
BOB

**Bee on Budget (BOB)
Vision**

Version <1.0>



**BEE ON
BUDGET**

POLLINATE YOUR PROFIT

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Revision History

Date	Version	Description	Author
07/02/23	<1.0>	Initial outline of document.	Simon, Emil, Eskil, Erik
08/02/23	<1.2>	Added information to the Positioning chapter, Project Goals and Alternative competition (4.6).	Simon, Emil, Eskil, Erik
14/02/23	<1.3>	Product Features and Bibliography.	Simon, Emil, Eskil, Erik
15/02/23	<1.4>	Added risk analysis (5.4) and User environment (4.3).	Simon, Emil, Eskil, Erik, Igor
19/02/23	<1.5>	Type checking and refactoring.	Emil
20/02/23	<1.6>	Added Product Owner information and Constraints.	Simon, Emil, Eskil, Erik
21/02/23	<1.7>	Added chapter 9 and 12.	Simon, Emil
22/02/23	<1.8>	Project overview (1.4) and Documentation Requirements.	Simon, Emil, Erik
23/02/23	<1.9>	Finished chapter 5.	Emil
27/02/23	<1.10>	Finished chapter 9 and 12	Simon, Emil
28/02/23	<1.11>	Finished Introduction chapter.	Simon, Emil, Eskil, Erik
01/03/23	<1.12>	Added Use-Case descriptions.	Simon, Emil, Eskil, Erik, Igor
06/03/23	<1.13>	Added final changes and revisions to document and revision history	Emil, Eskil, Igor
07/03/23	<1.14>	Type checking and quality assurance.	Emil, Simon, Erik, Eskil
08/03/23	<1.15>	Final check before delivery	Emil, Simon, Eskil, Igor, Erik
17/04/23	<1.16>	Added recommended changes after Sprint review meeting 1	Emil
24/04/23	<1.17>	Added small changes to risks and did a final read through	Emil

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Table of Contents

1.	Introduction	8
1.1	Purpose and scope	8
1.2	Definitions, Acronyms, and Abbreviations	8
1.3	Overview	9
2.	Positioning	9
2.1	Business Opportunity	9
2.2	Problem Statement	9
2.3	Product Position Statement	10
3.	Project goals	10
3.1	Impact goals	10
3.2	Result goals	11
3.3	Process goals	11
4.	Stakeholder and User Descriptions	11
4.1	Stakeholder Summary	11
4.2	User Summary	12
4.3	User Environment.	12
4.4	Key Stakeholder or User Needs	12
4.5	Alternatives and Competition	13
4.5.1	Mint	13
4.5.2	Personal Capital: Finance	13
4.5.3	DNB Spare	13
5.	Product Overview	14
5.1	Product Perspective	14
5.2	Summary of Capabilities	14
5.3	Assumptions and Dependencies	14
5.4	Risk analysis	14
5.4.1	Scrum team risks	14
5.4.2	Security and personal data	15
5.5	Cost, Pricing, and benefits	15
5.6	Quantifiable and non-quantifiable benefits	15
5.7	Loss of direct costs	15

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

5.8	Estimated costs	16
5.9	Licensing and Installation	16
6.	Product Features	16
6.1	Login/Registration Feature	16
6.2	Savings Plan Feature	16
6.3	Budget Feature	16
6.4	Add Expenses	16
6.5	Diagram Feature	16
6.6	Finance Calendar Feature	16
6.7	Graphical Interface	17
7.	Use-Case Descriptions	17
7.1	Register	17
7.2	Log in	17
7.3	Create Savings Plan	18
7.4	Show Savings Plan	19
7.5	Show Finance Calendar	19
7.6	Show Diagrams	20
7.7	Add Expenses	20
7.8	Show Budget	22
8.	Constraints	23
9.	Quality Ranges	23
10.	Precedence and Priority	23
11.	Other Product Requirements	23
11.1	Applicable Standards	23
11.2	System Requirements	24
11.3	Performance Requirements	24
11.4	Environmental Requirements	24
12.	Documentation Requirements	24
12.1	User Manual	24
12.2	Online Help	24
12.3	Installation Guides, Configuration, and Read Me File	25

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

12.4	Labeling and Packaging	25
13.	A Feature Attributes	25
13.1	A.1 Status	25
13.2	A.2 Effort	26
13.3	A.3 Stability	26
13.4	A.4 Target Release	26
13.5	A.5 Assigned To	26
13.6	A.6 Reason	27
14.	Bibliography	27

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Vision

1. Introduction

In our everyday life, people often struggle to have enough money to use after paying all our monthly expenses, because they struggle to structure their economy. Many have big problems with prioritizing and must take loans and pay by credit, which only escalates their economic problems to a greater degree. To help fix this, the Norwegian bank “BN bank” recommend that normal people create a budget so that they can track and prioritize their spendings in a way that is more sustainable for their everyday life [1]. Therefore, many people download budget apps that they use on their mobile phones or computers to have control more easily over their economy. The problem with some of these apps is that their functionality and user friendliness is good in some cases, but in other cases it is extremely hard to use and does not have the most efficient functionality. Because of this we have decided to create a new application ourselves that has the most important functionality for budget apps and displays it in quite a simple and effective way.

1.1 Purpose and scope

This document's goal is to outline the first assumptions and the application's planned design features. Furthermore, aside from client requirements this document also includes an assessment of potential risks and failures within the development process, costs of development and implementation of said application and a discussion of product security. The purpose of these descriptions is to provide the reader with a complete understanding of the planning and early design phases of this project to have a better grasp of the final product and supply a baseline for evaluation of the product and/or potential shortcomings.

1.2 Definitions, Acronyms, and Abbreviations

Definitions, acronyms, or abbreviations	Explanation
BOB	Bee on Budget, the name of the application
MVP	Minimum Viable Product
UML	Unified Modeling Language
HCI	Human Computer Interaction
JRE	Java Runtime Environment
GUI	Graphical User Interface
GDPR	General Data Protection Regulation

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

1.3 Overview

The Vision document has 13 sections. The first section in this vision document is the Introduction section, which gives the reader a brief overview of our vision. Next the document transitions into the Positioning and Project Goal chapters, which describe our business opportunity and our development goals. From chapter four to eight we delve deeper into the matter, and describe the product owner, the stakeholders and the product overview with the constraints and quality ranges. Then we describe how we plan to carry out the final product with which releases that will have the unique features and which requirements the application has in relation to hardware and software. In the last three chapters in the document, we will say the need of documentation, how the user manual will be implemented in the application, the feature attributes and lastly a list of all the sources we have used in the development of the application.

2. Positioning

2.1 Business Opportunity

The business opportunity for this project was started by the Product Owner for this application who wishes to get control of his economy. Today, having control of your own economy is one of the key parts that make your everyday life successful in this application the product owner wants the opportunity to create a future saving goal with the help of a standard budget application setup with the most common functions.

2.2 Problem Statement

The problem of	- Trying to aid the economic situation and plan the monthly spendings according to the user's income.
affects	- People who want control of their everyday life economy
the impact of which is	- Users not being able to control their economy and use loads of unnecessary money that could have been saved for later.
a successful solution would be	<ul style="list-style-type: none"> - An application with diagrams that shows the users spending history and gives the user a fitting solution. - An application that creates budgets by looking at the users spendings and incomes to automatically creates a reasonable budget that can make the user save the most amount of money.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

2.3 Product Position Statement

For	Lower to middle class people
Who	Control of economy
The (product name)	Bee on Budget (BOB)
That	Simple, elegant, and easy to use
Unlike	Mint, DNB Spare and Personal Capital.
Our product	Personalized for the product owners' needs

3. Project goals

During an application development process, it is important to note some project goals that can work as milestones in the process. For this project, we have chosen to set our goals in collaboration with our product owner, so that we can ensure that the project's result will satisfy his requirements. In addition to this we have also set some goals to have a collective understanding of the project group and what that job entails, and to learn the ability to plan and manage a group project. Since these goals vary in what they aim to achieve, we have divided these goals into three categories. The first category of goals are the impact goals, which are the goals that our product owner wants to achieve with the project. The second, are the result goals which describe the project result. The last chapter has the process goals that describe the development process the application will go through.

3.1 Impact goals

In relation to this application idea the Product Owner presented a personal wish for a user function that he wanted the scrum team to implement in the application's definitive version. This user function needed to include the possibility to create a future saving goal, with an inserted amount of money for each month or week. In addition, he wanted the application to have a quite simple and user-friendly design with most of the standard functionality that other budget applications have, for example DNB Spare. But in relation to DNB Spare, our app will include a less complicated user interface, which will be easier to use. Our will User interface will be much less time consuming to use since it will contain much bigger icons and buttons that will make it easier for users with bad sight to press the correct buttons in the application. This change also minimizes the possibility of missclicks and the buttons will become easier to click. Some of these functions will be the ability to create a budget with a standard set of categories, add expenses, add income, show finance calendar and a create budget function.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

3.2 Result goals

Throughout the development process, we will set interim goals for when the major parts of the application should be finished. We will have a Minimal Viable Product ready by week 13 of 2023, but the full release of the application will be in week 17 of 2023. The released version will have the core functions that the Scrum team and Product Owner agreed upon. In addition, it will be suitable for other stakeholders, like owners of small companies or ordinary people that might need the possibility to create a budget for their financial incomes and expenses in their everyday life.

3.3 Process goals

The interim goals for the project will be set by the start of each sprint, which lasts 2 weeks each. During these sprints we will get a deeper understanding of the Scrum as an agile development strategy, and how a working day is structured for a developer in a Scrum team. Additionally, the Scrum team will also learn much about Human-Computer Interaction (HCI) and the process to create a user-friendly application by trying to meet the Product owner's demands. By carrying out these goals, we hope to pass the subject with a good grade.

4. Stakeholder and User Descriptions

Ruben Aune came with the need of application to get an overview over his own budget. He wanted an application that could budget his money month by month and could make a saving goal.

4.1 Stakeholder Summary

Name	Description	Responsibilities
Sander Sørhagen	Is interested in investing and always on the lookout for a way to save a penny.	Monitors the projects progress. To see if the application has any interesting features he is looking for in his daily life.
Frederic Bendixen	Is interested if the application turns out the way it is advertised.	Just a person who is interested in the application.
Ryan Slater	Is interested in how we handle the financial approach in the application.	Monitors how the application supports financial support.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

4.2 User Summary

Name	Description	Responsibilities	Stakeholder
Ruben Aune	This is the end user of the program.	This user came with the requirement of the program.	

4.3 User Environment.

The application is meant to be used by one person at a time. The user can spend more time in the application when setting their budget or savings plan up. Other activities like just checking your budget, the finance calendar etc. will take less time. The application will only be used on a computer and not on the move. In the future the application could be added to mobile platforms, for example smartphones.

4.4 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Saving Plans	High	Application	See the wireframe for the current solution.	Creating a saving plan by using a previously created budget.
Show diagram	Medium	Application	See the wireframe for the current solution.	Creation different diagrams by using current budget and expenses.
Register user	Low	Application	See the wireframe for the current solution.	Creation a user to use in the application saved local on the computer. Using email and pin code.
Login	Low	Application	See the wireframe for the current solution.	Login to chosen user, with email and pin code.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Add Expenses	High	Application	See the wireframe for the current solution.	Add an expense to the budget using category, date, and price.
Show Calendar	Medium	Application	See the wireframe for the current solution.	See the current budget in the calendar, where you see the monthly expenses due date.
Budget	High	Application	See the wireframe for the current solution.	Look at expenses and sort them. See the amount of the budget used.

4.5 Alternatives and Competition

4.5.1 *Mint*

Mint was the first personal finance app to aggregate the user's accounts into one place. Mint's major strength is the overall usability of the app, it is easy to create saving goals or check your current net balance. Mint lets the user do budgeting by category and track upcoming bills. The major weakness of Mint compared to its competitors is the lack of investing tools [3].

4.5.2 *Personal Capital: Finance*

Personal Capital: Finance is one of Personal Capital's applications for mobile platforms. The budget application is free, but \$100,000 is needed to start an account with Personal Capital. The major strength of the app is the investing tools integrated into the app. The major weakness is that the budgeting tools are not as great as Mint [4].

4.5.3 *DNB Spare*

DNB Spare is a primary competitor in Norway, as it is made by the largest bank in Norway [2]. Spare has the primary strength of being integrated with the bank, so the experience for the user is seamless. The integration with DNB also allows for payments and purchases through the app, as well as integration with the global stock market. The major weaknesses of the application are that the UX and UI can be complicated for some users and the application requires an account with DNB to function optimally.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

5. Product Overview

5.1 Product Perspective

The application is a stand-alone product, and since everything is saved locally, the application is self-contained and runs regardless of the operating system. The only requirement is to have the Java Runtime Environment installed to run the application.

5.2 Summary of Capabilities

Table 4-1 Budget Application

Customer Benefit	Supporting Features
Customers will get an organized overview of their finances.	The budget feature lets you add all your expenses and incomes and categorize them.
Customers can plan their finances for the future.	Setting up a savings plan and having a finance calendar shows future expenses/savings.
Customers can see how their finances have changed over time.	The diagrams give an overview of changes over time.

5.3 Assumptions and Dependencies

The application will use JavaFx as the user interface. If JavaFx does not have features for all the functions planned, like for example diagrams, the project vision will need to change.

5.4 Risk analysis

5.4.1 Scrum team risks

While developing this application the Scrum-team have a medium risk of delays in the project. These delays might occur when many of the Scrum-team members become ill at once, or do not attend the weekly team-meetings and delay their contribution to critical parts of the project. To minimize the risk of this, each member in the scrum team has signed a group contract that says that a member will be punished if the rules have not been kept. One of the punishments is that the group member needs to buy coffee or energy drinks for all group members. If a group member breaks the rules many times, the other group members will discuss if it will be necessary to exclude a member from the team. There is also some risk connected to the stakeholders and product owner. For example, if we as the developers do not meet the requirements from the product owner and stakeholders. This could lead to the

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

scrum team losing a lot of credibility in the market and overall satisfaction from the product owner. Another risk is that the scrum team use too much time on trying to learn new and complicated programming tools, so that there will not be much time left to use at the end of the project. Because of this the scrum team would have lost a lot of valuable time, the overall quality of the project would be weakened, and the application would probably have many unfinished parts with many bugs.

5.4.2 Security and personal data

When creating a budget application, it is important to keep in mind that the data that the users leave behind might be overly sensitive. In competing budget applications, the users leave much information about their bank accounts and their income and expenses, as well as their loans and investments. Therefore, it is important to make an application that successfully takes care of the user's information before the application gets released. If the application had been released with these faults the user's security would be at substantial risk, and the application's success on the market would be exceptionally low and we would struggle to obtain any user recommendations. Therefore, the development should ideally create a separate database and store passwords here by encrypting them, like the developers of the Mint application have done. Within the given amount of time, we must create this application, we must minimize the functionality of the application. Because of this, the user will not be able to link the application to their bank account. Instead, the user will create a budget by manually inserting their expenses, but we will try to implement a login service that makes it possible to store your data by creating your own username and password.

5.5 Cost, Pricing, and benefits

If the product is to be distributed, it will all happen digitally and on a smaller scale so there will not be any cost associated with this. There is no planned marketing, but this might change if the scale of the product grows.

5.6 Quantifiable and non-quantifiable benefits

The application will bring multiple benefits to the user. The user will be able to create an organized overview of their finances using both a budget and savings plan feature. The application will be easy to use, leading to less frustration and stress surrounding their money management. The savings plan feature is beneficial to the user by letting them set goals to save money on a regular basis. The calendar lets the user add and see all future events, so they do not miss any upcoming invoices/payments.

5.7 Loss of direct costs

There is no cost for developing this application so the loss will be zero. The application is not made for commercial use, only the product owner and the stakeholders that have watched the application's development process will get access to it.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

5.8 Estimated costs

Considering the nature of the project and the client base, the costs of development, implementation and operation can be kept minimal. As this application is developed within the NTNU “Systemutvikling” course, all group members are effectively volunteers, and as such, do not require pay for work time.

The expected userbase is the sole client that this application is tailored towards, that is, the product owner, which allows the product to be designed to run locally on a single desktop. Therefore, the implementation and distribution of said product is limited, and the expected cost is none.

5.9 Licensing and Installation

The only 3rd party software needed is Java Runtime Environment. Other than this the application is fully independent, and all needed features will be created by the development team.

6. Product Features

6.1 Login/Registration Feature

A login and registration feature will allow the user to secure their information from others and have multiple users per device.

6.2 Savings Plan Feature

Will allow the user to both create and show a savings plan. This is a method for amassing money towards a financial goal. Additionally, the application will estimate the time needed to reach the goal.

6.3 Budget Feature

Allows the user to set up a budget with a table to get an overview of their finances.

6.4 Add Expense

The user can add expenses and categorize these to get a more organized view of the finances.

6.5 Add Income

The user can add their incomes to the budget with a suitable category, like for example work, investment profit or salary.

6.6 Diagram Feature

Allows the user to see how their finances have changed over time in a suitable type of diagram.

6.7 Finance Calendar Feature

A calendar will allow the user to see future expenses, incomes, and invoices they have added.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

6.8 Graphical Interface

A graphical interface will let the user easily navigate the application while also making it more aesthetically pleasing.

7. Use-Case Descriptions

In our application we have created an overview of all the application's areas of use by modeling it with a Use-Case diagram. In the Use case diagram. In the Use-Case we have included the most key features and coupled these with the user that is going to use them. These functions are registration/log in, login successfully, show diagram, show budgets, create savings plan, show savings plan, create savings plan, add expenses, add income, and show finance calendar. In the following chapters, we will explain these with Use-Case descriptions with the different conditions you need to fulfill to use the functions.

7.1 Register

UC-name	Register
Actor	User
Pre-condition	The program is installed and running correctly
Post-condition	The user has created an account
Primary Flow	<ol style="list-style-type: none"> 1. User enters full name. 2. User enters e-mail. 3. User enters a 4-digit pin code. 4. User picks a security question and an answer to it. 5. User clicks "Register account."
Exception	<ol style="list-style-type: none"> 2.1. User enters an invalid e-mail. <ol style="list-style-type: none"> 2.1.1. Application gives feedback to user about the invalid e-mail. 2.1.2. Go back to point 2 and enter a new e-mail. 3.1. User enters an invalid pin code. <ol style="list-style-type: none"> 3.1.1. Application gives feedback to user about the invalid pin code. 3.1.2. Go back to point 3 and enter a new pin code.

7.2 Log in

UC-name	Log in
Actor	User
Pre-condition	The user has created an account
Post-condition	The user has logged in
Primary Flow	<ol style="list-style-type: none"> 1. User enters e-mail. 2. User enters pin code. 3. User clicks "Login"
Exception	<ol style="list-style-type: none"> 1.1 User enters an invalid e-mail. <ol style="list-style-type: none"> 1.1.1 Application gives feedback to the user about invalid e-mail.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

	<p>1.1.2. Go back to point 1 and enter a new e-mail.</p> <p>2.1. User enters an invalid pin code.</p> <p>2.1.1 Application gives feedback to the user about invalid pin code.</p> <p>2.1.2. Go back to point 2 and enter a new pin code.</p> <p>4.1 User forgot pin code.</p> <p>4.1.1 User enters e-mail used for the account.</p> <p>4.1.2 User clicks "Enter email."</p> <p>4.1.3 User enters the answer to security question.</p> <p>4.1.4 User enters a new valid pin code.</p> <p>4.1.4.1 User enters an invalid pin code.</p> <p>4.1.4.2 Application gives feedback to user about the invalid pin code.</p> <p>4.1.4.3 Go back to point 4.1.4 and enter a new pin code.</p> <p>4.1.5 User clicks "Set new Pin Code"</p>
--	--

7.3 Create Savings Plan

UC-name	Create Savings Plan
Actor	User
Pre-condition	The user is logged in
Post-condition	A savings plan is created, and it can be shown
Primary Flow	<ol style="list-style-type: none"> The user logs in. The user presses the savings plan button. The user adds the monthly saving amount, the amount that the user already has saved and adds the savings plan date. The user is shown the new savings plan.
Exception	<p>1.1 The user enters an invalid e-mail.</p> <p>1.1.1 Application gives feedback to user about the invalid e-mail.</p> <p>1.1.2. Go back to point 1 and enter a new e-mail.</p> <p>1.2 The user enters an invalid pin code.</p> <p>1.2.1 Application gives feedback to the user about invalid pin code.</p> <p>1.2.2. Go back to point 1 and enter a new pin code.</p> <p>3.1 The max number of saving plans is reached.</p> <p>3.1.1 Application gives feedback to user about the limit.</p> <p>3.1.2 The user removes another saving plan.</p> <p>3.1.3 The user can create a new savings plan.</p> <p>4.1 No savings plan exists.</p> <p>4.1.1 The application gives feedback that no savings plan exists.</p> <p>4.1.2 Go back to point 3 and create a new savings plan.</p>

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

7.4 Show Savings Plan

UC-name	Show Savings Plan
Actor	User
Pre-condition	The user is logged in and a savings plan already exists
Post-condition	The savings plan has been shown to the user
Primary Flow	<ol style="list-style-type: none"> 1. The user logs in 2. The user shows one of their savings plans. 3. The user adds a new savings amount
Exception	<ol style="list-style-type: none"> 1.1 The user enters an invalid e-mail. <ol style="list-style-type: none"> 1.1.1 Application gives feedback to user about the invalid e-mail. 1.1.2. Go back to point 1 and enter a new e-mail. 1.2 The user enters an invalid pin code. <ol style="list-style-type: none"> 1.2.1 Application gives feedback to the user about invalid pin code. 1.2.2. Go back to point 1 and enter a new pin code. 2.1 No savings plan exists. <ol style="list-style-type: none"> 2.1.1 The application gives feedback that no savings plan exists. 2.1.2 Go back to point 3 and create a new savings plan. 3.1 The user enters an invalid value for the new savings amount. <ol style="list-style-type: none"> 3.1.1 The application gives feedback about the invalid value. 3.1.2 Go back to point 3 and enter a valid value.

7.5 Show Finance Calendar

UC-name	Show Finance Calendar
Actor	User
Pre-condition	<ol style="list-style-type: none"> 1. That the user is registered. 2. The user has logged in to the application. 3. Created a savings plan.
Post-condition	The saving plan has a specific date.
Primary Flow	<ol style="list-style-type: none"> 1. The user logs in. 2. The user creates a saving plan and adds it to a specific date. 3. The user goes back to the main menu and selects the “calendar function.”
Exception	<ol style="list-style-type: none"> 1.1 The user enters an invalid e-mail. <ol style="list-style-type: none"> 1.1.1 Application gives feedback to user about the invalid e-mail. 1.1.2. Go back to point 1 and enter a new e-mail. 1.2 The user enters an invalid pin code. <ol style="list-style-type: none"> 1.2.1 Application gives feedback to the user about invalid pin code. 1.2.2. Go back to point 1 and enter a new pin code. 2.1 The max number of saving plans is reached. <ol style="list-style-type: none"> 2.1.1 Application gives feedback to user about the limit. 2.1.2 The user removes another saving plan. 2.1.3 The user can create a new savings plan.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

	<p>2.2 The user enters an invalid value for the new savings amount.</p> <p>2.2.1 The application gives feedback about the invalid value.</p> <p>2.2.2 Go back to point 3 and enter a valid value.</p> <p>3.1 No savings plan is created.</p> <p>3.1.1 The savings plan dates are not shown in the calendar.</p> <p>3.1.2 Go back to point 2 and create a new savings plan.</p>
--	--

7.6 Show Diagrams

UC-name	Show Diagrams
Actor	User
Pre-condition	User is registered and logged in, and has made a budget with incomes/expenses
Post-condition	Diagrams have been shown
Primary Flow	<ol style="list-style-type: none"> The user logs in. The user shows diagrams.
Exception	<p>1.1 The user enters an invalid e-mail.</p> <p>1.1.1 Application gives feedback to user about the invalid e-mail.</p> <p>1.1.2. Go back to point 1 and enter a new e-mail.</p> <p>1.2 The user enters an invalid pin code.</p> <p>1.2.1 Application gives feedback to the user about invalid pin code.</p> <p>1.2.2. Go back to point 1 and enter a new pin code.</p> <p>2.1 No data to show in the diagrams.</p> <p>2.1.1 The user adds a new expense or income.</p> <p>2.1.2 Go back to point 2 and show diagrams.</p>

7.7 Add Expenses

UC-name	Add Expenses
Actor	User
Pre-condition	Has a user and logged in, set up their budget
Post-condition	Expenses added to budget
Primary Flow	<ol style="list-style-type: none"> The user logs in. The user presses the “add expense” button. The user chooses an expense category. Set the description of the expense. Set the amount of the expense. The user presses “the submit”. button.
Exception	<p>1.1 The user enters an invalid e-mail.</p> <p>1.1.1 Application gives feedback to user about the invalid e-mail.</p> <p>1.1.2. Go back to point 1 and enter a new e-mail.</p> <p>4.1 The user did not enter any expense description.</p>

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

	<p>4.1.1 The expense will be submitted, but the budget will not show any description.</p> <p>5.1 The user tries to enter an illegal expense amount. 5.1.1 The application prevents the user from typing the illegal symbols.</p> <p>5.2 The user did not enter any expense amount. 5.2.1 The method will not add any information to the budget.</p> <p>3.1 The user has not chosen an expense category from the list. 3.1.1 The user will not be able to create the expense. 3.1.2 Go back to point 3.</p> <p>4.1 The user has not chosen a recurring interval from the list. 4.1.1 The user will not be able to create the expense. 4.1.2 Go back to point 4.</p> <p>6.1 The user has not entered an expense amount. 6.1.1 The user will not be able to create the expense. 6.1.2 Go back to point 6.</p> <p>7.1 Nothing happens when the user presses “submit”. 7.1.1 One of the input values for the expense is invalid. 7.1.2 Go back to point 3 and enter a valid input for each value.</p>
--	--

7.8 Add Income

UC-name	Add Income
Actor	User
Pre-condition	Has a user and logged in, set up their budget
Post-condition	Income added to budget
Primary Flow	<ol style="list-style-type: none"> 1. The user logs in. 2. The user presses the “add income” button. 3. Choose an income category. 4. Choose a recurring interval. 5. Set the description of the income. 6. Set the amount of the income. 7. The user presses the submit button.
Exception	<p>1.1 User enters an invalid e-mail. 1.1.1 Application gives feedback to the user about invalid e-mail. 1.1.2. Go back to point 1 and enter a new e-mail.</p> <p>6.1 The user tries to enter an illegal income amount. 6.1.1 The application prevents the user from typing the illegal symbols.</p> <p>7.1 The user did not enter any income description. 7.1.1 The income will be submitted, but the budget will not show any description.</p> <p>7.2 The user did not enter any income amount.</p>

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

	7.2.1 The method will not add any information to the budget.
--	--

7.9 Show Budget

UC-name	Show Budget
Actor	User
Pre-condition	Has a user and logged in
Post-condition	The budget is shown to the user
Primary Flow	<ol style="list-style-type: none"> 1. The user logs in. 2. The user presses the “budget” button. 3. The user presses the “new expense/” button. <ol style="list-style-type: none"> 3.1 Choose an expense category. 3.2 Choose a recurring interval. 3.3 Set the description of the expense. 3.4 Set the amount of the expense. 3.5 The user presses the submit button. 4. The user presses the “new income” button. <ol style="list-style-type: none"> 4.1 Choose an income category. 4.2 Choose a recurring interval. 4.3 Set the description of the income. 4.4 Set the amount of the income. 4.5 The user presses the submit button. 5. The information gets shown in the budget.
Exception	<ol style="list-style-type: none"> 1.1 The user enters an invalid e-mail. <ol style="list-style-type: none"> 1.1.1 Application gives feedback to user about the invalid e-mail. 1.1 User enters an invalid e-mail. <ol style="list-style-type: none"> 1.1.1 Application gives feedback to the user about invalid e-mail. 1.1.2. Go back to point 1 and enter a new e-mail. 3.3.1 The user did not enter any expense description. 3.3.2 The expense will be submitted, but the budget will not show any description. 3.4.1.1 The user tries to enter an illegal expense amount. 3.4.2 The application prevents the user from typing the illegal symbols. 3.5.1 The user did not enter any expense amount. 3.5.2 The method will not add any information to the budget. 4.3.1 The user did not enter any income description. 4.3.2 The income will be submitted, but the budget will not show any description. 4.4.1The user tries to enter an illegal income amount.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

	<p>4.4.2 The application prevents the user from typing the illegal symbols.</p> <p>4.5.1 The user did not enter any income amount. 4.5.1 The method will not add any information to the budget.</p> <p>5.1 The budget does not show any income or expense. 5.1.1 The user did not add any expense or income before pressing submit.</p> <p>5.2 The budget does not show any category or recurring interval. 5.2.1 The user did not add any category or recurring interval before pressing submit.</p>
--	---

8. Constraints

The application will only allow for data to be saved locally as opposed to in the cloud. Since it is only locally, there will not be a possibility to get data from 3rd party sources, for example bank information. The application will be made for desktop use only and will not be used on other types of devices, for example smartphones.

9. Quality Ranges

The application will be made robustly to minimize crashes. The program will start within a few seconds, if not instantaneously and the saving and loading of files will be made not to slow down the process.

10. Precedence and Priority

The higher priority systems in this app are the Savings Plan feature, Budget feature, Add Expenses/Expense Categories, Add Income/Income Categories and the graphical user interface. These features are ranked as the most important for our application because we have explicitly been asked by our product owner to prioritize these for our minimal practical product release on the 30th of March. After this release date we will prioritize our lower priority requirements to create the final application. These requirements are the Login/Registration, Diagram and Finance calendar features.

11. Other Product Requirements

11.1 Applicable Standards

In this application, we are regulated by the General Data Protection Regulation (GDPR) because we handle the personal information the application handles in relation to the Login/Registration function and the added income and expenses. Additionally, we are regulated by all the most specific compliance standards since we want the application to compile on all platforms.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

11.2 System Requirements

Application will work on computers running Windows, MacOS, Linux or Solaris as their operating systems. The Java Runtime Environment is the only other software needed to run the application.

11.3 Performance Requirements

For our budget application the performance requirements are incredibly low since we have planned to make an application without the ability to have many users to use the application at once. Therefore, the application will not have any problems with the number of user requests for the system at the same time.

11.4 Environmental Requirements

The application requires a computer with at least one screen and keyboard/mouse. The program will not require more than 1 GB of disk space.

12. Documentation Requirements

When the application is finished, there will be a need to document the process and make a user manual to help the user understand the varieties of functions in the applications. In this section, we will discuss creating a user manual and an installation guide.

12.1 User Manual

The User Manual guides the user on how to install, run and effectively use the application. The length of the manual would ideally be as short as possible, by only having the most essential information. The manual's quality is not to be degraded by the ambition of a short manual, but rather be clear-cut and proper. Indexing the main windows in the GUI with subpoints of functions and a glossary of financial and technical terms would increase the useability of the manual. The manual will be shipped with the program as a digital format, so the formatting and printing constraints are next to none.

12.2 Online Help

The application will also supply online help through email. The email addresses that can be contacted are eskilal@stud.ntnu.no, emilkle@stud.ntnu.no, simonhou@stud.ntnu.no, erbj@stud.ntnu.no, igormd@stud.ntnu.no.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

12.3 Installation Guides, Configuration, and Read Me File

The Readme file will include any new features added since the earlier version and any updates to existing software. In addition, there will be an installation guide on how to set up the program (setting up Java runtime for instance), a user account, and how to get started.

12.4 Labeling and Packaging

The application logo was made on looka.com. Six different logos were made with slight differences in color and labels.

13. A Feature Attributes

13.1 A.1 Status

Proposed	<ul style="list-style-type: none"> - Saving to server. - Bank Information acquired by 3rd party member. - Other platforms than PC. For example, smartphones. - More advanced GUI. - More security, for example encryption.
----------	--

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

Approved	<ul style="list-style-type: none"> - Show diagrams function (Planned for final delivery) - Show financial calendar function (Planned for final delivery) - Add registration function (Planned for final delivery) - Add login function (Planned for final delivery)
Incorporated	<ul style="list-style-type: none"> - Create budget function (Planned for MVP) - Add expense function (Planned for MVP) - Show saving plan function (Planned for MVP)

13.2 A.2 Effort

The first release of the application is made in 3 weeks to get a minimum viable product done. The second and final release should also be done in 3 weeks and consist of the final product.

13.3 A.3 Stability

Because this is the development team's first time creating an application of this sort, there is a high probability of change during the development phase. The team has little to no experience with GUIs, so this is a part of the development with high uncertainty. The final product might differ from the wireframe as the team's knowledge of what functions are available is low.

13.4 A.4 Target Release

During the development process the application will have two releases. For the first release on the 30th of March 2023, we will have a minimal workable product ready with the most important functionality. For this release we will have created the create budget, add expenses, and create saving plan features with a basic version of the user interface. A month later, on the 28th of April 2023, we will have the final application ready with more features. The planned features for the definitive version are the login/registration, diagram, and finance calendar features, with a better version of the user interface.

13.5 A.5 Assigned To

In our development team we want all members to contribute to all parts of the project so that we all learn about the various parts of the development process, like UML, Scrum and HCI.

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

13.6 A.6 Reason

A conversation with the product owner was held on Discord on 10.02.2023 where they gave their requirements for the application. The product owner also gave feedback on a user test given to them on 27.02.2023 there they wanted slight changes to the wireframe.

14. Bibliography

- [1] BN bank, «BN bank,» 2023. [Internett]. Available: <https://www.bnbank.no/sparing/sett-opp-budsjett/>. [Funnet 1 March 2023].
- [2] DNB Bank ASA, «Spare-app,» 2023. [Internett]. Available: <https://www.dnb.no/sparing/spare-app>. [Funnet 8 February 2023].
- [3] Intuit, Inc., «Mint,» Intuit, 2021. [Internett]. Available: <https://mint.intuit.com/>. [Funnet 8 February 2023].
- [4] Personal Capital Corporation, an Empower Company, «Personal Capital,» Personal Capital Corporation, 2023. [Internett]. Available: <https://www.personalcapital.com/>. [Funnet 8 February 2023].
- [5] Wikipedia, «DNB ASA,» Wikipedia, 15 January 2022. [Internett]. Available: https://en.wikipedia.org/wiki/DNB_ASA. [Funnet 8 February 2023].
- [6] G. Williams, «How Safe Are Budget Tracking Apps and Software?,» U.S.News, 28 September 2022. [Internett]. Available: <https://money.usnews.com/money/personal-finance/articles/how-safe-are-budget-tracking-apps>. [Funnet 15 February 2023].

Bee on Budget	Version: 1.17
Vision	Date: 09/03/23

[7] Looka, « Design your own beautiful brand,» [Internet]. Available: <https://looka.com/>. [Funnet 27 February 2023].