COMSATS University Islamabad,
Park Road, Chak Shahzad, Islamabad, Pakistan

SOFTWARE DESIGN DESCRIPTION

(SDD DOCUMENT)

for

Madadgar

Version 1.0

By

Maham Riaz Keyani **CIIT/SP18-BSE-068/ISB**

Namra Waseem **CIIT/ SP18-BSE-097/ISB**

Supervisor

Sir Muzaffar Iqbal

Bachelor of Science in Computer Science (2018-2022)

Table of Contents

List of tables:

List of figures:

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by

Sir Muzaffar Iqbal

Signature_____

1. Introduction

In our society many people have **skills** and still do not have any jobs. It is due to **lack of resources**, education, or money. People have the skills but **right platform** to **showcase** their skills. Some people may find a platform, but they are usually paid. Our Application provides a platform that is **free of cost** and people who have the right skills can get the required exposure and can land many jobs.

And one of the major problems in our society specially for aged people and women is that they need help with **household** and **outdoor activities**. Generally, any person that needs a service which he or she cannot do themselves or they are **busy** doing other stuff can use our application to hire a service provider and get the job done. People can use work tutorials posted by the other users related to a service they are not willing to hire someone.

Our system provides **direct communication** between the customers and services without inclusion of any **third party**. Our system is **cost effect** and **efficient**. And a **great platform** for people seeking jobs and **experience**. It helps people who have need some kind of work done efficiently.

The system has **registration modules** both for service provider and customer and their accounts can be managed as well. Customer can request for a service and the nearby service providers **are notified**, and they can in turn respond or consider it. The service provider and customer can likewise **chat** or talk on call to fill in further things. Customer can **rate** the service provider account to his work or even also **complaint** about him if he tends to do anything wrong. Moreover, there is questions portal, where customers can ask questions for authenticated answers without actually hiring any service provider. The system does not provide refund functionality in case of any fraud once the **contract** has been signed between both service provider and customer. The system does not have any proper verification of documents of professionals. To ensure the authenticity of the service providers our system provides profile picture authentication using artificial intelligence. And to make the application scure users can unlock the application using face unlock.

Modules:

Module 1: User Registration and login Management

Module 2: Manage Users

Module 3: Manage Services

Module 4: Manage Complaints

Module 5: Frequently asked Questions Portal

Module 6: Administration Controls

Module 7: In-App Chat

Module 8: Notification generation

Module 9: Facial Unlock

Module 10: Profile Picture Validation using AI.

2. Design methodology and software process model

We will be using Object oriented methodology:

As we have a lot of experience in Object oriented programming. OOP makes understanding and managing the code easier. All the code is divided into classes and objects. By using the concepts of data hiding and encapsulation we can control what the user can access easily and what not. OOP provides reusability of code which saves extra coding efforts and reduces the sizes of code.

We will be using Incremental process model:

Incremental model allows divide the system into small subsystem which will make it easier for our team to divide the workload and integrate it afterwards. Incremental model does not restrict any feature to be implemented first so we will be able to choose the highest priority functionality to be implemented. Incorporating any changes in the system will be a lot easier with incremental model.

3. System overview

The service provider will be able to register by providing his personal information, skills, experience, and certification if he claims to be a professional of the field. The certification will be authenticated by the system and the service provider will be allowed to use the features of the system. The customer's account will be created using required information such as name, address, location. The location information will be used to detect the service provider nearby. Both the users will use their credential to login to the system. The users account information will be managed by providing the functionalities like **Add/remove Skills**: Skills the service provider has can be added or removed. **Update experience:** service provider can update experience in the profile. **Disable Account:** users can disable account according to their will. **Change personal Information:** Both type of users can update their personal information such as name, address, phone no. etc. **Facial Unlock:** set app security to facial unlock to maximize the privacy.

customer will post the service and notification will be sent to all nearby work-related service providers according to customer's location and filters given by the customer such as budget and service type. After being notified of the service request, service provider will respond or consider the request and then it will be displayed to the customer. Then customer after viewing the ratings of the service provider will talk to provider using chat or call function of the system and fill a contract form and will hire him. **Posting a Job**: The customer will be able to post a job request by entering the required information such as job title, description etc. a notification of the job will be sent to the service providers with related skills. Customer will be able to choose if he wants a work to be done physically or online. **Chat**: When service provider accepts a job request, they can chat about the problem and the solution of the problem and any details related to the problem and can talk about the bill of the service. **Contract**: After deciding on chat, Customer will create a contract form and if the conditions of both the parties are same according to what they have talked about then they contract will be finalized and accepted. Otherwise, Service Provider will be provided with certain minutes to decline the contract. **Tracking and progress**: once the job starts the customer can track the progress and the location of the service provider. **Rating**: once a job is complete the customer can rate the service provider and give feedback.

In case of any problem in work done by service provider or in case of fraud committed by service provider, customers can file a complaint about that service provider with proper reasoning and details. After proper investigation, service provider will pay the penalty.

If task is small and customer does not want to hire any service provider for it then customer can post a question on portal. Other customers and work-related people can comment on that post. Then work-related service provider can rate that comment to verify its authenticity. All the posted question will be stored in database and customer can search questions instead of posting a new one. customers will post a question in case they cannot afford or do not want to hire a service provider then they can post a question and check the replies on their post to do that job on their own.

All the feature and controls that a admin can have will be implemented. The administration will **receive compliant**. **Respond to complaints**. **Restrict accounts** based on complaints. **Make certificates** for the service provider for providing good services for a long period of time. And can make a **statistical analysis** of how the website is performing. **Statistical Analysis:** all the information about how the

website is performing over a certain period can be view. The details related to the traffic on the website, can than further be specified to traffic per month, week, or year. Number of new registrations, higher rated service providers.

The system will check if the profile picture uploaded by the service provider is Valid: It is a picture of a human being. Eyes are visible and clear. The picture is not a side pose. The picture is not blur. The picture is a closeup of face. The nose and the lips are visible.

3.1. Architectural design

In this project, the architecture followed is Model-View-Controller (MVC) which is a software architecture widely used in the development of mobile and web applications.
In MVC structure, the functionality is divided into 3 major parts:

1. Models
2. Views
3. Controller

Each part is implemented and tested independently. Helps keep the code is manageable form. It separates an application into the following components:

- Models for handling data and business logic. It is responsible for holding the functions and variables that are involved with what it is representing. It functions more like a class in the OOP.
- Controllers for handling the user interface and application by taking input from the user, sending it to the model to get the appropriate output and then sending it to the view to display the response to the user
- Views for handling graphical user interface objects and presentation. It will contain the markup, CSS, HTML amongst others used in the creation of the web page.

Models: Customer, administration, Service providers,
Controllers: Registration system, Hiring System, Notification system, In-app chat system, Tracking system, rating system, Profile picture authenticating system, faq portal system, Administration control system, statistical analysis system.
V**iews:** Customer client, service provider client, Administration client.

3.1.1. Modular decomposition:

This is the modular decomposition for the whole system, for the sake of explaining how the system works. The system might not be implemented exactly how it is explained here. However, it will explain the overall functionality of the system.

Registration:
All the users will have to register to the application to use the feature. The system will provide users with an option to be service provider or a customer as a signup form will be shown likewise.

Authenticating service provider's profile:
While registering the service provider must make sure his profile picture shows his face with the features completely visible. That will be test using artificial intelligence.

Database:

All the Information provided by the users will be entered into a database. The information stored in database will be used to give search results, manage account settings. And provide administration with controls to keep check on the functionality of the system.

Hiring Service Providers:
Customers will search a service by typing combination of characters into the search bar and the system will look up the query in the database and return the results. Once the customer chooses to hire a service provider he will need to register. If he is already registered the process of hiring will begin.

Notification:
When a service provider is hired a notification is generated and sent to service provider. The system will generate a notification that will be sent to the service providers with related skilled when a customer posts a job he needs to be done. The job request in saved into the database. The system will generate a notification when a job is being responded to and send it to the person who posted the job. When a customer is done creating the contract the system will notifies the service provider about it. When a service provider denies the contract, the system notifies the customer. When a service provider declines an accepted job offer system notifies the customer. When a complaint is registered system alerts the administration. When administration responds to the complaints the system notifies the customer.

Contract:
When both the service provider and the customer agree upon certain term the system provides them with a contract form in which the details are mentioned and a copy of the contract is sent to both the parties via email. And is also saved to the database.

Tracking:
The hired service provider will be tracked and will be asked to provide continuous feed back about the progress of the work being done that will be communicated to the customer.

Rating:
Once the job is done. The customer can provide review and rate the service and the service provider. Which will be visible to any person visiting the profile of that service provider. The rating will be stored in the database.

Statistical Analysis:
The system will keep track of the traffic on the website and provide information about the performance of the website by taking information from the database, looking at how many people signed up and what are the activities that are being performed more.

FAQ Portal:
Customers with queries can post a question and get answers from different service providers. Database will store all the posted question and their answers. The rating of the answers and the time when the question was asked and answered. People who have registered will be able to ask and answer the questions. If person searches A question the query will be sent to database and all the related question and answers will be displayed on screen.

3.1.2. Architectural Diagram:

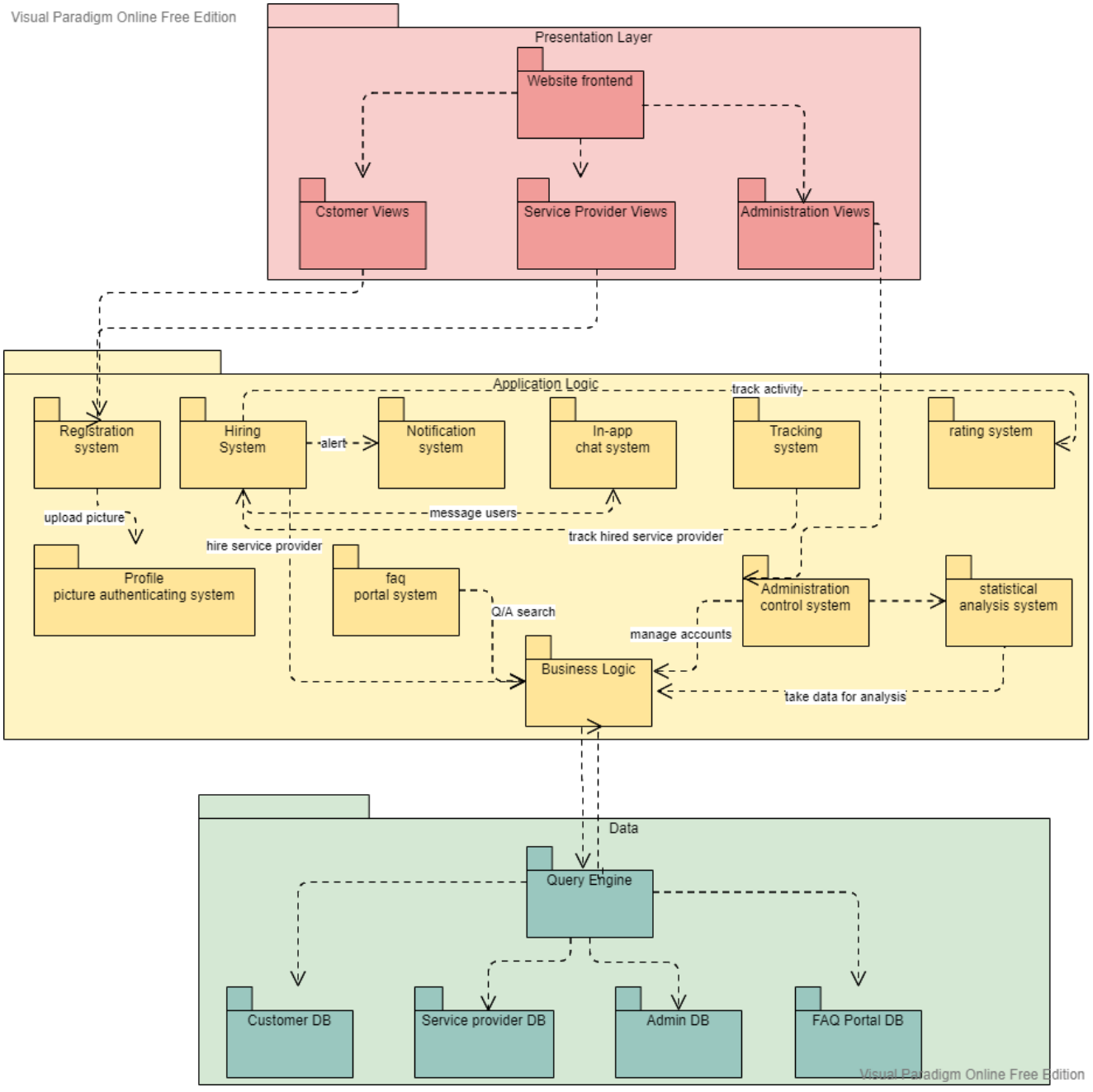

Figure 1 architecture Diagram

3.1.3. Process flow/Representation

3.1.3.1. Activity Diagram 1: register User

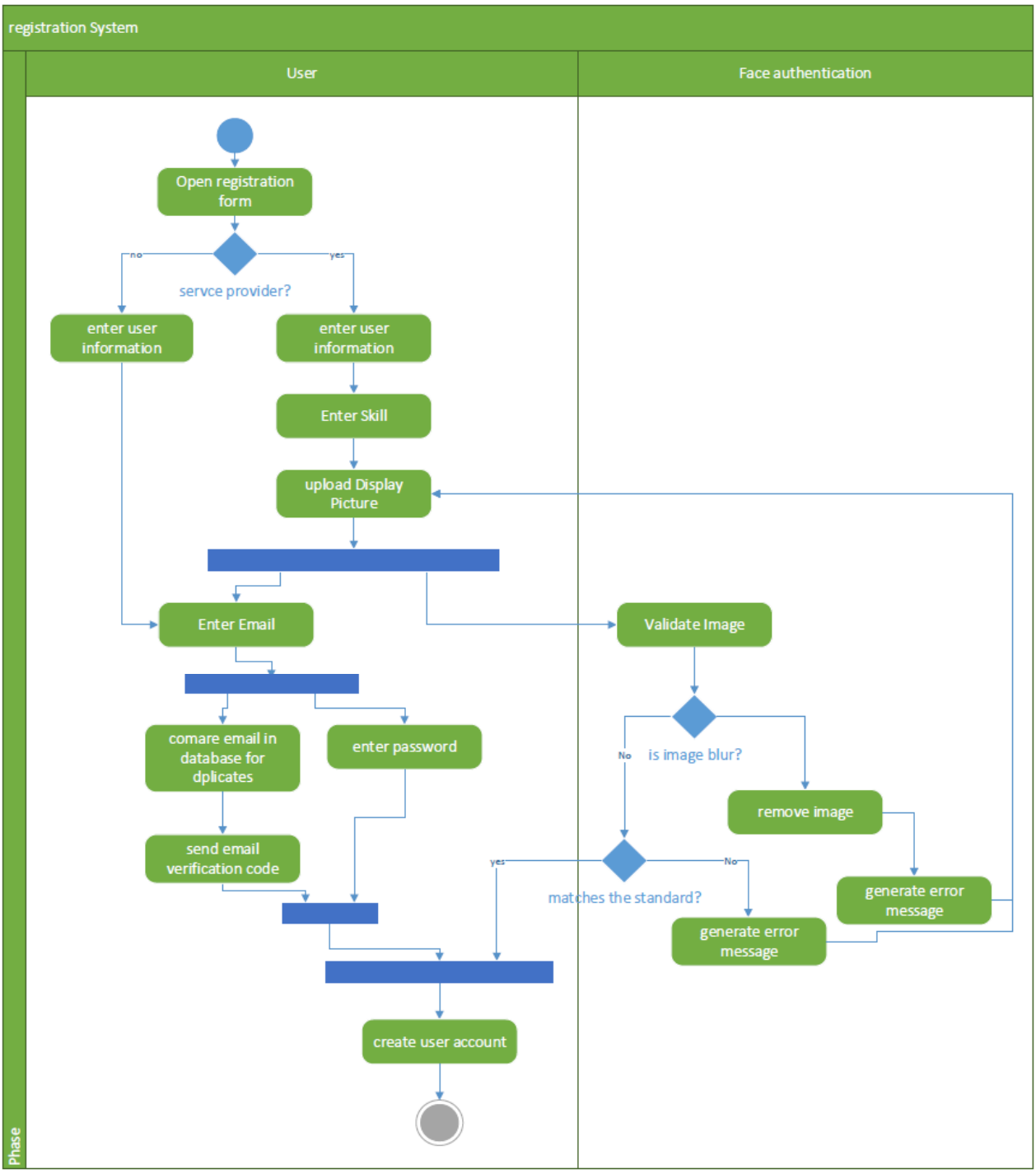

Figure 2 Activity Diagram 1: Register User

3.1.3.2. Activity Diagram 2: Hire helper

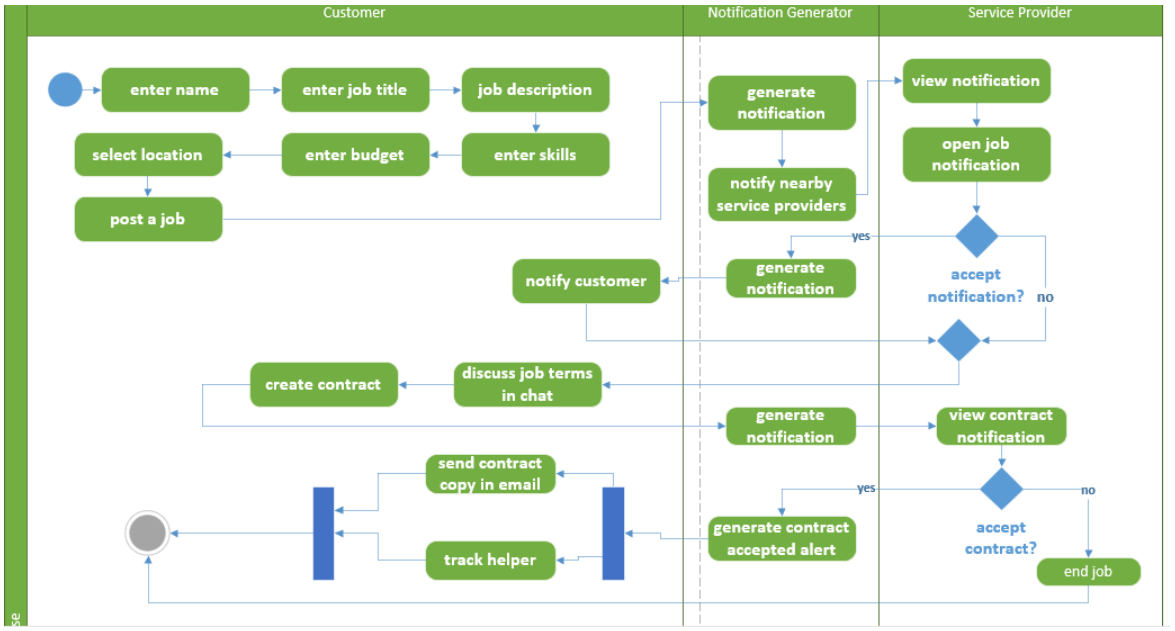

Figure 3 Activity diagram 2: hire Helper.

3.1.3.3. Activity Diagram 3: Post a Question

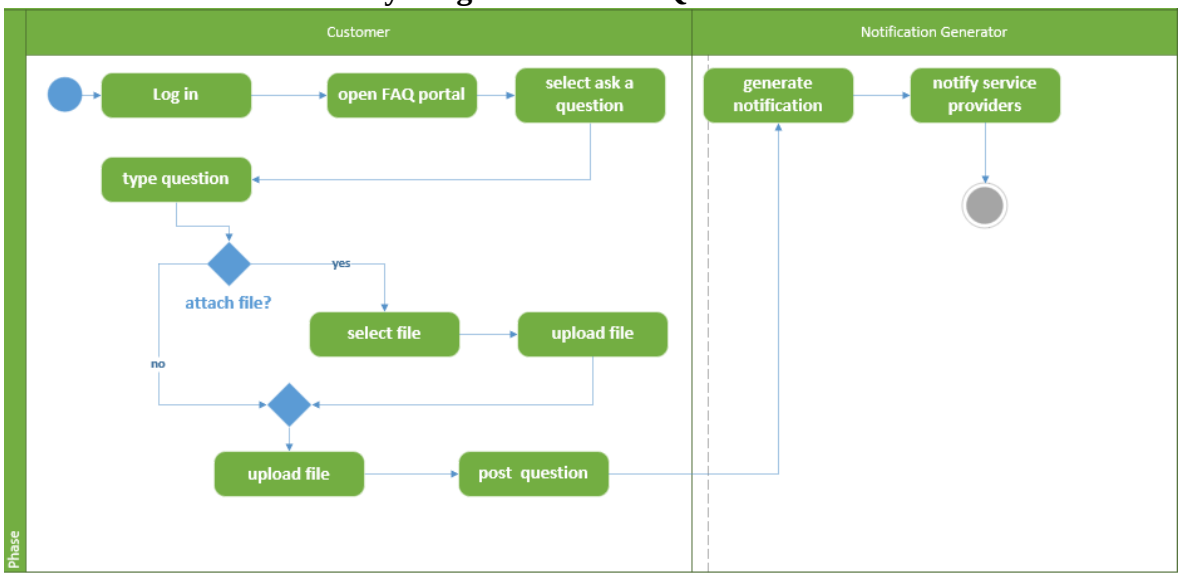

Figure 4 Activity Diagram 3: Post a question

3.1.3.4. Activity diagram 4: Web Analytics

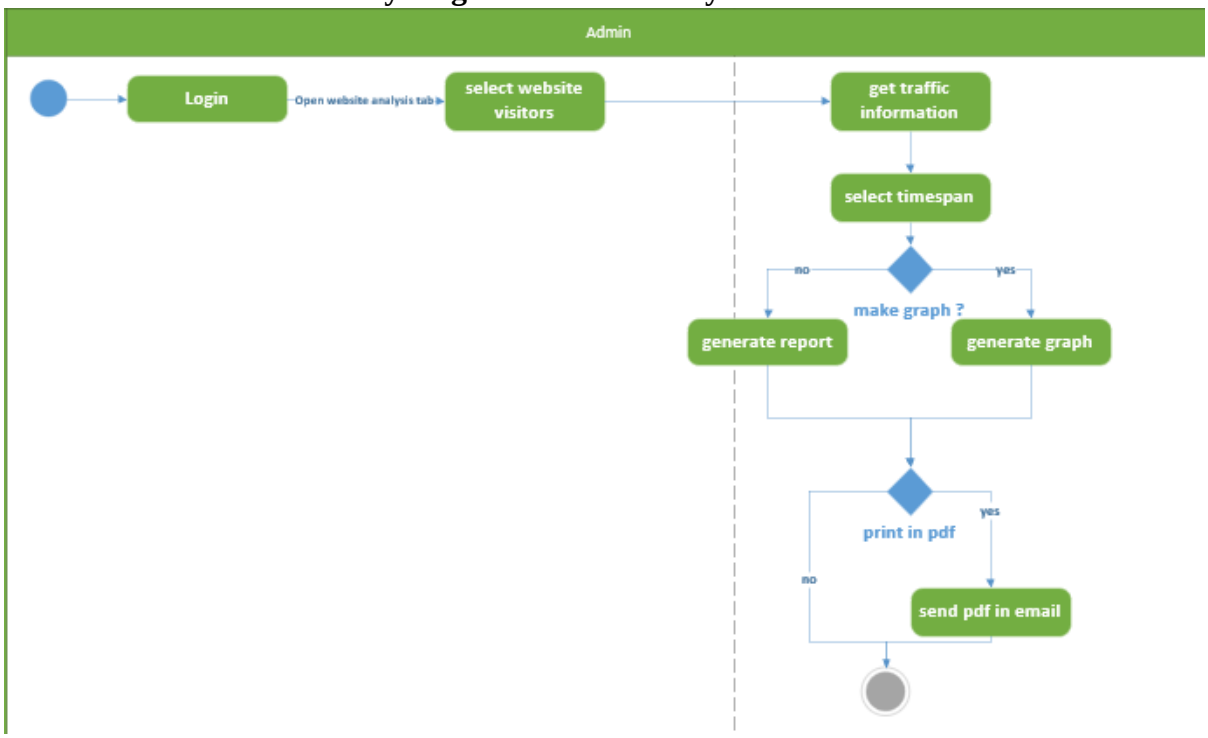

Figure 5 Activity diagram 4: Web analytics

4. Design models

4.1. Class Diagram:

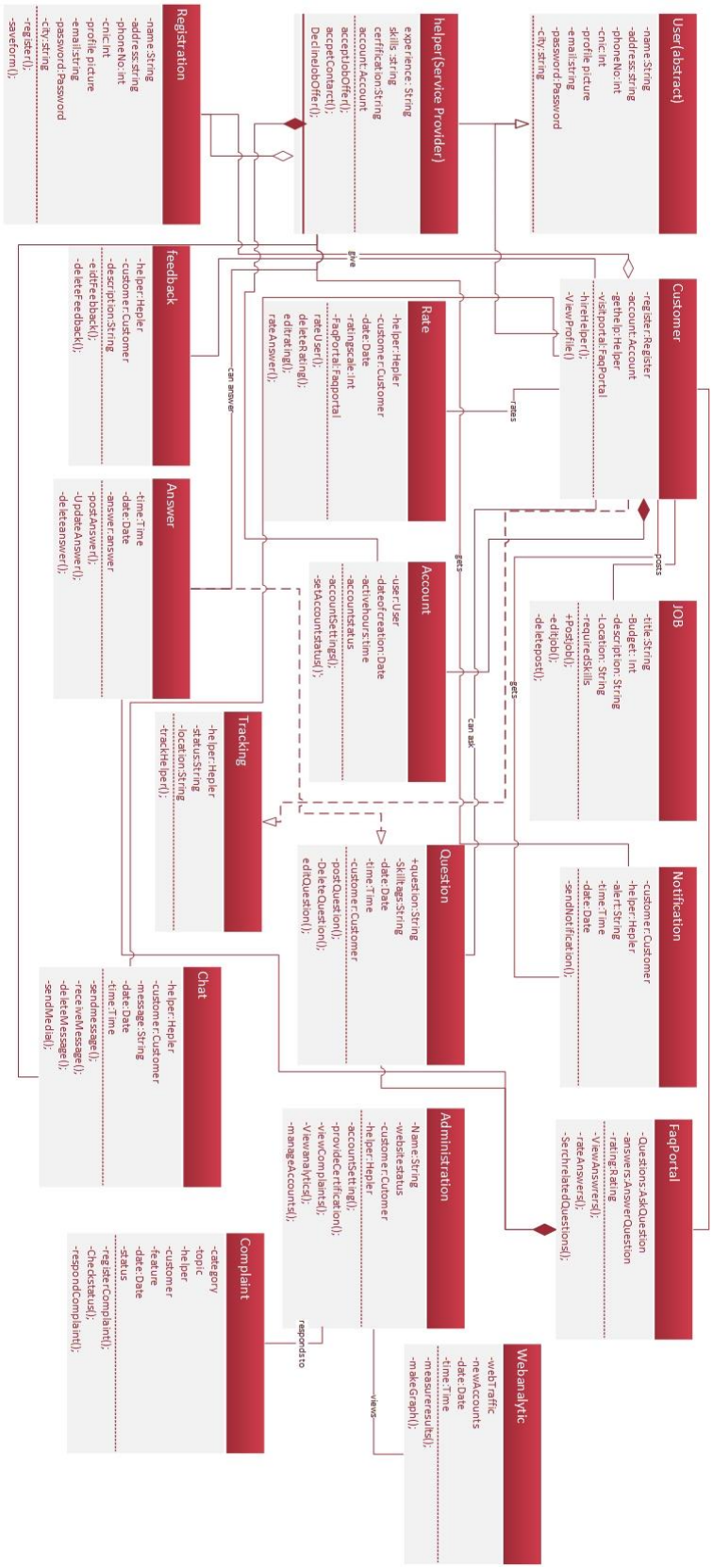

Figure 6: Class Diagram

4.2.Sequence Diagram1: register:

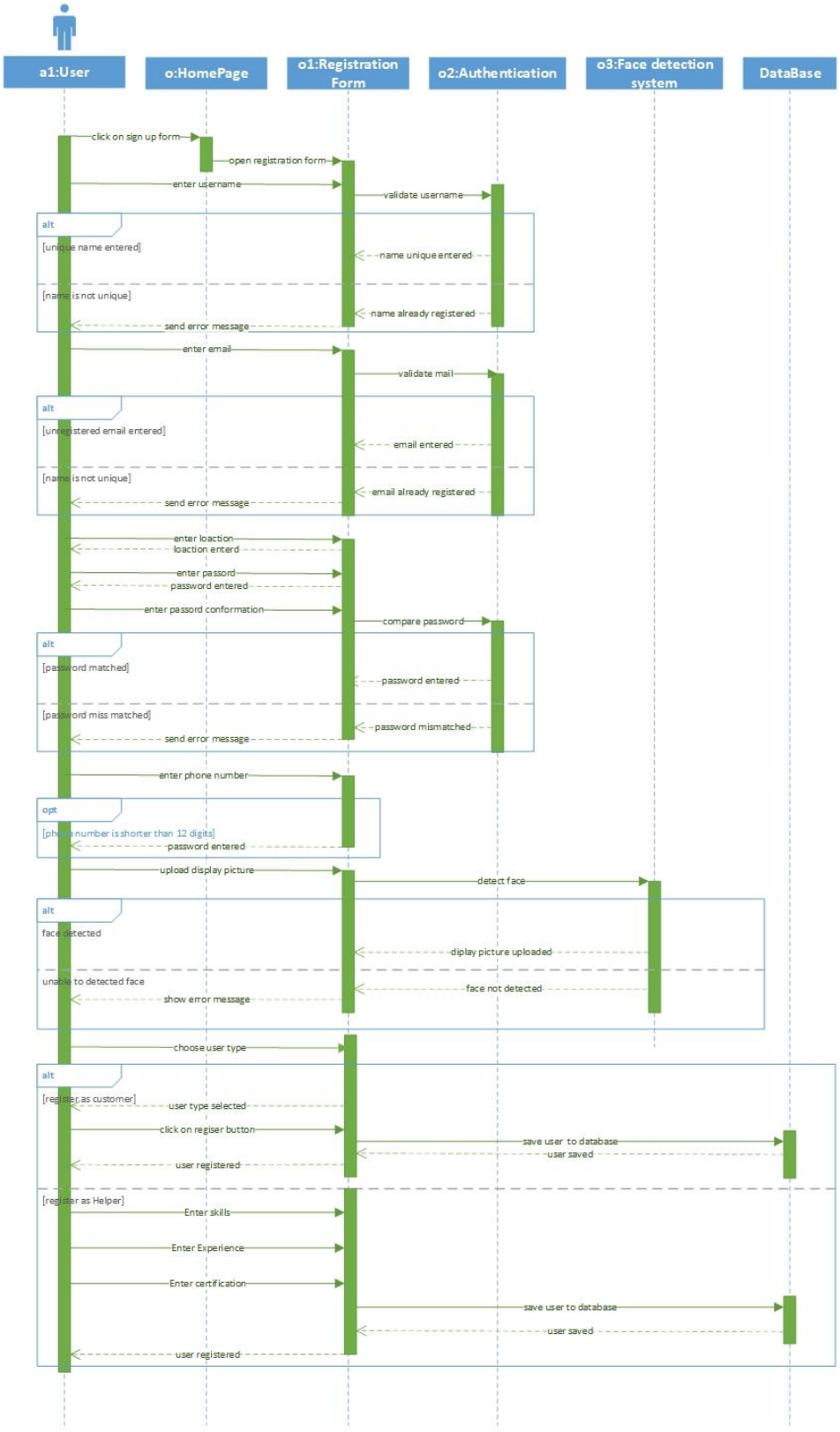

Figure 7 Sequence Diagram 1: Register

4.3.Sequence Diagram2: Hire Helper:

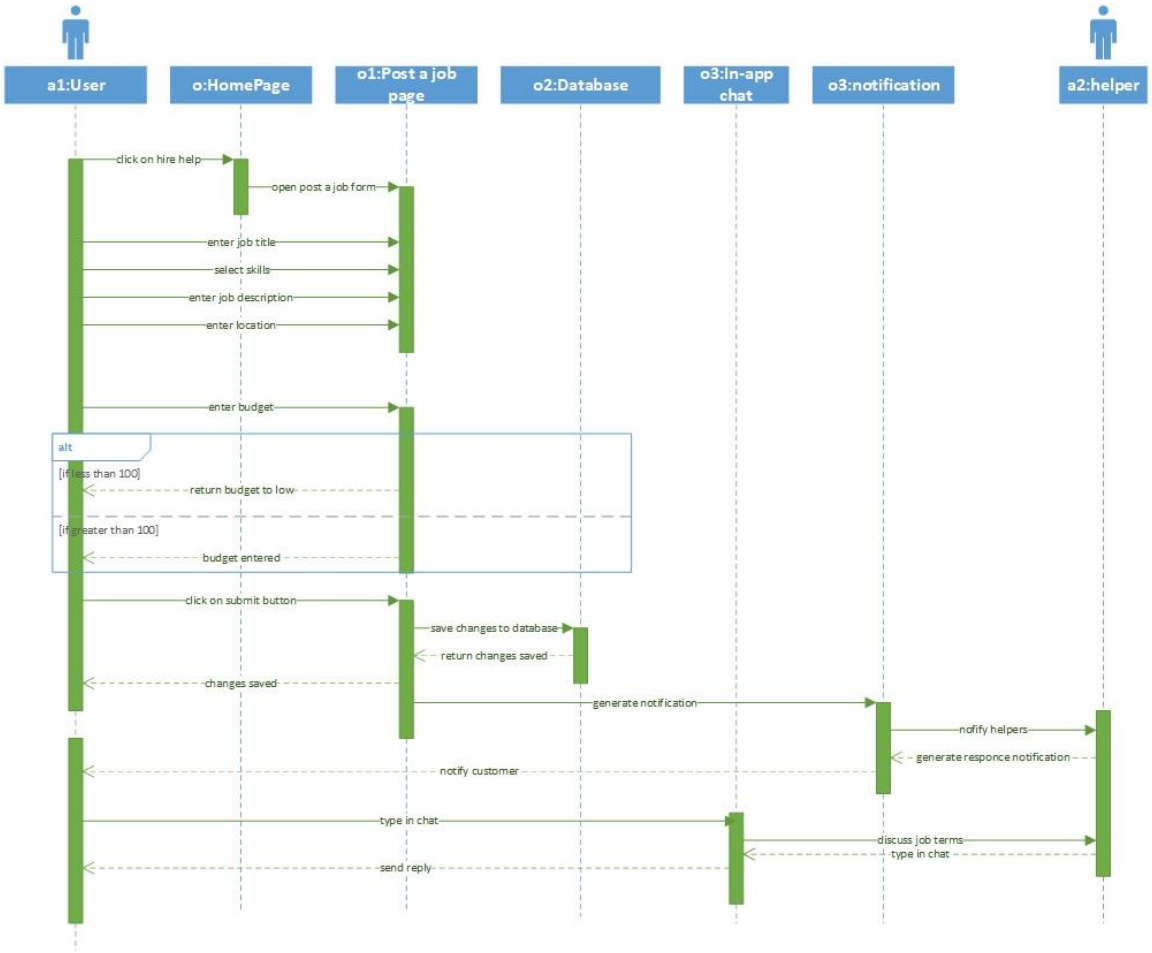

Figure 8 Sequence diagram: Hire Helper.

4.4.Sequence daigram3: Rate helper

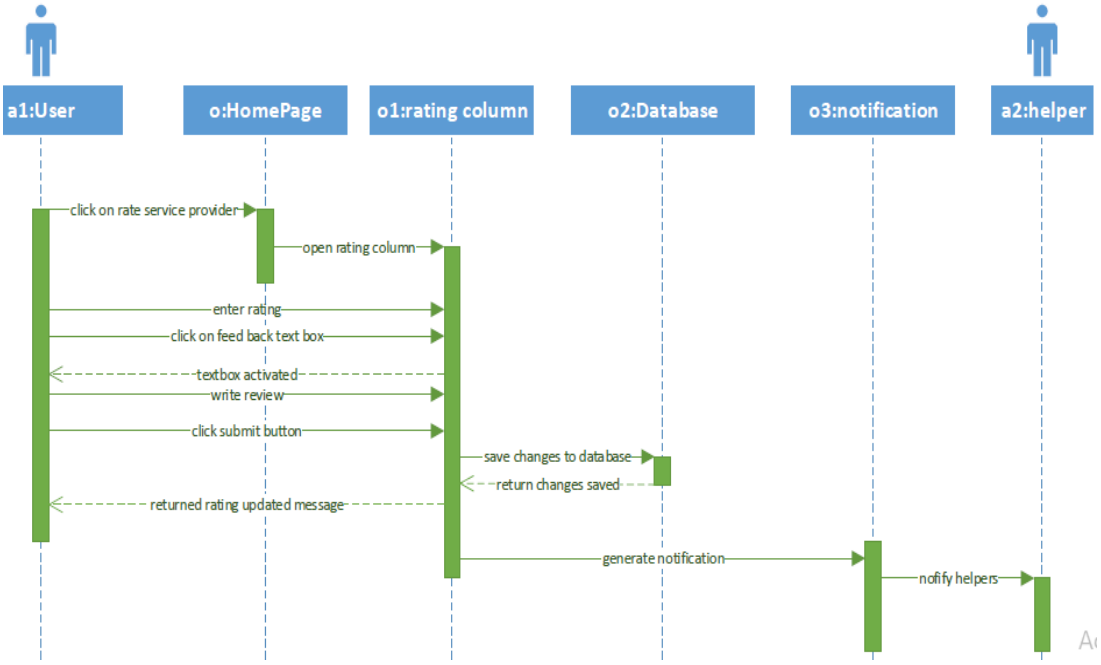

Figure 9 Sequence Diagram 3: Rate Helper

4.5.Sequence daigram4: Register complaint

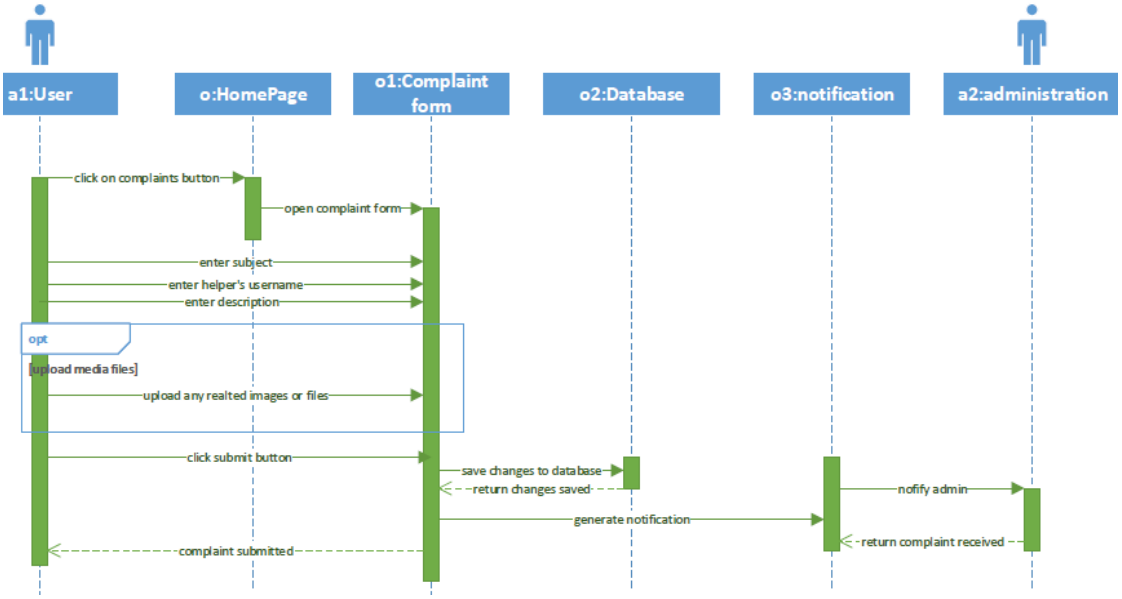

Figure 10 Sequence Diagram 4: Complaints

4.6.State machine diagram 1 Account:

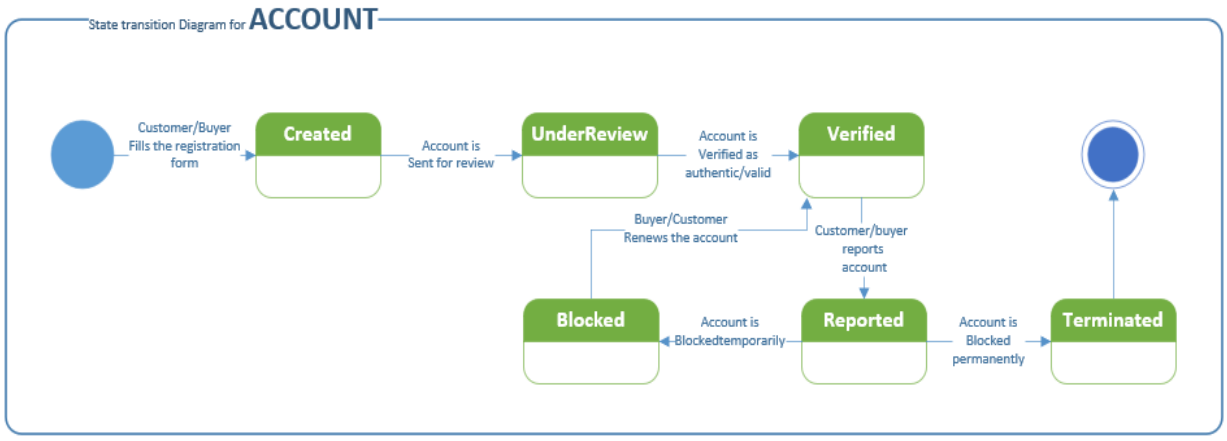

Figure 11 State Machine Diagram 1: Account

4.7.State machine diagram 2 Question:

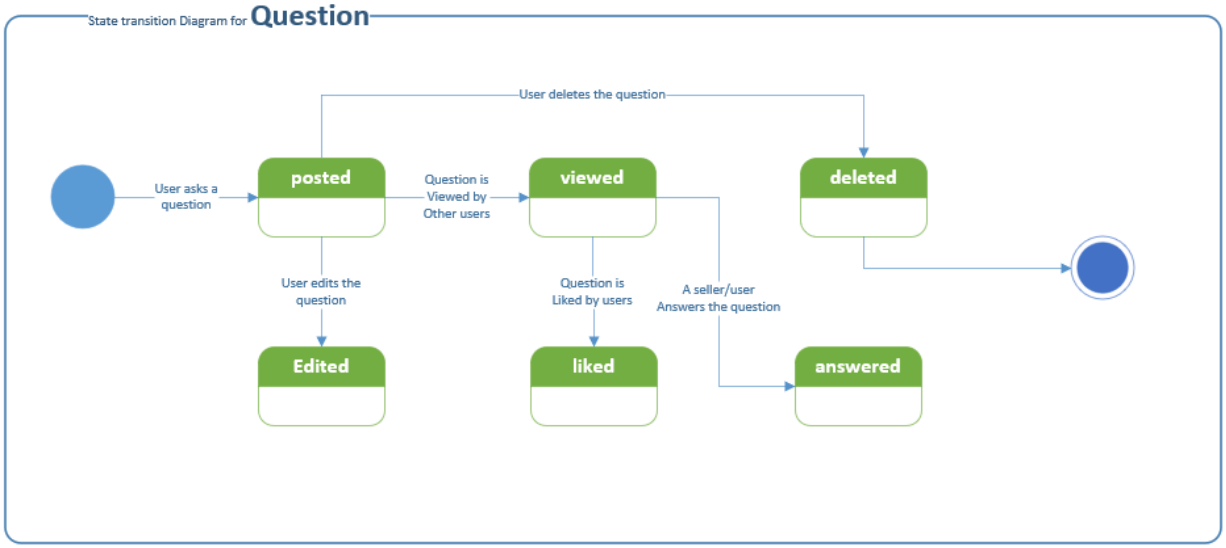

Figure 12 State machine diagram 2: Question

4.8.State machine diagram 3 Complaints:

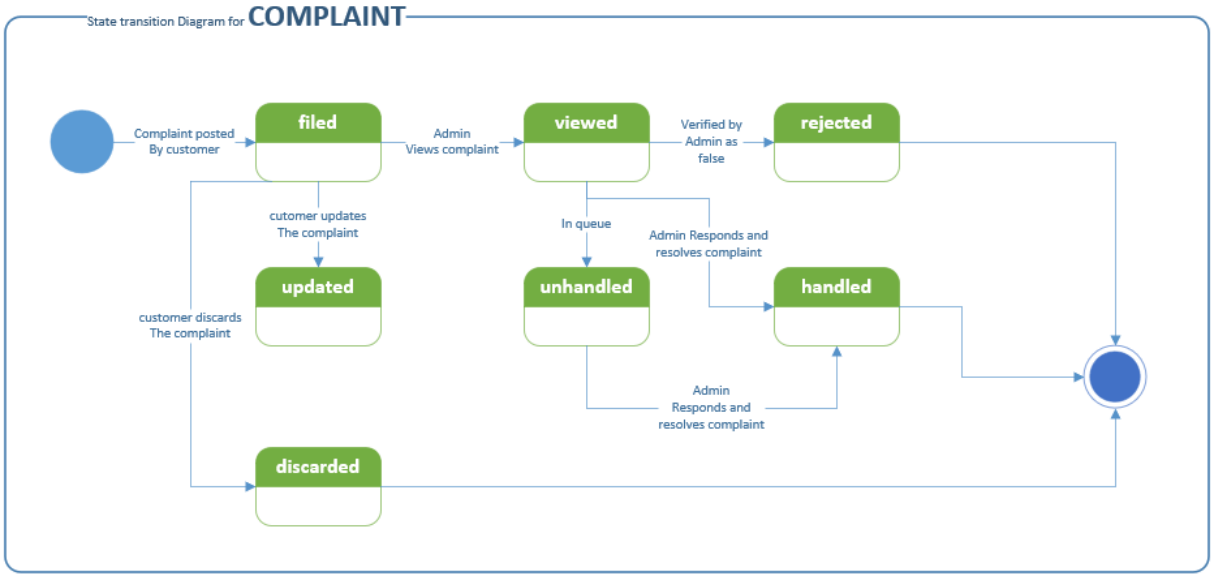

Figure 13 State machine diagram 3 Complaints

4.9.State machine diagram 4: JOB

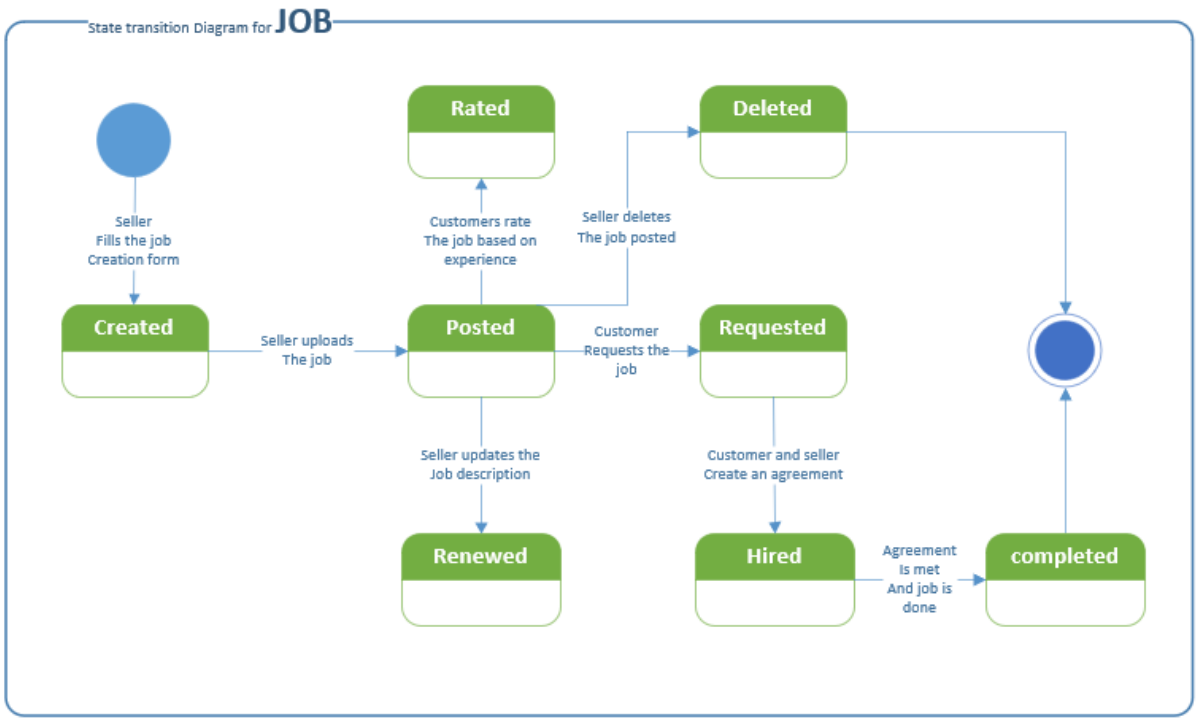

Figure 8 state Machine diagram 4 JOB

5. Data design

The users of our software system will be entering data into form. Registration form, where users will enter data to register to our website and the user information will be stored inside the database and a login form where the user will enter his credentials and the credentials will be cross checked in the database to gain access to the account.
Post a job form, user enters the required information in the from and the data is stored into the database. And the job post is visible to the nearby helpers.
Post a question form, users with queries related to work can ask a question by entering the top skills and question in the form which is stored in the database. And displayed in the FAQ Portal by fetching the data entered by users.
Answer a question, users respond to the questions posted by other users and the responses are saved to the database.

- Madadgar

5.1.Data dictionary

Table 1 Entity Account

Attributes/functions	Data type/ Return type	Entity	description
-user	User	Account	Object of user class
-dateofcreation	Date		Date when account was created
-activehours	Time		Time when account was actively being used.
-accountstatus	String		Accounts availability status
-accountSettings();	void		All the action from changing personal information to changing privacy settings
-setAccountstatus();	boolean		Set availability status

Table 2 Entity Administration

Attributes/functions	Data type/ Return type	Entity	description
-Name	String	Administration	Name of the admin member logged in
-customer	Cutomer		Object of customer class for, Details of customers accounts
-helper	Hepler		Object of Helper class for, Details of Helpers accounts
-webAnalytics	Webanalytics		Object of webAnalytics class
-accountSetting();	void		All the action from changing personal information to changing privacy settings
-provideCertification();	void		Providing certification to helpers with higher rating.
-viewComplaints();	void		View complaints filed by customers
-Viewanalytics();	void		View the results of web analysis
-manageAccounts();	void		Manage users accounts such as restricting accounts.

Table 3 Entity Answer

Attributes/functions	Data type/ Return type	Entity	description
-time	Time	Answer	Time when someone answers the question
-date	Date		Date when someone answers the question
-answer	String		The text that someone enters in the answer text box
-postAnswer();	void		To submit the answer.
-UpdateAnswer();	void		To edit the answer.
-deleteanswer();	void		To delete the answer.

Table 4 Entity Chat

Attributes/functions	Data type/ Return type	Entity	description
-helper	Hepler	Chat	Object of helper class for, Details of helpers accounts
-customer	Customer		Object of customer class for, Details of customers accounts
-message	String		The text message to be entered
-date	Date		Date when someone sends or receives a message
-time	Time		Time when someone sends or receives a message
-sendmessage();	void		To send the message.
-receiveMessage();	void		To receive message
-deleteMessage();	void		To delete message
-sendMedia();	void		To attach media in a message

Table 5 Entity Complaint

Attributes/functions	Data type/ Return type	Entity	description
+category	String	Complaints	Type of complaint
-topic	String		Topic of complaint
-helper	Hepler		Object of helper class for, Details of helpers accounts
-customer	Customer		Object of customer class for the person who is complaining
-date	Date		Date of complaint
-status	String		Status of the complaint progress
-registerComplaint();	void		Submit complaints
-Checkstatus();	boolean		Mark status of complaint
-respondComplaint();	void		Send complaints response

Table 6 Entity Customer

Attributes/functions	Data type/ Return type	Entity	description
-register	Register	Customer	Object of register class
-account	Account		Object of account class
-gethelp	Helper		Object of helper class
-visit portal	FaqPortal		Object of Faq portal class
-ViewProfile()	void		Open profile function
-hireHelper()	void		Hire helper function

Table 7 Entity FAQ Portal

Attributes/functions	Data type/ Return type	Entity	description
-Questions	Question	FAQportal	Object of question class
-answers	Answer		Object of answer class
-rating	Rating		Object of rating class
-ViewAnswrers();	void		Function to view answers
-rateAnswers();	void		
-SerchrelatedQuestions();	void		

Table 8 Entity Feedback

Attributes/functions	Data type/ Return type	Entity	description
-helper	Hepler	Feedback	Object of helper class
-customer	Customer		Object of customer class
-description	String		Variable to store feedback text.
-eidtFeebback();	void		Function to make changes to feedback.
-deleteFeedback();	void		Function to delete feedback

Table 9 Entity helper

Attributes/functions	Data type/ Return type	Entity	description
+experience	String[]	Helper	array to store experience
+skills	String[]		array to store skills
+cerfification	String[]		array to store certification
-account	Account		Object of account class
acceptJobOffer();	void		Function to accept offered job
accpetContarct();	void		Function to accept contract
DeclineJobOffer();	void		Function to decline job offer.

Table 10 Entity Job

Attributes/functions	Data type/ Return type	Entity	description
-id	int	Job	Integer to store unique ID
-title	String		Variable to store title of the job
-Budget	double		Variable to store the budget
-description	String		Variable to store job description
-Location:	String		Variable to store location.
-requiredSkills	String[]		Array to store skills
-editjob();	void		Function to edit posted job
+Postjob();	void		Function to post job
-deletepost();	void		Function to delete job

Table 11 Entity notification

Attributes/functions	Data type/ Return type	Entity	description
-id	int	Notification	Integer to store unique ID
-customer	Customer		Object of Customer class
-helper	Hepler		Object of Hepler class
-alert	String		Variable to store the message
-time	Time		Object of Time class
-date	Date		Object of Date class
-sendNotification();	void		Function to send alert message
generateNotification();	void		Function to generate alert message

Table 12 Entity Question

Attributes/functions	Data type/ Return type	Entity	description
+question	String	Question	Variable to store the text of question
-Skilltags	String[]		Array to store required skills
-date	Date		Object of Date class
-time	Time		Object of Time class
-customer	Customer		Object of Customer class

Table 13 Entity Rating

Attributes/functions	Datatype/ret type	Entity	description
-helper	Helper	Rating	Object of Hepler class
-customer	Customer		Object of Customer class
-date	Date		Object of Date class
-ratingscale	double		Variable to store the message
-FaqPortal	Faqportal		Object of FAQPortal class
rateUser();	double		Function to rate users
deleteRating();	void		Function to delete rating
updaterating();	void		Function to update rating
rateAnswer();	double		Function to rate answers

Table 14 Entity Registration

Attributes/functions	Data type/ Return type	Entity	description
-id	int	Registration	Integer to store unique ID
-name	String		Variable to store username
-address	String		Variable to store address
-phoneNo	int		Variable to store cnic
-cnic	int		Variable to store cnic
-profile picture	JPG,JPEG,PNG		Variable to store profile picture
-email	String		Variable to store email
-password	Password		Variable to store password
-city	String		Variable to store city
-register();	void		Function to register user
-saveform();	void		Function to save the incomplete registration form.

Table 15 Entity Tracking

Attributes/functions	Data type/ Return type	Entity	description
-helper	Hepler	Tracking	Object of Hepler class
-status	Boolean		Variable to store tracking status
-location:String	String		Variable to store tracking location
-trackHelper();	void		Function for tracking hire helper

Table 16 Entity User

Attributes/functions	Data type/ Return type	Entity	description
-id	int	User	Integer to store unique ID
-name	String		Variable to store username
-address	String		Variable to store address
-phone No	Int		Variable to store phone no.
-CNIC	Int		Variable to store cnic
-profile picture	JPG,JPEG,PNG		Variable to store profile picture
-email	String		Variable to store email
-password	Password		Variable to store password
-city	String		Variable to store city

Table 17 Entity Web Analytics

Attributes/functions	Data type/ Return type	Entity	description
-webTraffic	Double	Web analytics	Variable to store number of users visiting website
-newAccounts	Double		Variable to store number of new registrations on website
-date	Date		Object of Date class
-time	Time		Object of Time class
-measureresults();	Double		Function to calculate results.
-makeGraph();	void		Function to make graphs

6. Algorithm & Implementation

6.1. Algorithm 1: Registration

Step 1. User will click on register link to create account.
Step 2. Registration form will appear.
Step 3. The system will ask user to select category i.e, Helper or Customer.
Step 4. If the user selects customer category, fields related to Helper will be disabled for customer.
Step 5. User will fill all required fields in order to create account.
Step 6. User will click on register button after filling the form.
Step 7a. If the provided information is valid, the system will create user account and information will be stored in database.
Step 7b. Else, error message will be shown.

6.2. Algorithm 2: Register Complaint

Step 1. The customer will click on register complaint link.
Step 2. The complaint form will appears on the screen
Step 3. The customer will fill required fields i.e. service provider's name, job title, job description and contract.
Step 4a. The customer will click on register complaint button. The complaint will be registered if the provided information is valid. The registered complaint will be stored in database.
Step 4b. Else, error message will appear on the screen.

6.3. Algorithm 3: Hire Helper

Step 1. The customer will click on hire helper link.
Step 2. The customer will then fill all the required fields in "post a job" form.
Step 3. The customer will then click on post job button.
Step 4. The customer will get notification of the service providers' response to his/her job request
Step 5. The customer will view the notification.
Step 6. The customer will select one of the service providers who have accepted his/her job request
Step 7. The customer will then communicate with him through chat.
Step 8. The customer will click on create contract button if both parties agree on terms and conditions.
Step 9. The customer will then fill contract form and click on create contract button.
Step 10. The contract will be created.
Step 11. A softcopy of the contract will be sent to both service provider and customer through email, if the service provider accepts the contract.
Step 12. The service provider hired pop up will appear on the screen.

6.4. Algorithm 4: Rate Service Provider

Step 1. The customer will receive the notification of job completion by service provider.
Step 2. The customer will click on notification link and view the notification.
Step 3. The customer will rate the service provider out of five for the services.
Step 4. The system will updates service provider's profile.
Step 5. Rating will appear on the profile of the service provider.

6.5. Algorithm 5: Post a Question

Step 1. The customer will click on FAQ portal link.
Step 2. The customer will then click on Ask a question link.
Step 3. The customer will fill the title and body fields.
Step 4. The customer will then click on post a question a button.
Step 5. The question will be posted and will be stored in database.
Step 6. Service providers will receive notification of the question posted by customer.

6.6. Algorithm 6: Answer a Question.

Step 1. The service provider will click on view all questions button.
Step 2. The list of posted questions will appear on the screen.
Step 3. The service provider will then click on question to view.
Step 4. The service provider will then click on answer a question button.
Step 5. The service provider will type the answer and attach a file if required.
Step 6. The service provider will click on post answer button.
Step 7. The answer will be posted and stored in database.
Step 8. The customer will receive the notification of the answer.

6.7. Algorithm 7:Hire Helper.

Step 1: User clicks on be a helper button and form appears.
Step 2: User enters the skills he has. If the skills are less than 2 the system asks user to enter 1 more skill else moves to next field
Step 3: User enters experience.
Step 4: Users sets his least demand to do any kind of work. (budget)
Step 5: User clicks on register.

7. Software requirements traceability matrix

This section should contain a table that summarizes how each software requirement has been met in this document. The tabular format permits one-to-one and one-to-many relationships to be shown.

Table 7.1 Requirements Traceability Matrix

Req. Number	Ref. Item	Design Component	Component Items
FR-01	Class Diagram/sequence diagram/activity diagram	Registration	register();
FR-02	Class Diagram/sequence diagram	Registration	register();
FR-03	Class Diagram/sequence diagram	Registration	register();
FR-04	Class Diagram/sequence diagram	Registration	register();
FR-05	Class Diagram	Login	signIn();
FR-06	Class Diagram	Account/customer/hepler	Viewprofile();
FR-07	Class Diagram	Account/customer/helper	Accountsettings()
FR-08	Class Diagram	Account/customer/helper	Accountsettings()
FR-09	Class Diagram	Account/customer/helper	Accountsettings()
FR-11	Class Diagram	Account/customer/helper	Accountsettings()
FR-12	Class Diagram	Account/customer/helper/feeddback	Givefeedback(); viewFeedback();
FR-13	Class Diagram	Account/customer/helper/Job	viewjob();
FR-14	Class Diagram	Account/customer/helper/Job	acceptob();
FR-15	Class Diagram	Account/customer/helper/Job	withdrawjob();
FR-16	Class Diagram/ Sequence diagram	Account/customer/helper/In-app chat	openchat();
FR-17	Class Diagram/ Sequence diagram	Account/customer/helper/In-app chat	Sendmessage();
FR-18	Class Diagram/ Sequence diagram	Account/customer/helper/In-app chat	Receivemessage();
FR-19	Class Diagram	Account/customer/helper/Job	contract();

Table 18 Requirements Traceability Matrix

FR-20	Class Diagram	Account/customer/helper/Job	contract();
FR-21	Class Diagram	Account/customer/helper/Job	contract();
FR-22	Class Diagram	Account/customer/helper/Job	contract();
FR-23	Class Diagram	Account/customer/helper/Job	contract();
FR-24	Class Diagram	Account/customer/helper/Faqportal/Question	Postquestion(); Viewquestion();
FR-25	Class Diagram	Account/customer/helper/Faqportal/Answer	Answer();
FR-26	Class Diagram	Account/customer/helper/Faqportal/Answer/rate	rateanswer();
FR-27	Class Diagram	Account/customer	searchhelper();
FR-28	Class Diagram	Account/customer /Job	postjob();
FR-29	Class Diagram	Account/customer	hirehelper();
FR-30	Class Diagram	Account/customer /Job	deletejob();
FR-34	Class Diagram	Account/customer/helper /ratehelper	ratehelper ();
FR-35	Class Diagram	Account/customer/helper/tracking	Trackhelper();
FR-36	Class Diagram	Account/customer/complaint	registercomplaint();
FR-37	Class Diagram	Account /customer/complaint	Complaintresponce\();
FR-38	Class Diagram	Account/customer/helper/Faqportal/Question	serachquestion ();
FR-39	Class Diagram	Account/customer/helper/Faqportal/Question	Postquestion();
FR-40	Class Diagram	Account/customer/helper/Faqportal/Question	deletequestion();
FR-41	Class Diagram	Account/ helper/Faqportal/Answer/rating	viewratedAnswer();
FR-42	Class Diagram	Account/Administration/customer/ complaint	viewComplaints();
FR-43	Class Diagram	Account/Administration/helper	restrictAccount();
FR-44	Class Diagram	Account/Administration/helper	providecertificate ();
FR-45	Class Diagram	Account/Administration/webanalytics	Viewtraffic();
FR-46	Class Diagram	Account/Administration/customer/ complaint	respondComplaints();

8. Human interface design

8.1.Screen images

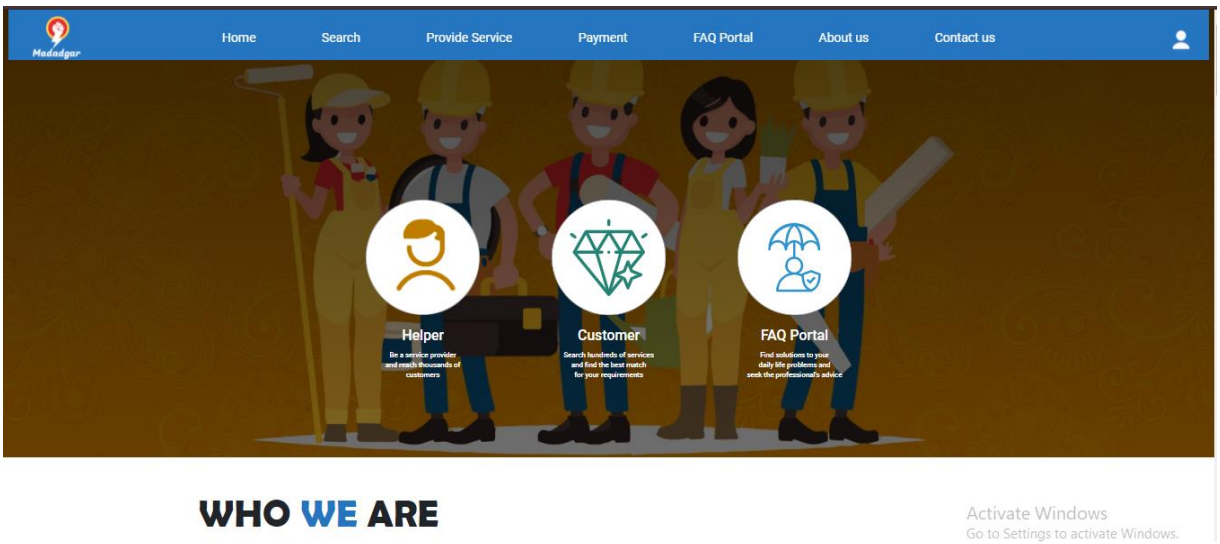

Figure 9 landing page

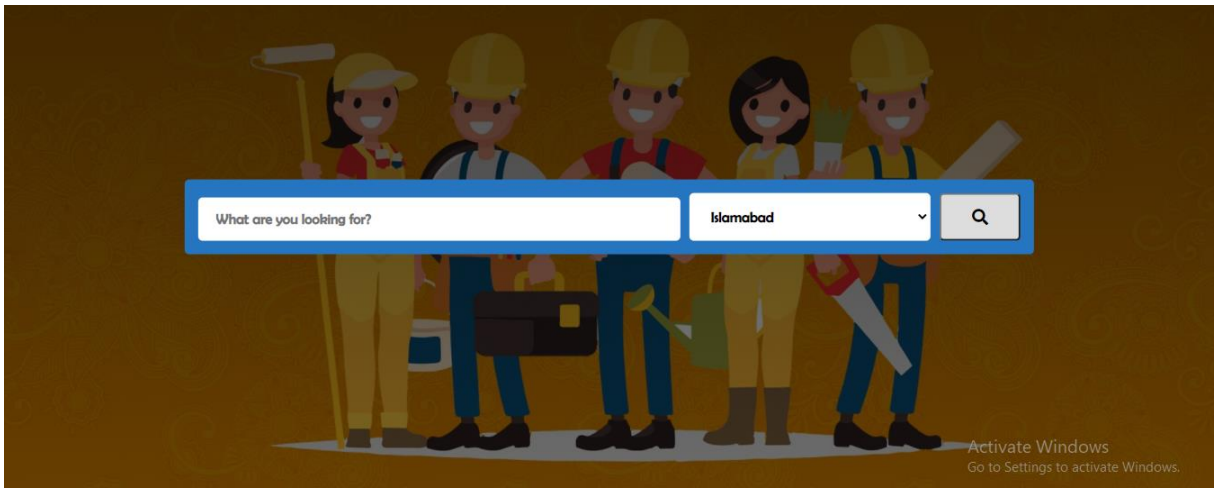

Figure 10 Search skill and helpers page

Madadgar application

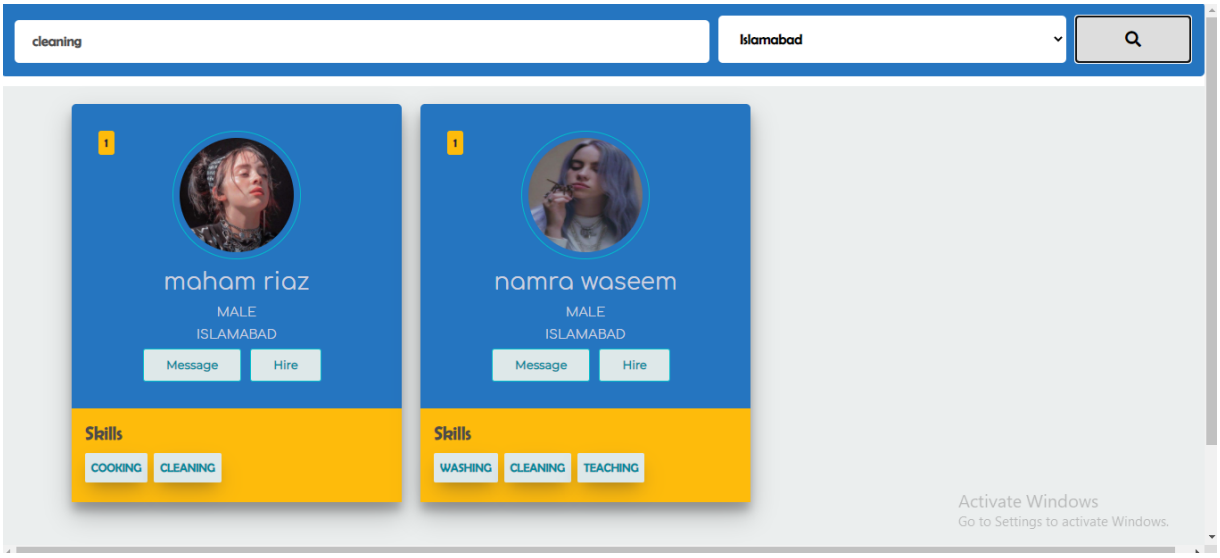

Figure 11 List of Helpers with related skills

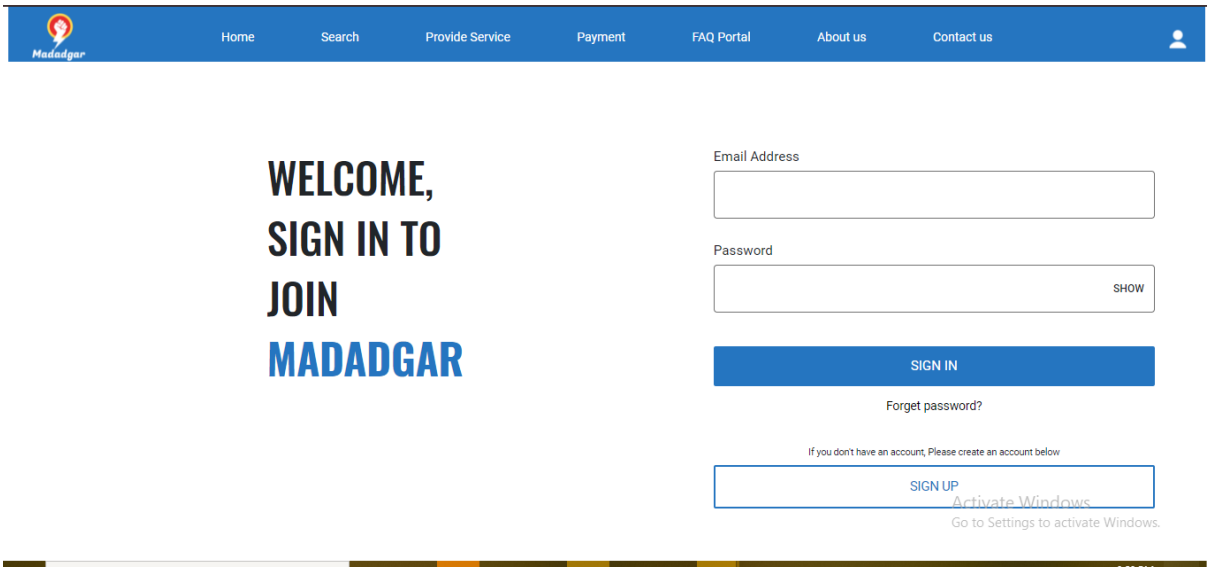

Figure 12 Login page

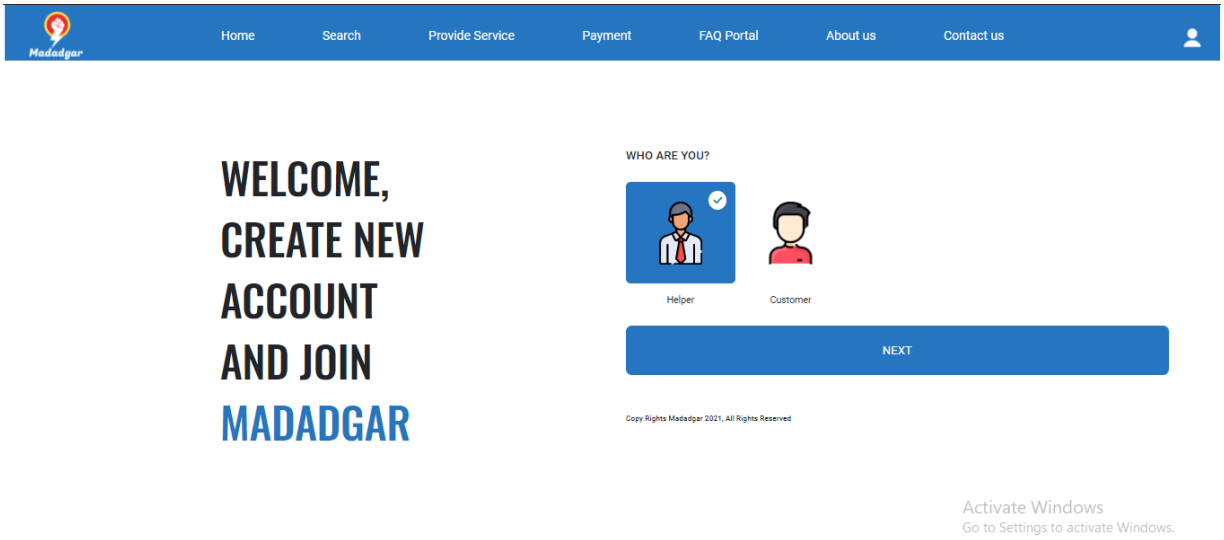

Figure 13 select user type page

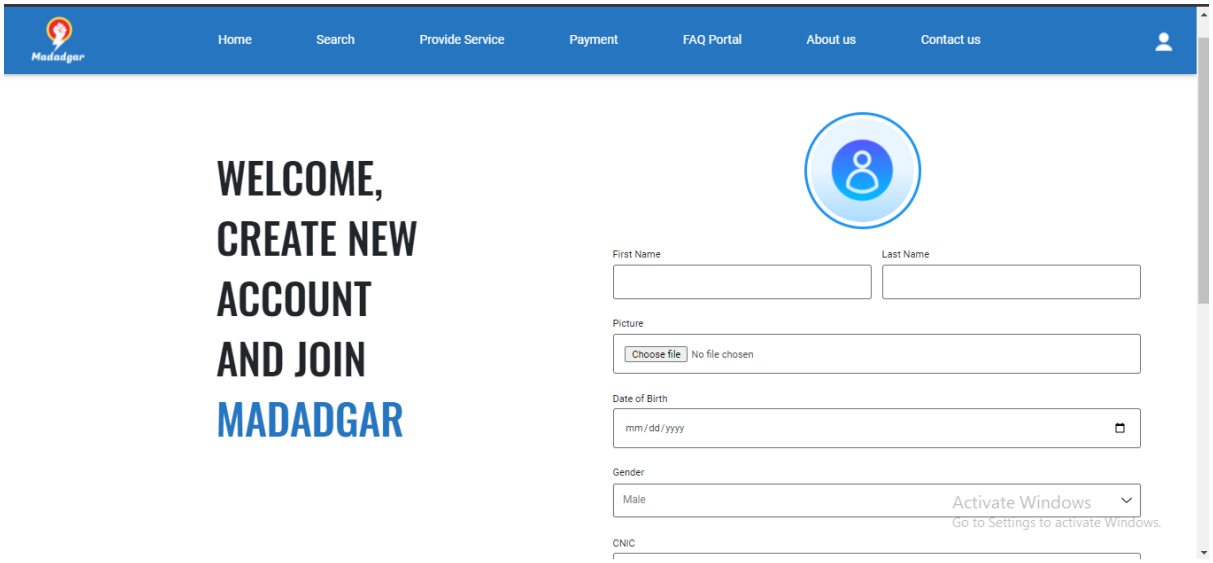

Figure 14 Sign up form

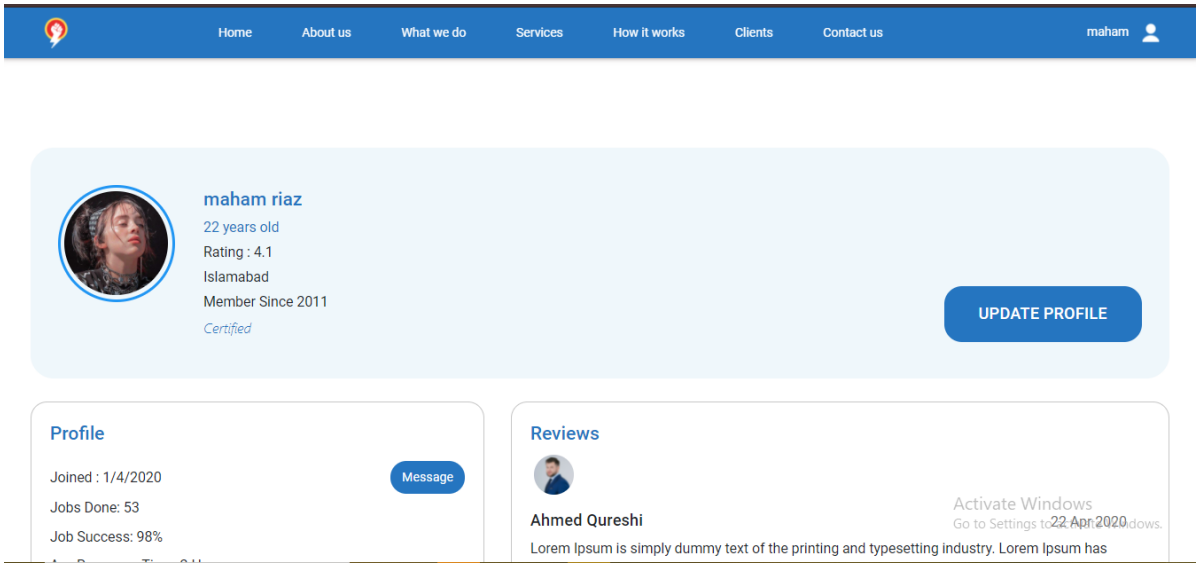

Figure 15 Helper's Profile

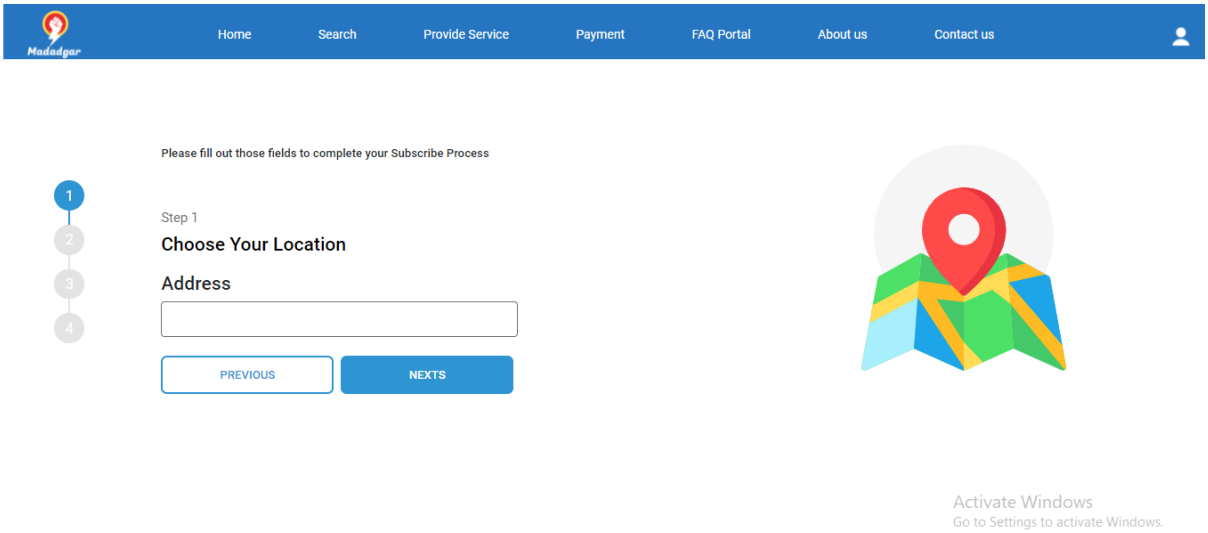

Figure 16 Payment

8.2. Screen objects and actions

Following is the discussion of screen objects and actions associated with those objects.

8.2.1. Homepage

This page will allow user to go to home page, get services page or FAQ portal directly. It will also allow user to create account and view feedback. They can select services, set location, view services and contact admin.

8.2.2. FAQ Portal

This screen will allow customers to view notifications, post their query, search any query, like and comment. Moreover it will allow service providers to view notifications, view all questions, post their answers, like and rate answers.

8.2.3. Customer's Portal

This page will allows customers to get services, set location, send and receive message, create contract, view notifications and post job request. They can edit their profile, search service provider, view their profile, view reviews, hire service provider, rate service provider, and file a complaint. This will also allow customer to go to FAQ portal directly. This page will also allow customers to log out their account.

8.2.4. Admin's Portal

This screen allows admin to view users, view and respond complaints, restrict user's account, view statistics, provide certification to service providers.

8.2.5. Service Provider's Portal

This screen allows service provider to select the services category, edit profile, upload certification, set location, add experience and add skills. They can chat with customer, view all job's requests, accept or decline job request, view ratings and feedback. They can view contract, accept or reject contract. This will also allow customer to go to FAQ portal directly.

Appendix I

- How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: http://agilemodeling.com/artifacts/

- How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link:
  http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf

- Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:
  - http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm ○ Software Engineering –A Practitioner's approach by Roger Pressman
- Architecture diagram: For guidance please follow the instructions mentioned in the link and book:
  - Ian Sommerville – Software Engineering 9th Edition– Chapter 6