Paintio(An android Based App for paint Job Booking)

By
Haroon Akbar

Supervised By
Mamona Afsheen

Department of computer science
Quaid-i-Azam University
Islamabad

Abstract

Paintio (Android app for paint job booking) application is an online mobile application that provides the customer to book a painter for his paint job. Through this app the customer can post the paint job and will be able to bid with the painter in order to get the best offer. Paintio will be beneficial to customer and painter as well. Painter will also be able to get the job by using this app so he will not have to roam around.

As there is no platform for paint job booking by biding for both painters and customers. Customers search the painters by going to different shops or by different references. With the help of this application customer will be able to find the painter with good ratings and desire rates.

- Customer can create a job by giving the attributes with the help of application.
- Customer can then post the job by giving the date and rate.
- Painters can search for the jobs from different cities.
- Painters can bid on a job by giving his price according to the job.
- After the job is done both the customer and painter can rate each other. This will eventually help for their future.

In this application, the profile of both the customer and painter will be maintained so for painters, it will help them to find lucrative and reliable jobs for them in future and for customer, it will help them to find the reliable and skillful painters. Both of them have to register to the application first by giving the credentials in order use the above features of the application.

Table of Content

List of Figures

List of Tables

1 Chapter: Introduction

1.1 Problem:

Now a day's people do not get the quality work regarding paint. And many painters are unemployed because there is no such platform for them to get work. And people are deceived by other people because the lack of exposure.

1.2 Idea:

Idea is to develop an android app for the painters which provides platform for both i.e. customer and painters to connect, communicate and bid with each other so the life becomes easy.

1.3 Project Overview:

Painter will register themselves on this application by creating their profile. Customer will be post paint jobs after creating profile on the application. Painters will be able to find jobs according to the region and will be able to bid on the job and can also pass remarks. Customer will then be able to select the painter for their particular job according to their requirement and ratings. After the job both will be able to rate each other.

1.4 Project Deliverables:

Software Project Management Plan SPSM:	1st deliverable
Software Requirements Specification SRS:	1st deliverable
Software Design Description SDD:	2nd deliverable
Software Test Documentation	2nd deliverable

Table 1.4.1 Project Deliverables

1.5 Software Process Model:

In the implementation of this project I prefer to use Waterfall Process Model which shows execution of processes in a sequential manner. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order. [1]

1.6 Roles and Responsibilities:

Following are the roles that define the foundations of the project.

Project Plan	• Tasks to be perform in every cycle of development. • Schedule for the next task to be done. • Progress according to the plan.
Requirements Specification and Analysis	• Specify and analyze every details of the application and proceed to the designing phase.
Designing	• Design the basics and complete application requirements.
Source Code	• Developing phase and providing source code of the application.
Test Plan	• Testing the application, using various testing tools and techniques
Final Deliverable	• After successfully completion of the previous mentioned phases final deliverable product will be provided with a demo.

Table 1.6.1 Roles and Responsibility

1.7 Tools and Techniques Used:

I will use object-oriented methodology. Also I will use Android studio to build this application using JAVA and database will be stored at SQL Server(xampp server). And for the web service I will use Anaconda Spyder using python.

1.8 Software Project Management Plan (SPMP):

	Name	Duration	Start	Finish	Predecessors
	Project Planning	4 days?	**11/6/17 8:00 AM**	**11/9/17 5:00 PM**	
	Intro to Project planning &	1 day?	11/6/17 8:00 AM	11/6/17 5:00 PM	
	Intro to projects & allocatic	1 day?	11/7/17 8:00 AM	11/7/17 5:00 PM	2
	Development of the projec	1 day?	11/8/17 8:00 AM	11/8/17 5:00 PM	3
	Review of Project planning	1 day?	11/9/17 8:00 AM	11/9/17 5:00 PM	4
	Collect Requirements	3 days?	**11/13/17 8:00 AM**	**11/15/17 5:00 PM**	
	Collect functional reqd	1 day?	11/13/17 8:00 AM	11/13/17 5:00 PM	
	Collect non - functional req	1 day?	11/14/17 8:00 AM	11/14/17 5:00 PM	8
	Meeting with customer	1 day?	11/15/17 8:00 AM	11/15/17 5:00 PM	9
	Defining Requirments	2 days?	**11/16/18 8:00 AM**	**11/19/18 5:00 PM**	
	Write it on paper	1 day?	11/16/18 8:00 AM	11/16/18 5:00 PM	
	Meeting with customer	1 day?	11/19/18 8:00 AM	11/19/18 5:00 PM	13
	Define Usecases	6 days?	**11/20/18 8:00 AM**	**11/27/18 5:00 PM**	
	Write main, alternative ser	1 day?	11/20/18 8:00 AM	11/20/18 5:00 PM	
	Identify system	1 day?	11/21/18 8:00 AM	11/21/18 5:00 PM	17
	Draw use cases	1 day?	11/22/18 8:00 AM	11/22/18 5:00 PM	18
	Reviews of usecases	2 days?	11/23/18 8:00 AM	11/26/18 5:00 PM	19
	Meeting with customer	1 day?	11/27/18 8:00 AM	11/27/18 5:00 PM	20
	Development of analysis	13 days?	**11/28/18 8:00 AM**	**12/14/18 5:00 PM**	
	Draw a data model	1 day?	11/28/18 8:00 AM	11/28/18 5:00 PM	
	Creates a model at custom	1 day?	11/29/18 8:00 AM	11/29/18 5:00 PM	24
	Draw behavioural model	1 day?	11/30/18 8:00 AM	11/30/18 5:00 PM	25
	Draw State transaction dia	1 day?	12/3/18 8:00 AM	12/3/18 5:00 PM	26

Name	Duration	Start	Finish	Predecessors
Draw package diagram	1 day?	**12/24/18 8:00 AM**	**12/24/18 5:00 PM**	
Find UML elements	1 day?	12/24/18 8:00 AM	12/24/18 5:00 PM	
Draw Component diagram	1 day?	12/25/18 8:00 AM	12/25/18 5:00 PM	39
Identify Classes and objects	5 days?	12/25/18 8:00 AM	1/1/19 5:00 PM	40
Draw Class diagram	1 day?	1/2/19 8:00 AM	1/2/19 5:00 PM	41
Draw Sequence diagram	1 day?	1/3/19 8:00 AM	1/3/19 5:00 PM	42
Create Design class	1 day?	1/4/19 8:00 AM	1/4/19 5:00 PM	43
Make interface design	2 days?	**1/7/19 8:00 AM**	**1/8/19 5:00 PM**	
Input Interface	1 day?	1/7/19 8:00 AM	1/7/19 5:00 PM	
Output Interface	1 day?	1/8/19 8:00 AM	1/8/19 5:00 PM	47
Review for refinement of I	2 days?	1/7/19 8:00 AM	1/8/19 5:00 PM	
Meeting with customer	1 day?	1/11/18 8:00 AM	1/11/18 5:00 PM	
Database Connectivity	2 days?	1/12/18 8:00 AM	1/15/18 5:00 PM	
Define Classes	1 day?	**1/16/19 8:00 AM**	**1/16/19 5:00 PM**	
Make Objects	1 day?	1/16/19 8:00 AM	1/16/19 5:00 PM	
Interconnect classes	1 day?	**1/17/19 8:00 AM**	**1/17/19 5:00 PM**	
Make relations between cla	1 day?	1/17/19 8:00 AM	1/17/19 5:00 PM	
Coding	19 days?	**1/18/19 8:00 AM**	**2/13/19 5:00 PM**	
Use meaningful variables	3 days?	1/18/19 8:00 AM	1/22/19 5:00 PM	
Give comments after each	1 day?	1/23/19 8:00 AM	1/23/19 5:00 PM	61
Building front end	5 days?	1/24/19 8:00 AM	1/30/19 5:00 PM	62

Name	Duration	Start	Finish	Predecessors
Define Classes	1 day?	**1/16/19 8:00 AM**	**1/16/19 5:00 PM**	
Make Objects	1 day?	1/16/19 8:00 AM	1/16/19 5:00 PM	
Interconnect classes	1 day?	**1/17/19 8:00 AM**	**1/17/19 5:00 PM**	
Make relations between cla	1 day?	1/17/19 8:00 AM	1/17/19 5:00 PM	
Coding	19 days?	**1/18/19 8:00 AM**	**2/13/19 5:00 PM**	
Use meaningful variables	3 days?	1/18/19 8:00 AM	1/22/19 5:00 PM	
Give comments after each	1 day?	1/23/19 8:00 AM	1/23/19 5:00 PM	61
Building front end	5 days?	1/24/19 8:00 AM	1/30/19 5:00 PM	62
Refining front end	10 days?	1/31/19 8:00 AM	2/13/19 5:00 PM	63
Testing the software	1 day?	1/14/19 8:00 AM	1/14/19 5:00 PM	
Dry Run	1 day?	1/15/19 8:00 AM	1/15/19 5:00 PM	66

Figure 1.8.1 Project Plan

1.9 Scope:

The product which is going to be presented in this document is called “Paintio”. This application is planned to be used by any user, to get their paint jobs done by posting the job, then biding with the painter.

Customer and painter will get register to the app. Customer will then select the type of work i.e. paint tasks. Customer fill the form for requirement and can also attach pictures and then post the job by giving the price and days from and to for the paint job.

Painter can see the job and can bid with the user for price, days and give remarks regarding bid.

Customer can accept one from the bidder. After getting the work done the user can select done option and then they can able to rate each other.

1.9.1 **User Functionalities:**

- **Customer**
  - Customer can Register to app
  - Customer can password reset
  - Customer can update profile
  - Customer can select job type.
  - Customer can post job.
  - Customer can view bids.
  - Customer can view bidder’s profile.
  - Customer can accept bid.
  - Customer can select done.
  - Customer can rate painter.
- **Painter**
  - Painter can Register to app
  - Painter can password reset
  - Painter can update profile
  - Painter can view job
  - Painter can bid on a job.
  - Painter can request to unblock.
  - Painter can request to increase work limit.
  - Painter can report customer.
  - Painter can give rating to customer.
- **Admin**

- Admin can view profiles
- Admin can block and unblock profiles.
- Admin can increase the work limit for certain period.

1.9.2 **Roles:**

Customer:

The customer must have **android mobile phone** and knows how to use it. His **literacy level** must be intermediate and must know the basic of English language.

In order to **get register** with the app customer must provide his full name, number, email address, password and have to sign the agreement (i.e. In case of any mishap, app provider will not be responsible and you will provide credentials if needed).

Then Customer can fill the **paint job form** which includes:

- Select (Exterior and Interior)
- Select Job task (Paint, Poitin+Paint, Scrubing+Paint, Poitin+Scrubing+Paint)
- Entity name (e.g. Room, wall, doors etc.)
- Check or Uncheck approximate value.
- Length
- Width
- No of Coating
- Total entities (i.e. Room or wall)
- Add Exclude Area (Width, Height)
- Add pictures

Customer can also add more entities.

In order to post the job the customer has to go through **bidding criteria**.

Customer has to fill the **bidding form** to start the bid which includes:

- Number of days the bid remains on the app.
- Minimum cost for the job.
- No. of days to complete the job (From-To).
- Address.
- Submit bid.

Customer can accept the bid. After accepting the bid, the customer and painter will be able to get contact information of each other (Number, Email and Address).

After the job is done the customer can select done and then rate the painter out of 5.

Customer can also report the Painter with writing the reason of reporting in the comment section.

Painter:

The Painter must have **android mobile phone** and knows how to use it. His **literacy level** must be intermediate.

In order to **get register** with the app Painter must provide his **full name, number, CNIC, email address, password and have to sign the agreement** (i.e. In case of any mishap, app and app provider will not be responsible and you will provide credentials if needed).

Painter can view the paint jobs and in order to view jobs Painter has to **complete the previous job**.

Painter can bid on 4 jobs per week but if he has to increase his **work limit**, he has to request the admin and have to pay admin.

Painter can **bid** on the paint job

- Painter can enter cost
- Painter can enter date (From-to)
- Painter can give remarks.
- Painter can then send the bid.

After the bid is accepted by the Customer the customer and painter/contractor will be able to get contact information of each other (Number, Email and Address).

Painter can request to **increase the Work Limit** to admin.

Painter can request to **Unblock**.

Painter can give rating to the customer.

Painter can report customer with writing the reason of reporting in the comment section.

1.10 DEFINATION, ACRONYMS AND ABBRIVIATIONS:

Acronyms	Abbreviations
SPMP	Software Project Management Plan
OTP	One Time Password
WL	Work Limit

Table 1.10.1 Definition and Abbreviations

2 Chapter: Requirements Gathering and Analysis

2.1 OVERALL DESCRIPTION:

I have defined product features and operations performed by the application. User first have to register them, whenever they visit my app they have to log in first with their email and password to perform operation. For registration a form will appear, they have to fill in the form comprising their complete information.

2.2 PRODUCT PERSPECTIVE:

Paintio is totally an independent application. It has three active actors i.e. Painters, admin and customer. Whenever they register themselves a password is created, which is compulsory for log in. Admin can view profiles. Admin can block and unblock profiles. Admin can increase work limit of painter for certain period.

2.2.1 **System Interfaces:**

Paintio is an Android based application. It is used by the Painters and Customers who are registered at our app to find jobs. It uses a server which store data in database. User can use tablet or android phones to access this application and after logging in they can perform operations.

2.2.2 **User Interfaces:**

User interface defines how a user can interact with the application so in order to use this application to perform functions, user requires an email and password which must be authenticated using OTP. A first-time user of the mobile application should see the log-in and registration page when he/she opens the application. If the user has not registered, he/she should be able to do that on the log-in page. Every user should have a profile page where they can update their profile.

2.2.3 **Software Interfaces:**

Our application can interact with Google Location Services in order to perform a filtered search for jobs or applicants in a particular area. Similarly it can interact with the database in order to get the information of the jobs of interest or required applicants.

2.2.4 **Hardware Interfaces:**

Since the mobile application don't have any designated hardware, it does not have any direct hardware interfaces.

2.2.5 **Communication Interfaces:**

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems

2.3 PRODUCT FUNCTIONS:

- This software helps the painter to find and bid for jobs.
- Customer will be able to post paint jobs for bidding.
- Painter will be able to bid for a particular paint job.
- System can also perform the following function:

1) Register
2) Login
3) Update profile
4) Password Reset
5) View profile
6) Create Task
7) Post Job
8) Place Bid
9) Submit Remarks
10) Accept bid
11) Rate and Comment
12) View Jobs
13) Request to unblock
14) Request to increase WL

15) View profile

16) Unblock profile

17) Increase WL

2.4 USER CHARACTERISTICS:

Users are assumed to have basic knowledge of the mobiles and Internet browsing, they must have the knowhow of English language and can read and write it. The proper user interface, online help and the guide to install and maintain the application must be sufficient to educate the users on how to use the application without any problems. The user interfaces will be highly interactive and user friendly.

2.5 ASSUMPTIONS AND DEPENDENCIES:

The users have sufficient knowledge of mobile/tablets. Mobile should have Internet access and Internet server capabilities. The users know the English language, as the user interface will be provided in English. The product can access the database

2.6 SPECIFIC REQUIREMENTS:

There are some functional and non functional requirements for the software application.

2.6.1 **Functional Requirements:**

System can validate all the correct emails and create the profiles. It can allow to post jobs from the registered users only. System shall allow admin to manage the profiles of painter and customer. It can allow all users to modify their data whenever needed and permit them to post jobs and place bids. System does not store any data regarding problems.

2.6.2 **Non Functional Requirements:**

Non functional requirements are basically software system attributes which are not directly involved in the system. Following are some of the non functional requirements:

2.6.2.1Reliability:

This application is reliable and responds well to all valid queries, system process the query and display the related result within no time. System will only send non ambiguous results.

2.6.2.2Maintainability:

For any type of system crash due to virus or any operating system issue there is a backup of data at the database to avoid data loss.

2.6.2.3Security:

As far as security requirement is concerned as there are only registered users so the one who is not a user cannot see the activities going on and necessary data is being showed at profile at first after choosing and contacting the applicant further data can be shown according to privacy policy.

2.6.2.4Availability:

This is a 24/7 system. Whenever you have the internet access you can perform the desired operations so if a user can make query, database can send the results.

2.6.2.5Portability:

As it is an android application so it will run only on Android phones only.

2.6.2.6Performance:

Load time for user interface screens is no longer than some milliseconds. Log in verification shouldn't take more than 1 or 2 seconds similarly for search queries system will display the related result in no time.

2.7 Use Case Diagram

Use Case Diagram is the representation of the interaction between user and system and depicting the specification of the use case. A use case diagram depicts the actors, use cases and the relation among them. Use Case details include description, primary actors, stake holders, pre-conditions, post-conditions and success scenario.

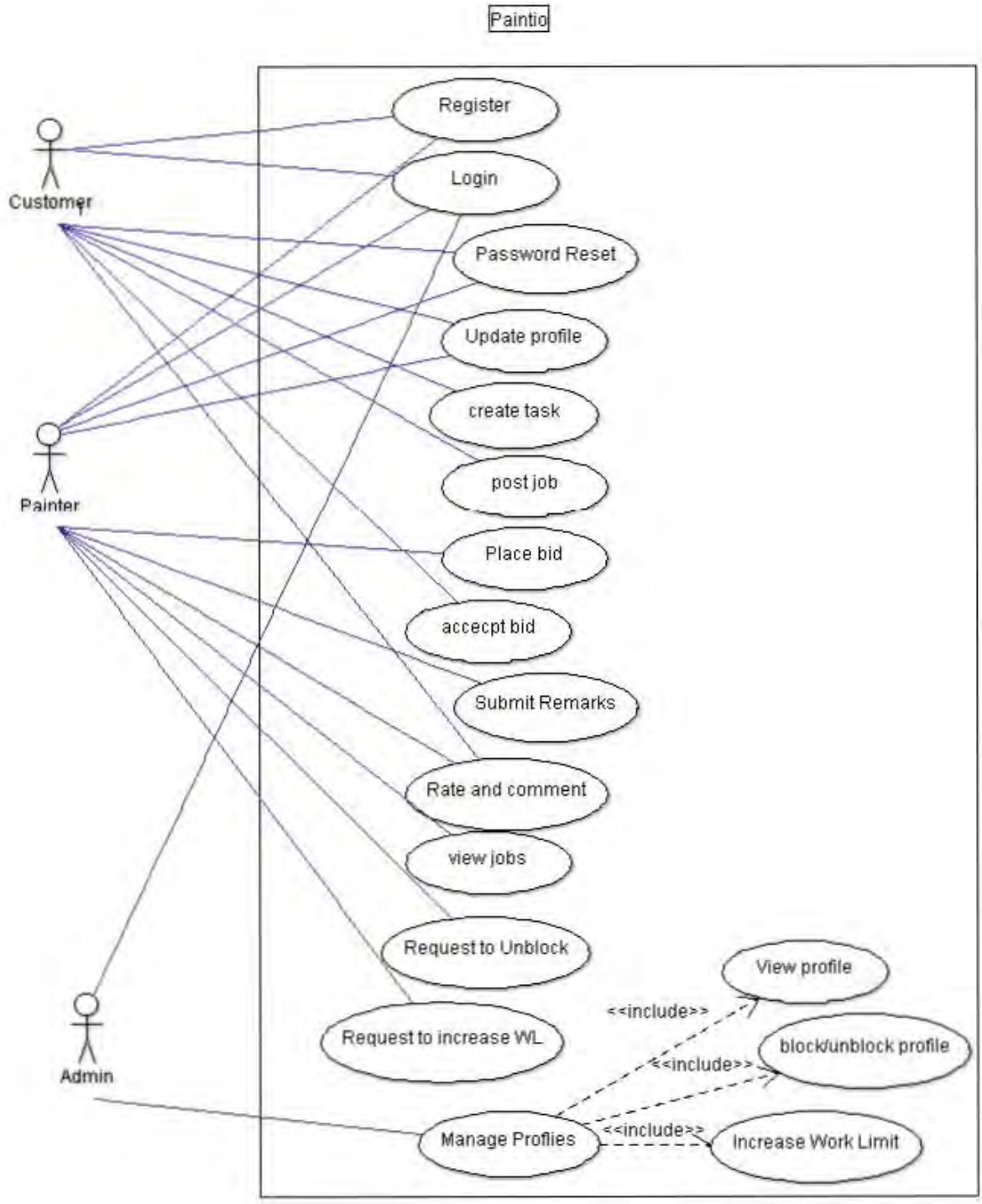

Figure 2.7.1 Use Case Diagram

2.8 Use Case Description:

A use case diagram is a graphic depiction of the interaction among the elements of the system. A list of use cases mentioned in diagram are described below in detail so that we can look more precisely how a user can interact with system. There are three main actors of this system.

A primary user is the one whose tasks are getting fulfilled by the system while a supporting actor in a use case in an external actor that provides a service to the system under design. It might be a high-speed printer, a web service, or humans that have to do some research and get back to us. [2]

2.8.1 **UC:1 LOG IN**

UC:1	LOG IN
Primary actor	Admin, Painter and Customer
Pre Condition	User must be registered
Post condition	User successfully logged in.
Main success scenarios	1. User selects log in option. 2. System asks to enter email and password 3. User enters valid email and password correctly. 3. System will check email and password. 4. User successfully logged in.

Table 2.8.1 Login

2.8.2 **UC:2 Register :**

UC:2	Register Painter and Customer
Primary actors	Painter and Customer

Pre condition	User must have his/her name , Phone, Cnic.
Post condition	User Successfully registered
Main success scenarios	1. User selects register as painter/customer option. 2. Form appears 3. System asks to enter required information. 2. Painter/Customer fills in the required information, 3. Painter/Customer submits form. 4. Painter/Customer successfully registered,
Alternate scenarios	System fails at any time 1. System gets back to previous page and discards all information. 3.a) Painter/Customer skips any information or entered incorrect information. 1. System asks to fill all information.

Table 2.8.2 Register

2.8.3 **UC:3 Password Reset:**

UC:3	Password Reset
Primary actors	Painter/Customer
Pre condition	Painter/Customer must be registered
Post condition	Painter/Customer successfully reset their password
Main success scenarios	Painter/Customer selects the password reset

	option. Painter/Customer enters the old password. Painter/Customer enters the new password. Painter/Customer selects the reset button
Alternate scenarios	System fails at any time System gets back to previous page and discards all information. A) Painter/Customer forgets the old password. System will send the password to the registered number upon selecting forget password option 1. System asks to fill all information.

Table 2.8.3 Password Reset

2.8.4 **UC4: Update Profile:**

UC:4	Update profile
Primary actors	Painter and Customer
Pre condition	User must be registered and logged in
Post condition	Profile will be updated successfully
Main success scenarios	1. User opens the profile. 2. Relevant page/ form are displayed by the system to the user. 3. User opens the form and fills in the information. 4. System prompts the message that profile successfully updated.

Alternate scenarios	System fails at any time 1. System erases all the changes previously made. 3. User doesn't fills in the correct/ complete information. 3. A) Form will not be submitted and ask to fill it completely.

Table 2.8.4 Update Profile

2.8.5 **UC:5 Create Task :**

UC:5	Create Task
Primary actors	Customer
Pre condition	Customer must be registered and logged in.
Post condition	Task will be created successfully.
Main success scenarios	1. Customer selects the create task option. 2. System displays interior/exterior option. 3. Customer selects interior/exterior option. 4. System displays the options i.e. Paint, Paint+Poitin, Paint+Scrubbing andPaint+Poitin+Sucrubbing option. 5. Customer selects the option. 6. System displays the form 7. Customer fills the form and selects the create task option.
Alternate Scenarios	System fails at any time. 1. it rolls back all the changes made 3.a) User doesn't fills in the correct/ complete

	information. a) system asks to fill it again properly.

Table 2.8.5 Create Task

2.8.6 **UC:6 Post Job :**

UC:6	Post Job
Primary actors	Customer
Pre condition	Customer must be registered and is logged In.
Post condition	Customer posted the job successfully.
Main success scenarios	1. Customer creates the task. 2. Customer fills the bidding form. 3. Customer selects the post job option.
Alternate Scenarios	System fails at any time. 2. it rolls back all the changes made 4.a) User doesn't fills in the correct/ complete information. a) system asks to fill it again properly.

Table 2.8.6 Post Job

2.8.7 **UC:7 Place Bid**

UC:7	Bid
Primary actors	Painter
Pre condition	User must be registered and logged in. There should be some job to bid on
Post condition	Painter successfully bid on the job.
Main success scenarios	1. Painter selects the job 2. Painter adds the price. 3. Painter adds the days. 4. Painter submits the bid.
Alternate scenarios	System fails at any time 1. Rolls back all the changes.

Table 2.8.7 Place Bid

2.8.8 **UC:8 Accept bid**

UC:8	Accept bid
Primary actors	Customer
Pre condition	User is logged in and Search is done.
Post condition	Applicant's pofile will be displayed.
Main success scenarios	1. User clicks on the profile icon. 2. System displays the profile and relevant information.
Alternate scenarios	System fails at any time 1. System rolls back the changes being made

	Click again on profile icon.

Table 2.8.8 Accept bid

2.8.9 **UC: 9 Rate and comment**

UC:9	Rate and Comment
Primary actors	Painter and Customer
Pre condition	Painter and Customer is logged in and profile is viewed/displayed.
Post condition	User rate and comment successfully.
Main success scenarios	1. Painter/Customer gives the rating. 2. Painter/Customer write the comments. 3. Painter/Customer select the submit option.
Alternate scenarios	System fails at any time 1. profile shuts down 2. Interface might not be displayed due to slow connection.

Table 2.8.9 Rate and Comment

2.8.10 **UC: 10 View Jobs**

UC:10	View jobs
Primary actors	Painter
Pre condition	Painter must be logged on.
Post condition	Painter successfully viewed the jobs.

Main success scenarios	1. Painter goes to the available job. 2. Painter can click on any job to view. 3. Painter views the job.
Alternate scenarios	System fails at any time 2. Rolls back all the changes. 1.a) Log in again and view jobs again.

Table 2.8.10 View Jobs

2.8.11 **UC: 11 Request to unblock:**

UC:10	Request to unblock
Primary actors	Painter/Customer
Pre condition	User must be logged in.
Post condition	Painter and customer will request to unblock successfully.
Main success scenarios	1. User clicks on the request to unblock option. 2. System sends the request.
Alternate scenarios	System fails at any time 2. System rolls back the changes being made.

Table 2.8.11 Request to Unblock

2.8.12 **UC:12 Request to increaseWL:**

UC:11	Request to increase WL
Primary actors	Painter

Pre condition	Painter must be logged in and his work limit much be expire.
Post condition	Request will be sent successfully.
Main success scenarios	1. Painter visits the Work limit interface. 2. Painter selects the work limit. 3. Painter submits the request.
Alternate scenario	System fails at any time 1. Rolls back all the changes made.

Table 2.8.12 Request to Increase WL

2.8.13 **UC:13 Manage Profiles:**

UC:12	Manage Profiles
Primary actors	Admin
Pre condition	User must be logged in.
Post condition	Admin manages the profiles successfully.
Main success scenarios	1. Admin select the profile and view. 2. Admin unblocks the profile. 3. Admin select the work limit and then select the increase work limit option.
Alternate scenarios	System fails at any time 1. Rolls back all the changes made by admin.

Table 2.8.13 Manage Profiles

2.9 DOMAIN MODEL:

Domain model is an object model of problem domain. It is based on real world concepts and their relationships, that is used to identify the relationships among all the entities within the scope of problem domain

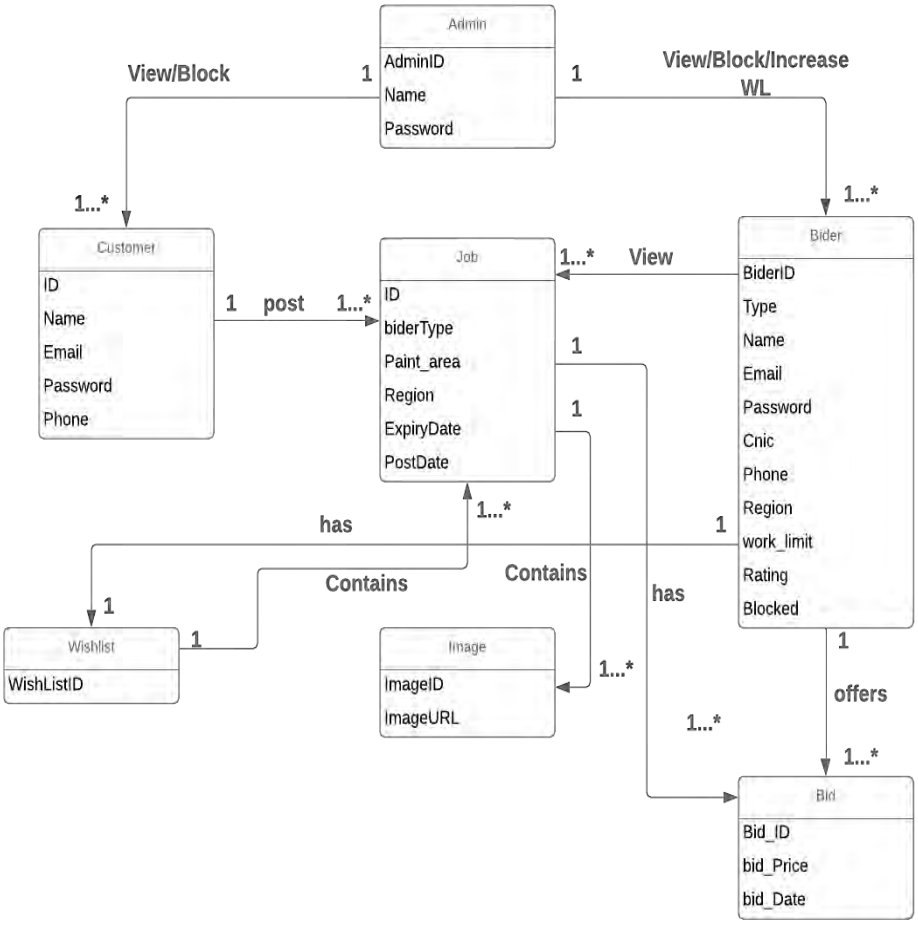

Figure 2.9.1 Domain Model

3 Chapter: Software Design Description

3.1 Design Overview

In the design overview, we can describe the system at architecture level and the complete structure of the system. In this we can explain how the user interacts with the system. In the class diagram, we can show that how the classes can interact with each other define the relationship. Sequence diagram is an interaction diagram, that represents the sequence of messages interacting with objects**.** [3]

3.2 SYSTEM SEQUENCE DIAGRAM:

A system sequence diagram illustrates input and output events related to our system. System is treated as a black box and the emphasis of the diagram is events that are generated by system for a particular scenario of use-case.

3.2.1 **Login**

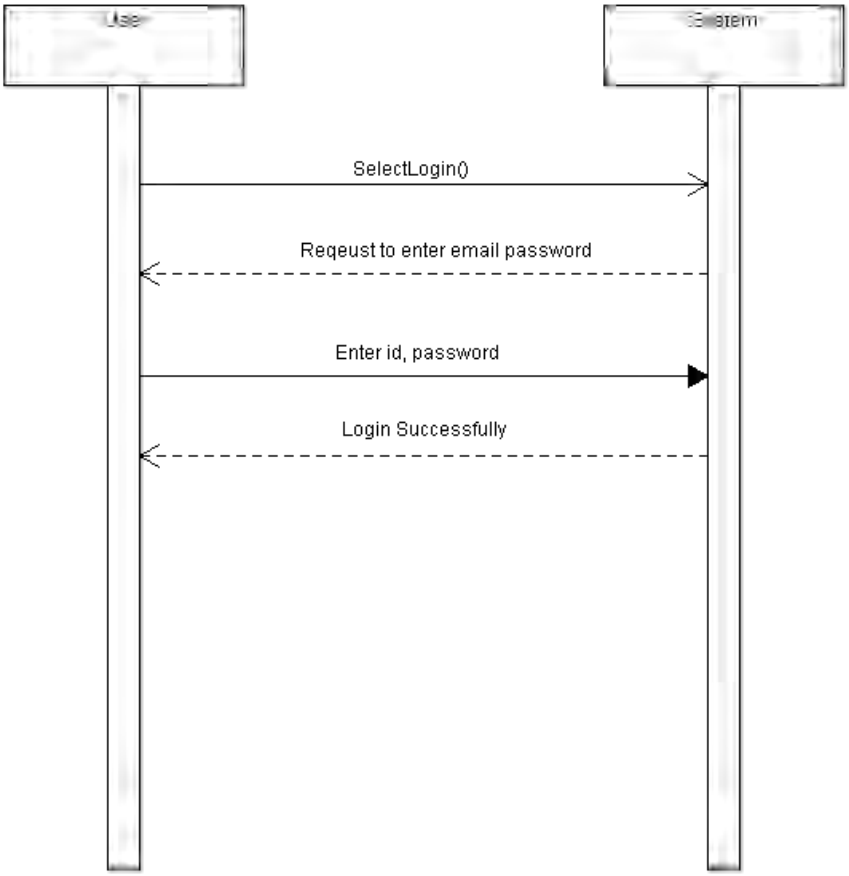

Figure 3.2.1 Login

3.2.2 **Register**

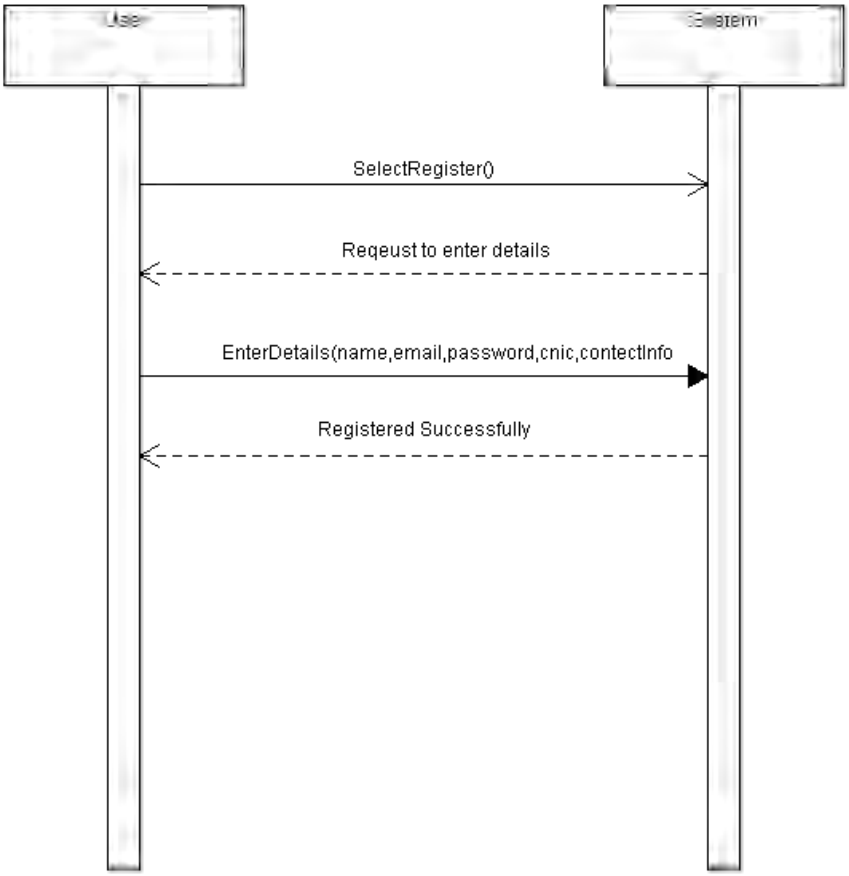

Figure 3.2.2 SSD Register

3.2.3 **Update Profile**

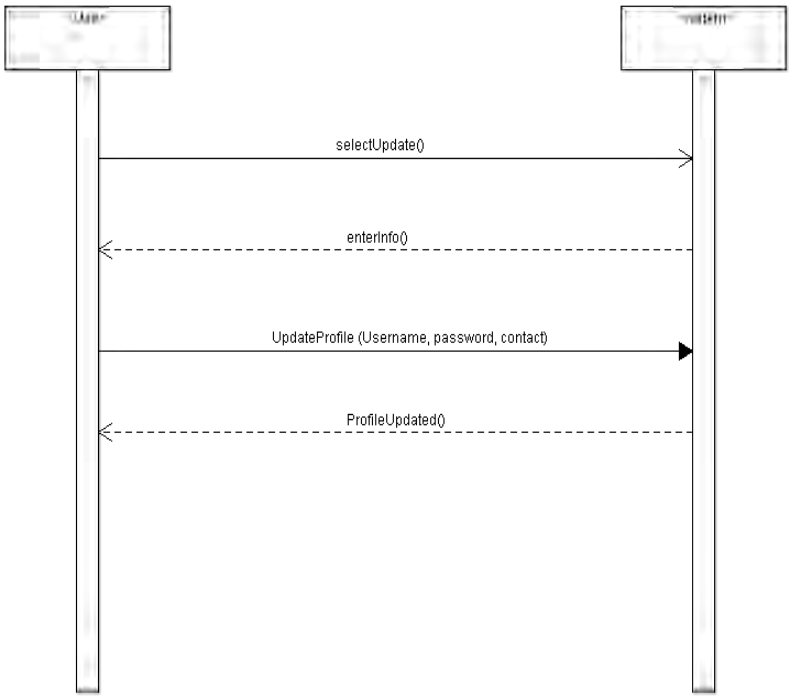

Figure 3.2.3 SSD Update Profile

3.2.4 **Create Task**

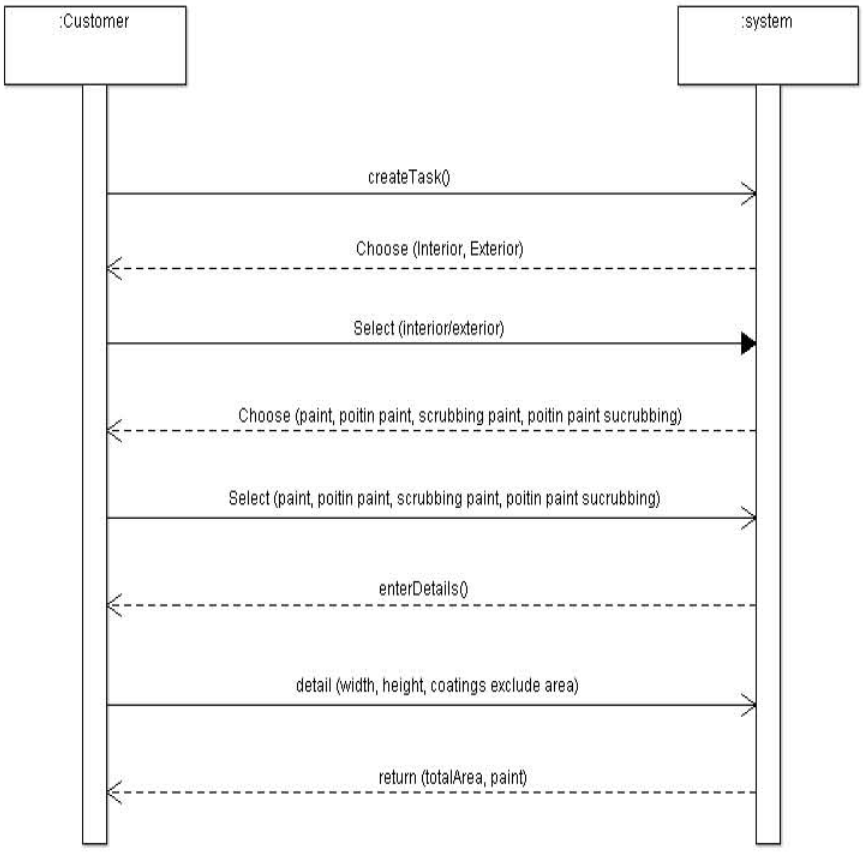

Figure 3.2.4 SSD Create Task

3.2.5 **Post Job**

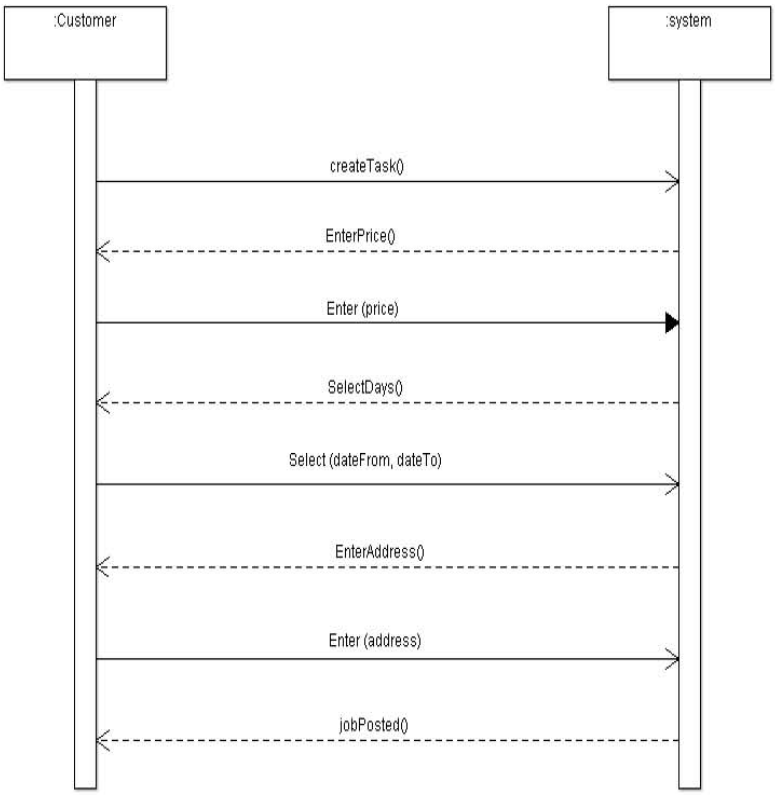

Figure 3.2.5 SSD Post Job

3.2.6 **Place Bid**

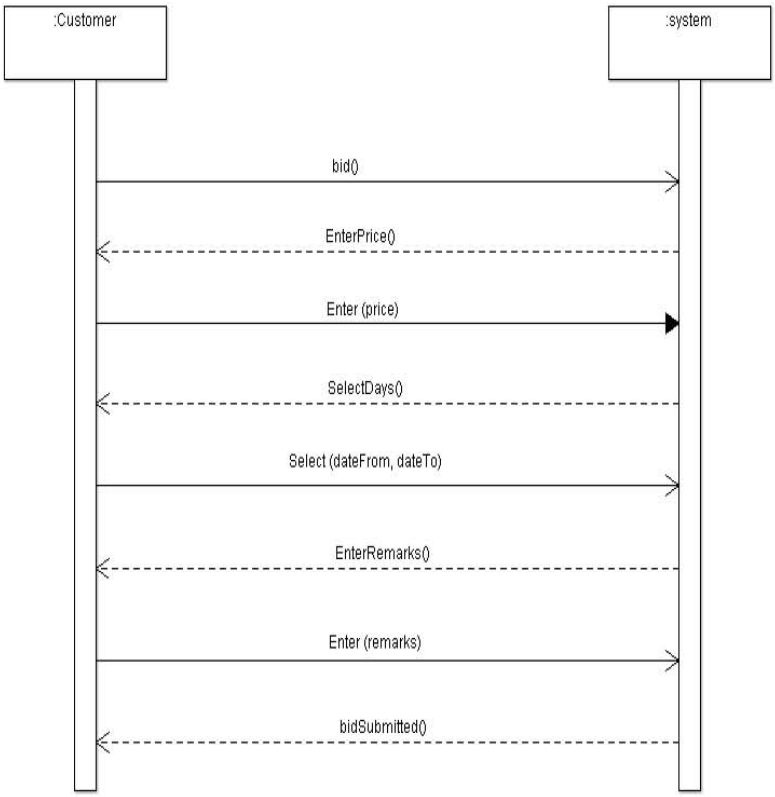

Figure 3.2.6 SSD Place Bid

3.2.7 **Rate And Comment**

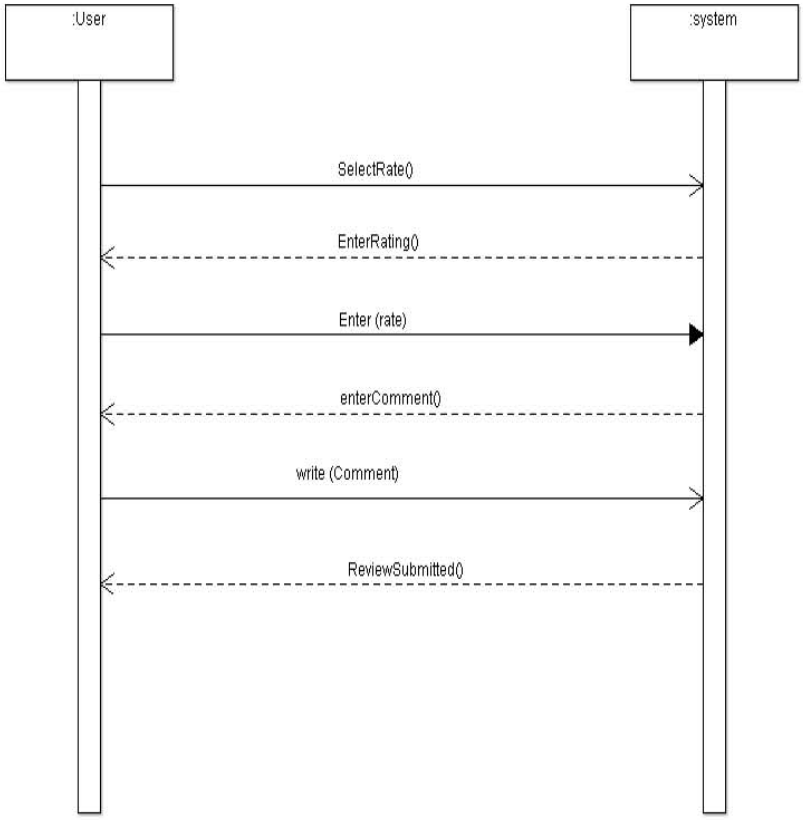

Figure 3.2.7 Rate And Comment

3.2.8 **Increase Worklimit:**

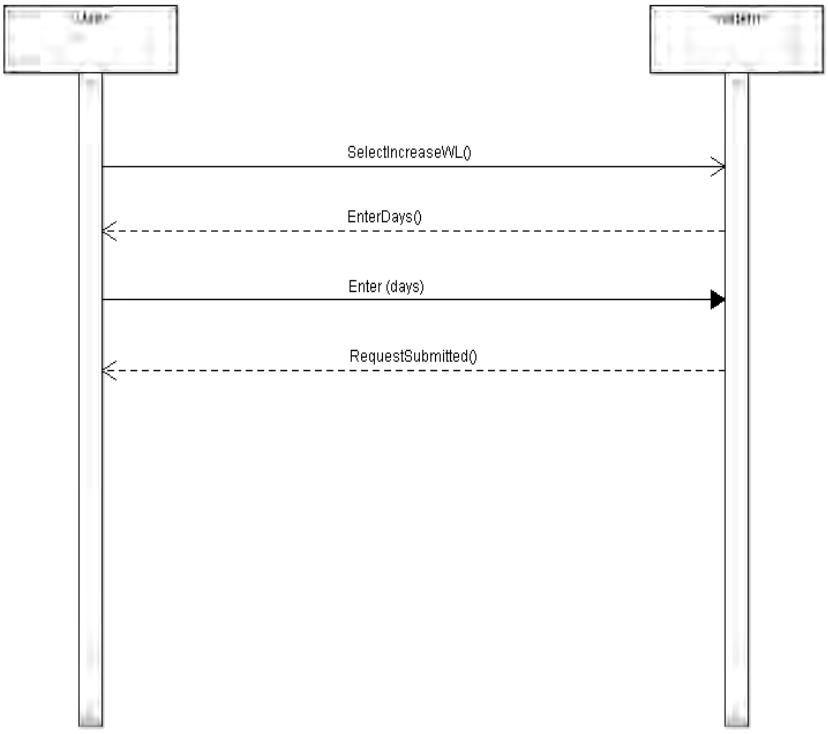

Figure 3.2.8 SSD Increase Worklimit

3.3 Class Diagram:

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. [4]

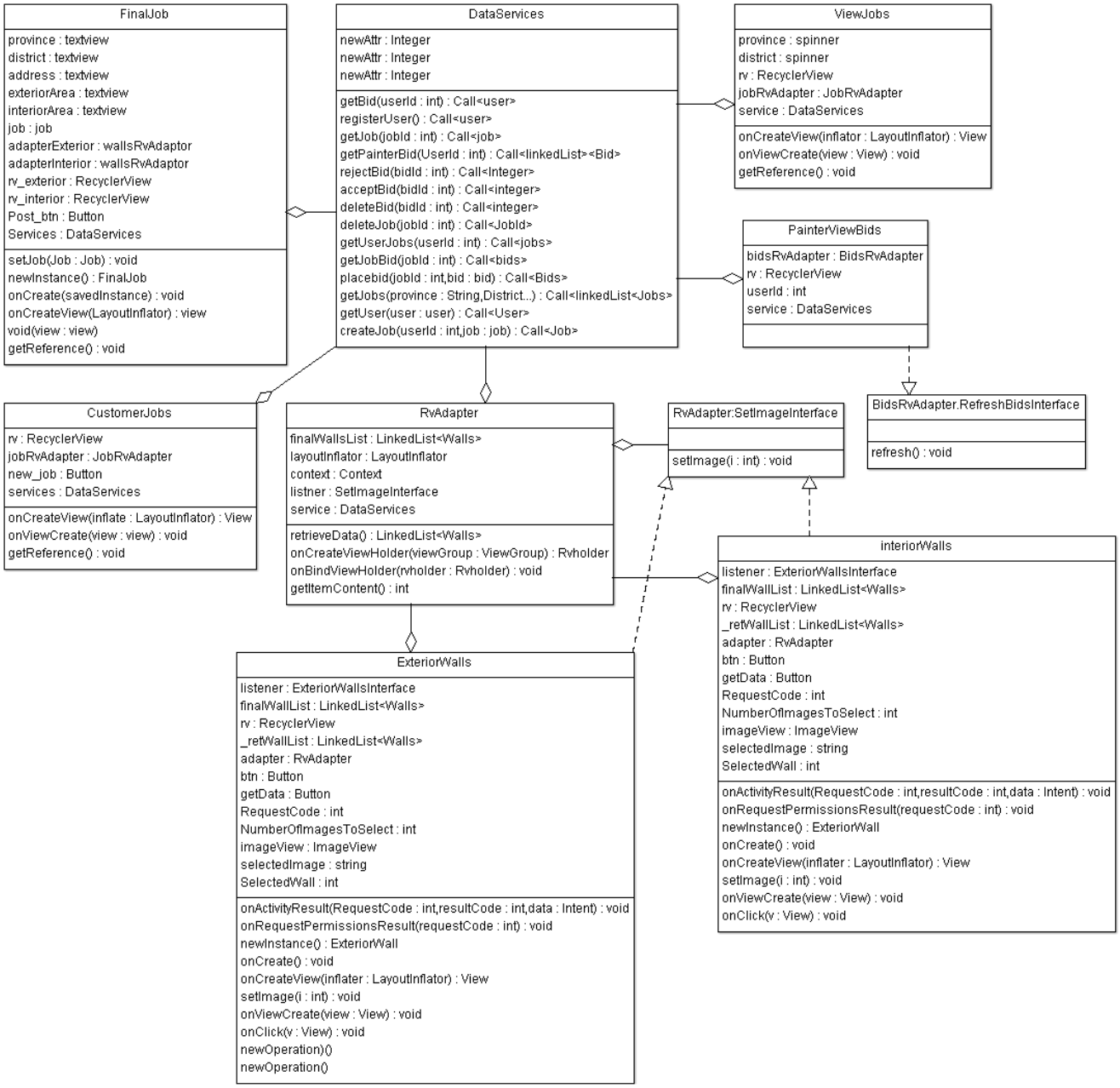

Figure 3.3.1 Class Diagram

3.4 Entity Relationship Diagram:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

The ERD draw below is in Crow's foot notation.

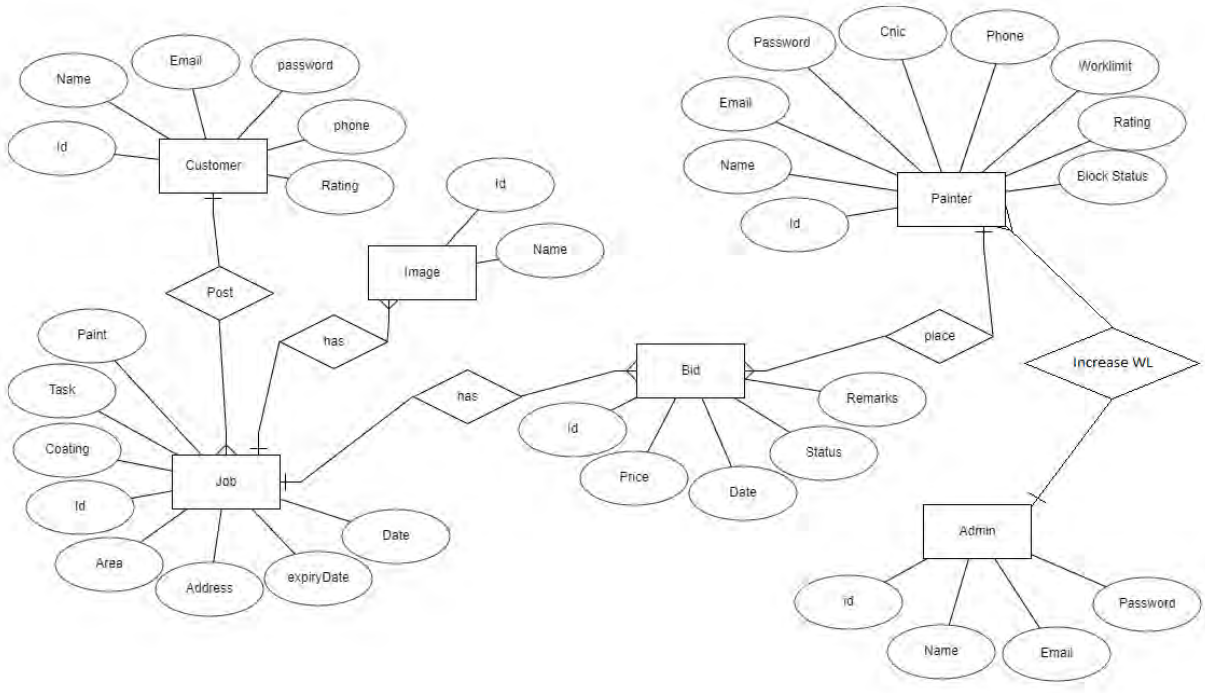

Figure 3.4.1 ERD

3.5 Architecture Diagram

Architecture Diagram is used to represent the components of system and interaction between them. System under discussion is based on “Three Tier” Architectural pattern. Three-tier architecture is a software architecture pattern in which the user interface (presentation), functional process logic (business rules), computer data storage and data access are developed and maintained as independent modules Interacting between components of system is shown in diagram. Double arrows represents the interaction from both sides.

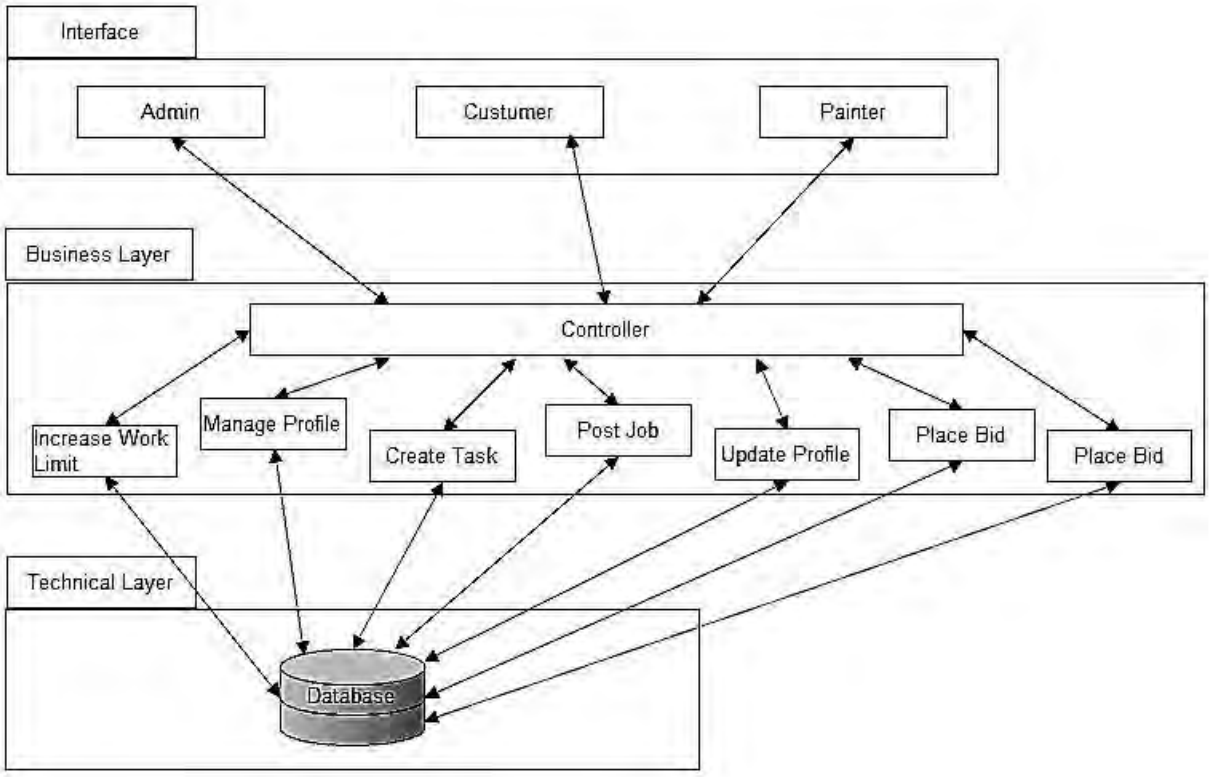

Figure 3.5.1 Architecture Diagram

3.6 Design Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

3.6.1 **Log In**

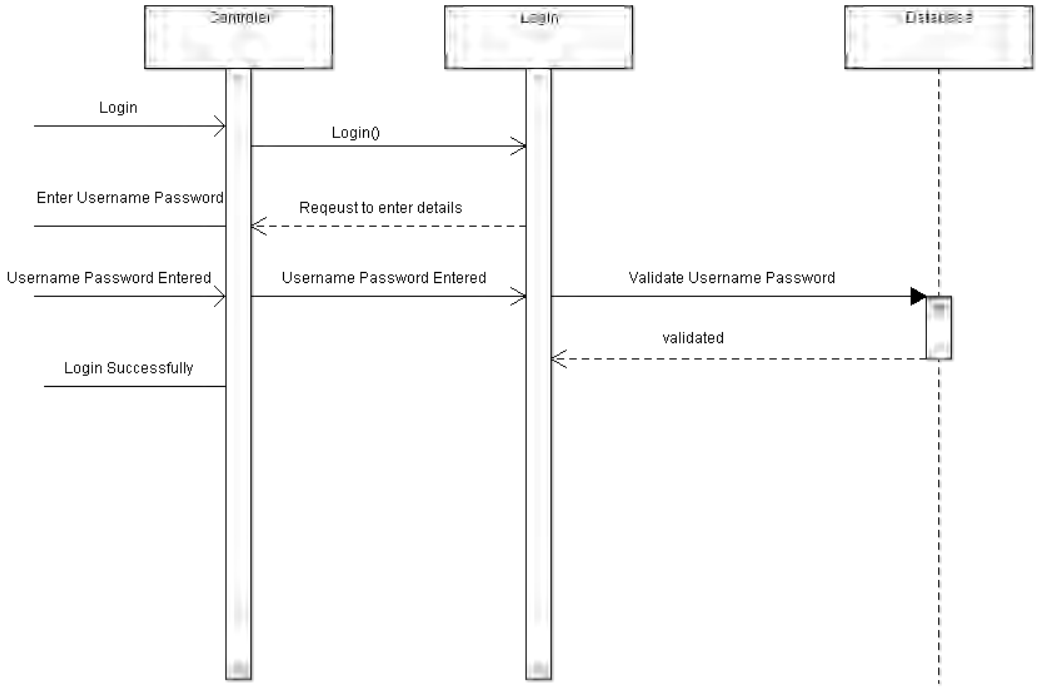

Figure 3.6.1 SD Login

3.6.2 **Register:**

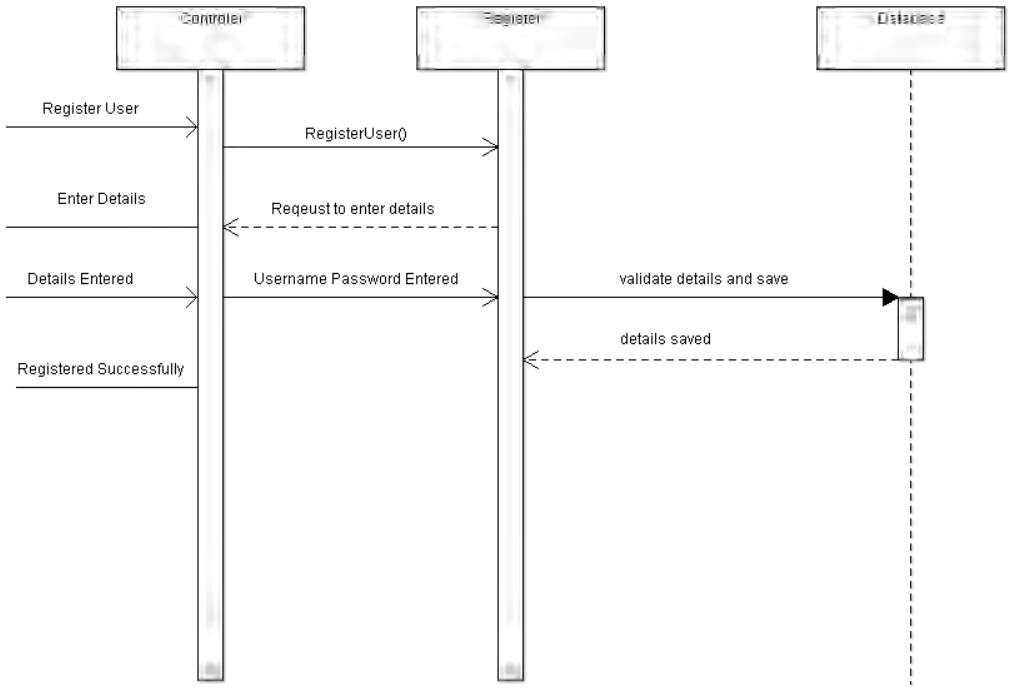

Figure 3.6.2 SD Register

3.6.3 **Add Job**

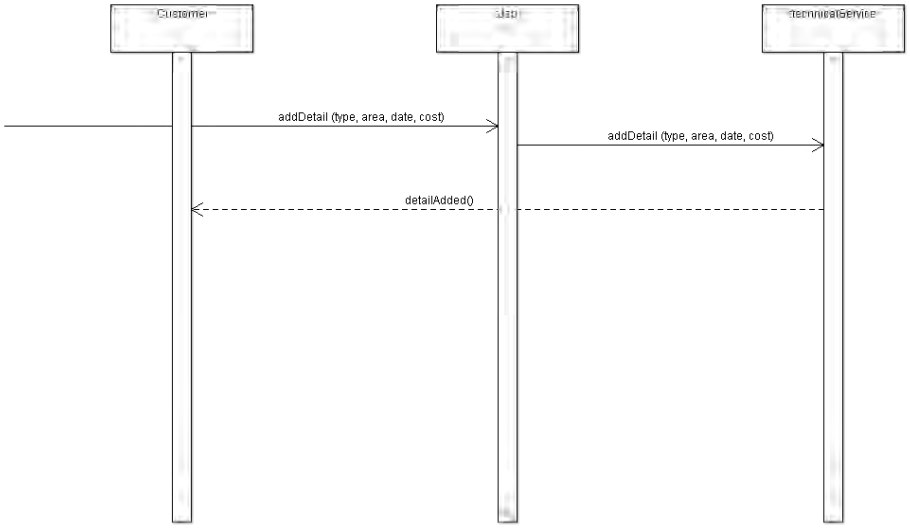

Figure 3.6.3 SD Add Job

3.6.4 **Bidding:**

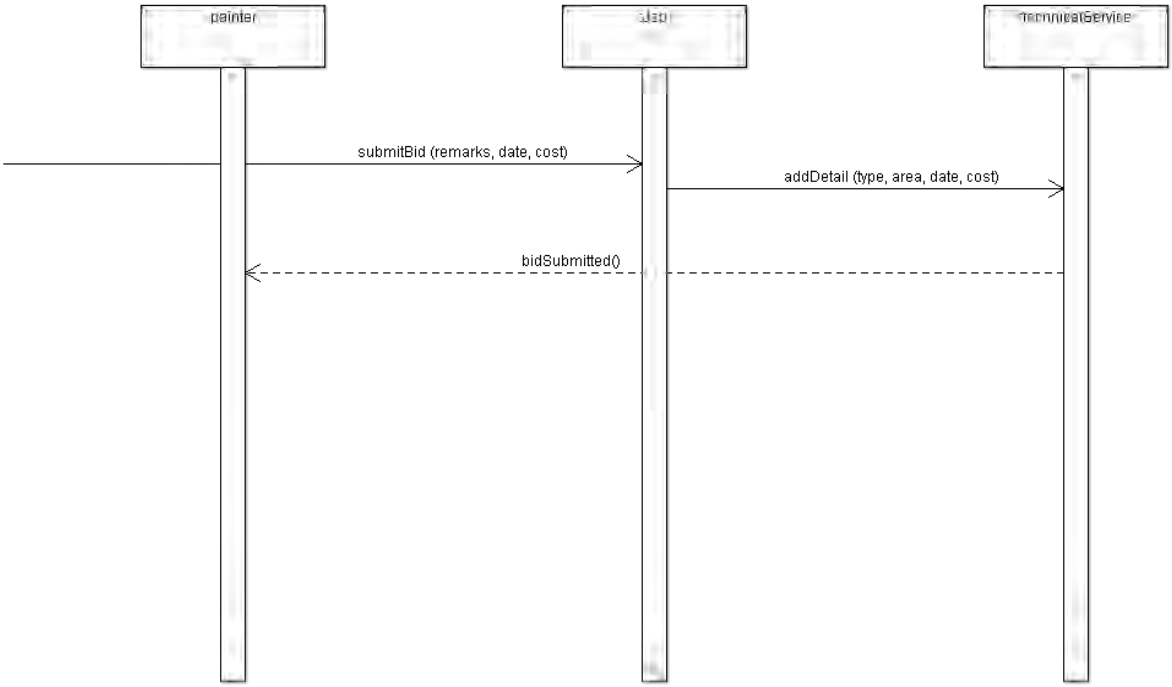

Figure 3.6.4 SD Bidding

3.6.5 **Increase WL**

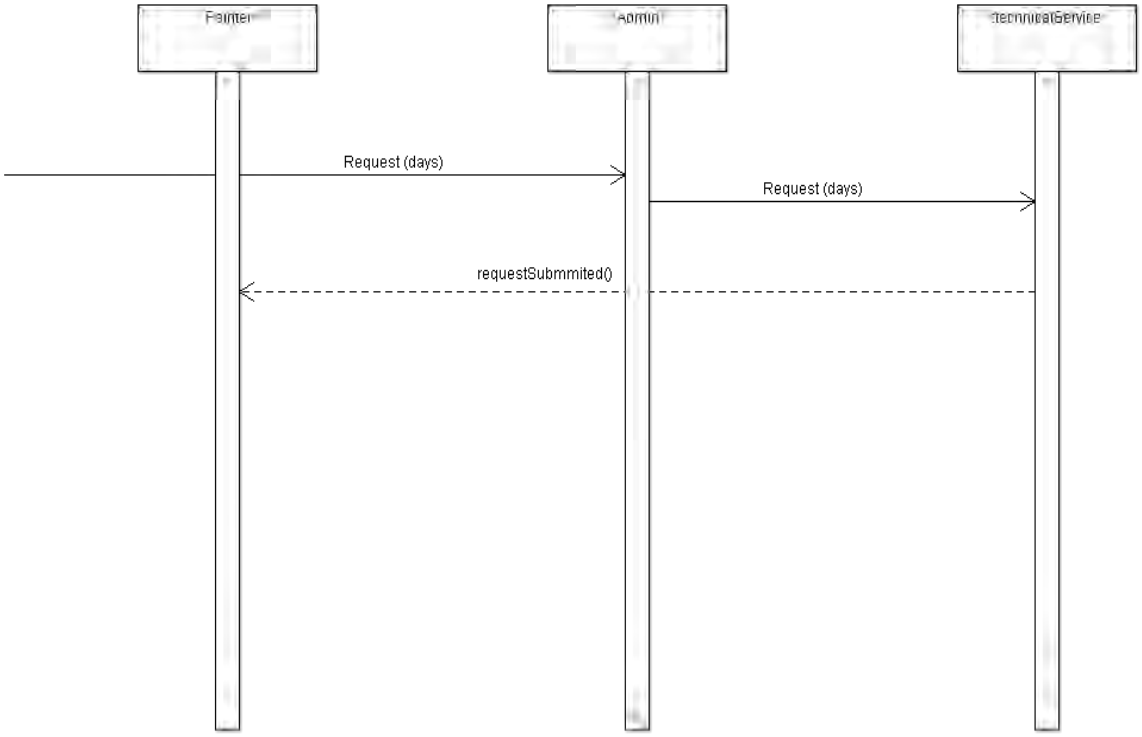

Figure 3.6.5 SD Increase WL

4 Chapter: Design Interface

4.1 Interface design

4.1.1 **Simple and Appealing**

The interface is simple to use, a user can also use it very easily and efficiently to get there required information. The images are used that makes application appealing.

4.1.2 **Responsive**

The interface is responsive this means it can adjust its size according to device display Like (landscape, portrait) etc.

4.2 User Interface Design:

User interface designs show that how the end user will interact with the system through the screens to perform the various task of the application. User interface allows a user to connect with a given technology. Many different kinds of user interfaces come with various devices and software programs. User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. User interface is part of software and is designed such a way that it is expected to provide the user insight of the software.

4.2.1 **Log in:**

This interface depicts how the user can log in to the system. User enters id and password.

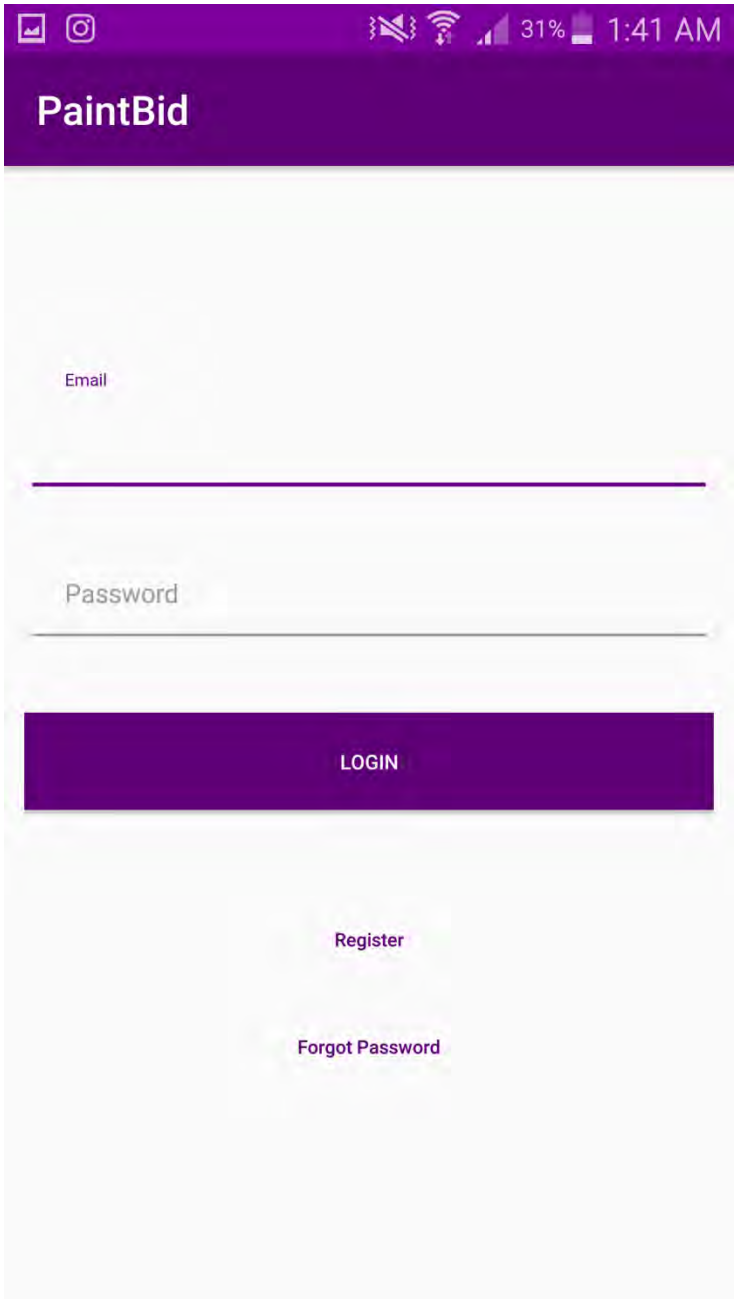

Figure 4.2.1 UI Login

4.2.2 **Register:**

This interface shows how the painter and customer can be registered. It has the following fields.

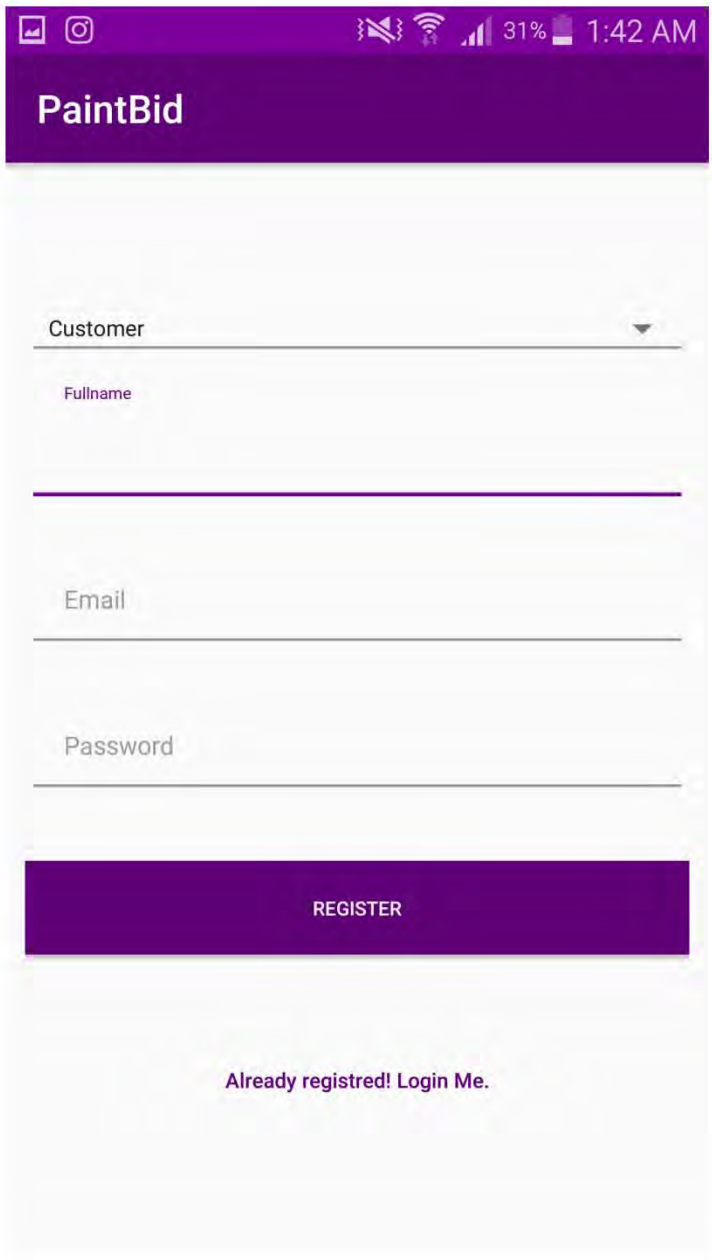

Figure 4.2.2 UI Register

4.2.3 **Select Paint Type:**

This interface shows the interaction when customer have to select the paint type i.e. interior or exterior.

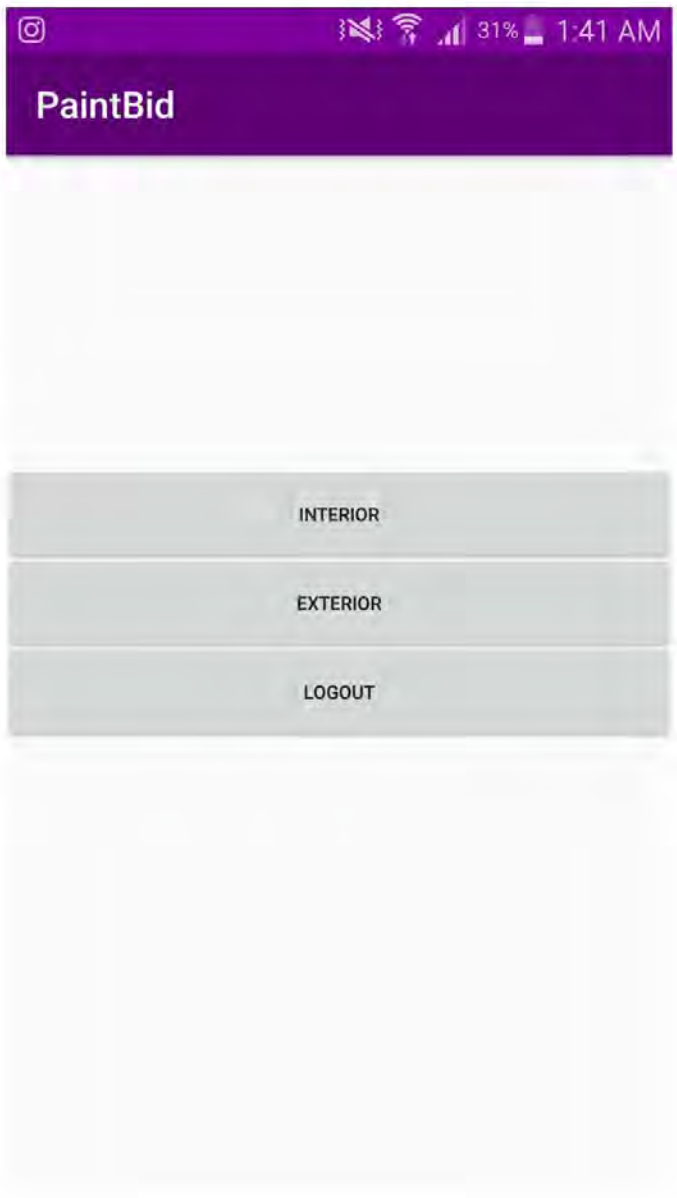

Figure 4.2.3 UI Select Paint Type

4.2.4 **Select Task:**

This interface depicts the interaction when customer has to select the task for his paint job.

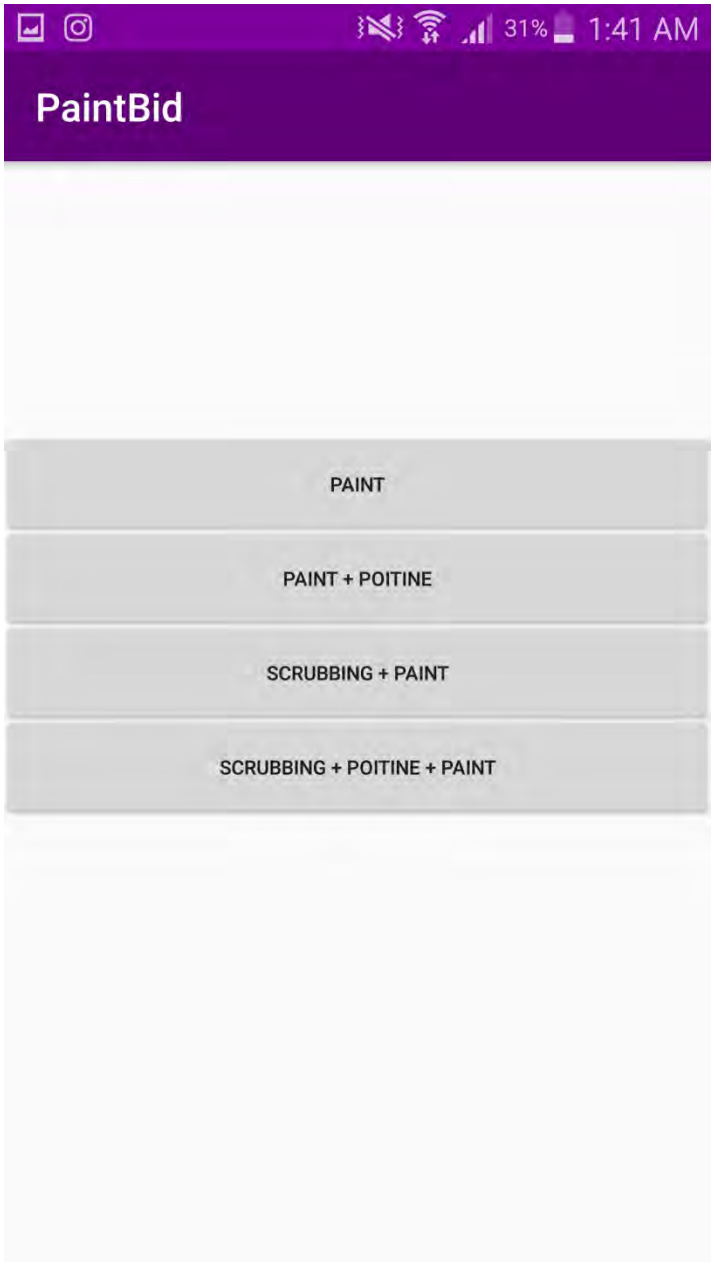

Figure 4.2.4 UI Select Task

4.2.5 **Task Form:**

This interface depicts the how to fill the task field of paint job. It consists of the following information regarding paint job.

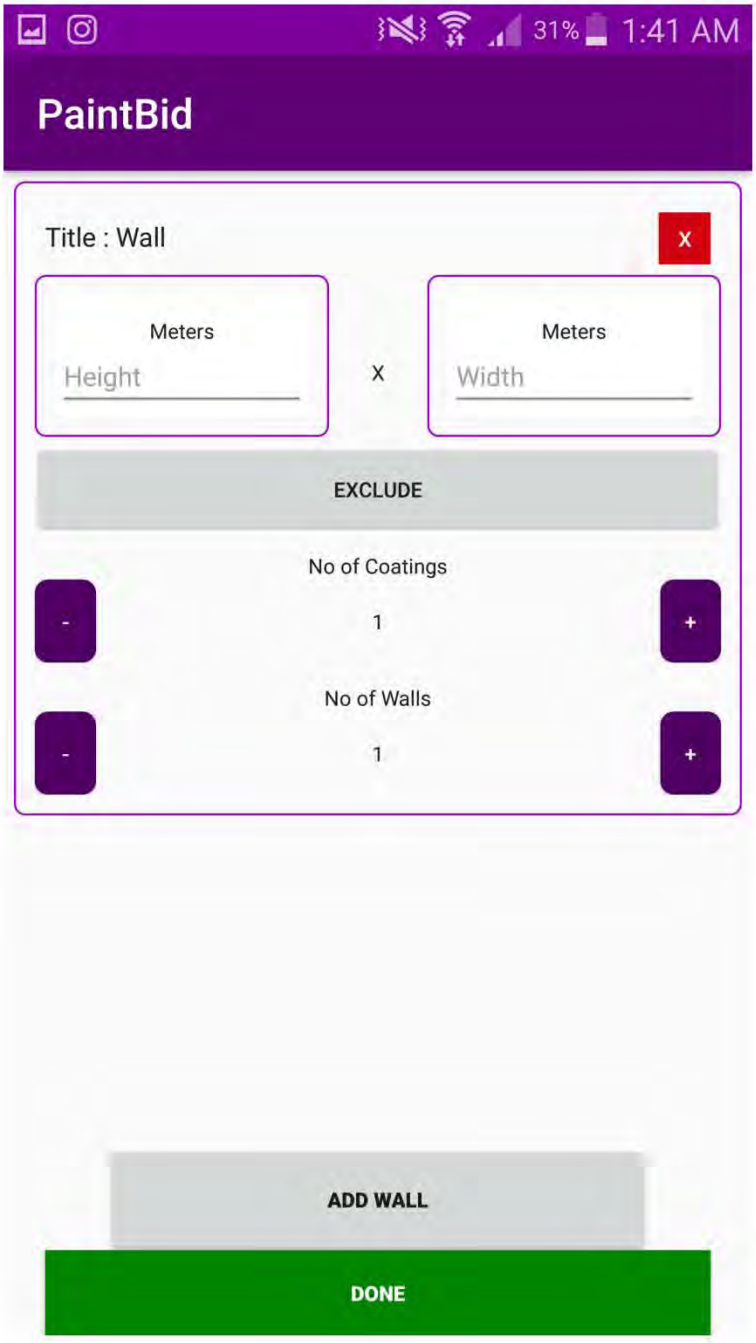

Figure 4.2.5 UI Task Form

4.2.6 **Paint Task Form**

This interface depicts the how to fill the task field of paint job. It consists of the following information regarding paint job. It also consists of the exclude area.

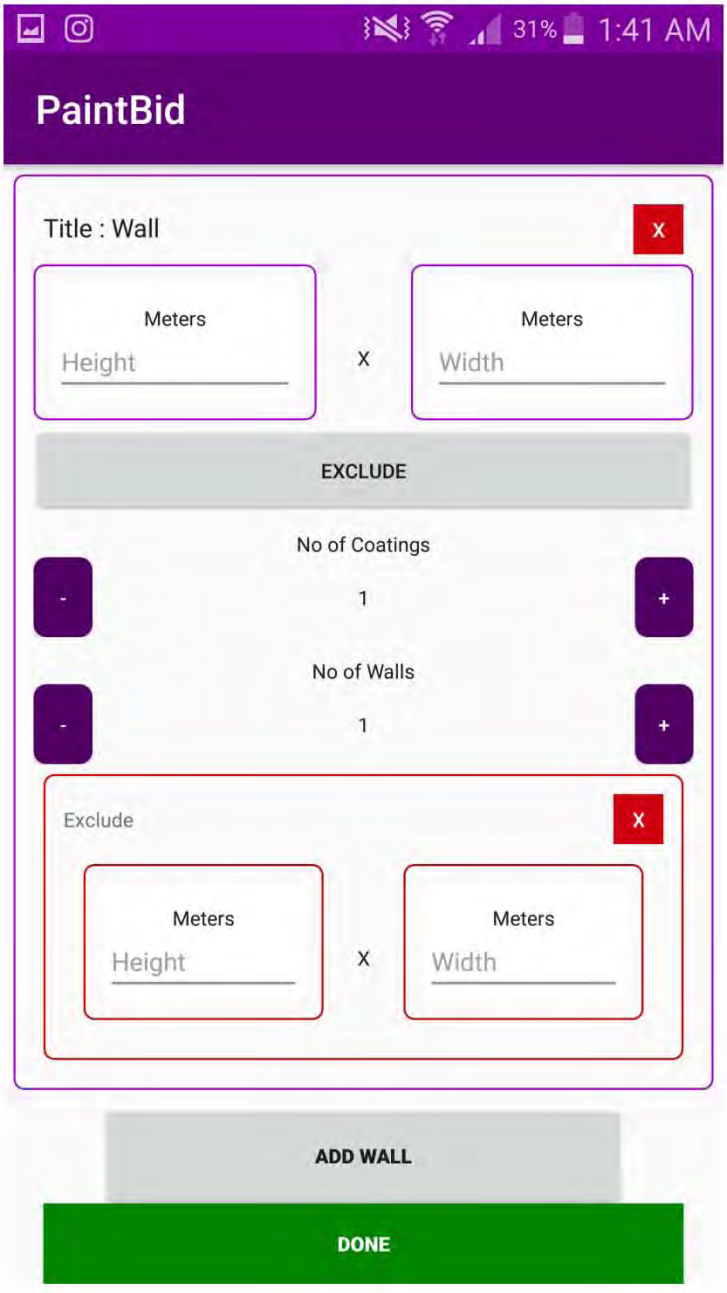

Figure 4.2.6 UI Paint Task Form

4.2.7 **View Jobs**

This interface depicts when the painter selects the region to search the bids in that particular region.

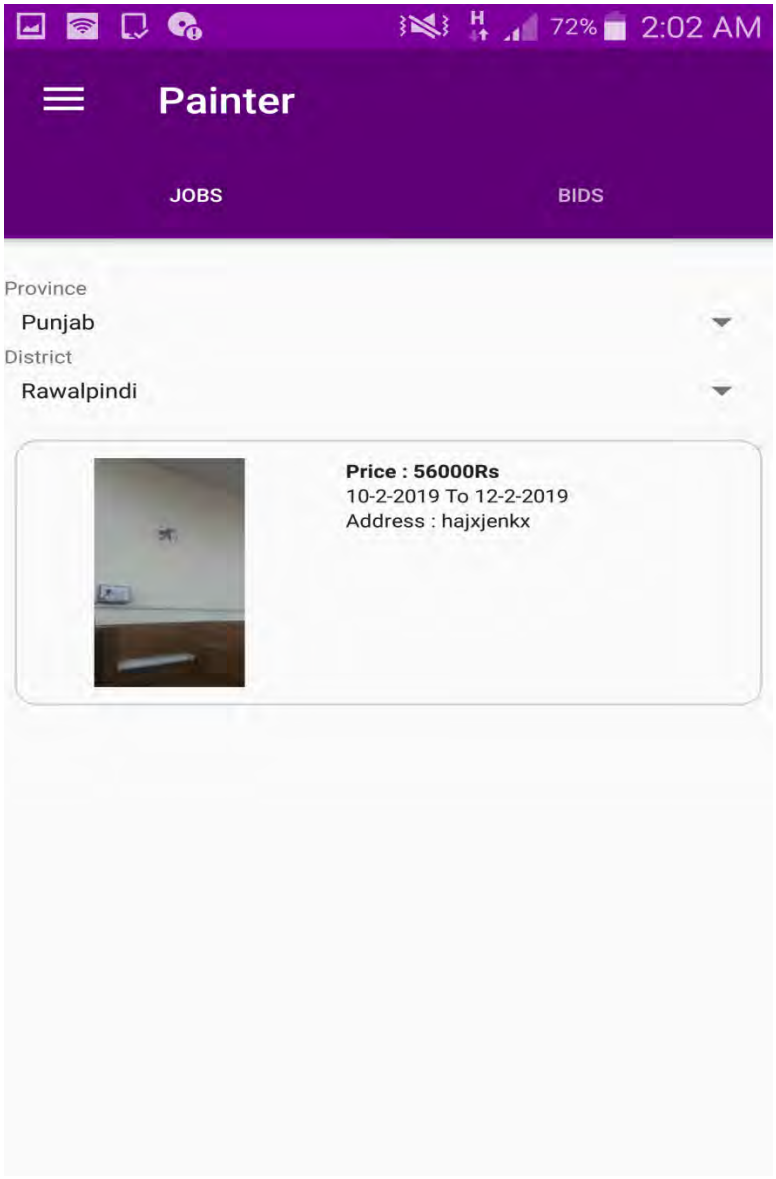

Figure 4.2.7 UI View Job

4.2.8 **Job Details**

This interface shows the job details of a particular job. It will consist of all the task of that particular paint job.

Figure 4.2.8 UI Job Details

5 Chapter: Software Testing

This chapter illustrates which will be used in this project for testing my application, testing tools and techniques and test cases.

5.1 Test Approach

A test approach is the test strategy implementation of a project and defines how testing would be carried out. Test approach has two techniques:

- Proactive: An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive: An approach in which the testing is not started until after design and coding are completed.

The reactive approach has been used in this project because by this approach, we can analyze the field or tool expert's knowledge extremity. We can analyze various risks linked to the project. Consider people, environment, and the company. Understand your project's nature and the business setup. [5]

5.2 Test Plan

A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. [6]

5.2.1 **Testing Technique**

I will use black box testing technique in which the internal structure of item being tested will not be visible as the name indicates that so tester will not be able to see inside just like black box. This technique can be used to test both functional and non functional requirements of the system and can indicate errors and missing functions etc.

5.3 Test Cases:

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

5.3.1 **Log In:**

ID	T001
Description	Users can Log in to the application
Tester	User
Set Up	Register user with id harrx@gmail.com and password 1234.
Instruction	1. Enter ID harrx@gmail.com 2. Enter Password 1234 3. Click log in button.
Expected Result	User logged in successfully.
Oracle	Pass

Table 5.3.1 TC Login

5.3.2 **Log in (Alternative Scenario)**

This test case will describe the alternative scenario of logging in to the system. The entered username and password will be checked from database if it wouldn't be found than user cannot logged in.

ID	T002
Description	User cannot log in to the system.
Tester	User
Set Up	Register user with ID harrx@gmail.com and password 1234.
Instruction	1. Enter ID harx@gmail.com 2. Enter password 123. 3. Click Log in button.
Expected Result	1. User with Id harx@gmail.com cannot be logged into the system. 2. There is an error in ID or password.
Oracle	Fail

Table 5.3.2 TC Login(Alternative Scenario)

5.3.3 **Register Customer**

This test case will tells about the success scenario of registering user when user fills in the required data correctly.

ID	T003
Description	Customer successfully registered with ID harrx@gmail.com and password 1234.
Tester	Admin
Set Up	Customer fills in registration form completely.
Instruction	1. Customer enters personal details correctly. 2. Customer enters Cnic . 3. Customer fills in contact information.
Expected Result	1. Admin verify the applicant. 2. Admin data saved in database. 3. Customer registered successfully
Oracle	Pass
Oracle	Fail

Table 5.3.3 TC Register Customer

5.3.4 **Update Profile:**

This test case will tells about the success scenario of registering advertiser when user fills in the required data correctly.

ID	T004
Description	Customer successfully updated the profile with ID harrx@gmail.com and password 1234 and contact 0308734
Set Up	Customer fills in update form completely.
Instruction	4. Customer updates name correctly. 5. Customer updates the contact info correctly
Expected Result	4. Customer data has been updated. 5. Contact no is verified by One Time Password
Oracle	Pass

Table 5.3.4 TC Update Profile

5.3.5 **Create Task:**

This test case will tell about the success scenario of creating a task of paint job.

ID	T005
Description	Customer successfully created the new task.
Set Up	Customer fills in task form completely.
Instruction	Customer selects the province. Customer selects the district. Customer enters the title of job. Customer enters the amount to start the bid. Customer enters the dates (from-to). Customer checks the box(i.e. Exterior, Interior). Customer selects the done button
Expected Result	Task has been created successfully.
Oracle	Pass

Table 5.3.5 TC Create Task

5.3.6 **Create Task (Alternative Scenario):**

This test case will tell about the alternate scenario of creating task for the paint job.

ID	T006
Description	Customer cannot create the new task successfully.
Set Up	Customer fills in task form completely.
Instruction	Customer selects the province. Customer selects the district. Customer enters the title of job. Customer enters the amount to start the bid. Customer checks the box(i.e. Exterior, Interior). Customer selects the done button
Expected Result	System shows error and asks customer to enter the date(From-To).
Oracle	Fail

Table 5.3.6 TC Create Task (Alternate Scenario)

5.3.7 **Place Bid:**

This test case will tells about the success scenario of placing bid by the painter to a particular paint job.

ID	T007
Description	Painter places the bid successfully.
Set Up	Painter fills in bid form completely.
Instruction	Painter selects the job. Painters add the bidding price. Painter selects the date (From-to). Painter selects the done button
Expected Result	Bid Placed Successfully.
Oracle	Pass

Table 5.3.7 TC Place Bid

5.3.8 **Place Bid (Alternate Scenario):**

This test case will tells about the alternate scenario of placing bid by the painter to a particular paint job.

ID	T008
Description	Painter cannot place the bid successfully.
Set Up	Painter fills in bid form completely.
Instruction	Painter selects the job. Painter selects the date (From-to). Painter selects the done button
Expected Result	System prompt message to fill the from completely.
Oracle	Fail

Table 5.3.8 TC Place Bid (Alternate Scenario)

6 Chapter: Future Enhancement

In this chapter we will discuss the features that will be added to this application. It describes the conclusion and future enhancements.

6.1 Conclusion

- People no longer need to go outside their home to find the good painters.
- Painters will be able to get direct jobs from the customer and they no longer need to search for paint jobs.
- Ratings will be maintained so customer will be able to find the best painter for their job.
- Ratings will be maintained so the painters will work for the reliable customers.
- Bidding module helps the customer to find the best price from the painters for their paint job.

6.2 Future Enhancements

In Future:

- In Future enhancement, users will be able to perform payment transactions through the application.
- Application will be uploaded on the Play Store so that users will be able to download it freely.

References

[1] P. Mohapatra, “Software Requirement Specifications Software,” in Software Engineering (A Lifecycle Approach), 5th ed. New Age, 2010, ch. 2.

[2] C. Larman, “Use cases and Functional Requirements,” in Applying UML and Patterns, 2nd ed., 1998, ch. 6.

[3] R. Pressman, “Design Concepts and Principles,” in Software Engineering: A Practitioner’s Approach, 7th ed., 1982, ch. 13.

[4] R. Pressman, “Software engineering A lifecycle approach,” in Software Engineering: A Practitioner’s Approach, 7th ed., 1982, ch. 14, sec. 5.

[5] I. Sommerville, “Software Testing,” in Software Engineering, 9th ed., 1982, ch. 8.

[6] M. Mohtashim et al. (2017, Jul 05). Black box Testing [Online]. Available: www.tutorialspoint.com/software_testing_dictionary/black_box_testing.html

References

References