**THE SUPERIOR COLLEGE LAHORE**

|  |
| --- |
|  |

**Faculty of Computer Science & IT**

**Department of Software Engineering**

**Final Year Project**

**PROJECT REPORT (Part-1)**

**[Out Task App]**

Project ID: **[write ID here Issued by FYP Manager]**

**Project Team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Program** | **Contact Number** | **Email Address** |
| Hamza Rafique | Bcsm-S17-072 | BSCS | 0304-4662898 | BCSM-S17-072@superior.edu.pk |
| Muhammad Suleman | Bcsm-S17-067 | BSCS | 0306-3133608 | BCSM-S17-067@superior.edu.pk |
| M. Zaryab | Bcsm-S17-088 | BSCS | 0300-7830313 | BCSM-S17-088@superior.edu.pk |

**[Zaman Aziz]**

([Designation])

**Project Report**

**[Out Task App]**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  | 1.0 |  | <Original Draft> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  | <Changes Based on Feedback From Faculty> |  |
|  |  |  | <Added Project Plan> |  |
|  |  |  | <Changes Based on Feedback from Supervisor> |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |
| --- | --- |
| **Project Supervisor** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

|  |  |
| --- | --- |
| **Project Manager** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| **Head of the Department** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

*This work is dedicated to my . . . . . .*

# Acknowledgements

I am really thankful to my supervisor *who has guided us well throughout the semester. As well as the people that provided us with their honest opinions and reviews. Who made time for us for conducting their interview from their busy schedule. We would also like to thank our friends that provided us with the opportunity and support for developing this project.*

# Executive Summary

People have a very busy routine due to which their home task remains pending and the work goes lengthy day by day. People go to the shop of plumber, Electricians etc. and hire them for the specific home task. This procedure is very time consuming. We are developing an application named as “Out Task”, with the help of this application people hire task performers like plumbers, electricians, labors at their home in a few minutes .People just post their requirement task and detail on this application in the form of deal and the nearest person who is eligible for this work/task can accept this deal and on the support the worker\task performer goes to the clients house and perform the task as the people wants. Another important purpose to make this application is that people are facing a very trouble to buy home kitchen items like soaps, cooking oil, meat, sugar etc. This is happening due to two reasons. One of them is busy working routine of people and the other major issue is corona virus. Now days, People are not going outside of their houses due to corona virus. They want all the things at their door step so this application is very helpful for people.

# Table of Contents

[Dedication iv](#_Toc505638416)

[Acknowledgements v](#_Toc505638417)

[Executive Summary vi](#_Toc505638418)

[Table of Contents vii](#_Toc505638419)

[List of Figures ix](#_Toc505638420)

[List of Tables x](#_Toc505638421)

[Chapter 1 1](#_Toc505638422)

[Introduction 1](#_Toc505638423)

[1.1. Background 2](#_Toc505638424)

[1.2. Motivations and Challenges 2](#_Toc505638425)

[1.3. Goals and Objectives 2](#_Toc505638426)

[1.4. Literature Review/Existing Solutions 2](#_Toc505638427)

[1.5. Gap Analysis 2](#_Toc505638428)

[1.6. Proposed Solution 2](#_Toc505638429)

[1.7. Project Plan 3](#_Toc505638430)

[1.7.1. Work Breakdown Structure 3](#_Toc505638431)

[1.7.2. Roles & Responsibility Matrix 3](#_Toc505638432)

[1.7.3. Gantt Chart 3](#_Toc505638433)

[1.8. Report Outline 3](#_Toc505638434)

[Chapter 2 4](#_Toc505638435)

[Software Requirement Specifications 4](#_Toc505638436)

[2.1. Introduction 5](#_Toc505638438)

[2.1.1. Purpose 5](#_Toc505638439)

[2.1.2. Document Conventions 5](#_Toc505638440)

[2.1.3. Intended Audience and Reading Suggestions 5](#_Toc505638441)

[2.1.4. Product Scope 5](#_Toc505638442)

[2.1.5. References 6](#_Toc505638443)

[2.2. Overall Description 6](#_Toc505638444)

[2.2.1. Product Perspective 6](#_Toc505638445)

[2.2.2. Product Functions 6](#_Toc505638446)

[2.2.3. User Classes and Characteristics 6](#_Toc505638447)

[2.2.4. Operating Environment 7](#_Toc505638448)

[2.2.5. Design and Implementation Constraints 7](#_Toc505638449)

[2.2.6. User Documentation 7](#_Toc505638450)

[2.2.7. Assumptions and Dependencies 7](#_Toc505638451)

[2.3. External Interface Requirements 8](#_Toc505638452)

[2.3.1. User Interfaces 8](#_Toc505638453)

[2.3.2. Hardware Interfaces 8](#_Toc505638454)

[2.3.3. Software Interfaces 8](#_Toc505638455)

[2.3.4. Communications Interfaces 9](#_Toc505638456)

[2.4. System Features 9](#_Toc505638457)

[2.4.1. System Feature 1 9](#_Toc505638458)

[2.4.1.1. Description and Priority 9](#_Toc505638459)

[2.4.1.2. Stimulus/Response Sequences 9](#_Toc505638460)

[2.4.1.3. Functional Requirements 9](#_Toc505638461)

[2.4.2. System Feature 2 10](#_Toc505638462)

[2.4.2.1. Description and Priority 10](#_Toc505638463)

[2.4.2.2. Stimulus/Response Sequences 10](#_Toc505638464)

[2.4.2.3. Functional Requirements 10](#_Toc505638465)

[2.4.3. System Feature 3 (and so on) 11](#_Toc505638466)

[2.5. Other Nonfunctional Requirements 11](#_Toc505638467)

[2.5.1. Performance Requirements 11](#_Toc505638468)

[2.5.2. Safety Requirements 11](#_Toc505638469)

[2.5.3. Security Requirements 12](#_Toc505638470)

[2.5.4. Software Quality Attributes 12](#_Toc505638471)

[2.5.5. Business Rules 12](#_Toc505638472)

[2.6. Other Requirements 12](#_Toc505638473)

[Chapter 3 13](#_Toc505638474)

[Use Case Analysis 13](#_Toc505638475)

[3.1. Use Case Model 14](#_Toc505638476)

[3.2. Fully Dressed Use Cases 14](#_Toc505638477)

[Chapter 4 15](#_Toc505638478)

[System Design 15](#_Toc505638479)

[4.1. Architecture Diagram 16](#_Toc505638480)

[4.2. Domain Model 16](#_Toc505638481)

[4.3. Entity Relationship Diagram with data dictionary 16](#_Toc505638482)

[4.4. Class Diagram 17](#_Toc505638483)

[4.5. Sequence / Collaboration Diagram 17](#_Toc505638484)

[4.6. Operation contracts 17](#_Toc505638485)

[4.7. Activity Diagram 18](#_Toc505638486)

[4.8. State Transition Diagram 18](#_Toc505638487)

[4.9. Component Diagram 18](#_Toc505638488)

[4.10. Deployment Diagram 19](#_Toc505638489)

[4.11. Data Flow diagram [only if structured approach is used - Level 0 and 1] 19](#_Toc505638490)

[Chapter 5 20](#_Toc505638491)

[Implementation 20](#_Toc505638492)

[5.1. Important Flow Control/Pseudo codes 21](#_Toc505638493)

[5.2. Components, Libraries, Web Services and stubs 21](#_Toc505638494)

[5.3. Deployment Environment 21](#_Toc505638495)

[5.4. Tools and Techniques 22](#_Toc505638496)

[5.5. Best Practices / Coding Standards 22](#_Toc505638497)

[5.6. Version Control 22](#_Toc505638498)

[Appendices 23](#_Toc505638499)

[Appendix A: Information / Promotional Material 24](#_Toc505638500)

[Reference and Bibliography 27](#_Toc505638502)

[Index 29](#_Toc505638503)

# List of Figures

1.1 Caption of first figure of first chapter 6

1.2 Caption of second figure of first chapter 7

2.1 Caption of first figure of second chapter 14

2.2 Caption of second figure of second chapter 22

2.3 Caption of third figure of second chapter 26

5.1 Caption of first figure of fifth chapter 49

5.2 Caption of second figure of fifth chapter 49

# List of Tables

1.1 label of first table of first chapter 6

1.2 label of second table of first chapter 7

2.1 label of first table of second chapter 14

2.2 label of second table of second chapter 22

2.3 label of third table of second chapter 26

5.1 label of first table of fifth chapter 49

5.2 label of second table of fifth chapter 49

# Chapter 1

# Introduction

**Chapter 1**

**Introduction**

Task means work which are necessary to do till a specific time. Tasks are based on the nature of relation of work with time. Some tasks are done in home but other tasks are meant to done from outside of the house such as buying different products from different shops. Our task out app work on the same principle. Also, it is providing the opportunities to the work less people to do the work. A person can post his or her need of buying specific things from home and can get that things staying in home. This app provides the facility to job seeker and as well as the employer to work under one roof. Both the job seeker and employer can register themselves in this app can find the required person which is suitable for both in a very limited time at one place. Both the job seeker and employer have facility to check the details of each other. Both can chat with each other through this app.

## Background

The idea of building this app comes from the freelancing apps like fiverr, freelancer and upwork. The background of making this application is to reduce the movement of people around. With some people the need of daily products can be fulfil that will help in decreasing the movement. Similarly, the weather also effect the need of people. Sometime it is not possible for every person to go out and get need filed. Our app will also help in those conditions. The recent covid-19 pandemic also lead in building of this application.

## Motivations and Challenges

We are developing this app for the ease of all people who could not go outside for some kind of reasons. We want to bring the change in the society so that there all need could be fulfill by staying at one place. Our challenge is to build this app in professional way and make the people use of this app for their own benefits.

## Goals and Objectives

Our goal is to develop this app in user friendly way so that any kind of person who even does not good in English. It will be efficient to use and less time consuming. Team work is a key to success, so we work as a team and we will develop this system in given duration of time.

## Literature Review/Existing Solutions

The existing solution is a manual system. We randomly goes to the shops and hair labor, Electricians, Plumbers etc. for doing our specific tasks. It is a very time consuming procedure. On the other hand super tasker mobile application is working on it but not completely because many people don’t know about it. They want personal information like CNIC of labors, Electricians and different task performing workers. People hesitate for entering the personal data on the mobile application because they think our data will leak from this application for doing crimes.

## Gap Analysis

Delivery options are available at only the big stores, big restaurants nowadays but they are very limited. That is why people has very less options to fulfil their need by staying home from every shop, market and restaurants. Sometime someone need a thing which is available only at one place and that place is not offering the delivery options and the person cannot reach to that place. This thing create problems for some person. This thing also becomes a reason to loss of customers.

## Proposed Solution

So, we are going to build an app which will help the persons not to personally visit a place for a specific thing. He or she can have that thing in home by using this application. This application will help the decreasing of the movement of people and along with it this will accommodate both the customers and other persons.

## Project Plan

We are developing an application for making the life of people easy. People hire task performers from this application to complete their home tasks.

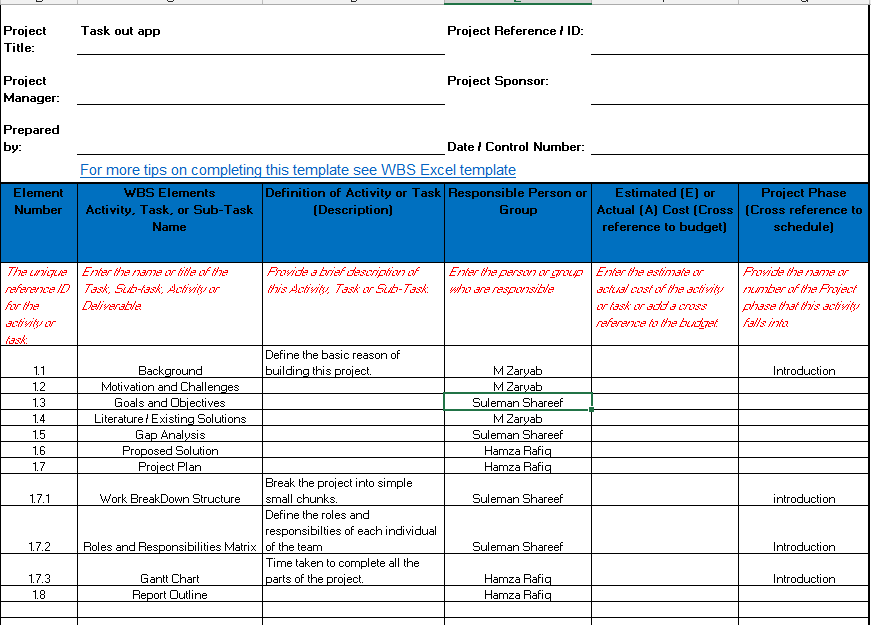
• First we have to gather requirements.

• Then we design our system including database and signup, login page.

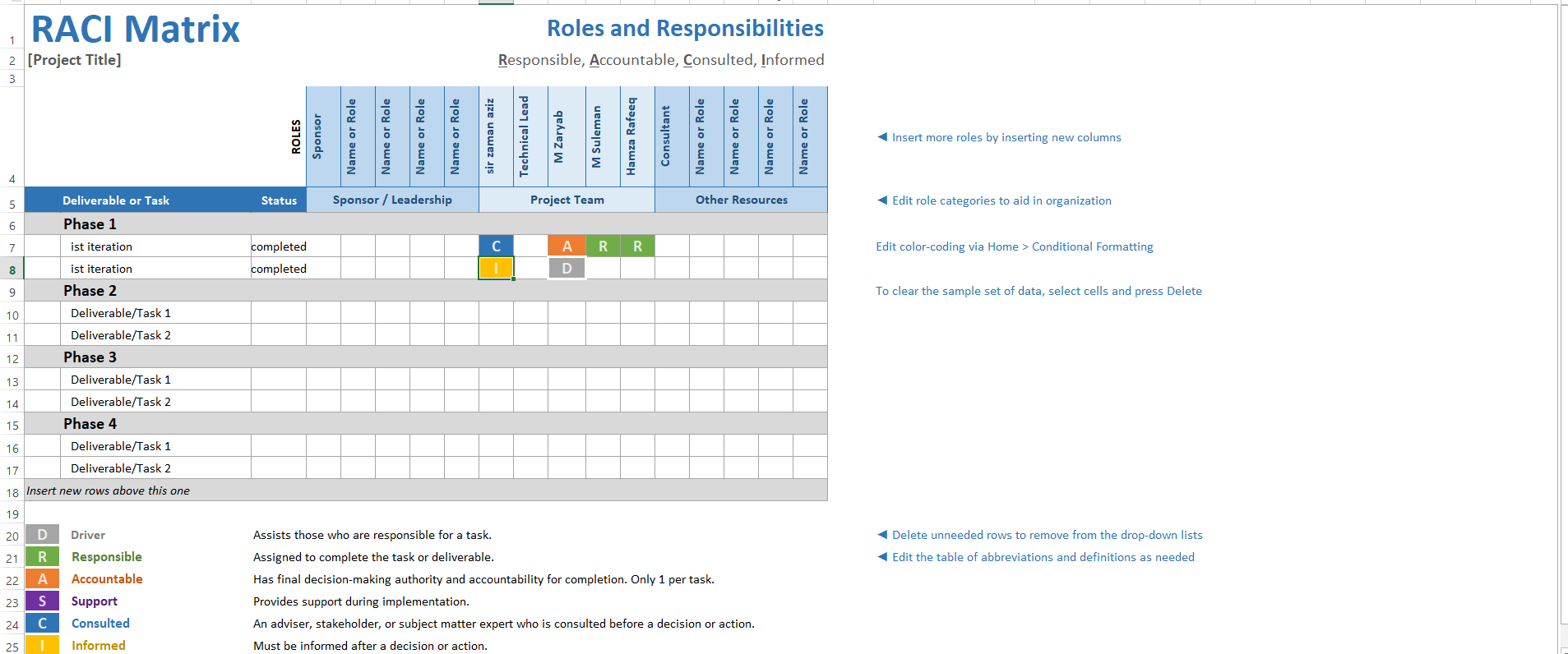
• Testing

• Deployment

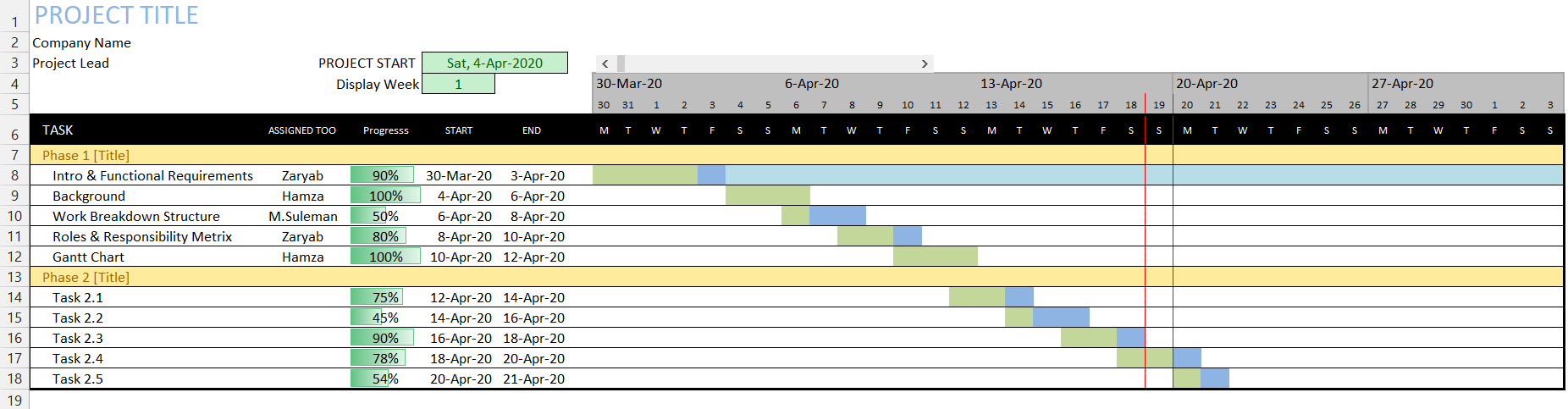
## Work Breakdown Structure



## Roles & Responsibility Matrix



## Gantt Charts



## Report Outline

* User / employee registration
* User / employee login
* User and employee can login with a registered account
* User and employee can be registered themselves
* User posting projects
* User can post his or her require work with a little description and budget
* Employee can browse and select from different projects
* User accepting available employee
* User can accept the employee who wants to work with.
* Payment status
* Payment status can be viewed and changed after the completion and payment
* Payment modes
* By Cash / By Card (Debit / Credit)

# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

## Purpose

This document presents the detailed explanation of the objectives, feature, app interface and applications of Task-Out App in real life. It also describes how the system performs and under which condition it must operate. In this document there will be listed all the features and functions of the software and its requirements.

## Document Conventions

|  |  |
| --- | --- |
| **SHORT FORM** | **FULL FORM** |
| ERD | Entity relationship diagram |
| DB | Database |
| SRS | Software requirement specification |
| UML | Unified Modeling Language |
| DFD | Data flow diagram |
| WBS | Work breakdown structure |
| IDE | Integrated development environment |
| GUI | Graphical user interface |
| HTTP | Hypertext transfer protocol |
| FTP | File transfer Protocol |
| WAN | Wide Area network |
| LAN | Local area network |

## Intended Audience and Reading Suggestions

It is restricted within the university premises. This has been implemented under the guidance of university professors. This project is useful for the house hold people and as well as the job seeker.

Reading suggestions are to start reading from the top till bottom and read that portion again if can’t understand what it means.

## Product Scope

**Moderator**

* Moderator can add job seeker and employer.
* Moderator can remove employer as well as job seeker.
* Moderator can notify job seeker and employer.
* Moderator can warn a job seeker and an employer.

**Job Seeker**

* Job seeker can login from his/her account.
* Job seeker can register his/her account.
* Job seeker can edit or delete his/her account.
* Job seeker can restore his/her password.
* Job seeker can view an advice.
* Job seeker can view pending jobs.
* Job seeker can search different jobs.
* Job seeker can search an employer.
* Job seeker can view the employer’s details.
* Job seeker can chat with the employer.

**Employer**

* Employer can login from his/her account.
* Employer can register his/her account.
* Employer can edit or delete his/her account.
* Employer can restore his/her password.
* Employer can post a job.
* Employer can delete a job.
* Employer can search a job seeker.
* Employer can see details about a job seeker.
* Employer can chat with the job seeker.
* Employer can track the job seeker with map.

**Administrator:**

* Administrator handles the application.
* He is concerned about its speed and optimization.

## References

<https://krazytech.com/projects> we used this website to get help related to SRS.

## Overall Description

## Product Perspective

Task means work which are necessary to do till a specific time. Tasks are based on the nature of relation of work with time. Some tasks are done in home but other tasks are meant to done from outside of the house such as buying different products from different shops. Our task out app work on the same principle. Also, it is providing the opportunities to the work less people to do the work. A person can post his or her need of buying specific things from home and can get that things staying in home. This app provides the facility to job seeker and as well as the employer to work under one roof. Both the job seeker and employer can register themselves in this app can find the required person which is suitable for both in a very limited time at one place. Both the job seeker and employer have facility to check the details of each other. Both can chat with each other through this app.

## Product Functions

Product functions are as follows:

1. **Log in**: This function authenticates the employer and job seeker.
2. **Post job**: Employer can post his or her job .
3. **Make appointment**: A valuable time can be set between the job seeker and employer.
4. **Create the profile**: Employer and as well as the job seeker can make his profile by using this function.
5. **Chat:**  This function helps the employer and job seeker to chat with each other.

## User Classes and Characteristics

|  |  |  |
| --- | --- | --- |
| **Classes** | **Functionality Privilege** | **Technical/Non-Technical** |
| **Employer** | Create profile, edit profile, chat  And upload status, View homepage, Make appointment, cancel appointment and post a job. | Technical |
| **Job seeker** | Create profile, edit profile, chat  And upload status, View homepage, view vedios, view blogs and search the jobs and employer and accept a job. | Technical |
| **Moderator** | Can add job seeker, employer as well as can remove them. | Technical |
| **Administrator:** | Concerned about the app speed and its optimization. | Technical |

## Operating Environment

* Spiral model is being used in the development of this app.
* For Database Use For Record like add profile, delete profile, update profile We Will Use Tool: SQLite.
* For front end and back end coding including user interface (server side, user side) and database connectivity with front end interfaces we used Android studio and java JDK.
* We used windows 7, 8, 8.1, 10 as operating systems.
* We used firebase to setup local servers.
* Web browser were used like google chrome.
* we used XML and Java as programming languages.

## Design and Implementation Constraints

* Default language will be English.
* We should follow the IEEE standards.
* Project will follow all the copyright and cyber lines of PTA (Pakistan Telecommunication Authority).

## User Documentation

It will provide specific guidelines to a pet parents for using the Taks out app. Further video slide show will be provided which will represent the whole system functions and how it works.

## Assumptions and Dependencies

Task out app is a self-contained project, the only explicit dependencies are those stated in the Design and Implementation Constraints.

## External Interface Requirements

## 

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## 

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

The systems require Data Base also for the store the data of the system like SQLite.

To access database required SQLite. Android studio. The system will run on Sql server so it requires any scripting language like Java, XML etc. User need web browser for interact with the system. Operating system must be with GUI supportability like windows (XP and higher versions), Linux, Macintosh (leopard or OSX version or higher), android OS (jellybean or higher).

## Communications Interfaces

Communication function required the internet protocol version 6 and it will follow HTTPS. It will use FTP for whole system with local server.

## System Features

1. **Log in**: This function authenticates the employer and job seeker.
2. **Post job**: Employer can post his or her job .
3. **Make appointment**: A valuable time can be set between the job seeker and employer.
4. **Create the profile**: Employer and as well as the job seeker can make his profile by using this function.
5. **Chat:**  This function helps the employer and job seeker to chat with each other.

## System Feature 1

This feature is one of the main features listed. It will help the employer to find the right person for the job.

## Description and Priority

This is the main component of our app which clearly defines its priority.

## Stimulus/Response Sequences

To remove the extra time which can be very hectic sometime this app will help to done.

## Functional Requirements

* REQ-SF1-1: authentication employer Registration
* REQ-SF1-2: authentication job seeker Registration
* REQ-SF1-3: log in/log out
* REQ-SF1-4: post job/delete job
* REQ-SF1-5: confirm job seeker
* REQ-SF1-6: chat
* REQ-SF1-7: profile creation
* REQ-SF1-8: Create pet profile, update profile
* REQ-SF1-9: view job status

## System Feature 2

File a Dispute

## Description and Priority

This feature is the backbone of our app for the employer which clearly defines its priority.

This feature helps the employer when his or her work is not done in time or in given

requirements. Then the employer can open dispute against the job seeker.

## Stimulus/Response Sequences

To increase the trust of employer on this product and helps him or her to get done his

work according to his or her requirements.

## Functional Requirements

* REQ-SF1-1: authentication employer Registration
* REQ-SF1-2: authentication job seeker Registration
* REQ-SF1-3: log in/log out
* REQ-SF1-4: post job/delete job
* REQ-SF1-5: confirm job seeker
* REQ-SF1-6: chat
* REQ-SF1-7: profile creation
* REQ-SF1-8: Create pet profile, update profile
* REQ-SF1-9: view job status

## System Feature 3 (and so on)

Report employer

## Description and Priority

This feature is the backbone of our app for the job seeker which clearly defines its priority.

This feature helps the job seeker when there is a problem with the employer in case of any

he can report about the problem and the employer in this feature.

## Functional Requirements

* REQ-SF1-1: authentication employer Registration
* REQ-SF1-2: authentication job seeker Registration
* REQ-SF1-3: log in/log out
* REQ-SF1-4: post job/delete job
* REQ-SF1-5: confirm job seeker
* REQ-SF1-6: chat
* REQ-SF1-7: profile creation
* REQ-SF1-8: Create pet profile, update profile
* REQ-SF1-9: view job status

**Functional requirements**

**Job seeker / employer login:**

User and employee can login with a registered account

**Job seeker / employer registration:**

User and employee can be registered themselves

**Employer posting projects:**

User can post his or her require work with a little description and budget

**Job seeker selecting projects:**

Employee can browse and select from different projects

**Employer accepting available job seeker:**

User can accept the employee who wants to work with.

**Hire:**

Employer can hire a job seeker by watching the details about the job seeker.

**Apply;**

Job seeker can apply for the different jobs which would be being shown.

**Payment status:**

Payment status can be viewed and changed after the completion and payment

**View details:**

Employer and job seeker can view the details of each other individually.

**Chat :**

Employer and job seeker can chat each other in the app.

**Report and disputes:**

Employer and job seeker can report the disputes about each other on the problems.

## Other Nonfunctional Requirements

## Performance Requirements

* Initial load time of this product Is very minimum.
* This product is based on the SQL-Lite.
* To make the search faster fast searching algorithms is used.
* App information will be fully secure through POS system.

## Safety Requirements

* The whole system secures. Only administrator and moderator can access the all data.
* The system is using POS system.
* The system will use HTTPS because it is more secure.
* The source code developed for the system shall be maintained in the Android studio

## Security Requirements

The security feature from having a log in for all employer and job seeker personal information. The log in detailed will be used in the app also. So, the chances of app getting intruded are very less.

## Software Quality Attributes

1. **Reliability :**

The product should be reliable to use and the product will sustain any condition.

* **Maintainability:**

The system will be easy to upgrade and it would be maintain easily.

* **Usability:**

This product can be used by any age of person.

* **Efficiency:**

This product will work according to required speed and will give performance within time.

* **Security:**

This product will be safe to use by any employer and job seeker and any kind of information will not be leaked or damaged for any personal use.

* **Testability:**

System will be easy to test and find defect. It required will be easy to divide in different modules for testing.

* **Reusability:**

This product has wide reusability range as almost all vet clinic have same type of modules.

## Business Rules

Illegal duplication of the reports should be strictly deal with. The system authentications should have full detailed of any person registered to the system.

## Other Requirements

* **Licensing requirement**

Not Applicable

* **Legal and copy rights**

All rights reserved by our team.

* **Application standard**

As per industry standard.

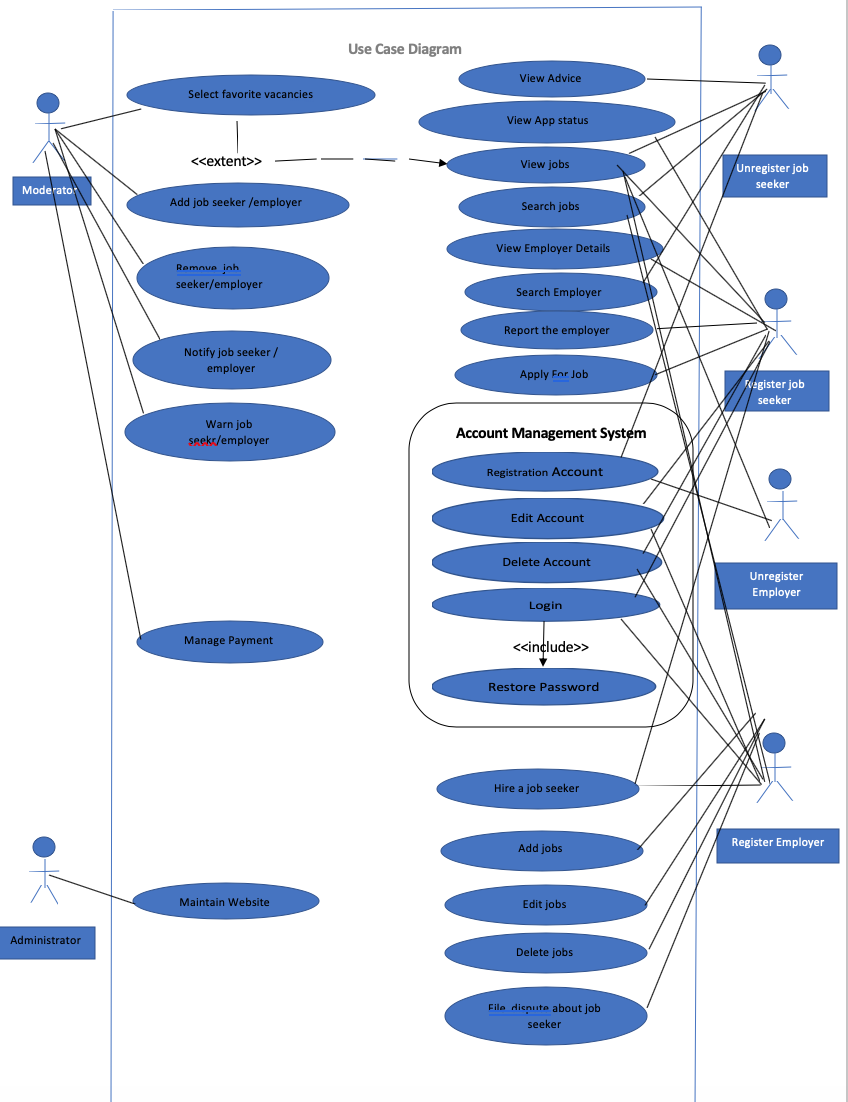
# Chapter 3

# Use Case Analysis

**Chapter 3:** System Analysis

Creating use cases to describing the whole scenario how it is going to work. Describing actors & their purposes. How users will Log In, create accounts make purchases. And users can edit order within specific time limit.

## Use Case Model



## Fully Dressed Use Cases

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | login |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer and the job seeker login in the app. This helps in authentication the original users of the app |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | 1. The user enters the ID and password and submit. 2. The system validate the account. 3. The system verifies the account. 4. Then the system show their relevant page. 5. The use case ends. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | View jobs |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to find the available jobs which are need to be done in a given time. |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validate the account. * The system verifies the account. * Then the system show their relevant page. * By selecting this usecase the available job will be shown. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Search jobs |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to search jobs which are need to be done in a given time. User can search the jobs which are related to its skills. |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * By selecting this use case the searched required jobs will be shown. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Search jobs |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to search jobs which are need to be done in a given time. User can search the jobs which are related to its skills. |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * By selecting this use case the searched required jobs will be shown. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Search employer details |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to search a specific employer and view his details |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Report the employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to report the administrator about the employer if they are no table to resolve their problem in the given time |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | chat |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to stalk with the employer in very efficient way and increase his chances to get hired. |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Report the employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the job seeker to report the administrator about the employer if they are no table to resolve their problem in the given time |
| **Primary Actor:** | Job seeker |
| **Secondary Actor:** | employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Add a job |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer to add and post a job so that job seeker can view that job and apply for that job in very quick time. |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Delete a job |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer to delete a job which is now not need to be done or somehow specification or the requirements of the job has been changed. |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Edit a job |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer to edit a job which is now not need to be done or somehow specification or the requirements of the job has been changed. |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Hire a job seeker |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer to hire a job seeker directly after viewing his details. |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | File dispute |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the employer to file a dispute against the job seeker if they do not come on one page after the job is near to done. |
| **Primary Actor:** | Employer |
| **Secondary Actor:** | Job seeker |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Add a job seeker/employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the moderator to add job seeker and employer is the app if they both meet the requirements of the app |
| **Primary Actor:** | moderator |
| **Secondary Actor:** | Job seeker/ employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Remove a job seeker/employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the moderator to remove job seeker and employer is the app if they both do notmeet the requirements of the app |
| **Primary Actor:** | moderator |
| **Secondary Actor:** | Job seeker/ employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Notify job seeker/employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the moderator to notify job seeker and the employer if they violates the rules of the policy of using this app. |
| **Primary Actor:** | moderator |
| **Secondary Actor:** | Job seeker/ employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | warn job seeker/employer |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the moderator to notify job seeker and the employer if they violate the rules of the policy of using this app. |
| **Primary Actor:** | moderator |
| **Secondary Actor:** | Job seeker/ employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

|  |  |
| --- | --- |
| **Use-case ID:** | Use\_case\_1 |
| **Name:** | Maintenance |
| **Created By:** | M Zaryab |
| **Date Created:** | 11-6-2020 |
| **Description:** | This use case allows the administrator to maintain the speed of site, report issues and problems about the app. Administrator using maintenance use case has the main role in the better functioning of this app. |
| **Primary Actor:** | moderator |
| **Secondary Actor:** | Job seeker/ employer |
| **Preconditions:** | 1.user have valid account. |
| **Postconditions:** | 1.The system display their relevant page. |
| **Main Flow:** | * The user enters the ID and password and submit. * The system validates the account. * The system verifies the account. * Then the system shows their relevant page. * Then the use case ended. |

# Chapter 4

# System Design

**Chapter 4:** System Design

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

## Architecture Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Domain Model

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Entity Relationship Diagram with data dictionary

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Class Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Sequence / Collaboration Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Operation contracts

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Activity Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## State Transition Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Component Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Deployment Diagram

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Chapter 5

# Implementation

**Chapter 5:** Implementation

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

## Important Flow Control/Pseudo codes

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Components, Libraries, Web Services and stubs

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Deployment Environment

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Tools and Techniques

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Best Practices / Coding Standards

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

## Version Control

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Appendices

# Appendix A: Information / Promotional Material

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this appendix all about*]

* 1. **Broacher**

* 1. **Flyer**
  2. **Standee**
  3. **Banner**
  4. **First Level heading [16 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + 1. **Second level heading [14 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + - 1. **Third level heading [12 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Appendix [no.]: Appendix Title

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

[*Between 4 to 8 lines describe what is this chapter all about*]

* 1. **First Level heading [16 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + 1. **Second level heading [14 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

* + - 1. **Third level heading [12 pt, Calibri, Bold, Left aligned]**

[Paragraph Text 12 pt, Calibri, 1.5 Line Spacing, Justified]

# Reference and Bibliography

**Reference and Bibliography**

[1] M. Sher, M. Rehman, “*Title of the Paper*” Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.

[2] ……

# Index

**Index**

**[A]**

**[B]**

**[C]**