

Software Project Plan

Beren Elçin Polat - 22070006064

Melike Aytaç - 21070006056

Yağmur Sabırlı - 21070006017

Sultan Esen Murat - 21070006039

Table of Contents

Overview 4

High-Level Functionality 5

Project Staffing 5

3.1 Assignment Staff 5

3.2 Requirements 6

3.3 Design 6

3.4 Iteration 1 6

3.5 Iteration 2 7

3.5.1 Interface Coding Group 7

3.5.2 Persistent Storage Group 7

3.6 Iteration 3 7

3.7 Optional Iteration 4 8

3.8 Testers 8

Software Process 8

4.1 Iteration 1 9

4.2 Iteration 2 9

4.3 Iteration 3 9

Software Engineering Methods 10

5.1 Coding Standard Verification 10

5.2 Code Reviews 10

5.3 Coding Standard 10

5.4 Object-Oriented Design 10

Schedule and Effort 11

6.1 Schedule 11

6.1.1 Iteration 1 11

6.1.2 Iteration 2 11

6.1.3 Iteration 3 11

Measurements 11

Project Risks 12

Software Tools 13

- 9.1 **Tools** 13
 - 9.1.1 **Git** 13
 - 9.1.2 **GitHub** 13
 - 9.1.3 **OpenJDK** 13
 - 9.1.4 **IntelliJ IDEA / VSCode** 13
 - 9.1.5 **Java GUI Toolkit** 13
 - 9.1.6 **Microsoft Teams** 13
 - 9.1.7 **Jira** 14

Hardware Support 14

- 10.1 **Development Machine** 14
- 10.2 **Testing Machine** 14
- 10.3 **CI/CD Build Server** 14
- 10.4 **Presentation/Demo Machine** 15

Software Support 15

- 11.1 **Git** 15
- 11.2 **GitHub** 15
- 11.3 **IntelliJ IDEA / VSCode** 15
- 11.4 **Java Development Kit (OpenJDK 17)** 15
- 11.5 **Microsoft Teams** 15
- 11.6 **Jira** 15

Personnel Support 16

1 Overview

This document contains the Software Project Plan for the Restaurant Automation system that will be produced spring semester. The schedule for this project starts in March 2025 and ends in June 2025. The members of this team are as follows:

Beren Elçin Polat

Sultan Esen Murat

Melike Aytaç

Yağmur Sabırlı

The Restaurant Automation System is a comprehensive software solution designed to streamline restaurant operations, with a focus on order management and inventory tracking. Tailored for small to medium-sized establishments, this system helps restaurant managers and owners efficiently handle customer orders, monitor stock levels, and optimize daily workflows. By automating routine tasks, it reduces administrative burden and enhances overall service quality.

The project scope for the Restaurant Automation system product will be to include a Java GUI based interface, and a database enabled backend for persistent storage.

We plan to have a final product complete by 4 June 2025 and will provide functional (without all functions defined) products at regular intervals throughout the semester.

Restaurant Automation System will be developed using a partially concurrent development cycle. The first iteration will be completed in its entirety, then the development team will split, with more programmers working on the interface programming, and a small group that will extend and increase the robustness of the persistent storage subsystem. The third and final iteration will include further expanding the storage options and adding security into the system at all levels.

If time permits, we may optionally perform fourth iteration which will add a web-based interface to Restaurant Automation System.

2 High level functionality

The Restaurant Automation System is designed to simplify order processing and inventory management for small to medium-sized restaurants. Instead of spending time on manual tasks, restaurant staff can efficiently take customer orders, track stock levels, and manage daily operations through an intuitive interface. By automating these core processes, the system reduces errors, saves time, and ensures smoother service during busy hours.

The Restaurant Automation System is designed to simplify order processing and inventory management for small to medium-sized restaurants. Staff will be able to take customer orders through a streamlined, intuitive interface that mimics the ease of a digital order pad or spreadsheet. From these entries, the system generates real-time updates to inventory levels, ensuring that stock usage is automatically tracked without the need for manual logging.

If an ingredient runs low or is out of stock, the system clearly marks the item, allowing the staff or manager to take immediate action, such as adjusting menu availability or restocking inventory—all with just a few clicks. This reduces last-minute surprises and helps maintain service quality even during peak hours.

Additionally, the Restaurant Automation System supports categorization of menu items and ingredients into groups and subgroups, making it easy to manage and view related items together. This classification extends to suggesting substitute ingredients or menu adjustments when certain stock items are unavailable, eliminating the extra step of manually identifying alternatives and ensuring consistent service with minimal disruption.

3 Project Staffing

3.1 Assignment staff

The effort for each part of this assignment is listed below:

- Software Project Plan – Beren Elçin Polat
- Requirements Document – Melike Aytaç and Sultan Esen Murat
- The Restaurant Automation System Design Diagram – Yağmur Sabırılı
- Implementation Plan – Beren Elçin Polat and Melike Aytaç
- Test Plan – Sultan Esen Murat
- Presentation – Yağmur Sabırılı

3.2 Requirements

The Restaurant Automation System will fulfill specific requirements for streamlining order processing and inventory management tailored to small to medium-sized restaurants.

Melike Aytaç and Sultan Esen Murat are the people most familiar with those requirements.

The participants in the group will be:

- Melike Aytaç
- Sultan Esen Murat

3.3 Design

The Restaurant Automation System will incorporate many of the design principle

- Yağmur Sabırı
- Beren Elçin Polat
- Melike Aytaç

3.4 Iteration 1

Iteration 1 we will focus on the completion of the fundamental working of the system reserving most of the work on the graphical user interface until iteration 2. The purpose for this is to keep the focus on making the software correct before making the interface production quality.

The participants in the group will be:

- Yağmur Sabırı
- Beren Elçin Polat
- Melike Aytaç
- Sultan Esen Murat - Developer

3.5 Iteration 2

Iteration 2 will split the coding between two groups. Team one will focus on the User Interface, making that area robust and easy to use. Team 2 will focus on implementing additional persistent storage options for the Restaurant Automation System.

3.5.1 Interface Coding Group

The participants in this team will be:

- Sultan Esen Murat – Lead Developer
- Beren Elçin Polat – Developer
- Melike Aytaç – Developer

3.5.2 Persistent Storage group

The participants in this team will be:

- Yağmur Sabırılı – Lead
- Beren Elçin Polat – Developer

3.6 Iteration 3

This iteration will focus on increasing correctness of the requirements of the overall system and cleaning up any interface issues that arise due to testing during iteration 2.

The participants in the group will be:

- Melike Aytaç– Lead Developer
- Yağmur Sabırılı– Developer
- Beren Elçin Polat – Developer
- Sultan Esen Murat– Developer

3.7 Optional Iteration 4

Iteration 4, if included would complete a web-based interface to the Restaurant Automation System. This would include an http server and provide authentication based on the mechanism already provided by the Restaurant Automation System.

The participants in the group will be:

- Beren Elçin Polat – Lead Developer
- Sultan Esen Murat– Developer
- Melike Aytaç– Developer
- Yağmur Sabırılı– Developer

3.8 Testers

The participants in this group will be:

- Melike Aytaç, for iterations 1
- Sultan Esen Murat for iteration 2
- Yağmur Sabırılı for iteration 3

4 Software Process

We plan to use an iterative development cycle with three iterations for development of the Restaurant Automation System software.

The three iterations are as follows:

4.1 Iteration 1

Complete the underlying system architecture before providing any real work on the user interface. This iteration will be used to provide a solid proof of concept that the system design we have created will suffice for the requirements of the system.

Midway through this iteration, Beren Elçin Polat and Melike Aytaç will begin design on the user interface to be implemented in iteration 2.

4.2 Iteration 2

This iteration will be performed by two teams:

- Team 1 – GUI Implementation
 - Complete the graphical user interface
- Team 2 – Persistent Storage
 - Extend the options that the backend of the product will use.
 - § MySQL
 - Provide persistent storage security for all data stored in the storage subsystem to increase security.
 - § File encryption using known strong cryptographic algorithms (AES, etc.)

4.3 Iteration 3

This iteration will focus on increasing correctness of the requirements of the overall system and cleaning up any interface issues that arise due to testing during iteration 2.

Testing will be the focus for this iteration with the testing group being expanded to three testers to make sure that all defects are found and resolved before final delivery.

5 Software Engineering Methods

This section will provide and overview what methods we will use to create this product and how we will perform quality assurance testing.

5.1 Coding standard verification

We will perform code checks to verify that the code meets the coding standard and catalog and fix any standards violations.

5.2 Code reviews

As a team, we will meet every other week to conduct code reviews.

5.3 Coding Standard

We will follow a coding standard for this project that will increase readability and maintainability.

5.4 Object Oriented Design

We will use the object-oriented method for program design and keep this design updated with fixes for any potential problems we may catch. We will also perform design reviews throughout the project to ensure that the existing design will meet the criteria for the product.

6 Schedule and Effort

6.1 Schedule

Please see the attached schedule for the specifics of the schedule. We plan to have the following milestones:

6.1.1 Iteration 1

15 April 2025

6.1.2 Iteration 2

Graphics team

26 May 2025

Persistent Storage Team

11 May 2025

6.1.3 Iteration 3

4 June 2025

7 Measurements

We plan to track the following measurements:

- Categorized Developer time
 - Coding – Time Spent programming
 - Designing – Time spent doing design and re-design

- Testing – Time spent testing
 - Email – Time spent communicating with the group
- Code metrics
 - Source lines
 - Number of Classes
 - Number of Methods

8 Project Risks

The risks associated with this project are:

- Incomplete understanding of the requirements
 - The requirements for the Restaurant Automation System are generally well understood; however, there is a possibility that restaurant staff perform certain order or inventory-related tasks that involve complex human decision-making and cannot be easily replicated by automation. In such cases, a limited functionality version of the system may need to be delivered to ensure essential operations are still supported.
- Misconfiguration of the software tools due to inexperience
 - If the tools chosen for development are overly complex, improper setup may lead to data loss or wasted time due to software issues. This could delay the project timeline significantly. If the problem becomes severe enough to require switching tools during development, additional time may be lost as the team must learn how to use the new tool effectively.
- Steep learning curve for the new technologies included with the project
 - The Java GUI libraries are a new technology for most members of the team. If the API is difficult to learn and interact with, this could lead to time lost trying to get the interaction to work correctly.

- Team inexperienced with windows graphic programming
 - Most members of the team have not had extensive experience programming with the Java graphic API and significant productivity issues may be the result. This issue could become extremely serious if the learning curve for the technology is too steep for the programmers to master in a short time.

9 Software Tools

We plan to use the following tools:

9.1 Tools

9.1.1 Git

A distributed version control system used for source code management and collaboration.

9.1.2 GitHub

Cloud-based platforms providing a web interface for Git, project management features, issue tracking, and collaboration tools.

9.1.3 OpenJDK

The official Java Development Kit used to compile and run the Java application across platforms.

9.1.4 IntelliJ IDEA / VSCode

Modern, full-featured Integrated Development Environments (IDEs) for developing Java GUI applications.

9.1.5 Java GUI Toolkit

A modern GUI toolkit for building rich desktop applications in Java, offering components for UI design, animations, and media.

9.1.6 Microsoft Teams

Used for team communication, file sharing, and coordination during development.

9.1.7 Jira

Tools used for issue tracking, task assignment, and agile project management to help organize the development process.

10 Hardware Support

We plan to make use of the following hardware:

10.1 Development Machine

- **Purpose:** Primary development workstation for coding and UI design.
 - **Processor:** Intel Core i7
 - **RAM:** 16 GB
 - **Storage:** 512 GB SSD
 - **OS:** Windows 11
-

10.2 Testing Machine

- **Purpose:** Used for testing builds and verifying GUI compatibility.
 - **Processor:** Intel Core i5
 - **RAM:** 8 GB
 - **Storage:** 256 GB SSD
 - **OS:** Windows 10
-

10.3 CI/CD Build Server

- **Purpose:** Runs automated builds and tests (can be physical or virtual).
 - **Processor:** Intel Xeon
 - **RAM:** 32 GB
 - **Storage:** 1 TB SSD
 - **OS:** Ubuntu Server 22.04 LTS
-

10.4 Presentation/Demo Machine

- **Purpose:** Used for showcasing the application to stakeholders or instructors.
- **Processor:** Intel Core i5
- **RAM:** 8 GB
- **Storage:** 256 GB SSD
- **OS:** Windows 11

11 Software Support

We will require the following software:

- **11.1 Git**
 - A modern, distributed version control system used for source code management and collaboration among team members.
- **11.2 GitHub**
 - A cloud-based Git platform providing a web interface for source control, code review, issue tracking, and collaboration tools such as project boards and pull requests.
- **11.3 IntelliJ IDEA / VSCode**
 - Popular IDEs used for writing, debugging, and managing Java GUI code. IntelliJ IDEA provides full-featured Java support, while VSCode offers a lightweight alternative with Java plugins.
- **11.4 Java Development Kit (OpenJDK 17)**
 - Used to compile and run Java applications. The OpenJDK provides a cross-platform standard runtime and toolset for Java-based software.
- **11.5 Microsoft Teams**
 - A team communication and coordination platform used for messaging, file sharing, meetings, and project updates.
- **11.6 Jira**
 - A powerful issue tracking and project management tool used to manage bugs, tasks, sprints, and progress throughout the development lifecycle.

12 Personnel Support

The Restaurant Automation System team will require support from the following personnel:

- **Prof. Dr. Mehmet Süleyman Ünlütürk :**

- Supervising project progress and ensuring alignment with academic and technical standards.
- Offering insights and recommendations for improving system design and implementation.
- Reviewing project documentation and providing feedback.