To-Do-List App

Vision

Version 0.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23.02.2021 | 0.1 | First draft | Everyone |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**Introduction**](#_30j0zll) **5**

[Purpose of the vision document](#_1fob9te) 5

[Scope of the vision document](#_3znysh7) 5

[Framework conditions](#_cijc4tk9r4vk) 5

[Definitions, Acronyms, and Abbreviations](#_2et92p0) 5

[Overview](#_3dy6vkm) 5

[**Positioning**](#_vn2klzsop1lr) **6**

[Problem statement](#_3f5iq2tn8t2z) 6

[Product Position Statement](#_17dp8vu) 6

[**Project goals**](#_3rdcrjn) **6**

[Efficiency goals](#_26in1rg) 6

[Result goals](#_lnxbz9) 6

[Process goals](#_35nkun2) 6

[**User Descriptions**](#_dnentzqq0dnq) **7**

[User Summary](#_z337ya) 7

[User Environment](#_3j2qqm3) 7

[User Profiles](#_2xcytpi) 7

[Student](#_1ci93xb) 7

[Student Assistant](#_29kc7uj4jhlf) 7

[Professor](#_ti4qb0k8t3co) 8

[Key User Needs](#_3whwml4) 8

[**Product overview**](#_421esrldtif0) **9**

[Features](#_hvj6usiq2fkm) 9

[Create a reminder](#_pnrhwt5makqq) 9

[Delete/edit reminder](#_gy9n7vgl19yx) 9

[Push notifications](#_69ymdprv3v7t) 9

[Mark an assignment as done](#_ctesz8wvmeii) 9

[Save reminders](#_m5zkurfftdfy) 9

[Product Perspective](#_49x2ik5) 10

[Summary of Capabilities](#_2p2csry) 10

[**Risk analysis**](#_64mqsichacu) **11**

[Risk Table](#_cctw4q19ufdr) 11

[Risk chart](#_ngt2wqt0bugy) 12

[**Installation**](#_8ch7jq1vzz9k) **13**

[**Constraints**](#_4f1mdlm) **13**

[**Quality Ranges**](#_2u6wntf) **13**

[**Precedence and Priority**](#_19c6y18) **13**

[**Other Product Requirements**](#_3tbugp1) **13**

[Applicable Standards](#_28h4qwu) 13

[System Requirements](#_nmf14n) 13

[Performance Requirements](#_1mrcu09) 13

[**Documentation Requirements**](#_ryxqpuirsbm3) **14**

[User Manual](#_111kx3o) 14

[Main wiki page](#_3l18frh) 14

[Installation Guides, Configuration, and Read Me File](#_206ipza) 14

[Labeling and Packaging](#_4k668n3) 14

[**A Feature Attributes**](#_2zbgiuw) **15**

[A.1 Status](#_1egqt2p) 15

[A.2 Effort](#_3ygebqi) 15

[A.3 Stability](#_2dlolyb) 15

[A.4 Target Release](#_sqyw64) 15

[A.5 Assigned To](#_3cqmetx) 15

[A.6 Reason](#_1rvwp1q) 15

Our Vision

# Introduction

This document's purpose is to collect, analyze and define the needs and features of the To-Do-List application. The document will focus on the capabilities needed by the target users. The details of how the To-Do-List application fulfills these needs are detailed in the use-case and supplementary specifications.

## Purpose of the vision document

This is the first and founding document regarding the project from the project group to our client. It will create a basis for the parties to decide whether they want to pursue. In the case all agree to pursue, this document will make sure they have a common understanding of the superior goals and scope of the project. The focus should be on our clients needs and the underlying reasons that these needs exist. Furthermore this document will contain information on how we intend to meet these needs. Unified process is an iterative process, which could mean that the document can be changed as the project encounters new hurdles.

## Scope of the vision document

This document is associated with our semester assignment in the subject IDATT1002. It covers the needs and goals of the project application.

* 1. **Reasoning for the project**

This project is an assignment given in the subject IDATT1002. Its purpose is to teach students to work on bigger projects and work together as a team.

## Framework conditions

The project is limited by some conditions. The most important being the size of the developer team. Our team consists of 6 developers, each having only 150 hours of disposable time. This limits the project's capability of addressing details.

Because of the nature of the project, being an assignment, there are some technical limitations as well. We are required to use java, javafx and Balsamiq for this project.

## Definitions, Acronyms, and Abbreviations

System or application may be used intertwining and refer to the finished product. Application may be shortened to app.

## Overview

In this project we are producing a to do list application, which will have the goal of keeping track of things a user needs to remember. The application will be for all people who are in the possession of a computer being able to run java. In terms of goals there are three categories. The efficiency goals involve becoming more efficient in larger projects. Our result goals involve being able to present and deliver a to do list application which answers all requirements within the deadline. Additionally we want to increase our competence within the subject, this falls under process goals.

Our three main users: student, student assistant and professor define the basic needs of the application. This leads us into the key application features. The user should at least be able to add events, dates and descriptions, remove events, mark as done and edit existing events. The risk analysis shows us that there are no significant risks associated with this project. Furthermore the document will describe our constraints and document requirements, as well as other project requirements.

## 

# Positioning

## Problem statement

|  |  |
| --- | --- |
| The problem of | having too much to remember at a given time. |
| the impact of which is | Forgetting important meetings and deadlines. |
| a successful solution would lead to | Keeping track of things you need to do. Not forgetting things you need to remember. |

## Product Position Statement

|  |  |
| --- | --- |
| For | All people who own a computer with java installed. |
| Who | People who often forget things or like to keep their events organized. |
| The (product name) | Application. |
| That | Keeps your activities organized and reminds you of them. |
| Unlike | Your own memory. |

# Project goals

## Efficiency goals

* Increase efficiency by scheduling working hours and organizing what to work on at a given time.
* Reduce the chance of not being able to deliver within a deadline by doing the work early and setting our own deadlines for different parts of the project.
* Increase efficiency by working on the code together, we will have multiple people working on the code at the same time.

## Result goals

* Present an to-do-list application with all the best features to help the user organize their day. The application should keep all relevant information like task description, priority, category, status, deadline, and start and finish date. The user should also be able to mark tasks as done, change the priority of the task and reorder tasks.
* Present a program that fits the description given by the requestor.
* Completed within the deadline 30/04/2021
* The application meets Datatilsynet’s ethical guidelines and mandatory requirements
* Pass the subject with a preferable grade.

## Process goals

* Improve skills in collaboration and communication.
* Improve competence within the subject.
* Get to know the other students and make bonds.

# 

# 

# User Descriptions

## User Summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| Student | Developers of the application | Develop, download and use the application |
| Student assistant | A student who works for the university as a help for other students. Helps the professor and can approve assignments | Guide the students and help them during the project. |
| Professor | User and grader of the application | Use and test application  Grade the project |

## User Profiles

### Student

|  |  |
| --- | --- |
| **Representative** | Self presented |
| **Description** | A student, studying at the university. |
| **Type** | Developer |
| **Responsibilities** | Developing and delivering the application as well as managing the project. |
| **Success Criteria** | Grade is preferable |
| **Involvement** | Develop and produce the application |
| **Deliverables** | The application |
| **Comments / Issues** |  |

### Student Assistant

|  |  |
| --- | --- |
| **Representative** | Presented by the professor |
| **Description** | A student who works for the university as a help for other students. Helps the professor and can approve assignments. |
| **Type** | Guider |
| **Responsibilities** | Project overview and guidance |
| **Success Criteria** | The guidance helps and improves the project. |
| **Involvement** | Guide the students and help them during the project. |
| **Deliverables** | Experience and knowledge |
| **Comments / Issues** |  |

### Professor

|  |  |
| --- | --- |
| **Representative** | Self presented |
| **Description** | Professor at the university and has experience within the subject. |
| **Type** | Expert |
| **Responsibilities** | Grading the project, and teaching the necessary background information. |
| **Success Criteria** | The submitted project answers to all criteria set |
| **Involvement** | Grading, setting deadlines and project requirements. |
| **Deliverables** | None |
| **Comments / Issues** |  |

## Key User Needs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Need** | **Priority** | **Concerns** | **Current Solutions** | **Proposed Solutions** | |
| Add new tasks | High |  | none | | The user can add events at the main screen with the use of a single button. |
| Rearrange tasks / Edit tasks | High |  | none | | Rearrange the tasks order |
| Add date | High |  | none | | When adding an event the user can select a due date for the event. |
| Add description | High |  | none | | When adding an event the user can add a description for the event. |
| Change priority | High |  | none | | Change the priority of the tasks |
| Remove events | High |  | none | | The user can select an event and then choose to remove it. With a confirmation screen, in case of a miss click. |
| Mark events as done | High |  | none | | The user can click on a button on the event to mark it as done. |
| Reminder | Low |  | none | | User get reminded of events in advance |

# Product overview

• Product perspective

The final product is a To Do application. The applications main features are the possibility of adding, keeping track of, and removing tasks, with additional functions described under *Product functions.*

• Product functions

As a minimum the following functions are required:

* User should be able to add new tasks
* User should be able to rearrange tasks
* User should be able to change priority of tasks
* User should be able to mark tasks as done
* Application should be able to keep tasks stored
* Application should keep all relevant information stored:
  + Start and finish date
  + Task description
  + Priority
  + Category
  + Status
  + Deadline

The application should have additional functions. These functions should be added according to feedback from user tests and other feedback.

## Features

The basic functionality of the system should be as follows. The system should be able to make a reminder for the user. This reminder should contain the date, what type of reminder, the importance of the reminder, and other relevant information. Furthermore the system must be able to push this as a notification to the user. More detailed features will be listed and detailed below.

### Create a reminder

The system needs to be able to process the input of the user and translate this into an appointment that can be viewed, accessed, edited and/or deleted as the user deems necessary. The system also needs to be able to display the date/time of the reminder, what type of reminder (e.g. chores, submission, homework), the importance of the reminder, a brief description of the reminder.

### Delete/edit reminder

The system needs to be able delete or edit all of the information relating to the reminder.

### Push notifications

The system should be able to push a notifications to the user(e.g through; email, push notification, popup)

### Mark an assignment as done

Whenever the assignment is done, the system needs to be able to mark an assignment as done after a user's input.

### Save reminders

The system needs to be able to save and keep track of the different reminders.

## Product Perspective

This application is a standalone, independent and totally self-contained app. It is important to have a to do app that is quick to use, easy access and with few, but important functions/features. Therefore the application should be easy to use, with minimal functions, keeping it user friendly, quick and competitive. In contrast with the majority of the competitors, this app is only offering the most important functions, eliminating less important “add-on features”.

## Summary of Capabilities

**Table 4-1 To-do-list app**

|  |  |
| --- | --- |
| **Customer Benefit** | **Supporting Features** |
| Helps the customer easily organize the day. | Create reminders and sort them by date and category. |
| Helps the customer remember. | The application will push notifications to remind the user. |
| The customer can keep track of done and yet to do elements. | The app will allow you to mark an element as completed. |

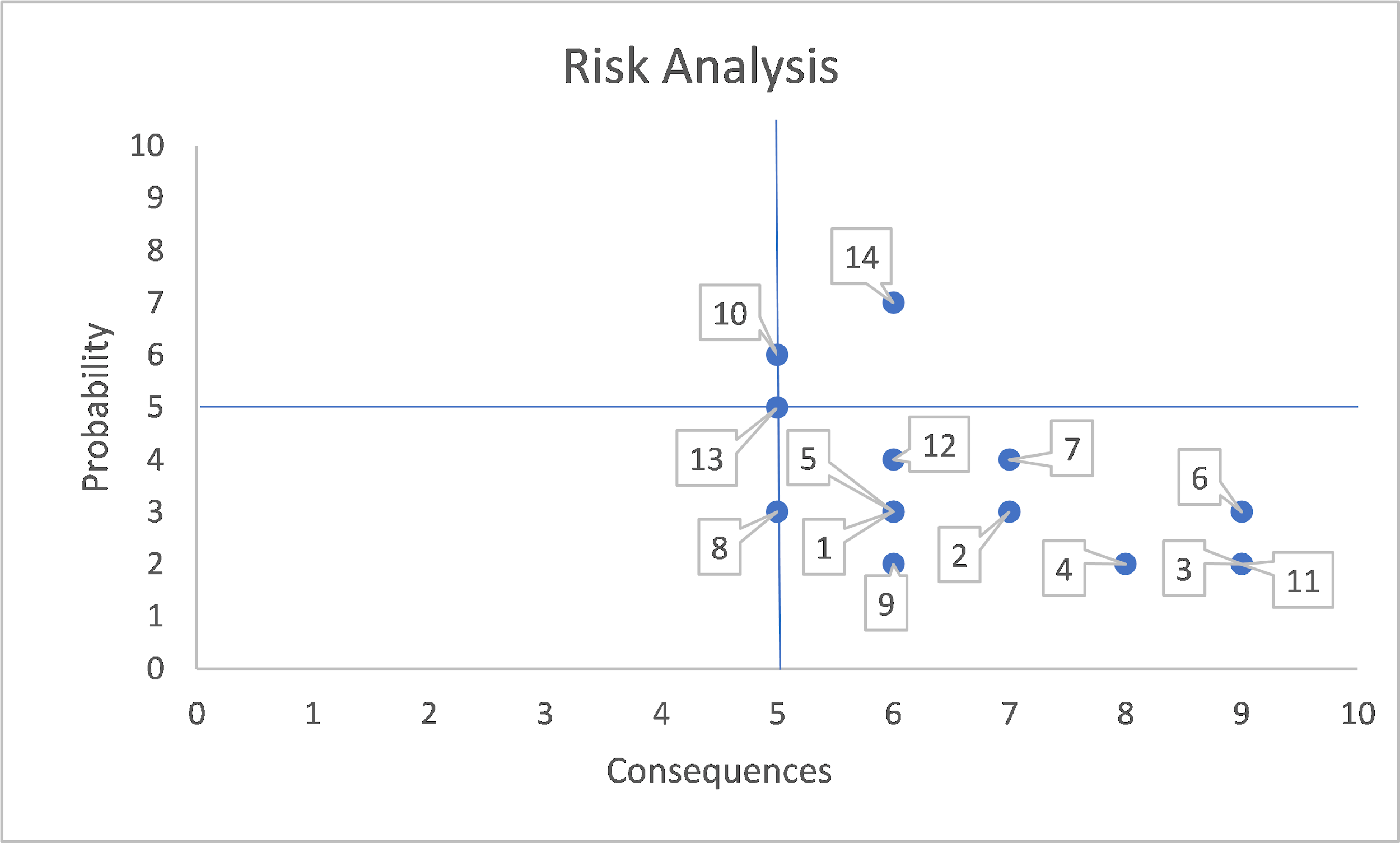
## 

# Risk analysis

## Risk Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Short Description** | **Impact** | **Probability** | **Significance** |
| **Technology** | | | |
| 1.The technology we have chosen is new and untested and may therefore have serious shortcomings | 6 | 3 | 18 |
| 2. The product may contain serious unseen bugs | 7 | 3 | 21 |
| 3.  The product does not fulfill the users criterias | 9 | 2 | 18 |
| 4.  The product violates ethical guidelines | 8 | 2 | 16 |
| 5.  One or more of the members computers stop functioning | 6 | 3 | 18 |
| 6.  Loss of work | 9 | 3 | 27 |
| **Interaction** | | | |
| 7. There may arise misunderstandings between the various parts | 7 | 4 | 28 |
| 8. Conflict can occur within the group during the project. | 5 | 3 | 15 |
| 9. One or more group members do not complete or do their work tasks. | 6 | 2 | 12 |
| 10. One or more group members become ill during the working period. | 5 | 6 | 30 |
| 11.  Unable to deliver our product within deadline. | 9 | 2 | 18 |
| 12.  There may be difficulties in obtaining the expertise or resources in the group | 6 | 4 | 24 |
| 13. Schedule delays | 5 | 5 | 25 |
| 14.  Covid prohibits the use of campus throughout the whole project time period | 6 | 7 | 42 |

## Risk chart



# Cost pricing and benefits

The product's cost will primarily be paying wages. This is because all software used is provided for free either through freeware(e.g. Google docs) or student licenses(e.g. IntelliJ, Balsamiq, Gitlab) from NTNU. The project is estimated to cost around 1.3 million NOK, this is largely wages for 150 work hours for 6 developers. As the finished product is distributed online, there will be no significant costs regarding distribution.

Comparing the cost and benefits of completing the application and all work that is associated with it versus doing nothing, it becomes clear that finishing all the work is beneficial. This is because this will ensure that we have a chance at a good grade. Doing nothing will ensure that we will fail the subject. Furthermore completing all the work will give us a good amount of experience with producing and completing larger projects, as well as we deliver a finished product to our clients.

# Installation

As this is a Java application it will be able to be installed on all compatible devices that support Java. This is because Java is a “compile once, run everywhere” language. The bytecode which the compiler produces on our machines will be able to run on all machines with Java support.

# Constraints

The system will primarily be limited by our(developers) ability to program the system. Other factors will be our choice of environment. We will be using Maven with JavaFx. As we are new to this environment, it will be unrealistic to expect that we will be able to extract all potential from Maven or JavaFx. Therefore our primary constraint will be our own abilities.

# Quality Ranges

Our goal is that the system will be near flawless. It will contain fault handling, and be very user friendly. As this is a relatively light application, performance will probably not be a big consideration and/or problem.

# Precedence and Priority

The priority feature will be to add a new reminder, without this feature the program will be essentially useless. Notifications to the user will be the least prioritized feature.

# Other Product Requirements

## Applicable Standards

As we are aiming to follow all applicable laws and rules this system will aim to be in accordance with GDPR and other EU and Norwegian laws. The system will also follow standardized programming norms, e.g. documentation, code structure, and other Java norms.

## System Requirements

The main requirement of the application will be that the hardware supports Java and JavaFx.

## Performance Requirements

The system will be relatively light to run, therefore performance requirements will not be a big factor. However the development will focus on using established methods, which will ensure good performance.

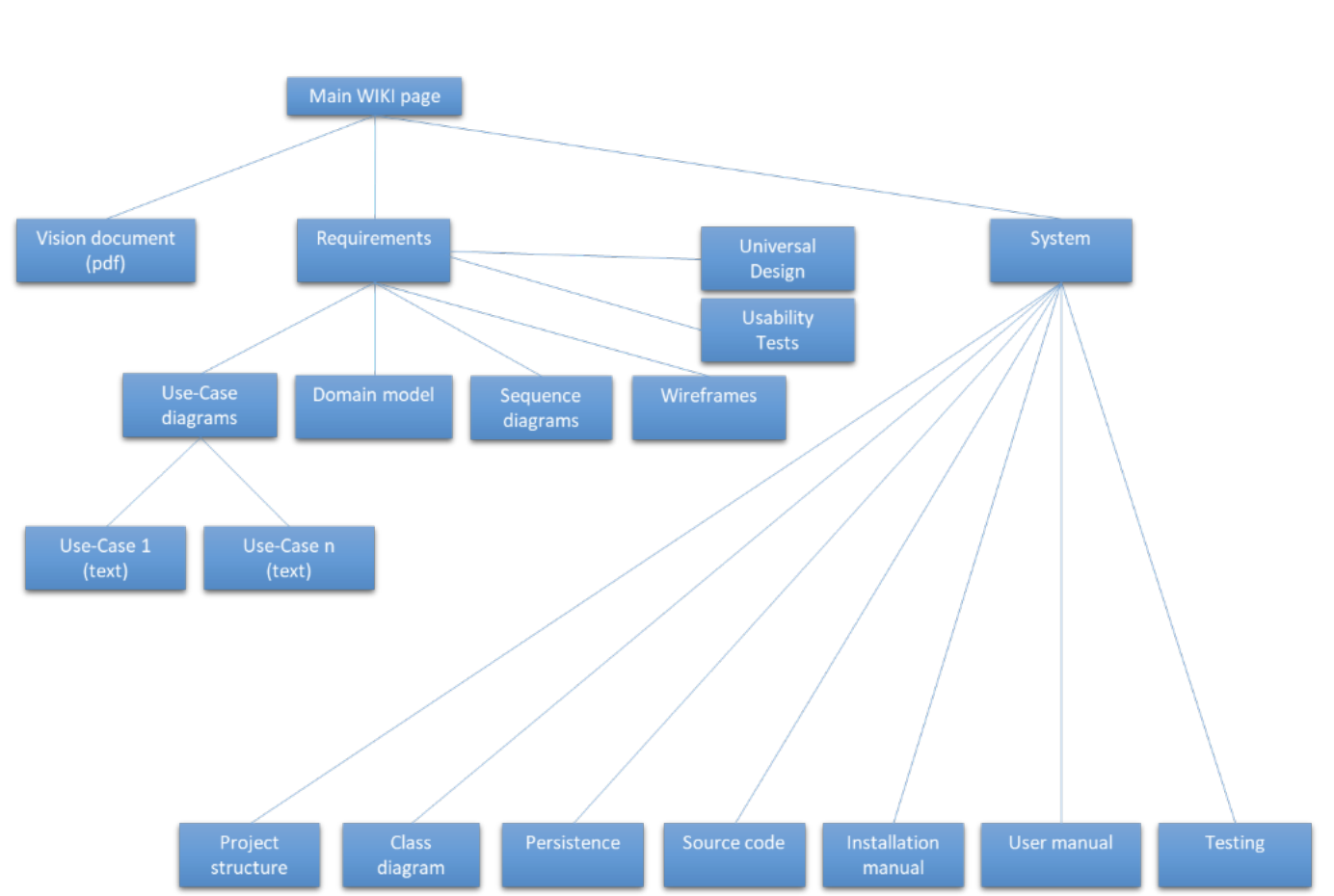
# Documentation Requirements

## User Manual

The user manual will contain relevant information regarding installing, using and uninstalling the application. It should be easy to read and its primary function will be to guide the user through the basic functionality of the application. More advanced functionality will be described in the wiki. The user manual will be publically available through our repository. It will probably not be available in printed format.

## Main wiki page

For more frequently asked questions there will be a FAQ available through our repository. Also available is a wiki which describes more advanced functionality of the program. The wiki will contain detailed descriptions of how the program functions, furthermore it will contain all of the documents listed in the image below(e.g. Domain model, this document, source code, UML-diagrams, wireframes)

**

## Installation Guides, Configuration, and Read Me File

The installation guide will detail how the average user would go about installing the program, and perhaps common issues during installations. A readme file will also be shipped with each release, and this will detail new functionality.

## Labeling and Packaging

The code will contain who authored the specific code. The graphical elements of the system will contain standardized icons and graphics. Furthermore there will be a license agreement available.

# A Feature Attributes

*[Features are given attributes that can be used to evaluate, track, prioritize, and manage the product items proposed for implementation. All requirement types and attributes need to be outlined in the Requirements Management Plan, however, you may wish to list and briefly describe the attributes for features that have been chosen. The following subsections represent a set of suggested feature attributes.]*

## A.1 Status

*[Set after negotiation and reviewed by the project management team. Tracks progress during definition of the project baseline.]*

|  |  |
| --- | --- |
| Proposed | *[Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management, and user or customer community.]*  F |
| Approved | *[Capabilities that are deemed useful and feasible, and have been approved for implementation by the official channel.]* |
| Incorporated | *[Features incorporated into the product baseline at a specific point in time.]* |

## A.2 Effort

*[Set by the development team. Because some features require more time and resources than others, estimating the number of team or person-weeks, lines of code required or function points, for example, is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.]*

## A.3 Stability

*[Set by the analyst and development team, this is based on the probability that features will change or the team’s understanding of the feature will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.]*

## A.4 Target Release

*[Records the intended product version in which the feature will first appear. This field can be used to allocate features from a* ***Vision*** *document into a particular baseline release. When combined with the status field, your team can propose, record, and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the* ***Vision*** *document but will be scheduled for a later release.]*

## A.5 Assigned To

*[In many projects, features will be assigned to "feature teams" responsible for further elicitation, writing the software requirements, and implementation. This simple pull-down list will help everyone on the project team to understand responsibilities better.]*

## A.6 Reason

*[This text field is used to track the source of the requested feature. Requirements exist for specific reasons. This field records an explanation or a reference to an explanation. For example, the reference might be to a page and line number of a product requirement specification or to a minute marker on a video of an important customer review.]*