Tampere University **Unit of Computing Sciences**

COMP.SE.610 Software Engineering Project 1
COMP.SE.620 Software Engineering Project 2

Group 4

Snowledge: Pallaksen Lumisovellus

Project plan

H278851, Katariina Collander
150551124, Abdullah Arif
H275550, Oona Laitamäki
K435282, Emil Calonius
K428358, Lotta Orsmaa
K436244, Tiina Tamminen
K439387, Juho Kumara

Version history

Version	Date	Author	Description
0.1	14.9.2021	Katariina Collander	First draft
0.2	16.9.2021	Katariina Collander	Wrote some quick infos
0.3	20.9.2021	Oona Laitamäki	Introduction and tools
0.4	21.9.2021	Katariina Collander	Notes and work division
0.5	22.9.	Lotta Orsmaa	Wrote UI and usability requirements
0.6	22.9.	Juho Kumara	Wrote study plan
0.7	23.9.	Abdullah Arif	Wrote constraints
0.8	24.9.	Emil Calonius	Wrote risks
0.9	24.9.	Katariina Collander	Wrote requirements, high level goals discussed with customer and other missing parts
1.0	24.9.	Katariina Collander	Finalished and returned the document
1.1.	28.9.	Katariina Collander	Additions and corrections based on project plan review

Contents

1 Introduction

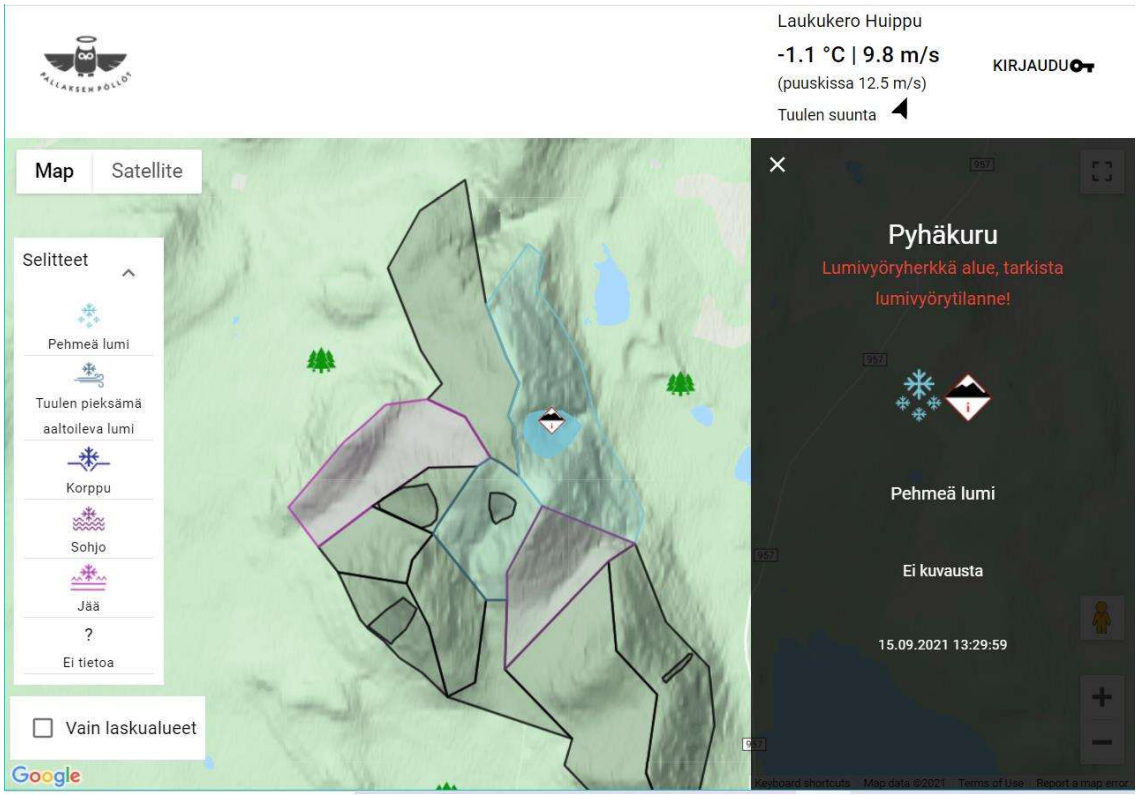
1.1 Purpose and scope of project

Purpose of this document is to give a big picture of the project, its schedule, risks, and progression. This document ensures that the project team remembers to take everything necessary into account when planning the project. This document is also an information source for the group and course personnel how the project is proceeding.

Scope of the project is software development regarding some topic provided by the customer, in this case improving an existing web application. Project is done based on the customer requirements and exploiting advice of Software Engineering Project course.

1.2 Product and environment

Product relating to this project is a web-based application called Snowledge by Pallaksen Pöllöt. The application provides snow information combined with weather conditions in different sections of Pallastunturi to people who are travelling in the Pallas area. Snowledge was produced together with the University of Tampere in a course which started in autumn 2020. The app can be accessed here: http://itc-pallas.rd.tuni.fi/ (link accessed on 23.9.2021). Application default view in the beginning of this project can be seen from picture 1.



Picture 1 Snowledge web application

The application has been running on Tavu Cloud so far, but during this project we should discuss if some other cloud provider could be used. Cloud provider should be considered specially in terms of cost-effectiveness. There are also some costs coming from usage of the Google Maps, as the application is using it as a base map.

This project is for further development of the application as there wasn't enough time to implement everything in the previous phase. Also based on the feedback from users there are multiple things that could be improved in the application. One of the most important goals is to improve the user experience.

User groups of this application are members of Pallaksen Pöllöt and people who are travelling in Pallas. There have also been Metsähallitus, rescue authorities and Finnish Meteorological Institute as application users up to this point. As there is no similar application in Finland, there is a great gap in the market for this product and commercialization can be seen as a possibility. However, commercialization requires increased cost efficiency in the application.

1.3 Customer's current system and other similar systems

The application has been in usage from the beginning of this year. It has provided great information for its users. So, the functionality provided by the application so far won't be changed dramatically.

There are already existing similar systems. Many of them are mobile applications, which provide information about snow and ski conditions. For example, one of these is onTheSnow Ski & Snow Report mobile application. But in Finland this web application seems to be one of the kind.

1.4 Project constraints

A test server is needed in order to start the project and understand in a better way. We'll have to ask TUNI for instructions how to access the test server which will take some time.

Most of the team members comes from different backgrounds professionally, almost every team member has to go through discussion study of the programming/Development languages e.g. React, Node.Js, VirtualBox, Vagrant.

Taking over a project that is already halfway through the development phase can bring constraints. It consumes to understand the previous developmental strategies, techniques, methods and tools and the team is constrained to use the already chosen solutions.

Integration of the map is a humongous part of the project and it needs to be changed. After discussing it with the client we are still on the phase of selecting the better map and payment for it.

1.5 Definitions, abbreviations and acronyms

GitHub	Version control system
MMT	Metrics monitoring tool for project management
Node.js	Back-end JavaScript runtime environment
React	JavaScript library

Tavu Cloud	Cloud provider
UI	User interface
User	Someone who uses the web application
Vagrant	Software for building virtual software development environments
VirtualBox	Software for virtualization
Snow record view	The right hand side panel where a segment's snowtype and its timestamp are presented
Snow record entry view	The dialog where the user enters a new snow record with snowtype

2 Project organisation

2.1 Group members

Katariina Collander, project manager
- Katariina.collander@tuni.fi, 040 484 88 36
- Master's student with studies focusing on software business and management.
- First time being a project manager
- Weekly hours dedicated to this project are 10-15.

Oona Laitamäki, senior developer
- oona.laitamaki@tuni.fi, 0400 546918
- Master's student with web & cloud as a major subject
- Some previous experience with React and node.js tools
- Weekly hours dedicated to this project are 10-15

Juho Kumara, developer
- juho.kumara@tuni.fi, 044 048 11 74
- Bachelor's student with computer science as a major subject
- Some previous experience with React and GUI programming
- Weekly hours dedicated to this project are 10-15

Abdullah Arif, developer
- Abdullah.arif@tuni.fi, 00923135481050
- Master's student with software web & cloud
- Some previous experience of developing with MySql
- Weekly hours dedicated to this project are 5-10

Tiina Tamminen, developer
- tiina.tamminen@tuni.fi, 044 2510909
- Bachelor's student with data processing as a major subject
- Some previous experience with UX designing
- Weekly hours dedicated to this project are 5-10

Lotta Orsmaa, developer

- lotta.orsmaa@tuni.fi, 050 348 15 25
- Bachelor's student with Computer Sciences as a major subject
- Some previous experience of UI/UX design and usability evaluation. Also a bit graphics design background.
- Weekly hours dedicated to this project are 10-15

Emil Calonius, developer
- emil.calonius@tuni.fi, 045 1134569
- Bachelor's student with Computer Sciences as a major subject
- Some experience with React and SQL from studies
- Weekly hours dedicated to project around 10

2.2 Customer

Juuso Holstein, juuso.holstein@pallaksenpollot.fi, representative of Pallaksen Pöllöt
Arto Hämäläinen, arto.hamalainen@harte.fi, representative of Pallaksen Pöllöt

2.3 Related organisations

End-users will be members of Pallaksen Pöllöt and people who wish to get information about skiing conditions while in Palla, such as tourists in Lapland Hotels and skiers asking for snow information from Metsähallitus.

The test project is running on a test server provided by the IT department of Tampere University.

3 Project goals and ending/termination

3.1 High level goals of the project group

- Implement a product that the customer is happy with and ensure that they are happy during the development phase, too.
- Learn new things about design, technologies involved and other aspects of project implementation.
- Staying in the project schedule.

3.2 High level goals of the customer

- Features of priority 1-3 should be finalised (possibly with some compromises)
- Improving the usability. Give the app a polished and finalised look.
- Creating new functionality to the app.
- Staying on track on how the project is progressing. Knowing well the situation/state of the app at the end of the project.

3.3 High level goals and deliverables of the project

- Features of priority 1-3 should be finalised (possibly with some compromises)
- Having continuous and honest communication between the team and the customer throughout the project.

3.4 Quitting criteria of the project

The project is terminated if the customer withdraws or if there are overwhelming resourcing difficulties. Termination of the project should be a common agreement between the customer, the team and the project coach.

3.5 Ending criteria of the project

The project will be ended when the course ends in December. In order to end the project, the team has to present a final report and test report. The project is ended as a common agreement between the customer, the team and the project coach.

4 Project and Process management

4.1 Methods and tools

The application has been developed with React (version 17.0.1), Node.js (version 14), NPM (version 6.14.6) and MySQL (version 2.2.5) so far, so we agreed to use them also in further development. Of course, these versions can be upgraded if they make restrictions for further development.

Oracle VirtualBox (version 6.1 ok) and Vagrant (version 2.2.18 ok) are used for creating a development environment as these tools have already been used in the previous development of this application. Virtual machine for running the application in development phase is running on Ubuntu 18.04. It has also been tested on Ubuntu 20.04. Oona Laitamäki, a senior developer, is the guru of these tools listed.

Teams and Telegram are tools agreed for communication and collaboration for this project. Also email communication is used. MMT tool is used for weekly reporting and working hours logging as well as for risk management.

GitHub is used as version control system as also the customer has access to it. For this project we are using GitHub's Kanban board to keep track on what the team has done and what still needs to be done. The team uses Scrum methods: the work is divided into iterations, the team meets with the customer regularly, and the team holds scrum ceremonies for demos and iteration reviews.

Project demos will be recorded and stored in team's Onedrive folder. Demos will also be presented in sprint review meetings and the customer can explore the current version of the app on the project's test server.

There will be at least documentation for project planning, test planning, test report and final report. Also, some architectural documentation would be good to have. Meetings are documented and stored in team's Onedrive folder.

4.2 Planned meetings

There are three types of regular meetings: internal weekly meetings, sprint reviews and sprint planning meetings.

The team meets weekly in weekly 1–2-hour internal meetings on Monday's. Before sprint 1 these meetings have been about general discussion, dividing the workload and

agreeing on common ways of working. In the coming sprints the weekly session is dedicated to reviewing what the team has been working on, updating the Kanban board, and solving possible issues. Team weekly meeting occurs every week until the end of the project.

The team meets with the project coach and the customer after each sprint in sprint review meetings. The first meeting is already scheduled, and the rest of the meetings will be scheduled in that meeting. In these meetings the Kanban board and weekly hour reports are reviewed. Done work is demoed to the customer and there will be time for comments and discussion between the meeting attendees.

After the sprint review the team has a meeting for sprint planning. In the meeting, the work for the sprint is reviewed, assigned and estimated. The team will also verify that the features have all the needed content, and the conditions of satisfaction are clearly understood.

4.3 Learning and study plan

Team members participating in programming should read the documentation presented in the original project on GitHub. Members should focus especially on the parts of the project they have agreed to implement. Programming-focused members should also recall technologies used, such as Git and React, if there has been some time since they last used them. Studying map API is also required for those team members implementing features using maps. (Juho, Emil?)

Every member of the team working with backend (Abdullah, Oona, Emil?) should read the backend documentation on GitHub project. MySQL and database study is required. Virtualbox and Vagrant are used in testing of the project, therefore are integral technologies to study for backend development.

The UI/UX team (Juho, Lotta, Tiina) focuses on studying methods to enhance usability and overall user experience of the product. Searching for usability issues and studying usability theory to find possible solutions to problems found are important tasks for the UI/UX team. Studying how to utilize user task analysis, user profiles and affinity maps is valuable regarding UI/UX planning.

Customer also provides some training and advice during the course. Training provided by the customer includes common skiing knowledge and introduction of new snow types (and how to utilize them in the project). Customer also advices team members during the course by meetings and in a Telegram group.

Commercialization requires first studying how to monetize the app. As presented by Pallaksen Pöllöt, possible methods to do monetizing are: voluntary user donations, one-time payments, and advertising (with possible partners). Additional studies depend on the selected method. For example, selecting advertising as a monetizing method requires studying how to implement possible advertising.
Studying internationalization is needed, if the application is aiming to other countries in the future.

5 Project iterations and timing

5.1 Iterations (sprints)

The main goals of iterations are listed here. Sprint 0 is for requirements gathering, sprints 1-5 are implementation sprints and sprint 6 is dedicated to quality assurance and system testing.

This is a first draft of the plan. The plan will be adjusted accordingly during each sprint planning. The features will be studied and researched in more detail during the implementation sprints.

Table 5.1. Project schedule.

Sprint no	Schedule (start/end date) High level content.
0	Weeks 37-38 (13.9.-26.9.) Requirements gathering, setting up environment and writing project plan
1	Weeks 39-40 (27.9.-10.10.) UI group designs map view and snow record views, implement new snow types
2	Weeks 41-42 (11.10.-24.10.) Design new icons and app layout changes, finalise map, implement snow record views
3	Weeks 43-44 (25.10.-7.11.) Design weather tab, implement layout changes and locking of map
4	Weeks 45-46 (8.11.-21.11.) Implement weather tab and snow record pictures
5	Weeks 47-48 (22.11.-5.12.) Finalise unfinished work, work on minimum priority features
6	Week 49 (6.12.-12.12) Quality assurance, system testing, fixing bugs

5.1.1 Iteration 0 (planning)

- We had our first team meetings, three meetings with the client and a meeting with our coach.
- Requirements were gathered and turned into features.
- Team members studied the features.
- The features were given priority order and divided into tasks.
- Project plan was written.

5.1.2 Iteration 1 (first implementation sprint)

- Choose the map, and its usage designed
- Design map view (hiding map details, colors)
- Design snow record view and snow record entry view
- Plan where to get weather data from and do required preparations for the solution
- Implement new snow types and new information associated with them:

  - Sub and base types
  - Skiability factor

5.1.3 Iteration 2

- Implement map view using the new terrain map
- Design new icons and segment patterns
- Design app layout (ensure no elements block the app, location of weather tab)
- Implement the new snow record view
  - Timestamp
  - Two snow types with visible base/sub-types
  - Skiability factor (thermometer)
  - Enabler: Pictures (implemented fully later)

5.1.4 Iteration 3

- Enabler: Implement pulling, storing and calculating of weather statistics and other data
- Implement new icons and segment patterns
- Implement locking the location of map and limit zooming
- Implement layout changes (affects mostly mobile)
- Design weather tab

5.1.5 Iteration 4

- Implement weather tab
- Implement adding and showing pictures attached snow records
- Implement showing sub-segments more independently in map view

5.1.6 Iteration 5

- Implement showing the user's location
- Evaluate and document the commercialisation factors of the project
- (Implement english version)
- (Implement exporting of snow records)

5.1.7 Iteration 6 (quality assurance)

- Fix bugs
- Ensure quality (Note that this will be done in other iterations, too.)

6 Requirements

6.1 Functional requirements

1. Use a terrain map instead of Google Maps in the map view
2. Improve the usability and design of the map view and its segments
3. Improve the app layout
4. Improve the record types by adding new snow types, redesigning the icons and implementing using two snow types for one snow record
5. Add new functionality to snow records: sub-snow types, skiability factor and pictures

6. Show weather data and statistics
7. Show the location of the user
8. Study and document possibilities of commercialisation of the app

6.2 Non-functional requirements goals

6.2.1 Usability goals

Improving the usability of the app is the most important goal of the project. The app should have a polished and finalised look. The map and the snow records should be simple, convey all important information and easy to access.

The usability of the current implementation has been evaluated and its issues acknowledged in the requirements. The goal is to solve all major usability problems. We plan to measure reaching our usability goals by user testing.

6.2.2 Performance goals

The app should display the map and its data in a few seconds with a relatively good connection.

6.2.3 Reliability goals

The data visible on the app should always be the newest record. The age of the record should be clearly showed so the user can estimate how reliable the record is.

6.2.4 Security goals

The login should be safe so no one without correct credentials can edit the segments or users of the app.

6.3 User interface requirements

6.3.1 Notifications

We need to make sure that the users understand that the data is not in real-time by some notification which should be user-friendly. Also, we need a more understandable avalanche warning to one of the areas in the Pallas fell.

6.3.2 Map base

Map base is presented more precisely and possibly in 2d form. It should be a topographic map but remain easily understood by beginner users also. The user's location should be showing on the map and the different areas of the Pallas fell should be highlighted and clickable. We explore different alternatives, such as Retkikartta for utilizing the map base.

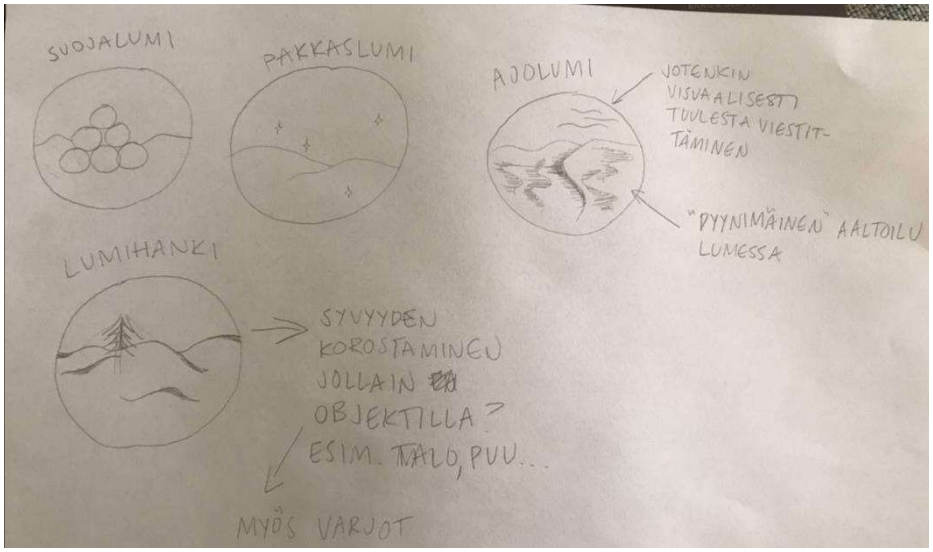
6.3.3 Navigation and layout

The navigation should be simple, clear and easy to use. The map, weather information and login could all be on separate pages. Navigating between the pages could be done, for example, using a navigation bar at the bottom of the screen.

The layout of the app can also be improved. On mobile other elements should not hide the map and on larger screens the map is so large that it shows unnecessary details. The map will be locked to a certain location, zooming will be limited, and depending on the chosen map some details will be hidden.

6.3.4 Snow types

We find out all of the snow types that the client wishes to add to the application and create more understandable icons and make the clickable parts on the map more distinguishable. We will add some shapes to the clickable parts on the map so that color would not be the only way to distinguish map areas. The snow types should be understandable to all users, not just the pros. Some beginning sketches presented below.



6.3.5 Weather data

Weather data and statistics are displayed on a separate page, while the most essential weather data is still shown on the map page. We have clarity on client's exact details on the information they want the app to display. One plan before finalising the requirements gathering was to take inspiration of the Google Weather forecast layout presented below. At this state the needs seem to differ from the example picture. More studying of the feature is required, and sketches will be drawn in iteration 1.

7 Risk management

7.1 Risk list

Table 7.1. Project risks.

Risk ID	Explanation , severity/impact , probability , size/importance
Sickness etc.	One or more team members being unable to contribute fully to the project. Reasons can include illness, personal issues, accidents or lack of time. Severity and probability depend on how many members are unable to contribute fully and how much time they can give. Severity = 3, probability = 2, size = 6.
Change in group size	One or more team members dropping the course entirely. Severity = 4, probability = 2, size = 8.
Personal technical issues	For example, with Git or running the app locally. Severity = 3, probability = 4, size = 12.
Lacking in skills	It is very likely that there are some aspects that require studying during the project. Severity = 2, probability = 5, size = 10.
Miscommunication with client	For example, there are different views on the project between client and project group due to miscommunication. Severity = 3, probability = 3 size = 9.
Miscommunication within group	For example, someone misunderstands their task. Severity = 3, probability = 3, size = 9.
Team-wide technical issues	For example, while running the app in the test server. Severity = 4, probability = 3, size = 12.
Under-estimating tasks	Issues with estimating tasks to require less time than actual. This can lead to delays or even having to leave out some features. Severity = 4, probability = 3, size = 12.
Over-estimating team's skills or time	Issues with estimating the hourss possible for the team members to dedicate or the project or their skill level. This can lead to delays or having to leave out some features. Severity = 4, probability = 3, size = 12.
New approaches not considered	Client is stuck on the ways how development was done in the first phase of the project and does not consider new approaches proposed by the team. Severity = 3, probability = 1, size = 3.
First-time project manager	First-time project manager might not realise everything required. Severity = 2, probability = 2, size = 4.

7.2 Risk monitoring

Risk monitoring will be part of the team meetings and sprint review meetings. In the team meetings, the project manager goes through the recognised risks with the whole team. In sprint review the most important risks are reviewed with the customer and the

coach. Our goal is to ensure that everyone in the team feels comfortable to share their project related issues in an early phase.

The MMT tool is used for monitoring risks and possible new risks are added there, too. The severity and probability of each risk is re-evaluated in MMT every week.

8 References

--

9 Open issues

The map API to use will be decided during iteration 1. The choice of the map affects the implementation work of other features, too.

10 Ideas for further development

- A mobile app could be created to make it easier for users to use the app on the go and to add advertisements to the app more easily