Flagship

Main Report
Team 1
Version <1.2>

Revision History

Date	Version	Description	Author
16.04.2020	1.0	Initial draft	Mats Sollid Eide, Lars-Håvard Bråten
17.04.2020	1.1	Continuous work	Lars-Håvard Bråten, Arvid Kirkbakk, Mats Sollid Eide
18-24.04.2020	1.2	Final version	Lars-Håvard Bråten, Mats Sollid Eide, Karl Labrador, Eivind Berger-Nilsen, Arvid Kirkbakk, Robin Christoffer Vold

Table of Contents

Main Report

This report is the outcome of the project assignment given to us in the course IDATT1002 Software Engineering at the Norwegian University of Science and Technology in the spring semester of 2020.

The objective of the project assignment was to build a photo management application based on Java and familiarize the student with development tools and technologies, as well as giving the student practical experience in working in a team and implementing knowledge from previous programming courses.

In the report, we write about how our teamwork progressed and look at how the result of our product is in accordance to our early vision. We also reflect on our effort and how we could have done things differently, and further envision how our product could be developed after the assignment, and look at new technologies that we have learned to use to meet the requirements of the assignment.

The team would like to thank our student assistant, Morten Nordseth, for guidance and help throughout the project process.

We would also like to thank the participants of our Wireframe tests, Maria Brandbo, Åsa, Jørn and Simon Vold, Mali Storbækken and Malin, for giving the team great feedback to further improve the user interface.

Karl Labrador | Lars-Håvard Bråten | Mats Sollid Eide

Eivind Berger-Nilsen | Robin Christoffer Vold | Arvid Kirkbakk

2 Introduction

2.1 Purpose

The purpose of this report is to document and evaluate the project process. It is meant to give a thorough, convenient and easy overview of the project, assignment and clients, in addition to describing the process, tools, and technologies used to complete the project.

2.2 Scope

This main report document is associated with a project in the course IDATT1002 Software Engineering at NTNU and Team 1 (“Flagship”). It is limited to this project and team only.

2.3 Definitions, Acronyms, and Abbreviations

Term	**Definition**
SQL	Structured Query Language. Used to communicate with a database.
MySQL	A relational database management system.
Hibernate	Object-relational mapping tool
IDATT1002	The course “Software Engineering” at NTNU.
JavaFX	Java library for creating a GUI for our application
GUI	Graphical user interface
Metadata	Text information that is embedded into an image file.
NTNU	Norwegian University of Science and Technology
OpenStreetMap	Open source, editable world map
Trello	Online kanban style planning software

2.4 References

Apache Maven:

- Maven – Welcome to Apache Maven

Apache Maven JavaDoc plugin:

- https://maven.apache.org/plugins/maven-javadoc-plugin/javadoc-mojo.html?fbclid=IwAR3aW28OcnBza1zKtjwIWZA5uUUJ9V9Kqg4C-N-CQHgnkSSMyzGiQ41sKGc

Drewnoakes Metadata Extractor library:

- drewnoakes/metadata-extractor: Extracts Exif, IPTC, XMP, ICC and other metadata from image, video and audio files

GitLab page:

- https://gitlab.stud.idi.ntnu.no/flagship/app-product/

Hibernate API documentation:

- Overview (Hibernate API Documentation)

JavaFX 11 API documentation:

- https://openjfx.io/javadoc/11/

JDK 11 API documentation:

- https://docs.oracle.com/en/java/javase/11/docs/api/index.html

OpenStreetMap API documentation:

- API v0.6

WCAG 2.1 Perceivable:

- https://www.w3.org/TR/WCAG21/#perceivable

Fetching UUID Windows & Mac:

- https://stackoverflow.com/questions/49488624/how-to-get-a-computer-specific-id-number-using-java

Fetching UUID on Linux:

- https://www.includehelp.com/java-programs/method-for-get-system-uuid-for-linux-machine.aspx

2.5 Overview

The report will start with a description of the assignment. It will then go through the use of hardware, software, literature, and collaborative tools in addition to explaining the distribution of the workload on a broader level.

Later, an overview of the documentation is presented before the implementation of the project is explained. In this section we give a description of the project process before reflecting on it. The section also contains an evaluation of what we have achieved relative to the goals stated in the Vision Document, as well as an assessment of how the final product complies with Don Norman's principles of interaction design and Universal Design. We will subsequently comment on how the process compares to our project plan.

In the final section we will compare our risk analysis in relation to actual events from the project before discussing further work with assumptions & dependencies, a product perspective, and an explanation of the program's features. Following this, a summary is presented along with a link to the GitLab repository.

3 Assignment Description

This assignment is part of IDATT1002 Software Engineering, at NTNU. Our assignment was to create an application from the ground up, with the Java programming language, for uploading image data and browsing images via a MySQL database. Additionally, we were tasked to create map functionality for the application. Furthermore, the application was given a set of requirements that had to be fulfilled. These are:

1. The application has to be a standalone Java RE application.
2. It needs to use the MySQL server that our University provides and ORM technologies as taught by our courses.
3. Use a single connection pool, only adding connection pools if we're running multiple threads.
4. Test **at least** all the classes that read metadata with Junit.
5. Relate to or follow Don Norman's principles of interaction design.
6. Carry out usability tests after the first and second iterations.
7. Application must be designed according to Universal Design principles.

Our client is a fictional computer consulting company commissioned to create a system according to these specifications, represented in our case by Grethe Sandstrak.

4 How the assignment was solved

4.1 Methods and standards used

Java Development Kit 11.02, JavaFX,

Programming Environment: Java Development Kit 11.06, JavaFX 11.02 and JPA.

Standards: Universal Design and Don Norman's Principles of Interaction Design, WCAG 2.1 (perceivable)

4.2 Use of literature and Internet

The literature the team has taken advantage of in this project mainly consists of the documentation of the various APIs that have been put to use. These APIs include: Java 11 API, Hibernate API, OpenStreetMap API, JavaFX API, Drewnoakes Metadata Extractor library, and the iText API.

We have made some use of online information for our program and its dependencies.

- Source on fetching UUID on Windows and Mac: https://stackoverflow.com/questions/49488624/how-to-get-a-computer-specific-id-number-using-java
- Source on fetching UUID on Linux: https://www.includehelp.com/java-programs/method-for-get-system-uuid-for-linux-machine.aspx

4.3 Overview of hardware used

- MySQL server provided by the university.

One of the project requirements was to use NTNU's servers, as provided by the client.

- Personal desktops and laptops

There were no other special hardware needs for this project.

4.4 Overview of software used

JetBrains IntelliJ IDEA: an integrated development environment for developing computer software.

Saros: A plug-in for IntelliJ IDEA (and other IDE's) which allows two or more users to edit the same project in real time.

Balsamiq WireFrames: A low fidelity wireframing tool.

Google Drive: Online file-sharing system.

GitKraken pro: An alternative to GitHub Desktop, a downloadable Git-client.

Trello: Online kanban-style planning tool. Used for task delegation and progress tracking.

Discord: Desktop text-and voice chat client. Used to conduct meetings and communicate.

4.5 Distribution of workload

The team aimed for an as equal distribution of workload as possible but most members had several fields of focus to reach a deadline and not hamper the process in any major way. When the team members had spare time, it was spent helping the other team members so that everything would be completed in a timely manner.

- **Eivind Berger-Nilsen:** In charge of GUI through JavaFX. Primarily paired with Robin Vold.
- **Robin Vold:** Metadata Extraction, map functionality and a minor role in GUI programming.
- **Karl Labrador:** Team leader. Had a role in all aspects of the programming process. Main fields were Hibernate and MySQL database connection and communication.
- **Arvid Kirkbakk:** Main fields were Metadata Extraction and PDF creation. Minor role in Hibernate and MySQL database communication and testing.
- **Mats Sollid Eide:** Main roles in PDF creation and formatting. Minor role in Hibernate and MySQL communication and testing.
- **Lars H. Bråten:** Chief of Wiki and Documentation, producing class and sequence diagrams, and automatic Javadoc Pipelines. Minor role in database management, DAO, Logging and Hibernate. Paired with Karl

4.6 Overview of documentation

In addition to this main report, several other documents will also be delivered. This includes a Collaboration agreement, Gantt chart, Timesheets with status reports, Meeting invitations and minutes, a Vision Document, and a link to the JavaDoc on GitLab pages. In addition to this, the team has also made a fully fledged wiki page with diagrams, wireframes, user tests, explanations and installation/user manuals.

5 Implementation of the project

5.1 The process of realizing the project

5.1.1 The development cycle

We will start this section with a brief overview of how the team proceeded while carrying out the project. The project was carried out through an iterative development cycle. It consisted of three iterations:

- The first iteration laid the groundwork for the upcoming software construction. This iteration included constructing a simple wireframe model resembling our planned product. We also wrote the Vision Document for our project during this period, in order to clearly communicate to our customer the requirements our application would fulfill. A domain model was also developed to visualize the general structure of our planned application. The team additionally outlined a project plan in the form of a Gantt chart, so that a manageable plan could be followed throughout the project cycle. During this iteration we also set up the environment for our version control system, using GitLab, and also decided on other collaboration tools like Trello and Google Drive.

- The second iteration consisted of constructing a minimal viable product (MVP). During this iteration, functionality was prioritized over design, and core functionality over optional functionalities. Our MVP ended up not making use of any database communication, as it had proved problematic in the days leading up to our demonstration. Documentation was handled in parallel with the software construction.

- The third and final iteration was mainly about finishing the application and all related documentation and attachments to be delivered. In our case this included implementing certain features lacking from the MVP, such as searching, album collections, and integration of database communication. It also meant revisiting existing code and documentation to ensure a polished finish in all the aspects of our final delivery.

5.1.2 Meetings

In order to secure a stable and steady workflow, the team decided from the very first day that having all team members meet regularly would be crucial. Several of the NTNU campuses and buildings served as meeting points, including Realfagsbygget, Akrinn and Lysholmbygget. Meetings were both regular and irregular. The regular meeting took place every Monday and consisted of formal updates on how everyone was progressing. Additionally, the team would meet two or three times a week to work together on the project. The timing for these less formal meetings were usually decided during the prior meeting.

The team also conducted meetings with our study assistant Morten Nordseth. During these meetings the team received valuable feedback on various tasks we were occupied with at the time. Furthermore, he has helped clarify details surrounding a handful of the documentation to be delivered.

5.1.3 Work delegation

The team members were generally free to choose what they wanted to work on. Once an early agreement had been made on which aspects should be worked on, each team member could select the task they were most motivated to tackle. Using issue boards further emphasized the relatively free nature of the task delegation. After the team members had been working with their separate aspects for a while, some team members were swapped around in order to familiarize themselves with other features. In these transitional phases, those who previously had previously worked on a task would help out those taking over.

5.1.4 How the coronavirus impacted the project

The national breakout of the coronavirus started seriously during the fourth week of the project workflow. We realized at this point that further meetups were out of the question, and multiple team members decided to travel back to their homes at this stage. The solution for keeping contact without conducting meetings in person, became the text and voice chat service Discord. All in all, this switch did not hinder the team communication considerably, but it did change the nature of our meetings somewhat. After the lockdown started, the team had some difficulties delegating tasks. This was solved by creating workgroups, and utilizing the IntelliJ add-on Saros to work together. Saros enables you to work together at the same time in the same files, similarly to how Google Docs works.

The guidance meetings with the course lecturer were also affected by the outbreak, and the second meeting was carried out through Blackboard Collaborate. Our experience with this type of meeting was generally positive, and we have the impression that this alternative solution was not a detriment to the project in any way.

The project aspect which received the greatest impact from the virus outbreak situation was the usability testing. The testing for the first iteration, i.e. the wireframes, was fortunately completed before the outbreak. We were also going to conduct a usability test of the MVP from the second iteration, but the MVP was completed while the coronavirus outbreak was in its most devastating phase. In order to take the necessary security measures with regards to public health, we decided to not approach any test subjects with our MVP. This meant that we had to move on to the final stage of production without the valuable user feedback such a test provides.

In conclusion, the outbreak of the coronavirus had little impact on the communication within the team, beyond the short phase of putting the new communication tools to use. It may however have had an effect on the final product, considering the lack of feedback resulting from the cancellation of the second usability tests.

5.2 Reflections on the project process

5.2.1 What we are content with

Team members have to a high degree been present throughout the project. People have attended meetings and put the necessary work hours into their delegated tasks. This may be seen in light of our penalty system laid down in the collaboration agreement, where penalty points have been sparsely incurred due to the high level of conscientiousness in the team. In return, this has resulted in a high level product, and has helped us complete all deliverables on time.

The usage of tools have been very successful as well. For instance, working in different branches within our Git repository has allowed us to solve multiple tasks in parallel, which has been working out flawlessly. Utilizing Maven as our project management tool has also saved the group a lot of hassle. Furthermore, both Trello and our tidely organized GitLab Wiki page has given the project a level of overview that we are highly content with.

The use of the Add-on Saros has also proven very successful. This allowed us to work in pairs in real time, as if two people were working on one computer. By doing this, we effectively negated the impact social distancing had on our team work.

5.2.2 What we could have done differently

Perhaps the most important aspect of a project of this character, is to gather valuable experience about the process. This includes making some acknowledgements about what could have been differently, and which we will likely take into consideration when working on future projects.

The different guidelines for achieving Universal Design could have been followed more strictly from the very beginning of the project. While the team always had the idea of Universal Design in mind while developing the application, the concrete guidelines published by the Center for Universal Design and the Web Accessibility Initiative were not emphasized until the later stages of development. Ensuring sufficient compliance with these guidelines required reworking some of the code, which would have been avoided if they were followed from day one.

5.3 Evaluating the achieved results

5.3.1 Achievements in relation to project goals stated in the Vision Document

- Efficiency goals:
  The Vision Document states that we wish to deliver a product in accordance with the specifications agreed upon with the customer. Our evaluation is that the product we have developed achieves the requirements outlined in the Vision Document, and thus also this goal.

  As a team goal, the Vision Document declares that we wish to stay ahead of deadlines by making the right prioritizations. When it comes to the formal deadlines, the team has always managed to finish the required tasks on time and in a proper manner. It also mentions that the team should seek to prevent excessive last minute workload. The team has generally succeeded in this regard, with the Vision Document as a small exception, where some extra work hours had to be put in during the final day (and night) before delivery.

  The order of priorities listed in the Vision Document was followed to a significant degree, and helped the team implement the necessary features before their respective deadlines. For example, the MySQL connectivity, which is listed as the number one priority, came a long way already on the first day. The map feature, which had a lower priority, was not properly implemented until after the MVP had been finalized. Sharing images through social media appears on the bottom of the list and did not end up in the final product. This prioritization was necessary in order to properly implement the more important features.

  In addition to the formal deadlines, we also implemented deadlines of our own once we employed a sprint-style workflow. Each team member did for the most part finish their delegated task by end of each sprint. These deadlines were less strict, as uncompleted tasks were simply moved over to the next sprint.

  Furthermore, the team set out to gain experience and knowledge throughout the project. This goal has been achieved by every team member. The various technologies and techniques the team has increased their knowledge with include object relational mapping, sophisticated testing, pipelines, JavaFX using FXML, Maven, iText, and additional third party APIs. The team has also gained

valuable experience with the development process. This includes team organization, using a version control system, and completing a broad range of documentation.

The Vision Document furthermore states that all members should be kept updated on knowledge and information in each field. This goal was not achieved to the same degree that we initially envisioned. Team members generally only had a surface level understanding in the field they had not been directly involved with. For instance, those who worked primarily on testing knew little about the details of the map implementation, and vice versa. The goal may have been too ambitious or poorly formulated. While it is true that each member has a good overview of the different aspects of the production, the knowledge gain has been in clear favor to their own set of working areas.

Keeping the group morale high was our final team goal, and all of the team members agree that this has been achieved and have had a positive experience in this regard.

- Result goals:
  Our primary result goal was to make a working application in accordance with the requirements. This goal has by our evaluation been met. We also stated in the Vision Document that we aim to achieve the grade A. It remains to be seen whether this objective will be met. The final result we set as a goal was an increased wealth of knowledge. By the end of the project, every member had achieved this by familiarizing themselves with the various technologies that were utilized throughout the project.

- Process goals:
  The team set themselves a handful of goals relating to how the project process would be carried out. We aimed to use Hibernate as our ORM technology, and this was indeed implemented successfully. We also managed to stick to our preferred JDK version, JDK 11. The team built a working database communicator in the sense that an API layer using Hibernate was implemented as a bridge between the database and the front-end program. Finally, we set ourselves the objective of creating an image metadata reader in Java. Instead of hard coding this utility ourselves, the team made use of an existing API for this purpose.

5.3.2 Evaluation with regards to Norman's principles

At the beginning of the project, the team set themselves the goal of creating a product in accordance with Don Norman's six principles of interaction design. This section discusses whether this has been achieved, by evaluating the application with regards to each of the principles.

- Visibility:
  The application maintains a clear overview of all the features throughout the process of using it. That is, every major feature can be accessed through a single click, rather than being hidden behind multiple layers of menus. Whether the user is using the grid view, the map view, or looking at a single image, the other tabs remain available and highly visible. This keeps the user from getting lost and missing out on important functionalities.

  When it comes to communicating what the available features will do when interacted with, the application takes great advantage of knowledge users already have by convention. For example, most users will immediately have a good idea of how the search bar is used. Tab layouts are also something users are familiar with, and most of the buttons display text which efficiently describes their purpose. The plus and minus buttons used for zooming in and out on the map are another example of clear and concise communication through means of convention.

  All in all, the program scores well both when it comes to keeping features visible and clearly communicating their purpose.

- Feedback:
  Because the application is built with JavaFX, many essential forms of feedback are already available by default. Buttons, tabs, search bars, and row elements are all highlighted when selected. Subtle feedback like this makes the application feel responsive.

- Constraints:
  There are currently a handful of constraints in the program. Clicking on an image will open it in a new tab, but the user is prohibited from making duplicate tabs by clicking on the same image multiple times. The zooming functionality in the map

view is also constrained. If the user zooms far enough in or out, the zoom button will grey out and prevent unreasonable zoom levels (although the zoom out button could grey out earlier).

In addition, the program puts a constraint on which file types the user can select when adding images to the application. Only a list of image file types will be accepted. This prevents users from adding files with no practical use in the program, which could potentially cause confusion. Besides preventing confusion, it prohibits incorrect use of features not supporting non-image files, such as the PDF generator.

The implementation of these constraints increases the likelihood of the application being used as intended.

- Mapping:
  The program is controlled through simple and conventional methods, where the effect of each action is expected and well established. Using scroll bars or the mouse wheel to move up and down is intuitive enough that users do not have to think about how they are controlling the application. Navigating the map by dragging it around also maps well, and is a standard way of map navigation.

- Consistency:
  The application carries a simplistic and consistent design. The buttons performing image-related actions look the same, and the different tabs share the same design. It is therefore easy to understand which elements function as buttons, and which are tabs. Having many different button and tab styles would clutter this seemingly obvious differentiation between the different element types. Other minor elements, such as scroll bars and tab drop-down menus, keep a consistent design regardless of where they appear in the program.

  Important features break with the consistency in order to make them more visible. This is the case with the option to add images, which in addition to being present in the right-click menu in the file overview, also appears in the top menu in a more visible manner. Making key features more distinct is beneficial to the user experience.

- Affordance:
  The appearance of the program closely matches what it can do. Having a high degree of visibility of all the system features makes the user aware of the different actions that are available to them at all times. To further improve the affordance the application could be launched with some images already included. This would give users the immediate impression that this is an app for dealing with images, and give them an idea of how the different features work.

5.3.3 How the product complies with the principles of Universal Design

The team has taken necessary considerations to ensure a product which is available for comfortable use for users with different abilities and prior technical knowledge. Both the 7 principles of Universal Design and the Web Content Accessibility Guidelines (WCAG) 2.1 principle 1 - Perceivable. The specifics on how these guidelines have been abided are documented on our GitLab Wiki pages.

5.4 Achievements in relation to the project plan

At the start of the project, the team outlined our project plan in the form of a Gantt chart. The chart displays the planned duration of all the parts which make up the project. In this section we will evaluate whether the project plan matches the actual duration of the different tasks.

The various activities listed in the Gantt chart have generally been carried out during their planned time frame, with deviations being at most a couple days off schedule. The exception to this is the usability test for our MVP, which was never performed due to the public situation regarding the coronavirus outbreak. By actively using our project plan throughout development, we have maintained a steady and controlled workflow during the entire process.

5.5 Reviewing our risk analysis

Our Vision Document lists a number of events which could potentially cause trouble during development, and which measures would need to be taken in the event of their occurence.

The risk with the highest evaluated probability was about potential difficulties implementing Google maps. This risk came true, as the team had to scrap the use of GmapsFX in favour of OpenStreetMap due to difficulties implementing it. This cost us some time, but we still managed to complete it because the team already had a set plan on what we should do in case of such an event. Luckily, none of the other risks came to fruition.

Overall, the evaluations the team has made during the risk analysis appear to have been well grounded.

6 Further work

6.1 Product perspective

We envision this product to have a wide array of different applications and useful situations. It can be used as a tourism app, allowing people to tag the areas they visit and increasing exposure for landmarks and locations most people never even knew existed. With proper integration, it can be used for reporting events, allowing people the chance to look through time and space to the exact date and location of any given event as long as it has been reported to the system. With a wider audience, this application will only get better, more expansive with potential global ramifications.

6.2 Product functions

This product would benefit greatly from Cloud Services. We also wanted to implement a collection marker, where you could mark up an area and return the exact number of images in one location. Sharing to social media directly from the app is also something which easily could be implemented.

6.3 Assumptions and dependencies

The application is intended for use on desktops or laptops running Windows, MacOS or any kind of Linux distro. The program requires the computer's UUID to work. We implemented the functionality for all three platforms, but as of now, we can only guarantee that the program will work on Windows computers, as we have not been able to test this on computers running MacOS or Linux. The program also requires Java and a stable internet connection.

7 Repository

Link to GitLab Wiki:
https://gitlab.stud.idi.ntnu.no/flagship/app-product
Link to GitLab pages JavaDoc:
http://flagship.pages.stud.idi.ntnu.no/app-product/apidocs/
The program does not require any usernames or passwords to function. All you need is a computer with Java installed and a stable internet connection.

8 Summary

In essence, this assignment is part image management application and part database functionality. This project has taken us through all phases of systems development except maintenance. Planning - Analysis - Design - Implementation - Testing and Integration.

Additionally, it gives us three fields of study relevant to our course. Image/file data extraction, ORM technology through Hibernate and GUI creation with JavaFX. It allows us to develop our skills as programmers as well as engineers and managers.
In addition to these technical fields, we got the opportunity to work in a team in a simulated professional environment, which is a great way to prepare the team for employment at the end of our studies.

It is said that a great engineer is not only skilled in his field of expertise but he also provides clear and thorough documentation. Through this project we get to see a glimpse of how much work goes into the creation of a system, not just the finished product. From actual coding and design to a rigorous documentation process, this project has been both strenuous and a valuable learning experience.

It also teaches us compassion and consideration. When you are working to develop a system in a team, not only do you need to be mindful of your companions and the clients wishes, as well as potential future clients. The needs of your team must be met as well as your clients. And one of the best ways to show consideration is to provide detailed and sufficient documentation and testing for as many variables as is feasible. Doing this the correct way helps building a great work ethic and good workplace habits. The teamwork was excellent, with every member being available and willing at a moments notice. We followed our plan and exceeded it to some extent, and problems were met with swift and decisive action. All things considered, the team is happy to finally be finished with this project, and even though there are things that could have been done differently, we can all say that we are delighted with the end result.