Project Plan

Authors:
Erik Bjørnsen
Simon Husås Houmb

Supervisor:
Peter Stefan Nussbaum

January 2025

Table of Contents

List of Figures

List of Tables

1 Goals & Constraints

1.1 Background

Skogkurs is a non-governmental organization established in 1958. The institute operates as a partnership, with 36 forestry organizations and scientific institutions as its members. Its activities are nationwide and encompass a wide range of topics, including forest management, construction and maintenance of forest roads, and forest operations and techniques [10]. Through its initiatives, Skogkurs aims to enhance the competence of professionals within the forestry industry and to promote knowledge of forests and nature to schools and the general public [11].

The Nordic forestry industry faces increasing challenges in ensuring stable timber transport throughout the year. Changes in climate have extended the snow-free season, creating variability in forest road conditions due to shifts between frozen, dry, and rainy periods. Addressing these challenges requires digital tools that can assess which forest roads are accessible during different weather conditions.

Recent research has focused on developing methodologies for digitally classifying forest road load-bearing capacities based on soil types and weather conditions. For example, a nationwide pilot study demonstrated that soil types, such as well-drained materials like glacial deposits, are more suitable for year-round use, while finer sediments require specific conditions like freezing or drying. These findings support the potential for fully digital solutions to predict road accessibility based on a combination of weather patterns and road construction characteristics. For more details see [Appendix A].

1.2 Project Goals

This section outlines the project's goals, which include product, impact, and learning goals.

1.2.1 Product Goals

The primary goal of the project is to develop and test a prototype system for fully digital modeling of forest road load-bearing capacity under varying conditions throughout the year. The solution will incorporate various geological and meteorological parameters, such as soil type and moisture, weather forecasts, and precipitation data, to generate an accurate classification of forest roads. This classification will be presented through an interactive map-based website. The roads will be color-coded using a traffic-light system, where green indicates safe roads, yellow signals caution, and red highlights unsafe roads. The system will provide users with a forecast for road conditions at least a week into the future, enabling better planning and decision-making. Furthermore, the system will prioritize ease of use, with an intuitive interface designed for transport managers, allowing them to make informed route choices based on real-time data and forecasts.

1.2.2 Impact Goals

- Reduced uncertainty for transport managers when setting the routes using forest roads.
- To validate the prototype's feasibility and effectiveness by conducting tests with end-users, such as transport managers, to assess its performance and usability in real-world scenarios.

1.2.3 Learning Goals

- Gaining insight in implementing interactive maps and geospatial data on web pages.
- Leveraging RESTful APIs for efficient data integration.
- Acquiring hands-on experience collaborating with real-world companies and products.
- Conducting user tests and implementing feedback.
- Developing a deeper understanding of the software development life cycle while actively practicing agile methodologies, like Scrum and Kanban.
- Enhancing application performance by implementing concurrency and optimizing parallel processing.
- Expanding proficiency in containerization techniques, particularly through hands-on experience with Docker.
- Implementing OpenStack deployment and configuration using Terraform for efficient infrastructure management.
- ***From NTNU [7]:***
  - Has in-depth knowledge of a selected topic within the subject area.
  - Has knowledge of research and development work within the topic.
  - Can identify, formulate and solve a relevant engineering problem.
  - Can apply knowledge and relevant results from research and development work to solve theoretical, technical and practical problems within the topic of the bachelor thesis and justify their choices.
  - Can apply engineering methods and work methodically.
  - Can document and disseminate engineering work.
  - Can plan and carry out engineering work.

- Disseminates professional knowledge to various target groups both in writing and orally in Norwegian and English.
  - Has insight into scientific honesty and understanding of ethical issues.
  - Has insight into environmental, health, social and economic consequences of products and solutions within their field and can put these in an ethical perspective and a life cycle perspective.
  - Integrates previously acquired knowledge and is able to acquire new knowledge in solving a problem.

1.3 Constraints

1.3.1 Temporal Constraints

- The set deadline for the final report is 20th of May.
- The presentation of the bachelor's thesis is scheduled for 4th or 5th of June.

1.3.2 Product Constraints

- The product will be using HTML 5, which requires newer versions of browsers.
- The product must be able to integrate with existing APIs.
- All software will be deployed on NTNU-hosted servers.
- The product must be accessible to users with varying levels of technical expertise, such as transport managers, requiring an intuitive and user-friendly interface.

1.3.3 Legal Constraints

- The product must comply with the licensing terms and conditions of all third-party services, including map distributors, external APIs, and any code libraries, frameworks, or tools used in its development and deployment.

2 Scope

2.1 Domain

The project domain encompasses the fields of geographic information systems (GIS), forestry, meteorology, and computer science, with a specific focus on digital modeling, data-driven decision support, and web-based visualization.

The project aims to integrate geospatial analysis, predictive modeling, and user interface design to improve the classification and usability of forest roads under varying environmental conditions.

2.2 Delimitation

- The product will primarily focus on mapping Gjøvik and its surrounding areas, as including larger regions would result in an excessive amount of data.
- The weight of vehicles using the roads will not be factored in when determining trafficability.
- The trafficability forecast for forest roads will extend up to a maximum of two weeks, as the product owner does not require a longer time frame.

2.3 Task Description

The project aims to develop and test a prototype system for fully digital modeling of forest road load-bearing capacity under varying conditions throughout the year. The development process can be divided into the following areas:

1. **Data collection and Integration:**
   - Decide and gather relevant geological and meteorological data, which may include:
     - Superficial deposits, soil moisture, ground water, ground frost.
     - Weather forecasts.
     - Historical and real-time road conditions.
   - Identify and implement suitable data sources and APIs for continuous updates.
2. **Classification and Forecasting:**
   - Develop a rule-based model to classify road conditions based on environmental factors.
   - Implement a traffic-light classification system (Green = Safe, Yellow = Caution, Red = Unsafe).
   - Extend the model to provide forecasted road conditions at least a week in advance.
3. **Web-Based Visual and User Interface:**
   - Design and develop an interactive map-based website for intuitive accessibility.
   - Implement a GIS-based visualization with real-time updates and historical road condition tracking.
   - Ensure that the system is optimized for transport managers, with a user-friendly interface that allows efficient decision-making.
4. **Testing, Validation and Refinement**:
   - Evaluate the accuracy and usability of the system by testing with real-world data.
   - Conduct user testing with transport managers or forestry stakeholders to assess the effectiveness of the interactive interface and forecasting capabilities.

- Incorporate potential user feedback.

5. **Documentation and Future Work:**
   - Provide detailed documentation of the system architecture, data sources, and model.
   - Provide a detailed user guide of the product.
   - Suggest potential improvements, such as machine learning model enhancements, additional data sources, mobile application integration, optimization, or further improvements of the app interface.

3 Project Organization

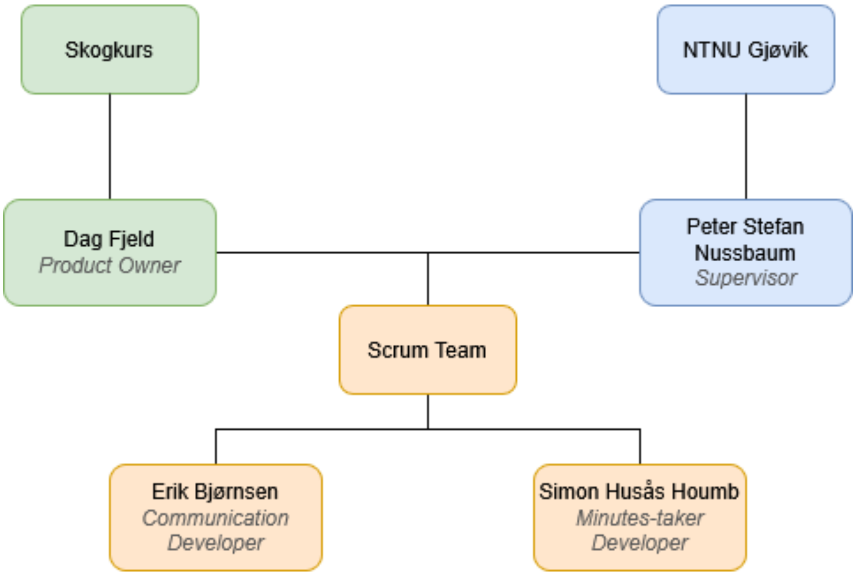

Figure 1: Project Organization Diagram

3.1 Responsibilities & Roles

- Simon Husås Houmb
  - Minutes-taker: Tasked with documenting the proceedings for each meeting that is held.
  - Developer
- Erik Bjørnsen
  - Communication: Responsible for communicating with Product Owner and Supervisor by mail, etc.
  - Developer

Since both group members share similar backgrounds, responsibilities for specific aspects of the development are not predefined. Instead, they will collaborate on all parts of the project, ensuring that everyone has an opportunity to learn and contribute equally.

3.2 Routines & Group Rules

3.2.1 General

- The members of the group should work a minimum of 30 hours on average per week, preferably on campus.
- The Core hours are every weekday 10:00 to 15:00. Group members are expected to work during this period.
- Decisions that have a major impact on the final product should be discussed with the other member of the group.
- All work hours must be recorded with Traggo.

3.2.2 Sickness

- If a group member is sick or for other reasons cannot work, it is still expected that they do their best to get some work done.
- If necessary, the other group member is expected to cover for the sick member to ensure a consistent workload.

3.2.3 Meetings

- Meetings with the supervisor will be held every Tuesday at 12:00, unless not necessary or the time is not suitable.
- Sprint meetings will be held every Monday, while also having daily scrum meetings.
- The proceedings of each meeting will be documented by the minutes-taker.

3.2.4 Communication

- Mutual respect, civility, and humility are required by the group members.
- Disagreements are welcome as they are a natural part of working in a team and are vital to any nuanced discussion.

3.2.5 Accountability Measures

- In the event of a conflict, the group will first attempt to resolve it internally. If the issue remains unresolved, a third party may be consulted for assistance. This could include the product owner, the supervisor, or another NTNU employee.
- If a group member falls behind in their work, they will be required to use the Easter vacation to catch up.
- For minor issues, the group member at fault will be expected to provide a treat (e.g., cinnamon bun, "databrus", beer, "GT") for the other member as a gesture of goodwill.

4 Planning, Follow-up, & Reporting

4.1 Selection of SDLC model

For this project, the team needed to select an appropriate system development life cycle (SDLC) model. Key factors considered in this decision included the clarity and flexibility of the requirements, team size, project timeline, product delivery goals, and prior experience [4].

The requirements provided by the Product Owner are intentionally ambiguous and flexible, allowing the team to prioritize the few fixed requirements upfront while iteratively refining the flexible ones over time. The team is composed of two members with similar experience levels, enabling close collaboration and effective decision-making. With a short project timeline of four months, a structure of 2-week sprints aligns perfectly with Scrum's iterative and adaptive approach, ensuring steady progress and regular opportunities for feedback.

Since our project lacks clearly defined requirements and requires ongoing collaboration with the Product Owner, Scrum was the best fit for our team and project. Traditional methodologies like Waterfall were not suitable, as they rely on fixed requirements and a linear development process, which would limit our flexibility. Extreme Programming (XP), while valuable for high-collaboration environments with strict engineering practices like pair programming and test-driven development, was not fully suitable as our project does not emphasize these practices to the same extent [1]. Similarly, Lean development focuses on minimizing waste and maximizing efficiency but is less structured in terms of iterative planning, which we need to manage evolving requirements effectively [6]. By adopting Scrum, we can iteratively refine requirements and adapt to changes throughout development [4]. To implement Scrum effectively, several key practices will be integrated into our workflow [8].

- **Sprint Meetings:** The project will be divided into 2-week sprints. At the start of each sprint, sprint meetings will include sprint planning, reviews, and retrospectives. During the planning phase, the team will select tasks from the product backlog to form the sprint backlog for the upcoming sprint. The review phase will focus on assessing progress and determining whether adjustments to the product backlog are needed. The retrospective phase will identify areas for improvement in the Scrum process itself.
- **Daily Scrum Meetings:** Short daily meetings will be conducted to discuss the progress of ongoing tasks, identify potential obstacles, and ensure alignment between team members.
- **Scrum Master:** Given that the team consists of only two members, we have decided to share the role of Scrum Master. Both members are responsible for ensuring adherence to the Scrum framework and continuously working to improve team efficiency. This collaborative approach allows us to maintain flexibility while upholding Scrum principles throughout the project.
- **Kanban:** To complement Scrum and further enhance workflow visibility, the team will utilize a Kanban board to track and manage tasks on GitHub. The Kanban board will consist of columns representing different stages of the workflow, such as Product Backlog, Sprint Backlog, In Progress, In Review, Done, and Discarded.

4.2 Plan for Status Meetings

Overview of the regular two-week status meeting cycle. For a more detailed view of all sprint meetings, see Figure 2.

Week	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
1	Sprint Meeting	Daily Scrum Supervisor (12:00)	Daily Scrum	Daily Scrum	Daily Scrum
2	Daily Scrum	Daily Scrum Supervisor (12:00)	Daily Scrum	Daily Scrum	Daily Scrum

Table 1: Plan for Status Meetings

5 Quality Assurance Organization

5.1 Documentation

Proper documentation is crucial, as there is a possibility that others may use the code in the future. To address this, both the code and the development process will be thoroughly documented throughout the project. All tasks will be tracked and recorded as GitHub issues, ensuring a clear history of the project's progress. Additionally, the proceedings of all meetings will be recorded by the designated minutes-taker (see Section 3.1). Furthermore, a comprehensive user guide detailing how to use the product will be created.

5.2 Source Code

All source code will adhere to industry-standard documentation practices and linting rules specific to each programming language to ensure clarity and maintainability. Although all source code should be documented, self-documenting code will be a central concept in naming variables and functions. For Go, the project will follow Google's official style guide [5]. All code should also handle errors properly to ensure the availability and integrity of the product.

GitHub commits should adhere to a defined standard [3]. Each commit message must include a relevant keyword that categorizes the task performed, such as feat, fix, docs, style, refactor, test, or chore. Additionally, each message should provide a brief and concise description of the changes made.

5.3 Storage

To ensure that no valuable data is lost during the project we will follow the 3-2-1 backup strategy. According to Seagate, the 3-2-1 rule has the following requirements [9]:

- 3 Copies of Data
- 2 Different Media
- 1 Copy Offsite

This strategy is used to prevent total data loss by copying data to different mediums and locations, ensuring that a disaster does not destroy all copies. A total loss of project assets would be catastrophic for the project, it is therefore important to minimize this risk as much as possible.

The original copy of data is mainly stored by the associated SaaS provider, i.e. Overleaf for LaTeX-documents, GitHub for repositories, and Slack for communications. A central backup server in SkyHiGh will continuously take backups of all repositories under the GitHub organization [2], along with copies of Traggo data. The two different mediums are covered by the two backups, as SkyHiGh mainly uses HDDs and our local PCs have SSDs.

Informal copies of repositories will also be taken throughout the project as a natural part of development. Slack data and the Kanban board on GitHub with issues will be exported and copied from a local PC to the backup server monthly. See table below for an overview of our strategy.

Asset	Main	Backup	Offsite
LaTeX-documents	Overleaf	GitHub repositories	Local PC
Source code & repositories	GitHub [2]	SkyHiGh backup server	Local PC
Kanban with issues	GitHub [2]	SkyHiGh backup server	Local PC
Minutes	GitHub [2]	SkyHiGh backup server	Local PC
Traggo	SkyHiGh	SkyHiGh backup server	Local PC
Slack data	Slack	SkyHiGh backup server	Local PC

Table 2: 3-2-1 Backup Strategy

5.4 Plan for Inspections & Testing

- Implement CI/CD pipelines on GitHub to automate processes as much as possible, starting early in development.
- The pipelines will include:
  - Static Code Analysis and Linting
  - Unit and Integration Testing
  - Deployment
- Conduct user testing with the Product Owner and potential end-users once the MVP is complete. Interviews and follow-ups will be conducted to measure satisfaction and gather feedback for the MMP.
- For unit and integration testing, we aim for 80% or higher coverage, with a primary focus on critical components and an emphasis on test quality over quantity.
- To monitor errors, identify potential availability and integrity issues, and facilitate debugging, a logging system will be implemented.

5.5 Tools

This section provides a detailed overview of the tools that the group intends to use throughout the project. These tools have been selected to facilitate various aspects of the project, including development, collaboration, deployment, and documentation. The table below outlines each tool, its type, and its specific use case within the project workflow.

Name	Type	Use-case
OverLeaf	Online LaTeX-Editor	Report writing
JetBrains IDE Family	IDE	Code Development
VS Code	IDE	Code Development
GitHub [2]	Code Repository	Version Control & Kanban
Docker	Containerization Platform	Containerizing Deployment
OpenStack SkyHigh	Cloud Computing Platform	Deployment
Postman	API Collaboration Platform	API Development
TeamGantt	Gantt Diagram Tool	Gantt Chart
draw.io	Diagramming Tool	Other Diagrams
Traggo	Time-Tracking Tool	Tracking Time
Slack	Communication Platform	Communication
OpenLayers	JavaScript Library for Interactive Maps	Interactive Map

Table 3: Tools

5.6 Risk Analysis

This section presents potential risk scenarios identified for the project. Each scenario is evaluated using a matrix that assesses likelihood and severity, categorized into four levels ranging from green (low risk) to red (high risk). Additionally, each scenario includes proposed measures or mitigations, prioritized from low to high.

Likelihood / Severity	**Minimal**	**Marginal**	**Moderate**	**Major**	**Critical**
Certain					
Likely					
Possible					
Unlikely					
Rare					

Table 4: Risk Matrix (5x5)

Risk #	**Description**	**Likelihood**	**Consequence**
1	One or more group members get sick over a longer period, making it hard to work on the project.	Unlikely	Major
2	Source code and/or documents become lost and unrecoverable due to issues with their storage location.	Unlikely	Critical
3	Infrastructure and systems could be attacked in a cyber attack, compromising the availability of the systems.	Unlikely	Major
4	The project's objectives expand beyond the initial plan, consuming additional time and resources.	Possible	Major
5	The sources for geotechnical data are found to be too unreliable or inaccurate.	Possible	Critical
6	The sources for maps and geotechnical data become inaccessible.	Rare	Major
7	Delayed meetings or feedback from supervisor, causing slight adjustments to the schedule.	Possible	Minimal

Table 5: Risks

Risk #	**Mitigation & Measures**	**Priority**
1	The sick group member should work as much as possible. Good documentation and communication will make it easier for the other member to step in and help with the workload.	Medium
2	Backups of all work should be made and stored in at least one separate location (see Section 5.3).	High
3	Proper and strict security rules need to be in place to limit access to only needed ports. Load-balancing and rate-limiting should be used to ensure availability.	High
4	Clearly define project goals and milestones. Review the scope with stakeholders to prevent unnecessary additions.	High
5 & 6	Research alternate data sources early in the project and implement modular and flexible design in the code for easier transition.	High
7	Practice good communication with the supervisor and include buffer time in the project plan.	Low

Table 6: Risk Mitigation & Measures

6 Execution Plan

This section details the project's execution plan, presented through a Gantt chart. The plan is organized into distinct milestones, activities, and decision points for clarity and effective tracking.

6.1 Gantt Chart

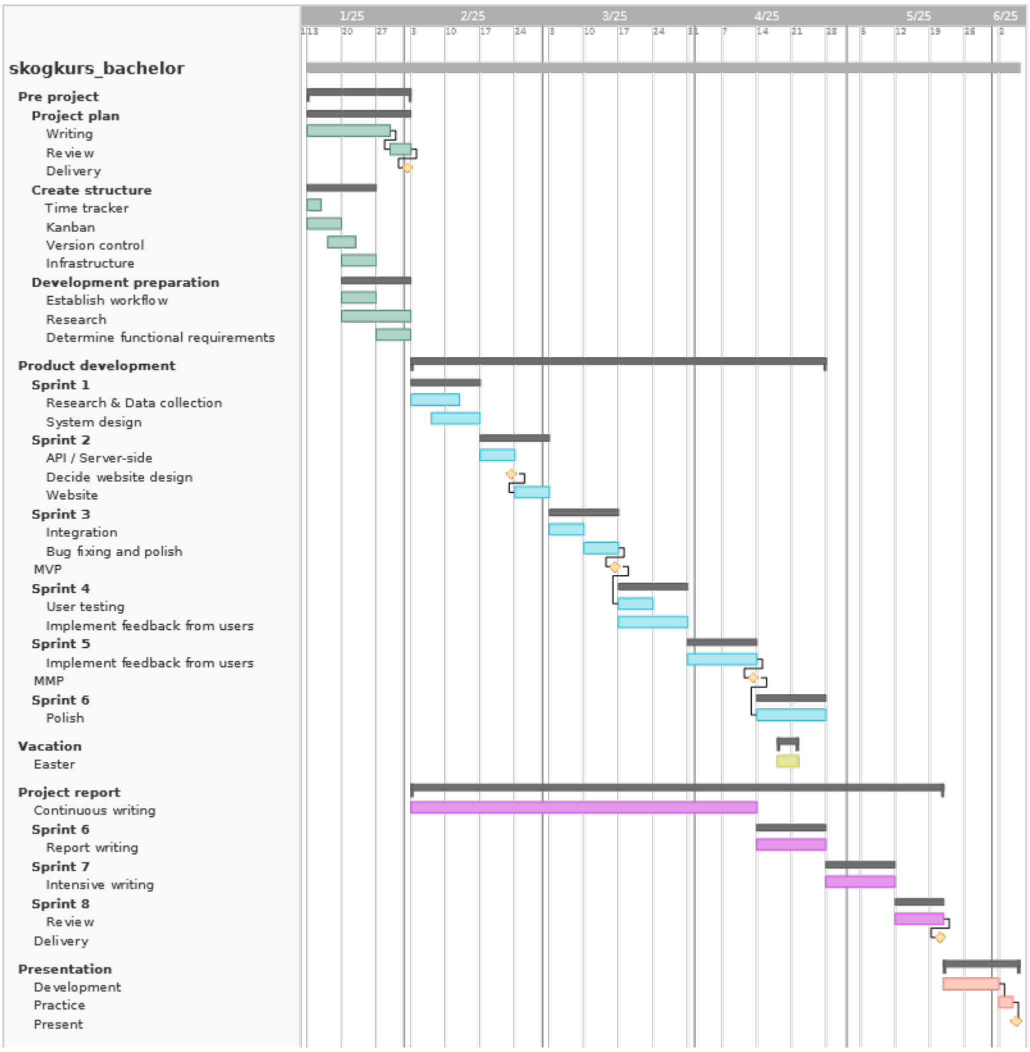

Figure 2: Gantt Chart

#	Name / Title	Start Date	End Date	Predecessors
1	skogkurs_bachelor	2025-01-13	2025-06-04	
1.1	Pre project	2025-01-13	2025-01-31	
1.1.1	Project plan	2025-01-13	2025-01-31	
1.1.1.1	Writing	2025-01-13	2025-01-28	
1.1.1.2	Review	2025-01-29	2025-01-31	1.1.1.1
1.1.1.3	Delivery	2025-01-31	2025-01-31	1.1.1.2
1.1.2	Create structure	2025-01-13	2025-01-24	
1.1.2.1	Time tracker	2025-01-13	2025-01-14	
1.1.2.2	Kanban	2025-01-13	2025-01-17	
1.1.2.3	Version control	2025-01-16	2025-01-21	
1.1.2.4	Infrastructure	2025-01-20	2025-01-24	
1.1.3	Development preparation	2025-01-20	2025-01-31	
1.1.3.1	Establish workflow	2025-01-20	2025-01-24	
1.1.3.2	Research	2025-01-20	2025-01-31	
1.1.3.3	Determine functional requirements	2025-01-27	2025-01-31	
1.2	Product development	2025-02-03	2025-04-25	
1.2.1	Sprint 1	2025-02-03	2025-02-14	
1.2.1.1	Research & Data collection	2025-02-03	2025-02-11	
1.2.1.2	System design	2025-02-06	2025-02-14	
1.2.2	Sprint 2	2025-02-17	2025-02-28	
1.2.2.1	API / Server-side	2025-02-17	2025-02-21	
1.2.2.2	Decide website design	2025-02-21	2025-02-21	
1.2.2.3	Website	2025-02-24	2025-02-28	1.2.2.2
1.2.3	Sprint 3	2025-03-03	2025-03-14	
1.2.3.1	Integration	2025-03-03	2025-03-07	
1.2.3.2	Bug fixing and polish	2025-03-10	2025-03-14	
1.2.4	MVP	2025-03-14	2025-03-14	1.2.3.2
1.2.5	Sprint 4	2025-03-17	2025-03-28	
1.2.5.1	User testing	2025-03-17	2025-03-21	1.2.4
1.2.5.2	Implement feedback from users	2025-03-17	2025-03-28	
1.2.6	Sprint 5	2025-03-31	2025-04-11	
1.2.6.1	Implement feedback from users	2025-03-31	2025-04-11	
1.2.7	MMP	2025-04-11	2025-04-11	1.2.6.1
1.2.8	Sprint 6	2025-04-14	2025-04-25	
1.2.8.1	Polish	2025-04-14	2025-04-25	1.2.7
1.3	Vacation	2025-04-17	2025-04-21	
1.3.1	Easter	2025-04-17	2025-04-21	
1.4	Project report	2025-02-03	2025-05-20	
1.4.1	Continuous writing	2025-02-03	2025-04-11	
1.4.2	Sprint 6	2025-04-14	2025-04-25	
1.4.2.1	Report writing	2025-04-14	2025-04-25	
1.4.3	Sprint 7	2025-04-28	2025-05-09	
1.4.3.1	Intensive writing	2025-04-28	2025-05-09	
1.4.4	Sprint 8	2025-05-12	2025-05-20	
1.4.4.1	Review	2025-05-12	2025-05-20	
1.4.5	Delivery	2025-05-20	2025-05-20	1.4.4.1
1.5	Presentation	2025-05-21	2025-06-04	
1.5.1	Development	2025-05-21	2025-05-30	
1.5.2	Practice	2025-06-02	2025-06-03	1.5.1
1.5.3	Present	2025-06-04	2025-06-04	1.5.2

Table 7: Gantt Chart in Text-Format

6.2 Milestones

- Milestone 1: 31st of January - Deadline Project Plan and Project Contract
- Milestone 2: 14th of March - MVP Done
- Milestone 3: 11th of April - MMP Done
- Milestone 4: 20th of May - Deadline Report
- Milestone 5: 4th or 5th of June - Presentation of Thesis

6.3 Activities

- 7th of March - Crash Course in Report Writing and Presentation Techniques.

6.4 Decision Points

- 21st of February - Choose the Design of the Website

Bibliography

[1] Agile Alliance. *Extreme Programming*. URL: https://www.agilealliance.org/glossary/xp/ (visited on 31st Jan. 2025).

[2] Erik Bjørnsen and Simon H. Houmb. *skogkursbachelor*. URL: https://github.com/skogkursbachelor (visited on 30th Jan. 2025).

[3] Conventional Commits. *Conventional Commits*. URL: https://www.conventionalcommits.org/en/v1.0.0/ (visited on 28th Jan. 2025).

[4] GeeksForGeeks. *How to Choose the Best SDLC Model for Your Project*. URL: https://www.geeksforgeeks.org/how-to-choose-the-best-sdlc-model-for-your-project/ (visited on 30th Jan. 2025).

[5] Google. *Go Style*. URL: https://google.github.io/styleguide/go/ (visited on 24th Jan. 2025).

[6] Ben Lutkevich and Valerie Silverthorne. *Lean software development*. URL: https://www.techtarget.com/searchsoftwarequality/definition/lean-programming (visited on 31st Jan. 2025).

[7] NTNU. *IDATG2900 - Bachelor Thesis*. URL: https://www.ntnu.edu/studies/courses/IDATG2900 (visited on 23rd Jan. 2025).

[8] Ken Schwaber and Jeff Sutherland. *The 2020 Scrum Guide^{TM}*. URL: https://scrumguides.org/scrum-guide.html (visited on 17th Jan. 2025).

[9] Seagate. *What is a 3-2-1 Backup Strategy?* URL: https://www.seagate.com/gb/en/blog/what-is-a-3-2-1-backup-strategy/ (visited on 27th Jan. 2025).

[10] Skogkurs. *Forestry Extension Institute*. URL: https://skogkurs.no/english/ (visited on 21st Jan. 2025).

[11] Skogkurs. *Om Skogkurs*. URL: https://skogkurs.no/om-skogkurs/ (visited on 20th Jan. 2025).