Elsa Lassfolk

User Experience App Design for Visually Impaired Elderly

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

27 Feb 2023

Abstract

Author:	Elsa Lassfolk
Title:	User experience app design for visually impaired elderly
Number of Pages:	53 pages + 4 appendices
Date:	27 Feb 2023
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Mobile Solutions
Supervisor:	Ulla Sederlöf, Senior Lecturer

.

The purpose of the study was to provide a feasible solution to the smartphone app design process that can improve usability for elderly users while using the application.

This study investigated the user experience and user interface theory and the existing problems that stopped elderly from using apps. At the end of investigation, a project app that fitted the needs of desire of the target user group was built.

Before building the application, market research was conducted to find out what kind of apps visually impaired elderly users were using. User interviews were carried out with the target user group. After that, user story and persona were made. Affinity diagram was drawn for brainstorming and ideation. Ideas were turned into a paper app prototype, a wireframe app prototype and app flow diagram. Theory on Jetpack Compose Accessibility and the common principles on accessibility were examined and put into practice in the app implementation. Voice user interface flow dialogue was included in the app design. User testing of competitive apps were conducted on TalkBack, Google Voice Access, Google Live View, Google Map and the app itself. Finally, based on the findings and results, the app was amended.

As a result of this project, an application was produced. With accessibility taken into careful consideration, the user interface was ready for visually impaired elderly users. A better voice user interface flow dialogue and visual interface were created to better meet users' needs.

Keywords: UX, UI, usability, elderly, accessibility, prototype, affinity diagram, user persona, Jetpack Compose, wireframe, user testing

Contents

List of Abbreviations

UX: User Experience

UI: User Interface

GUI: Graphical user interfaces

VUI: Voice user interfaces

WCAG: Web Content Accessibility Guidelines

ADA: Americans with Disabilities Act

W3C: World Wide Web Consortium

1 Introduction

Advanced medical technology has prolonged the life expectancy of humanity. In 2019, 703 million people were above 65 years old. In 2050, the number will be doubled to 1.5 billion, which is 16% of world population [1]. Thanks to the development of information technology and networks, daily needs are fulfilled by using smartphones. Finnish smartphone penetration rate in 2019 was 92% [2]. Smartphones play a crucial role in daily lives of both elderly and youngsters. Youngsters learn how to use a smartphone at an early age. As for elderly, it is not the case.

Very often, elderly users are not used to smartphone technology. The smartphone interface is overly complicated, and its target group is mainly youngsters [3]. Elderly users' needs and requirements are often left out in the design phase of smartphone applications [4; 5; 6]. It is hard for elderly to use the smartphone apps and navigate through them even if they want to.

Senior users may have impairments and activity limitations because disability in older age is common. Impairments and limitations have led to their requirements being different in smartphone apps. Visually impaired elderly requires an interface with larger texts or more contrast of colour. Totally blind elderly may need voice assistance and voice control. Designers often neglect the needs and requirements of elderly users. Therefore, elderly often find it hard to interact with the apps and sometimes find themselves totally lost.

This study is divided into four parts. Firstly, the user experience and user interface theory is examined. Secondly, the existing problems that stop elderly from using apps are looked into and market research will be conducted to find out what kind of apps visually impaired elderly users are using. User interviews will be carried out with the target user group. After that, the user story and persona will be made. Affinity diagram for brainstorming ideas to the apps will be drawn. In the third part, an app will be made based on the findings. Before making the apps, ideas will be turned into a paper app prototype, a wireframe app prototype and app flow diagram. Theory on Jetpack Compose Accessibility and the common principles

on accessibility will be inspected and put into practice in the app implementation. Voice user interface flow dialogue will be included in the app design. In the end, user testing will be conducted with competitor apps such as TalkBack, Voice Assistance, Google Live View and Google Map. The app will be tested by users and revised based on the findings and the final product will hopefully meet the needs of visually impaired elderly users.

2 Project planning and specification

When I first learned about accessibility, it was an eye-opening experience for me. The more I learnt, the more I understood there was a way to make mobile applications accessible to people with disabilities. As I have an acquaintance, who had been visually impaired for 30 years and had been scared of using smartphones and mobile applications. I thought it would be a perfect opportunity to create technology that people with similar disabilities can use. Hence, ideation of this project began.

Before app implementation, a more thorough understanding of the concept behind UX, UI and accessibility were looked into. Then user interviews, user study and market research were conducted to learn more about the needs and pain of the target user group. Persona was drawn, to see the commonalities of the target users, to be empathetic towards them and to make product design decisions based on the findings. After that, storyboard and wireframe were drawn to visualise how the app should be, how the users will use the app and what are the screens needed in the app.

The app was built with Android Studio and Jetpack Compose. It was an app targeting Android phone users. Since Jetpack Compose is a toolkit for building native UIs on Android [7] and it is built around Kotlin, Kotlin will be used as the main programming language in the app. For easier version control, the source codes have been uploaded to Github: https://github.com/ElsaLassfolk/FindYourWay.

Prototype of the app was then shared with the target user for user testing. Changes were made accordingly to meet the needs of the user. However, final product of the app was not published in the Android app store. As the app only has user interfaces, the logic for navigation and VUI (voice user interface) still needs to be implemented. In addition, user testing for a larger target group and adjustments should be done before it could be published publicly.

3 Theoretical Background

First of all, user experience (UX) and user interface (UI) are design principles commonly used by designers in website and app designs. They often go hand-in-hand in the design process. They can improve the usability and increase the satisfaction of users towards the app. Before implementing these principles, designers should first know what they are and what should be done beforehand. As they are the backbone of the project app, theories behind UX and UI will be discussed below, before we move on to the app design and implementation.

3.1 User Experience (UX)

User experience (UX) design is about how well a website or app works and how users feel about it. Very often users have a two-sided relationship about a product. They either hate it or love it. Especially when the product does not work well, users will stop using it. With research and tests throughout the design process, UX designers can find out what are the needs and requirements of the users, what they value and what pains them. With the needs and requirements being met by designers, the end product will be usable, valuable and help users [8].

Another UX definition is from Forrester (2009, p.2): “Users’ perceptions of the usefulness, usability, and desirability of a Web application based upon the sum of all their direct and indirect interactions with it” [9]. It is the relationship between the product and users. In the design process, it is impossible for designers to guess how the users would interact with the product itself. Only through user testing and interviews, designers will be able to find out. Therefore, after the project app was made, user interviews and testing were carried out to see how users interact with the app and how they felt about it.

User experience is widely used in the design world and many experts agree it is a common law that should be adopted by every designer. It is a human-first approach to web design; emphasis is on users’ needs and how they feel while

using the product. Hence, user experience is focused on usability and empathy, which will be discussed below.

Usability

Usability is one of the core values in UX design. User experience designers ensure the design has intuitive navigation, onboarding UX and a logical structure to allow the user to interact and find their way in the app without obstacles. The flow should be smooth, easy and not frustrating.[10] Since every user is different, some of them have physical or mental disabilities, some are older, some lack technical knowledge and have no idea how to use technology. UX designers should ensure that the app is not just targeting a certain target sex/age/group. It should be for everyone.

Empathy

Designers should be empathetic towards their users; see an application through the user's eyes and experience the things they do.[11] To foster and build empathy, mirroring and active listening are the keys. Designers should gather information about users' hopes, concerns and fears, so that they can build a bond with the users and to realise what they need - not what designers think the person should need.[12] Designers should actively listen to users because it is the only way to understand what he/she is hearing without any judgement [12]. After all, we cannot empathise with those we do not hear. Listening with open-heart and actively will not only understand the user needs, but also their constraints and context of the situations.

3.2 User Experience Design Thinking Process

UX design thinking process is a user-focused design that involves users from the beginning until the end. It is divided into different stages: discover, define, ideate, prototype and test:

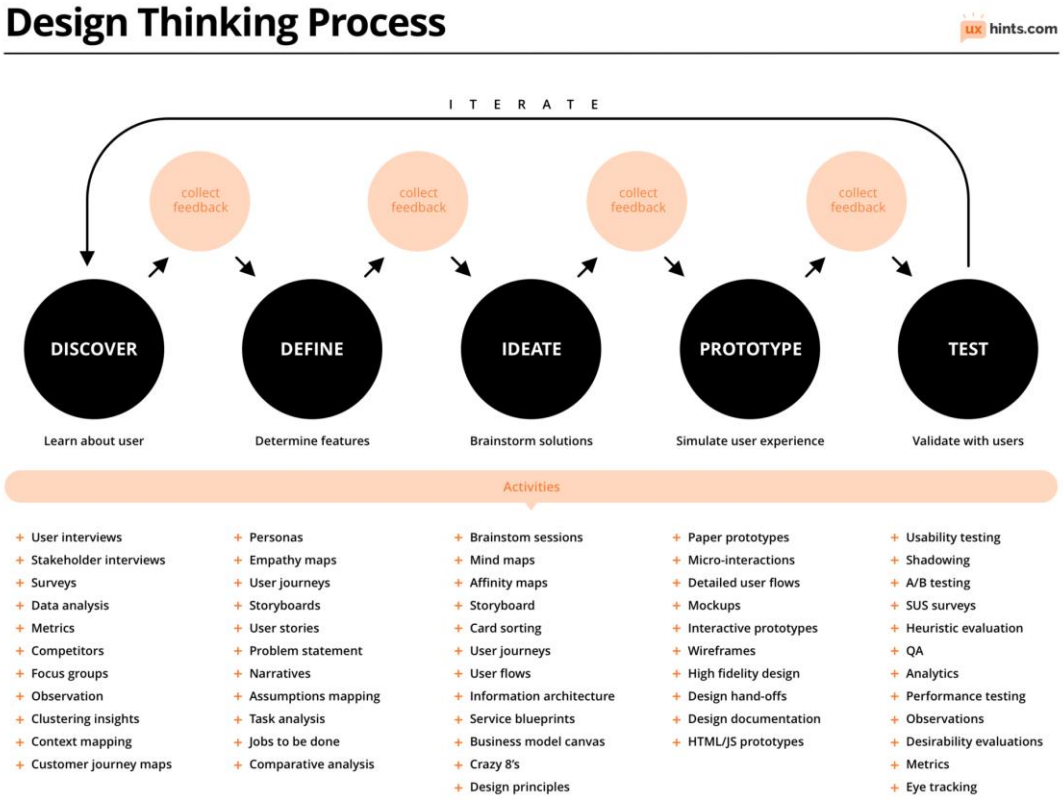


Figure 1. UX Design Process [13].

Figure 1 explains the design thinking process that UX designers need to follow.

In figure 1, whenever the previous design stage is finished, feedback is collected and used as input in the next stage. Feedback collection is a very essential element in the process. It is important for the designer to listen and keep an open mind to all the opinions. In every stage, there are different activities designers should carry out.

Discover

In the design thinking process, the first stage is the discovery stage. Designers should look into competitor analysis, market research, focus groups, stakeholder interviews, etc. As it is important to find out what are the trends and unique traits that can be implemented in the products. Designers can understand the behaviour pattern of the users, the target group, market size, similar trends and

create a better user experience in the end. [14] Through user surveys and interviews, designers can also find out what is the target group, what are their needs and pains. [14]

Define

After designers have heard about their target users, they can use users' characteristics to build a user persona. [8]

Figure 2. User persona made by UX designer [8].

Figure 2 is an example of a user persona often used by a UX designer. User persona is a user profile that represents the personality and characteristics of the target user group, which should contain all the details of the user such as demographic, needs, wants, values, fears, tech profile and quotes that summarise how they feel and sound like. [8]

After a persona is created, designers can create a storyboard. A storyboard tells a story through panels of images that map the story's main events chronologically. It should contain the visuals, captions and scenario. [15] Figure

3 below is an example of a storyboard. It has panels of images that show the story, its main events and the sequence that those events happen.

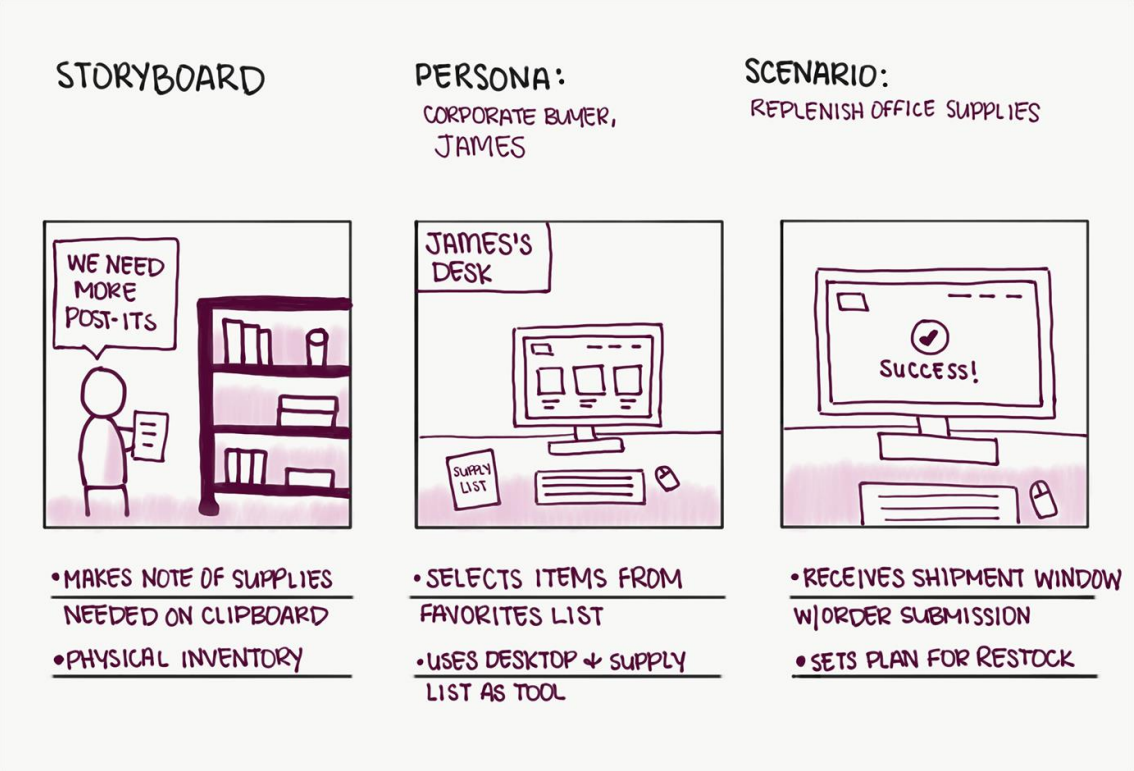


Figure 3. Storyboard through main event visuals in chronological order [15].

Ideate

After defining the target user, designers can brainstorm ideas for the possible solutions. Storyboard can be a means to generate ideas. Using a storyboard to sketch how a user uses a feature, it can provide a better picture of the user's environment and visualise the potential experience [15].

Prototype

Paper prototype is an excellent tool in the first step of information architecture. First of all, to make a paper prototype, designers lay out each individual screen of an app or website on a piece of paper. Then, each screen is linked with flows so that users can navigate from one screen to another without difficulty. [16] Lastly, the paper prototype is shared with the user for user feedback. Wireframes and high-fidelity prototypes can be produced after user feedback.

Usability testing

Throughout the process of app design, UX designers work closely with users and put them to usability testing. It is the practice of testing how easy a design is to use with a group of representative users. The main objectives of usability testing are to find out:

- whether users can complete the task independently;
- assess the mental state and performance of the users;
- check how much the users like using it;
- identify problems;
- find solutions [17].

In usability testing, moderators are often present. They should be empathetic towards users and listen to them without any judgement. Changes are made in the design after the test results are analysed.

3.3 User Interface (UI)

User Interface (UI) is the process designers use to build interfaces, which is mainly focused on looks or style. Designers control what visitors can see on a website to the smallest details and create interfaces that serve the app's purpose, easy to use and pleasing. [18] Besides the look of the interface, designers also make sure the interaction is engaging and effortlessly flowing from one screen to another in the app. A good user interface promotes effective interaction between the user and the program through contrasting visuals, clean design and responsiveness [19]. UI designer ensures the design includes effective visuals and efficient response. Not only the looks or style are present, usability should also be focused on.

Formats of UI

When users interact with design, user interfaces are the point of contact. User interfaces come in 3 formats:

- Graphical user interfaces (GUIs) - Users interact with visual elements on digital control panels;
- Voice user interfaces (VUIs) - Users interact through their voices. E.g. Siri on iPhone;
- Gesture-based interfaces - Users interact with the program through bodily motions: E.g., in virtual reality games [18].

In this study, GUIs and VUIs are focused primarily.

3.4 Difference between UX and UI

UX and UI are important in digital design. They are both human-centred designs, concerning how users are satisfied with the product. However, there are small differences between them.

Table 1. Comparison between UX and UI [8].

UX	UI
Human-first approach.	Visual-focusing approach.
Focus on how users feel.	Focus on how users see.
Focus on whether it can satisfy the needs of users.	Focus on whether the design is interactive for users,whether the flow is natural.
Rely on user research, focus group, interviews and user testing.	Rely on design research and trends, industry analysis and web design principles.

UX design is a human-first approach to web design. It focuses on how the user feels and how designers should meet users' needs. The design product should be valuable and useful to users. UI design is a visual-focusing approach, the focus is more on the style and design, such as colour, buttons, fonts, menus and

animations. Users should find the design easy to use, interactive and confident while using.

A good UX app often includes appealing UI design with clean and responsive elements. UX and UI together can maximise user experience, meet their goals and produce an app that is pleasing, engaging and easy to use. In our study, we will combine these two disciplines together.

3.5 Accessibility and Usability

According to Gilbert, there are 57 million people with disabilities in the United States. However, 90% of the websites or apps lock them out, when they are trying to experience the convenience of mobile banking, connectedness of social media and online shopping. [20] The reason behind is that apps are often designed for normal people without any disabilities and disabled users are neglected in the design process. To make the app accessible for everyone, it is essential to keep accessibility in mind during the design process.

Users may have different disabilities and limitations. It could be mobility impairment, visual impairment or cognitive impairment. Designers should carry out user research during design and development to understand who the users are and what kind of disabilities they may have.

There are several guidelines and tools that designers can use to design for accessibility. According to Braithwait, there are four ways to build more accessible websites:

- Web Content Accessibility Guidelines (WCAG);
- Measure usability;
- Advanced Technology;
- Design principles [20].

WCAG provides clear guidelines for companies to make the web content accessible for everyone. Contributed by individuals and corporations in many

countries, these guides cover every aspect - from text and image use to code and mark-ups.

In Finland, the act on the Provision of Digital Services also requires government authorities, institutions and associations governed by public law to make their online service compliant with accessibility requirements. The service and its contents should meet accessibility requirements, the accessibility of the service and its content must be evaluated. The service should also include an electronic feedback channel so that users can provide feedback in regards to the accessibility of the service. Authorities such as Kela (Social Insurance Institution of Finland) have to comply with this act. [21]

Guidelines and tools ensure a broad spectrum of users are involved in the development stage. User testings are conducted to evaluate the following five areas:

- Learnability - How easy it is for users to learn how to use the website;
- Efficiency - How well users adapt to use the website to complete a task;
- Memorability - How easy can users remember how to navigate the website to re-do a task;
- Errors - The number of errors occurred when a user engages with a website;
- Satisfaction - How appealing the website is for users [22].

User testing is essential in the design process for accessibility. By exposing users to the interface and interactions early in the design process, it can make the product accessible to everyone and minimise the number of changes after it has been released in the market.

Advanced technology such as artificial intelligence (AI) and voice user interface (VUI) can be used in the design for accessibility. For example, a website is scanned and checked for the colour ratio of text used or alt text for images with AI [22]. Users who have motor skills disabilities and visually impaired users who cannot read the screen can both use their voice for data input.

Design principles raise accessibility by making the visual elements accessible to everyone:

- Font size - at least 16px to avoid readability issues;
- Contrast - focus on creating distinct differences between two elements. The focus can be size or colour. e.g., black text on light background;
- Discoverability - use contrast to help users to find information easily;
- Repetition - repeat a single element multiple times to help users remember and guide them through the website;
- Hierarchy - arrange visual information that implies importance. e.g., headline of text is larger than the text under it [22].

Usability often refers to how easy a product is to use and whether the product can achieve the users' goals. With accessibility taken into account throughout the design process, usability can also increase since even users with disability can use the product.

4 Current State Analysis

How can we incorporate UX and UI into an app for the visually impaired elderly? What are the existing problems and how can we solve them?

4.1 Existing problem

In this study, target users for the project app were seniors over 60 years old, who had impaired or zero vision. For current state analysis, we first looked at the physical problem they had.

4.2 Limitation due to ageing

The World Health Organisation reported that the proportion of the world's population over 60 years will almost double from 12% to 22% between 2015 and 2050 [23]. With the increasing number of elderly people, more older adults will have to embrace technology. Technical companies are finding ways to market their platform and software to them. However, many elderly users find it hard to navigate themselves in the digital design. They find it frustrating because the design either does not meet their needs or it is hard to use. Elderly often have physical limitations and visual acuity due to ageing [24]. They might have hearing loss, loss of fine motor control, loss of attention, psychomotor impairments and other learning and memory impairments [25]. These limitations can lead to difficulty and frustration while using apps. UX designers should take these limitations into account and come up with a solution to help elderly interact with the app and navigate through.

4.3 Elderly neglected in UX design process

Elderly are often neglected in the design process of smartphone applications. Their needs and requirements are not taken care of [26-28]. In UX design, it is important to focus on the users - find out what they need, what pains them and what motivates them. To find out, UX designers should include the user group in research, interviews, surveys and user testing.

4.4 Stakeholder interview

To find out more about the problem that elderly face when using smartphones, a group of people were recruited to the stakeholder interview. They were all above 65 years old with different degrees of physical limitations and visual acuity. The first interview question was to find out if they were using smartphones or normal mobile devices such as Doro. If they were using smartphones, we looked into the kind of apps they had. Then we moved on to the wishes they would like to have as an app and what pains them.

After collecting the feedback, the direction of the app was more clear - an app that targeted elderly who had some degree of physical or eye limitations. Another user interview was carried out on the app prototype, with transcript recording and more questions.

5 Implementation - Building a navigation app for Visually impaired elderly

In this study, the UX experience design process was used. The first step was to discover who the users were, what made them suffer and what interests they had. User study was carried out as a result.

5.1 User study

User interviews were conducted with 3 seniors aged 70 or above. The transcript was saved in Appendix 2. The interviewees had limitations such as impaired vision, swelling fingers from arthritis and lack of knowledge in the digital world. One of them had never used any apps in her life. None of them were technology-savvy.

Interview questions were set as follows:

- What kind of app(s) are you using? Do you like it? What are the great features?
- What kind of app would you like to have? What are your requirements for a good app?
- What do you not like in an app? What frustrates you the most?
- Any app you would like to have?

Table 2. Results from user interviews.

Name	Apps using	Why are you using those apps?	What is frustrating in an app?	Apps that you want or can help you?
Tarja	Whatsapp, HSL Reittiopas, Facebook, Camera, Video	Not too complicated, useful and solve my problem	Swelling fingers, small texts, difficult interface	None, happy with what I have
Tepa	None	solve my problem, easy to	Swelling fingers, difficult	Google maps that I can use

		use even without perfect vision, easy to navigate	interface, too small texts for my eyes, bad colour contrasts	even though I am quite blind.
Carina	Whatsapp, Facebook, Camera, Video	Texts are big enough, useful app, easy to use	Swelling fingers, small text, not sure what to do next	None, happy with what I have

Results of the interview are shown in table 2 above. Throughout the interviews, all of the participants complained about the font size. Sometimes they needed a magnifying glass to see the details. They had difficulty in pressing the correct key and interacting with the app due to their “swelling fingers”. They felt that smartphone apps were made only for youngsters. They would like to have an app that could solve their problem and is easy to use despite their physical limitations.

During this research, an app idea had come up. It was a navigation app targeting vision impaired elderly with audio clues and visual clues that are big enough to be seen.

After the user interview, a persona was drawn. As mentioned earlier, persona is very much used by UX designers and it is very important in the design process. Through persona, we can understand the needs of the users.

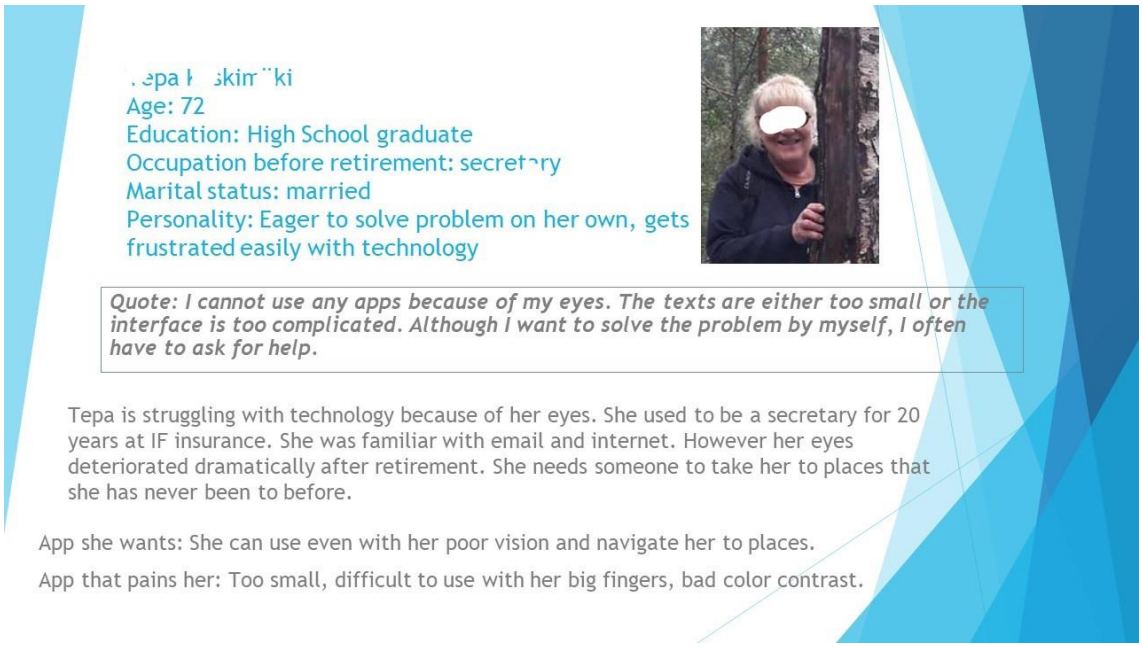


Figure 4. User persona made after the user interview

As illustrated in figure 4, user persona is a fictional character and a representation of the intended user group: visually impaired elderly who have colour vision defects and little knowledge in using digital devices and apps, but would like to have an app that could make her life easier.

5.2 Market research

Similar apps in the market tailor made for visually impaired users were investigated. They were navigation apps with accessibility functions, which could be downloaded for free on Android and iOS devices:

App investigation 1: RightHear

The app provides navigation to nearby locations, which are restaurants or shops who partnered with the app company.
Pros: Many features such as indoor navigation and audio clues.
Cons: Difficult to use, too many options and too small icons. Cannot find any landmarks close to Espoo and Helsinki. Companies or shops have to pay to be listed on this app. It is basically useless in Finland.

App investigation 2: SoundScape

The app provides navigation to a place or address through a search function. It can explore places nearby
Pros: Provide audio clues while navigating the designated place.
Cons: Only available on iOS and there are bugs. It does not pin-point one single shop in a mall, instead, it only leads users to the mall. It does not work in a multi-level building.

Market research results

Both of the apps use audio clues for users. However, users have complained about the RightHear app that the visual elements are too small and not easy to

use. As for SoundScape, it was said that when users are inside a building, it is impossible to navigate if it has multiple floors.

The market trends, likes and dislikes of the two existing apps in the market were taken into consideration. The project app would include audio clues, bigger visual elements and an easy-to-use interface.

5.3 Ideation

User story was done to brainstorm ideas and what were the features needed in the app.

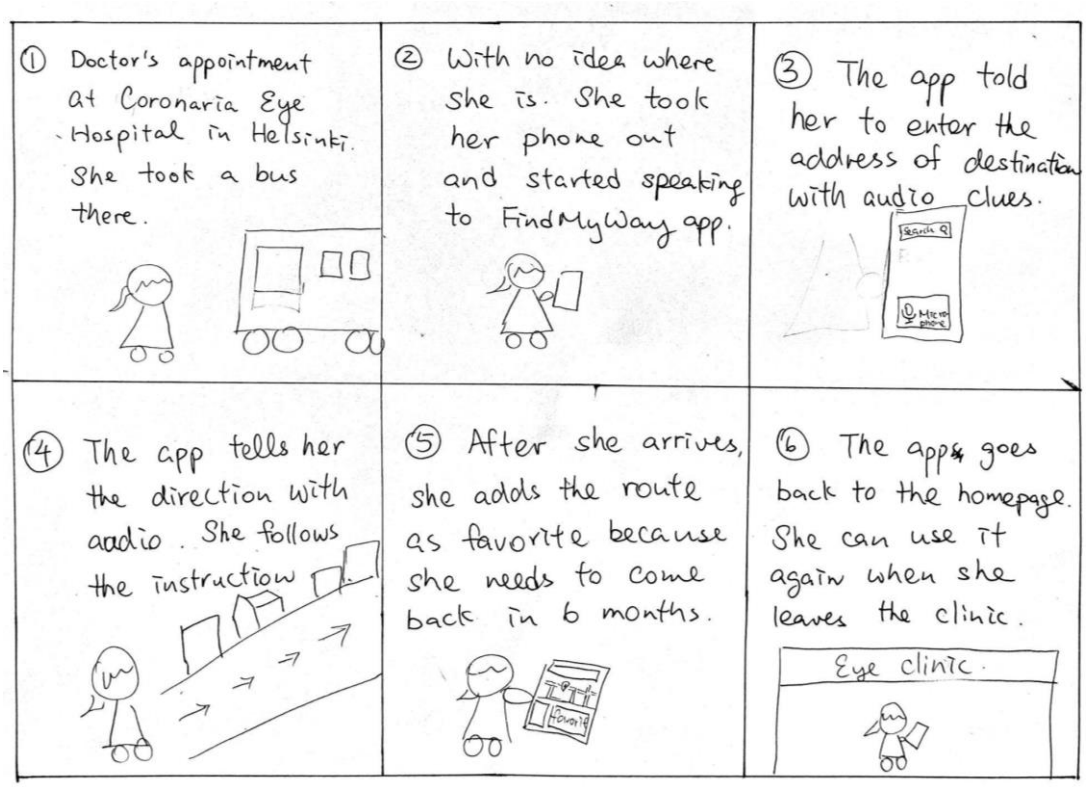


Figure 5. User story

In figure 5, we could see that a user story was drawn. It was separated into different scenes: why she needed to use the app, how she would use it and what the apps could do for her.

Then, an affinity diagram was drawn (please refer to appendix 1). Affinity diagram was used to generate ideas for the app. It is a method used in UX design to quickly organise:

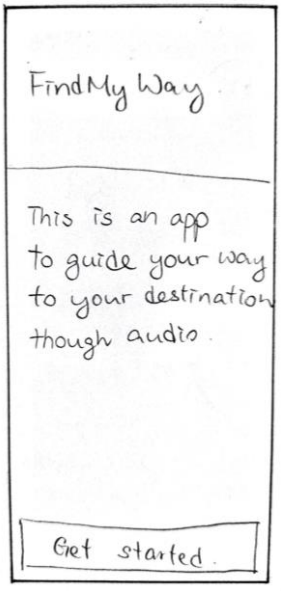
- observations or ideas from a research study;
- ideas that surface in design-ideation meetings;
- ideas about UX strategy and vision [29].

Sticky notes were generated with the ideas written on, then sorted into categories and prioritised one by one.

Affinity diagrams give ideas about what kind of UI elements needed from the accessibility point of view, and what were the navigation elements to be included to meet the needs of users.

5.4 Paper prototype

A paper prototype of the app was created:



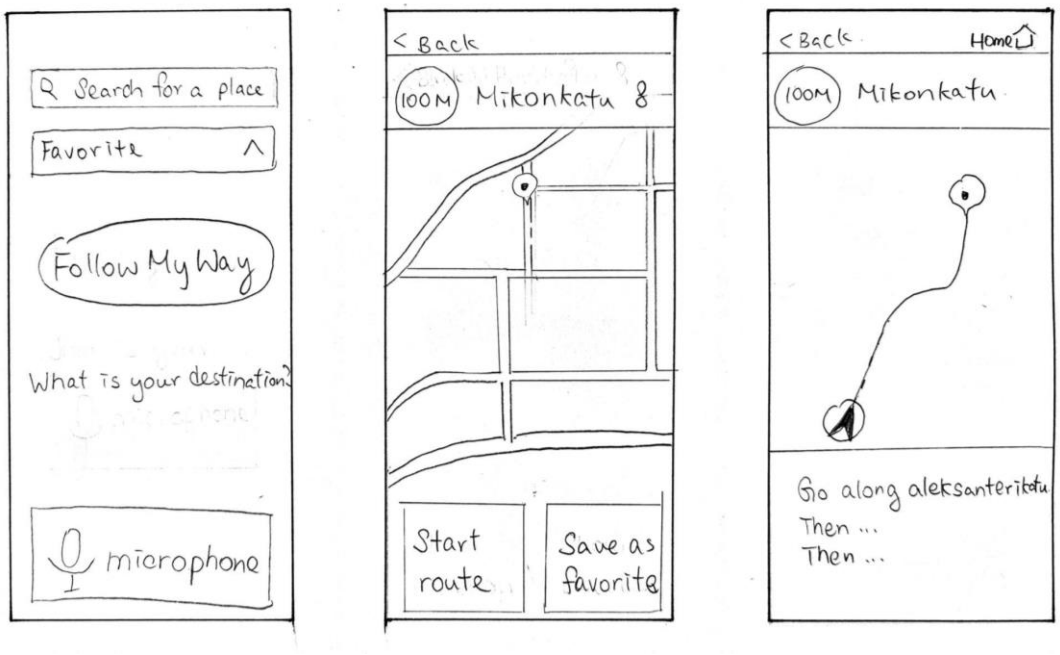


Figure 6. Paper prototype

A paper app prototype includes what views should be included in an app. Figure 6 above showed the paper prototype of the project app. In total, there were 4 views in the app:

1. info and introduction page
2. main page
3. map page
4. navigation page

5.5 Wireframe

A wireframe prototype (figure 7 below) was made based on the paper prototype. It was a set of images that display the functional elements of the app.

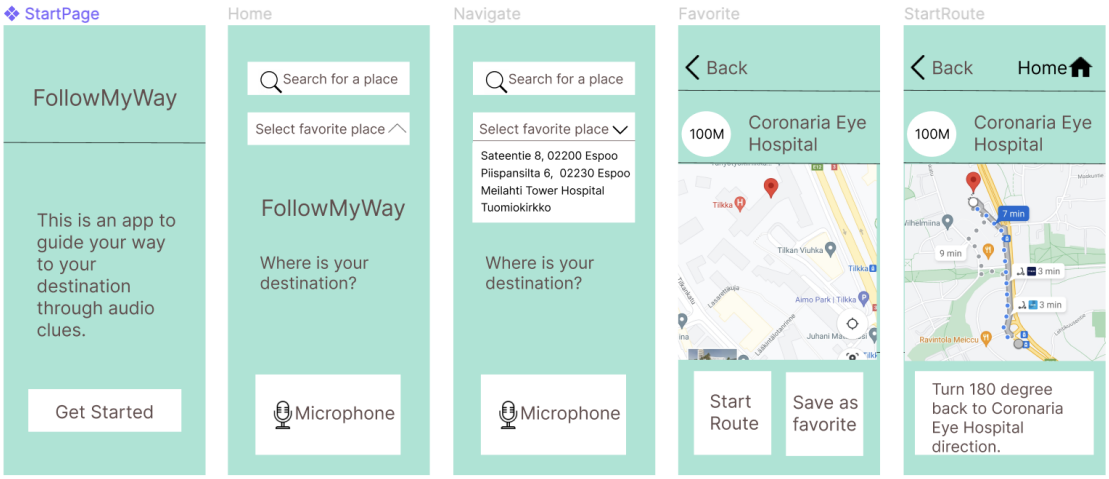


Figure 7. Wireframe prototype

While making the wireframe prototype, the needs of the user were taken into consideration. The icons were big, the fonts were large and there was enough colour contrast, so that even users with visual impairment could see them. Since the target user was not technology-savvy, icons were named and specified accordingly. Users could use voice recognition to input the destination address and use audio clues to control the app.

5.6 App flow

Usability is one of the most important elements in UX design. A high usability app is normally one that is easy to interact with and there is a natural workflow. Therefore, a flow diagram was created for user testing:

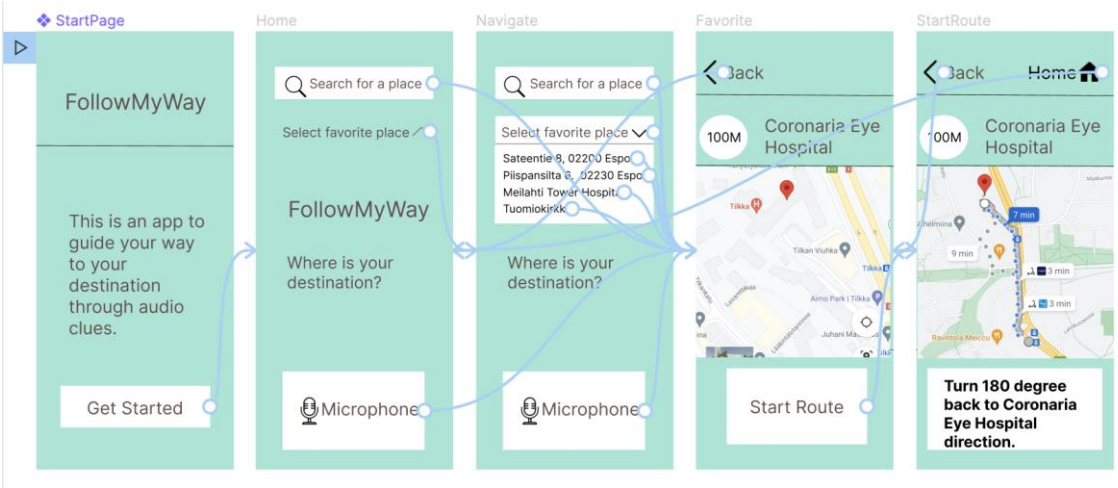


Figure 8. Flow diagram: wireframes with app flows.

Figure 8 above is a flow diagram: a series of wireframes connected by app flows. In the flow diagram, every UI element was linked with each other. For example, after users pressed the “Get Started” button, it would link to the main page. After user input (audio), it could link to the map page, and after pressing or saying “Start Route”, it will link to the route page and the audio clues will tell the user how to get to the destination. Audio clues were used to make sure the user can navigate from one UI element to another. A smooth and natural flow was ensured.

5.7 Accessibility

One of the goals in this app was to make sure that the users could use the app without difficulty. To achieve that goal, accessibility features below were taken into account:

- Colour blind palette;
- Large text and standard font style;
- TalkBack;
- Switch Access;
- Speech Recognition with Voice User Interface;
- Visual Cues and Google Live View;
- Jetpack Compose Accessibility.

Color contrast

Our target user was visually impaired elderly, who have colour vision defects. Since 1883, colour vision defects in glaucoma have been mentioned. The colour defects are divided into “blue-yellow” and “progressive red-green blindness”. It means that the patients either are colour-blind to red and green colour or blue and yellow colour [30]. In their eyes, those colours are just different shades of brown as listed on the right side (Protanope Simulation) in Figure 9 below.

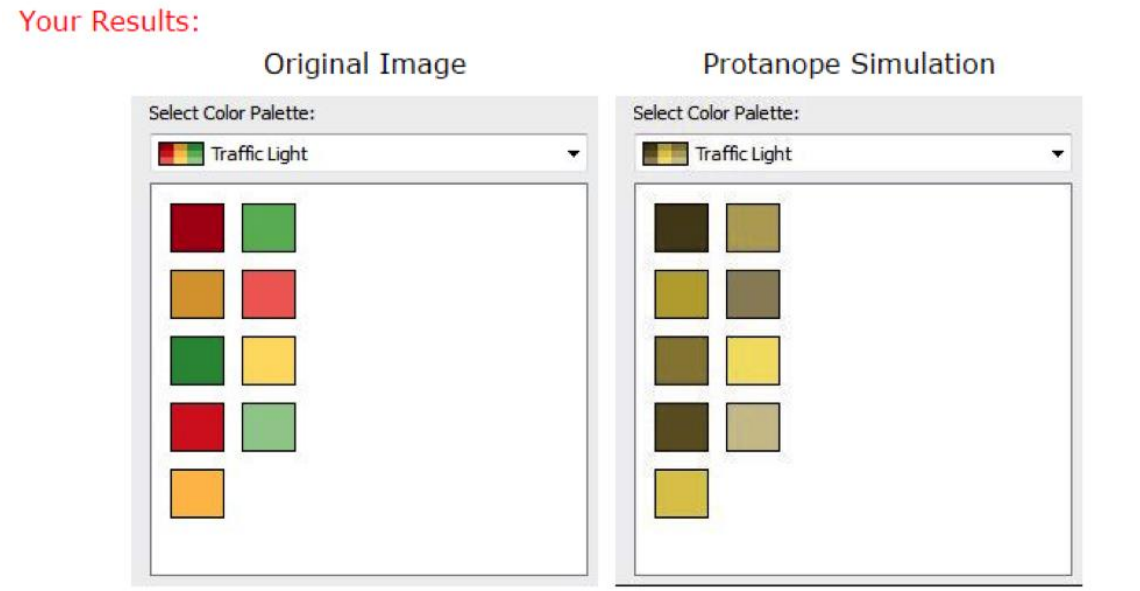


Figure 9. Simulation of Tableau's stoplight colours using protanope simulation [31]

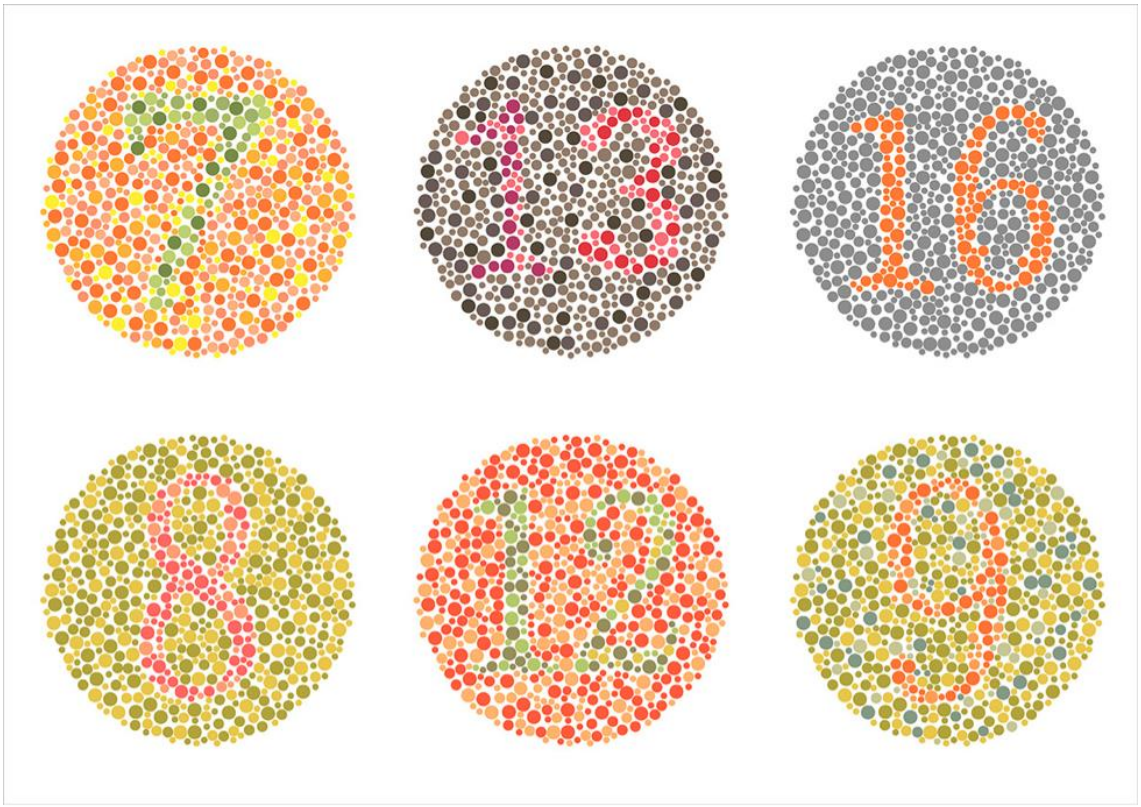


Figure 10: Colour test for the colourblind [31]

Figure 10 is a colour test for the colourblind. People with colourblind have difficulty to spot the numbers inside the circle because red and green colour have very little contrast in the eyes.

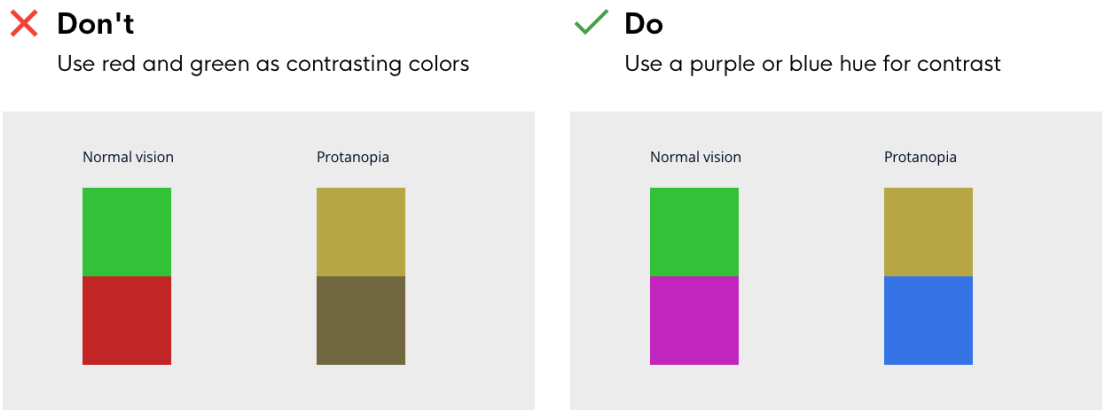


Figure 11: Do's and Don't in colour contrast for the colourblind. [32]

When designing for the colourblind, we should avoid using red and green as contrasting colours. As illustrated in figure 11, designers should use a purple or blue hue for contrast.

Colour palettes have been developed to increase accessibility for the colourblind. However, designers should not convey information only through colours because colour-blinded individual may have a totally different experience. It is important to include visual attributes such as shapes, iconography, text, contrast, and spacing makes information accessible to colourblind people.[33]

Colour contrast ratio should also be taken into consideration. Based on Web Content Accessibility Guidelines (WCAG), colour contrast refers to the difference in brightness or perceived "luminance" between two colours. It is often used to measure accessibility compliance [34]. For example, the brightness difference between two colours ranges from 1:1 (white on white background) to 1:21 (white on black background). The minimum ratio required by WCAG is 1:4.5.

Besides colour contrast, WCAG clearly stated that the contrast level for text to be deemed accessible, it should meet the following requirement:

- Headings: Contrast ratio for large-scale text should be a minimum of 3:1;
- Main Text: For text other than headings, a contrast ratio of at least 4.5:1 should be maintained;
- Logotype: Text which is a part of a logo or brand has no contrast requirement;
- Incidental Text: Text which is part of an inactive user interface, pure decoration, invisible, or part of a picture, and does not convey meaningful information, has no contrast requirement [35].

To make sure the colour contrast is enough for users, the accessibility scanner was used to check the contrast ratio of the app. The app passed the test, except for the logo on the first page.

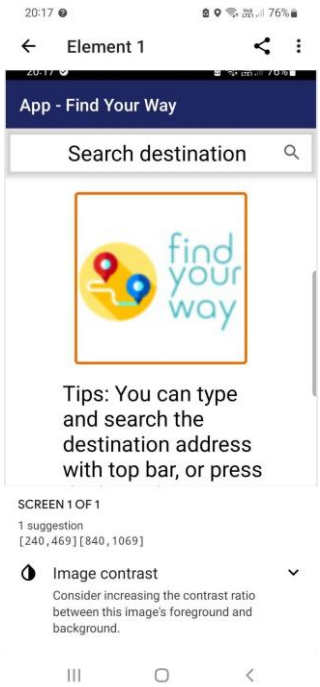


Figure 12. Main screen on the app with logo that has not enough contrast ratio

In figure 12, the app logo was highlighted by the accessibility scanner as it did not give enough contrast ratio. The logo colour scheme was changed to black and white and it passed the accessibility scanner test afterwards.

Large text and standard font style

The 3 key factors to determine whether a font is accessible are size, colour and contrasts [35]. Custom fonts should be easy to read and legible. Full text justification with too much white space, italics and other font styles are not recommended because it may be difficult for users to read. [36] Preferred fonts are Tahoma, Calibri, Helvetica, Arial, Verdana, and Times New Roman [35]. These fonts have high legibility, good width, height and thickness.

Elderly users often complain the texts are too small in apps. To solve the problem, larger font sizes are used. Some of the experts suggested that font size in the app should not be less than 16px. It is also said that users should be able to adjust the font size. [37] However, when the font sizes increase, the amount of space on the screen decreases and text truncation will happen. Text truncation is often used in digital design. It is the act when the text gets shortened or removed [38]. As per Apple developer's accessibility guide on text size and weight, when text size increases, text truncation should be kept to a minimum [39]. Designers should aim to display as much useful text in the largest accessibility font size as possible. Text truncation with scrollable regions can confuse elderly and decrease usability.

In the project app, except for the top bar and navigation bar. The font size was above 25dp throughout the app. The main buttons for navigation were 400x90dp, much greater than the 48x48dp touchable target required by Jetpack Compose accessibility. The custom font was Arial, which was legible and easy to read.

TalkBack

TalkBack is a screen reader that takes the content on screen and reads it out loud to users [40]. TalkBack is built on Android's AccessibilityService APIs. Since it is open source, the source codes can be found on GitHub. Visually impaired users can listen to TalkBack and navigate in the app without looking at the screen.

To test the Jetpack Compose Accessibility setting in this app, TalkBack had been switched on for user testing. Since the user may use TalkBack to navigate through the app, every element on the UI should be investigated so that the UI sounds neat and easy to understand.

Firstly, visual elements on the screen should be described by contentDescription parameter, so that TalkBack can announce the elements. Throughout the app, visual elements were marked with ContentDescription, as illustrated in listing 1 below:

```
IconButton(onClick={}{
Icon(
ImageVector=Icons.Default.Search,
contentDescription=stringResource(id=R.string.Enter_address)
)
}
```

Listing 1: contentDescription describes the element in words.

Secondly, the onClickLabel parameter was used to add semantic meaning to the click behaviour of a composable [40]. TalkBack could then explain to the user what will happen when the user interacts with this composable. Listing 2 showed how onClickLabel was used in the app:

```
Button(
onClick={navController.navigate(Screen.Main.route)},
modifier=Modifier
.clickable{
onClick={},
onClickLabel=stringResource(id=R.string.Click_to_LiveView)
)
)
```

Listing 2: Clickable function to tell user what will happen after Live View button is clicked.

When TalkBack was on, it announced “click button to see live view”. User would understand what would happen after it has been clicked.

Switch Access

Switch Access is another accessibility service provided by Android. Users who have problems pressing the buttons and icons to interact with the app, can use external switches. There are several kinds of switches:

- External switch : External device sends a keystroke signal to Android. It is connected by USB or Bluetooth;
- External keyboard: Configure a USB or Bluetooth keyboard to work as a switch device;
- Use buttons on Android devices: Built-in buttons on Android phones can be assigned to actions [41].

In the project app, the user could use an external switch to connect to her app and pass action from there. Using buttons on Android devices would not be possible because users might accidentally touch the volume buttons while using TalkBack. External keyboard was also inconvenient while navigating on the street.

Speech Recognition with Voice User Interface (VUI)

World Wide Web Consortium (W3C) mentioned that speech recognition is about recognizing words for speech-to-text (STT) transcription, virtual assistants as well as other speech user interfaces. It is used to dictate text in a form field, navigate and activate UI elements. [42] It is for people who prefer to speak, rather than to type. It is a feature that is very crucial to our target user who has limitations in eyes and fine mobility.

For those who prefer to speak and avoid touching the phone screen, the voice user interface can be triggered by voice. To ensure the words are being ingested and processed by the device, feedback is necessary to let users take affirmative or corrective action [43]. Audio playback, output text or real-time text can be used to confirm the speech. Audio playback is a simple playback to confirm the interpretation of speech. Real-time text is textual feedback that appears immediately as the user speaks. Output text is the one that is transformed and

amended after the user has finished speaking. [43] Since elderly users prefer a simpler interface, the amount of text elements should be limited. Real-time text and output text will not be used in the current app design. To check whether the speech was correct, users will be asked a “yes” or “no” question. If the user says “no”, VUI will be triggered and users can input the address again through speech.

In VUI design, it is essential to create prototyping VUI conversations with dialogue flows. A dialog flow is a deliverable that outlines the following:

- Keywords lead to interaction;
- Branches represent where the conversation could lead to;
- Example dialogues for the voice assistant and the user [44].

Here is a dialog flow for the project app:

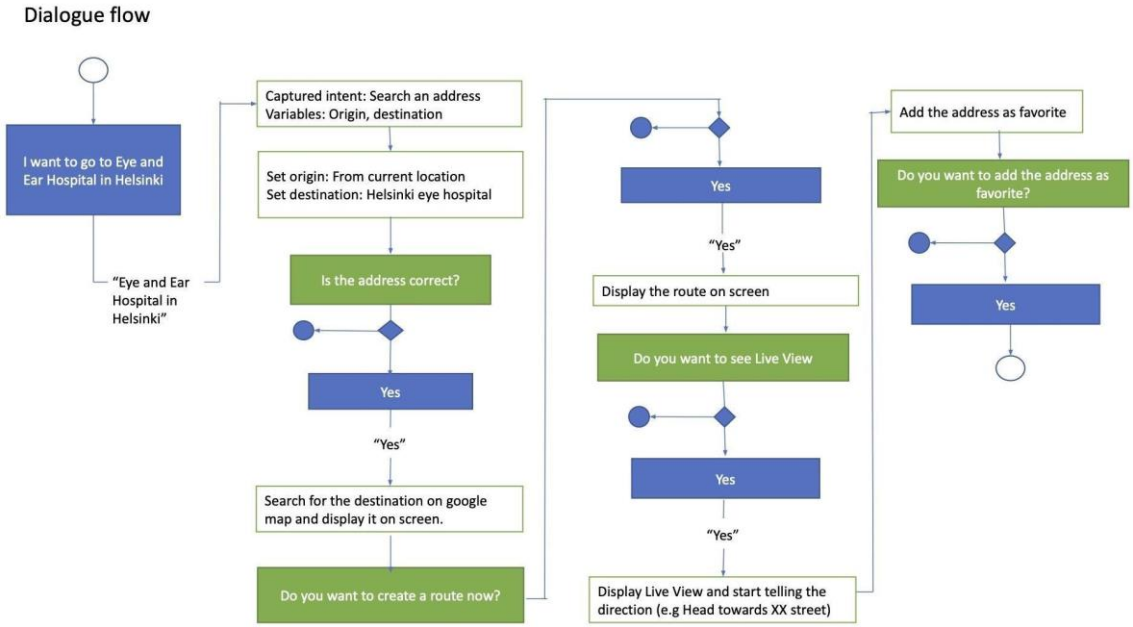


Figure 13. A dialog flow illustrating the intent, slot and overall conversation.

In figure 13, example dialogues between the voice assistants and the users are outlined in the flow. The dialogue flow diagram also shows where the conversation could lead to. Keywords that lead to interaction are also included.

Visual cues and Google Map Live View

Including visual cues in user interface design can help elderly because of the reduced abilities regarding divided attention and fluid intelligence [45]. As mentioned earlier, many elderly users find it frustrating to navigate themselves through different UI elements and screens in an app. Visual cues are added in the user interface, so that the design is easy to decipher, clear and not difficult to interact with [46].

Designers should think like the users and create designs that users can understand. It is not enough that the design is clear, it should also be easy to understand and interact with.

In the beginning of the project app design process, two wireframe prototypes were created:

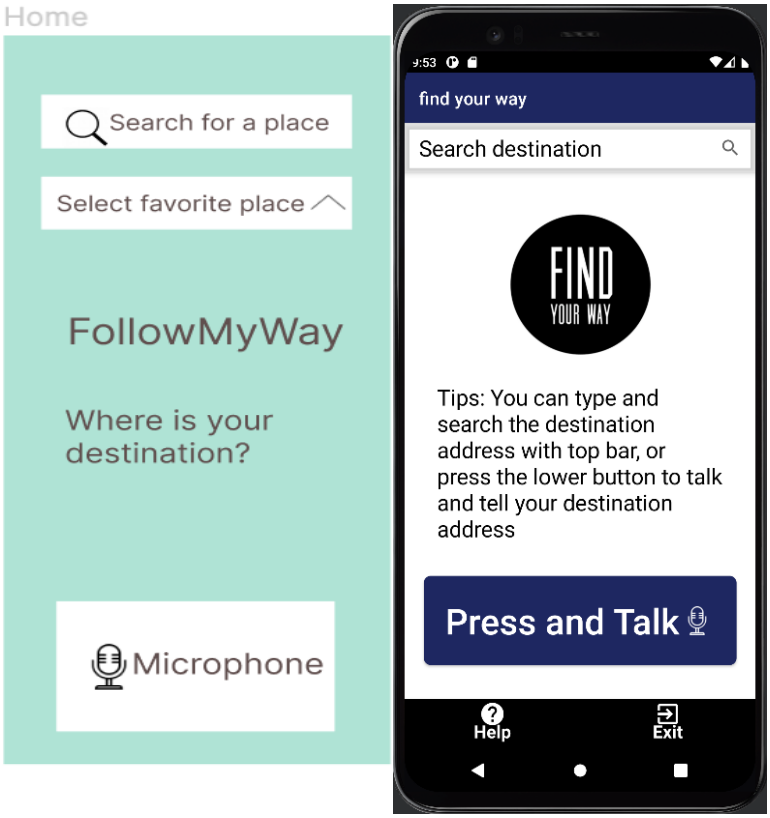


Figure 14. App design prototype – Main screen with microphone icon. Prototype on the left was the original version, to the right the improved version.

As illustrated in figure 14, in the original prototype (left picture), the button's text was "Microphone". It did not reflect that it was a button for speech recognition. After consideration, the button text was changed to "Press and Talk" (right picture), so that users could understand what this button was for.

In Google Map Live View, it displays the navigation instructions on the phone, while pointing a phone camera at buildings, street signs, or any element of scenery that Google Maps can recognize [47]. Navigation instructions provide visual cues to users, telling them which direction to go and where to turn. Google Map Live View was tested and examined in the project app design process, as it could be useful in the app development for visually impaired elderly.

Jetpack Compose accessibility

As mentioned earlier in the project specifications, Jetpack Compose is used while building the app. It is a modern toolkit for building native Android UI, which uses composition to describe the UI of the app. Composition is a tree-like structure that is made of composables which describe the UI. Next to the Composition, Semantics tree is a parallel tree describing the UI that can be understood for Testing framework and Accessibility services.

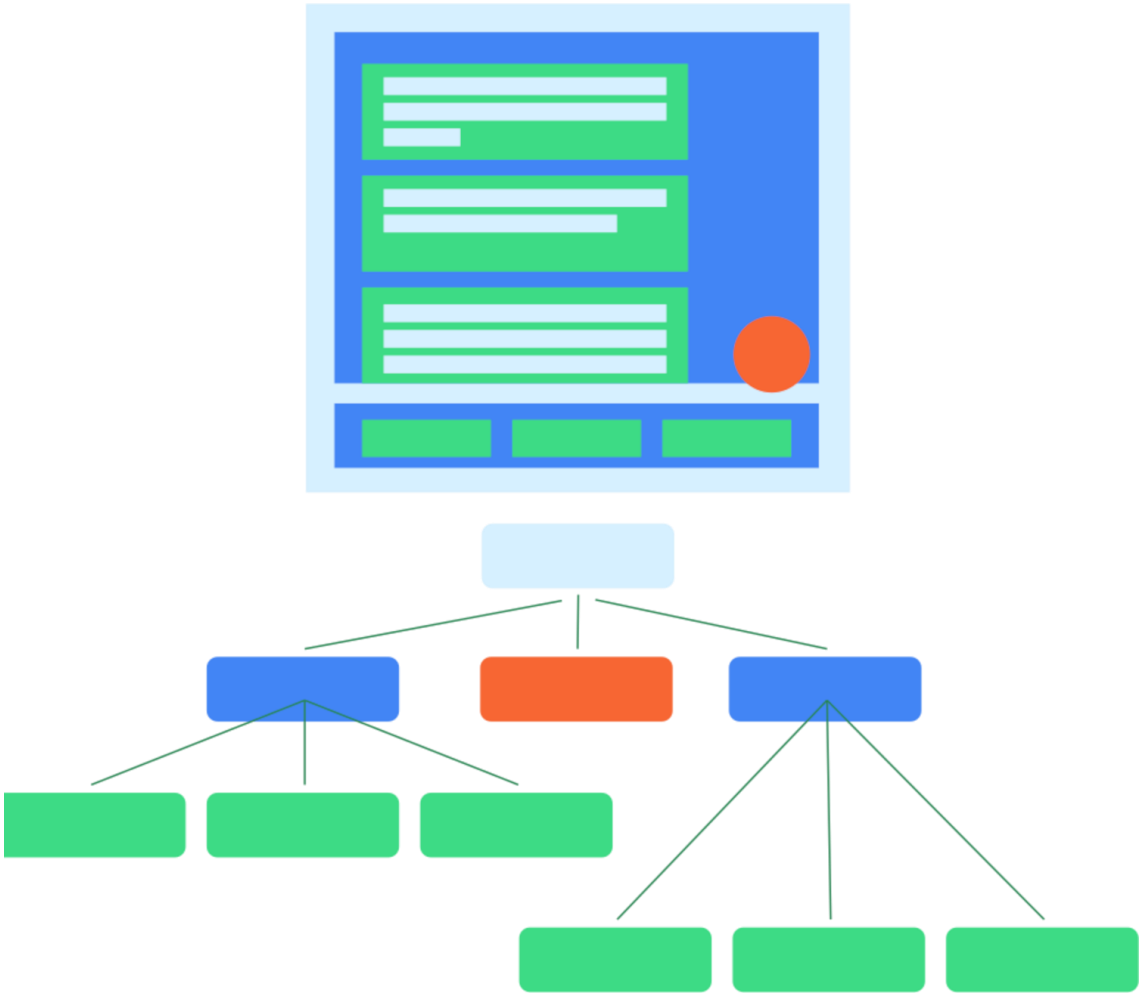


Figure 15. A typical UI hierarchy and its semantics tree. [48]

Figure 15 showed a typical UI hierarchy and semantic tree. Semantics tree is automatically generated if composable and modifiers are used in the app. All nodes in the UI tree with the seminar meaning have a parallel node in the semantics tree.

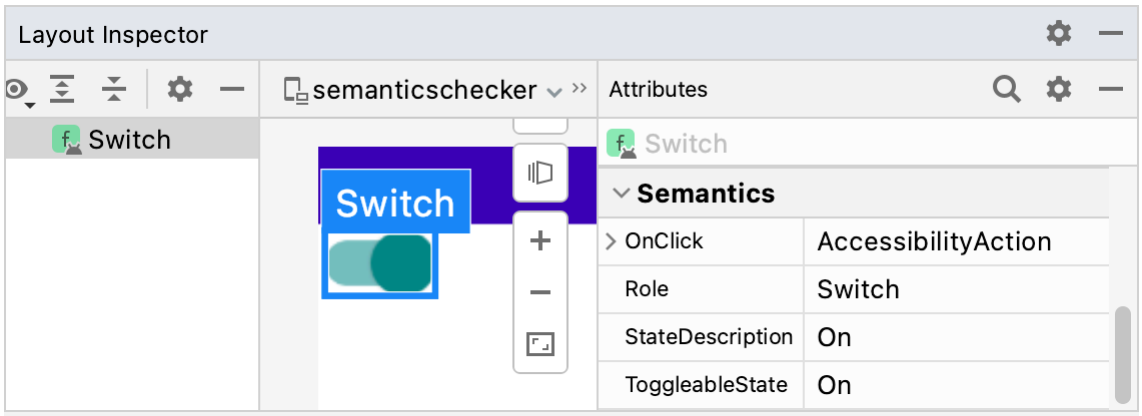


Figure 16. Layout Inspector showing the semantics properties of a Switch composable. [48]

In mobile app development, layout inspector is used to examine a compose layout. In figure 16, when the layout inspector is opened for a switch composable, semantic properties of the composable are visible. With AccessibilityAction included in OnClick, it describes the app to users with specific needs and give accessibility focus to the node. [48]

When designing an app for users with limitations, accessibility should be taken into great consideration. In Jetpack Compose, principles below were followed to ensure accessibility for users:

- ContentDescription with Talkback;
- Custom merging;
- Add actions to lists;
- Touchable target size;
- Headings;
- Switches and checkboxes;
- State descriptions. [49]

ContentDescription with Talkback

Since TalkBack reads everything on a screen, whether it is trivial or not, there is no hierarchical representation for the UI. Instead of reading everything trivial, contentDescription is used. Talkback will only read the text content of the node.

```
IconButton(
    onClick = { },
    icon = { Icon(Icons.Default.ArrowBack) },
    modifier = Modifier.semantics {
      contentDescription = "press to go back to previous page"
    }
)
```

Listing 3: Describe visual elements with contentDescription in Jetpack Compose. [49]

From listing 3, the back button can include a contentDescription, to guide the user back to the previous page.

Click labels can add semantic meaning to the click behaviour of a composable. Clickable elements generally do not provide any information about what clicking can do. When TalkBack is used, a generic default description will be given such as “Double tap to activate”. [40] By using onClickLabel in clickable, a specific description of the label can be included:

```
@Composable
fun StartRoute(startRoute: () -> Unit) {
   Card(
       onClick = startRoute,
       // R.string.action_start_navigate = "start navigating"
       onClickLabel = stringResource(id = R.string.action_start_navigate)
    ) {
   }
}
```

Listing 4: Describe visual elements with onClickLabel in Jetpack Compose. [49]

In listing 4, once the user clicked the “Start Route” button on the app, TalkBack would say “Click button to start navigating”.

Custom merging

In Jetpack compose for accessibility, it is essential to merge multiple composables elements into one focusable entity [49]. As having too many focusable elements can lead to confusion as the TalkBack will navigate them one by one. In the project app, before using custom merge, TalkBack will read “100M” and “Coronaria Eye Hospital” one after another in the UI below.

Figure 17. Multiple composable example from existing app design.

Figure 17 shows multiple composables in the same view. With the merge multiple composables function, it would read “100M Coronaria Eye Hospital”, as we could see from figure 17.

Composables can be merged using semantic modifier with its mergeDescendents property:

```
@Composable
private fun PostMetadata(metadata: Metadata) {
   // ...
   Row(Modifier.semantics(mergeDescendants = true) {}) {
        TextField(
            // ...
        )
        Spacer(Modifier.width(8.dp))
        Column {
            Text(
                // ...
            )
      CompositionLocalProvider(LocalContentAlpha provides ContentAlpha.medium)
{
                Text(
                    // ..
                )
            }
        }
   }
```

Listing 5: Merge multiple composables with mergeDescendents in Jetpack Compose. [49]

In listing 5, mergeDescendants is used to merge multiple composables.

Add actions to lists

TalkBack can read what is on the screen. When there is a list of items and icons for each of the list items, TalkBack will first read what is the item, then the icon. In a long list of items, it will be very repetitive and can easily annoy the users. The same action mentioned will repeat itself many times until the end of the list. To avoid frustration, designers have to remove the behaviour of icons and make sure accessibility service will not select it. [49]

The follow example is about a list of book names and a bookmark icon behind it:

```
@Composable
fun PostCardSimple(
  /* ... */
  isFavorite: Boolean,
  onToggleFavorite: () -> Boolean
) {
  val actionLabel = stringResource(
    if (isFavorite) R.string.unfavorite else R.string.favorite
  )
  Row(modifier = Modifier
    .clickable(onClick = { /* ... */ })
    .semantics {
      // Set any explicit semantic properties
      customActions = listOf(
        CustomAccessibilityAction(actionLabel, onToggleFavorite)
      )
    }
  ) {
    /* ... */
    BookmarkButton(
      isBookmarked = isFavorite,
      onClick = onToggleFavorite,
      // Clear any semantics properties set on this node
      modifier = Modifier.clearAndSetSemantics { }
    )
  }
}
```

Listing 6: Add action to list in Jetpack Compose. [49]

With the codes in listing 6, behaviour of the bookmark icon is removed. When TalkBack reads the list of books, it will only read the name of the books, not the icons.

Touchable target size

Any UI elements that can be clicked, touched or interacted should have a width and height of at least 48dp [49]. Sometimes the UI elements are sized dynamically and resized based on the size of their content. Designers should use sizeIn modifier to set a lower dimensions limit to ensure the UI element is large enough. In addition, designers can also add padding to the elements:

```
Row(
        // ...
    ) {
        // ...
        CompositionLocalProvider(LocalContentAlpha provides
ContentAlpha.medium) {
            Icon(
                imageVector = Icons.Default.Close,
                contentDescription = stringResource(R.string.cd_show_fewer),
                modifier = Modifier
                    .clickable { openDialog = true }
                    .padding(12.dp)
                    .size(24.dp)
            )
        }
    }
    // ...
}
```

Listing 7: Add clickable function to add padding and include overall touchable target size. [49]

In listing 7, clickable function is used to add padding and increase overall touchable target size. With padding added after the clickable function, the touch target will be at least 36dp. With IconButton Function, the touch target will be at least 48dp. The IconButton function also adds a ripple indication to show the users that the element is clickable.

```
@Composable
fun PostCardHistory(post: Post, navigateToArticle: (String) -> Unit) {
    // ...
    Row(
        // ...
    ) {
        // ...
        CompositionLocalProvider(LocalContentAlpha provides
ContentAlpha.medium) {
            IconButton(onClick = { openDialog = true }) {
                Icon(
                    imageVector = Icons.Default.Close,
                    contentDescription = stringResource(R.string.cd_show_fewer)
                )
            }
        }
    }
    // ...
}
```

Listing 8: IconButton function. [49]

Headings

When a screen contains a lot of text, such as in a News App, users with visual limitation may not be able to find the section they are looking for. With Jetpack

Compose accessibility, designers can indicate which parts of the texts are headings. [49]

```
@Composable
private fun Paragraph(paragraph: Paragraph) {
   // ...
   Box(modifier = Modifier.padding(bottom = trailingPadding)) {
       when (paragraph.type) {
           // ...
           ParagraphType.Header -> {
               Text(
                   modifier = Modifier.padding(4.dp)
                     .semantics { heading() },
                   text = annotatedString,
                   style = textStyle.merge(paragraphStyle)
               )
           }
           // ...
       }
   }
}
```

Listing 9: Semantic function indicates which part of the texts are heading. [49]

With the code in listing 9, heading semantic property can be set to indicate which parts of the composable are headings. User will be able to find the sections they are looking for.

Switches and checkboxes

Switch and checkbox are read out loud about their checked state as they are selected by TalkBack [40]. Checkboxes are often separated from their labels by default. It is repetitive, difficult to understand and annoying when Talkback reads out the switch and checkbox. To make it easier for the users to understand, labels and checkboxes can be linked together and the whole row will be made toggleable. It is illustrated in figure 18 below.

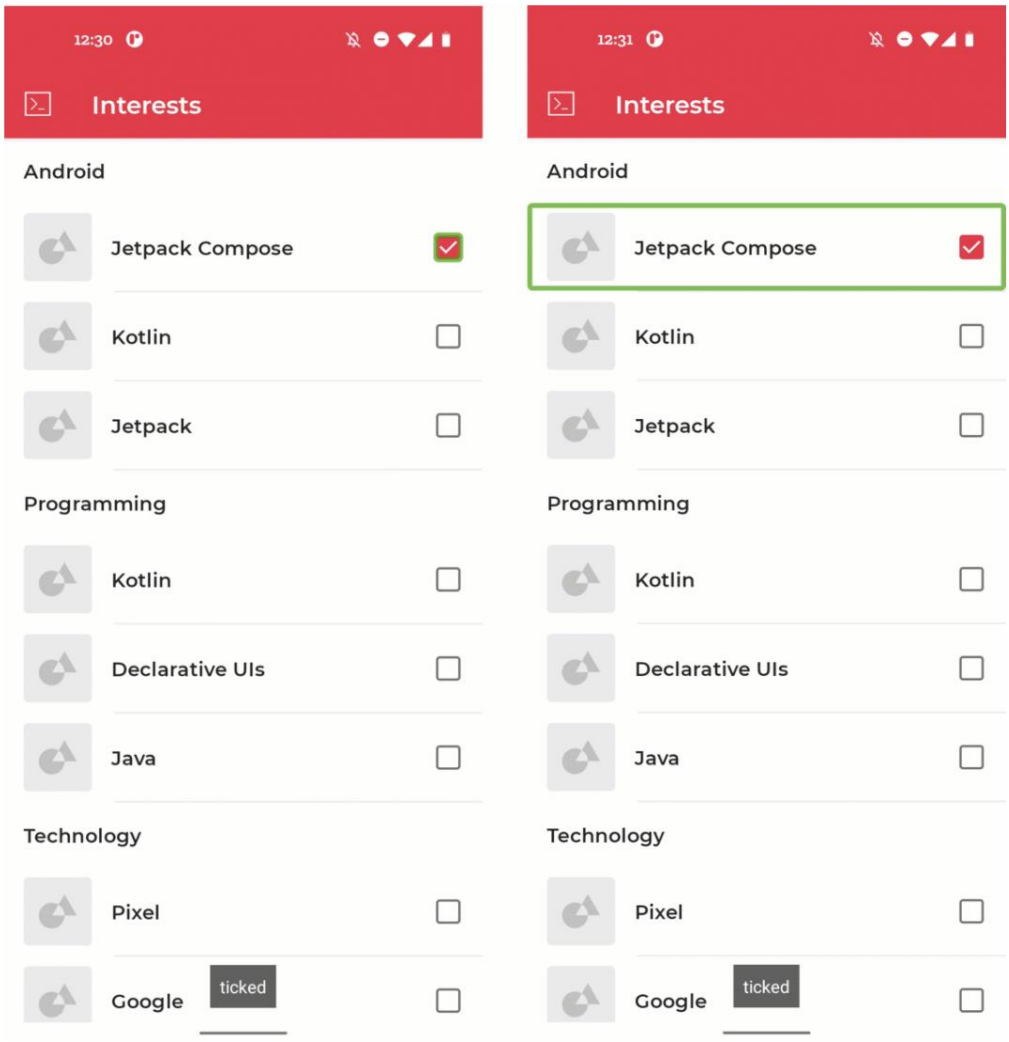


Figure 18. Working with checkboxes. Before (on the left) vs after (on the right). [49]

```
@Composable
private fun TopicItem(itemTitle: String, selected: Boolean, onToggle: () ->
Unit) {
    // ...
    Row(
        modifier = Modifier
            .toggleable(
                value = selected,
                onValueChange = { _ -> onToggle() },
                role = Role.Checkbox
            )
            .padding(horizontal = 16.dp, vertical = 8.dp)
    ) {
        // ...
        Checkbox(
            checked = selected,
            onCheckedChange = null,
            modifier = Modifier.align(Alignment.CenterVertically)
        )
    }
}
```

Listing 10: Label and checkedbox are combined and become toggleable. [49]

In listing 10, the checkbox toggleable function is lifted and transferred to the whole row. After TalkBack announces the checkbox, it will say “ticked, Jetpack Compose” instead of “ticked”.

State descriptions

State description can be added to toggleable elements. Instead of announcing “ticked” or “not ticked” by TalkBack, a description can be added using stateDescription inside semantic modifier:

```
@Composable
private fun TopicItem(itemTitle: String, selected: Boolean, onToggle: () ->
Unit) {
   // ...
   val stateNotSubscribed = stringResource(R.string.state_not_subscribed)
   val stateSubscribed = stringResource(R.string.state_subscribed)
   Row(
       modifier = Modifier
           .semantics {
               stateDescription = if (selected) {
                   stateSubscribed
               } else {
                   stateNotSubscribed
               }
           }
           .toggleable(
               value = selected,
               onValueChange = { _ -> onToggle() },
               role = Role.Checkbox
           )
           .padding(horizontal = 16.dp, vertical = 8.dp)
   ) {
       // ...
       Checkbox(
           checked = selected,
           onCheckedChange = null,
           modifier = Modifier.align(Alignment.CenterVertically)
       )
   }
}
```

Listing 11: Toggleable element state Description added by using stateDescription in semantics. [49]

With listing 11, TalkBack can now say “subscribed” or “not subscribed” instead of “ticked” or “not ticked”.

In the project app, only 3 functions from Jetpack Compose accessibility: ContentDescription with Talkback, onClickLabel and CustomMerging were

implemented. As the app did not contain switches and checkboxes, toggleable elements nor headings.

5.8 Competitive app testing

To experience the apps with accessibility in the market, live testing was conducted. The following apps were tested:

- Google Map Live View
- TalkBack
- Voice Assistance (Google)

Google Map Live View

As mentioned earlier, Google Map Live View could be useful for app development in this project. Live testing was carried out to explore the features, whether it was suitable for the visually impaired elderly.

First of all, the speech recognition for address input on Google Map was good. The app understood the street name, as well as the street number most of the time. The only time it did not work was when “Eye Hospital” was spoken. The output text was “I Hospital.

Then, after switching to Live View, the phone camera was switched on and pointed to buildings, street signs, or any element of scenery that Google Maps could recognize. The app then displayed the navigation instructions on the phone. The process was quite smooth. However, the phone had to be up right the whole time, otherwise the Live View would disappear and switch back to street view.

Visual cues were found throughout the process while navigating in the area. They were big and easy to understand. The live view visual cue is illustrated in figure 19 below.

Figure 19. Google Map Live View with Visual Cue in Augmented Reality

However, while navigating with the Live View, the audio navigation was not clear enough. For example, if a street was long, there were no audio cues while walking on the street. Visually impaired users would not know if they are still on the right track or not, until they finished walking on the street and took the next turn.

While navigating, a wrong turn was made deliberately to check if the app detected it and would remind the user to turn right back to the correct direction. However, the app did not provide any cues to correct the path.

Figure 20. Google Map Live View, while walking parallel to the designated path

In figure 20, user was walking parallel to the original path, but no cues were provided to correct the path. In addition, some of the visual cues were very small. They may not be visible for visually impaired users.

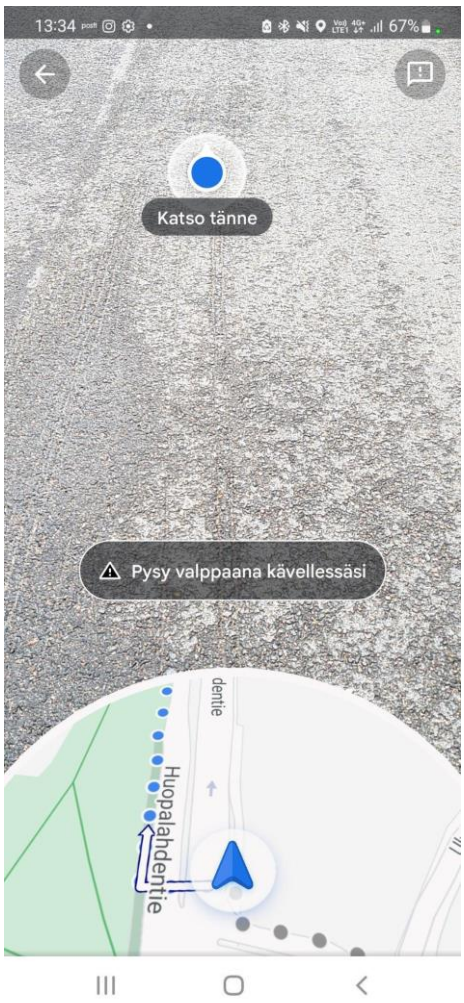


Figure 21. Google Map Live View visual cue “Look here”

In figure 21, the visual cue “Look here (in English)” was so small and not visible that visually impaired users could easily miss it.

In order to implement this feature to the app for visually impaired elderly, more audio cues should be given throughout the navigation process so that users can follow along the way without feeling lost. Visual cues on the app should be bigger. Immediate cues should be given if users make a wrong turn or go the wrong path.

TalkBack

TalkBack was switched on before opening Google Map, to imitate the scenario for a visually impaired user. After TalkBack was turned on, it was especially hard to interact with the mobile phone. TalkBack read everything on the screen, which was time consuming and annoying. It was impossible to locate the Google Map app with TalkBack because it was nested inside a folder. Although it kept saying “double click” to activate the apps, nothing happened even if the said action was made. In conclusion, TalkBack was hard to use and not responsive.

Google Voice Access

Then, Google Voice Access was tested. Google Map could not be opened with Voice Access by using the command “Open Google Map”, as the app’s name was actually “Map”. User had to know exactly what the name of the app was to open it.

When pairing it with Google Map, user had to click the search bar by hand before entering the address via speech recognition. It did not work when the user tried to navigate to the search bar with voice command on voice access.

Moreover, Google Voice Access worked perfectly while trying to carry out simple commands such as calling someone. However, while trying to type a SMS with speech recognition, it failed.

Also, users have to learn how to use Google Voice Access. As previously mentioned, words are being ingested and processed by the device in VUI. Users should learn what keywords should be used to dictate the app.

5.9 App prototype

First prototype of the app was made after taking accessibility into account.

Figure 22 shows how the app looked like:

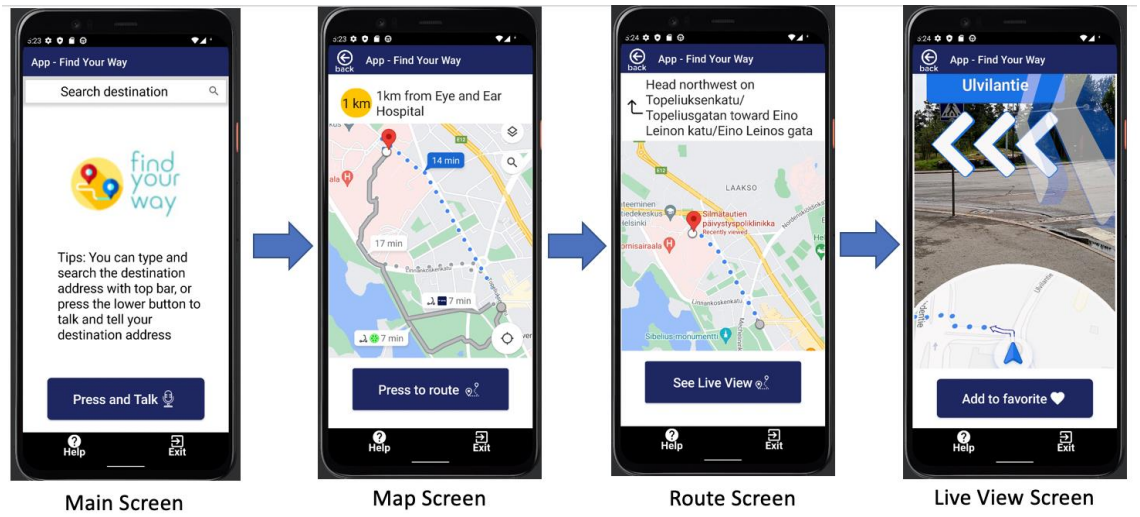


Figure 22. First app implementation with flow included.

The app flow was natural and all the UI elements were connected with each other. Users could navigate through different screens with ease. If they had any problem, they could click the help button. Below shows a help screen UI with information to help users navigate in the app.

Figure 23. Help screen on the app.

5.10 User testing

User testing is crucial in UX design process, as users feedback can generate ideas to improve the app.

User testing was conducted with the target user. In the beginning of user testing, user had been briefed about the purpose of this app. Since this was only a UX design for the app, logic for Google map and Google live view had not been included. To compensate for that, competitive apps such as Google Map and Google Live were used in the user testing. TalkBack was also turned on to provide an all-rounded experience for the visually impaired user.

User testing questions were given to the user:

- Can you name some competitors of this product?
- What features do you find most valuable and why?
- What do you think about the design?
- What prevents you from completing a task?
- What was your overall impression of this product?
- If you could change something in this product, what would it be and why?

The transcript was saved in Appendix III.

Then, the user mentioned she heard about Google Map. During the interview, she asked why she could not simply use Google Map, instead of this app. A brief discussion was carried out and a quick demo was done with the Google Map. She understood Google Map was not user-friendly for her. The icons were too small and not easy to find. She would not be able to complete any task with it.

In a nutshell, the overall experience of the user using the UX app was positive. Colour contrast was good. She liked the app flow, it was easy for her to use. While presenting the UI elements with TalkBack she said it was quite clear. She

could understand everything on the UI. What she valued most was the speech recognition element, because it was quite hard for her to read what was on the UI. She could see only the main buttons such as “Press to Talk”, “Go to Map”, “See Live View” and “Add to favourites”. But she said they were enough.

However, some of the UI elements were too small for her, such as the visual cues on the Google Live View, the top bar and bottom navigation bar.

What she did not like about the UX app was the TalkBack app. It was complicated to use and she found it hard to interact with the app. She found the Google Map difficult to use if she could only rely on the audio cues. While navigating with the map, she would not know which side was north sometimes. The app would not tell her if she was facing north or not. When she made a wrong turn, the app would not remind her either. The audio cues stopped abruptly sometimes. She had to double click to activate the audio cues. She found the Google Live View impossible to use, as the visual clues were too small for her eyes.

Lastly, she mentioned that if she could change the app, she would like to have a 2-way-interaction with the app via audio. It would be great if she could tell the app what she wanted to do, where she would go and the app would confirm if she was going in the right direction or not. If she made a wrong turn, it would tell her to turn right back.

6 Discussion

After the user testing, the app was revised based on the results. The font size of the main buttons is now 40sp and button size is now 400 x 100dp. Since the user said the UI elements sounded clear and was easy to comprehend on TalkBack, there was no need to adjust the Jetpack Compose accessibility elements. Figure 24 below showed the revised user interface of the app:

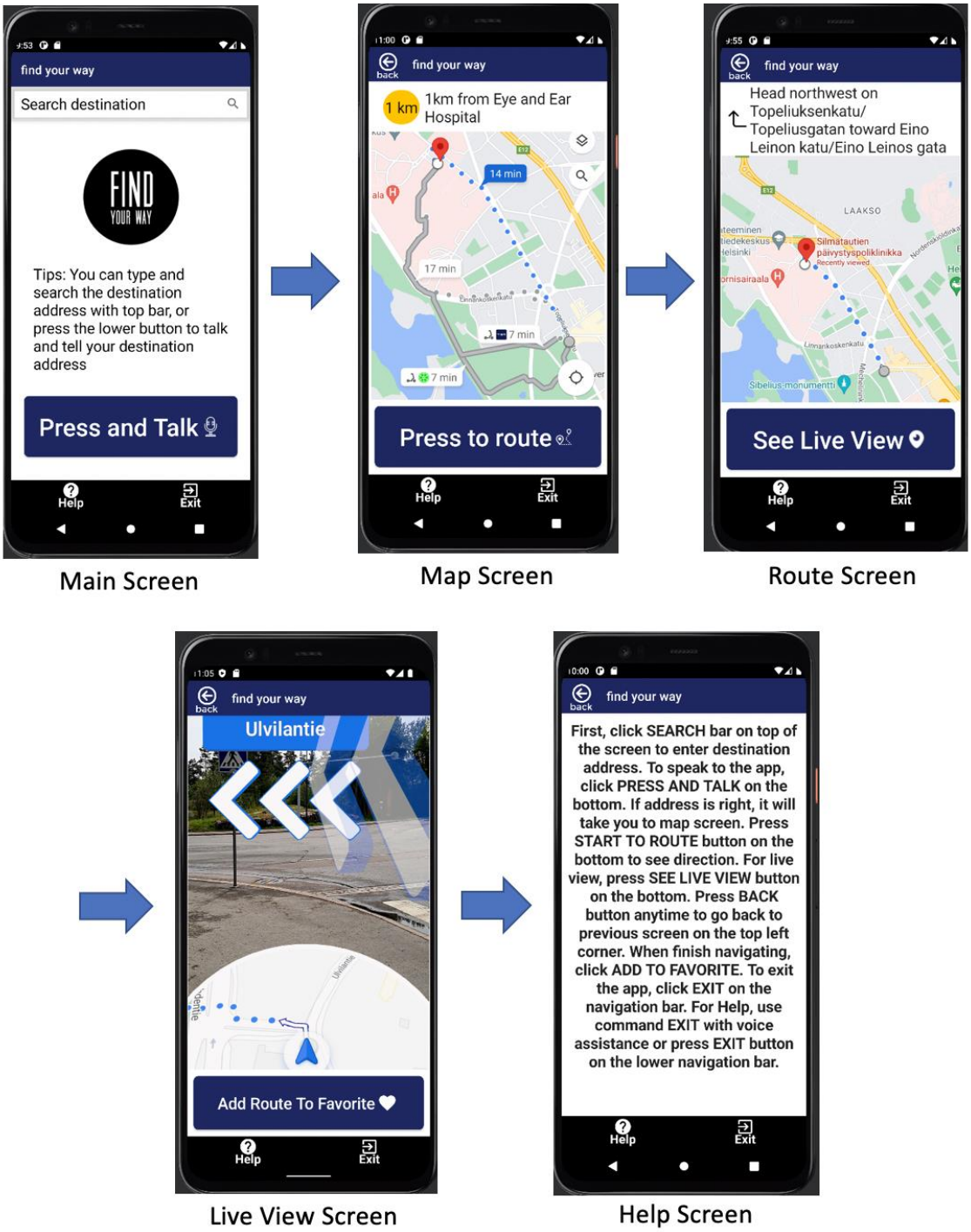


Figure 24. Revised user interface for the app

In addition, VUI dialog flow was updated to better serve her needs as she has to rely on voice recognition on this app. It is attached as Appendix 3 and as follows in figure 25:

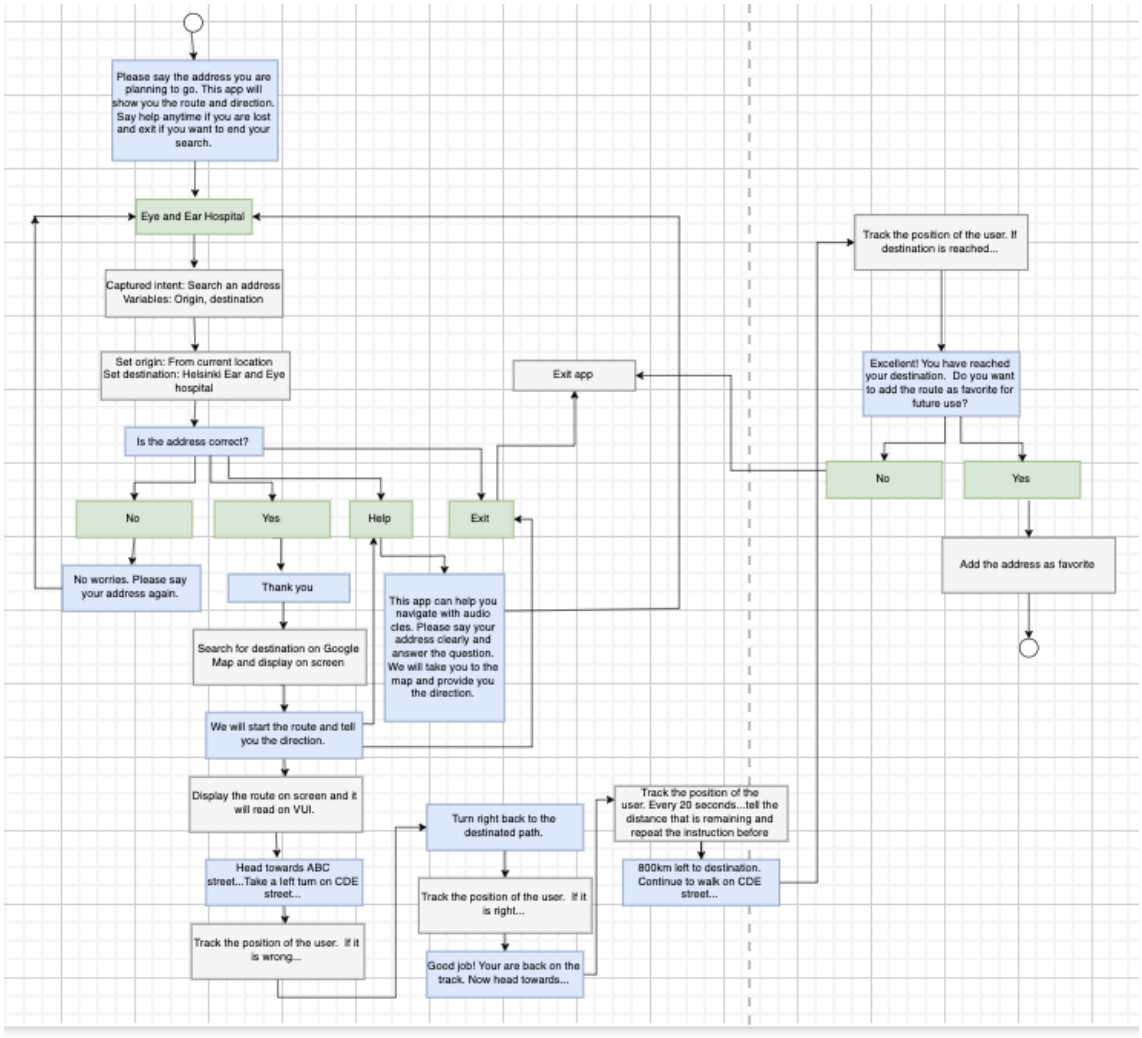


Figure 25. Revised dialog flow for the app's VUI.

Live View was removed in this VUI, as she could not use it during the user testing.

Since the user would like to have a better 2-way conversation with the voice recognition and a dialogue flow that includes more audio cues, in the new VUI, it would tell her what kind of options she has while using this app. The VUI will stay on during her entire journey. Apart from telling her where to go, it will also remind her if she has made a wrong turn and how to get back on the right track. Once she is back on the right track, it would turn her "Good job! You are back on

track…”. Every 20 seconds, the VUI will tell her how much distance is remaining and repeat the direction given before. Once she arrives, it will tell her “Excellent! You have reached your destination”.

Moreover, the help function has been added to the VUI, as elderly users need helping clues sometimes due to cognition problems. Exit function is also implemented because elderly get irritated with technology easily and they just want to exit the app immediately.

Lastly, with this new VUI, user can simply talk to the app and receive information. Feedbacks like “Good Job! “You are back on track” or “You have 200 metres remaining in your journey” can reassure she is not lost and she is on the right path. If she makes a wrong turn, the app will keep giving her hints until she is back to the right track.

7 Conclusion

As previously mentioned, elderly users are often neglected in the design process of smartphone applications. Their needs and requirements are not taken care of. This study discussed design process guidelines to develop a UX design for the visually impaired elderly.

Throughout the design process of the project app, accessibility was taken into account and implemented into a project app that had high usability and fitted the needs of the target user group. Accessibility scanner was used to check the contrast ratio of the app, as a low ratio means the visually impaired users would not be able to see the elements clearly. Then UI elements like font size, button size, visual cues were considered. The app was also tested with TalkBack, to check if all the elements were labelled clearly and understandable for visually impaired users. With user interview and user testing, the app prototypes and its final product were revised several times with better UI elements that better fit the needs of the target user group.

However, there is no one single solution to UX design. Designers should research and investigate what the target users' needs are, what pains them and what kind of functions would serve their needs. Designers should have empathy towards the users and think like them. Focus groups, surveys, interviews and user testing should be carried out throughout the process, they should not wait until the product is finished for user testing. All these processes allow designers to bond with the users, understand what they need and make an app which they find useful and satisfied about.

Lastly, with all of the above steps included in the design process, designers can produce a clean and responsive design that is pleasing, engaging and easy to use to maximise user experience and meet their goals.

References

1. World Population Ageing 2019: Highlights, pp. 1, 2019.

2. Global Mobile Consumer Survey 2019 Nordic Cut, pp.6, 2019.

3. Q. Li and Y. Luximon, “Older adults' use of the mobile device: usability challenges while navigating various interfaces,” Behaviour & Information Technology, vol. 39, no. 1, pp. 1–25, 2019.

4. F. J. García-Peñalvo, M. Á. Conde, and V. Matellán-Olivera, “Mobile apps for older users–the development of a mobile apps repository for older people,” in Proceedings of the International Conference on Learning and Collaboration Technologies, pp. 117–126, Heraklion, Crete, Greece, June 2014.

5. H. Hassan and M. H. N. Md Nasir, “The use of mobile phones by older adults,” ACM SIGACCESS - Accessibility and Computing, vol. 92, no. 92, pp. 11–16, 2008.

6. R. Leung, L. Findlater, J. McGrenere, P. Graf, and J. Yang, “Multi-layered interfaces to improve older adults' initial learnability of mobile applications,” ACM Transactions on Accessible Computing (ACCESS), vol. 3, no. 1, 2010.

7. Jetpack compose [Internet]. Android Developers. [cited 2022 May 27]. Available from: https://developer.android.com/jetpack/compose?authuser=1

8. Lazarovich MN. UX vs. UI design: What’s the difference? [Internet]. Elementor. 2021 [cited 2022 May 7]. Available from: https://elementor.com/blog/ux-vs-ui/?utm_source=google&utm_medium=cpc&utm_campaign=13060922353&utm_term=&gclid=EAIaIQobChMI77iozpqv9wIVxAh7Ch2ldgkEEAAYAyAAEgL_MfD_BwE

9. Best practices in user experience (ux) design: Design compelling user experiences to wow your customers [White paper]. Forrester. 2009 [cited 2022 May 7]. Available from: https://www.forrester.com/report/Best-Practices-In-User-Experience-UX-Design/RES54101

10. Cardello J. UX design process: a simple (but complete) guide [Internet]. Webflow. 2021 [cited 2022 May 7]. Available from: https://webflow.com/blog/ux-design-process?utm_source=google&utm_medium=search&utm_campaign=gen

eral-paid-workhorse&utm_term=keyword-targeting&utm_content=dynamic-search-ads-t1&gclid=EAIaIQobChMIrs7G_qGv9wIVCW8YCh0zEgZVEAAYAyAAEgI1xPD_BwE

11. Dam RF, Siang TY. What is empathy and why is it so important in design thinking? [Internet]. The Interaction Design Foundation. Interaction Design Foundation; 2023 [cited 2023 Jan 28]. Available from: https://www.interaction-design.org/literature/article/design-thinking-getting-started-with-empathy

12. Stull E. UX fundamentals for non-UX professionals: User experience principles for managers, writers, designers, and developers [Internet]. 1st ed. New York, NY: APRESS; 2018 [cited 2022 May 7]. Available from: https://learning.oreilly.com/library/view/ux-fundamentals-for/9781484238110/html/464548_1_En_21_Chapter.xhtml

13. Urse A. The curse of the UX design process [Internet]. Research Loop - Everything about User Research and UX Design. 2021 [cited 2022 May 7]. Available from: https://researchloop.medium.com/the-curse-of-the-ux-design-process-5d1562bde555

14. Adhav R. UX research and market research: Know the difference [Internet]. designial. 2020 [cited 2022 May 7]. Available from: https://www.designial.com/ux-research-and-market-research-know-the-difference/

15. Storyboards help visualize UX ideas [Internet]. Nielsen Norman Group. [cited 2022 May 7]. Available from: https://www.nngroup.com/articles/storyboards-visualize-ideas/

16. Information architecture guide for UX architects & designers [Internet]. Ideas. 2020 [cited 2022 May 7]. Available from: https://xd.adobe.com/ideas/process/information-architecture/information-ux-architect/

17. What is Usability Testing? [Internet]. The Interaction Design Foundation. [cited 2022 May 9]. Available from: https://www.interaction-design.org/literature/topics/usability-testing

18. Harris P. What Is User Interface Design? New York, NY: PowerKids Press; 2017.

19. Indeed Editorial Team. What is a user interface? Definition, types and user interface example [Internet]. Indeed Career Guide. 2020. Available from: https://www.indeed.com/career-advice/career-development/user-interface

20. Gilbert RM. Inclusive design for a digital world: Designing with accessibility in mind. 1st ed. Berlin, Germany: APress; 2019.

21. Let’s make services accessible [Internet]. Web Accessibility Requirements. [cited 2022 Nov 29]. Available from: http://webaccessibility.fi

22. UX accessibility design best practices [Internet]. Ideas. 2021 [cited 2022 May 7]. Available from: https://xd.adobe.com/ideas/principles/web-design/what-is-accessible-design/

23. Ageing and health [Internet]. Who.int. [cited 2022 May 7]. Available from: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

24. Falahaty K, Cheong LS, Isa MBH. Disability among elderly people with visual impairment in two welfare homes in Malaysia. Biomedical and Pharmacology Journal [Internet]. 2015 [cited 2022 May 7];8(2):1369–82. Available from: https://biomedpharmajournal.org/vol8no2/disability-among-elderly-people-with-visual-impairment-in-two-welfare-homes-in-malaysia/

25. Assistant Secretary for Public Affairs. Web usability and aging. 2013 [cited 2022 May 7]; Available from: https://www.usability.gov/get-involved/blog/2008/12/aging-and-usability.html

26. F. J. García-Peñalvo, M. Á. Conde, and V. Matellán-Olivera, “Mobile apps for older users–the development of a mobile apps repository for older people,” Proceedings of the International Conference on Learning and Collaboration Technologies, pp. 117–126, 2014.

27. H. Hassan and M. H. N. Md Nasir, “The use of mobile phones by older adults,” ACM SIGACCESS - Accessibility and Computing, vol. 92, no. 92, pp. 11–16, 2008.

28. R. Leung, L. Findlater, J. McGrenere, P. Graf, and J. Yang, “Multi-layered interfaces to improve older adults' initial learnability of mobile applications,” ACM Transactions on Accessible Computing (ACCESS), vol. 3, no. 1, 2010.

29. Affinity diagramming: Collaboratively sort UX findings & design ideas [Internet]. Nielsen Norman Group. [cited 2022 May 7]. Available from: https://www.nngroup.com/articles/affinity-diagram/

30. Pacheco-Cutillas M, Edgar DF, Sahraie A. Acquired colour vision defects in glaucoma-their detection and clinical significance. Br J Ophthalmol

[Internet]. 1999 [cited 2022 May 9];83(12):1396–402. Available from: https://bjo.bmj.com/content/83/12/1396

31. Shaffer J. 5 tips on designing colorblind-friendly visualizations [Internet]. Tableau. [cited 2022 May 9]. Available from: https://www.tableau.com/about/blog/examining-data-viz-rules-dont-use-red-green-together

32. Designing with visual impairments in mind | icapps blog [Internet]. icapps. [cited 2022 Nov 29]. Available from: https://icapps.com/blog/design-visual-impairments

33. Don’t use color alone to convey information (colorblind) - Access Guide [Internet]. Accessguide.io. [cited 2022 May 9]. Available from: https://www.accessguide.io/guide/colorblind

34. WebAIM. WebAIM: Contrast and Color Accessibility - Understanding WCAG 2 Contrast and Color Requirements [Internet]. webaim.org. 2021. Available from: https://webaim.org/articles/contrast/

35. Accessible fonts: How to choose a font for web accessibility [Internet]. Siteimprove. [cited 2022 May 7]. Available from: https://www.siteimprove.com/glossary/accessible-fonts/

36. Apple Inc. Text size and weight - Accessibility - human interface guidelines - apple developer [Internet]. Apple.com. [cited 2022 May 7]. Available from: https://developer.apple.com/design/human-interface-guidelines/accessibility/overview/text-size-and-weight/

37. Let users control font size [Internet]. Nielsen Norman Group. [cited 2022 May 9]. Available from: https://www.nngroup.com/articles/let-users-control-font-size/

38. Wilson C. What does the truncation of the text mean? [Internet]. truncations.net. [cited 2022 May 7]. Available from: https://www.truncations.net/what-does-the-truncation-of-the-text-mean

39. Material Design [Internet]. Material Design. [cited 2022 May 7]. Available from: https://material.io/design/usability/accessibility.html

40. Semantics and TalkBack with jetpack compose · Bryan Herbst [Internet]. Bryanherbst.com. [cited 2022 May 7]. Available from: https://bryanherbst.com/2020/11/03/compose-semantics-talkback/

41. Control your Android device with Switch Access [Internet]. Android Developers. [cited 2022 May 7]. Available from: https://support.google.com/accessibility/android/answer/6122836

42. W3C Web Accessibility Initiative (WAI). About W3C WAI [Internet]. Web Accessibility Initiative (WAI). 2022 [cited 2022 May 27]. Available from: https://www.w3.org/WAI/about/

43. Baker J. Voice user interfaces (VUI) — the ultimate designer's guide [Internet]. Muzli - Design Inspiration. 2018 [cited 2022 May 9]. Available from: https://medium.muz.li/voice-user-interfaces-vui-the-ultimate-designers-guide-8756cb2578a1

44. Goossens F. Designing a VUI — voice user interface [Internet]. Multimodal UX. 2018 [cited 2022 May 9]. Available from: https://medium.com/voice-experience-design/designing-a-vui-voice-user-interface-c0b3b9b57ace

45. Kohlbacher F, Herstatt C. The silver market phenomenon: Business opportunities in an era of demographic change [Internet]. Kohlbacher F, Herstatt C, editors. Berlin, Germany: Springer; 2008. Available from: https://books.google.at/books?id=JfJDAAAAQBAJ

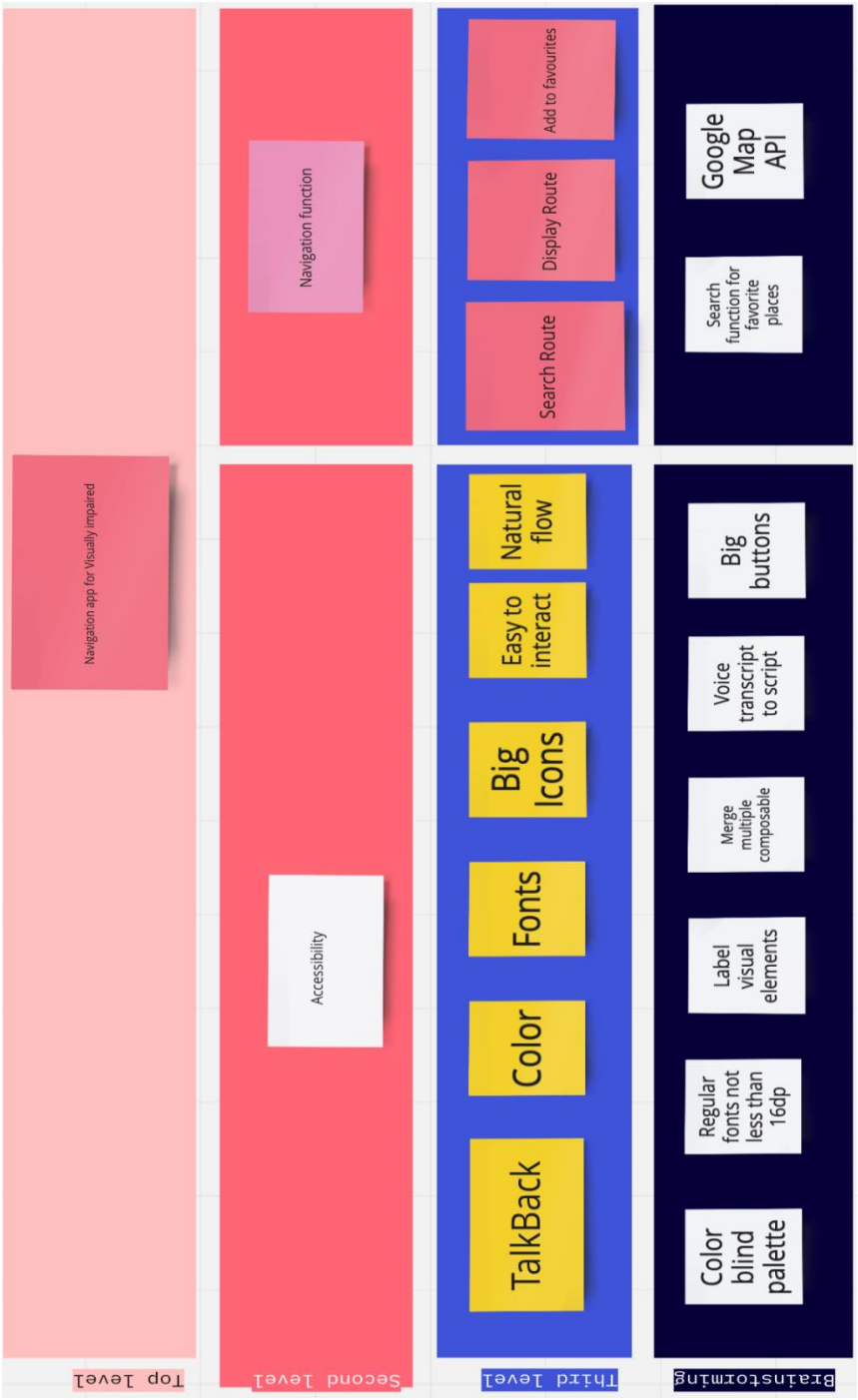
46. Hanks B. Accessibility design: How to master interface design for older adults [Internet]. Screen Recording Software for Mac & PC | CloudApp. Screen Recorder Software by CloudApp; 2020 [cited 2022 May 27]. Available from: https://www.getcloudapp.com/blog/accessibility-design/

47. Use Live View on Google maps [Internet]. Google.com. [cited 2022 May 27]. Available from: https://support.google.com/maps/answer/9332056?hl=en&co=GENIE.Platform%3DAndroid

48. Semantics in Compose [Internet]. Android Developers. [cited 2022 May 9]. Available from: https://developer.android.com/jetpack/compose/semantics

49. Accessibility in Compose [Internet]. Android Developers. [cited 2022 May 7]. Available from: https://developer.android.com/jetpack/compose/accessibility

Affinity Diagram



User Interview

Interviewee: Tarja (TR), Tepa (TK), Carina (CK)

Date: 7 May 2022

What kind of apps are you using?
TR: Whatsapp, HSL Route Guide, Facebook, Camera,
TK: None
CK: Whatsapp, Facebook, Camera, Video

Why are you using those apps?
TR: I connect with my friends with Whatsapp because it is easy to use. I use the HSL Route Guide because it solves my problem. It can check the bus schedule for me. Camera and Video I love because I want to share what is happening in my life with my friends through them.
TK: None. I do not own a SmartPhone. I tried to use it but it is too complicated and I cannot see well. I own a Doro mobile (not a smartphone).
CK: I am from Russia and I want to connect with my friends overseas with these apps. They are quite easy to use, unlike the other apps.

What frustrates you in an app?
TR: Complicated features. Too small touchable icons. My fingers are too big for them. It is impossible to type the right alphabets with the phone's keyboard.
TK: Small fonts and icons. Bad colour contrast. I have glaucoma.
CK: Small touchable icons. Online banking is getting so complicated.

Any apps you would like to have or can help you in life?
TR: None. I do not want more apps.
TK: Maybe Google Map with audio aid. I want to use it even if I do not see well.
CK: I am happy with the apps I can use now.

User Testing

User: Tepa

Date: 30 May 2022

Question 1: Can you name some competitors of this product?
Google Map

Question 2: What features do you find most valuable and why?
The speech recognition feature. I can press the button and start talking to the app instead of finding the small icons/views to type in.

Question 3: What do you think about the design?
The buttons and font size are quite big. There is good colour contrast. Although the main button in the bottom of the screen can be even bigger as I rely on it throughout the app.
The app flow is quite easy and even as old as I am can understand. Google Map can be improved (AI?). Sometimes I do not know which side is north. How can I follow the route by just listening to it? I cannot see what is on the map anyhow. The voice of the Google Map stops abruptly. Sometimes I have to press the screen to hear the clues for map direction. I cannot use the Live View at all. As the clues are too small and complicated for me.

Question 4: What prevents you from completing a task?
Missing compass direction of the map. Not knowing which side is north. I need to know where the main button is, as I cannot see it so clearly.

Question 5: What was your overall impression of this product?
The interface is clear. The main buttons are on the smaller side for me. I do not understand why there is a need for this app because there is Google Map in the market already.

Question 6: If you could change something in this product, what would it be and why?

Voice recognition function. It would be great if I could talk while navigating through the app. I can say commands like “help” anytime. The app would be able to tell me which direction I am going. If I make a wrong turn, it would tell me to turn right back.

TalkBack is hard to use. I have to swipe all the time so that it will keep reading from one UI element to another UI element. It would be nice if I could have a two-way interaction with this app. The app could tell me the direction, confirm my commands, correct the path if I go the wrong way.

VUI Dialogue Flow