Project Plan

TrendTrack Design Co.

Date	Sept 30, 2024
Author	Ivet Kalcheva
Status	Completed (Version 1.0)

Version History

Version	Date	Author(s)	Changes	State
0.1	03/09/24	Ivet Kalcheva	Initial draft, basic structure	Draft
0.3	15/09/24	Ivet Kalcheva	Added more detailed scope and phases	Draft
0.8	16/09/24	Ivet Kalcheva	Redefining	Under Review
1.0	30/09/24	Ivet Kalcheva	Finalized document	Completed

Distribution

Version	Date	Receivers
0.8	20/09/24	Frank Coenen and Bart Rabeling
1.0	11/10/24	Frank Coenen and Bart Rabeling

Contents

Project Assignment

Context

TrendTrack Design Co. is a fashion manufacturing company based in Eindhoven, Netherlands. Specializing in high-quality fabrics and trims, the company was ranked among the top 10 reliant distributors in Europe for 2019.

However, their outdated paper-based inventory system leads to inefficiencies for employees and limited product visibility for clients. Moreover, their lack of an online presence restricts their customer base to already established fashion brands, who also place orders exclusively by phone.

Goal of the project

The primary goal of this project is to create a web application that updates the company's inventory management system for employees and streamlines delivery, and order processes for clients.

Digitize Inventory Management	Enhance Product Visibility	Improve Client Experience

Scope and predictions

Inside Scope	Outside Scope
Design documents for the application	24/7 customer support
Implementation of user-friendly interface	Localisation for multiple languages
Real-time data and integration	Marketing and promotion
User roles and permissions	Integration with external company system
User-specific notifications	User manual for employees
Advanced filtering and search	Future system scalability
Sales Analytics	Mobile application development

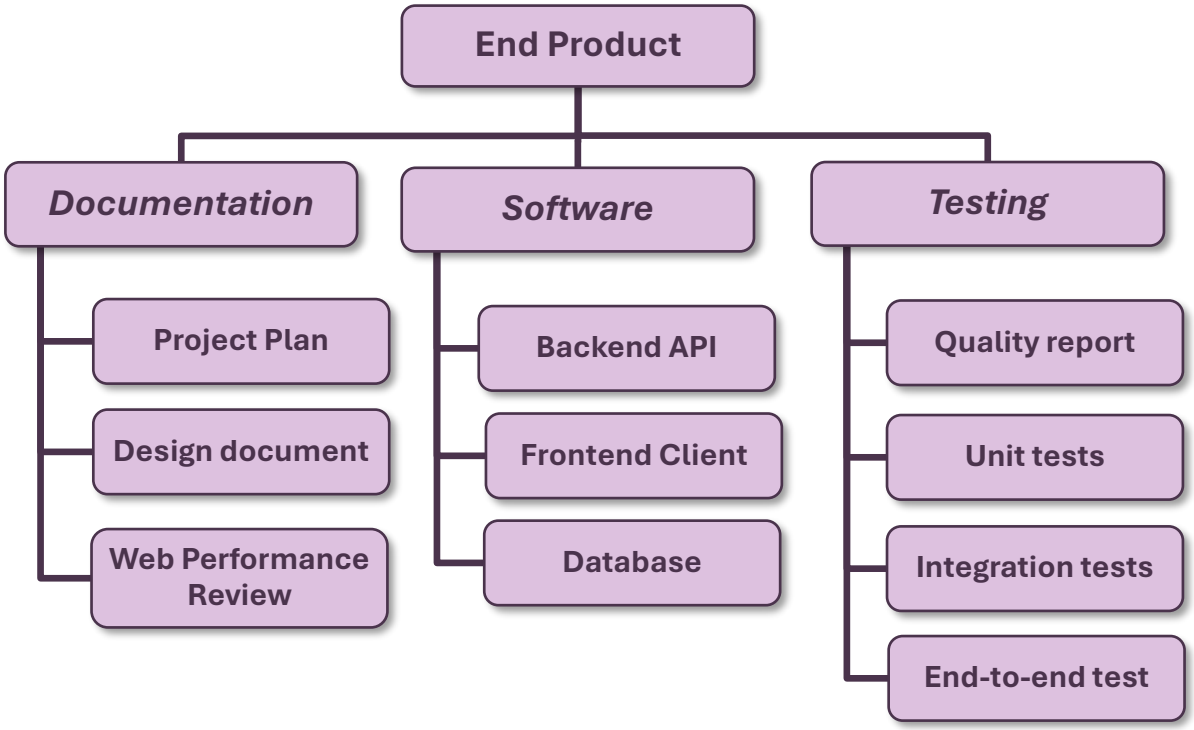
Strategy

The project will adopt an **Agile** approach using the **Scrum** framework. This method creates a flexible development process, allowing for continuous feedback, quick adjustments to changing requirements and a focus on delivering functional features in a structured manner.

End products

- Project Plan
- Design document
- Web Performance Review document
- REST API in Java Application
- REACT Front-End application
- Database system
- Quality report
- Unit tests
- Integration tests
- End-to-end test

Product Breakdown Structure (PBS)



Project Organisation

Stakeholders and team members

Company and project owner

- **Name:** Marian van den Scheer
- **Availability:** Through email (on weekly basis)
- **Email:** marian.vdscheer@trendtrack.nl

Lead developer

- **Name:** Ivet Kalcheva
- **Availability:** Monday to Friday
- **Email:** i.kalcheva@student.fontys.nl

Teacher

- **Name:** Bart Rabeling
- **Availability:** Mo 09:00-12:00, We 09:00-12:00
- **Email:** b.rabeling@fontys.nl

Teacher

- **Name:** Frank Coenen
- **Availability:** We 13:00-16:00, Fr 09:00-12:00
- **Email:** f.coenen@fontys.nl

Communication

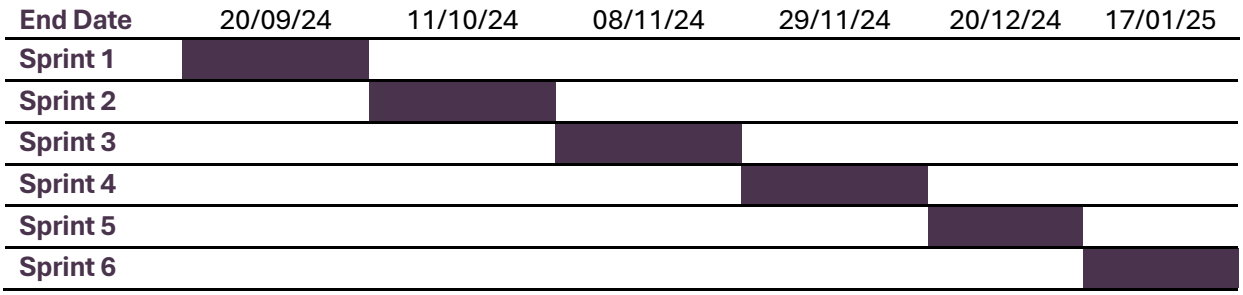
- **GitLab:** Used for version control, project management and collaboration.
- **Jira:** Responsible for task management, sprint planning and track of progress.
- **Sprint reviews:** Held to review progress (with teachers).
- **Weekly meetings:** Scheduled to align on project progress (with the client).
- **Weekly feedback:** Provided on progress to ensure flexibility (with teachers).

Activities and Time Plan

Phrases of the project

- **Phrase 1 (Requirements Gathering and Documentation):** Conduct stakeholder meetings to understand the company's needs, gather requirements, and define scopes and deliverables. Create user stories and design documents to outline system features.
- **Phrase 2 (Research and Fundamental Project Setup):** Research technologies and architectures suitable for the system. Set up development environments, frameworks, and establish CI/CD pipelines to ensure smooth development and integration processes.
- **Phrase 3 (Development of Core Modules):** Build the core functionality of the application, including user authentication, inventory management and product catalogue features. Implement user roles, permissions and basic filtering capabilities.
- **Phrase 4 (Implementation of Advanced Features and Customizations):** Develop advanced features like product filtering, user-specific notifications and sales analytics. This phase also includes refining the user interface and adding validation and error-handling mechanisms.
- **Phrase 5 (Testing, Feedback, Finalization and Documentation):** Perform comprehensive system testing (unit, integration and end-to-end tests) and collect feedback from stakeholders. Address any bugs or issues, finalize system features and ensure all documentation is completed. Prepare for the final sprint demo, handover and project wrap-up.

Time plan and milestones



- **Sprint 1:** Initialized project plan, environment setup, Git repository (with CI/CD pipelines) and initial system framework.
- **Sprint 2:** Frontend design and development, initial UI setup and integration with basic features.
- **Sprint 3:** Backend core functionality (user authentication and product management), database setup and unit tests.
- **Sprint 4:** Advanced features (notifications, filtering and sales analytics), role-based authorization and user interface refinement.
- **Sprint 5:** Finalization of remaining features, thorough system testing (end-to-end tests), bug fixing and quality assurance.
- **Sprint 6:** Final sprint demo, handover, deployment, performance review and complete project documentation.

Testing Strategy and Configuration Management

Testing strategy

- Unit test coverage > 80%
- Technical debts <5 days (subjective)
- Maintainability index for the module/code > 90
- Functional test automation > 75%
- Functional test passed
- No known defects

Test Environment and Required Resources

- Automated testing deployment using **CI/CD configuration** via GitLab.
- Test environment in **Docker**.

Configuration management

- **Tool:** GitLab
- **Strategy:** Branching Strategy (using Feature, Develop, Release and Main Branch)

Risks

Risk	Prevention activities	Mitigation activities	Probability
Data Loss	Regular backups using GitLab repositories.	Restore from latest GitLab backup.	Low
Integration Issues	Implement continuous integration (CI) and frequent testing.	Regular testing to identify and fix issues early.	Medium
Laptop Failure	Use GitLab to ensure continuous project updates.	Obtain a replacement laptop (from ISSD) and access from GitLab.	Low
Delays in Delivery	Break tasks into smaller parts.	Reprioritize tasks and focus on high-priority features.	Medium
Network Issues	Work offline using local environments.	Use a mobile hotspot or find a stable connection.	High

Appendix

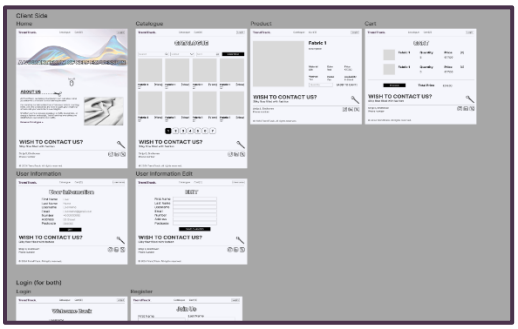


Figure 1: Figma Design

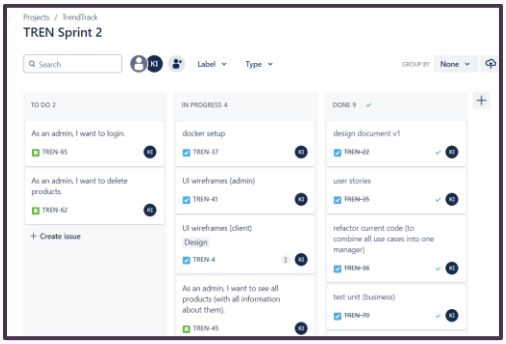


Figure 2: Jira Board

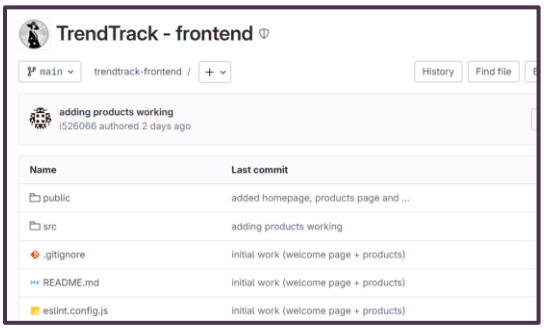


Figure 3: GitLab Repository - Frontend

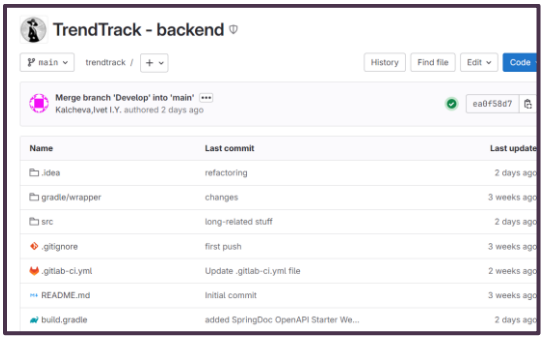


Figure 4: GitLab Repository - Backend