

---

# “ReVend” - Online Marketplace

## Project Plan

---

**Date:** 20.05.2024 (*last modified*)

**Author:** *Desislav Hristov*

**Version:** 4.0

---



## Contents

<b>Introduction</b> .....	3
<b>1. Project Goal</b> .....	3
<b>2. Project Scope</b> .....	4
2.1 Preconditions.....	4
<b>3. Client &amp; Team</b> .....	4
<b>4. Strategy</b> .....	5
<b>5. Research questions and methodology</b> .....	5
5.1 How can user experience be optimized to encourage the adoption of sustainable shopping practices? .....	5
5.2 What are the critical security measures needed to protect user data and transactions on the platform?.....	5
<b>6. Deliverables</b> .....	6
6.1 Documentation .....	6
6.2 Front-End Application .....	6
6.3 Back-End Infrastructure.....	6
6.4 Database Schema.....	6
6.5 Security Framework .....	6
<b>7. Non-Deliverables</b> .....	6
<b>8. Constraints</b> .....	7
8.1 Constraints Imposed by Semester Requirements.....	7
8.2 Self-Imposed Constraints .....	7
<b>9. Phasing Based on Sprints</b> .....	8
Sprint 1: Project Setup and Initial Framework .....	8
Sprint 2: Architecture Design and Integration.....	9
Sprint 3: Backend Integration and Quality Assurance .....	9
Sprint 4: User Experience and Security Enhancements .....	9
Sprint 5: Finalization and Security .....	9

## Introduction

In today's fast-paced consumer market, a significant challenge is the efficient circulation of goods, particularly when items outlive their initial purpose or fall out of personal favor. Many usable goods are prematurely discarded, contributing to clutter and waste, while others seek these very items in secondary markets. The gap between potential buyers and sellers remains wide due to a lack of a streamlined, trustworthy platform that facilitates easy exchange. By providing a streamlined and trustworthy platform, [ReVend Marketplace](#) seeks to facilitate the easy exchange of such items.

### 1. Project Goal

The goal of the ReVend Marketplace project is to create an online platform that facilitates the easy exchange of pre-owned goods among users, thereby extending the lifecycle of products and reducing waste. The project seeks to:

- [Why](#): Address the environmental impact of consumer goods by promoting reuse and recycling.
- [Preferred Situation](#): Establish a user-friendly, secure, and efficient platform where individuals can list items for sale, purchase pre-owned goods, and participate in an item exchange program.
- [Advantages](#): Reduced environmental footprint, cost savings for users, and increased awareness of sustainable shopping practices.
- [Value Addition](#): Enhances the company's commitment to sustainability, provides a community for like-minded individuals, and opens revenue channels through marketplace transactions.
- [ICT Product Possibilities](#): Implementation of features such as user profiles, secure payment systems, product listings, and a review system to ensure trust and reliability.

## 2. Project Scope

The task at hand is to create an application of our choice for an individual assignment that will be graded by the assessors. Key focus areas include:

- **Facilitates Easy Navigation:** Ensure users can effortlessly find what they're looking for, with clear categorization, a straightforward layout, and an intuitive search function.
- **Supports Efficient Listing and Management:** Provide sellers with easy-to-use tools for listing their items, including uploading photos, setting prices, and managing their listings with real-time updates.
- **Promotes User Engagement:** Create interactive elements such as ratings and reviews to keep users engaged and invested in the platform.
- **Incorporates Real-Time Notifications:** Utilize real-time notifications to keep users informed about important updates, messages, or transaction statuses, enhancing the user experience.

### 2.1 Preconditions

The platform will be developed using **React** for the frontend and **Java Spring Boot** for the backend, with adherence to responsive design principles to ensure accessibility across devices. Existing technology choices, including the use of a relational database and **RESTful API** design, have been predetermined.

## 3. Client & Team

Team:

- Hristov, Desislav - email: [457570@student.fontys.nl](mailto:457570@student.fontys.nl)

Assessors:

- Coenen, Frank F.W.J.

- Schriek, Erik H.J.D. van der
- Oliveira De Arruda Camara, Amália M.A.

## 4. Strategy

The project will adopt an [Agile](#) development approach, specifically [Scrum](#), due to its flexibility, emphasis on customer feedback, and ability to adapt to changing requirements. This iterative methodology will facilitate regular evaluation of progress and ensure alignment with user needs and business goals.

## 5. Research questions and methodology

### [5.1 How can user experience be optimized to encourage the adoption of sustainable shopping practices?](#)

- Approach: Conduct user surveys and A/B testing to gather feedback on the platform's usability and features.
- Methodology: Workshop and Lab contexts for iterative design and testing.

### [5.2 What are the critical security measures needed to protect user data and transactions on the platform?](#)

- Approach: Review of current best practices in e-commerce security.
- Methodology: Desk research and Expert Interview.

**As the project progresses, additional research questions may emerge, necessitating updates to the project deliverables and approach.**

## 6. Deliverables

### 6.1 Documentation

- Platform Design Documents
- Project Plan

### 6.2 Front-End Application

- A fully functional React-based web application featuring user-friendly navigation, interactive elements, and a visually appealing interface.

### 6.3 Back-End Infrastructure

- A robust Java Spring Boot RESTful API that handles all server-side logic, database interactions, and client-server communications securely and efficiently.
- Implementation of business logic.

### 6.4 Database Schema

- A well-structured MySQL database schema, optimized for performance and scalability, including all necessary tables, relationships, and indices.
- Integration of JPA/Hibernate for object-relational mapping to facilitate smooth and efficient data operations.

### 6.5 Security Framework

- Implementation of a comprehensive security framework, including secure user authentication and authorization processes using JWT for role-based access control.
- Protection against common security threats like SQL injection.

## 7. Non-Deliverables

### What will not be delivered:

- Integration with third-party job portals.
- Maintenance of the website after delivery.

- User manual for the website.

## **8. Constraints**

### **8.1 Constraints Imposed by Semester Requirements**

**These constraints are defined by the academic framework, deadlines, and deliverables set by the educational institution for the semester.**

- **Time:** Fixed deadlines for project milestones and the final delivery date can limit the scope of work, necessitating prioritization of core features over nice-to-have additions.
- **Technical:** The choice of technology stack (React, Java Spring Boot, MySQL, etc.) may limit certain functionalities or integrations, especially if relying on legacy systems or specific third-party APIs.
- **Resource:** The project must be completed with the available resources within the academic setting, including access to software, hardware, and any institutional support services.
- **Regulatory and Compliance:** Legal and regulatory requirements, especially concerning user data protection (like GDPR), payment processing, and online commerce, can restrict certain features or necessitate additional measures for compliance.
- **Security:** The need to ensure high levels of security, particularly for user data and transactions, can impact development timelines and require additional resources for implementing robust security measures.

### **8.2 Self-Imposed Constraints**

**These constraints are the additional limitations or challenges we've identified based on our project vision, personal goals, or perceived project needs.**

- **Operational:** The operational environment, including hosting servers, development and production environments, and deployment pipelines, can impose limitations on scalability and performance.

- **User Adoption:** The ability to attract and retain users, influenced by market competition, user preferences, and platform usability, can be a significant constraint on the success of the marketplace.
- **Integration:** Challenges in integrating with external systems, payment gateways, or other third-party services due to API limitations, costs, or technical incompatibilities.
- **Scalability:** Potential limitations in the platform's ability to scale efficiently with increased user load or data volume, influenced by the initial architectural and infrastructure decisions.

## 9. Phasing Based on Sprints

### Introduction to Phasing

In alignment with the Agile development approach, the ReVend Marketplace project will be executed in phases, divided into multiple sprints. Each sprint is planned to last two weeks and aims to deliver specific, incremental functionalities of the platform. This phasing ensures continuous progress, frequent assessments, and the ability to incorporate feedback rapidly.

### Sprint Breakdown

Here's a tentative breakdown of the sprints and their focal points. It's important to note that the nature of Agile allows for flexibility, and sprint goals may be adjusted based on ongoing feedback and project needs.

### Sprint 1: Project Setup and Initial Framework

- **Goals:** Establish the project's foundational elements, including the setup of development environments, initial architectural design, and basic project scaffolding.
- **Deliverables:** Project plan, exclusive GIT repository for Backend, Backend (Layering, dependency inversion and injection using Spring), CI/CD environment initialization



## Sprint 2: Architecture Design and Integration

- **Goals:** Refine architecture, establish design decisions, and integrate frontend with backend.
- **Deliverables:** Continuous Delivery: Dockerfiles for backend and frontend, Initial frontend setup and exclusive Git repository, Backend: CORS configuration, Design document version 1, C4 Model diagrams (Levels 1, 2, 3)

## Sprint 3: Backend Integration and Quality Assurance

- **Goals:** Enhance backend functionality by integrating with the database, ensure code quality and coverage with testing and analysis tools.
- **Deliverables:** Design document version 2, Initial backend to database setup, Unit testing, Database versioning with Flyway, SonarQube installed and running, SonarQube plugged into pipelines with a new stage after tests

## Sprint 4: User Experience and Security Enhancements

- **Goals:** Improve the user interface based on feedback, refine design documentation, and implement robust authentication and authorization mechanisms.
- **Deliverables:** UX feedback report and resulting UI improvements, Design document version 3 with CI setup diagram, Authentication and authorization implementation, Backend login service using JWT, Role-based authorization on two services, Frontend login feature connected to backend and token storage, one feature calling a secured backend service, Continuous integration with SonarQube (all tests passing, business layer > 80% unit test coverage)

## Sprint 5: Finalization and Security

- **Goals:** Finalize design documentation, enhance application security, implement WebSocket feature, and ensure the application meets MVP requirements.

- **Deliverables:** Final design document, Security report addressing OWASP top 10 risks, Code review for broken-access control and injection cases, WebSocket feature implementation, MVP features completed, Continuous integration with SonarQube (all tests passing, business layer > 80% unit test coverage)