To-Do-List App

Vision

Version 0.1

*[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]*

*[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]*

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

[[This subsection describes what the rest of the Vision document contains and explains how the document is organized.]](#_bvt8ag193c9h) 5

[**Positioning**](#_1t3h5sf) **5**

[Problem Statement](#_2s8eyo1) 5

[Product Position Statement](#_17dp8vu) 6

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

[Efficiency goals](#_26in1rg) 6

[Result goals](#_lnxbz9) 7

[Process goals](#_35nkun2) 7

[**User Descriptions**](#_1ksv4uv) **8**

[User Summary](#_z337ya) 8

[User Environment](#_3j2qqm3) 8

[User Profiles](#_2xcytpi) 9

[Student](#_1ci93xb) 9

[Student Assistant](#_29kc7uj4jhlf) 9

[Professor](#_ti4qb0k8t3co) 9

[Key User Needs](#_3whwml4) 10

[**Product overview**](#_421esrldtif0) **10**

[Features](#_hvj6usiq2fkm) 11

[Create a reminder](#_ktr7w1nyrvo3) 11

[Delete/edit reminder](#_5wrckf5v47yg) 11

[Push notifications](#_pv0s1hvcgs3x) 11

[Mark an assignment as done](#_f0xzfe2fbc6w) 11

[Save reminders](#_5f2u51rgo4d5) 11

[Product Perspective](#_49x2ik5) 12

[Summary of Capabilities](#_2p2csry) 12

[Assumptions and Dependencies](#_147n2zr) 12

[**Risk analysis**](#_x4txll6w19yy) **12**

[**Cost pricing and benefits**](#_awtqepxr9yx8) **14**

[Quantifiable and non-quantifiable benefits](#_ihv636) 15

[**Licensing and installation**](#_rxk66vtoez5o) **16**

[**Product Features**](#_vx1227) **16**

[Create a reminder](#_3fwokq0) 16

[Delete/edit reminder](#_1v1yuxt) 17

[Push notifications](#_fdyt84yt3ae8) 17

[Mark an assignment as done](#_cgf8hthqoovk) 17

[Save reminders](#_btmeeikkeud5) 17

[**Constraints**](#_4f1mdlm) **17**

[**Quality Ranges**](#_2u6wntf) **17**

[**Precedence and Priority**](#_19c6y18) **17**

[**Other Product Requirements**](#_3tbugp1) **17**

[Applicable Standards](#_28h4qwu) 17

[System Requirements](#_nmf14n) 17

[Performance Requirements](#_1mrcu09) 18

[Environmental Requirements](#_46r0co2) 18

[**Documentation Requirements**](#_2lwamvv) **18**

[User Manual](#_111kx3o) 18

[Online Help](#_3l18frh) 18

[Installation Guides, Configuration, and Read Me File](#_206ipza) 18

[Labeling and Packaging](#_4k668n3) 18

[**A Feature Attributes**](#_2zbgiuw) **18**

[A.1 Status](#_1egqt2p) 19

[A.2 Effort](#_3ygebqi) 19

[A.3 Stability](#_2dlolyb) 19

[A.4 Target Release](#_sqyw64) 19

[A.5 Assigned To](#_3cqmetx) 19

[A.6 Reason](#_1rvwp1q) 19

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 stakeholders” and 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.

*[A brief description of the scope of this* ***Vision*** *document; what Project(s)it is associated with and anything else that is affected or influenced by this document.]*

* 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.

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the* ***Vision*** *document. This information may be provided by reference to the project’s Glossary.]*

## Overview

## *[This subsection describes what the rest of the* ***Vision*** *document contains and explains how the document is organized.]*

# Positioning

## Problem Statement

|  |  |
| --- | --- |
| The problem of | Make a to-do-list application. |
| the impact of which is | Forgetting important things like meetings and deadlines. |
| a successful solution would be | Keeping track of things you need to do. Not forgetting things you need to remember. |

## Product Position Statement

*[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]*

|  |  |
| --- | --- |
| For | All people who own a smartphone or has access to the internett |
| 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. |

*[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]*

# Project goals

* The application meets the users requirements
* Deliver the application within the deadline
* The application meets Datatilsynet’s ethical guidelines and mandatory requirements

The project's objectives are the basis for:

* To be agreed with the client about what will be the outcome of the project
* Being able to plan and manage
* To assess retrospectively the outcome of the project that was as agreed with the client
* To have a common understanding of the project group for what the job entails

The project objectives, we often formulate based on the problems we have identified and described. However, there may be cases where the project is started in conjunction with a new business model. Then the basis for the project objectives may be the same targets for new business model.

When we formulate objectives, we emphasize that the objectives are:

1. Operational, they must be:
   1. Measurable, it must subsequently be possible to detect if the target was reached
   2. Steerable, they must be indicative when it comes action options
2. Clearly formulated, it must be reasonably certain that anyone reading the objectives reads the same. (We can say that the opposite is " poetic " goal, the purpose of poetry is to create associations so that the individual can bring forward their own images and moods )

It is useful to distinguish between three types of goal: efficiency, result and process goals.

## Efficiency goals

**Effect - the impact of the project**. Effect describes what the client wants to achieve with the project. Effect can be linked to the organization's strategic plans or tactical plans. Efficiency objectives describe typically an effect:

* Increased productivity
* Improved working environment
* More effective work processes

Example of efficiency goals:

* Reduce corporate loss of interest on late payment of invoices by 50% compared to the current situation
* Reduce the resources needed for invoice processing by 80% compared to the current situation
* Increase efficiency by scheduling working hours and organizing what to work on when.
* 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, multiple people working on the code at the same time.

## Result goals

**Result goals - the result of the project**. The results objectives describe what specifically will be available as a result when the project is finished. The result objectives describe a typical product (e.g. an IT system) and have:

* The product quality / features
* When it will be completed – time / progress

Example of performance:

* Introduce a common computer system for ordering and invoice processing (description of the overall quality and characteristics) during the 2nd quarter of 2008.
* Enter agreements with our 10 largest suppliers on a monthly based handling of invoices for all purchases
* Develop new work procedures in the billing department so that control of an invoice differentiated by the invoiced amount within 1 quarter of 2008

We can say that the effect goals are the primary basis for at least doing something, and that result goals are what we choose to do to achieve the desired effect

* 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

## Process goals

In some, but not all projects it can also be useful to formulate a third type: **process goal**. These are goals that are not directly related to what the project will produce, but the **process** the project will go through to make a product. Short formulated, we can say that the process objectives are the effects the project work will have on its project participants and will be a combination of individual and collective expectations for the project. Process goals may involve:

* Competence
* Cooperation Training
* Be familiar

It is less common to formulate process goals, but in student projects it is natural - human resources development, corporate training and good grades will be typical project objectives in a student project. An example of process goals are taken from a project where students aimed to:

* Realize self-development in terms of improved skills in collaboration and communication
* Get to know other students
* To pass the subject

In such a case, the process goals are important to clarify because it is important that participants are familiar with each other's expectations and ambitions with what they decided to do.

* Pass the subject with a preferable grade.
* Improve skills in collaboration and communication.
* Improve competence within the subject.
* Get to know the other students and make bonds.

# User Descriptions

*[To effectively provide products and services that meet your stakeholders’ and users' real needs, it is necessary to identify and involve all of the stakeholders as part of the Requirements Modeling process. You must also identify the users of the system and ensure that the stakeholder community adequately represents them. This section provides a profile of the stakeholders and users involved in the project, and the key problems that they perceive to be addressed by the proposed solution. It does not describe their specific requests or requirements as these are captured in a separate stakeholder requests artifact. Instead, it provides the background and justification for why the requirements are needed.]*

## User Summary

*[Present a summary list of all identified users.]*

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| *Student* | *Final user 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* |

## User Environment

The application will have to be used on a computer with Java installed.

*[Detail the working environment of the target user. Here are some suggestions:*

* *Number of people involved in completing the task? Is this changing?*
* *How long is a task cycle? Amount of time spent in each activity? Is this changing?*
* *Any unique environmental constraints: mobile, outdoors, in-flight, and so on?*
* *Which systems platforms are in use today? Future platforms?*
* *What other applications are in use? Does your application need to integrate with them?*

*This is where extracts from the Business Model could be included to outline the task and roles involved and so on.]*

## User Profiles

*[Describe each unique user of the system here by filling in the following table for each user type. Remember user types can be as divergent as gurus and novices. For example, a guru might need a sophisticated, flexible tool with cross-platform support, while a novice might need a tool that is easy to use and user-friendly. A thorough profile needs to cover the following topics for each type of user.]*

### 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

*[List the key problems with existing solutions as perceived by the stakeholder or user. Clarify the following issues for each problem:*

*• What are the reasons for this problem?*

*• How is it solved now?*

*• What solutions does the stakeholder or user want?]*

*[It is important to understand the* ***relative*** *importance the stakeholder or user places on solving each problem. Ranking and cumulative voting techniques indicate problems that* ***must*** *be solved versus issues they would like addressed.*

*Fill in the following table—if using Rational RequisitePro to capture the Needs, this could be an extract or report from that tool.]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| Add events | High |  | none | | The user can add events at the main screen with the use of a single button. |
| 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. |
| 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. |
| Edit Events | Medium |  | none | | The user can select an event and then choose to edit its elements. |
| Reminder | Low |  | none | | User get reminded of events in advance |

# Product overview

*[This section provides a high level view of the product capabilities, interfaces to other applications, and system configurations. This section usually consists of three subsections, as follows:*

*• 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.

*• Assumptions and dependencies]*

*HER!*

## 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.

*[List and briefly describe the product features. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. Each feature is an externally desired service that typically requires a series of inputs to achieve the desired result. For example, a feature of a problem tracking system might be the ability to provide trending reports. As the use-case model takes shape, update the description to refer to the use cases.*

*Because the* ***Vision*** *document is reviewed by a wide variety of involved personnel, the level of detail needs to be general enough for everyone to understand. However, enough detail must be available to provide the team with the information they need to create a use-case model.*

*To effectively manage application complexity, we recommend for any new system, or an increment to an existing system, capabilities are abstracted to a high enough level so 25-99 features result. These features provide the fundamental basis for product definition, scope management, and project management. Each feature will be expanded in greater detail in the use-case model.*

*Throughout this section, each feature will be externally perceivable by users, operators or other external systems. These features need to include a description of functionality and any relevant usability issues that must be addressed. The following guidelines apply:*

*• Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.*

*• If you are using the Rational RequisitePro toolkit, all need to be selected as requirements of type for easy reference and tracking.]*

### 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 subsection of the* ***Vision*** *document puts the product in perspective to other related products and the user’s environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]*

Anta at brukeren ikke har en todolist app, hva kan appen vår gi?

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

*[Summarize the major benefits and features the product will provide. For example, a* ***Vision*** *document for a customer support system may use this part to address problem documentation, routing, and status reporting without mentioning the amount of detail each of these functions requires.*

*Organize the functions so the list is understandable to the customer or to anyone else reading the document for the first time. A simple table listing the key benefits and their supporting features might suffice. For example:]*

**Table 4-1 Customer Support System**

|  |  |
| --- | --- |
| **Customer Benefit** | **Supporting Features** |
| New support staff can quickly get up to speed. | Knowledge base assists support personnel in quickly identifying known fixes and workarounds. |
| Customer satisfaction is improved because nothing falls through the cracks. | Problems are uniquely itemized, classified and tracked throughout the resolution process. Automatic notification occurs for any aging issues. |
| Management can identify problem areas and gauge staff workload. | Trend and distribution reports allow high level review of problem status. |
| Distributed support teams can work together to solve problems. | Replication server allows current database information to be shared across the enterprise. |
| Customers can help themselves, lowering support costs and improving response time. | Knowledge base can be made available over the Internet. Includes hypertext search capabilities and graphical query engine. |

## Assumptions and Dependencies

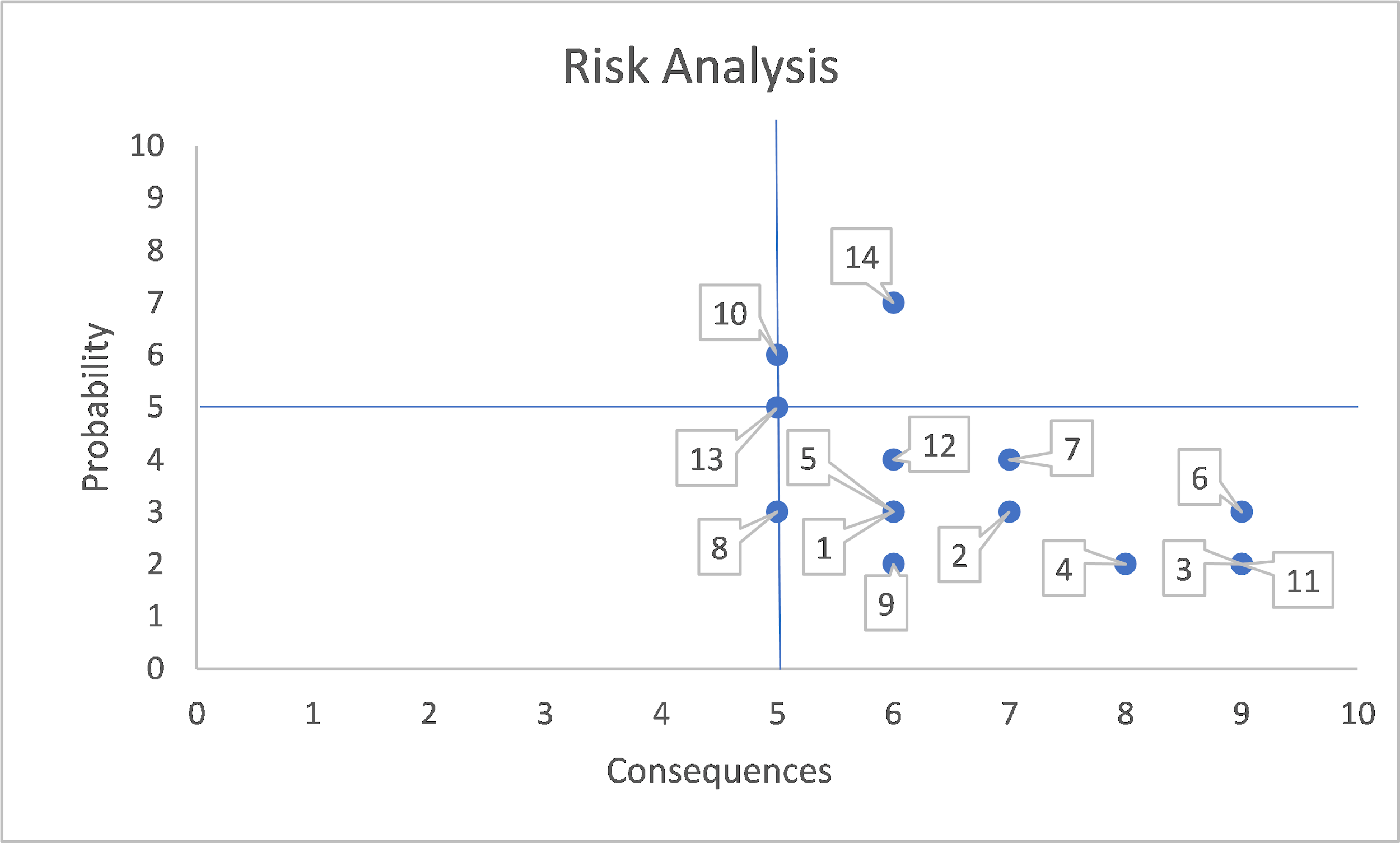
*[List each of the factors that affect the features stated in the* ***Vision*** *document. List assumptions, that if changed, will alter the* ***Vision*** *document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the* ***Vision*** *document will need to change.]*

Assumptions:

* This app will be available for the target group within the deadline.
* This app will deliver the product asked for in the project assignment.

# Risk analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |



# Cost pricing and benefits

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.

*[For products sold to external customers and for many in-house applications, cost and pricing issues can directly impact the application’s definition and implementation. In this section, record any cost and pricing constraints that are relevant. For example, distribution costs, (# of diskettes, # of CD-ROMs, CD mastering) or other cost of goods sold constraints (manuals, packaging) may be material to the project success, or irrelevant, depending on the nature of the application.]*

*[Set by Marketing, the product manager or the business analyst. All requirements are not created equal. Ranking requirements by their relative benefit to the end user opens a dialog with customers, analysts, and members of the development team. Used in managing scope and determining development priority.]*

This is an important basis part of the project to be implemented. The purpose is to determine whether the benefits of the project are worth the cost of implementing it. It is not certain, that we conduct a cost / benefit analysis of a project. It depends on whether the customer wishes such an analysis.

In the analysis we will first make an estimate of the effect of the project - expected benefits. Then we make estimates of what the project will cost. Finally, we compare the two variables to assess whether the benefits are worth the costs.

A cost / benefit analysis can be made more or less advanced and scientifically. In its most advanced form, it is very laborious and a task for experts, i.e. economists. In smaller projects, it may be sufficient with a simplistic statement of expected benefits and expected costs and not to mention the actual process by discussing this with the client will have value even if the analysis is simple.

Whether we initiate an advanced or a simple analysis, it is very important that before we start making it clear that we compare:

* In general, we compare it to complete a project against the so-called 0 - alternative - to do nothing.

A simple cost / benefit analysis has the following elements:

# Licensing and 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.

*[Licensing and installation issues can also directly impact the development effort. For example, the need to support serializing, password security or network licensing will create additional requirements of the system that must be considered in the development effort.*

*Installation requirements may also affect coding or create the need for separate installation software.]*

# 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.

*[Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.]*

# 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.

*[Define the priority of the different system features.]*

# Other Product Requirements

*[At a high level, list applicable standards, hardware or platform requirements, performance requirements, and environmental 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.

*[List all standards with which the product must comply. These can include legal and regulatory (FDA, UCC) communications standards (TCP/IP, ISDN), platform compliance standards (Windows, UNIX, and so on), and quality and safety standards (UL, ISO, CMM).]*

## System Requirements

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

*[Define any system requirements necessary to support the application. These can include the supported host operating systems and network platforms, configurations, memory, peripherals, and companion software.]*

## 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.

*[Use this section to detail performance requirements. Performance issues can include such items as user load factors, bandwidth or communication capacity, throughput, accuracy, and reliability or response times under a variety of loading conditions.]*

## Environmental Requirements

*[Detail environmental requirements as needed. For hardware- based systems, environmental issues can include temperature, shock, humidity, radiation, and so forth. For software applications, environmental factors can include usage conditions, user environment, resource availability, maintenance issues, and error handling and recovery.]*

# Documentation Requirements

*[This section describes the documentation that must be developed to support successful application deployment.]*

## 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.

*[Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial versus reference manual strategy, and so on. Formatting and printing constraints must also be identified.]*

## Online Help

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.

*[Many applications provide an online help system to assist the user. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, and so forth) with aspects of technical writing, such as organization and presentation. Many have found the development of an online help system is a project within a project that benefits from up-front scope management and planning activity.]*

## 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.

*[A document that includes installation instructions and configuration guidelines is important to a full solution offering. Also, a Read Me file is typically included as a standard component. The Read Me file can include a "What's New With This Release” section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the Read Me file.]*

## 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.

*[Today's state-of-the-art applications provide a consistent look and feel that begins with product packaging and manifests through installation menus, splash screens, help systems, GUI dialogs, and so on. This section defines the needs and types of labeling to be incorporated into the code. Examples include copyright and patent notices, corporate logos, standardized icons and other graphic elements, and so forth.]*

# 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.]*