IDATG1002 - Software Engineering

# MetIma

*Authors:*
Mikkel Aas

Magnus Johan Gluppe

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

April, 2020

# Table of Contents

# 1 Preface

## 1.1 The Study Program

The study program is concentrated around software engineering. You learn how to make user friendly and secure software for the internet and mobile devices. You also learn about good work habits by using modeling, testing and high code quality.

It is a practical study program. The theoretical knowledge you learn is quickly utilized in practical problem solving. During the study you will cooperate with peers to develop large scale computer system.

## 1.2 The purpose of the project

The purpose of the project was to put theory into practice. During the course we learned about object oriented analysis, unified modeling language, agile development methodology, vision and requirement analysis, software architecture, and many other subjects. The project allowed us to learn more about those subjects by using them to build a software. We also learned how to share information and knowledge with team members and a client.

# 2 Introduction

## 2.1 The purpose of the report

The purpose is to give the reader information about the development process of the application. The intent is for the reader to understand how the project was solved. This includes the whole project from start to finish, including the requirements and thought process behind the decisions that were made during development.

## 2.2 Scope

This report contains the information about the development process for the application MetIma. The name MetIma is a combination of the words *metadata* and *image*. The application is primarily an imaging application which reads the metadata of images and saves it in a accessible database. This report covers all the different elements from the process, as well as how the process was structured.

As for the application itself, and its position in the market, MetIma is part of a very competitive space. With already well established similar applications like Picasa and Photoscape, MetIma will have a hard time acquiring users that already use these applications. At the same time, one of MetImas differentiating features is that it is very lightweight and tailored towards users that prefer a simple to use interface. This could entice users to switch over to MetIma if it offers all the functionality they need in a software like it. MetIma also has the advantage of being commissioned by a customer, so it has a guaranteed user base.

### 2.3 Definitions, Acronyms and Abbreviations

In the report, there are a lot of terms used that might not have an obvious meaning by themselves. Therefore, a list of these terms and their explanations have been included:

- IDE - **I**ntegrated **d**evelopment **e**nvironment.
- NTNU - Norwegian University of Science and Technology. (**N**orges **t**eknisk-**n**aturvitenskapelige **u**niversitet)
- WCAG - **W**eb **C**ontent **A**ccessibility **G**uidelines
- MVP - **M**inimum **v**iable **p**roduct
- GUI - **G**raphical **u**ser **i**nterface
- VoIP - **V**oice over **I**nternet **P**rotocol
- DevOps - Software development (Dev) and Information-technology operations (Ops)
- ORM - **O**bject-**r**elational **m**apping

### 2.4 Overview

This report explains what the project is and how it was solved. There is also additional information regarding the application, and its functions. The report goes into detail about how the development team collaborated and solved problems during the project.

## 3 Assignment Description

The assignment we received was to create an imaging application which read the metadata of an image, and stored it in a database. The user had to be able to register images, and when images are added, their metadata is read and added to the database. The user also had to be able to search based on metadata and on the tags added by the user. Lastly the user had to have the possibility to create a PDF photo album. The goal was to develop this application from scratch.

The client which has given the assignment is Evil inc. They needed an imaging application which reads the metadata of images.

The technical requirements the application had to fulfill is as follows:

- Must be a standalone Java application.
- Use the MySQL database at the university and use ORM technology.
- Use a connection pool with one connection.
- Test at least all the classes that read image metadata with JUnit.
- Has to follow the Web Content Accessibility Guidelines.

The project also had some process requirements: The project had to be carried out in three iterations.

1. The 1st iteration focused on the vision and the requirements. A wireframe of the application had to be made and usability testing had to be done. The prototype and the vision document had to be provided in the 1st iteration.

2. In the 2nd iteration the focus was placed on the MVP. This was a prototype with minimal functionality. This prototype was also usability tested. The first version of the GitLab WIKI was also made.
3. In the 3rd iteration the focus was on finishing the application and all documents.

# 4 How the assignment was solved

## 4.1 Use of Literature and Internet

A programmer has a variety of different tools at their disposal. The most used website has been Stack Overflow. It is a discussion forum where you can post code related questions and have other programmers answer and provide examples. [1] To ensure this project goes smoothly the development team was sent to a Software Engineering course. For the projects structure, vision document and the report, the team has used the resources made available to us through this course. This includes the lectures we attended and the multitude of templates provided by the lecturer for this course.

## 4.2 Project Responsibilities

The work on the project was loosely divided between the four team members. Each team member had an area of responsibility: Mikael was the group leader, Magnus had the responsibility of organizing meetings, Jakob was responsible for the documentation and Mikkel was head of design. However, when it came to developing the application itself, the roles were more fluid. Everyone collaborated and worked in a group, what you worked on was mostly decided in that work session. The one constant was that Mikkel has focused on the design aspects of the project.

## 4.3 Methods and Standards Used

Due to the structure of the project we chose to use Scrum. This is a very agile model with a high amount of customer involvement. This was done by having several partial deliveries of the application for review. While the requirements of the project were not changed, this provided us with valuable feedback on the progress of the application. This made it possible for the development team to make adjustments and focus on working software.

For organizing the project we used the Unified Process, or UP. The Unified Process is structured in a series of faces, each which is complete when you reach a milestone. An example of this is when the MVP was done, or when the application was connected to a database. Phases can overlap, and often do, as not all group members worked on the same task at the same time.

## 4.4 Brief description of standard software used

- Java 11.0

  The entire project was written in Java 11.0. Java is a class-based and object oriented general purpose programming language. The language is designed to have as few implementation dependencies as possible, and it runs on every platform that support Java. [2]

- IntelliJ IDEA

  The IDE of choice was IntelliJ and all group members used this development tool. IntelliJ IDEA is an integrated development environment for developing computer software. It is developed by JetBrains and written in Java. It features tools like coding assistance, that improve efficiency while coding. [3]

- GitLab

  The collaboration was done using GitLab, where we all shared a project. GitLab is a web-based DevOps life-cycle tool that provides a Git-repository manager. The Git-repository manager has features such as issue tracking, and continuous integration and deployment. [4]

- Javafx Scene Builder

  For GUI we opted to use Javafx Scene Builder instead of writing fxml documents from scratch. Javafx Scene builder is a software that lets the user create fxml files by dragging and dropping controls from a palette.

- BalsamiQ Wireframes

  When creating our MVP we based the design on the previously made wireframe. The wireframe was made using a software called BalsamiQ Wireframes. BalsamiQ Wireframes is a graphical user interface website wireframe builder application. It contains pre-built widgets that the user can arrange using a drag-and-drop editor. We chose to use the Google Drive version of the application because it was the easiest to share.[5]

- Overleaf

  We used overleaf to write all the documents. Overleaf is an online Latex editor. It is a modern collaborative authoring tool that helps make science and research faster, more open and more transparent. [6]

# 5 Implementation of the Project

## 5.1 Predefined Goals

Before the project commenced, there were three main goals in the vision document, as well as a few smaller goals. There were three main goal categories; efficiency, result, and process. While the efficiency and result goals partially overlap, they were important enough to keep separate.

- Efficiency goals

  The application should enable an easier method of organizing a digital image collection. By giving the user an automated way to have their images easily search- and browse able, it will be easier to work with a large collection of images. The application will also be easy to use for all users no matter their computer proficiency.

- Result goals

  The main result goal for the MetIma application was to increase the productivity of the user. Both for storing and accessing digitally stored images. The less time a user has to spend doing mundane tasks, the better. The other result goal is increased earnings in a business environment where "time is money". While this overlaps with the main result goal, it is more of a commercial goal. A user can spend more time working in a business and earning money than organizing images, which usually does not net a lot of money for businesses.

- Process goals

  A goal during the process was to improve the developers competence. By using new libraries, tools and strategies, the developers would improve their personal development skills. This newfound skill is useful both in future projects and later in this project if code needs refactoring. Since this was a group project, the collaborative part of the process was also very important. The developers had to work closely both with each other, and potential users for usability testing. This provides experience that will be very valuable in the future.

## 5.2 Evaluation of Final Result

### 5.2.1 Goals

- Efficiency goals

  The final application very much fulfills the efficiency goals that were set. All images added to the application are easily searchable. The application can display results for search terms based on image name, user specified tags, and automatically extracted metadata. While the tags must be specified manually by the user, the application will automatically choose an image name based on file name, save metadata in a readable format for the user to access.

- Result goals

  The result goals overlap somewhat with the efficiency goals, and therefore are mostly fulfilled as well. The application allows the user to add many images at once, and set names and tags for all of them before they are added to the gallery. It therefore makes it much quicker for the user to later access these images, which fulfills the goal.

- Process goals

  While writing the application, a lot of different solutions and libraries were tested before ending up with a final solution. By trying out and testing different solutions, the developers learned several skills and obtained information about possible solutions that, while not in this project, can be very useful elsewhere. Collaborative skills from working with solutions like Git were also greatly improved, as Git was used extensively throughout the entire project.

### 5.2.2 Technical Requirements

As mentioned in section 3, "Assignment Description", the application had a few technical requirements it had to meet. The final MetIma application fulfills all of these requirements.

The application is written using Java and Java libraries. It does not require any other software to work, other than a database connection, which is not a part of the application itself. This follows the first requirement, which is "Must be a standalone Java application".

The next two requirements, "Use the MySQL database at the university and use ORM technology" and "Use a connection pool with one connection", were also both fulfilled. While it does not specify to only do this and not include other solutions, it was decided to not branch out too far from the requirements and do unnecessary work that was not required.

For code testing, there was only one requirement, which was to "Test at least all the classes that read image metadata with JUnit". This was done internally, and there are several tests run when the application is compiled to make sure it works properly before it's distributed.

The application had to be designed according to WCAG 2.1 principle 1, perceivable. This was done by implementing text alternatives for non-text content so that it can be changed into other forms people need, such as speech.

## 5.3 Problems and Solutions

During the development process there were several problems with the application that had to be solved. Some of these problems came from planned features that turned out to be hard to implement, and some came from refactoring and reworks that had to be done mid-development. These reworks or refactors were the result of solutions that weren't good enough, or a mistake in the planning that was not predicted beforehand.

The first big issue that was discovered was that the planned page for adding images was very ineffective and lacked options that would be very basic in an application like MetIma. With the planned design, it was not possible to more than one image at a time. This made the application really slow to use for a user that wanted to add a large amount of images.

A temporary and dirty solution to this was to allow the selection of more images, but the page design did not have room to set names or tags for the images. So all images added at the same time received the same image name and tags. While this was a solution, it was still not satisfactory. This led to a complete rewrite of both the design of the page and the controller, which resulted in more time spent than planned on this part of the application.

The second big problem occurred when setting up the application for working with a database instead of a temporary solution that was set up for testing purposes. For extracting the image metadata, a library called "metadata-extractor" was used. The library provides a class for storing and working with extracted metadata. Then original plan was to store this whole object in the database, which turned out not to be possible without rewriting most of the library.

While there were a lot of attempts made at trying to make the class serializable, it required so much work that it was decided to develop a different solution. The final solution ended up being a simple hashmap that stored the human friendly name of a tag and it's value. This meant some data loss, as the raw tag id was lost, but for our use case, that value was not needed and could be discarded.

The third and last problem that had to be solved was the amount of memory the application was using. Since the application has to display the images to the user, all these images have to be loaded into memory so they can be displayed when the user opens the gallery. In our application, when the user opened the gallery, the application would load images into memory and keep them there. If the user went back to the gallery after having been on a different page, it would load the images into memory again, even if they already were there and increase the memory usage.

The unsatisfactory solution to this was to create new image objects and redraw the images for each time the user viewed the gallery page. Before the fix, the images were supposed to be cached in memory, but our solution was unable to retrieve the cached images and kept trying to cache them over and over again. The solution makes the application slower the more images the user has in the gallery, but it makes sure the application does not use huge amounts of memory.

## 5.4 Evaluation of Risk

In the vision document some risk were laid out which could be detrimental to the success of the project. It was considered unlikely that the development team would exceed the budget and deadline for the launch of the application. We have been working on a comfortable budget and have had limited expenses. The main expense is salary, since the developers use their personal computers and development tools were supplied by the client. The development team managed to stay under budget because the project was completed earlier than expected. Another concern was planning to create something over-ambitious, that the developers could not deliver. The foundations we set in the planning phase proved to be solid, and the application is well within the realm of feasibility.

The risk that was considered most likely to derail the project was that the developers could not exclusively work on the MetIma project. This partially happened where the developers also had other responsibilities they had to focus on. This lead to the work was preformed in creative sprints, where large amounts of programming was done in a short span of time. Other times the project was left unattended for longer periods of time. This is reflected in the GitLab activity and time sheet. However,the project has been completed in a satisfactory way on schedule.

## 5.5 Teamwork and Workload

As mentioned in section 4.2, "Project responsibilities", all developers had gotten their own working areas and responsibilities. While the plan was to keep this separation of roles throughout the while development process, after a while the tasks started mixing and the roles became less clear. This was mostly because of our main process goal, which includes ”improve the developers competence” and it seemed in the developers best interest to try their hand at different tasks as long as it did not have negative consequences on the product.

## 5.6 Evaluation of Teamwork

The teamwork aspect of the project has been on of the aspects that has worked the best. There has been no personal conflicts, and all the developers have worked well together. For the most part the work has been conducted in a group setting, where all four developers program at the same time. This made it possible to consult with each other and collaborate on tasks. Since everyone's responsibilities started mixing, it made the most sense to work in a group. For future projects we want to focus more on individual work, where a developers does not feel forced to work at a certain time, and work more at their own pace. In this case more defined tasks would make it easier to work individually.

The teamwork for this project has been conducted in a very informal manner. All the developers on the development team are familiar with each other, and are on good terms. This has positive and negative aspects. While the project has gone very smoothly, the group has ignored many aspects of the planning process. This was because we were in frequent contact with each other We simply contacted each other on Discord and initiated a work session. Milestones and goals have been met, not because of proper planning, but because the group happened to work enough.

During the development of the application the development team utilized different collaborative tools to coordinate the workflow. GitLab and Overleaf was mentioned in section 4.4 ”Brief description of standard software used”

- Google drive

  Google drive is a freemium cloud storage software which is integrated with your google account. [7] This was useful for storing non overleaf documents like our collaboration agreement. Google drive is organized such that everyone shares a cloud storage, and can access the same documents.

- Discord

  We used discord as our main communication platform. Discord is a freeware VoIP application designed for video game communities. It specializes in text, image, video, and audio communication between users in a chat channel. [8]

Due to complications the group was unable to meet up in person, therefore these collaborative tools have been essential for completing the project.

# 6 Further Work

## 6.1 Product Perspective

### 6.1.1 Necessity

The need for the product was recognized when our client contracted us to develop it. We decided to provide the product as the scope seemed manageable based on our previous experience. The scope of the product was an imaging application that could read image metadata and sort images based on tags, filenames and metadata. It was going to be a GUI based desktop application with a database connection. We gathered the requirements for the product based on the scope and began inventing.

### 6.1.2 Invention

The product was designed based on the requirements. We made a domain model that covered the functional aspects of the program and a Wireframe for the GUI layout. The application was then created based on its design. We made tests to ensure that the program worked as intended under the development phase. However, the tests were mainly focused on the data handling.

### 6.1.3 Release

The product is going to be packaged for outside release as a jar file. The product will go through further external testing which will be done by Evil inc. The product will be handled off to an operational.

### 6.1.4 Support

The product may be supported by a new operational group and a user service group. And the code may be developed further and kept up to date be these new groups.

### 6.1.5 Desupport

Since this product is in a very competitive field, there may be no need for this specific application. Therefore there is a chance the product will be phased out in favor of better applications made by larger corporations.

## 6.2 Product Functions

This product is an imaging application which saves image metadata. It provides functions for the user to save images and view their metadata.

The main features in the MetIma application is as follows:

- Gallery: shows all the images which are added in a gallery.
- PDF exporting: exports all images currently shown in the gallery.
- Metadata reading: the application reads the image metadata and stores it in a database. The metadata can be shown when the user clicks on an image.
- Search: the gallery can be searched based on tags, filename and metadata.
- Add images: the user can add multiple images.
- Remove images: images can be removed from the gallery.
- Edit images: the tags and filenames of the images are editable.

## 6.3 Assumptions and Dependencies

### 6.3.1 Assumptions

The application assumes that the user has some basic knowledge operating a computer. We assume the user has an interest in viewing meta data, this is a feature that is more useful for photographers and similar professions.

### 6.3.2 Dependencies

- This application relies on Java and JavaFX. The user needs to have java runtime environment installed on their computer
- The application also relies on the user being connected to the internet, this is to save the data to a database.

## 7 Repository

This is the link to the git repository: `https://gitlab.stud.idi.ntnu.no/group-2/imaging-application`

To run the program the password has to be changed in the persistence.xml.

The username is already entered, which is: jakobfk

The password is: yBQOt5I4

This is the link to the GitLab WIKI: `https://gitlab.stud.idi.ntnu.no/group-2/imaging-application/-/wikis/home`

This is the link to the GitLab pages JavaDoc: `http://group-2.pages.stud.idi.ntnu.no/imaging-application/docs/`

# Bibliography

[1] About - stack overflow, month =.

[2] Wikipedia contributors. Java (programming language), April 2020. Available at https://en.wikipedia.org/wiki/Java_(programming_language.

[3] Making development an enjoyable experience, April 2020. Available at https://www.jetbrains.com/idea/features.

[4] Empower your team to thrive remotely, April 2020. Available at https://about.gitlab.com/.

[5] Scene builder, April 2020. Available at https://gluonhq.com/products/scene-builder/.

[6] About us, April 2020. Available at https://www.overleaf.com/about.

[7] Margaret Rouse. What is google drive?, March 2018. Available at https://searchmobilecomputing.techtarget.com/definition/Google-Drive.

[8] Phil Hornshaw. What is discord?, March 2020. Available at https://www.digitaltrends.com/gaming/what-is-discord/.

# Appendix

## A Vision Document

# Vision Document

Mikkel Aas Magnus Gluppe Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

February 2020

## Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 26.02.2020 | 0.1 | First revision | Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog |
| 27.02.2020 | 0.2 | Second revision | Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog |
| 28.02.2020 | 0.3 | 1st iteration submission | Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog |
| 19.03.2020 | 0.4 | 2nd iteration submission | Magnus Gluppe, Mikael Krog |
| 21.04.2020 | 0.5 | Responding to feedback | Magnus Gluppe, Mikael Krog |

# Contents

# 1 Introduction

This document outlines the vision of MetIma, an image application. MetIma is a newly founded development team that has been subcontracted by Evil Inc. to create an image application. The goal of the project is to design and develop an application that gathers metadata and organizes images. Evil Inc. has large amounts of image data from their users, and need a convenient way to organize and store them. We aim to deliver a product that is easy to use, while still being sophisticated. User interaction and design are key values in our development team. The product should be designed with good programming principles in mind, and use modern programming philosophy. This will ensure that the MetIma application is not rapidly outdated and stays relevant in the market.

## 1.1 Purpose

This vision document outlines the the purposes and aspirations for our image application MetIma. Here we state what we wish to achieve and how we want to achieve it.

## 1.2 Scope

The MetIma application can have potential use for both consumers and private businesses. However, the scope of this project is to creating a functioning and well design application. We want MetIma to be usable on a wide variety of devices and operating systems. However, the development team is limited to four relatively inexperienced programmers and a timetable of 3 months, so the end product cannot expect a substantial market share or high levels of sophistication and polish.

## 1.3 Definitions, Acronyms and Abbreviations

- SQL - Structured Query Language
- NTNU - Norwegian University of Science of technology
- NOK - Norwegian Krone
- ORM - Object-Relational Mapping
- Inc. - Incorporated

## 1.4 Overview

The rest of the vision document describes what goals we have for this project, and how we intend to achieve them. It also contains risk analysis and cost estimations. The end product we aim for is described in the vision document, and the different requirements we have are included.

# 2 Positioning

## 2.1 Problem Statement

Digitally stored images are the norm today, and can be a struggle to sort and organize. When the images are stored in a file system, there is no easy native way to organize these images unless the user does it manually. This makes it cumbersome to store images digitally. This exact problem affects a lot of users. A solution to this problem is an application that can take these images and organize them automatically for the user, which is what MetIma will be able to do.

### 2.2 Product Statement

This product is made a corporation who had a demand for organizing their image data, but can be of use to any individual that stores images and photos digitally. Today, that covers almost every person in the developed world. Many of these individuals are in need of a way to organize these files, and that is where our application comes in. Evil Inc. currently owns the license for MetIma, but the application can be of interest for other businesses in the future. The application is connected to a database, but processing happens locally on your computer,

## 3 Project goals

### 3.1 Efficiency goal

We want our application to enable an easier way to organize digital image collections. By giving the user an automated way to have their images organized, it will be easier for them to work with their collection and find the images they need when they want them. The application will be easy to use for all users no matter their computer proficiency.

### 3.2 Result goals

The main result goal for this application is to increase productivity for the user. The user in this case is Evil Inc., a private business. Both for storing and accessing the digitally stored images. The users productivity is greatly dependent on the users technical proficiency. However, an indication of a reached goal is a significant speed increase in the time it takes for a user to organize a set of images and later to access a specific image.

The other result goal is increased earnings in a business environment. While this also partially included in the increased productivity goal, this goal is more of a commercial goal. In a world where ”time is money”, the less effort a user has to put in to doing mundane tasks, the better. A user can spend more time with their clients, or have the resources to bring in more clients. Increasing the number of clients, results in higher revenue.

### 3.3 Process goals

During the process we want to improve the developers competence in coding and development. By learning to use different libraries, tools and strategies the developers will improve their personal skills. They will be able to apply this newfound competence in future projects, and also in this project by refactoring code at a later stage in the development process.

Taking into consideration that this is a student project, there are a lot of positive learning experiences that can be obtained. We have to work closely with other students, getting to know them better and creating a functionally working environment. There are several new concepts in the realm of documentation and project planning, this can be vital for future projects in a business setting.

## 4 User description and Stakeholders

### 4.1 Market Demographics

MetIma aspires to reach a vast audience for our product, by using open source software and cutting edge technology. We strive to have the MetIma application work cross platform on all devices. This would make MetIma a very flexible product, anyone with a smartphone or computer can enjoy our product. However, as our company is a currently unknown startup, the initial launch will be on a smaller scale.

## 4.2 Stakeholders

The stakeholders of the project is the development team themselves and Evil Inc. The success of the application is vital for all the developers future careers. Evil Inc.'s stakes in the application is the daily operations of their business and

| Name | Description | Responsibilities |
|---|---|---|
| Developer | The developer is in charge of developing and maintaining the application | – Developing the product<br>– Maintaining the application<br>– Meet the user requirements<br>– Ensure the viability in the marketplace |
| Client | This is the person or organization that has commissioned this project | – Paying the developers.<br>– Give specific requirements for the product.<br>– Using the application in a responsible way. |

## 4.3 Users

| Name | Description | Responsibilities |
|---|---|---|
| User | End user. The employees that will be using the application in their work | – Getting familiar with the application, to be able to utilize it in an effective manner.<br>– Provide useful feedback for future improvements.<br>– Use the platform regularly so it becomes part of their routine and workflow.<br>– Maintaining security so no sensitive information is compromised. |

# 5 Product Overview

## 5.1 Product Perspective

Our product is independent and self-contained. It does not belong to a larger system. This product is an alternative to other photo gallery solutions.

## 5.2 Risk Analysis

There are multiple risks which could be detrimental to the project:

| Risk | Risk Probability |
|---|---|
| The team could overreach, and try to develop something unachievable, which would make our end product unusable. | unlikely |
| The developers may not be able to exclusively work on this project, this might cause delays. | Very likely |
| There is a risk of exceeding our budget. | unlikely |
| Not being able to finish the product before the deadline. | unlikely |

## 5.3 Estimated costs

The development costs of the project is all our developers salary. Our developers hourly salary rate is 1470 NOK. Each team member is estimated to spend 150 hours on the project. +/- 10%. We have four developers working on this project. This means our expected development cost is a minimum of about 800 000 NOK and a maximum of about 970 000 NOK. Location and development tools are supplied by NTNU. Therefore, we have no other estimated costs.

# 6 Product Features

This section will present the first draft of the functional features of the MetIma application.

## 6.1 The home page

The first page the user is represented with is the home page. The homepage contains three buttons and the title of the application. Since the two main features of the application are to view a gallery and add new images, we thought it would be natural to have these two options as buttons on the homepage. The third button on the home page toggles dark mode for the application.

## 6.2 The gallery page

The gallery page mainly consists of an area where all the pictures in the gallery are presented to the user. The second biggest feature of the gallery page is the search bar. The search bar allows the user to search for images based on tags, filenames, and metadata. The user will be able to export the search results to a PDF document with an export button. There will also be home button and an add new image button that grants the user quick access to their respective pages.

## 6.3 The add new image page

To add a new image the user either has to click the button on the home page or use one of quick access buttons that are placed in the top left corner. When the user clicks the button they will be prompted to choose either a singular image, or a folder of images on the computer. If the user chooses a singular image they will be taken to a page where they can set the filename and add appropriate tags. However, if the user chooses a folder, all the pictures will have the filename followed by a sequence of numbers, and they will all share the same tags. Before the image or images are added to the program, the user will have to confirm or deny the import by clicking either add or cancel. The page will contain relevant quick access buttons in the upper left corner.

### 6.4 The image-view page

The last and simplest page is the view image page. It encompasses two main features. The first is the ability to view the selected image. The second is the availability to view a list of the metadata of the image, which you can simply scroll through. This page will also contain quick access buttons.

## 7 Product Requirements

For this project we have a multitude of requirements. Including System requirements, technical requirements and documentation requirements.

### 7.1 System Requirement

The only real system requirement for our application is to have Java Runtime Environment installed on your computer. The application is written in Java 11, and therefore Java will be needed to run the program.

### 7.2 Technical Requirements

The technical requirements describes which functions the application has to have. The technical requirements for the application are as follows:

1. The application must be made as a standalone Java application.
2. Use the MySQL database at the university and use ORM technology the way it has been taught on the programming course.
3. Use a connection pool with one connection. This is to improve performance and avoid using too many database connections.
4. Test at least all the classes that read image metadata with JUnit.
5. The application has to follow Web Content Accessibility Guidelines 2.1 Perceivable.

### 7.3 Documentation Requirements

The project has multiple documentation requirements. These define what we must document during the development:

1. The project has to be carried out in three iterations.
2. The team must use collaboration tools as part of the project. Such as GitLab, Google Drive etc.
3. The team must submit a main report which includes evaluation of the teamwork, experiences dealing with the project-work and cooperation within the team.
4. The main report must contain the following attachments: collaboration agreement, Gantt chart, time-sheet, meeting invitations and minutes, vision document, link to GitLab WIKI pages and a link to JavaDoc on GitLab pages

## B Project Manual

# Project Manual

Mikkel Aas Magnus Gluppe Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

April 2020

# Collaboration Agreement for Gruppe 2

Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog

February 2020

## 1 Introduction

The Collaboration agreement is based on a collection of goals, role responsibilities, procedures and guidelines for interaction within the team. The agreement is developed and complemented by the team members with their own interpretations of what you mean by these and how to achieve the goals.

## 2 Goals

### 2.1 Effect Goals

1. To become acquainted, build trust in each other and increase motivation for the project there will be held regular study sessions, where the members will work as a team and help each other if any problems should arise. There will also be social gatherings to get more comfortable with each other.
2. Respect each other's opinions and suggestions.
3. Finish the individual assigned work, as to not slow down and / or cause problems for others work.

### 2.2 Result goal

The group should be able to hand in all deliveries on schedule. To achieve this, all members will have a clean overview of everyone's work, and the progress of each individual. Members can then see who has done what, and what remains to be done. There is also the option to assign more people to tasks if deemed necessary.

## 3 Roles and responsibilities

- **Project leader**
  Krog was elected project leader by acclamation. His responsibilities involve but are not limited to, delegating assignments, keeping the schedule, solving potential conflicts and the overall vision of the project.
- **Organization responsible**
  Gluppe has been given the assignment of organising the teams meetings. He will be responsible for leading and documenting the meetings during the project. Considering the groups familiarity with each other, the summons will be quite informal, but still enforced.

- **Documentation responsible**
  All group members are responsible for documenting their work, both their work hours and what work they have done on the current task through GitLab. However, Karlsmoen has the responsibility of coordinating all the teams documentation into a cohesive, easily comprehensible document.

- **Head of Design**
  The UI/UX design is an important part of the application. Aas has the responsibility of making sure the UI/UX is up to standard, and meets all requirements. While other members will work on the UI/UX as well, everything must be approved by the head of design if it is to be included in the final design.

## 4 Procedures for the teamwork

- **Meetings**
  Meetings are to occur at least once a week. The organization of meetings is handled by Gluppe, therefore he summons the group to meetings. The idea is one formal meeting a week, with additional working sessions. This number can be increased if deemed necessary.

- **Notification in case of absence or other incidents**
  If a member is late for a meeting or cannot attend, the member must contact the one responsible for organising the meeting. Either contact them on Discord or by phone. The person that is late for the meeting is still allowed to attend, unless it is a repeated problem, which will require more attention to solve.

- **Documents**
  The main method for sharing documents is through Google Drive and GitLab. Google Drive will mainly be used for documents such as this. The main report for the project will be written in Latex using Overleaf. GitLab will be the main resource for sharing code and documentation. Both manual documentation of the code, and auto-generated JavaDoc is going to be handled on GitLab.

- **Policy for monitoring tasks**
  All tasks should be listed as issues on the GitLab project. The issues will be placed on a Kanban board, and updated according to the progress being made on them. This way all members will have a good overview of how the project is coming along.

- **Submission of teamwork**
  To ensure the quality of all members' work there will be a great use of GitLab commit logs to inspect the quality of members' work. This way there will be more than one pair of eyes on anything added to the project. All group members must coordinate with Karlsmoen to ensure proper documentation.

## 5 Interaction

- **Attendance and preparation**
  For meetings, lectures and work sessions there will be a specified time to meet. To make room for errors out of members control, the session will start 15 minutes after the agreed upon meeting time. During the 15 minutes between the set meetup time and actual start, it is encouraged for members to prepare themselves. Any follow-up meetings will be decided upon during the meeting, and follow the same structure as a normal meeting.

- **Presence and commitment**
  During collective work sessions, it is expected that everyone focuses on actual work, and takes dedicated breaks when needed. If a person decides to take an individual break, the person should not disturb the other still working members, and preferably should leave the working area.

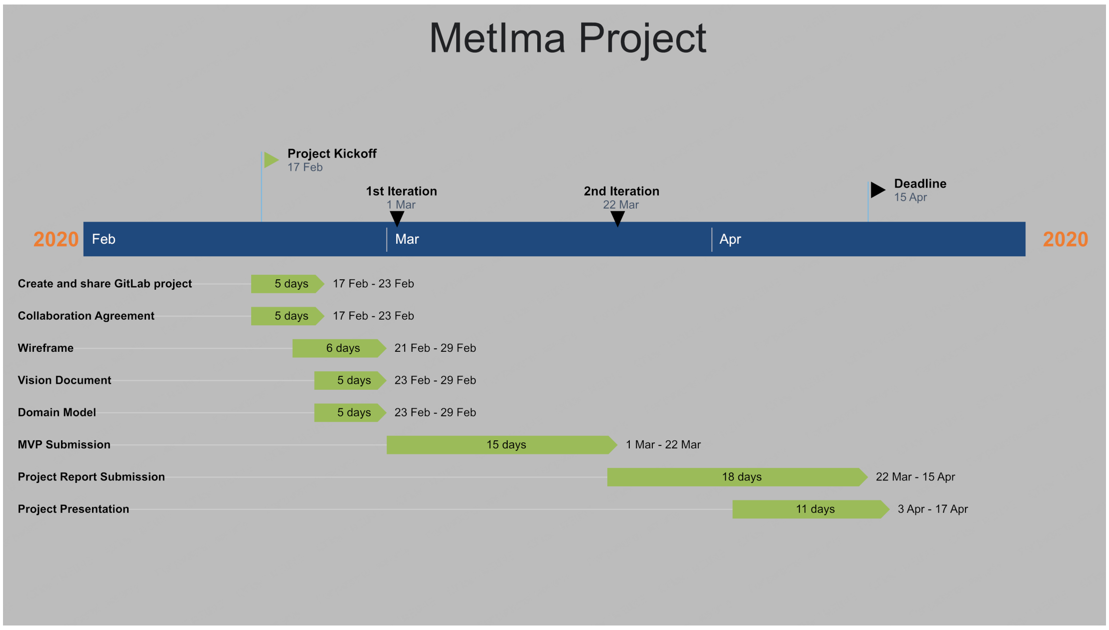
- **How to support each other**
  During meetings, there will be a summary of the work that has occured since the last meeting. Every member then has the opportunity to show off what they have done and receive feedback. This allows for everyone to have a good overview of how the project is coming along, and to help each other if someone is having trouble with their tasks.

- **Disagreement, breach of contract**
  Considering the previous work experience and familiarity of the group, there is not expected any major conflicts. However, if one arises, the involved group members will try to solve it themselves. If this turns out to be hard, the problem will be tackled by the group leader. If this does not help either, there will be brought in a mediator from outside the group, possibly from the school. If the breach of contract is critical enough, the team member will be removed from the project.

## 6 Signatures

| | |
|---|---|
| Project Leader | 25- 2, 2020<br>Date |
| Magnus Gluppe<br>Organization Responsible | 25. 2. 2020<br>Date |
| Jakob Karlmoen<br>Documentation Responsible | 25/2 - 2020<br>Date |
| MA<br>Head of Design | 25/2 -2020<br>Date |

# Gantt Diagram



# Meeting Summons

Mikael Falkenberg Krog    Magnus Gluppe    Jakob Frantzvåg Karlsmoen
Mikkel Aas

February - April 2020

## Contents

# 1 Meeting summons

## 1.1 Meeting summon: Project Group 2 28.02.2020, 11:00, Atriet A-bygget NTNU Gjøvik

Magnus Gluppe is the meeting Chairman and minutes responsible for this meeting.

Agenda

- Case no. 01/2020: Opening meeting
- Case no. 02/2020: Approval of agenda
- Case no. 03/2020: Eating protocol during meetings
- Case no. 04/2020: Startpoint and endpoint for a work sessions
- Case no. 05/2020: Group versus individual work
- Case no. 06/2020: Briefing everyone on individual work

Please contact me if you are unable to attend the meeting.

## 1.2 Meeting summon: Project Group 2 19.03.2020, 19:30, Online Call

Mikael is the meeting Chairman and minutes responsible for this meeting

Agenda

- Case no. 01/2020: Opening meeting
- Case no. 02/2020: Approval of agenda
- Case no. 03/2020: Compression?
- Case no. 04/2020: Workings of export
- Case no. 05/2020: Adding folders instead of individual images
- Case no. 06/2020: Consistency in UI
- Case no. 07/2020: Communication

Please contact me if you are unable to attend the meeting.

## 1.3 Meeting summon: Project Group 2 03.04.2020, 12:00, Online Call

Mikkel is the meeting Chairman and minutes responsible for this meeting

Agenda

- Case no. 01/2020: Opening meeting
- Case no. 02/2020: Approval of agenda
- Case no. 03/2020: Delegate report work
- Case no. 04/2020: Optimization
- Case no. 05/2020: Write JUnit tests
- Case no. 06/2020: The edit image buttons
- Case no. 07/2020: Connect to database

Please contact me if you are unable to attend the meeting.

### 1.4 Meeting summon: Project Group 2 22.04.2020, 12:30, Online Call

Jakob is the meeting Chairman and minutes responsible for this meeting

Agenda

- Case no. 01/2020: Opening meeting
- Case no. 02/2020: Approval of agenda
- Case no. 03/2020: What is left to do on the application
- Case no. 04/2020: What is left to do on the report
- Case no. 05/2020: How has the project been?

Please contact me if you are unable to attend the meeting.

# 2 Minute

## 2.1 Minute from project meeting in Project Group 2

**Time/location:** 28.02.2020, 11:00, Atriet A-bygget NTNU Gjøvik
**Present:** Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog
**Absent:** No one
**Moderator:** Magnus Gluppe

**Case no 2/2020**
Approved by acclamation.

**Case no 3/2020**
Anyone can eat during the meeting or work sessions, but we should avoid noisy food like chips. At important deadlines, the group can go out to eat.

**Case no 4/2020**
Issue undecided, we like the approach of working until we are done with what we started. However, this method can have its disadvantages. Not everyone is productive at the same time, and might wish to end a session early and pick up the work later. If no one else does this, you can feel obligated to stay and produce a sub-par product.

**Case no 5/2020**
When we are further along with the project, it will become easier to work individually. The group still wishes to have joint work sessions, even if our tasks do not overlap. This ensures a certain amount of structure for everyone, and we can ask each other for help.

**Case no 6/2020**
In this part of the project, all the group members have cooperated on various documents and diagrams. So this point is not very relevant to this stage of the project.

28.02.2020, Magnus Gluppe

### 2.2 Minute from project meeting in Project Group 2

**Time/location:** 19.03.2020, 19:30, Online Call
**Present:** Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog
**Absent:** No one
**Moderator:** Mikael Krog

**Case no 2/2020**
Approved by acclamation.

**Case no 3/2020**
We have decided that compression is not to be prioritized, and instead try to find other solutions. We may come back to compression later.

**Case no 4/2020**
The export function is currently under development.

**Case no 5/2020**
We have some ideas to import multiple images is imported at once. We will begin testing this later.

**Case no 6/2020**
It will be taken into consideration.

**Case no 7/2020**
We will keep on using discord as communication, it is working well.

19.03.2020, Mikael Krog

### 2.3 Minute from project meeting in Project Group 2

**Time/location:** 03.04.2020, 12:00, Online Call
**Present:** Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog
**Absent:** No one
**Moderator:** Mikkel Aas

**Case no 2/2020**
Approved by acclamation.

**Case no 3/2020**
We have chosen to write different segments that will be delegated evenly to the group members.

**Case no 4/2020**
The program is using too much RAM. There is still a few things we can try to fix this. We can try to Make scenes static. This ensures that one scene has one controller and not several controllers, which might save RAM.

**Case no 5/2020**
We have to write JUnit tests to ensure that every section of the application meets its design and behaves as intended.

**Case no 6/2020**
We should combine the two edit buttons into one. This makes the GUI more user friendly and removes clutter.

**Case no 7/2020**
We have to move our database from an embedded database within the application to a remote database on the NTNU servers. This will be done with simple configurations in our persistence.xml document.

03.04.2020, Mikkel Aas

### 2.4 Minute from project meeting in Project Group 2

**Time/location:** 22.05.2020, 12:30, Online Call
**Present:** Mikkel Aas, Magnus Gluppe, Jakob Karlsmoen, Mikael Krog
**Absent:** No one
**Moderator:** Jakob

**Case no 2/2020**
Approved by acclamation.

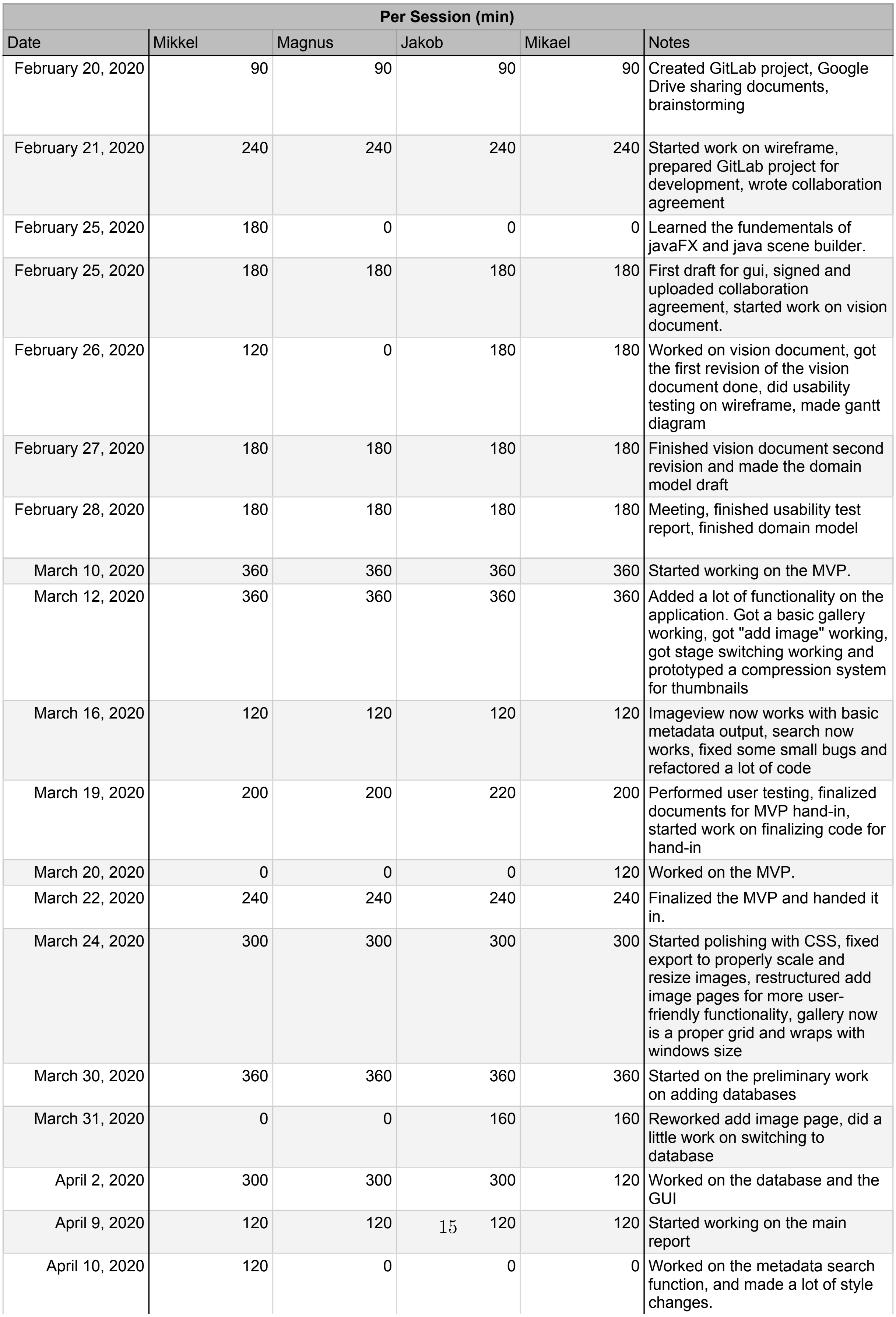
**Case no 3/2020**
Need small cosmetic fixes. There are also other small changes that could be made, but are not important so they are not a priority. Compile to jar? Make sure JavaDoc is good and consistent.

**Case no 4/2020**
Need to flesh out "how the assignment was solved" and "overview". "Further work" needs a lot of work. Some parts need a little padding. Conclusion needs to be written.

**Case no 5/2020**
Probably should have started writing the report earlier. There was a lack of interest in the subject matter, therefore we did not give it the attention it deserves and could have been better at following it more closely during the project.

22.04.2020, Jakob Karlsmoen

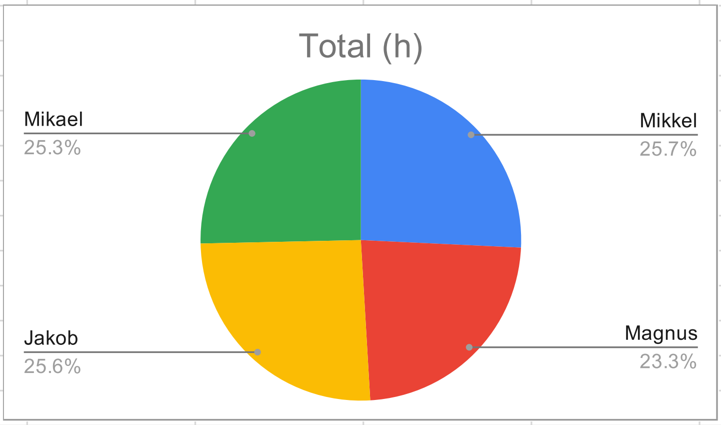
## Timetable



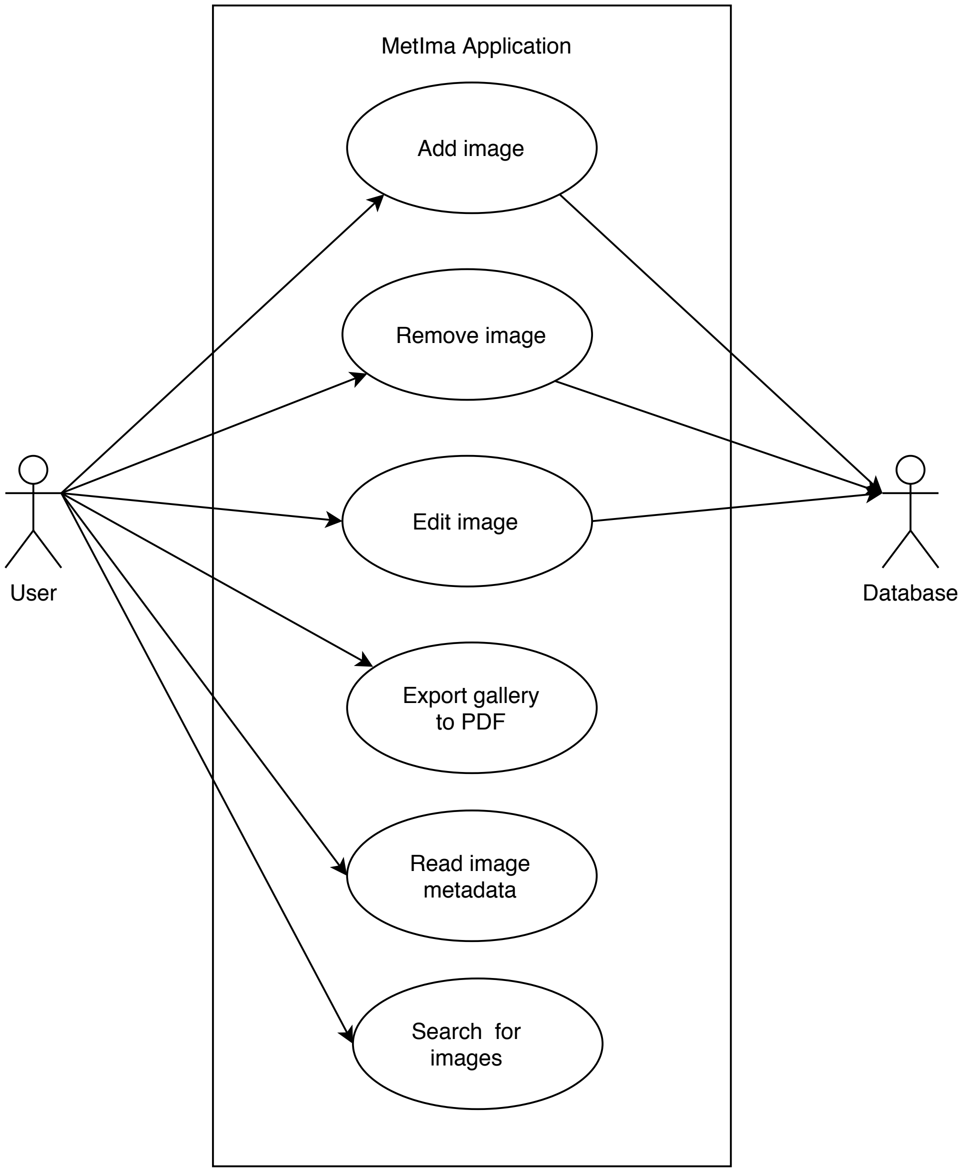
| Per Session (min) | | | | | |
|---|---|---|---|---|---|
| Date | Mikkel | Magnus | Jakob | Mikael | Notes |
| February 20, 2020 | 90 | 90 | 90 | 90 | Created GitLab project, Google Drive sharing documents, brainstorming |
| February 21, 2020 | 240 | 240 | 240 | 240 | Started work on wireframe, prepared GitLab project for development, wrote collaboration agreement |
| February 25, 2020 | 180 | 0 | 0 | 0 | Learned the fundementals of javaFX and java scene builder. |
| February 25, 2020 | 180 | 180 | 180 | 180 | First draft for gui, signed and uploaded collaboration agreement, started work on vision document. |
| February 26, 2020 | 120 | 0 | 180 | 180 | Worked on vision document, got the first revision of the vision document done, did usability testing on wireframe, made gantt diagram |
| February 27, 2020 | 180 | 180 | 180 | 180 | Finished vision document second revision and made the domain model draft |
| February 28, 2020 | 180 | 180 | 180 | 180 | Meeting, finished usability test report, finished domain model |
| March 10, 2020 | 360 | 360 | 360 | 360 | Started working on the MVP. |
| March 12, 2020 | 360 | 360 | 360 | 360 | Added a lot of functionality on the application. Got a basic gallery working, got "add image" working, got stage switching working and prototyped a compression system for thumbnails |
| March 16, 2020 | 120 | 120 | 120 | 120 | Imageview now works with basic metadata output, search now works, fixed some small bugs and refactored a lot of code |
| March 19, 2020 | 200 | 200 | 220 | 200 | Performed user testing, finalized documents for MVP hand-in, started work on finalizing code for hand-in |
| March 20, 2020 | 0 | 0 | 0 | 120 | Worked on the MVP. |
| March 22, 2020 | 240 | 240 | 240 | 240 | Finalized the MVP and handed it in. |
| March 24, 2020 | 300 | 300 | 300 | 300 | Started polishing with CSS, fixed export to properly scale and resize images, restructured add image pages for more user-friendly functionality, gallery now is a proper grid and wraps with windows size |
| March 30, 2020 | 360 | 360 | 360 | 360 | Started on the preliminary work on adding databases |
| March 31, 2020 | 0 | 0 | 160 | 160 | Reworked add image page, did a little work on switching to database |
| April 2, 2020 | 300 | 300 | 300 | 120 | Worked on the database and the GUI |
| April 9, 2020 | 120 | 120 | 120 | 120 | Started working on the main report |
| April 10, 2020 | 120 | 0 | 0 | 0 | Worked on the metadata search function, and made a lot of style changes. |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| April 16, 2020 | 240 | 240 | 330 | 240 | Worked on main report |
| April 17, 2020 | 240 | 240 | 240 | 300 | Continued work on main report, did bugfixes on applicaiton |
| April 21, 2020 | 240 | 240 | 300 | 300 | More work on main report, design fixes and bug fixes. |
| April 23, 2020 | 180 | 120 | 180 | 120 | Design changes and bug fixes on application. Also continued working on the main report |
| April 26, 2020 | 120 | 120 | 0 | 120 | Worked on the main report. |
| April 27, 2020 | 240 | 240 | 240 | 240 | More work on main report. Packaged application to jar. Worked on GitLab WIKI |
| April 28, 2020 | 180 | 180 | 180 | 180 | Finished up the project and prepared for hand-in |

| **Total (h)** |  |  |  |
|---|---|---|---|
| Mikkel | Magnus | Jakob | Mikael |
| 84.8 | 76.8 | 84.3 | 83.5 |

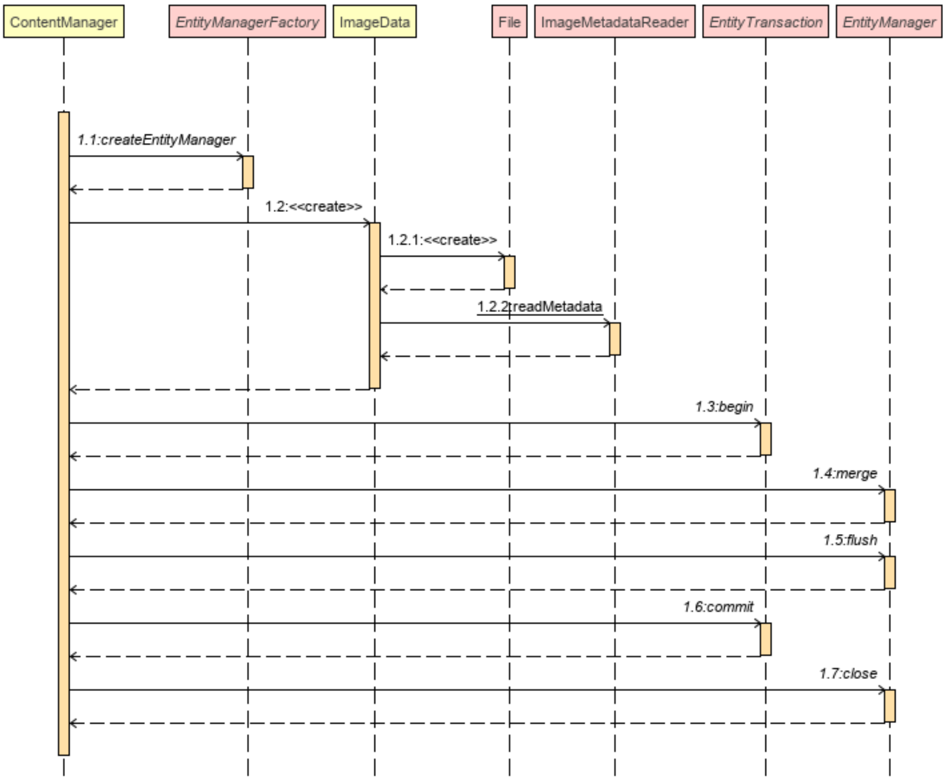


## C Use-Case Diagram

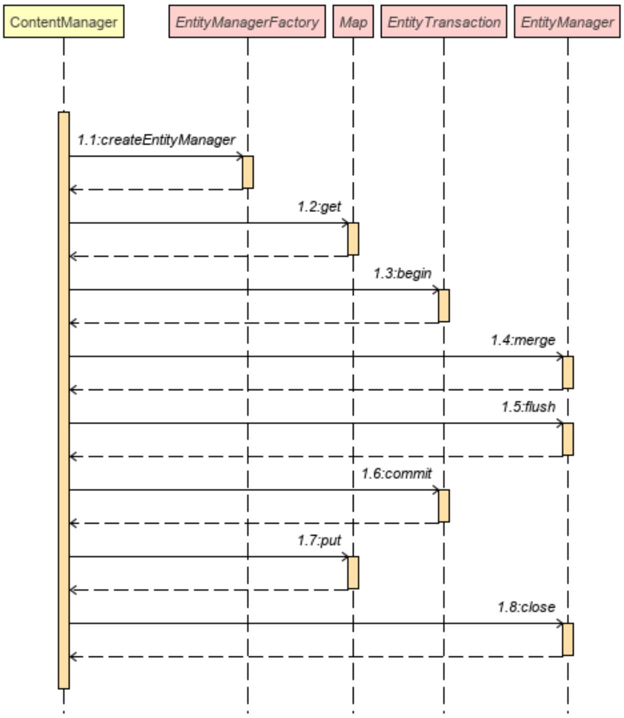


## D Sequence Diagrams

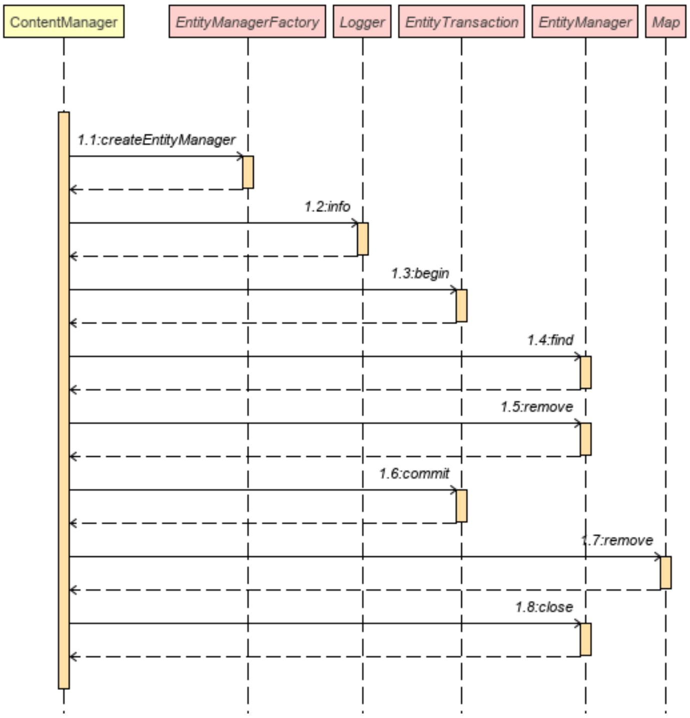
Add image sequence diagram:



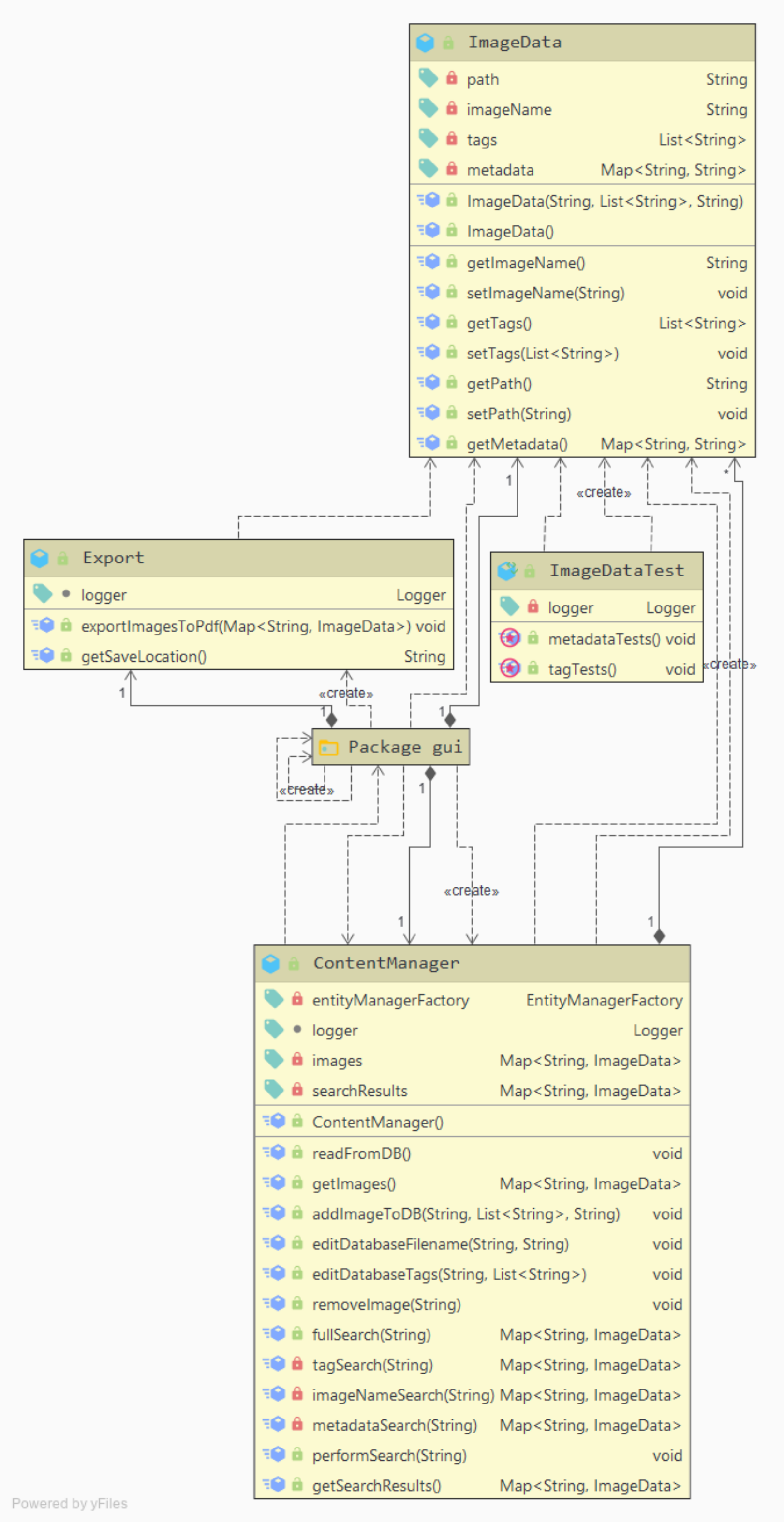
Edit image sequence diagram:



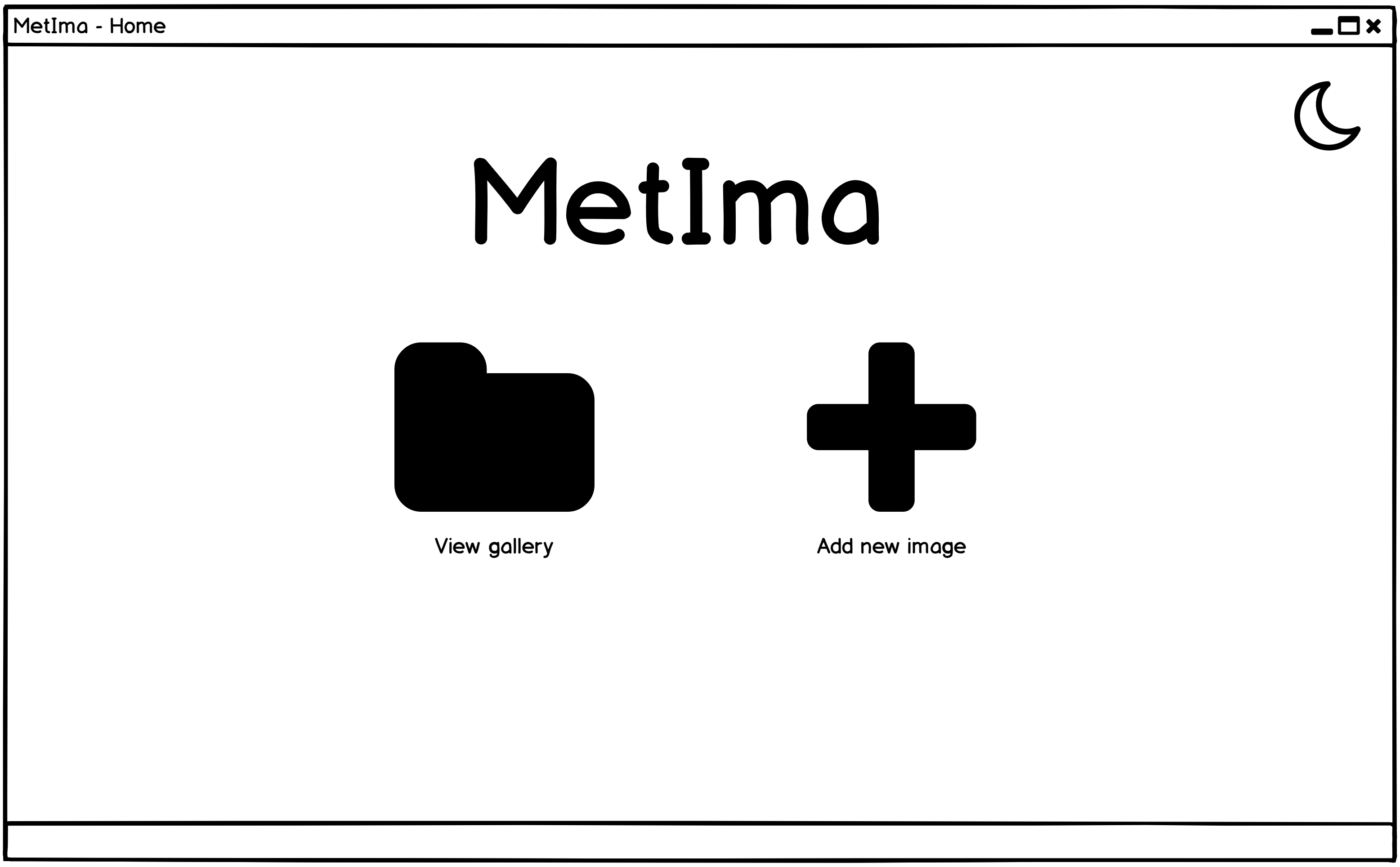
Remove image sequence diagram:



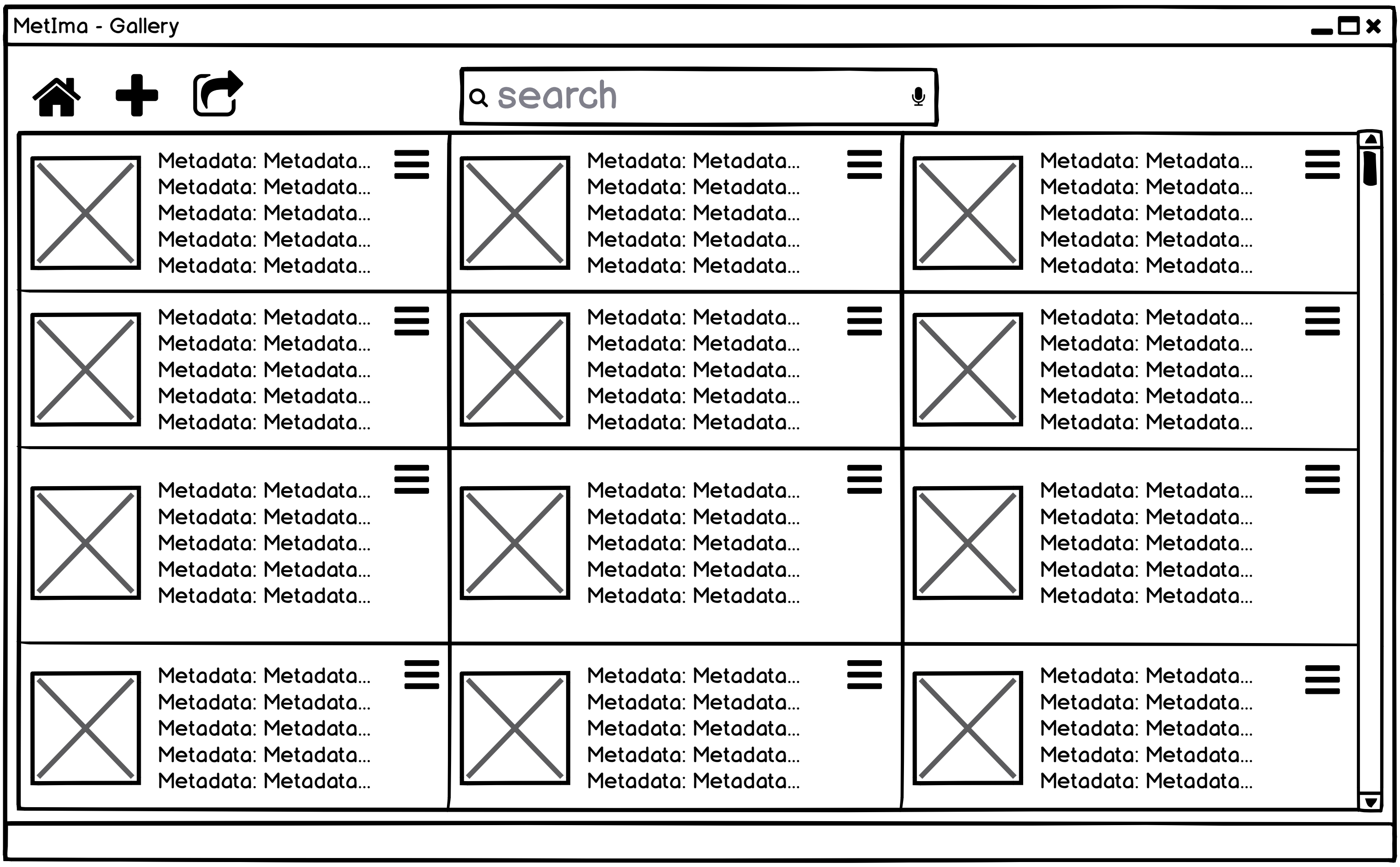
# E Class Diagram

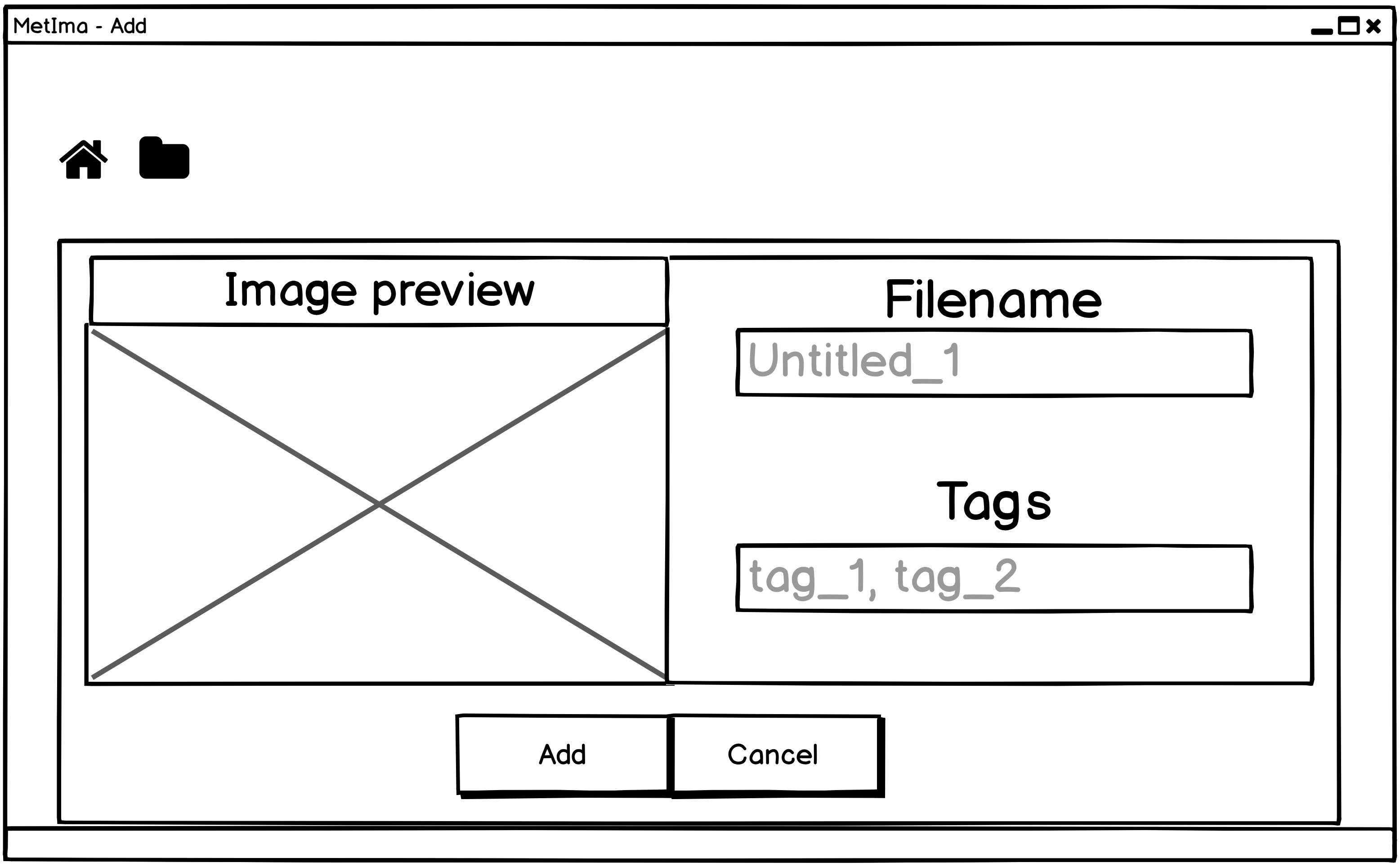

ImageData
path String
imageName String
tags List<String>
metadata Map<String, String>
ImageData(String, List<String>, String)
ImageData()
getImageName() String
setImageName(String) void
getTags() List<String>
setTags(List<String>) void
getPath() String
setPath(String) void
getMetadata() Map<String, String>
1
«create»
*
Export
logger Logger
exportImagesToPdf(Map<String, ImageData>) void
getSaveLocation() String
ImageDataTest
logger Logger
metadataTests() void
tagTests() void
«create»
1
«create»
1
1
Package gui
«create»
1
«create»
1
1
ContentManager
entityManagerFactory EntityManagerFactory
logger Logger
images Map<String, ImageData>
searchResults Map<String, ImageData>
ContentManager()
readFromDB() void
getImages() Map<String, ImageData>
addImageToDB(String, List<String>, String) void
editDatabaseFilename(String, String) void
editDatabaseTags(String, List<String>) void
removeImage(String) void
fullSearch(String) Map<String, ImageData>
tagSearch(String) Map<String, ImageData>
imageNameSearch(String) Map<String, ImageData>
metadataSearch(String) Map<String, ImageData>
performSearch(String) void
getSearchResults() Map<String, ImageData>
Powered by yFiles

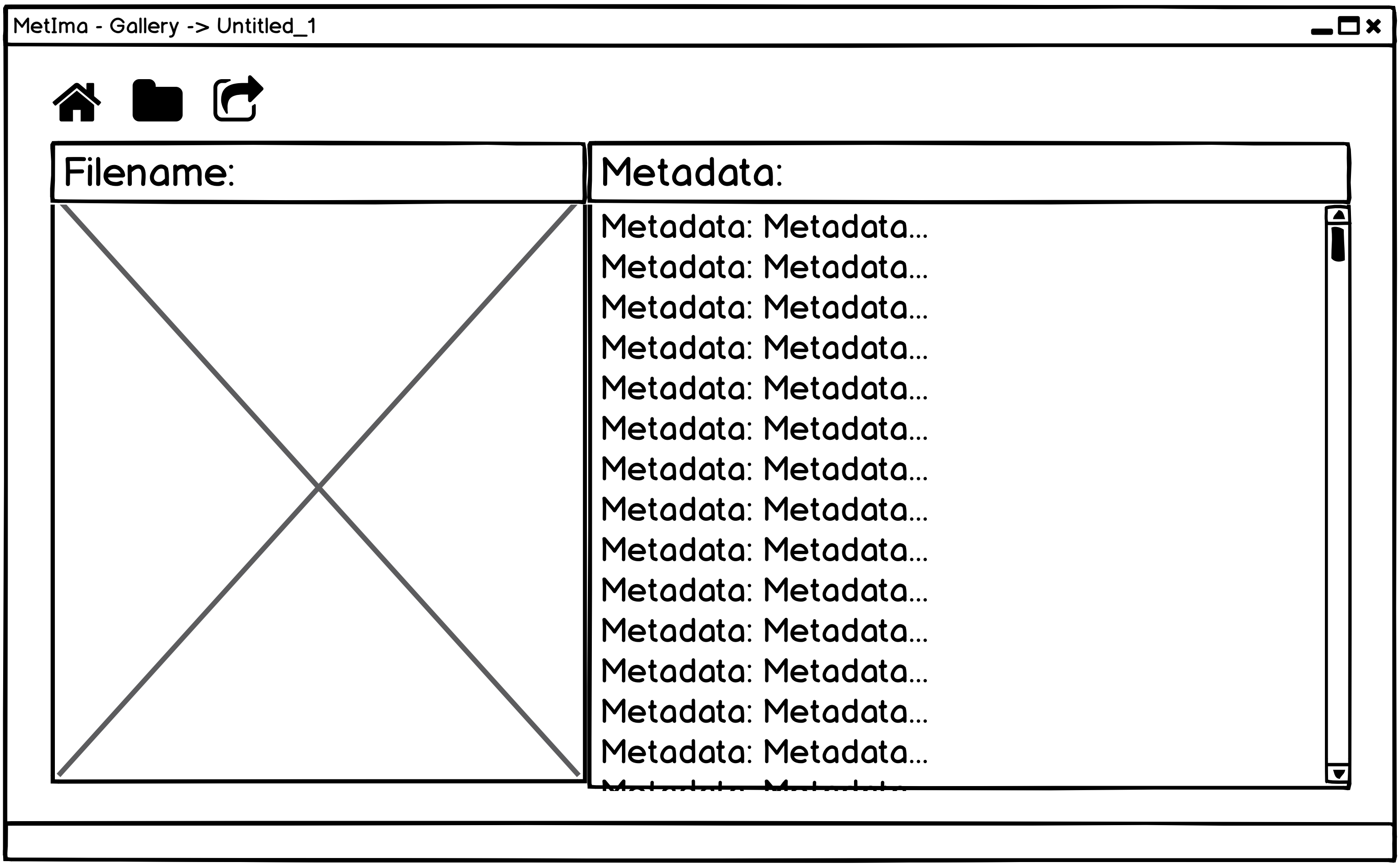
## F Wireframe



MetIma - Home


MetIma - Gallery
search
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...


MetIma - Add
Image preview
Filename
Untitled_1
Tags
tag_1, tag_2
Add
Cancel



Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...
Metadata: Metadata...