REVISION HISTORY

Date	Version	Description	Author
15.02.2025	0.1	Name and description of the project, Gantt chart and an abbreviation have been added. 2.2.2 has been filled.	Tahsin Karcı
15.02.2025	0.2	Risks have been analyzed with its probabilities and effects. Potential indicators and strategies have been decided.	İrem Akova, İris Akdemir
16.02.2025	0.3	Activities and comments are added in section 3.	Emre Tekin, Elifnur Boncuk
16.02.2025	0.4	GitHub repository and software development tools have been added. 2.2.2 has been updated.	Mert Turan
16.02.2025	0.5	Responsibilities are added in section 3.	Emre Tekin, Elifnur Boncuk
16.02.2025	1.0	Table of contents page's numbers have been updated, transferring from Google Docs to Microsoft Word has been made, last touch has been done in the document.	Tahsin Karcı

TABLE OF CONTENTS

Revision History **1**
1 Identification **3**
1.1 Document overview ***3***
1.2 Abbreviations ***3***
1.2.1 Abbreviations 3
1.3 References ***3***
1.3.1 Project References 3
2 Software Development Activities **3**
2.1 Software development process ***3***
2.1.1 Overview of process phases 3
2.1.2 Technical documentation 4
2.1.3 Deliverables 4
2.2 Software development tools ***5***
2.2.1 Workstation 5
2.2.2 Requirements management and documentation 5
2.2.3 Software Design 5
2.2.4 Coding and automated tests 5
2.2.5 Configuration management 5
2.3 Software development rules and standards ***5***
3 Responsibilities **6**
3.1 Activities and responsibilities ***6***
4 Risk Assessment **6**
4.1 Risk Analysis ***6***
4.2 Risk Planning ***7***

1 Identification

1.1 Document overview

This document contains the software development plan of software RMS.

RMS is a restaurant management system which manages the reservation of the tables, orders and payments. This desktop application can be used by both staff and customers. Customers can order entries, make reservations, pay the bills while the staff can organize the background work according to that, checking the supply chain, scheduling the works and monitoring the demands.

1.2 Abbreviations

1.2.1 Abbreviations

RMS: Restaurant Management System
VS: Visual Studio
UML: Unified Modeling Language
IDE: Integrated Development Environment
JDK: Java Development Kit
SRS: Software Requirement Specification
STP: Software Test Plan
SDD: Software Design Document
STR: Software Test Report

1.3 References

1.3.1 Project References

#	Document Identifier	Document Title

2 Software Development Activities

The section lists and describes the software development activities of RMS software development project.

2.1 Software development process

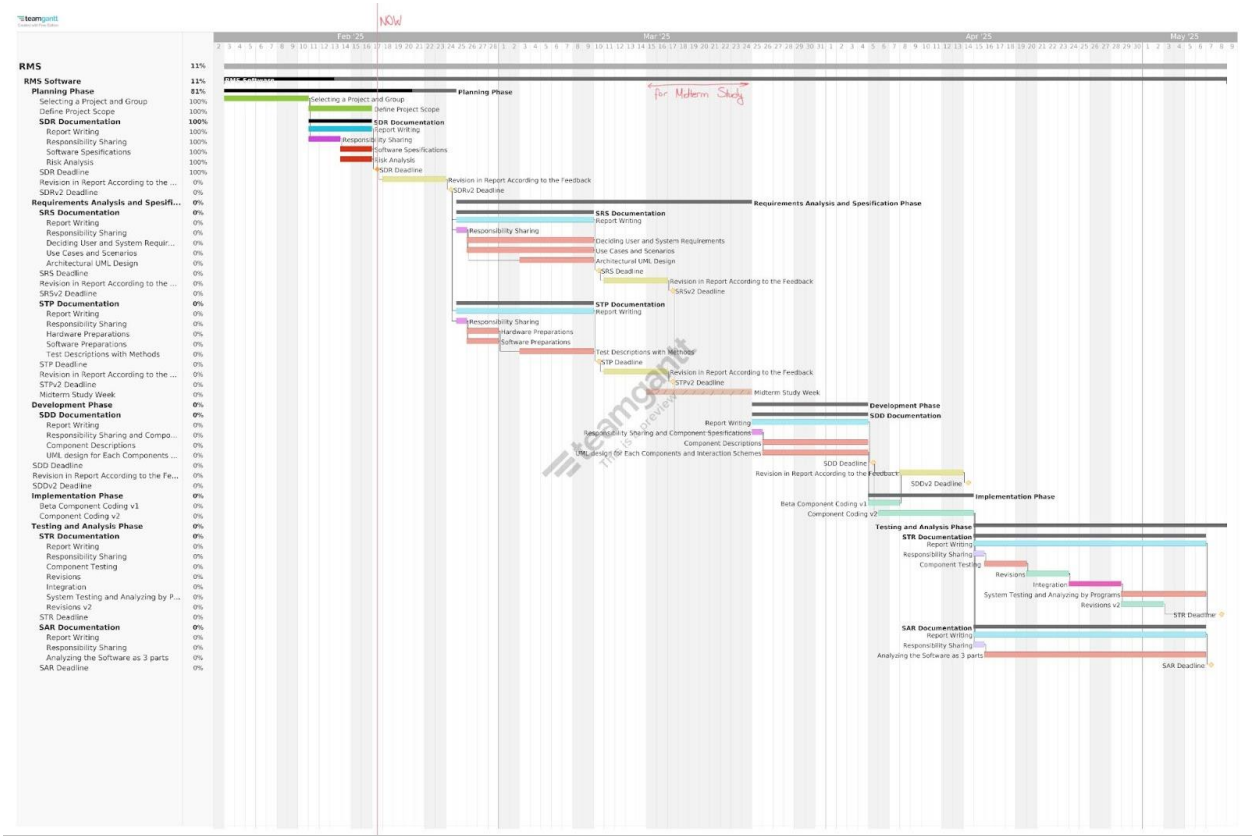
This is a course project, which adopts the waterfall model as the software development process.

2.1.1 Overview of process phases

The software development process for the project will be composed of the following phases:

- Planning
- Requirements Analysis
- Design
- Implementation
- Testing and Analysis

These phases will follow each other sequentially, where each phase starts just after the completion of the previous one. The following Gantt chart depicts the planned start date and duration for the phases.



2.1.2 Technical documentation

The following documentation is produced during the software development phases:

- Software specification: SRS, STP
- Software detailed conception: SDD
- Software tests phases : STR
- Software analysis: SAR

2.1.3 Deliverables

The following items will be delivered at the end of the process:

- Technical documentation as outlined in Section 2.1.2
- Software and its configuration files

2.2 Software development tools

2.2.1 Workstation

6 Laptops:

- 32 GB RAM, 1TB SSD, Intel Core i5 11300H @3.10 GHz, RTX 3050 Ti, Windows 11
- 8 GB RAM, 475 GB SSD, AMD Ryzen 5 4600H @3 GHz, GTX 1650, Windows 11
- 16GB RAM, 926 GB SSD, Apple M2 Pro, macOS 15.2
- 24GB RAM, 512 GB SSD, Apple M3, macOS 15.2
- 8GB RAM, 256GB SSD, Intel Core i5-1035G1 @1.00GHz (1.19GHz), Windows 11
- 16 GB RAM, 512GB SSD, Intel Core i7-9750H CPU @ 2.60GHz 2.60 GHz, NVIDIA GeForce GTX 1650, Windows 10

2.2.2 Requirements management and documentation

- Microsoft Word
- Google Docs
- Microsoft Excel
- GitHub Projects

2.2.3 Software Design

- Argo UML open source tool
- Microsoft Visio

2.2.4 Coding and automated tests

- Java 21
- SQL 15.0
- İntelliJ IDEA 2024.1
- JDK 21
- Docker 26.1.1
- GitHub Repository: https://github.com/Mert-Turan/RMS

2.2.5 Configuration management

GitHub[1] will be used for software configuration management and tracking issues regarding the software development. A public repository will be created for this purpose.

2.3 Software development rules and standards

UML[2] will be used for software design documentation.

[1] http://www.github.com
[2] http://www.uml.org/

3 Responsibilities

3.1 Activities and responsibilities

Activity	**Responsibility**	**Comment**
Project management	Tahsin Karcı	Responsible for requirement analysis (understanding restaurant needs) & risk assessment
Configuration tools management	Mert Turan	Responsible for the GitHub repository
Setting up the Development tools	Everybody	Responsible for setting up VS Code or Intellij
Database design	İrem Akova & İris Akdemir	Responsible for creating a relational schema for orders, menus, customers & reservations (ER Design)
Frontend development	Tahsin Karcı & Emre Tekin & Elif Nur Boncuk	Responsible for designing the Frontend of RMS
Backend development	Everybody	Responsible for writing the Backend of RMS

4 Risk Assessment

4.1 Risk Analysis

Risk	**Probability**	**Effect**
Key staff are unavailable at crucial situations	High	Serious
The organization is restructured	Moderate	Tolerable
The time required to develop the software is underestimated.	Moderate	Serious
Software tools cannot be integrated.	High	Tolerable
The size of the software is underestimated.	High	Tolerable
Customers are not able to understand the effect of requirements changes.	Moderate	Tolerable

4.2 Risk Planning

Risk	**Strategy**
Staff unavailability	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Organizational problems	Engage with new management to show how the project contributes business goals.
Unrealistic deadlines	Reevaluate project deadlines and milestones with the team, reorganizing them as necessary to make it manageable.
Defective tools	Replace potentially defective tools or investigate new integration methods.
Underestimated size of software	Re-evaluate the scope and size of the software. Reorganize the workload.
Requirements changes	Derive traceability knowledge to determine requirements change effect; maximize knowledge underlying in the design.

Risk type	**Potential indicators**
People	Lack of staff motivation; lack of work due to poor health conditions.
Organizational	Lack of action by senior management.
Estimation	Failure to meet the agreed schedule.
Tools	Inefficient software tools.
Requirements	Requests may change regarding too many requirements; negative feedback from customers.