tour kan een verhaal worden verteld en kan een foto of plaatje toegevoegd worden.

Voor een 'Roombeek 2020' app lijkt het wenselijk dat alleen herdenkingsverhalen over Roombeek beschikbaar zijn voor gebruikers. Hiertoe kan de Stadslegendes app omgebouwd worden om het 'Roombeek 2020' beeldmerk te dragen en deze functionaliteit goed te ondersteunen.

Ons concept voor een 'Roombeek 2020' verhalen app richt zich op het volgende:

- Eén of meerdere officiële rondleidingen door 'Roombeek 2020'
    - Verhalen worden in goede kwaliteit opgenomen
    - Bij elke locatie wordt een foto toegevoegd waarbij 'toen' en 'nu' vergeleken kan worden
    - GPS- navigatie zorgt voor accurate routebegeleiding
- Gebruikers van de app kunnen zelf tours creëren waarin zij zelf hun eigen ervaringen kunnen delen
- Moderatie van de content die gebruikers kunnen toevoegen voor de periode van het event
- Ontwikkeling van de app om het perfect passend te krijgen bij de Roombeek Herdenking

Onderstaand een prijsindicatie voor dit concept. Natuurlijk is dit heel summier en moet er nog uitgebreid gekeken worden wat er mogelijk is. Graag ga ik het gesprek aan om beter duidelijk te krijgen wat de eisen en wensen voor deze app zullen zijn.

Met vriendelijke groet,

Jasper Schütz
Stadslegendes

Functioneel ontwerp:

Roombeek 2020 App

Doel van de app:

Roombeek 2020 staat stil bij de gebeurtenissen rond de vuurwerkramp van 20 jaar geleden. De Roombeek 2020 app ondersteunt door een aantal verhalen over de ramp aan de fysieke locaties in Roombeek te koppelen. De verhalen vormen gezamenlijk een route door Roombeek.

De gebruiker:

De gebruiker kan de app openen om meer te weten te komen over de gebeurtenissen rond de vuurwerkramp. Deze informatie wordt aangeboden in een route die fysiek door de gebruiker afgelegd wordt. De gebruiker loopt dus van locatie naar locatie onder begeleiding van de app en kan de verhalen beluisteren via de speaker of headphones van de telefoon. Om een verhaal te beluisteren moet een gebruiker ook echt op de juiste plek aanwezig zijn!

De informatie:

De verhalen zijn afkomstig uit interviews met de betrokkenen. De teksten worden duidelijk voorgedragen door een spreekster of spreekster. De teksten worden aangeleverd door een derde partij, waarna ze klaar worden gemaakt voor de app. Een verhaal duurt maximaal 2 minuten om de interesse te behouden.

→ Wie levert exact de teksten aan?
→ Wie zorgt voor het inspreken van de teksten?
→ Weten de geïnterviewden dat ze in een app komen?

De route

De route leidt langs een 8-tal luisterpunten voor verhalen en een aantal fysieke locaties waar iets te doen is tijdens de herdenkingsmaand. Een ideale route duurt ongeveer een uur. Daarbij wordt 45 minuten gelopen en 15 minuten geluisterd. Als gebruikers even blijven hangen in de fysieke locaties loopt de gebruiksduur wat op. Voor nu ligt de route nog niet vast, dus de route hieronder is een indruk van de route in de app:

Design Document v1.0.1

1 December 2020
Revised on 5 April 2021

Bart-Jan Herweijer
151679, student CMGT

1. Table of Contents

2. Roombeek 2020

A design document on the creation of a mobile application to commemorate the Fireworks disaster which happened in Roombeek, Enschede on 13rd of May 2000.

Involved parties:

The following parties take part in the creation, development, and support of the application.

Client:

Roombeek workgroup, consisting of the stakeholders.

Stakeholders:

Gemeente Enschede, Chantal Euverman
Huis van Verhalen, Hadassa Meijer, Albert Haarman
Prismare, Evert van de Beld
AKI school for fine arts, Marc Bouwmeester
MuseumFabriek, Edwin Plokker
Tetem, Wilja Jurg
Vijvervrijdagen, Henri Workel

Product owner & content:

StadsLegendes, Jasper Schütz

Developer:

Brtld interactive, Bart-Jan Herweijer

3. Overview

This document is written to assist the development of the Roombeek 2020 app, which will be described more in detail throughout this document. The assignment was commissioned by the workgroup of Roombeek 2020.
The mobile application is part of the graduation process of Bart-Jan Herweijer, student CMGT at Saxion Universities, and takes place from September 2020, until April 2021. The initial lifetime of the app would have been in May 2020 but has been postponed and pushed one year later due to the Covid-19 crisis which hit the global pandemic in early 2020.

4. Context & scope

The context was initially declared along with the scope in the implementation plan. These two chapters have been added in this design document to have all relevant information in this document.

Initial scope

Now that it has been over 20 years ago, most of the neighborhood has been reconstructed. Under the guidance of Pie de Bruijn, a famous city architect and urban planner, the area has been gentrified at a certain extend. Most of the (obsolete) textile factories have been replaced by 'hip' company buildings, of which many have a cultural or creative character, sometimes even by upgrading and renovating those old factories. Many of the social houses have disappeared, making room for a higher social class.
Roombeek has been reconstructed from the ground up, focusing on a mixture of living, work and leisure. A kind of revolutionary and progressive concept in the early years of this millennium.
Since then, many tourists from all over the world have been visiting the neighborhood, where old and new meets. A unique mixture has been created of museums, like the *MuseumFabriek*, *Tetem* and *Rijksmuseum Twenthe*, cultural heritage from the old textile industries in the early stage of industrialism, and contemporary architecture.

The clients

Each year there is still a memorial service to commemorate the fireworks disaster. This takes place at the main monument, where once was the fireworks factory of SE fireworks. Each 13nd of May at 15:20 people come in and homage the fallen.
Because it will be 20 years after date, some of the local initiatives have decided to give extra attention to this anniversary. Together they have formed a workgroup, investigating, bundling and executing the different initiatives that have been coined in the neighborhood.

Roombeek 2020-workgroups

The initiatives have been collaborating in a workgroup. This consists of members from the *Gemeente Enschede*, *MuseumFabriek*, *AKI school for fine arts*, *Huis van Verhalen*, *Tetem*, *the*

creative campus CeeCee and the neighborhood association and community center. Each of them has researched ways in to contributing to those commemorations. There has been decided that the whole month of May there will be extra activities and expressions towards the memorial of the disaster.
There has been set up multiple tracks, from educational programs to cultural heritage, and one of those is an interactive tour. This will not take place with a tour guide, but people will be accompanied by an application. This interactive location based (audio) tour will be topic of this applied research.

Stadslegendes

Stadslegendes, in English best translated as 'Urban Myths' is a mobile application of the company Pronksnor. Jasper Schutz, owner of Pronksnor, has come up with the idea of creating location-based audio tours through the usage of the application Stadslegendes. The app will be very suitable for the tour in Roombeek, because it will be not depending on tour guides, thus providing people an individual experience where they can find out more of the history of Roombeek.
A first version of Stadslegendes has been made in 2019 by a number of ICT-students from Saxion. However, maintenance has been proven difficult and also number of users are 'disappointing'.
The project of Roombeek 2020 will be a good opportunity to refactor the code, make it available on iOS and Android devices, and *white label*, or rebrand it to the needs of the clients. Moreover, it may persuade the users to also use the regular app and grow its user base. This will attract more users to the platforms which will help to generate a critical mass for further development and maintenance of the platform and application, making it a viable project.

Summary

In May 2020 the neighborhood Roombeek will commemorate the fireworks disaster which took place 20 years ago. One of the projects will be an application, with which people can get a tour through the area, being served more information on some of the so-called hotspots. The objective is to present some personal stories of survivors of the disaster, to empathize with them and bond with the city. And of course, to pay respect and homage to those who did not survive. To show the resilience of the city of Enschede, and how people can find comfort after disaster.

Indicators of success

There are a number of indicators which could determine the success of the project. The fact of building a software application divides the indicators in functional & technical specifications.
The functional indicators specify manly from a user's perspective, while technical aspects are more focused around the design & develop process.

Functional

4. Users must be able to independently walk a tour, accompanied by the app
5. Users should get extra information on the fireworks disaster and its impact through personal stories

6. Stories should be available by physically visiting a number of hotspots along the route

Technical

5. Application should be up and running in the Google Play store and Apple store on May 1st.
6. Because of numerous different devices, the application should be lightweight as possible
7. Platform and code should be easy to develop & maintain
8. Should have an intuitive user interface and interactions, due to the target group is supposedly elderly people (whom in general have a bigger interest in history and city tours).

The users

The users of the application will be very broad, as basically the application can be downloaded by anyone who is interested and has a smartphone. Therefore, user interactions and interface should be very clear intuitive.
The workgroup has decided that the main target group would be adults, with an interest in history or in special towards the neighborhood Roombeek. Users will be notified of the application through diverse channels, like public and social media. The local and national newspapers will be informed of the project, which might cause for a rise in visitors and users. Besides that, also posters will be used at the partners of the project. *Huis van Verhalen,* which provides the content for the app, also hosts tours themselves. During that month they will channel visitors towards usage of the application.

The market

This chapter will focus on the market. It will briefly describe all the current trends, along with the technical advancements needed to conduct the research and, more importantly, the viability of the project. Which opportunities and threats come into play in designing and developing the application?

Current trends

There are a number of trends within the current market visible. Some relevant trends will be briefly introduced below.
LOCATION BASED APPLICATIONS: There has been a rise in the usage of location-based applications. With apps like *Snapchat* and of course *Pokémon Go* depending heavily on location-based technology and interactions, you can see that from a market-pull perspective location-based application are getting more and more common nowadays. Apps like *Uber* disrupt the taxi industry due to the real-time location updates from both the users as well as the uber drivers. This creates opportunities to app developers, when they make smart use of location-based technology.
PERSONALIZED) EXPERIENCES: As we are shifting more and more towards a 'sharing-economy' you see an increase in the need for personalized experiences. These custom-tailored experiences invoke a connection with the target group, mainly because it fulfils their needs

to be original and have a sense of belonging. Of course, the increased state of individualism takes a big part in this too.

Competitive Analysis

Due to the fact that the project has already been assigned there is a different look on the competition. Moreover, because of the 'local' market and short duration of the project it is serving a niche market.
However, if we look at competitors, we can definitely learn from their cases, making use of the best practices. This is done more in detail in chapter 7.1 .

Scope

The following chapter will narrow down the scope of the project. Of course, there are a number of restrains, requirements and edge cases in the project. This chapter describes the general outline of the research and the finished product.

Moscow-list

By making use of a Moscow-list we can set the priorities and requirements for the project.

Must Have	Should have	Could have	Won't have
Working application for IOS & Android	Location awareness (through GPS /Bluetooth / 2/3/4G triangulation)	Real time updates on current location	A lifetime support for both platforms
A map with overview and pins on hotspots	A route between the hotspots	Navigational system between the hotspots	Augmented reality capabilities
A clear visual identity	Designed elements which align with corporate identity	Splash screens and animations	A multiplatform social media campaign or advertisement
Easy and intuitive control	User interface an UX-elements tested with target audience	Walkthroughs and aiding systems for impaired & technically challenged.	Extensive support through email / phone / help section website
Options to manage the content	A manageable backend	A content management system	Extensive options for users to control content

Budgets

There is a budget available for the development and maintenance of the project. Initially, the budgets were unclear, but due to the *Stichting vuurwerkgelden*, who is in charge of money targeted for the reconstruction of Roombeek, a budget could be requested. For the whole project, a budget of 10K is requested. This is divided between development (± 5K) and project management (including overhead and unforeseen) & content creation (5K).

An overview of the available budgets and agreements can be found in the attached offer which was approved by the committee.

5.Goals & Non-Goals

The goals & non-goals lay down a first foundation for the project. The goals are loosely related to the requirements and the size and timespan of the projects. The milestones will be later derived from these goals.
A more descriptive summary of the mobile application can also be found in the chapter Proposed Solutions and more specific the functional design.

The general goals of the project will be:

- The project will run from September 2020, up till June 2021, and will be implemented by the product owner and developer
- The goal of the project is to develop a digital and interactive prototype to be used during the memorial month of the fireworks disaster, which will take place in May 2021
- Through a mobile application, visitors can take a guided tour, containing some (personal) stories which relate to the fireworks disaster

The application will meet the following traits:

- The application will run on mobile phones with Android & iOS
- The application will be location-aware
- The application will feature several 'stories' that are available through 'hotspots' along a route/tour.
- The app will be able to determine the proximity of the user to the hotspots.
- The stories featured will be firsthand experiences of people having who were closely related to the fireworks disaster, which happened on the 13th of May 2000 in Roombeek, Enschede
- The application will run during the commemoration month of May 2021, but there is an urge to embed the stories in communal history.
- The research will be done on how to make this part of the communal history, whether its availability will prevail the mentioned period, additional tours are added, or other platforms are found to add the content
- The application will be built modular, so extensions in the future can be realized.
- The content will be placed in a backend, making updates of content convenient
- At a later stage, a user could be able to log in and add content themselves.

6. MILESTONES

Throughout the project, some milestones have to be reached for it to have a good delivery. But first, we will focus on the deliverables which have to be met for graduation to be successful.
The indicators for success can be found in chapter 4.3 of this document.

Deliverables

The deliverables will be:

- An implementation plan
- A written report
  - Including an in-depth analysis of the research phase & testing phase
- A design document
- A functioning mobile application for iOS & Android
- Codebase
- Final presentation
- Promotional activities (TBD)

The milestones summed up in the document focus on the development of the application rather than the whole project (research + documenting). The milestones can be used to further specify sprints and make schedules out of this.

Milestones

Throughout the projects, there are certain difficulties and challenges to overcome, these are represented in the so-called milestones. These milestones represent (partial) functionalities, or tasks in the development process, and are arranged logically in order. They are loosely formed by a waterfall method, meaning the prior objectives should be met before continuing to a new milestone

1. Setting up a development environment

The first step in the development process is to set up a development environment. This is needed to start building the application. The Flutter framework has been chosen. It is installed by following the installation guides on https://flutter.dev. Next to the installment of the SDK (software development kit), VS Code is chosen as the IDE (integrated development environment). To be able to run on iOS devices as well, XCode tooling is installed too.

2. *Skeleton application*

After installing all the tooling, a new project can be created through flutter create. This will set up a typical 'Hello World' example. From this example on, the skeleton of the app can be created, navigational elements throughout the app can be made. After that, it can be fleshed out per page and functionality.

3. *Choice and installment of dependencies*

Flutter makes use of so-called packages, which add extra functionality to the framework. These dependencies enable working with other software, backends, or extend the widgets available.

4. *Hook up backend*

The backend will be linked after installing all the packages. For this project, Firebase has been chosen as a backend. Due to the fact it is a product from Google, as Flutter is, they work together quite well. It's also known as BAAS (backend as a server), due to the fact it offers rich features like a real-time (NoSQL) database and extensive authentication and analytics out of the box. Therefore, it is suitable for prototyping.

5. *Add maps & user location*

After adding in all the packages and have the database in place and integrated into the application, it's time to add a map view and enable user location. This will be shown on the map and is needed to calculate the proximity of the users to the hotspots.

6. *Add marker at hotspots and preliminary content*

The hotspots are added to the backend. These will be shown on the map as markers. Next to coordinates, each hotspot will also make use of geohash. This is an encoded value that will help determine the distance between a user (location) and a point (marker) on the map. A mockup of content will be placed in the database too, this will help to flesh out the app more and the ability to show and design detail pages.

7. *Range of hotspots where they can be triggered*

After adding both the map-view, user location & hotspots, user testing can begin. This milestone is focused on making the technique applicable in a proper way. The proximity of the hotspots will be tested and finetuned.

8. *Making a route of the hotspots (draw a line, navigational support)*

Ideally, when the hotspots should form a route, this is visually represented on the screen. This can be done by drawing a line between the points on the map, or number the markers. During this phase, the navigation between the different hotspots will be exanimated and a suitable solution will be chosen for both the routing as well as the navigation between the different points of interest on the map.

9. *Create and add final content*

After having the first prototype fully operational, it's time to add in the final content. There is still some discussion ongoing on whether video or audio content delivers the biggest impact and which form is most suitable for an application like this. During this phase, experimentation with different kinds of content is key. What drives the best user experience? The discussion has been written out in the functional & technical design within this document.

10. Option to auto-play & notifications

Next to testing and deciding the form of the content and navigation throughout the route, there will be research done on how to optimize the user experience (if needed) by making use of notifications or other background techniques as auto-play when in range of a hotspot. Notifications can also be used to (re-)engage users more or provide helpers to finish the current route.

11. Prepare application for placement in the stores

As all the core features are implanted and the app is working as expected, it will be time to prepare the applications for a push to both the Google Play Store as well as iOS App Store. There are strict guidelines to be met, and this phase will clean up the code and make sure it follows the design and coding conventions for it to be accepted in the stores. The application will have a final build that can be pushed to the stores and assure availability for a wide range of mobile devices.

12. Findable and downloadable in Apple & play store

The final step is to make sure all the metadata is added to become visible, findable, and downloadable from the stores. This does not include promotional activities.

Continuous Improvement

Besides having milestones to keep track of the progress and goals of the project, there will also be tasks that will maintain focus throughout the project. These tasks are called continuous improvement and are intertwined throughout the whole process. The goal of continuous integration is writing and testing more consistent code.

Testing

Throughout all the phases, testing is needed. The goals for each phase have been briefly addressed in the milestones.
Testing needs to be done both on virtual as well as physical devices. A test is passed when no compiling errors occur, and functionalities work as described.
At a later stage, more specific tests like unit, widget, and integration tests will be added through the build-in flutter tooling.
Where possible performance and optimization will be applied.
Testing of the design choices and process will be handled in a different chapter.

Extra (Non-Goals)

The extra milestones are non-goals. They are outside the scope of the initial project. So, they will not have a due date or user stories attached. However, there is a strong desire from the client *Stadslegendes* to develop the application after the deliverance of the Roombeek project. With this kept in mind, bonus points can be obtained for adding these milestones, however, it should never go at the expense of other, more prioritized features.

Options to easily add or update content

The client of the project, *Stadslegendes*, does appeal to the idea that the application can be used in other situations as well. The 'white labeling' of the project is kept in mind throughout the development but is a non-goal. But ideally, the possibility to change content over time is very sound. This can be done in the backend directly, but it would be appropriate to create a CRUD system for this. It is undecided yet whether it should be possible from the application itself, or a different matter, like a simple web app.

CMS like features

The adjustments in content can be evoked from the firebase console directly. However, a CMS-like feature as described above is eligible. This could also mean, that users can be authenticated and have a role within the system.

Login system

The first version of the app will not register users in the database. However, when the app would grow (in users or tours), it would be nice to have the users logged into the system. This could mean at a later stage they could add in content themselves, but also the progress of the tours can be stored 'in the cloud' instead of only locally on the device.

Profile pages & content creation users

For a user to track its progress, as well as the options to add content, comments, etc to the database, a user needs to be logged into the database. Therefore, a profile page to handle all user actions and updating their profile is needed.
Next to the profile page, there also need ways to upload content to the server & database and make them properly linked with all the assets and properties. Some additional research is needed on how to take and upload for instance photos, audio, or video files.

Promotional activities

The promotional activities have yet to be determined in collaboration with the stakeholders. For now, there has been reached out to *Enschede Promotie*. When closer to the launch date, also (local) press is going to be informed, hoping for more exposure.

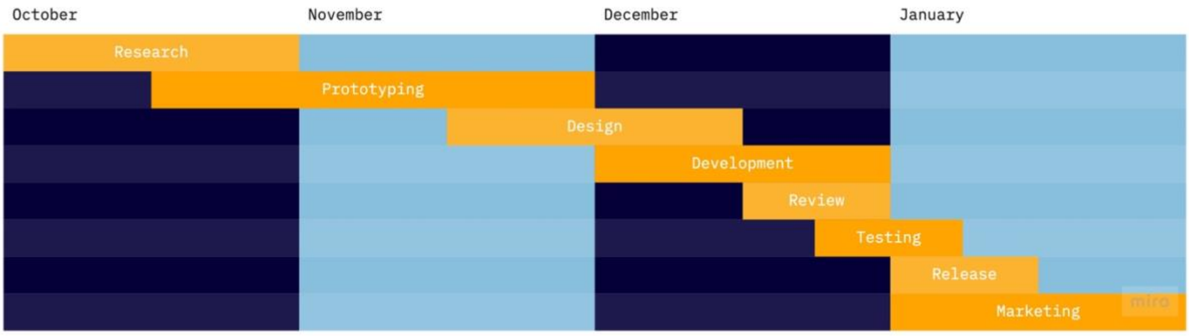
Sprints & schedules

Good time management is essential in finishing the project. All the milestones need to be achieved.
Because of the size of the project, only making a small mobile application, there will be no user stories written and used. Instead, the milestones can be used as input for the sprints. There are 12 milestones in total. Through prototyping, six of them have already been accomplished, and are ready to be implemented in the final application.

Schedules

Here is a rough schedule of the development process. It is presented in a Gantt chart. Besides the development, the project also consists of writing a report. The findings in this

Design document will form conclusions together with the 360-scan including theory. All of this will add to the design methods in the final research report.



Deadlines

For the development track, there have been several deadlines. These are related to the project and status updates back to the client, as well as in the deadlines put up by Saxion. The following deadlines should be met

1 September => start of the project

15 September => hand in the preliminary implementation plan

29 September => hand in the final implementation plan

2 October => meeting with graduation teacher

5 November => stakeholder meeting and progress update

1 December => handing in the draft design report

10 January => preview of prototype and user testing with the clients

6 April => hand in all research and documentation

20 April => final presentation

23 April => deliver the project

May 2020 => the memorial month in which the app will be mostly used, on standby

June => finished project and looking to hand over

Sprints

Sprints will be scheduled. Every two weeks, the developer will have a meeting with the product owner, Jasper Schutz, to discuss progress and set new goals. These sprints will be further determined in the retrospect and kick-off of the sprints. Because of the agile method, these sprints have not been scheduled at the beginning of the process, but instead along the way. Of course, respecting the deadlines and following loosely the milestones as described above in chapter 6.2

7. Existing solutions

Before starting a new project, it’s always good to research what's already available. This can be both through competitive analysis, as well as by looking at the feasibility of a project. With a feasibility check like this, it will be easier to make choices in the design process and justify those choices.

Competitor’s analysis

There are multiple applications in stores that provide a similar experience to what's expected of the Roombeek app.
In the following chapter, we look at some of those competitors, to see how they’ve handled functionality and display the various UI elements.
From a published mobile app, it’s often hard to see how it’s made: only the finished (compiled) product is accessible through the store. Nevertheless, some lessons can be learned by looking at different apps, trying to achieve the same goals.
For good order, I have only tested mobile apps which were:

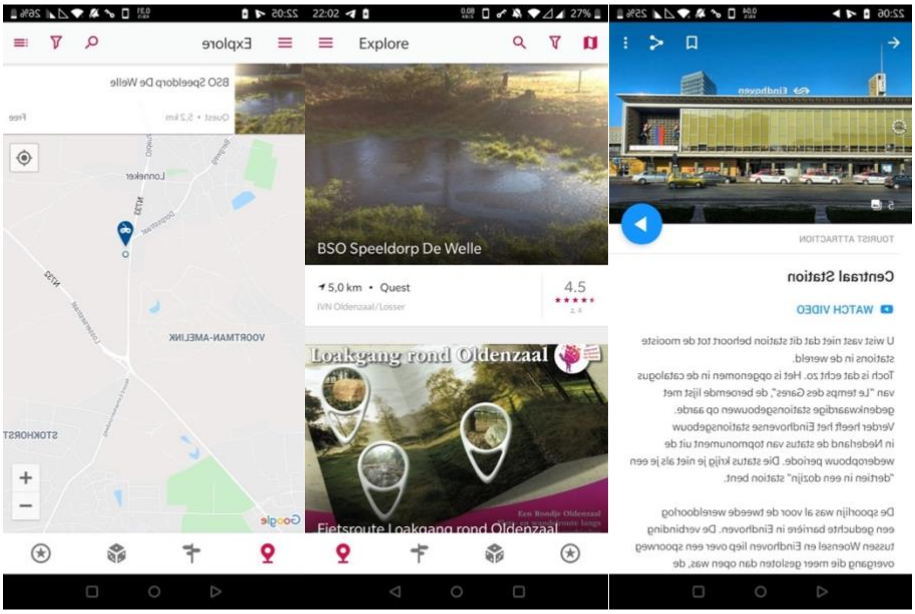
A. Available in my region
B. Feature local tours
C. Make use of Dutch content

In the final subchapter, we take a look at some tutorials and other resources which can be used in designing our solutions. But before that, a choice must be made on how to develop an app, namely in which language will be coded and which framework is used to build.

izi.Travel

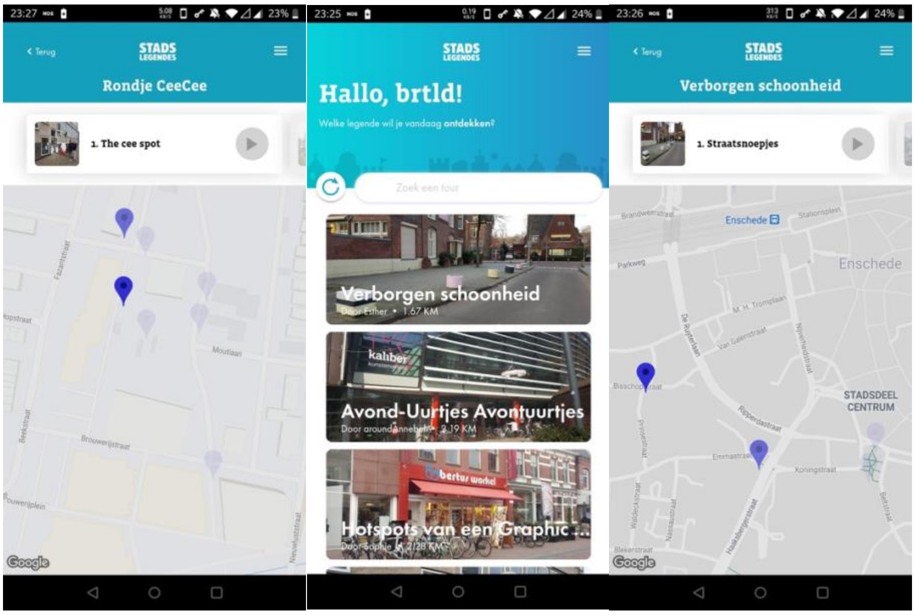
This platform has its origin in the Netherlands and has been around since 2011. According to their website, they want to become the go-to platform for the travel industry, providing tours for the cultural and tourist sector.

The app is very well thought out. Next to many different routes you can take, there is also a free walk tour, where you can visit hotspots without going through the whole route of the tour.
They have also a nice option for users to create their own audio fragments and tours and store those on the map. The downside is that is a big difference in quality between the tour because some of them have been made professionally while others are not. Navigation is not in-app either. It does draw a route between the markers, but real-time navigation is present. At the start of the tour, you can navigate through the phone’s native app like Maps. The UI of the app follows material design guidelines, and although it is very extensive it makes good use of the space and widgets.



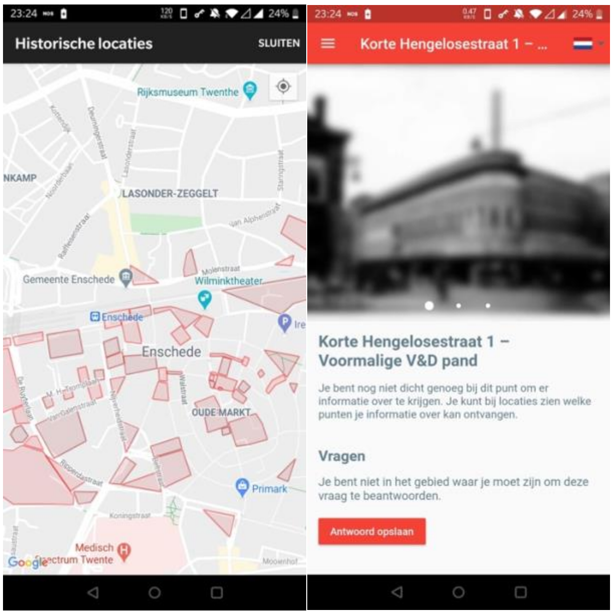
Stadslegendes

Stadslegendes was the project before this project and led to the Roombeek app.
Stadslegendes has been developed in 2018 by a group of students commissioned by Pronksnor, led by Jasper Schutz.
It also focuses on location-based audio tours, only the personal stories were crowd-sourced. Meaning is dependent on users to add content.
A prototype was made for Android and is still working today. However, the code was quite complex and hard to maintain.
This project can be a first step toward making the second version of Stadslegendes, but many features are beyond the scope of this project.



Stichting Historische Sociëteit Enschede-Lonneker

Stichting Historische Sociëteit Enschede-Lonneker has developed an app which features a city tour through the center of Enschede. In this tour, historical images are shown of some of the old historic sites. Not all buildings in the photos are still existing. The buildings are drawn on a map using a polygon. When inside that polygon, the user can answer a question, and all the hotspots form a quiz on the history of Enschede. When in proximity, the blur filter on the photos is also revealed. The app offers no form of geo-navigation and uses geofencing of the planes drawn on the map, to see if use is in the proximity of the historical location.



Korte Hengelosestraat 1 – Voormalige V&D pand

Je bent nog niet dicht genoeg bij dit punt om er informatie over te krijgen. Je kunt bij locaties zien welke punten je informatie over kan ontvangen.

Vragen

Je bent niet in het gebied waar je moet zijn om deze vraag te beantwoorden.

Uit in Enschede

On one of the websites of Enschede Promotion, the local tourist center, some walking tours, and routes can be found. In the routes are some markers added, which contain some additional information as description and photos. There are even some interactive components, featuring audio fragments. The routes make use of written navigation and can update a user's location on the map.
The routes can be found at https://www.uitinenschede.nl/routes

Frameworks

There are many ways to build apps. From the so-called 'no code' platforms to high-level libraries to low-level methods like native app development, the choice of a framework is one of the biggest factors in the outcome of the project. In general, frameworks help in doing some heavy lifting, making it easier to build the application and cut down time greatly. Most frameworks also offer additional libraries consisting of extra functionalities which can be imported.
Below you'll find a summary of some of the more popular frameworks for creating a mobile application.

React native

React is an open-source framework to develop mobile applications. It is designed and maintained by Facebook. It was released in 2015 and has gained a lot of traction, mostly due to the popularity of ReactJS, the web counterpart.
It largely resembles the same code & syntax as ReactJS, although it does not make use of the Virtual DOM, but instead has a JavaScript interpreter running in the background of the application.
React is a framework that uses JavaScript for both the logic and declaring the UI.

+ big community, lots of tutorials & packages available
- steep learning curve, with JSX-templated JavaScript & react hooks

Native application development (Java/ C++ / Objective-C/ Swift)

Over the years, (mobile) app development has been on the rise. Of course, the market has grown substantially, but also the number of tools has been drastically increased and made it easier to develop an app.
Native app development is writing code for a specific platform, with low-level code. With this in mind, you'll have to maintain at least two codebases for the two biggest OSes (Operating Systems), namely iOS & Android.
Android development is often written in Java, while iOS has been focusing on developing its own language Swift. Because developing platform-specific, the processes of the app are often tailor-made, bringing the fastest performance available.

+ fast performance, directly communicate with the internal processes of a mobile phone, without translations
- only platform-specific, so multiple codebases needed for a multi-platform app

Other (JS) frameworks (Cordova / Xamarin / Kotlin)

During the rise of mobile app development, roughly 10 years ago, also other tools have been developed, to mimic or translate (web-)technologies to the mobile market. Besides HTML / CSS /JS being encapsulated in a wrapper, also C# is commonly used in mobile development (think of Kotlin, Xamarin, or Unity3D). Cordova, formerly known as PhoneGap), was one of the first major platforms to offer such hybrid web applications for mobile devices. Since then, a lot of frameworks have delivered likewise features.

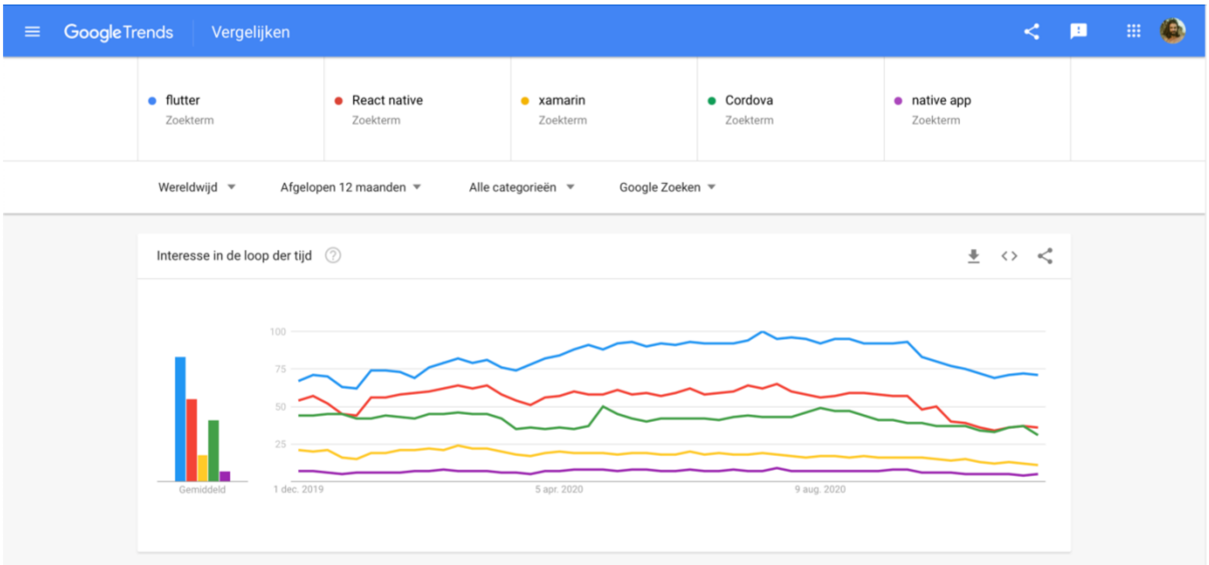
+ different options & languages available
- not as optimized as native coding

PWA

A PWA (Progressive Web App) is a web application running through the browser. Although over the years, many functionalities, like *Location, Bluetooth, WebRTC (video conferencing), WebGL, and other app-like features, like service workers (offline support) and push messages,* have been added to the specifications.
Progressive Web Apps are build using HTML, CSS & JS and have a manifest file that has metadata, like name, colors, icons, and URLs.
Although most of the modern features work on browsers, the lack of support on notifications on iOS browsers is a big loss. And besides that, location access is only available

and updated as the web app is in the foreground. These missing features will lay heavy restrictions on the design of the app, as it can only be used to navigate while the screen is on and the app is showing.

+ easy to use, as it makes use of basic HTML / CSS / JS
- not all functionalities are available of iOS



Flutter

Flutter is relatively new to the scene of app development. It was released in 2017 and is described as a 'mobile UI Framework'. Flutter compiles native code under the hood, giving it the benefits of native app development. However, a single codebase can be used, although specific Android and iOS UI elements can be implemented using Google's Material Design (standard) & Cupertino widgets (enhanced). This will make the look and feel of the app also very native. These UI elements can be styled, later on, to give it more of an own identity, without breaking with the design rules. Besides mobile app development, flutter can also be used for embedded systems and web apps (experimental).

Dart

Flutter uses Dart as its language. Dart compiles native Android code (Java or Kotlin) and native iOS code (Objective-C or Swift). It resembles other programming languages like Java, JavaScript, or TypeScript. It combines concepts both out of Object Orientated languages and Functional programming, so it can have advanced features. Classes and Objects, functions, widgets (components), and variables make up most of the concepts and tools used. It is a statically typed language like Typescript, meaning you always have to declare the type of variable. It is Async by default, with streams and futures, which can be compared to promises in JS.

Everything's a Widget!

In Flutter (and Dart) everything is approached as a widget. These widgets are placed in a tree to structure them. Even the data elements are considered to be widgets. This has the benefit of working nicely modularly. but building a tree of widgets can become quite overwhelming, as even styling, like padding or margins, is handled by a widget (encapsulating the widget which will use the styling). The settings of a widget, called attributes, can be added as a property of a widget, customizing the elements to the needs. There are many widgets available in Flutter, making the UI design easy at first, as long as you understand how to manipulate them.

Comparison & Justification

Below I made a table to compare the characteristics of the different frameworks. It has been loosely filled in according to the information researched throughout the project and personal experiences.

Framework	*React*	*native*	*hybrid*	*Flutter*
Popularity	++	+	+	+++
community	++	+	+	++
tutorials	+++	+	+	++
documentation	++	++	++	+++
performance	++	+++	+	++
Level of expertise	++	+	+++	++
Learning curve	+	+	+++	++

Packages & libraries

There are numerous packages available. Packages are extensions in the Dart environment. Where JS has NPM, has Flutter the package system. These packages are set up in the *public.YAML* file. This will install the dependencies. In the Dart files, the needed packages are imported at the top of the document. For an overview of chosen packages, to make a connection to the back end, compose map-component, or handle geolocation of the user, there is a [section in the Technical design](#).

Tutorials & resources

During the prototype phase, the goal is to find fittings solutions to the technical challenges of the app. These will include the usage of Flutter, implementing Cloud Firestore, Geo queries, and state management of the app.

The following tutorials have been proven of use to grasp the concept and help write the code:

Flutter Real-time Geolocation Firebase => an excellent tutorial on how to use Flutter and Firebase combined with geographical data and geolocation
https://fireship.io/lessons/flutter-realtime-geolocation-firebase/

Flutter File Uploads => making use of the camera and upload to Firebase Cloud Storage:
https://fireship.io/lessons/flutter-file-uploads-cloud-storage/

Flutter Provider with Firebase => how to implement simple provider for state management:

https://fireship.io/lessons/advanced-flutter-firebase/

Flutter & Firebase App Tutorial => getting acquainted with Firebase in combination with Flutter:
https://www.youtube.com/playlist?list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ

Build Mobile Apps with Flutter and Google Maps (Cloud Next '19) => an overview of what flutter can, including live coding of a geolocation-based app
https://www.youtube.com/watch?v=RpQLFAFqMlw

8. Proposed solutions

Below is the core of the design document written out. After doing initial research into possibilities, some choices have been made and justified. Below is a summary of what and how the choices are made. They have been broken down into different fields of design and experience.

Functional design

The functional design describes the functionalities the app will meet from a user's perspective. It is closely related to the User Experience design further down this document. The base of the functional design has already been shared with the client. This has been part of the agreements after the offer of the project was accepted. These (de)briefing documents, offer and functional design are written in Dutch and attached to this document.

The goal of the app

Roombeek 2020 is commemorating the events which took place during and after the fireworks disaster 20 years ago. The Roombeek 2020 app supports linking a number of stories and witnesses of the disaster by a physical location in Roombeek. The stories will create a route through Roombeek

The user

The user can open the app to get to know more about the events which took place during the fireworks disaster. This information is offered in a route that is meant to travel and visit physically. The user will be walking between locations under the guidance of the app and can listen to the stories through the speaker of its phone or headphones. For a user to unlock and listen to the story, he needs to be in proximity to that specific point of interest or hotspot.

The content

The stories will be recorded from interviews with those who were involved, whether it will be residents, emergency services, or bystanders. These personal stories will give some insight into the impact of the disaster, and how have been coping with that disaster, grief, and the resilience of the city and its people.
Jasper Schutz will be in charge of collecting the stories and content. At the moment it is unclear if only audio files will be used, or whether video files are giving a better experience. This will be tested throughout the development phase.
A story should not last over two minutes, to keep the attention of the user. Prior testing showed that a story longer than two minutes becomes less likely to be finished (with the attention of the user).
Huis van Verhalen is working on another project: Oral history, where eyewitnesses of the disaster will be interviewed by journalist Marco Krijnen. This oral history project is reporting the events and personal stories of 22 people who were affected by the disaster. But this only for the historical report and will not be published.
However, Marco can make a preselection of interesting stories and people who can tell about this.

These people can later be interviewed for a contribution to the Roombeek app.

The route

The route will lead by approximately 10 points of interest. These so-called hotspots will have stories attached to them. Besides the stories, plans have been made to showcase other memorabilia or references to the event of the disaster.
The route should take about one hour to finish. This will consist of 40-45 minutes of walking between the points of interest, and about 15-20 min of listening to the stories. Although if people tend to stick around certain points, the route could take longer.
The global outline of the route is fixed, although exceptions can be made if the contents require this.

Conditions

Besides the functional requirements described above, there are more conditions to take into account.
The app will have support until the 1st of June 2021. After that, the stakeholders will evaluate, and the project can be turned over or extended.

Product: Functional requirements

The following functional requirements need to be met for a successful launch. Besides the functional requirements, some of the bullets in the list below are more contextual or content-based.

1. Users will be able to download the mobile app through the official stores
2. The app will have a clear visual identity in line with the other involved parties like *Stichting Roombeek* & *Huis van Verhalen*.
3. The app should handle information on the disaster in an integer and non-biased or judgmental any way.
4. The stories should be appropriate for all ages.
5. The content of the stories will be in Dutch; the other UI elements could be multilingual.
6. The stories can only be unlocked in the concurrence of a point of interest.
7. After unlocking the stories, the stories could listen at any given location.
8. The stories are read by a clear voice which is pleasant to listen to and easy to follow.
9. The route will be predetermined by the client
10. The stories must be approved by the client upon launch

Technical design

How does the app work?
What are the considerations in chosen technical solutions?
What are the existing solutions?

Frontend

The front end represents all the objects and elements which are represented on the screen of the user. The frontend also couples the user interface to the database through business logic, like reading sensors, detect button clicks, or retrieving data from the backend

Flutter

The front end will be made using Flutter. All of the code will be written in Dart. The benefits of Flutter have been discussed earlier in this document. Due to the growing popularity, good documentation, and many packages that could speed up the development process and save time, it will be a good choice to provide a firm base for the application.
At a later stage, the workings and examples of code can be added to this document.

Backend

The backend is where the data is stored and structured. It is usually accessed by using API calls to the backend. These requests in the Application Programming Interface will get certain data in a model, which then can be presented by the (updates in the) frontend. In content management systems, the administrator pages for controlling the content are also often called a backend, although one can interact directly with the content or data, it still is considered a user interface, and therefore not an actual backend (as backend often refers to the server code). Firebase offers a console where you can view and alter the data(-models).

Firebase

Firebase is used as a backend since it is very easy to set up and tailored to work with modern frameworks as flutter. Next to the real-time database, it offers extensive user management and analytics out of the box.

Firestore

Within Firebase you have the option for either the 'classic' Realtime Database and the Cloud Firestore. The first one is typically used for smaller applications that heavily rely on real-time updates. Data is stored in a single JSON file and can be read a written instantaneously. Whereas Cloud Firestore can handle bigger data entries. As a NoSQL database, it relies on the usage of collections that can contain documents while both can contain data. This tree-like structure provides easier handling of large datasets also to the fact you can perform queries on the data, as supposed to the Realtime Database.

APIs

Application Programming Interfaces make it possible to connect the frontend and backend of a system. Within the application the business logic of the UI makes calls to the backend, requesting data or data manipulations. APIs can call internal services (database) as well as external services. Though an endpoint it can gather data from external APIs.
Because Firebase is a Backend as a service, all of the calls will be to external APIs.
Besides using Firebase for data storage, it can also be invoked to provide user statistics through analytics and A/B testing, cloud storage, authentication, and cloud messaging.
Which of the features will be used besides the Cloud Firestore is yet to be determined.

Google maps

To show a map component to the user, Google Maps API is implemented in the software. These API calls are used to show the map screen and place markers on the screen. Although there could have been chosen for a free and open-source API, like Map box or OpenStreetMap, there is a free tier in using Google Maps. Beneficial is the plugins available to the Google Maps API and components, like the chosen *geoflutterfire*, which combines Firestore data through queries based on geolocation.

3rd party plugins are called packages in the Flutter environment and can be found on https://pub.dev and extend the possibilities of Flutter by making it possible to import new Widgets.

Packages

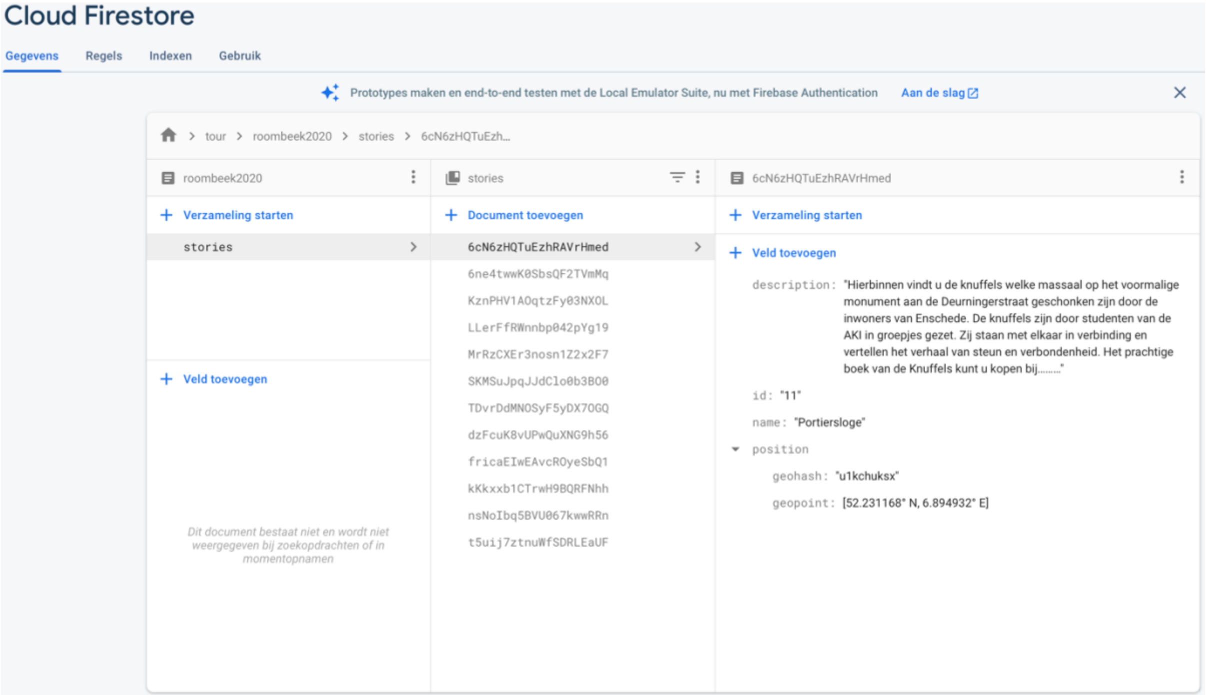
To make all the parts from the frontend, backend, and the various external systems through API calls work, Flutter turns to packages. These packages can be found on https://pub.dev and feature additional functionality which can be integrated into the applications.

The following packages will be used:

	Standard Flutter	**Firebase**	**Geolocation & maps**
Essential	flutter	firebase core	google_maps_flutter
	material design	cloud_firestore	location / geolocator
optional	Cupertino	firebase_auth firebase_analytics	geo_coding flutter_polyline_points
computing		Geoflutterfire	
model	provider		

Data structure & state management

The data will be structured in Cloud Firestore, which consists of collections & documents. Some documents sit a collection, for example, a collection(stories) consists of documents(story) that hold values like title, position, coordinates, description & a media file, or link. These attributes are all stored in the document. High in the data structures, you'll find different tours (all of them having several stories) or user entries.


Cloud Firestore
Gegevens
Regels
Indexen
Gebruik
Prototypes maken en end-to-end testen met de Local Emulator Suite, nu met Firebase Authentication
Aan de slag
tour > roombeek2020 > stories > 6cN6zHQTuEzh...
roombeek2020
stories
6cN6zHQTuEzhRAVrHmed
Verzameling starten
Document toevoegen
Verzameling starten
Veld toevoegen
Veld toevoegen
Dit document bestaat niet en wordt niet weergegeven bij zoekopdrachten of in momentopnamen

```
stories

6cN6zHQTuEzhRAVrHmed
6ne4twwK0SbsQF2TVmMq
KznPHV1AOqtzFy03NXOL
LLerFfRWnnbp042pYg19
MrRzCXEr3nosn1Z2x2F7
SKMSuJpqJJdClo0b3B00
TDvrDdMNOSyF5yDX70GQ
dzFcuK8vUPwQuXNG9h56
fricaEIwEAvcROyeSbQ1
kKkxxb1CTrwH9BQRFNhh
nsNoIbq5BVU067kwwRRn
t5uij7ztnuWfSDRLEaUF

description: "Hierbinnen vindt u de knuffels welke massaal op het voormalige
             monument aan de Deurningerstraat geschonken zijn door de
             inwoners van Enschede. De knuffels zijn door studenten van de
             AKI in groepjes gezet. Zij staan met elkaar in verbinding en
             vertellen het verhaal van steun en verbondenheid. Het prachtige
             boek van de Knuffels kunt u kopen bij........"
id: "11"
name: "Portiersloge"
▾ position
    geohash: "u1kchuksx"
    geopoint: [52.231168° N, 6.894932° E]
```

For state management, there are a lot of models you can choose from. The most popular for Flutter are the BLoC-pattern, MVVM, Inherited Widgets, or RX-variants.

Provider is the recommended package from the Flutter documentation; therefore, this solution is chosen. It is also one of the easiest state managers to use. It is a combination of the *ChangeNotifer* and *InheritedWidget* methods, and uses the following structure:

Models > hold models (blueprints) of the data model
Services > binds data (and external services to the data model (populates the models)
Widgets > can access the globally declared properties of the data model

This data model and state management will be described more in detail in the next version of this document.

Product: Technical requirements

1. The app should be lightweight: making it available for a large number of mobile devices
2. The app will not abuse the user data, if any
3. The app will be GDPR-compliant
4. The app will not make excessive use of the resources of the phone or user.
5. The app will be using the phone’s location data (antenna-triangulation or GPS) to obtain the user’s position
6. The app could feature a login system and user profiles at a later stage

User Experience design

What is the main target group and how does it make use of technology?
How will the user make use of the application?

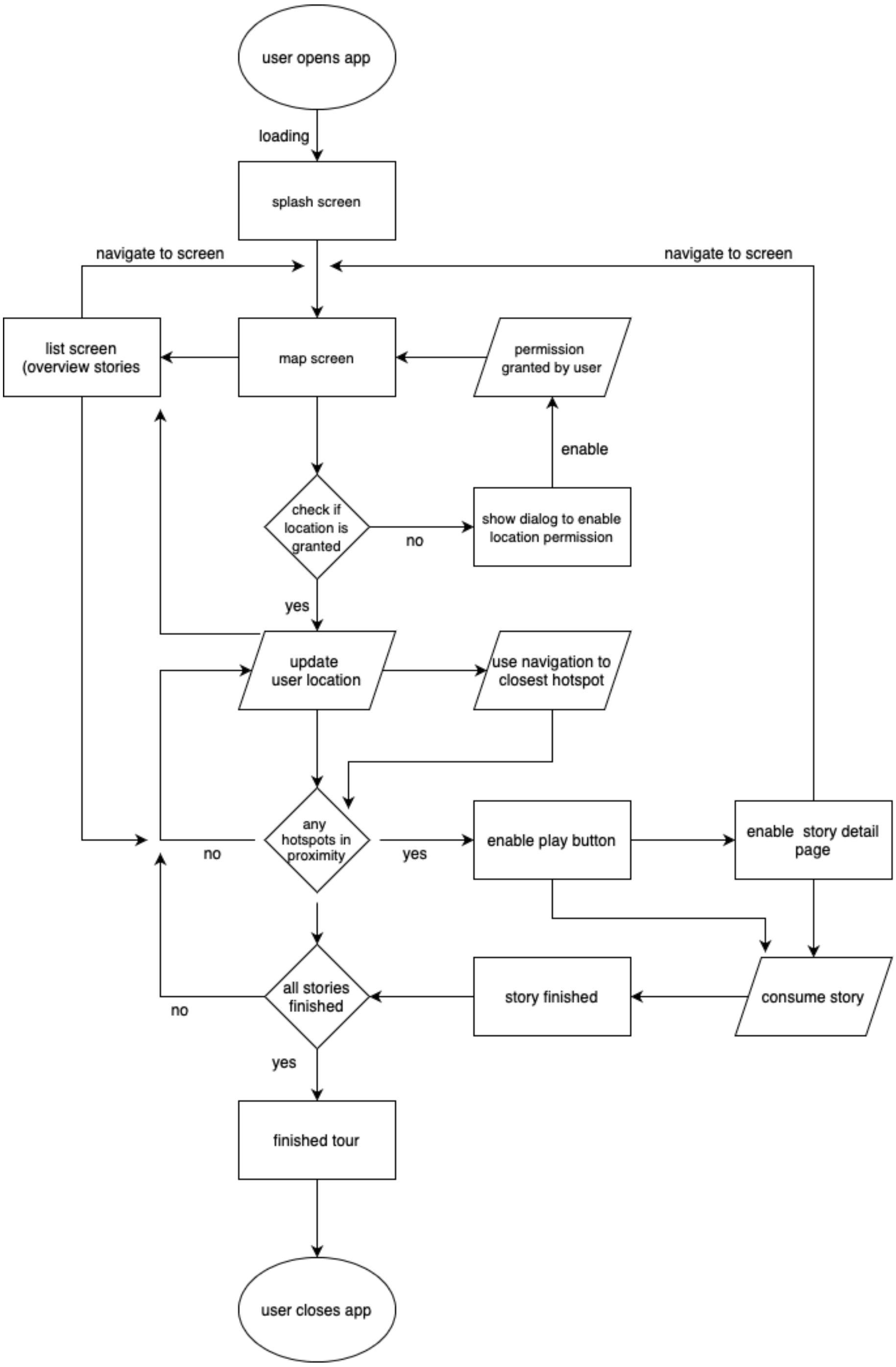
What flows are there to work out?

Product: User requirements

1. The app will be easy to use
2. The UI should be intuitive
3. The app will feature a quick how-to section or walkthrough to explain the working
4. The app will have a recognizable visual identity
5. Users should be able to use the app without signing in.
6. The user should explicitly be asked to grant permission for location tracking
7. Location tracking should only happen as the app is in use (either in the foreground or background) but should not send requests after the app has been shut down.
8. The app will have a contact (minimum of email address) for reaching out to the team in case of questions or remarks.
9. The app will have a disclaimer on safe usage throughout traffic.

Product: Flowchart

The following diagram describes a simple user flow throughout the app.



User interface design

What does the target audience expect?
What does the target audience already use on their mobile phone?
How can the actions become explicit through design?

Material design

Because Flutter uses Material Design standards, designing the UI is relatively simple. It consists of many widgets widely used in all different applications so will feel very familiar to users.
Later on, these material elements can be further designed to give them a custom look.

States of the application and components

Needs to be worked out

Geo-navigation

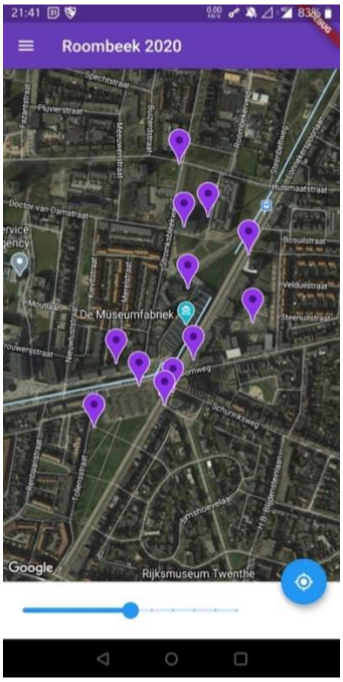
This chapter will describe the different options to navigate a route along with the stories. Geo-navigation is one of the biggest challenges in the project. There are different options gathered through a brainstorm and ordered by complexity, starting at the simplest.

1. User free walk mode instead of a predetermined route
2. Draw a line between the markers
3. Add an audio description to the next point of interest
4. Add a compass arrow and a distance to a selected point of interest
5. Implement real navigational support (like google maps)

There will be an iteration in the design process dedicated to this challenge and will be added to the report once finished.

Product: Mock-ups

Mockups will show the design of the view layer.
Need to be added in.



Graphical design

How should the application look and feel? Is there a corporate identity available? What is the message to be put across regarding visual outlets? Should it be a familiar or a project on its own?

Research on the Graphical Design.

During various meetings with the stakeholders, the design of the app has been discussed. The project team will have a free hand in making graphical design choices.

However, during the meeting in November, the workgroup did propose to use a visual style that aligns with the current initiatives and parties involved in the process. Examples of visual identity can be found at Roombeek.nl, Huis van Verhalen & the previous versions of Stadslegendes.

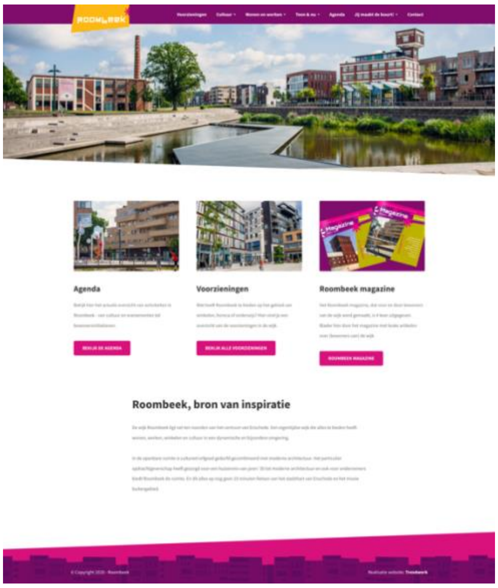
In most visual identities, there are some colors frequently used. Diverse hues of purple, pink and blue can be found. The accent colors are orange hues.

In the app Stadslegendes, the palette uses a gradient of blues.

Logo

The app will need the use of a logo in various places. At the home screen of a phone, it is used to access the app (i.e., a shortcut). Therefore, a recognizable icon should be chosen. The icon can also be used in the splash screens (loading screen) and on the app bar (the header navigation).

Screenshots of the website of two of the involved parties: on the left, www.roombeek.nl and on the right www.huisvanverhalen.nl



Voorzieningen · Cultuur · Wonen en werken · Toen & nu · Agenda · Zij maakt de buurt! · Contact

Agenda

BEKIJK DE AGENDA

Voorzieningen

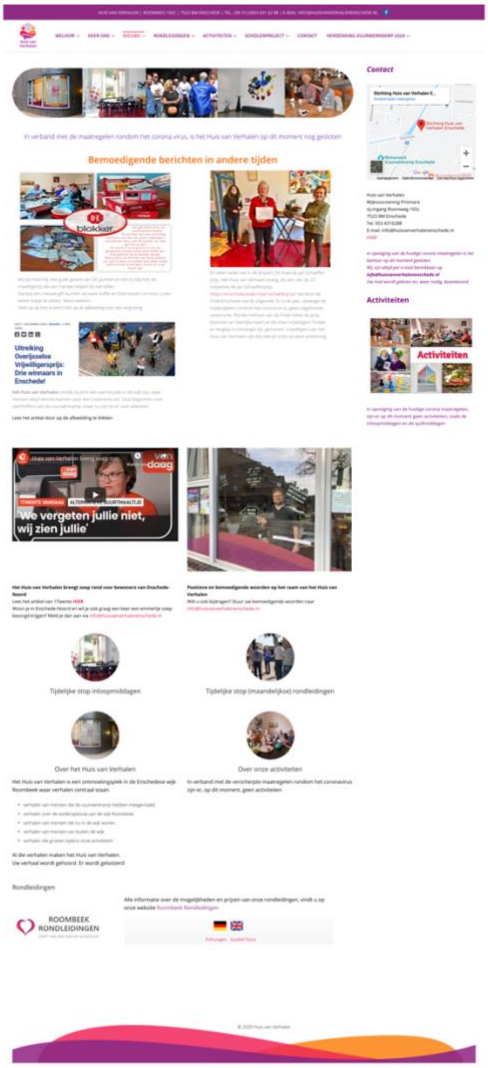
BEKIJK ALLE VOORZIENINGEN

Roombeek magazine

ROOMBEEK MAGAZINE

Roombeek, bron van inspiratie

© Copyright 2020 - Roombeek



WELKOM · OVER ONS · NIEUWS · RONDLEIDINGEN · ACTIVITEITEN · SCHOLENPROJECT · CONTACT · HERDENKING VUURWERKRAMP 2020

In verband met de maatregelen rondom het corona virus, is het Huis van Verhalen op dit moment nog gesloten

Bemoedigende berichten in andere tijden

Uitreiking Overijsselse Vrijwilligersprijs: Drie winnaars in Enschede!

Lees het artikel door op de afbeelding te klikken

Contact

Activiteiten

Het Huis van Verhalen brengt soep rond voor bewoners van Enschede-Noord
Lees het artikel van 1Twente HIER
Woon je in Enschede-Noord en wil je ook graag een keer een emmertje soep bezorgd krijgen? Meld je dan aan via info@huisvanverhalenenschede.nl

Positieve en bemoedigende woorden op het raam van het Huis van Verhalen
Wilt u ook bijdragen? Stuur uw bemoedigende woorden naar info@huisvanverhalenenschede.nl

Tijdelijke stop inloopmiddagen

Tijdelijke stop (maandelijkse) rondleidingen

Over het Huis van Verhalen

Het Huis van Verhalen is een ontmoetingsplek in de Enschedese wijk Roombeek waar verhalen centraal staan.

- verhalen van mensen die de vuurwerkramp hebben meegemaakt
- verhalen over de wederopbouw van de wijk Roombeek
- verhalen van mensen die nu in de wijk wonen
- verhalen van mensen van buiten de wijk
- verhalen die groeien tijdens onze activiteiten

Al die verhalen maken het Huis van Verhalen.
Uw verhaal wordt gehoord. Er wordt geluisterd

Over onze activiteiten

In verband met de verscherpte maatregelen rondom het coronavirus zijn er, op dit moment, geen activiteiten

Rondleidingen

Alle informatie over de mogelijkheden en prijzen van onze rondleidingen, vindt u op onze website Roombeek Rondleidingen

ROOMBEEK RONDLEIDINGEN

© 2020 Huis van Verhalen

Product: Visual Identity

The logo was requested from Indra Simmons, who is head of the Stichting Roombeek. In the past, this foundation has also published a magazine. Indra sent over an InDesign file containing some of the graphical design and visual elements.

The logo is an asterisk shaped symbol. It represents the crossing of the main roads in Roombeek: the Roomweg, Lonnekerspoorlaan and Deurningerstraat and Lasondersingel. In my personal opinion, it also slightly resembles fireworks in the sky, but I do not know whether this is intentionally.

9.Testing

The following chapter will be dedicated to testing methods. In software development, testing is very important. It assures the functionalities of the software will work on a variety of devices and within user expectations.

The technical tests, like error-free code, compliant to the publishing platforms like Google Play & IOS app store, are already briefly described in the continuous testing methods in chapter 6.2.1.

However, testing is also part of the design process and helps to determine the way to tackle problems and optimize solutions in a profound matter, which are both technically feasible as well as contributing to the goals of the app and user.

Below you'll find a summary of different testing methods before a final choice will be made regarding two of the choices to be made for main features, namely:

1. What type of content suits the application and its goals best?
2. What options are there for navigational support on a map screen between the story hotspots?

These two design questions will be handled in depth. But of course, at earlier stage feedback has been received from a target group and the stakeholder presentations.

Different testing methods reviewed

During software and app development, there are multiple occasions where testing can be useful, throughout the process, and help develop useful solutions to the proposed problems which the software application will address. These testing methods have been selected to explain and focus on. Some of them will be handled in-depth, while others will be targeted more loosely. First will be an overview of general testing methods, after that a detailed description of testing methods and expected results.

Lastly, there will be room to evaluate the chosen methods and descriptions of how those tests have provided input in the design and implementation of the product. This feedback loop takes an important part in the design thinking process, as it allows for fast(er) iterations and a more profound and usable outcome.

The software tests should ideally perform on a real device instead of an emulator or virtual device. Time will tell if this is feasible to practice on a large scale, due to the ongoing Covid-19 pandemic.

The tests should be performed on multiple real devices with different specifications, but the minimum will be to test on Android and Apple mobile devices.

In general, the following test methods are common to test software:

- User testing → is there a need to solve the problem your software is targeting
- Functional testing → test if everything is working according to specifications
- Usability testing → test to see if the users can perform tasks in the app (UI /UX)
- User acceptance testing → Beta testing
- Stress test → can you break it?

User testing

This is usually the first step in the design process. The application should address some problems or needs the users will have. It is informative from testing the ideas behind the application. Is there a need or are the users 'experiencing a pain' that can be solved by the proposed solution?
It is all about coining the idea of the application to a broad spectrum of users.
It does belong in the empathize phase of the design thinking methods.

Usability testing

With usability testing, the focus lies on how users are using the application. Are the user flows, interactions and user experiences formed logically and expected as the user performs tasks within the application? This can be done throughout the whole design process, usually starting with brown paper sessions, storyboards, paper prototypes, and mock-ups. Testers get to perform some tasks, often written down in scenarios. The key is to observe user behavior and look for where to optimize the user flow.

Brown paper sessions

The brown paper sessions are the initial brainstorms to map out functionalities. This is often done with the clients: the product owner & stakeholders.
It often consists of writing down the problem which is trying to be solved by the application. Next to this, also possible solutions come into play, as well as additional features. Everything will be written down on sticky notes and ordered logically.
Often you see after the brown paper sessions some personas are written, who all have a specific need to fulfill. Later on, in the other usability tests, these personas and their objectives can provide the scenario of the task you want a user to perform.

Wireframes /Paper prototype / mockups

The initial design phase of the application: Requirements get translated into functions and UI. This is the early stage of user experience and user interface design.
These first designs can be tested swiftly to make sure the logic is at the right place, and the consecutive actions follow an intuitive user flow.
Often wireframes start by sketching out the different screens with UI elements. Wireframes often test the placement of the components on the screen, whereas paper prototypes take this one step further and combine these paper sketches with user interactions (simulating

what happens when a user presses a button for instance. Digitalized versions are often called mockups: where the prior experience of the wireframes and paper prototypes get combined with the graphical style, resulting in a mockup of the Graphical User Interface, or GUI.
Nowadays, these three principles often get combined in computer programs like Adobe's XD, Sketch, or Figma, creating digital mockups with usable and testable interactions, by linking screens together with actions like button presses.

Moderated usability testing

Let users perform predefined actions (user stories) within the app in a controlled environment. This demo should only focus on the internal team. Actual testing with other groups should be done in more 'real-life scenarios' with actual devices (on location). Moderated usability testing is however convenient in an early stage of development because the auditor gets a lot of feedback by observing the testers.

User acceptance testing

Is the delivered software according to specs and does it live up to expectations and requirements? This so-called beta testing should ideally be done by a diverse target group, consisting of stakeholders (does the app meet up to the expectations and requirements?), experts (does the application follow the logic in UI/UX?), and the target audience (is the user able to achieve the goals in the app?).
This is scheduled for the end of March when the basic functionality is all ready for review. Two stakeholders will test the app, along with some professionals in IT and application development.
During testing either the users can be observed or be questioned in-depth afterward.

Stress testing

These types of tests will try to abuse the computer program of application. What happens if a user derives from expected flow and user interactions? The designer has an idea of how the user will be making use of the application, but during stress testing, it is checked what happens if this is not used as intended. Stress testing intends to see if the application can be broken or show unintended behavior by misusing it.

Functional testing

Functional testing is a more technical test method, focusing on quality assurance (QA), and goes deeper into what the application is doing. It tests if the components are working individually and together to perform the tests described in the functional specifications.

Analytics

As part of the firebase tools, Google Analytics can be implemented easily. Analytics show real-time statistics of the users in the application. Not only the number of users can be tracked, but also their time using the application, and the number of reads and writes of the database along with other events in the app. Because it is tightly coupled with the other tools of the Firebase backend services, also the notifications can be tracked, or users can be segmented when they make a profile.

Due to privacy concerns and the design choice not to have user profiles, these features will not be implemented in this research. Users will be tracked anonymously as 'guests', making only the active number of users available and the response to notifications (if any).

A/B testing

After the application becomes available in the app stores, it is possible to experiment through A/B testing. This is usually done by implementing subtle changes on two versions (hence the name), to see which solution or interface element is most efficient. It helps to optimize the app experience for users. This might be done after the project has gone live, but due to the time constraint, this will be not implemented before the soft launch.
On the other hand, throughout earlier stages of the development process, we will be using a different version of certain screens alongside each other, to see how the user responds. This will assure an increase in the usability of the application for the target group.
An example may be the different versions build of the map screen. Starting with a basic version with only pins on the map, up till a more advanced version where different types of route navigation will be tested. This process is described more in-depth in chapter 9.2.3

Chosen tests and methods

In this document, we will zoom in on two different tests, where has been discussion on the exact feature or contents of the app. The test will contain different methods and setups. These will be loosely placed among the different categories of testing and suited methods will be looked into. Next to desk research (literature and expert interviews/reviews), a field test will be conducted.
This means in the app multiple screens or content types will be used, and the effects of those will be observed and questioned after the test. And an initial opinionated swot-analysis will be done by the graduate student.

Usability – Mockups, Wireframes & UI-elements

After doing initial research through competitive analysis and earlier ideation phase, in which the features were roughly sketched out, on the 26th of November a presentation was held with early results.
Besides stating the problem and proposed solutions, also some mock-ups and screenshots were made. Some belonging to the prototype, other from the internet or comparable applications. This presentation helped fuel an open discussion on usability and the choice of UI elements.
This presentation was given during Feedback Friday, an event of the cee spot. Six professionals attended, who are working in creative industries, varying from marketeers, web developers, graphic designers, and business owners. The session was very open and informal, making use of peer feedback among the members of the creative community.

Outcomes and conclusion

Outcomes of the session were in general: Look at how the commonly used apps like (Google) Maps do their UI. Not only is this familiar to the users, but a lot of research has been conducted by those big tech parties to optimize usability.

Goals:

- Make an intuitive and functional application, by applying good UX
- Make a clear UI, built upon material design principles that have good usability

Used methodology:

- User testing
- Usability testing
- UX testing

Used methods:

- Open (expert) discussion
- Mildly moderated
- Mockups

Design phase:

- Inspiration
- Ideation

Attachments:

- Presentation

Content test – Video vs Audio

During the various stakeholder meetings, there have been requests on whether a video was possible to use instead of audio. The videos that were already shot by Marco Krijnsen in the assignment of de MuseumFabriek and Huis van Verhalen could be used, or even better, the videos could be recorded at the same location at the hotspot. This would cause a feeling of recognition with the users and could help to make a more immersive experience.
The development team agreed upon making a prototype: making a single detail screen containing an audio story and a detail screen containing a video.
The graduate student makes an initial SWOT analysis to map out the pros and cons of the different options. These have been mentioned in the presentation to the stakeholders.
The most important conclusions should be incorporated in the final report

Goals:

- Decide on the exact contents and specifications of the application
- Investigate which medium brings across the message the best and look into levels of immersion (without being Invasive).
- Take into account the emotional load of the options. What will be most suited from an emotional perspective

Used methodology:

- User testing
- UI / UX testing
- Functional testing

Used methods:

- SWOT analysis (biased)
- prototype

Design phase:

- Ideation
- prototyping

Attachments:

- stakeholder presentation
- SWOT analysis
- Screenshots of implementations

Functional Test of feature – Navigational possibilities.

Initial ideas where the app would take users to follow a predefined path. This route would also take place along and inside cultural institutes like Prismare and de MuseumFabriek. However, this might be hard as it still will be unclear whether these buildings are allowed to be open and host visitors. And ideally, the route could also be experienced outside opening hours.

During the ideation phase, 5 concrete proposals were described for using navigational elements to complete the route of hotspots.

1. Draw a route on the map between the markers
2. Number the markers
3. Turn by turn navigation
4. Compass arrow for direction and distance to the selected hotspot.
5. Guide the user to a hotspot through external navigation app like (Apple / Google) Maps

With the competitive analysis, it became clear none of the apps would use turn-by-turn navigation, and also by researching the different packages no extension was found to make this alongside the Google Maps plugin. An alternative, *Mapbox*, does provide this turn-by-turn navigation but is harder to use in conjunction with *Geoflutterfire* and other used libraries.

Outcomes and conclusion

Due to earlier design choices, implementing turn-by-turn navigation will be hard to realize as no libraries have been found which would add these functionalities in a relatively easy way.

The idea of having a fixed route should be not followed that strictly (nudging users to a preferred route and consecutively waypoint is not desired, especially with the buildings like Prismare and MuseumFabriek being still closed for the public).

On the other hand, it would be logical if the hotspot at Huis van Verhalen would come as the last one, which will make it easier for a user to discuss their experiences and possible bereavement.

Ideally, a combination of the different navigational elements should be used (assumption?

Goals:

- Research the different methods to make use of navigational elements within a map screen in an application
- Decide which methods are most feasible (from a development perspective) and profound for a user to complete the tour.

Used methodology:

- Brainstorm
- Desk research
- Usability testing
- A / B testing

Used methods:

- Prototyping
- Stakeholder interview

Design phase:

- prototyping
- testing

Attachments:

- screenshots
- SWOT analysis
- Source code