Tampere University **Unit of Computing Sciences**
COMP.SE.610 Software Engineering Project 1
COMP.SE.620 Software Engineering Project 2

ProjectM

MobileMESH

Final Report

Roope Mantere
Neea Tienhaara
Jahid Sagor
Katja Heiskanen
Elsa Huovila
Vertti Nuotio
Diyaz Yakubov

Version history

Version	Date	Author	Description
0.1	28.04.2023	Neea & Katja	Chapters 2,3,4…
0.2	02.05.2023	Neea	Chapter 3, 7 and 8
0.3	03.05.2023	Katja	Chapter 5
0.4.	04.05.2023	Diyaz	Chapter 1 and 6
0.5	05.05.2023	Roope	Chapter 7
0.6	05.05.2023	Jahid	Chapter 10
0.7	05.05.2023	Neea	Polishing

Contents

1 Introduction

1.1 Purpose of the report

The purpose of this final document is to provide a comprehensive overview of the Mobile Mesh project, its outcomes, and the lessons learned during its execution. The document aims to describe how the project was carried out, the results it delivered, and how well it performed in meeting the initial requirements and objectives. Additionally, it seeks to highlight the areas where improvements could be made and the lessons that the team learned during the project's lifecycle. Ultimately, this report aims to serve as a valuable resource for stakeholders who are interested in understanding the project's outcomes and identifying opportunities for future projects.

1.2 Product and environment

The objective of the project was to develop an application that facilitates self-organizing MESH network among scouts. The application utilizes WLAN technology to establish a network between mobile devices (Figure 1.1). The app runs on a mobile device, and the goal was that the application could be used in a context where there is no GSM connection available.

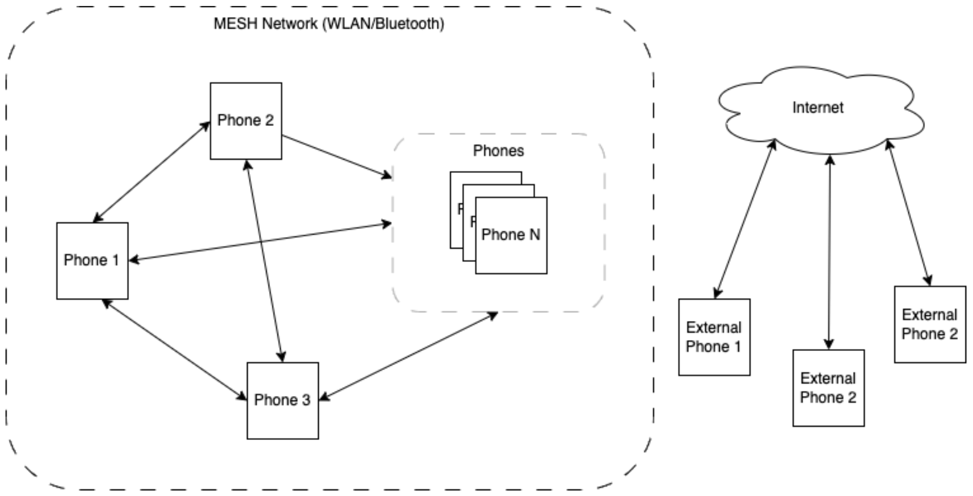


Figure 1.1 - MESH Network

The intended user group for this application are scouts. Scouts will be moving in remote environments, where there is no existing internet network, and need to share information. The application works as a platform for information sharing and reporting.

The application was supposed to include built-in trust mechanisms to authenticate new nodes, ensuring the security and reliability of the network. The user interface was intended to be designed intuitive and user-friendly, allowing the users to easily navigate the app’s features. Figure 1.2 illustrates the usage of the application via a context view diagram.

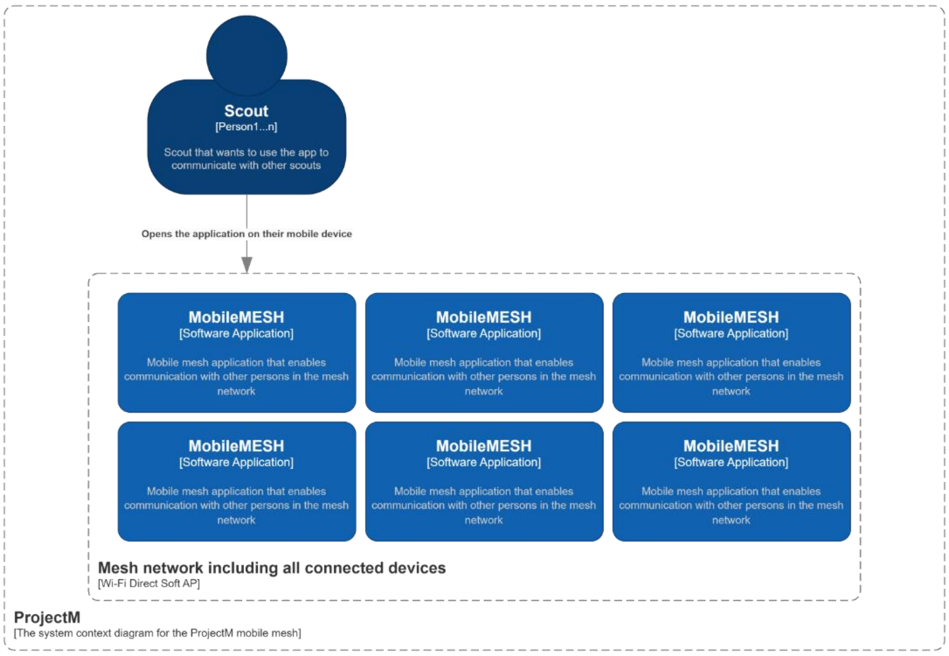

Scout
[Person1...n]
Scout that wants to use the app to communicate with other scouts
Opens the application on their mobile device
MobileMESH
[Software Application]
Mobile mesh application that enables communication with other persons in the mesh network
Mesh network including all connected devices
[Wi-Fi Direct Soft AP]
ProjectM
[The system context diagram for the ProjectM mobile mesh]

Figure 1 Context view

1.3 Definitions, abbreviations, and acronyms

Node	Mobile device that is part of the MESH network
Device	Synonymous with node
MVP	Minimum viable product
UI	User interface
UX	User experience
GDPR	General Data Protection Regulation in EU
IDE	Integrated Development Environment, tools for creating the project
GUI	Graphical User Interface
WLAN	Wireless Local Area Network

2 Project organisation

2.1 Group members

Neea Tienhaara

- E-mail: neea.tienhaara@tuni.fi
- Phone number: 04578404536
- Role: UX/UI, frontend

Katja Heiskanen

- Email: katja.j.heiskanen@tuni.fi
- Phone: 0443650755
- Role: UI/UX, frontend coding help
-

Elsa Huovila

- Email: elsa.huovila@tuni.fi
- Phone: 0449763607
- Role: UI/UX, frontend

Diyaz Yakubov
- Email: diyaz.yakubov@tuni.fi
- Phone: 0413131390
- Role: Project Manager

Roope Mantere
- Email: roope.mantere@tuni.fi
- Phone: 0400600703
- Role: Developer, backend

Vertti Nuotio
- Email: vertti.nuotio@tuni.fi
- Phone: 0407486523
- Role: Developer, backend

Jahid Sagor
- Email: jahid.sagor@tuni.fi
- Phone: 0415707913
- Role: Developer, backend

Customer

Manu Setälä, Solita
- E-mail: manu.setala@solita.fi
- Phone number: 0505577910
- Role: customer contact

Janne Niinivaara, Solita
- E-mail: janne.niinivaara@solita.fi
- Role: Head of Defence & Security, case owner

2.2 Other stakeholders

Pekka Mäkiaho, Tampere University staff
- E-mail: pekka.makiaho@tuni.fi, pekka.makiaho@istekki.fi
- Role: project coach

Scouts / military
- Role: end-user

3 Project implementation

3.1 Communication

3.1.1 Regular meetings

Meetings listed below were held regularly throughout the project course.

Customer meetings / biweekly sprint reviews:
- On Thursdays at 8am
- All group members and customer participated; coach participated every other week for the sprint review
- Group manager (Diyaz Yakubov) called the meeting
- Mostly remote meetings

Internal group meeting – everyone:
- On Tuesdays at 6pm
- Every group member participated
- Was held online on Teams

Internal group meeting – UX/UI team:
- On Wednesdays at 12pm
- Neea, Katja and Elsa participated regularly and at the end of the course Vertti joined the meetings for testing purposes
- Meetings at Linna

Internal group meeting – Dev team / everyone who wants to attend
- On Mondays at 6.30pm
- At least Jahid, Vertti and Roope participated
- Meetings were held remote

3.1.2 Tools for communication

These are the tools that were used regularly throughout the project course:
- Telegram was used for quick communication. There was a chat for only group members, separate chat for the UX/UI team and a chat between customer and the group (which was rarely used).
- Teams was used for meetings.

3.2 Tools and technologies

Down below are lists considering the used tools and technologies.

Tools for development:
- Android studio was selected as an IDE for developing android application. It was an easy choice considering that the client specifically wanted an android application, and Android studio is an IDE made for that purpose.
- Kotlin is a modern default language for android development, hence why it was chosen for this project.
- GitHub was selected as a code management tool for the project. Likewise, its Projects section is suitable for managing kanban board.

Tools for reporting:

- MMT was a course default tool for reporting, risks and time management.

Tools for testing:

- Google Accessibility Scanner was used to test the accessibility of the UI.

3.3 Sprints

Throughout the project, we completed a total of seven sprints, each lasting two weeks, with the exception of sprint 5, which our team extended to three weeks, and the last quality assurance sprint, sprint 6, which was only one week long. During sprint 0, we concentrated on gathering requirements, studying, and getting started in general. In sprints 1 and 2, we designed the user interface both by pen and in Figma, while also creating the first version of the chat feature. From sprints 3 to 5, the UX/UI team worked on implementing the designed views into the application in Android Studio, while the development team focused on multiple subjects, with mesh being the primary focus throughout these sprints. In sprint 6, which lasted one week, we focused on improving the application's overall quality by fixing bugs and accessibility issues. However, due to many team members already having worked many extra hours, achieving perfect quality was not possible.

We did not keep track of the planned and done work numbers during the sprints. Since we cannot offer those numbers, the similar numbers from weekly reports can be found in table 3.1. below. However, the sprint backlog number in MMT seems to include the sprint backlog, in progress and in review items all summed up. Since we wanted to confirm that the customer was happy with the items in “in review” pile before moving those items to the done pile, there were still items in review pile at the end of the sprint waiting for confirmation. Keeping this in mind, most of the numbers in the sprint backlog number that can be found below come from in review pile that was not by then cleared.

Table 3.1. Planned and done work numbers from weekly reports

Sprint number	**Sprint backlog**	**Done work**
0	2	4
1	14	7
2	6	16
3	11	17
4	12	30
5	15	43
6	11	48

Still, it is necessary to also point out that we started continuously having lingering work items in the “in progress” pile after the first sprints. Most of the lingering work items were somehow related to the mesh, which turned out to be very difficult to create and which took a lot of resources. Even so, the sprint backlog was not in reality in that much of an imbalance with the done work as one could think from only viewing the table 3.1.

At the end of the project, there were two items remaining in the product backlog. These two items were user stories, one aimed at reducing the programs battery consumption, and the other focused on developing a mission-specific tool for dividing tasks and checking their completion. Unfortunately, these tickets had to be left out due to time constraints. The total number of items implemented in the project is 59.

Prioritisation of the product backlog itself was quite successful. Still, it must be said, that more work items were taken per sprint than could be completed from sprint 3 and onwards.

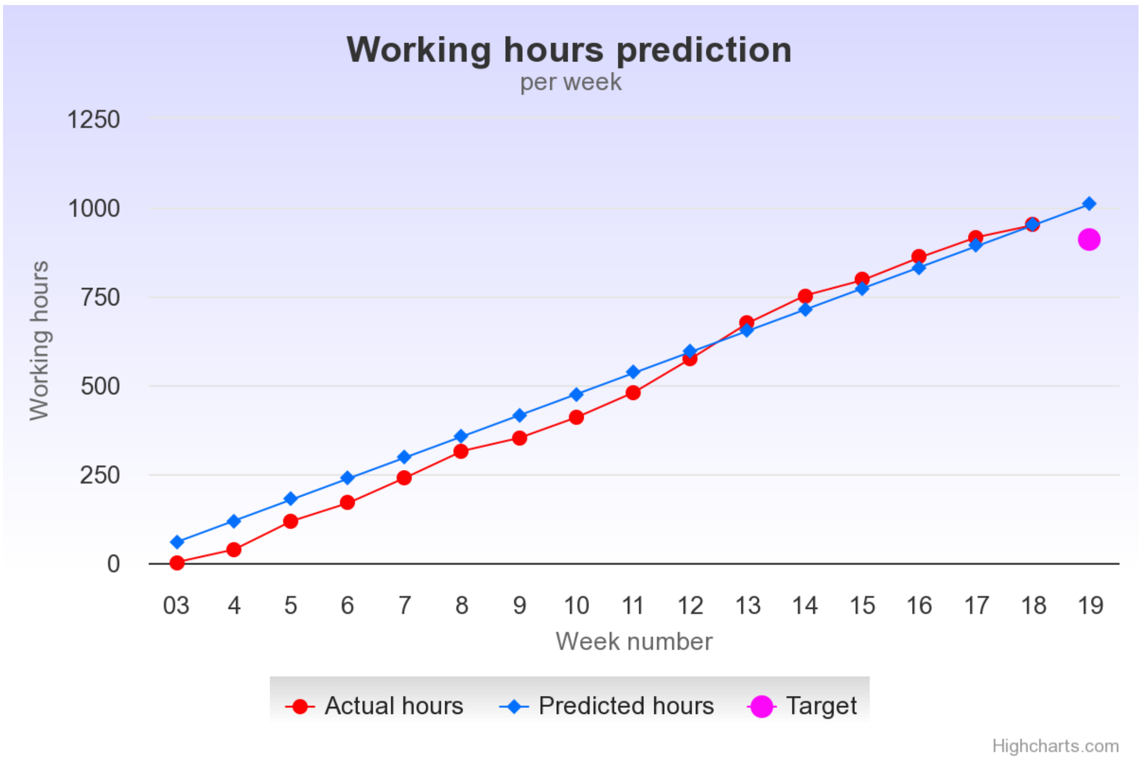


Figure 2 working hours prediction chart

3.4 Deliverables and outcomes

Actual product was delivered to the customer after acceptance testing held in 5.5.2023.

The group estimates that the product is 75% complete. To enhance its usefulness and prepare it for use, several improvements are necessary. Specifically, the mesh joining process needs to be made more intuitive, enabling users to join an existing network without waiting for an invitation or creating their own network. Also, the user interface should be designed to make it easy for users to understand the available options and underlying processes. Additionally, all the issues with mesh must be resolved to make the product useful.

While there are still some issues to be resolved, it is important to note that this product is intended to be a mere showcase mesh technology / proof-of-concept. Therefore, it may be considered to be in an acceptable state of readiness.

Here is information related to the code itself:

- Total amount of source code (SLOC): ~2447
- Self-coded lines: ~80%
- Totally reused X % (code used as is): ~5%
- Partially reused Y % (modified code): ~15%
- Number of classes and methods: 23
- Number of views / main windows: 16

3.5 Restrictions and limitations

There ended up being a few functionalities uncompleted and already visually finished views left unused due to time limitations and technical difficulties.

Here is a list of functionalities that did not make its way to the final version of the application, at least in the planned way:

- Viewing location of other group members and sharing location **non-textually** (not in coordinates in the chat)
- Ability to join a network. **Currently only the device that created the network can add a device to the network.** This was not the intended way, and due to time limitations, the adding happens in a screen that opens after pressing create a new network button. This makes the application less usable, which we do fully realize, but we were not able to fix due to time limitations.
- Task view. A system where scouts can share tasks to be done and mark them as complete was not completed due to not having enough resources. Still, these things can be reported quite easily in the chat even without a tool, hence why this item was not prioritized.
- Prioritizing devices by distance. Closest devices are not prioritized in the application.

3.6 Third party components, licenses and IPRs

The MobileMesh software product is composed entirely of proprietary components and there exist no dependencies or obligations that have been incurred through the use of third-party components or software libraries. The intellectual property rights of the software, including any software libraries, components, and dependencies have been explicitly acquired.

The client has all rights to the product and its future development.

3.7 GDPR, ePrivacy and related matters

Our android mobile application creates a wireless mesh network between devices, and during the development, several GDPR and ePrivacy considerations were taken into account. One important consideration was obtaining explicit consent from users to collect and process their personal data, such as device information and communication data. To achieve this, users are asked for necessary data permissions when the app is first launched and informed about the reasons for data gathering. Additionally, users have control over their data, including the ability to delete or modify their information through their phone’s settings. Users can also confirm whether they want to form a connection with a specific device.

Another important consideration was implementing strong data security measures to protect user data from unauthorized access or disclosure. While the WiFi-direct framework provides some level of security, no additional security measures were implemented due to time restrictions. This is something that should be considered in further development to enhance user data protection.

4 Working hours

Estimated working hours were 10h/per week for per person. Some of the group members also stated that they would be willing to work extra hours weekly if it was necessary for the completion of their own part.

Realized working hours are available in the tables below and can also be seen in the following pictures.

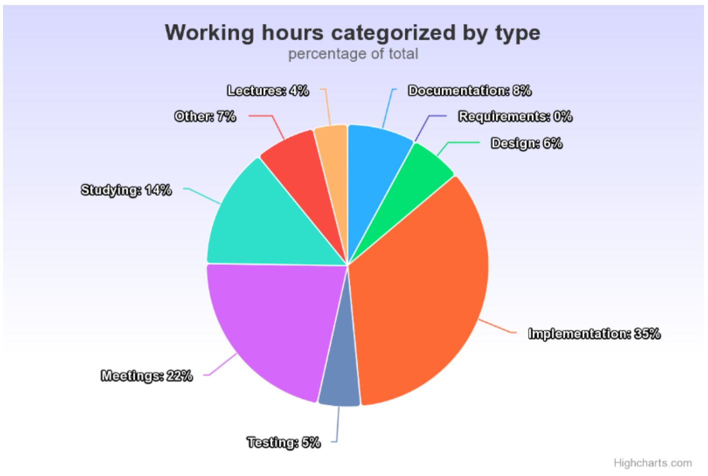


Figure 3 Working hours by type

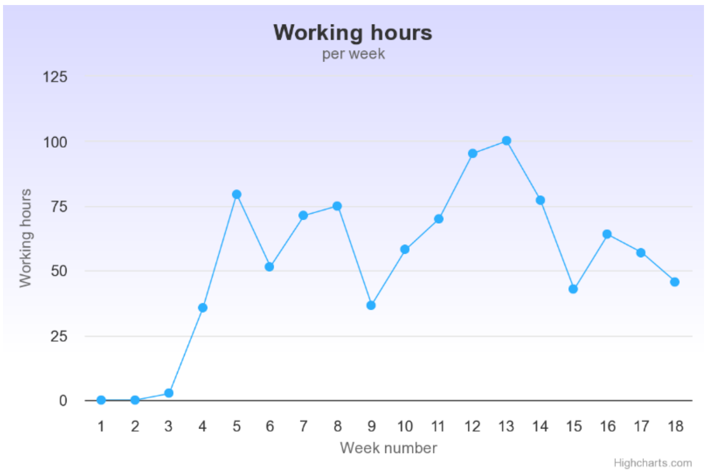


Figure 4 Working hours

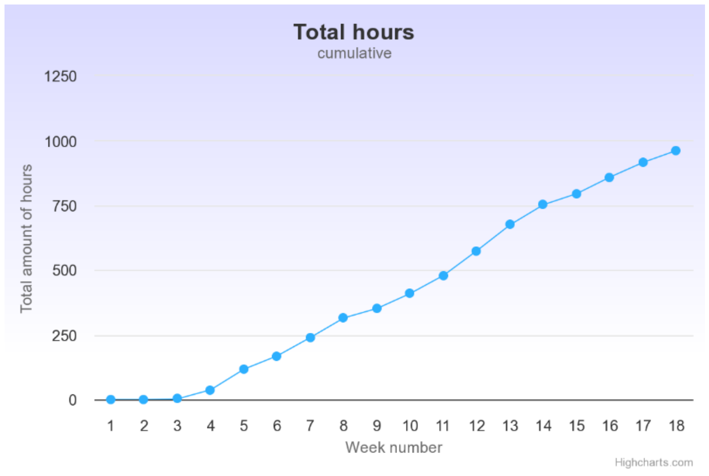


Figure 5 Total hours

Table 4.1. Realized working hours in person-hours by person and task

	Katja	Elsa	Neea	Roope	Vertti	Jahid	Diyaz
Documentation	13.5	6.5	7.75	6.9	1.75	6.5	37
Requirements	0	0	0	0	0	0	0
Design	16.75	15.25	17.5	3.5	4	0	0
Implementation	43.5	63.75	63.75	47.15	65.5	49	0
Testing	7	4	6.25	11	14.75	3	1.5
Meetings	30.5	32.25	25.5	21.75	37	33.5	32.75
Studying	18.75	13	16	27.5	19.75	31	11
Other	1.5	5.25	2.5	7.25	5.5	1.5	40.5
Lectures	6.5	7.25	5.75	6	2	5.75	5.5
Total	138	147.25	145	131.5	150.25	130.25	128.25

Table 4.2. Realized working hours (weekly totals / project).

	Weekly total
Week 3	3
Week 4	36
Week 5	80
Week 6	52
Week 7	71
Week 8	75
Week 9	37
Week 10	58
Week 11	70
Week 12	95
Week 13	100
Week 14	77

	Weekly total
Week 15	43
Week 16	64
Week 17	57
Week 18	34
Week 19	51
Total	966

4.1 Personal contributions to project

Diyaz was responsible for all the project management activities, including but not limited to project planning, risk management, task tracking, and progress reporting. He actively monitored the project's progress and ensured that it met the expected quality standards. Besides these, Diyaz did most of the paperwork related to the project. Additionally, he effectively communicated with all stakeholders and coordinated with the development team to ensure timely delivery of project milestones.

Dev team consisting of Roope Mantere, Vertti Nuotio and Jahid Sagor planned and implemented the back-end code and functionality for the application. Also, the dev team was deeply involved in testing the application and doing research on how to utilize the chosen technologies for the application.

Roope implemented the first 1 to 1 connection between the devices together with Vertti and after that Roope was involved in rewriting the device data structures such as the logic of saving data of the device. Also, Roope made the first demo of the application implementation of key structures Roope was more involved in testing and creating testing project for unit tests. After that he allocated time in to refactoring and bug fixing of the MESH with Vertti.

Vertti took the main responsibility for creating the MESH connection and improving it, and implementing the database for storing messages and networks. Most of the work was focused on researching, implementing, and fixing the MESH management system, but work was also done to integrate the backend with work done by the UI team.

Jahid implemented part of device information and store data locally. After that, Jahid worked on giving in-app the notification, also contributed to design the network view with the UI team. The work also included testing and UI improvement ideas.

UI team consisting of Elsa Huovila, Neea Tienhaara and Katja Heiskanen designed and implemented the UI of the app, as well as connected it to existing functionality. In addition, a lot of documentation was written by each member. UI team also participated in most of the system tests. Although the members assisted each other in many tasks, the roles were roughly the following:

Elsa implemented the view of network details in chat and worked on connecting the UI views together. Elsa also worked on some of the onboarding screens as well as the create network views together with Neea. Towards the end, Elsa did the accessibility testing and the separate test report based on that, as well as implemented some fixes to the issues that were identified during the test together with the UI team. Elsa also did some documentation, mainly with the project plan.

Neea implemented chat view, the navigation bar and some of the onboarding screens. Neea also worked on sending notifications, worked with Elsa to complete create network views and helped with finalizing some views in general. Neea also participated in writing project plan, test plan and final report.

Katja implemented the network and settings views, as well as came up with the color scheme for the app and made finalizing color and scaling edits for the UI based on the accessibility testing. In documentation, participated in writing project plan, test plan and final report, made presentation slides, and wrote most of the contents in test report.

4.2 Peer feedback

We did not utilize peer review that much as a group; it was up to individual members to submit feedback and reflect on what they received. The form of peer feedback as an evaluation form, although understandable for grading purposes, did not encourage group discussion for us as there were so many other things to do. The frequency of feedback was alright in our opinion.

5 Quality assurance

5.1 General description of testing

Testing included system and acceptance tests. In addition to testing the functional requirements of the app via test cases in system testing, we tested the usability of the app. There were 5 basic functional requirements for the app, and testing the process of those revealed many bugs to be fixed. We then made the most critical improvements to the backend and the UI based on them. Testing was mostly done in the two last sprints due to the team being behind schedule.

System testing was conducted by several group members taking turns in test sessions. Testers include Katja Heiskanen, Neea Tienhaara, Elsa Huovila, Vertti Nuotio and Roope Mantere. Testing used the Android Studio environment to run tests and debug processes at the same time. Tests were run on several phones using different versions of the Android system. Test results were logged into tables in the test report.

We also invited a test user to a system testing session to test the usability point of view and got comments from them on what was unclear. An accessibility test was done by using Google’s Accessibility Scanner on the application, which revealed issues with the scaling and contrast of the UI, that were then fixed. The customer made acceptance tests, which were conducted at the end of the course.

5.2 Bug reporting

We found 8 clear bugs in total, 4 of which are still open. Most of the bugs encountered are critical to the intended use of the app. It was difficult to track the root cause of some of the issues, so there might still be bugs that we didn’t discover during testing.

Severity ranking:
1 – Critical, prevents the app’s intended use
2 – Moderately hinders the app’s intended use
3 – Usability issue, doesn’t affect functionality

Test case	Bugs	Severity and status
1.Verify that the app can establish a MESH network between two devices using WLAN or Bluetooth technology	**(1) Device doesn't send network request to all devices** (2) After forming a connection, the other device is not visible (3) Messages are stuck in a "Waiting" mode	**(1) 1, open** (2) 3, fixed (3) 1, fixed
2.Verify that the app can authenticate new nodes joining the network and maintain the security and reliability of the network	**(4) Messages not received by all devices** (5) The new device couldn't find the original device (6) Issue with connecting a new device to 2-device network **(7) The process of connecting to a network is confusing to user**	**(4) 1, open** (5) 1, fixed (6) 1, fixed **(7) 3, open**
4.Verify that the app can handle multiple devices joining and leaving the network without affecting the network's overall functionality	**(8) In some cases, crashes on the original or new device after creating the network**	**(8) 2, open**

5.3 Conclusions on product's quality

We were unable to reach all the functional requirements for the app in the given timeframe. The app generally passes 3 out of 5 of the required test cases, but there are situations where those tasks can also fail due to bugs. The app's security is questionable due to the existing bugs.

The abandoned requirements were sharing location data in the app and ensuring that it would work in an environment where there is no GSM connection available. The former did not get implemented due to many bugs revealed in the basic connectivity, which took a lot of time to try to fix. The latter was not tested due to the difficulty of getting into an environment without existing network and the fact that testing it would have quickly become very time consuming.

The usability of the app also has some issues due to last minute changes to the logic of the app, which we didn't reserve enough time to redesign the UI. In summary, the process of creating/joining the network proved somewhat difficult for the user without instructions, and especially because we encountered the bug where all participating devices did not receive the network join request. On the other hand, the chat view and sending a message which were implemented according to plan, was very clear and intuitive for the user, as well as viewing network details and leaving the network.

The application is not error-free enough for production use and should absolutely be additionally tested and developed before any further releases! However, as an application which was primarily meant to be a proof-of-concept, it fulfils its purpose as expected and with further development could have use.

The customer has concluded that they can use it within their drone- and other showcases.

6 Risks and problems

6.1 Foreseen risks

The table below shows the list of identified risks.

Risk ID	Explanation, severity/impact, probability, size/importance
R1	Workload is too high. Some group members decide to drop the course because they have no time to do all the required things.
R2	There is an imbalance in the work amount between group members. Some group members might have a hard time getting the required hours while some might be working overtime. Those with not enough hours might not be able to pass the course and those with too many could get tired.
R3	There are health issues/burnout among the group members. Not everyone is able to contribute as much to the project because of this.
R4	Technical challenges are not feasible for the team or there aren't enough competencies. Some areas of the project might not be completed.
R5	Student events, such as those related to Vappu, keep the group members busy. Some areas of the project might be rushed / required hours may not be reached.
R6	Customer significantly changes the requirements in the end of project.

The only risk that "partially" worked out is the R4 risk. To mitigate that risk, we proposed a mitigation action: Allocate more time for acquiring new technical skills. Despite dedicating more time to study, we were not able to overcome all the technical challenges of this project within a given timeframe.

6.2 Risks not foreseen

One of the unforeseen risks we encountered during the project was that the team picked up the wrong tool for solving the problem. We had assumed that the tool we had selected at the beginning of the project would be sufficient for the task (Wi-Fi Direct library as a communication library/protocol). However, as we progressed through the development, we realized that the tool was not meeting our needs and was actually causing more problems than it solved. This was a significant setback for the project, as it resulted in a delay in the delivery timeline, as we had to take time to research and find out ways of dealing with limitations of the technology. Additionally, it was a learning experience for the team as we realized the importance of thoroughly evaluating and selecting tools at the beginning of a project, and not assuming that the tool we have selected will be sufficient for the entire project lifecycle. We have since noted how essential is a more comprehensive evaluation and selection process for tools in project planning to avoid this issue in the future.

7 Not implemented in this project

Throughout the project we were met with numerous technical difficulties that we had to take into account in our planning process. We had to prioritize some features and discard others. In this chapter we aim to discuss about the features that were left outside the scope of the project and also give some future development ideas to further refine project M.

7.1 Rejected ideas

Many great ideas had to be discarded simply because the technical difficulties we encountered took a lot longer to solve than we anticipated. Voice communication was rejected because of time constraints. Map view was too complicated to be implemented according to the assigned developer, so we postponed it until we decided to leave it out to make room for bug fixing also, the ability to edit network name and add a description to it was discarded due to the rush to complete the final demo version. Task view was removed early on as an unnecessary, but it could be added in the future development. Location sharing was discontinued after a failed attempt to implement by the assigned developer for being too difficult to implement. The application is in dark mode only since we thought that it would be better to have only dark mode than light mode. Message End to End encryption was also discarded because it was deemed unnecessary due to the basic Wi-Fi framework providing enough encryption to deliver a MVP.

What comes to third-party frameworks we were planning to use serval mesh for our project, but it deemed to be too outdated and contain too much unnecessary features for our project, so we decided to go with our own solution. This wasn’t roach either and probably one of the main reasons that we accumulated so many bugs.

7.2 Further development

The source code we produced will probably be a good basis to continue research and work on. The rejected ideas that we discussed in the last chapter can easily be implemented to the project by accommodating enough resources and time for the work. Many of the rejected ideas were created in team wide brainstorming sessions where we thought of things that could be useful for the target audience.

8 Lessons learnt

During the group project, we learned a lot about creating a mesh, the difficulties related to it, and new technologies such as Kotlin and Android Studio as many of us had not used them before. We also learned about how to work in a project environment in general.

One important lesson we learned was to pay attention to how much work can be done in a certain amount of time and not overload ourselves with multiple tasks. There were also clear imbalances in the workload between group members, due to varying skill levels, roles and knowledge of the topic. Our group had a lack of senior developer experience in Android.

We also realized that we should have had more development meetings between the development team and UX/UI team from the start to make the work more seamless and avoid doing unnecessary work. The UX/UI team created many screens that were not used due to a lack of time to implement the features, resulting in wasted hours and less usable application that could have been avoided by simply working together more closely. It would have been more productive to start design and development with a very small-scale MVP, before implementing any additional views or functions.

We received compliments about our work in general during the first few sprints and continuously received compliments about our course deliverables such as written plans.

Moving forward, we need to prioritize collaboration between teams, set more realistic goals, and make better use of our time to improve the outcome of future projects.

9 Comments about the course

This course was a great opportunity to learn about projects and teamwork. Working in a team allowed us to learn from each other's strengths and weaknesses, which helped us grow as individuals and as a team. However, the workload was quite high, given that we only earned 5 course points. The amount of required meetings and documentation also took a lot of time away from actually working on the project, and we did not use the meeting time as effectively because there wasn’t time to implement much new features in between them without working overtime.

It was challenging to balance the course with other responsibilities, but it also taught us how to manage our time effectively. Overall, the course was a valuable experience that will help us in our future endeavours.

10 Statistics

Project name: MobileMESH
Size of the group: 7

Project Manager
Diyaz Yakubov

Developers, backend
Roope Mantere
Vertti Nuotio
Jahid Sagor

UI/UX, frontend
Neea Tienhaara
Elsa Huovila
Katja Heiskanen

Name of customer
Manu Setälä, Solita
Janne Niinivaara, Solita

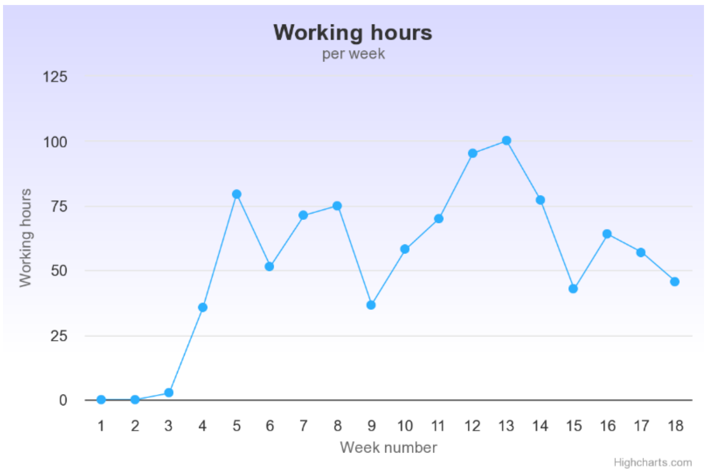
List of tools and technologies

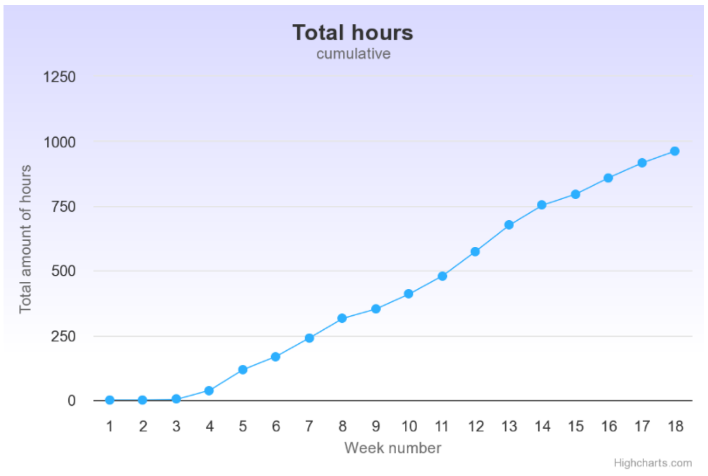
- Github
- Android Studio
- Figma
- Programming language – Kotlin

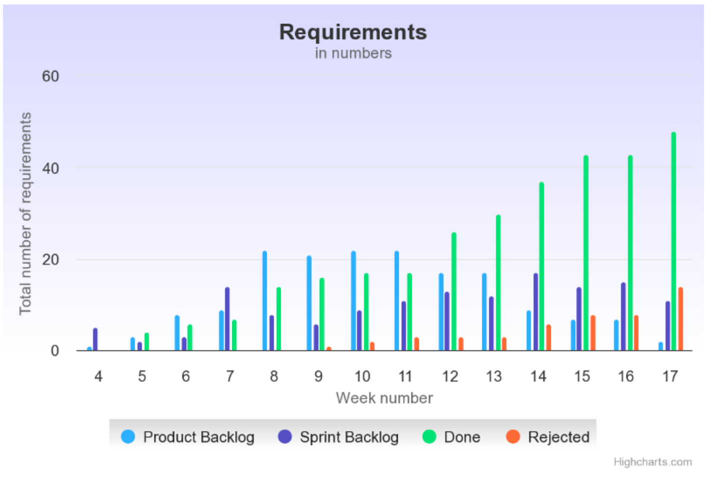
Code-related information

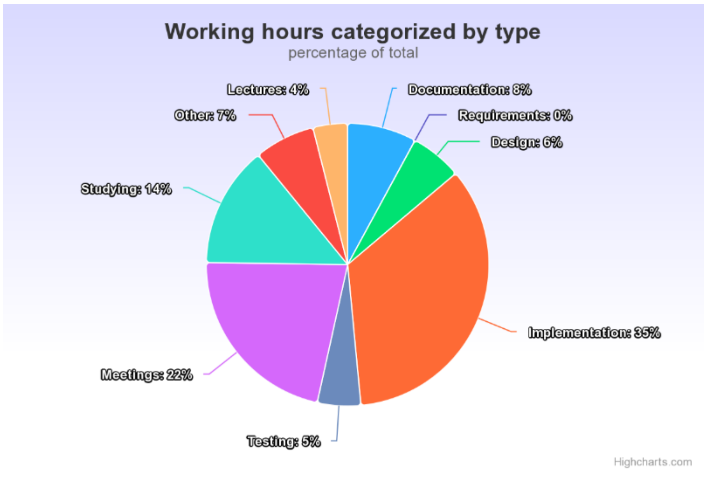
- Total amount of source code (SLOC): ~2447
- Self-coded lines: ~80%
- Totally reused X % (code used as is): ~5%
- Partially reused Y % (modified code): ~15%
- Number of classes and methods: 23
- Number of views / main windows: 16

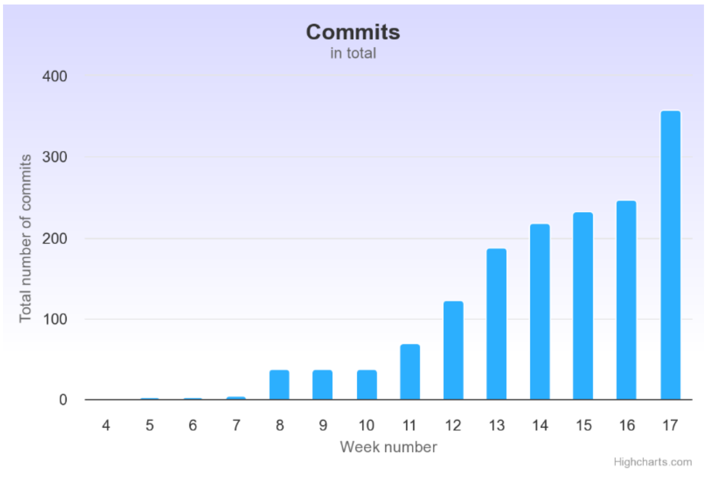
Charts

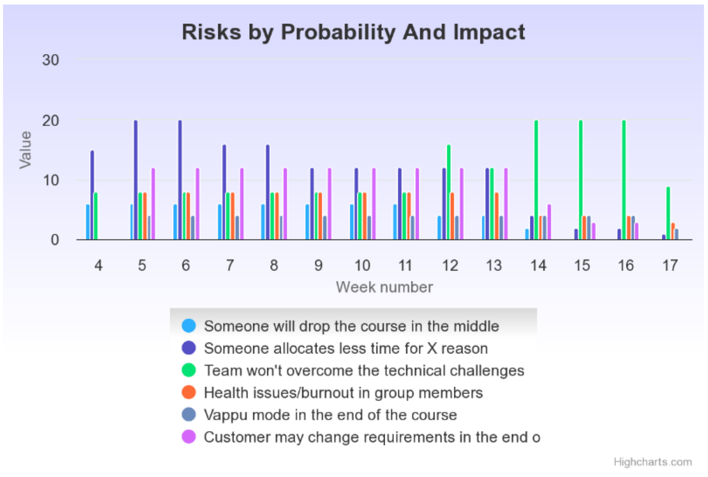





Requirements
in numbers
Total number of requirements
60
40
20
0
4 5 6 7 8 9 10 11 12 13 14 15 16 17
Week number
● Product Backlog
● Sprint Backlog
● Done
● Rejected
Highcharts.com


Working hours categorized by type
percentage of total
Lectures: 4%
Documentation: 8%
Other: 7%
Requirements: 0%
Design: 6%
Studying: 14%
Implementation: 35%
Meetings: 22%
Testing: 5%
Highcharts.com

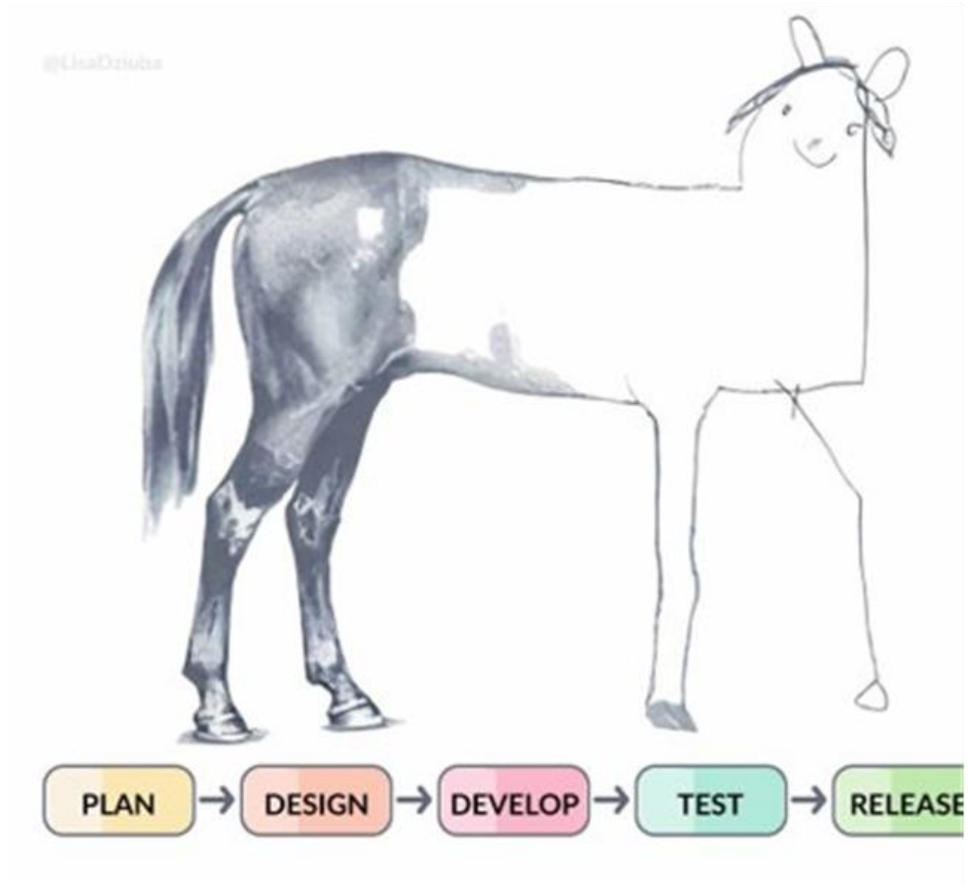

Commits
in total
Total number of commits
400
300
200
100
0
4 5 6 7 8 9 10 11 12 13 14 15 16 17
Week number
Highcharts.com



Personal thoughts

- Members teamed up very well
- Communication, work distribution, transparency, collaborative atmosphere
- A challenging error took on too big a project using untested technology

What we made



“75% of planned work”

Customer feedback

"I am so HAPPY with this application!"
-Manu Setälä, Solita