Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/10/2020 | 0.1 | Completed sections from 2.1 to 3.1. Overview and abbreviations parts filled. | Abdullah Saydemir  Aleyna Ölmezcan  Burcu Arslan  Emin Sadikhov  Esad Simitçioğlu  Özge Yılgür |
| 15/10/2020 | 0.8 | Gantt chart, risk analysis and risk planning tables are added. | Abdullah Saydemir  Aleyna Ölmezcan  Burcu Arslan  Emin Sadikhov  Esad Simitçioğlu  Özge Yılgür |
| 22/10/2020 | 1.1 | Reference table added back to section 1.3.1  Gantt Chart redesigned in 2.1.1.  Provided workstation CPU models in the section 2.2.1  Added details to section 2.2.3 | Abdullah Saydemir  Aleyna Ölmezcan  Burcu Arslan  Emin Sadikhov  Esad Simitçioğlu  Özge Yılgür |
| 23/10/2020 | 1.2 | Version info is provided for IDEs and Junit is added as the testing tool in section 2.2.4  Table contents rearranged in sections 4.1 and 4.2 | Abdullah Saydemir  Aleyna Ölmezcan  Burcu Arslan  Emin Sadikhov  Esad Simitçioğlu  Özge Yılgür |
| 25/10/2020 | 1.3 | Added links to footer at page 5.  Changed Gantt Chart orientation to landscape to provide better view. | Abdullah Saydemir  Aleyna Ölmezcan  Burcu Arslan  Emin Sadikhov  Esad Simitçioğlu  Özge Yılgür |

**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 4

2.2.1 Workstation 4

2.2.2 Requirements management and documentation 4

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 6

# Identification

## Document overview

This document contains the software development plan of software OzU-Garage.

OzU-Garage is a desktop-based software application that allows OzU students to sell/buy products such as books, furniture, electronics and tickets. Students can log in with OzU mails and become buyers or sellers. They can open the advertisements of the products they want to sell or as a buyer, they can review the advertisements and choose the suitable for them.

## Abbreviations

### Abbreviations

OzU-G: OzU-Garage

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

## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
|  |  |  |

# Software Development Activities

The section lists and describes the software development activities of OzU-G software development project.

## Software development process

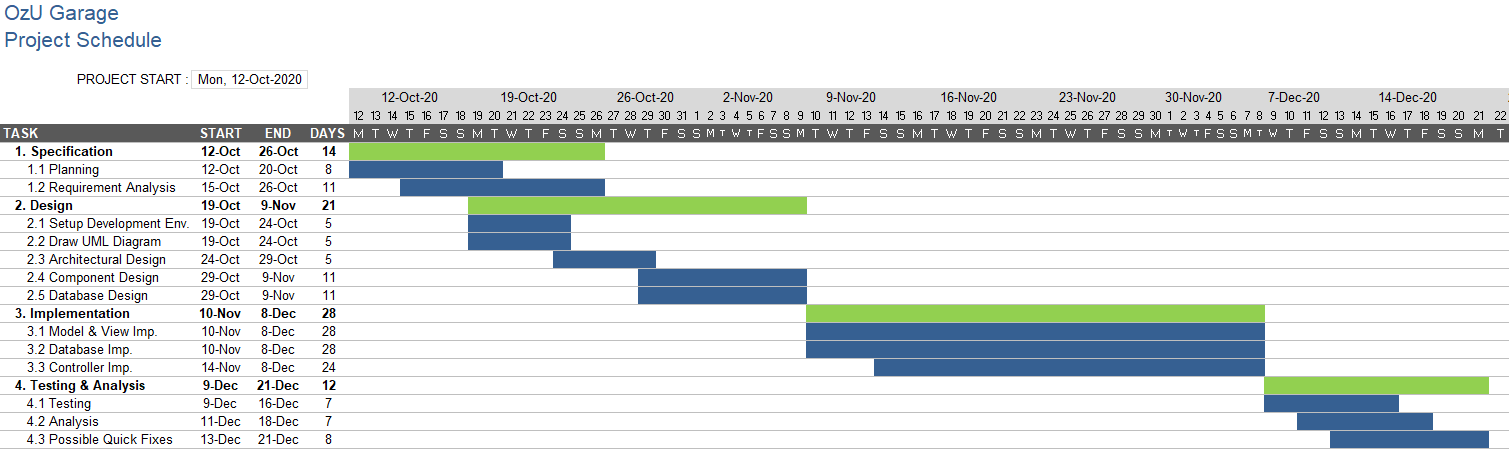
This is a course project, which adopts the waterfall model as the software development process.

### 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.



### 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

### 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

## 

## Software development tools

### Workstation

6 laptop computers and a cloud server machine with following configuration

* Intel® Core™ i5-7300HQ CPU, 2.50 GHz, 8 GB RAM, Windows 10
* Intel® Core™ i5-7360U CPU, 2.30 GHz, 8 GB RAM, MacOS Catalina
* Intel® Core™ i7-9750H CPU, 4.5 GHz, 16GB RAM, Windows 10
* Intel® Core™ i7-4702MQ CPU, 2.20 GHz, 8 GB RAM, Windows 10
* Intel® Core™ i5-6402P CPU, 2.80 GHz, 16 GB RAM Windows 10
* Intel® Core™ i7-77002HQ CPU, 2.80 GHz, 16 GB RAM Windows 10

### Requirements management and documentation

* Microsoft Word
* Zoom
* Microsoft Excel

### Software Design

Describe tools used for software design:

* Argo for UML Diagram
* Java Swing Library for GUI Design
* Google Cloud Server to store user and product information

### Coding and automated tests

* Java JDK 8
* JUnit 5
* IntelliJ IDEA v2020.2
* DataGrip v2020.2
* MySQL Workbench v8.0.20

### Configuration management

GitHub[[1]](#footnote-2) will be used for software configuration management and tracking issues regarding the software development. A public repository will be created for this purpose.

## Software development rules and standards

UML[[2]](#footnote-3) will be used for software design documentation.

Main coding language of the project is Java. Coding conventions of Java can be found at: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

In database development MySQL coding guidelines will be followed which can be reached at:

<https://dev.mysql.com/doc/dev/mysql-server/8.0.12/PAGE_CODING_GUIDELINES.html>

# Responsibilities

## Activities and responsibilities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Responsibility** | **Comment** |
| Project management | Emin Sadikhov | Responsible for project progress and making sure that project should finish in a timely manner. |
| Configuration tools management | Özge Yılgür | Being admin in GitHub account of the project. |
| Setting up the Development tools | Abdullah Saydemir | Making sure that everyone has necessary development tools. |
| Database design | Esad Simitçioğlu, Emin Sadikhov | Responsible for database connections. |
| GUI | Aleyna Ölmezcan, Abdullah Saydemir | Responsible for the GUI Design and development of the project. |
| Controller | Burcu Arslan, Özge Yılgür | Responsible of combining the parts of the database and GUI. |
| Model | Esad Simitçioğlu, Emin Sadikhov | Drawing the UML Diagram and creating the initial classes. |

# Risk Assessment

## Risk Analysis

|  |  |  |
| --- | --- | --- |
| **Risk** | **Probability** | **Effects** |
| The time needed to develop the software could be underestimated. | **High** | **Serious** |
| Software tools cannot work together in an integrated way. | **High** | **Tolerable** |
| Any group member could withdraw the course. | **Moderate** | **Serious** |
| Any group member could be ill or unavailable. | **High** | **Serious** |
| Lack of adjustability for requirements change. | **Low** | **Serious** |

## Risk-Planning

|  |  |
| --- | --- |
| **Risk** | **Strategy** |
| Time problems | Develop a more organized plan. |
| Software problems | Change for compatible software. |
| Staff problems | Reorganize the team by giving each member new tasks. |
| Requirement changes | Dividing code into versions so that it can be recoverable. |

1. http://www.github.com [↑](#footnote-ref-2)
2. http://www.uml.org

   3 https://cloud.google.com

   4 https://www.jetbrains.com/idea

   5 https://junit.org/junit5

   6 https://www.mysql.com/products/workbench

   7 https://www.jetbrains.com/datagrip

   8 https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html [↑](#footnote-ref-3)