|  |
| --- |
|  |
| BTWBeheer  Project plan – Individual project |
| |  |  |  | | --- | --- | --- | | Nies,Midas M.F.A. | Demand based - Semester 3 | Software engineering HBO ICT Fontys | |

Contents

[About this document 2](#_Toc145064797)

[Project overview 2](#_Toc145064798)

[Project Objectives 2](#_Toc145064799)

[Target audience 2](#_Toc145064800)

[Project Scope 3](#_Toc145064801)

[Moscow 3](#_Toc145064802)

[Must-Have 3](#_Toc145064803)

[Should-Have 3](#_Toc145064804)

[Could-Have 3](#_Toc145064805)

[Won’t-Have 3](#_Toc145064806)

[Project deliverables 4](#_Toc145064807)

[Project planning 4](#_Toc145064808)

[Introduction week 4](#_Toc145064809)

[Sprint 1 Initialization and Setup 4](#_Toc145064810)

[Sprint 2 Core Functionality 4](#_Toc145064811)

[Sprint 3 Core Functionality 5](#_Toc145064812)

[Sprint 4 Enhancements and Refinements 5](#_Toc145064813)

[Sprint 5 Testing, Optimization and Deployment 5](#_Toc145064814)

[Closing week 5](#_Toc145064815)

[Visual planning 6](#_Toc145064816)

# About this document

The project plan is a strategic document that provides a concise overview of the "BTWBeheer" bookkeeping web application project. It outlines project objectives, scope, timeline, and resource allocation while addressing key aspects such as risk management, quality assurance, and security. Serving as a roadmap for the project team and stakeholders, this document ensures clear communication, roles, and responsibilities, guiding the project towards its goals while allowing for adaptability to changing circumstances.

# Project overview

BTWBeheer is a web-based bookkeeping application designed for small entrepreneurs and businesses to manage their finances efficiently. Inspired by the success of digiBTW, our goal is to create a user-friendly and feature-rich platform that simplifies bookkeeping tasks, helps users stay tax-compliant, and provides valuable insights into their financial health.

# Project Objectives

The project's primary objective is to develop the BTWBeheer Bookkeeping Web Application, a user-focused financial management tool that competes effectively with existing platforms like DigiBTW. While recognizing that time constraints may limit the breadth of functionalities, the project will prioritize achieving excellence in every included feature. This entails meticulous attention to detail and ensuring that the implemented functionalities are flawless and user-friendly. By prioritizing quality over quantity, the project aims to deliver a streamlined, secure, and highly functional bookkeeping solution that can hold its own in the competitive landscape of financial management applications for small businesses and entrepreneurs.

# Target audience

BTWBeheer caters to small entrepreneurs and business owners by providing a streamlined and user-friendly financial management solution. Designed to address the unique needs of small enterprises, the application offers essential bookkeeping features without overwhelming complexity, empowering users to efficiently manage their finances and drive business growth.

# Project Scope

The scope of the **BTWBeheer** Bookkeeping Web Application project encompasses the development of a comprehensive financial management platform tailored to the needs of small entrepreneurs and small businesses. This includes the creation of a user-friendly interface, transaction management features, expense tracking capabilities, income recording tools, automated tax calculations, and reporting functionalities. The project also involves user authentication and data security measures to protect sensitive financial information. The application will be designed for scalability and efficient database management to handle growing user bases. In line with the **MoSCoW** (Must-Have, Should-Have, Could-Have, Won't-Have) prioritization, the project will primarily focus on implementing the "Must-Have" features while considering "Should-Have" and "Could-Have" features for future enhancements, and consciously excluding "Won't-Have" elements to ensure project feasibility and delivery within scope.

## User stories

### Must-Have:

As a small business owner, I want to be able to easily enter and track my revenue and expenses in the BTWBeheer application so that I can maintain an accurate record of my financial transactions.

As a user, I want to keep track of my profit within the BTWBeheer application to understand my business's financial performance over time.

As a small business owner, I want the BTWBeheer application to calculate and display my VAT (Value Added Tax) obligations based on my transactions so that I can ensure tax compliance.

As a user, I want to maintain a list of my business relations (e.g., clients, suppliers) in the BTWBeheer application to easily reference their contact information and transaction history.

As a small business owner, I need the ability to generate invoices within the BTWBeheer application, including itemized details, so that I can bill my clients accurately.

As a user, I want to create and send quotations through the BTWBeheer application to potential clients, including details of the proposed services and costs.

As a small business owner, I need to track the time I spend on various tasks or projects within the BTWBeheer application, allowing for hourly rate calculations and accurate billing.

As a user, I want to receive reminders for VAT declaration deadlines within the BTWBeheer application to avoid missing important tax filing dates.

### Should-Have:

As a user, I would like the BTWBeheer application to automatically send payment reminders to clients for outstanding invoices to help improve cash flow.

As a small business owner, I want the option to track my rides for business purposes within the BTWBeheer application, including mileage and expenses related to transportation.

As a user, I would like to register projects within the BTWBeheer application, associating them with clients and tracking hours worked for more accurate billing.

### Could-Have:

As a small business owner, I would appreciate the ability to manage my bank and cash transactions within the BTWBeheer application, enabling comprehensive financial tracking.

As a user, I would find it helpful if the BTWBeheer application included product and stock management features to monitor inventory levels and sales.

## Moscow

### Must-Have

* Keeping track of Revenue, payments / expenses, profit, and VAT
* Keeping track of relations
* Invoice generation
* Quotation generation
* Time registration
* VAT declaration reminders

### Should-Have

* Automatic invoice payment reminders
* Ride registration
* Project registration for easier billing (hourly rate per project)

### Could-Have

* Bank / cash registration
* Product & stock management

### Won’t-Have

* Multi employee registration, for big companies
* Automatic VAT declaration with tax authorities
* Personal bookkeeping (employment salaries)

# Project deliverables

The project deliverables for the final portfolio will include a ZIP file containing all products delivered throughout the group project and individual project, such as research papers, zipped source code (excluding binaries), configurations, test reports, design documents, and presentations. Additionally, there will be a comprehensive reader's guide in PDF format, offering descriptions and explanations for each item within the portfolio. This guide will provide general information about each product and detail how they contributed to achieving the learning outcomes, as perceived by both the student and teachers. The final chapter of the guide will present reflective insights on the entire process and the resulting products. Ultimately, the end grade for the semester will be based on the combined results of the group project and the individual project. The portfolio will be updated iteratively, ensuring that progress and structure are continually discussed and refined in collaboration with the semester coach and teachers.

# Project planning

## Introduction week

**Week 1** 04/09 – 10/09

In this week I will focus on planning and deciding on what my project will be. Also, I will do research to find out what development stack I will be using.

## Sprint 1 Initialization and Setup

**Week 2** 11/09 – 17/09 **Week 3** 18/09 – 24/09 **Week 4** 25/09 – 01/10

* Define project scope and objectives.
* Set up stack (Node.js, Express.js, React, MySQL).
* Create initial database schema.
* Set up version control and project structure.

## Sprint 2 Core Functionality

**Week 5** 02/10 – 08/10 **Week 6** 09/10 – 15/10 **Vacation** 16/10 – 22/10  
**Week 7** 23/10 – 29/10

* Implement user authentication.
* Build basic UI components.
* Create API routes and endpoints.
* Develop core features.

## Sprint 3 Core Functionality

**Week 8**  30/10 – 05/11  
**Week 9** 06/11 – 12/11  
**Week 10** 13/11 – 19/11

* Implement user authentication.
* Build basic UI components.
* Create API routes and endpoints.
* Develop core features.

## Sprint 4 Enhancements and Refinements

**Week 11** 20/11 – 26/11  
**Week 12** 27/11 – 03/12  
**Week 13** 04/12 – 10/12

* Add additional features and functionality.
* Improve user interface and user experience.
* Refactor code for better organization.
* Begin testing and bug fixes.

## Sprint 5 Testing, Optimization and Deployment

**Week 14** 11/12 – 17/12  
**Week 15** 18/12 – 24/12  
**Vacation** 25/12 – 31/12  
**Vacation** 01/01 – 07/01  
**Week 16** 08/01 – 14/01

* Extensive testing of application components.
* Optimize performance.
* Finalize user interface refinements.
* Address and fix any issues found during testing.
* Prepare the application for deployment.
* Deploy the application to a hosting environment.
* Create user documentation or guides.
* Perform final testing and debugging.
* Prepare for project presentation and handoff.

## Closing week

**Week 17** 15/01 – 21/01

This week we need to hand in the final product. Which is due on Friday, January the 19th.

## Visual planning

A calendar with different colors and numbers

Description automatically generated

# Technology stack

## Front-End

* React

## Back-End

* Node.js
* Express

## Database

* MySQL

## ORM (Object-Relational Mapping):

* Sequelize

## Authentication and Authorization

* Passport.js

## Deployment

* Hosting on cloud, if possible, on Fontys servers.
* Docker for containerization (optional).