Karoliina Ingman, Eetu Oinonen, Oleg Ivantsov, Miika Auvinen

Inventory Management System Project Plan

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

24.1.2025

Content

1 Project Overview

- **Project Title**: Inventory Management System
- **Problem Summary**:
    - Inefficient inventory management can lead to challenges such as inaccurate stock levels, overstocking, stock-outs, and difficulty managing customers and suppliers. These issues may arise from poor tracking systems, lack of organization, or ineffective tools, causing operational disruptions and revenue losses.
    - This project aims to solve these problems by developing an intuitive and user-friendly inventory management system.
- **Intended Audience/Users**:
    - Business administrators responsible for inventory and supply chain management.
    - Small and medium-sized enterprises (SMEs) require intuitive and reliable inventory management tools.
- **Main Features/Components**:
    - User Authentication
    - Product Management
    - Supplier and Customer Management
    - Stock Transactions Management
    - Reports and Analytics

2 Project Objectives

1. **Enhance Security Through User Authentication**

- **Specific***:* The objective focuses on implementing secure login and encrypted password storage.
- ***Measurable****:* Success is measured by having functional registration, login, and encryption of all passwords.
- ***Achievable****:* It uses well-established protocols (e.g., bcrypt).
- ***Relevant****:* Security is a key requirement for inventory systems to protect sensitive data.
- ***Time-bound****:* Delivered by **Sprint 1**.

2. **Streamline Product Inventory Management**

- ***Specific****:* This includes tools for adding, editing, deleting, and bulk uploading product data.
- ***Measurable****:* The module must handle at least 30 product entries and bulk uploads via CSV/Excel.
- ***Achievable****:* Using databases and libraries for bulk uploads ensures feasibility.
- ***Relevant****:* Accurate and easy product management is crucial for inventory tracking.
- ***Time-bound****:* Delivered by **Sprint 2**.

3. **Optimize Stock Tracking with Transaction Management**

- ***Specific****:* Focuses on recording stock in/out movements and linking them to suppliers or customers.
- ***Measurable****:* System must support tracking of at least 500 transactions and include filters for date, product, and type.
- ***Achievable****:* Database transaction systems can handle these requirements.
- ***Relevant****:* Accurate stock tracking is essential for business operations.
- ***Time-bound****:* Delivered by **Sprint 3**.

4. **Streamline Customer and Supplier Profiles Management**

- ***Specific:*** Develop features to add, edit, and delete customer and supplier profiles, with the ability to link stock transactions to the respective profiles.
- ***Measurable:*** Ensure the system can handle at least 500 customer profiles and 200 supplier profiles while maintaining accurate linkage to transactions.
- ***Achievable:*** Use a relational database to store and organize profiles with support for search and filtering.
- ***Relevant:*** Proper management of customers and suppliers improves operational efficiency and accountability in stock movement.
- ***Time-bound:*** Delivered by **Sprint 3**.

5. **Empower Data-Driven Decisions with Analytics and Reporting**

- ***Specific****:* Tools will generate/export reports on stock trends, value, and transaction history.
- ***Measurable****:* Reports must generate within 5 seconds for up to 1 year of data.
- ***Achievable****:* Leveraging libraries for PDF/Excel exports makes this feasible.
- ***Relevant****:* Decision-making relies on accurate analytics and reports.
- ***Time-bound****:* Delivered by **Sprint 4**.

6. **Provide a Functional Prototype for Future Development**

- ***Specific****:* Combines all features into a functional prototype
- ***Measurable****:* All features should be tested and work before delivery.
- ***Achievable****:* Completing work in structured sprints makes this realistic.
- ***Relevant****:* A functional prototype is essential for the project's success and smooth transition for the next project course.
- ***Time-bound****:* Delivered by **the end of the Sprint 4**.

3 Scope and Deliverables

Included in Scope:

- User authentication with secure login and encrypted password storage.
- Features for adding, editing, deleting, and bulk uploading products.
- Supplier and customer management tools.
- Recording and tracking of stock in/out transactions.
- Reporting and analytics tools for transaction history and stock trends.
- An intuitive user interface tailored for ease of use.

Excluded from Scope:

- Advanced features like AI-based demand forecasting.
- Integration with third-party e-commerce or ERP systems.
- Real-time updates using barcode scanners.

Deliverables:

- **User Authentication Module:**
  - A secure login and registration system with encrypted password storage.
- **Product Management Module:**
  - Tools for adding, editing, deleting, and bulk uploading product details.
- **Supplier and Customer Modules:**
  - Features for managing supplier and customer profiles and linking them to transactions.
- **Stock Transactions Module:**
  - Functionality for recording and tracking stock in and out movements.
- **Reports and Analytics Module:**
  - Tools for generating transaction reports with filtering options and exporting them to PDF, CSV and/or Excel.
- **User-Friendly Interface:**

    - A clear and accessible UI tailored for all users, including older individuals.

4 Project Timeline

Phase 1: Requirement Gathering and Planning

- **Start Date:** January 13, 2025
- **End Date:** January 24, 2025
- **Milestones:**
    - Finalize project scope and objectives.
    - Gather and document user requirements from the user stories.

Phase 2: Design

- **Start Date:** January 23, 2025
- **End Date:** February 2, 2025
- **Milestones:**
    - Create UI wireframes and database schema.
    - Prepare architectural diagrams (three-tier structure).

Phase 3: Development (Sprints)

- **Sprint 1:**
    - **Start Date:** January 13, 2025
    - **End Date:** January 27, 2025
    - **Milestones:** Implement User Authentication and Registration. User schema. Login / Register GUI.
- **Sprint 2:**
    - **Start Date:** January 27, 2025
    - **End Date:** February 10, 2025
    - **Milestones:** Develop Product Inventory Management and Management of Suppliers features.
- **Sprint 3:**
    - **Start Date:** February 10, 2025
    - **End Date:** March 3, 2025

    - **Milestones:** Implement Stock In/Out Transactions and Management of Customers.
- **Sprint 4:**
    - **Start Date:** March 3, 2025
    - **End Date:** March 14, 2025
    - **Milestones:** Add Reports and Analytics functionalities.

Phase 4: Testing and Debugging

- **Start Date:** January 23, 2025 (~ from Development start date, Continuous process)
- **End Date:** March 14, 2025
- **Milestones:**
    - Conduct unit testing, integration testing, regression testing and user acceptance testing.
    - Fix identified bugs.

Phase 5: Documentation

- **Start Date:** March 3, 2025
- **End Date:** March 14, 2025
- **Milestones:**
    - Deliver user manuals and final reports.

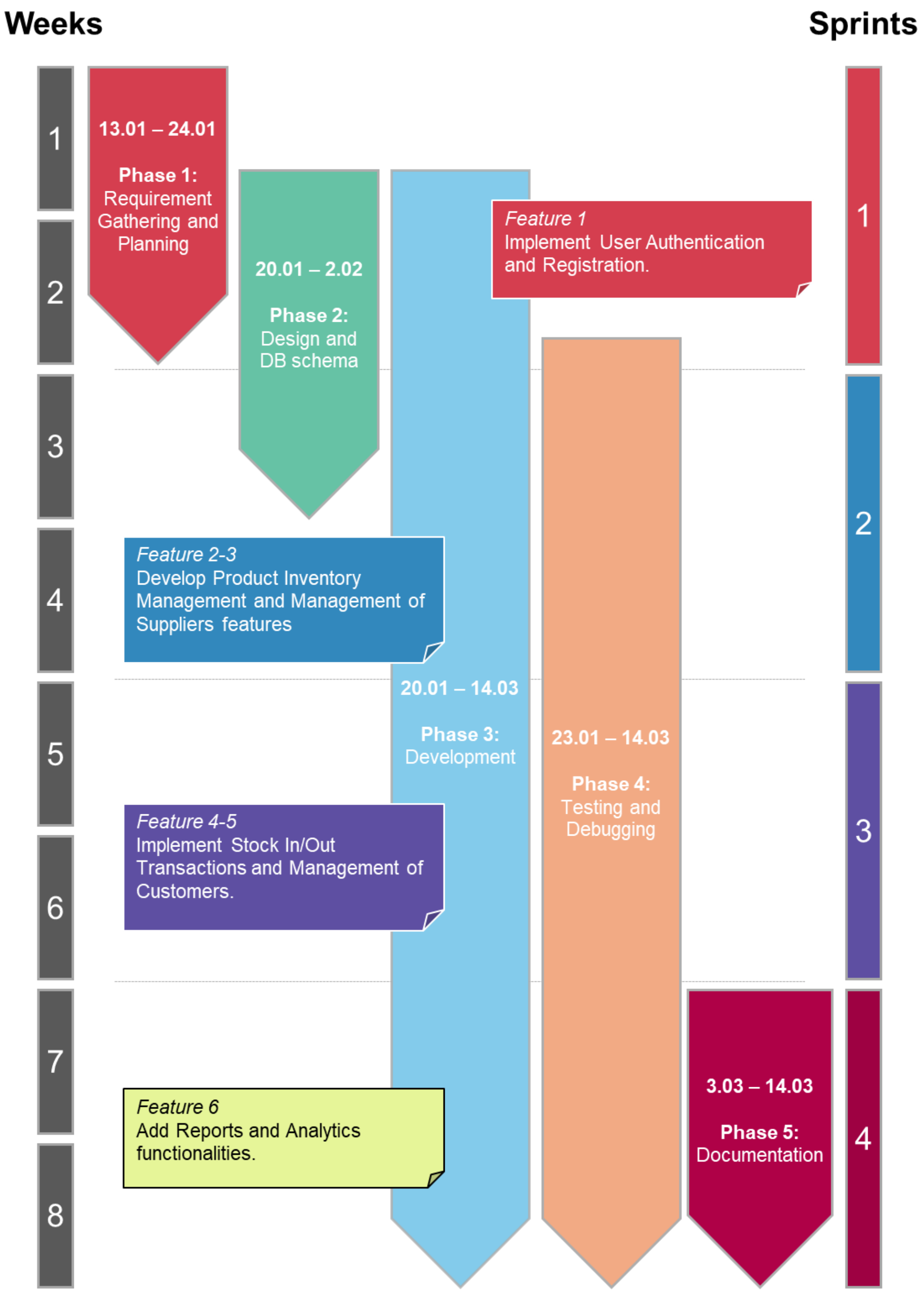
Weeks
Sprints
1
2
3
4
5
6
7
8
13.01 – 24.01
Phase 1: Requirement Gathering and Planning
20.01 – 2.02
Phase 2: Design and DB schema
20.01 – 14.03
Phase 3: Development
23.01 – 14.03
Phase 4: Testing and Debugging
3.03 – 14.03
Phase 5: Documentation
Feature 1
Implement User Authentication and Registration.
Feature 2-3
Develop Product Inventory Management and Management of Suppliers features
Feature 4-5
Implement Stock In/Out Transactions and Management of Customers.
Feature 6
Add Reports and Analytics functionalities.
1
2
3
4

Chart 1. Project Timeline

5 Work Breakdown

The table below shows the work breakdown in general terms, the specific distribution is tracked by using a monitoring tool (e.g. Trello) and is determined as the project progresses.

Table 1. Work Breakdown

Task	Description	Dependencies	Assigned To
Requirement Gathering	Collect and document user stories.	None	Entire Team
UI Design	Design wireframes for the UI components.	Requirement Gathering	UI Designer
Database Design	Design schema for MariaDB.	Requirement Gathering	Database Specialist
User Authentication Module	Develop login and registration features.	UI and Database Design	BE and FE Developers
Product Inventory Module	Implement CRUD operations for inventory management.	Database and UI Design	BE and FE Developers
Supplier Management	Implement supplier-related CRUD functionalities.	Database and UI Design	BE and FE Developers
Customer Management	Implement customer-related CRUD functionalities.	Database and UI Design	BE and FE Developers
Stock In/Out Transactions	Develop functionality for tracking stock movement.	Product Inventory Module, Supplier and Customer Management	BE and FE Developers
Report Generation Module	Generate and (export) reports for stock and transactions.	Stock Transactions Module	BE and FE Developers
Testing and Bug Fixing	Conduct end-to-end testing and fix bugs.	Completion of each feature	Entire Team
Documentation	Create user and technical manuals.	System Implementation	Entire Team

6 Resource Allocation

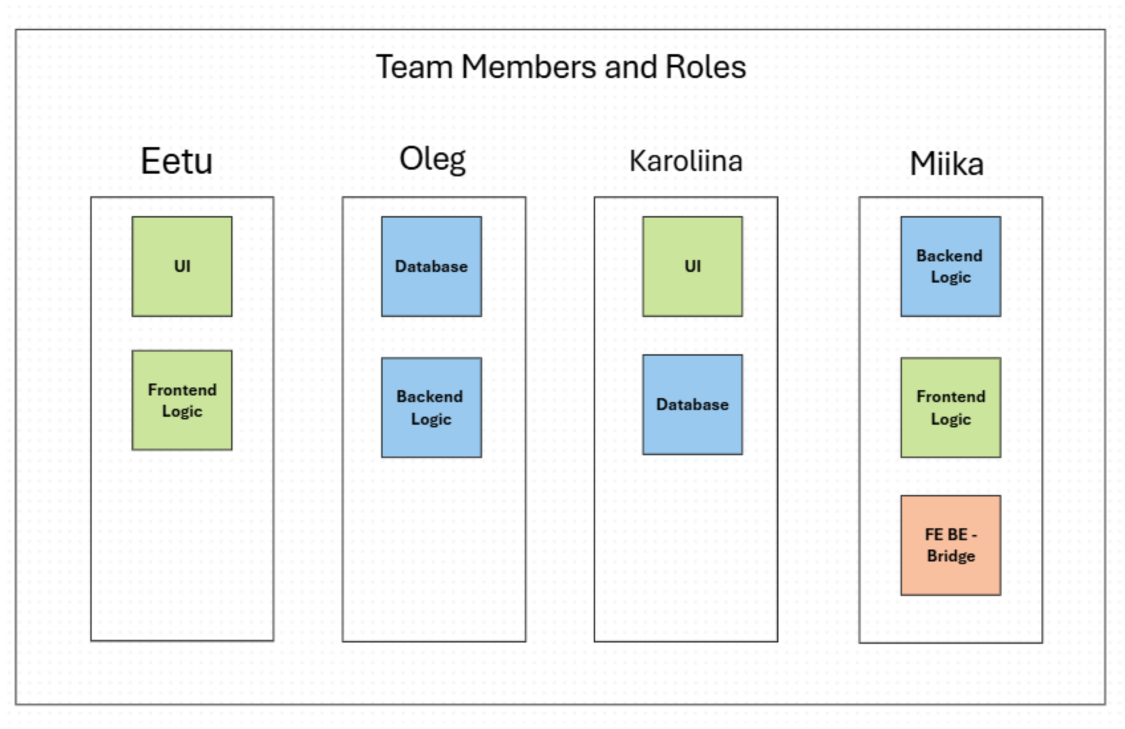

Image 1. Team Members and Roles

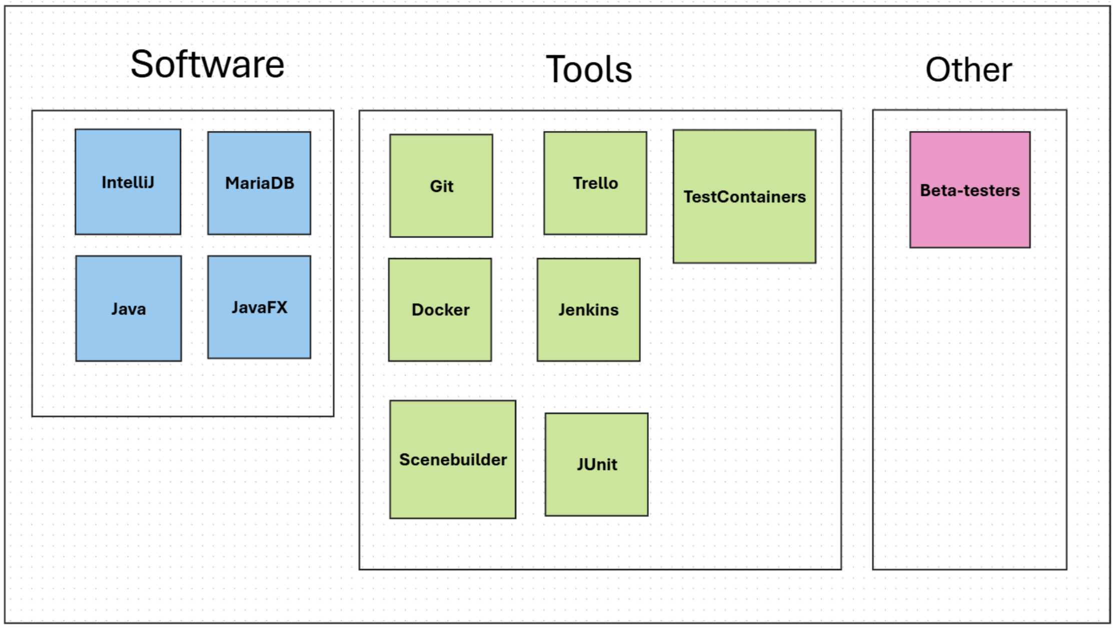

Image 2. Software, Tools

7 Risk Management

Risk 1: Communication Problems

- **Description:** Misunderstandings regarding responsibilities, roles, or project progress, especially since the team is inexperienced in software development. This can lead to duplicate efforts and delays.
- **Likelihood: Medium**
- **Impact: Medium to High (3–4 on a scale of 1–5)**
- **Mitigation Strategies:**
  - Actively use project management tools (e.g. Trello) to clarify responsibilities and track progress.
  - Conduct daily scrum meetings to ensure alignment.
  - Encourage team members to inform others about upcoming changes proactively.

Risk 2: Issues Integrating Frontend, Backend, and Database

- **Description:** Integration problems between the frontend, backend, and database could result in delays and disrupt development of other features.
- **Likelihood: Medium**
- **Impact: Medium (3 on a scale of 1–5)**
- **Mitigation Strategies:**
  - Set up a CI pipeline to automatically test the integration of the FE, BE, and database
  - Hold regular meetings or stand-ups between frontend, backend, and database teams to address integration challenges promptly.

Risk 3: Underestimating Development Time for Complex Features

- **Description:** Miscalculations of the time needed for complex features can lead to missed deadlines, resource reallocation, or cutting planned features.
- **Likelihood: High**

- **Impact: High (4 on a scale of 1–5)**
- **Mitigation Strategies:**
  - Reserve extra time in the schedule for buffer periods.
  - Break down complex tasks into smaller, more manageable tasks to improve accuracy in time estimates.
  - Focus on prioritizing and delivering the most crucial features first.

8 Testing and Quality Assurance

- **Unit testing:** JUnit, testing methods and objects individually to ensure they perform as intended. ***Success criteria***: 1) All methods and functions achieve at least 70% code coverage. 2) Tests pass for both valid inputs (positive testing) and invalid inputs (negative testing).

- **Integration testing:** JUnit and Docker, usage of containers to encapsulate code to be tested. Doing this will ensure that no external factors affect the outcome while testing parts of the system together. ***Success criteria***: modules can exchange data correctly without errors (e.g., frontend calls backend successfully, and backend stores data in the database).

- **Regression testing:** Docker and Jenkins, testing code in isolated testing environment whenever new code is pushed to Git. This will notify if code that previously worked, broke after new code was pushed to Git. ***Success criteria***: 1) Automated test suites detect and report regressions immediately after new code is pushed. 2) Deployment pipelines in Jenkins run successfully with no build failures.

- **User acceptance testing:** Beta-testers test the application and analyze if it is logical and easy to use. ***Success criteria***: 1) 80–90% positive feedback from beta-testers. 2) All core functionalities work without defects.

9 Monitoring and Reporting Mechanisms

Progress Tracking:

- Use a Kanban board (e.g., Trello) to track task status in real-time.
- Sprint reviews to assess completion of user stories.
- Weekly progress reports to monitor timeline adherence.

Reporting:

- Sprint meetings (e.g. Sprint Reviews) to discuss progress, risks, and blockers.
- Daily stand-ups for team updates during development (Discord).
- Final project presentation to stakeholders (e.g. teacher) at the end of each phase.

10 Documentation

Planned Documentation:

- **Requirement Specification:** Includes detailed user stories.
- **Technical Documentation:** Architecture, database schema, 3-tier application schema.
- **User Manual:** Instructions for using the system.
- **Test Reports:** Results of system, integration, and unit testing.
- **Final Project Report:** Summary of work completed, and lessons learned.