Bachelor's thesis

Information and Communications Technology

2021

Laanaya Salim

USABILITY AND USER TESTING ON THE MOBILE APPLICATION "DIGIHELPPARI"

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor's degree in Information and Communications Technology

2020 | 26

Salim Laanaya

USABILITY AND USER TESTING ON THE MOBILE APPLICATION "DIGIHELPPARI"

The importance of application testing along with the consequences that defects have on the final product has played a huge role in the development process. With the constant increase of mobile applications in today's large market, it has become an important process to ensure software quality. Thus, companies have taken the initiative to increase their attention to detail, leaning more on different application testing methods. The objectives of this thesis are to show the usability and user testing processes in an application, using the heuristic method.

The research and testing in this thesis were based on an application created for the ASPA association by theFIRMA, a project-based learning environment of Turku University of Applied Sciences. The development process was categorized in the following order: Planning, identifying testing types, test case, manual testing/UI and UX testing, usability testing, performance testing, security and compliance testing, device tests, and summary. The study began with conducting research into usability and user testing through a heuristic evaluation approach. This method gives the proper insight into the interface and its compliance with recognized usability principles. The heuristic evaluation approach provides specific information on which exact features and elements need to be changed, improved, or discarded and thus helps understand if an application's future is obsolete for a user.

The results of this thesis were a fully conducted mobile application test using heuristic evaluation. By analyzing the application and its feedback, a fully working application was created. These findings can also aid in other testing methodologies used in other mobile application testing.

KEYWORDS:

Heuristic Evaluation, UI, UX, manual testing, usability testing, user testing, mobile application

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikan kandidaatin tutkinto

2020 | 26

Salim Laanaya

KÄYTETTÄVYYS JA KÄYTTÄJÄN TESTAUS MOBIILISOVELLUKSEEN ”DIGIHELPPARI”

Sovellustestauksen merkityksellä sekä virheiden seurauksilla lopputuotteella on ollut valtava rooli kehitysprosessissa. Kun mobiilisovellukset kasvavat jatkuvasti nykypäivän suurilla markkinoilla, siitä on tullut tärkeä prosessi ohjelmistojen laadun varmistamiseksi. Siksi yritykset ovat tehneet aloitteen lisätä huomiota yksityiskohtiin tukeutuen enemmän erilaisiin sovellusten testausmenetelmiin. Tämän opinnäytetyön tavoitteena on osoittaa sovelluksen käytettävyys ja käyttäjien testausprosessit.

Tämän opinnäytetyön tutkimus ja testaus perustuivat Turun ammattikorkeakoulun projektipohjaisen, theFIRMA:n luomaan sovellukseen ASPA:n asukkaille. Kehitysprosessi luokiteltiin seuraavaan järjestykseen: Suunnittelu, testaustyyppien tunnistaminen, testitapausten ja komentosarjojen suunnittelu, manuaalinen testaus / käyttöliittymien ja käyttöliittymien testaus, käytettävyystestaus, suorituskyvyn testaus, tietoturva- ja vaatimustenmukaisuuden testaus, laitetestit ja yhteenveto. Tutkimus alkoi tutkimalla käytettävyyttä ja käyttäjien testausta heuristisen arviointimenetelmän avulla. Tämä menetelmä antaa oikean käsityksen käyttöliittymästä ja sen yhteensopivuudesta tunnustettujen käytettävyysperiaatteiden kanssa. Heuristisen arviointimenetelmän avulla se tarjoaa tarkkaa tietoa siitä, mitä tarkkoja ominaisuuksia ja elementtejä on muutettava, parannettava tai hylättävä, ja auttaa siten ymmärtämään, onko sovelluksen tulevaisuus vanhentunut käyttäjälle.

Tämän tutkielman tulokset olivat täysin suoritettu mobiilisovellustesti käyttäen heuristista arviointia. Analysoimalla sovellusta ja sen palautetta luotiin täysin toimiva sovellus. Nämä havainnot voivat auttaa myös muissa testausmenetelmissä, joita käytetään muissa mobiilisovellusten testauksissa.

ASIASANAT:

Heuristinen arviointi, Käyttöliittymä (UI), Käyttäjäkokemus (UX), Sovellustestaus, manuaalinen testaus, käytettävyystestaus, käyttäjien testaus, mobiilisovellus

CONTENTS

FIGURES

PICTURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
ARA	Housing Finance and Development Center
IT	Information Technology
OP	Operating System
RAM	Random Access Memory
SRS	Software Requirements Specification
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

1 INTRODUCTION

The idea and implementation of this thesis were commissioned by the ICT project office called theFIRMA. The application was created for the organization ASPA. TheFIRMA is a public facility created by and located at Turku University of Applied Sciences, Turku, Finland, while ASPA has its locations spread all over Finland and its headquarters located in Tampere, Finland.

TheFIRMA maintains its focus on helping students in learning and improving their skills by being involved in real-world projects, allowing for more possible job opportunities and experiences (theFIRMA,n.d). TheFIRMA also hires students for internships along with providing students a specialization track that they are interested in, such as project management, website development, etc.

These projects are real customer-based and provide several different opportunities in different sectors of Information Technology (IT). The following services are provided by theFIRMA (theFIRMA, n.d):

- Software development: The development and designing of software, applications, games, etc. According to the customers' needs.
- Website development: The production of websites from simple design to full packaged updates. Also, the including of graphical design and technical implementation.
- Testing: The planning and implementation of testing and debugging for programs, applications, websites, games. Etc.
- Marketing material: Using the guidelines of graphical design the development of logos, advertising posters, brochures, social media advertisements, along with other marketing material according to the customer demands.
- Requirements Specification: The creation of proper needed documents and requirements of the software in question.
- Technology Camps: Organizing technology camps allowing children and teenagers to become familiar with programming.

On the other hand, the ASPA organization was founded by disability organizations in 1995, has branches in 68 different locations around Finland and is responsible for providing customers with disabilities home and services based on different individual

needs. Their work ensures that everyone has the opportunity to live independently. The following services were introduced by ASPA as of the year 2019: Renting and building affordable housing for those who need daily support in their lives. They provide homes to about 200 new residents each year, employing 400 employees, and the operations are supported by Veikkaus' income and ARA grant. The total customer amount provided by their housing is currently 1,335. Also participating in this project as the producer was Gamu ry. Gamu ry is a small and flexible organization whose mission is to support the finding of new methods for different organizations.

The residents of ASPA are those who cannot live completely on their own due to mental health reasons, but the aim is to help them achieve an individual life regardless. In doing so, their daily tasks such as calculating income, everyday use of the calendar, and overall personal tasks have become unrecognized and hard to do. Their methods have been orientated more towards handwritten techniques. These methods are not only inconvenient but also time-consuming and untrustworthy. Thus, the idea came about creating an application called "Digihelppari" for the community residing at ASPA centers that would help in these matters. The application would guide them in their everyday tasks and at the same time provide them with a fun and innovative approach using modern-day technology. The application which started with the creation in Android consists of five sections which are: home page, calendar, financial information, notes/reminders, and useful links. Using these, the customer receives the necessary information to help guide them better in daily situations, making the application a more efficient, reliable, and cost-effective solution. The objectives of this thesis are to conduct a heuristic evaluation test and investigate how it plays a role in the final product. It is expected that the application will have a more fluent user interface for its customers to use.

Nowadays, mobile solutions have taken over the market. Smartphones especially harvest the greatest number of shares. According to the data released by International data corporation, as of June 3, 2020, the total smartphone shipments globally was a total of 1.2 billion smartphone shipments. Also, taking into consideration that due to COVID-19 there was an 11.9% decline in sales. Regarding portable devices, most refer to, tablets, laptops, phones, watches, etc. These are all handheld devices that provide fast delivery and require a faster response time. With the vast amount of growth in mobile devices and their applications, testing has become more demanding and complex. Hence, it has become ever more important to fully test the software. (Canalys, 2020)

2 BACKGROUND

2.1 Web-Based Software Testing vs. Mobile Application Testing

Many key factors need to be taken into consideration when discussing software testing and mobile testing. The main key difference between the two is that software testing refers to web-based or desktop applications whereas mobile testing is mobile-based. Often the two are confused with each other, but when in fact they differ from each other quite considerably.

Software testing is an essential and valued part of software engineering. It is a process of comparing the actual output to the expected output and is essential in the development process. Both tests are similar in that they help to identify errors, missing requirements, functionalities, etc. regarding what is required of the actual system to be defect-free. The process involves the execution of system components to determine one or more points of interest. The process of either web-based software testing or mobile application is indeed structurally the same and holds many similar key factors, but their differences are crucial.

It is important to note the following differences between web-based software testing and mobile testing (Nec Software, 2021):

- Mobile applications are developed for a certain type of operating system, usually either for Android or IOS systems. This being mentioned, application must fill the requirements of for example regulated processing performance, screen rotation, and display size, swipe area/click area, battery life, etc. Most application development software provides these features already included, making this process easy, whereas web-based software testing is orientated towards operating systems such as Mac, Windows, or Linux.
- Mobile devices are naturally smaller and depending on the mobile device can be in different display sizes. For example, iPhone 5 has a 4” display, whereas the iPhone 6 has a 4.7”. Desktop and laptop browsers come with a predefined ratio.
- Mobile phones have much less capable of storage and RAM (Random access memory) when compared to computers.
- Mobile phones use and support different network coverage from 3G, 4G, to Normal Wi-fi. Some might not have access to the internet at all. The following

characteristics must be taken into consideration: is the user going to need this application at times without internet? Or will the user need internet at all given times?

- Mobile applications do not have the capability of choosing between operating systems such as Android, Windows OP, IOS, etc. that come with different hardware limitations. Computers usually follow the same basic logic when it comes to browsers. Only a limited number of differences exist between different browsers. For example, Google Chrome is not very different compared to Mozilla Firefox or Internet Explorer. The layout stays mostly consistent on computers.
- If the given application is directed towards a certain audience, it should meet the needs of supporting multiple inputs and/or outputs like voice, sound, keyboard, fingerprint sign-in, text to speech, etc.

2.2 Heuristic usability evaluation

The method used on this application was heuristic usability evaluation. The process is used in assuring that an application is user-friendly in terms of ease of use and simplicity. Overall, usability testing guides the application in finding and identifying the bugs and errors to improve the customer's overall experience. Heuristic usability evaluation is a usability engineering method of finding the problems and concerns in a user interface design. This evaluation method being the most used for user interface design was first developed by Jakob Nielsen and Rolf Molich in 1990. Both had several years of experience in teaching and consulting regarding usability engineering and this gave them the idea of improving usability inspection in the wide field of human-computer interaction. The final Heuristic evaluation was then released by Nielsen in 1994 in his book Usability Engineering.

The published Heuristic rules are of the following: (Nielsen, 1994):

1. **Visibility of system status**:
   The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time.
2. **Match between system and the real world**:
   The system should speak the user's language, with words, phrases, and

concepts familiar to the user, rather than system-oriented terms, follow real-world conventions, and make information appear in a natural and logical order.

3. **User control and freedom**:
   Users often choose system functions by mistake and will need a marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. There is a clear and visible way to exit the current interaction, for example a “Cancel button”.
4. **Consistency and standards**:
   Users should not have to wonder whether different words, situations, or actions mean the same thing. Follows the consistency within a single intended product or a family of products.
5. **Error prevention**:
   Even better than good error messages is a careful design that prevents a problem from occurring in the first place. Error prevention can either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall**:
   Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use**:
   Accelerators—unseen by the novice user—may often speed up the interaction for the expert user so that the system can cater to both inexperienced and experienced users. Allows for both experienced and inexperienced users to use the system or application.
8. **Aesthetic and minimalist design**:
   Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors**:
   Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. **Help and documentation**:
    Even though it is better if the system can be used without documentation, it

may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

The heuristic evaluation usually consists of several different evaluators who examine the interface and give feedback and judge the usability based on the heuristic rules. Although the heuristic evaluation can be conducted individually, it becomes very hard for one to spot all the usability problems in the given interface making it a non-reliable solution. Usually, this method requires experience from many different evaluators to improve the effectiveness of the method and making it a reliable solution. If an evaluation is large or complicated it also might become necessary to split it into sections, as presented in Jakob Nielsen's case study, where 19 evaluators were used to find 16 usability problems in a voice response system that allowed customers of the bank to access their accounts. (Nielsen, 1994) This study provided evidence that there was no correlation between the evaluators and the usability problems they found. An experienced evaluator was able to point out many of the usability problems but had missed even some of the easier ones. On the other hand, a less experienced evaluator did the opposite by finding all the easier problems and missing the harder ones. This shows also that every test needs to take into consideration its test audience, whether they are unsuccessful or successful testers because it plays a key role in the outcome. As mentioned earlier, the experience from many different evaluators (in most cases above 3) has guaranteed to show an increased percentage in the number of problems found, shown in Figure 1. It becomes very difficult to address all problems when the number of evaluators is less than 3.

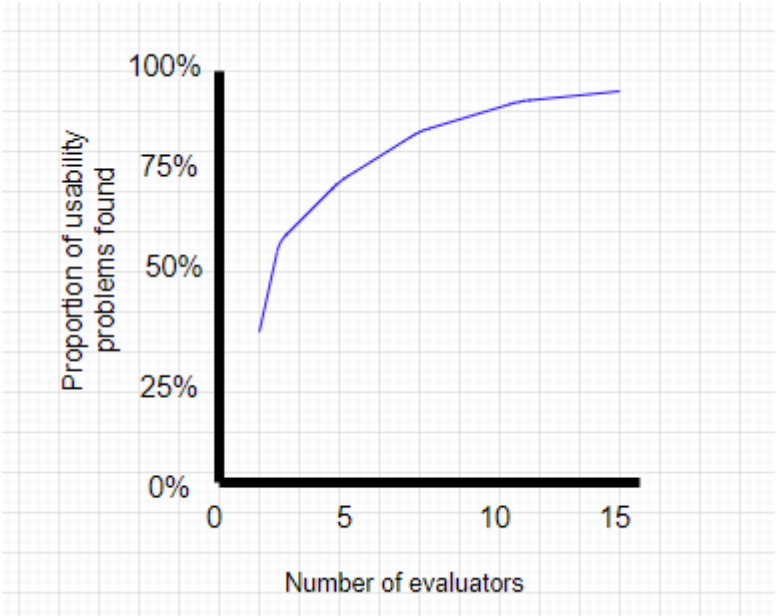


Figure 1. Illustration showing the correlation in the number of evaluators to the percent of problems found. (Nielsen, 1994).

2.3 Other evaluation types

The research of test methods is a very important step in every application development process. It is crucial to know and understand different test types as to what they do and how they can be used. Provided are some examples of other evaluation types (Gilmore, 2019):

Compatibility testing - Compatibility testing is a type of testing that is non-functional and checks that the application will work on a variety of operating systems, networks, versions, devices, and hardware. These tests focus on providing the customer with a working product that functions and reacts in every scenario. The application could react differently under some circumstances all dependent on what browsers, OP's, devices, etc. are being used.

Functional testing – Functional testing concentrates on ensuring that the application can function properly according to its customer's needs. These functionality tests can be for example, checking that the user is sign-in and out functions, push notifications, installations, upgrades, etc.

Usability testing – Usability testing ensures that the application or software is user-friendly according to the users in question. This type of testing allows for a better customer experience that is always looking for ways to improve the experience and error-

free. Some examples would be to check for, application response time, layout design, configuration, etc.

Performance and load testing - Performance and load testing is a test that puts the application under a certain workload amount that checks how well it can perform. In usual cases, if the application goes through an overload where it crashes or shuts down it shows its unsuitability to handle a certain workload.

Security testing - Security testing is testing conducted on the software that checks that it has the proper confidentiality, authenticity, and integrity. Usually, this can be checked through different application behavior cases and this testing shows how the program would handle these cases when it comes to security.

3 APPLICATION DEVELOPMENT

3.1 Application specifications of requirements

Throughout the process of creating an application, it is first necessary to create an application specification of requirements. These requirements are called software requirement specification (SRS) which is the product idea written on paper. Usually addressed in that document is the answering of some of the following questions: What needs your solution will cover, what future it needs and how it will work, what load amount should it resist, what is the target audience, when should the product be delivered, Etc.

It was discussed by the commissioner of this thesis that the planning and implementation of the testing would be freely chosen based on what was most ideal for the users.

The application in this case was built on top of a native web application. This would ensure that the application would operate on different platforms and implementing updates would be easier. In creating the application, it was highly important to specifically address each section regarding the end-user. As the users, in this case, are individuals with disabilities it was highly important to pay attention to color detail, text font, text size, along with the healthcare professionals that would aid in using the applications. There were four main users of the application as follows (theFIRMA, 2019):

1. User

The main user of the application. The main user can use the application according to their own needs, several times a day if needed.

2. User assistant

The user assistant is assigned by the administrator and is given the proper access rights as needed. This assistant can be a support person, relative, friend, etc. The assistant is responsible to monitor, help, and control the main user's performance in each everyday task.

3. Spokesperson

The spokesperson is an entity or organization that would organize events and communicate through the application about their events, courses, etc. The application

administrators would be the ones to decide whether the notifications would be received from the spokesperson.

4. Administrator

The administrator holds and maintains the application's full usage rights. Administrators run in the background and do not affect the application users. They are responsible for the proper updates, giving user rights, and fixes regard the application.

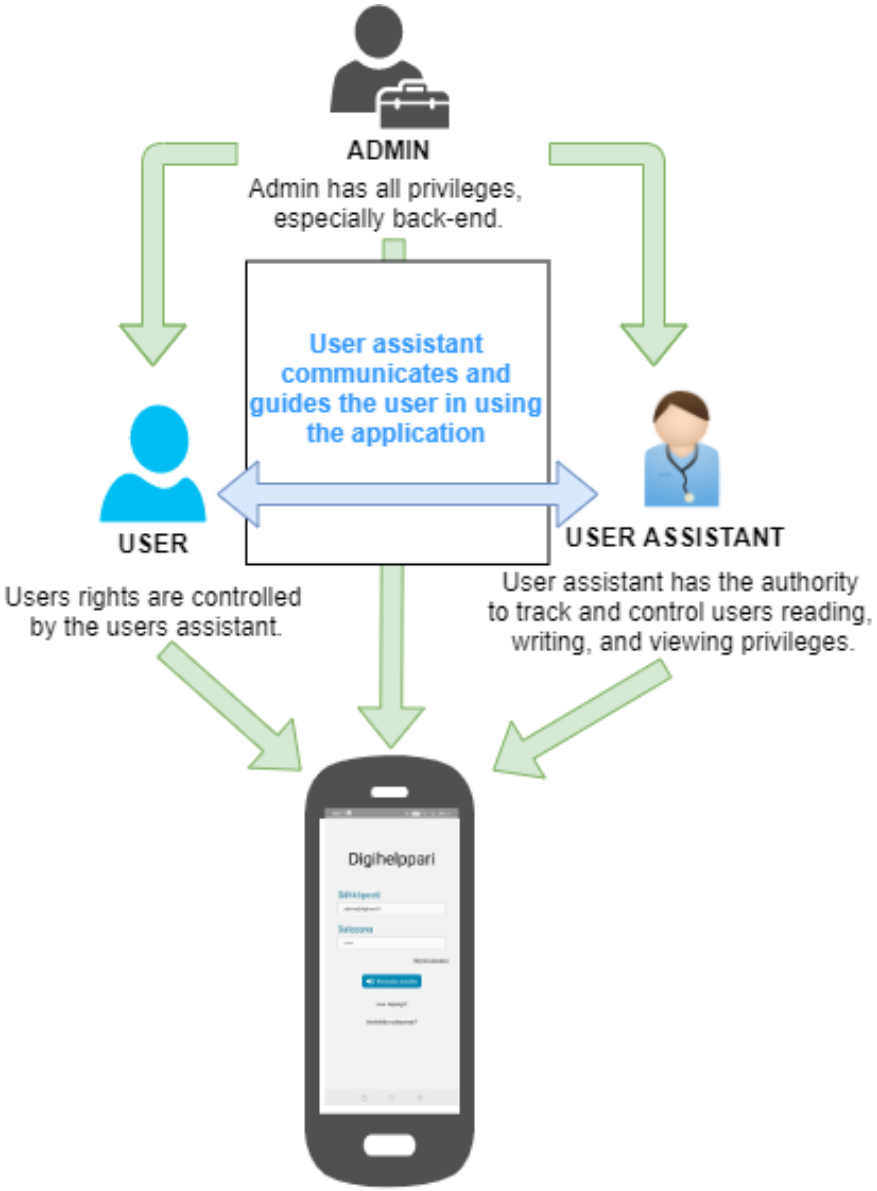


Figure 2. Diagram showing how the application works.

3.2 Test planning phase

The testing planning phases were an important part of the development of the application, so it was necessary to implement it correctly. The initial intention of the testing was to first run through two tests; one being done with the first test group, while the second test would be done in person with the intended audience (ASPA) for the application. But, because of covid-19 and its restrictions, it was decided that it would be more suitable to run the tests only with random test groups while pushing the test for the intended audience for a later date. As a result, the intended audience was not recorded in this thesis.

The tests were divided into three different test runs/phases, to provide valid confirmation of each test. The first two tests were conducted by the same group of three individuals, while the last test phase being a different group. The first and second group (blue) represents the more experienced evaluators, while the third group (green) represents the less experienced evaluators (Figure3). A start conference was held to discuss what had to be addressed and how. The application was then individually reviewed by each tester and reported and reviewed with the group at the following meeting. The third test was then conducted by a different group of two individuals. The same basis was kept regarding the testing, where a start conference was also held to introduce the application and how the tests were to be done, following a review by each tester and a reporting. The major test covered in the beginning were the home page, calendar, financial, notes/reminders, and useful links. The test results were then prioritized on a scale of one to three at the end of each change statement. Priority one meaning that the functions must be changed immediately and affect functionality, priority two meant the functions that should be changed to increase simplicity but do not affect functionality, and lastly, priority three meant that they are minor fixes that would make it visually better, but not a mandatory need for change. A clearer representation of the testing can be seen below. (Figure 3)

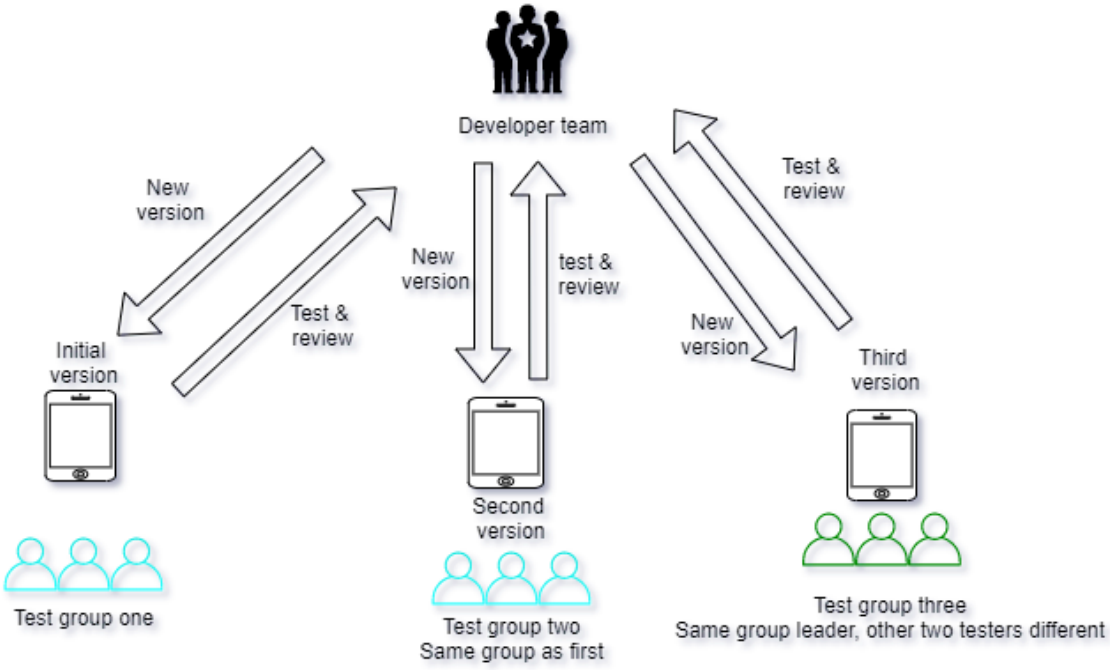


Figure 3. Testing development process.

3.2.1 Testing requirements

The tests conducted had to have consistency implemented which meant that testing requirements would also need to be taken into consideration. Each of the three tests was divided properly to give enough time for the developer team to work on the fixes. Test users were asked to give precise details regarding each remark they give about the application so that the developers can clearly understand the comments made. It was also necessary to give a prioritization score for each remark as shown in figure 4. This shows the importance of each test remark found and what role it plays in the development of the application. It is also important to note that each evaluator was presented and given the heuristic rules (section 2.2) as a guide when testing, in doing so, the evaluators were given at least a general understanding of the heuristic rules so that there can be some common ground between the two different test groups. Furthermore, the application categories were also split into main sections, which made it easy for the testers to analyze and split these into sections as mentioned in section 2.2.

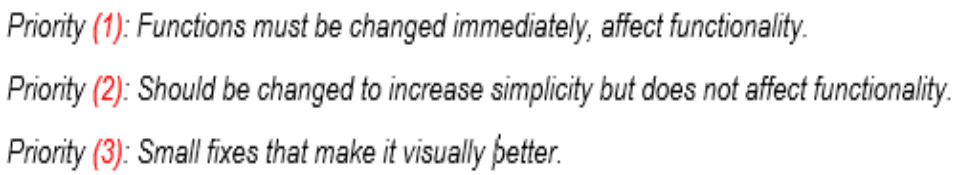


Figure 4. Prioritization of each test remark.

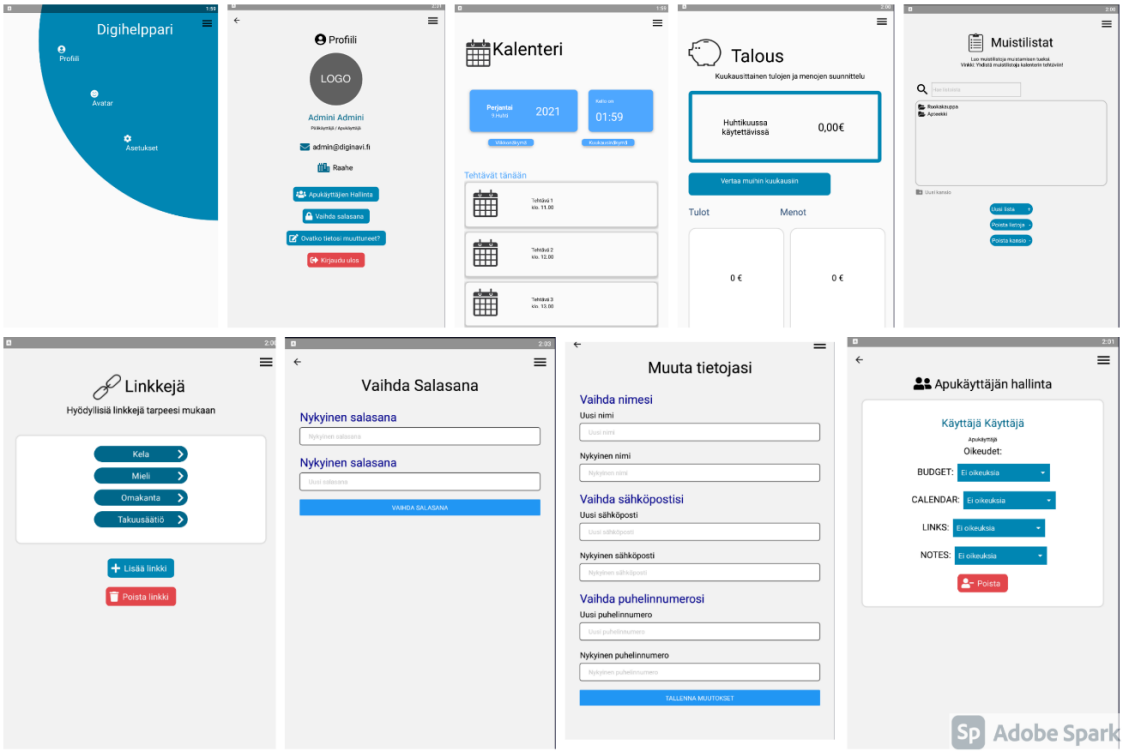
3.3 Testing

3.3.1 First test phase

In the testing process, different tasks and situations were tested several times to understand the general functionality of the application. The initial test version was given to each test user simultaneously so that there would be no obscurity present to any of the testers, as seen in picture 1. Using the heuristic rules (Nielsen, 1994) as guidance, each tester would go over the application individually without any communication to other testers. The amount of time each user took on the evaluation was not measured, since this would have been difficult to measure and of no major importance on the tests. Each remark was written down by evaluators along with a priority marking as shown in figure 4. Each test result per test user was then addressed in a meeting with one another on a later date.

The results of each remark showed to be of similar structure. Most of the remarks of priority one was found among each user. Although, there were some bugs found by users that were not covered by others. As each tester is different, this was expected and matched to the research done and presented in section 2.2. Remarks falling under priority category 1 When it came to the minor priorities (2-3) it was evident that everyone had different opinions and preferences. The evidence shows that there was an increase

in different opinions when the priority was between two and three. This is simply due to personal preference and was not subjective to the development of the application.

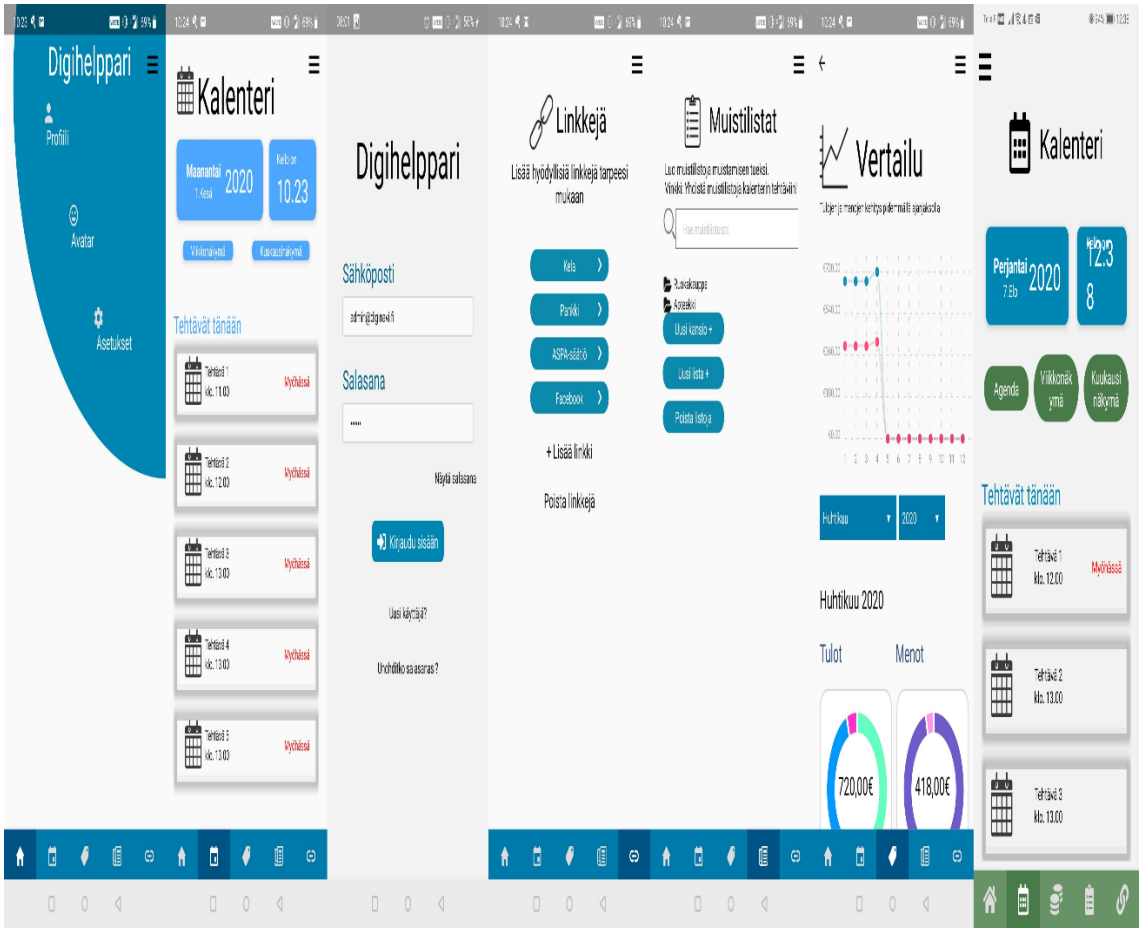


Picture 1. The initial layout of the application before the first test.

3.3.2 Second test phase

As mentioned earlier, the second group of test evaluators (blue) was kept the same as shown in the testing process in figure 3. The same process was used as in testing phase one. The reason for this was because the application was still under development, so some areas were still not to be tested. The second version of the application was given to each test user simultaneously, as seen in picture 2. Using the heuristic rules (Nielsen, 1994) as guidance, each tester would go over the application individually without any communication to other testers. The amount of time each user took on the evaluation was again not measured. Each test result per test user was then addressed in a meeting with one another on a later date.

Having done one test already the second test had brought upon more validation in the testing approach used. By not having changed evaluators, the tests showed to have a more strict result. In other words, there were comments and analyses made that were very precise and in full detail. This provided high-end feedback to the developers in making the necessary changes to the application.



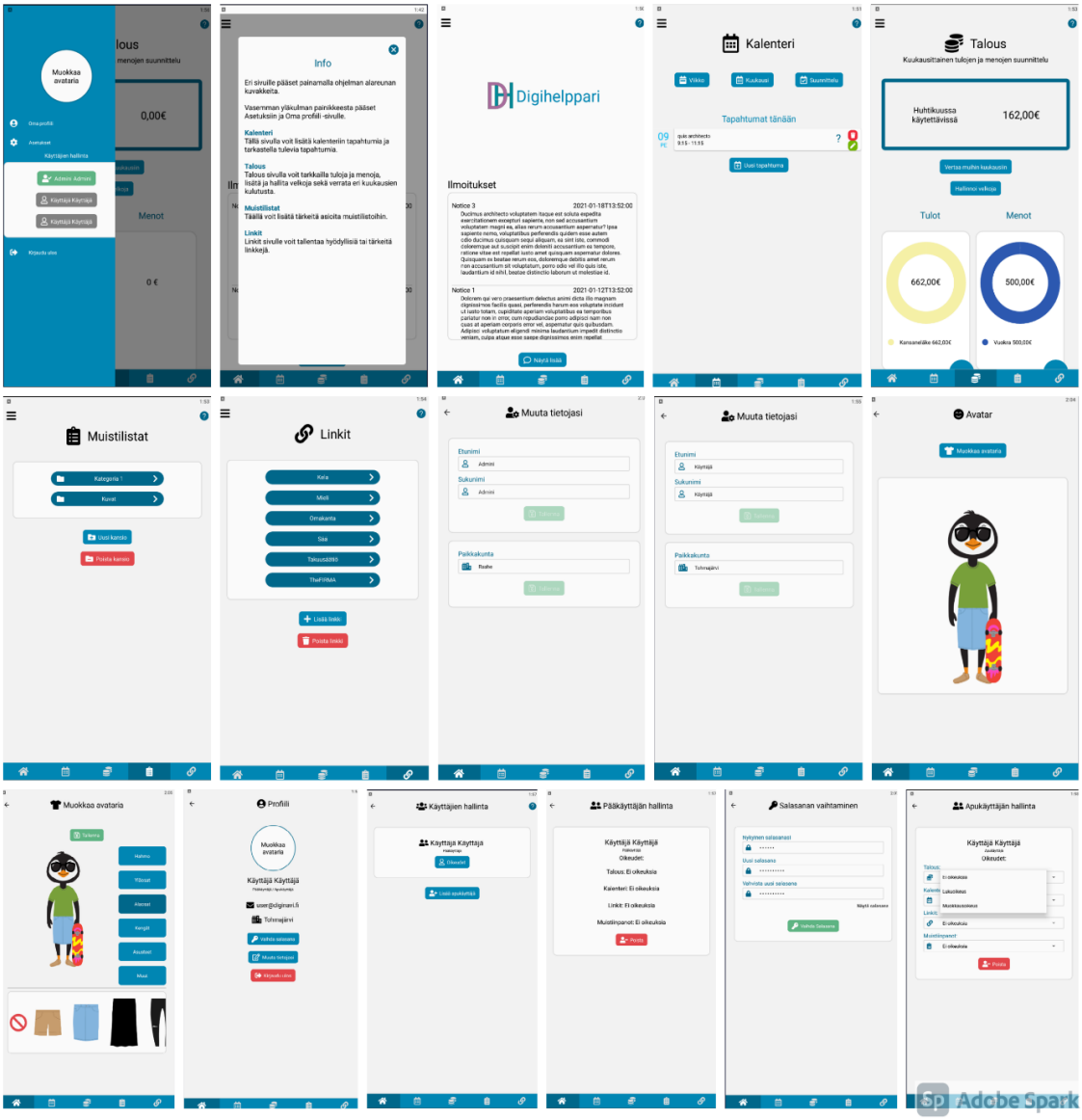
Picture 2. The layout of the application after the first test.

3.3.3 Third test phase

At this point, the final evaluation group (green) was changed for the test phase (Figure 3). By changing the groups, it gave a new perspective along with a new point of view into testing the application. Also, as mentioned earlier in section 2.2 having a different class of evaluators allows for a percentage increase when finding problems in the application.

The evaluators, in this case, fall under the category of less experienced evaluators, as presented in section 2.2. This meant that each evaluator had less background knowledge of how application testing is done along with the methodology used in a heuristic evaluation. Although, before testing, each test user was indeed provided with a set of instructions as to how the test would be held and in what order. Provided to each test user was an introduction about what the application is and what its intended uses are. In doing so, it was also necessary to give each test user a guided review of the heuristic usability evaluation as presented in section 2.2. This gave the evaluators a sense of what to expect from the application without seeing the product itself, allowing for a broad-minded approach to the evaluation. The updated application was given to the test group and was then analyzed individually (picture 3). The amount of time each user took on the evaluation was again not measured. Each test result per test user was then addressed in a separate meeting with one another on a later date.

The results of the final test confirmed that changing the group members indeed does play a major role in finding new problems and issues from the application. There were many security, UI, and UX issues found that were not addressed in the first two tests. Not taken into consideration that there were also some issues found in the first two tests that the developers had missed or were not fixed. There were also some key minor priority opinions brought about that made the application more user-friendly. This validates the facts brought about in section 2.2 when discussing the significance of having a different background in evaluators. The results in test three showed confirmation that some of the most basic usability problems were revealed by the inexperienced evaluators in comparison to the experienced evaluators. Thus, by having brought a different group into the final testing phase helped with the final application results. Also, by introducing inexperienced evaluators became present that the testing methods they used were slightly different. As stated in section 2.3, other test methods were used in the final test such as security testing and compatibility testing.



Picture 3.The layout of the application after the second test.

3.4 Future development

Once the first phase of test development is finished the second phase of testing will follow. This thesis studies the testing done using the heuristic evaluations approach on an application. There are plenty of other application evaluation methods that can be used for future testing, presented in section 2.3. The tests presented in this thesis were conducted with a selected group of testers that were not the originally intended

audience of the application. By conducting further tests with the intended group (ASPA) would confirm the further validity of the application and its intended use. This way we can assure the quality of the final product. Is the final product suitable for the intended user? Is the final product able to perform without any errors or bugs? Does the final product meet the requirements of the customer? These are only a few of the questions that must be answered in the later testing phases.

When developing an application, it becomes difficult to achieve all the needs in only one version of the system. The need for consistent updating, fixing, and new versions is mandatory to create a perfect software. Once the application is presented to the final intended users, there will be further needed changes according to their needs which is highly important to take note. Since the intended user's opinion is of the highest priority, it becomes crucial to make the needed changes and updates according to their preferences. For testing done in real-time, it may be interesting to study user acceptance testing with the intended users of the software. Also known as a Beta test, this type of testing happens with the end users at their location and verifies the software before the final release.

4 CONCLUSION

This thesis aimed to conduct a mobile test on the application “Digihelppari” using heuristic evaluation. The application would successfully aid the residents of ASPA, along with others, in their daily lives given any circumstances. In doing so, daily tasks such as managing finances, scheduling, and personal tasks would become a problem of the past.

Having this in mind, the testing phase of everyday solutions such as applications, websites, games, etc. is becoming more rigorous and demanding due to the consistent complexity of the system. Throughout the years, these testing methods are always changing and updated as the technology matures, meaning that it has become ever so important to fully cover the proper testing. In some test cases, the process might take even several phases depending on the circumstances of the development process. There might be scenarios where the end-user tends to be slightly more demanding regarding the overall design or the functionality of the system in general needs more development. In the case of this thesis project, the scheduling for the project was not what was originally anticipated. The initial plan was to first conduct the testing on a random group and after conducting it on the intended test audience. As a result, the intended scheduling and testing were changed due to the situation with Covid-19.

In conclusion, the thesis was conducted aiming to achieve a completed test for the application. The first phase was successful, but still, some modifications were to be made at the time of writing. Even after the three tests presented in this thesis, the application needs more development. In doing so, the product will find itself in the hands of the intended users and possibly even other locations. The next phase of the project would be to conduct a test with the intended users at the ASPA association, where the application would undergo further development. After this, if it is successfully tested and updated only then would it be a finalized product.

REFERENCES

Adanza, F. (2017). What is the difference between mobile and web app testing? Retrieved March 23, 2021, at https://www.getzephyr.com/insights/what-difference-between-mobile-and-web-app-testing

Bezsmolna, V. (2021). What is Mobile App Testing and Why is It Important? Retrieved April 5, 2021, at https://bitbar.com/blog/what-is-mobile-app-testing-and-why-is-it-important/

Canalys. (2020). Worldwide smartphone shipments Q4 2020 and full year 2020. Retrieved April 27, 2021, at https://www.canalys.com/newsroom/global-smartphone-shipment-Q4-2020#:~:text=In%20Q4%202020%2C%20worldwide%20smartphone,for%20a%20%2D12%25%20decline.

Gamu Ry. (2021). Gamu, Builder of digital bridges. Retrieved May 31, 2021, at https://gamu.fi/

Gao, A., Bai, X. Tsai, W. Uehara, T. (2021). Mobile Application Testing: A Tutorial. Retrieved April 7, 2021, at https://www.computer.org/csdl/magazine/co/2014/02/mco2014020046/13rRUEgs2EK

Gilmore, L. (2019). 10 mobile testing types and approaches. Retrieved April 25, 2021, at https://testlio.com/blog/10-mobile-testing-types-and-approaches/

Nec Software. (2021). Mobile application testing vs Web application testing. Retrieved March 23, 2021, at https://www.nexsoftsys.com/articles/difference-between-mobile-web-software-testing.html

Nielsen, J. (1994). How to conduct a heuristic evaluation. Retrieved April 20, 2021, at https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Softwaretestinghelp. (2021). What is software compatibility testing? Retrieved April 25, 2021, at https://www.softwaretestinghelp.com/software-compatibility-testing/

Testuff. (2014). Mobile Vs. Web-Based Software Testing: Are They the Same?. Retrieved April 5, 2021, at https://www.testuff.com/mobile-vs-web-based-software-testing-are-they-the-same/#:~:text=Whereas%20Web%2Dbased%20software%20testers,then%20you%20have%20software%20updates.

TheFIRMA. (2019). Requirement's specification of the "Digihelppari" application. Retrieved April 26, 2021, at file:///C:/Users/The%20Legend/OneDrive/School/THESIS/Diginavi_Aspa_Vaatimusm%C3%A4%C3%A4rittely.pdf

TheFirma. (2021). ICT-projektitoimisto. Retrieved May 6, 2021, at https://thefirma.fi/