



**03-134142-068** M. Umar Moeid

**03-134142-057** M. Awais

**03-134142-113** M. Haseeb Haider

## **Karigar Hub**

In partial fulfilment of the requirements for the degree of  
**Bachelor of Science in Computer Science**

Supervisor: Munazza Sher

Department of Computer Sciences  
Bahria University, Lahore Campus

June 2018



# Certificate



We accept the work contained in the report titled

“KARIGAR HUB”,

written by

M. Umar Moeid

M. Awais Aftab

M. Haseeb Haider

as a confirmation to the required standard for the partial fulfilment of the degree of  
Bachelor of Science in Computer Science.

Approved by:

Supervisor: Ms. Munazza Sher

\_\_\_\_\_  
(Signature)

June 4<sup>th</sup>, 2018



**DECLARATION**

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
<b>03-134142-068</b>	M. Umar Moeid	
<b>03-134142-057</b>	M. Awais Aftab	
<b>03-134142-113</b>	M. Haseeb Haider	

Date : \_\_\_\_\_



Specially dedicated to  
my beloved grandmother, mother and father  
(M. Umar Moeid)  
my beloved grandmother, mother and father  
(M. Awais Aftab)  
my beloved grandmother, mother and father  
(M. Haseeb Haider)





## ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, Miss Munazza Sher for her invaluable advice, guidance and his/her enormous patience throughout the development of the research.

In addition, we would also like to express our gratitude to our loving parent and friends who had helped and given us encouragement.

M. Umar Moeid  
M. Awais Aftab  
M. Haseeb Haider



## **KARIGAR-HUB**

### **ABSTRACT**

This project is about an android application along with a website that will help customers with solving their daily life home based problems, like someone in need of mechanic, plumber, carpenter all of those professional workers who are needed for daily home based tasks. People find it difficult in getting the right kind of professionals and waste a lot of their time and effort to reach them and have to book an appointment in order for their task to get done. All of these issues will be resolved by using our application which can be used through the use of technology known as an android phone, where customers can easily request for the worker(specified types of professionals) the request will be received by only those workers, who are linked with Karigar-Hub(android app). Then it will be the worker decision to accept or decline the request, only that worker will get allocated to the user who will accept the request and on both their mobile phones live tracking would be started so they can easily see each other's location. Customer will pay according to the time of his work, customer can also request day-based workers for day task.



## TABLE OF CONTENTS

<b>DECLARATION</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>vii</b>
<b>ABSTRACT</b>	<b>ix</b>
<b>TABLE OF CONTENTS</b>	<b>xi</b>
<b>LIST OF FIGURES</b>	<b>xv</b>
<b>LIST OF SYMBOLS / ABBREVIATIONS</b>	<b>xvii</b>

### CHAPTERS

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
 <b>2</b>	 <b>Software Requirement Specification</b>	 <b>5</b>
	2.1 Description	5
	2.1.1 User Classes and Characteristics	5
	2.1.2 Operating Environment	6
	2.1.3 Design and Implementation and Constraints	6
	2.1.4 Assumption and Dependencies	6
	2.2 System Use Cases	6
	2.2.1 Register (U1)	8
	2.2.2 Login (U2)	8
	2.2.3 Edit Account (U3)	9

2.2.4	Logout (U4)	10
2.2.5	Request for Worker (U5)	10
2.2.6	Live Tracking (U6)	11
2.2.7	Generated Request for Worker (U7)	12
2.2.8	Reviews (U8)	13
2.2.9	Billing (U9)	14
2.2.10	Start/End of Work (U10)	15
2.3	Other Non-functional Requirements	16
2.3.1	Performance Requirements	16
2.3.2	Safety Requirements	16
2.3.3	Security Requirements	16
2.3.4	Software Quality Attributes	16
2.4	Other Requirements	17
2.4.1	Response Time	17
2.4.2	Accuracy of Interpretation	17
2.4.3	Battery Life	17
2.5	System Requirements Chart	18
3	<b>DESIGN AND METHODOLOGY</b>	<b>19</b>
3.1	Design	19
3.1.1	Domain Model	20
3.1.2	Sequence Diagram	20
3.1.4	Collaboration Diagram	24
	<b>3.1.4.1 Login</b>	<b>24</b>
	<b>3.1.4.2 Register</b>	<b>24</b>
3.1.5	Design Class Diagram	26
3.1.6	Data Model	27
4	<b>IMPLMENTATION</b>	<b>33</b>
4.1	<b>DATABASE</b>	<b>33</b>

4.1.1	Karigar	33
4.1.2	User	34
4.1.3	Karigar available	34
4.1.4	Customer Requests	34
4.1.5	Working Karigar	34
4.1.6	Work complete	34
4.1.7	Advantages for using Firebase.	35
4.2	Software used for Android development:	36
4.2.1	Android studio:	36
4.2.2	JSON	36
4.3	Errors during Android Development:	36
4.3.1	Firebase Connectivity Error:	36
4.3.2	Firebase retrieval error:	37
4.3.3	Module Integration Error:	37
4.3.4	Linking a java file with multiple Xml files:	37
4.3.5	Auto Acceptation of Customers Request:	37
4.3.6	Extending an Activity:	38
4.4	Tools used for web development	38
4.4.1	PHPStorm:	38
<b>5</b>	<b>USER MANUAL</b>	<b>39</b>
<b>6</b>	<b>CONCLUSION AND RECOMMENDATIONS</b>	<b>57</b>
6.1	Conclusion:	57
6.2	Recommendations:	57
6.2.1	Strong internet connection:	57
6.2.2	Free mobile storage:	57
6.2.3	Running out of battery:	58
6.2.4	Work satisfaction level:	58
	<b>REFERENCES</b>	<b>59</b>

**LIST OF TABLES**

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
Table 2-1	Register	8
Table 2-2	Login	9
Table 2-3	Edit Account	10
Table 2-4	Logout	10
Table 2-5	Request for Worker	11
Table 2-6	Live Tracking	12
Table 2-7	Generated Request for Workers	13
Table 2-8	Reviews	13
Table 2-9	Billing	14
Table 2-10	Start/End of Work	15
Table 2-11	System Requirements	18



**LIST OF FIGURES**

<b>FIGURE</b>	<b>TITLE</b>	<b>PAGE</b>
Figure 2-1	Use Case Diagram	7
Figure 3-1	(Refined and Updated Use Case)	19
Figure 3-2	(Domain Model)	20
Figure 3-3	(Sequence Diagram: Login)	21
Figure 3-4	(Sequence Diagram: Register)	22
Figure 3-5	(Sequence Diagram: Reviews)	23
Figure 3-6	(Sequence Diagram: Logout)	23
Figure 3-7	(Collaboration Diagram: Login)	24
Figure 3-8	(Collaboration Diagram: Register)	24
Figure 3-9	(Collaboration Diagram: Reviews)	25
Figure 3-10	(Collaboration Diagram: Logout)	25
Figure 3-11	(Class Diagram)	26
Figure 3-12	(Fill in Cardinality)	28
Figure 3-13	(Fully Attributed ERD)	30
Figure 3-14	(Rapid Application Development)	31

**LIST OF SYMBOLS / ABBREVIATIONS**

GPS	Global Positioning System
API	Application Programming Interface
GB	Gigabyte
RAM	Random-Access Memory
MB	Megabyte
GUI	Graphical User Interface
UML	Unified Modelling Language
RAD	Rapid Application Development
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
SDK	Software Development Kit
APK	Android Package Kit
PHP	Personal Home Page



## CHAPTER 1

### INTRODUCTION

#### 1.1 Background

Mobile has become a basic contact tool, everyone prefers to possess and take it with them. This technology has established a base to conquered the traditional desktop based approach. With advancement in the field of technology and with the development of android phones web and android applications are gaining popularity amongst the people. Mobile application that tend be helpful in daily life becomes the habit of user. Android a mobile based Operating system developed by Google Company based upon Linux Kernel designed for mobiles with touch screens like smartphones and tablets.

Our project name is “Karigar-Hub”. The name demonstrating its significance is kind of hub for handymen. Karigar-Hub is based upon android application and a website, through which we are providing and delivering the professional, helpful and high-quality services at home and in any area where handymen are linked with the application. The concept is derived from UBER (Transport Company). Karigar-Hub allowing customers to book right kind of handyman for their job through their mobile phones easily meeting the needs of many customers, without wasting their time. Mostly people get difficulty in finding the right kind of handyman for the task as it consumes a lot of time to find them. If the innovation can do these works effectively then why to waste the time.

## **1.2 Problem Statements**

Nowadays people has become extremely busy in their lives and have very less time for their house held problems. People get to much difficulty in finding right kind of handymen to resolve their problems and often result in wasting a lot of time and effort to find themselves a suitable handymen for their desired job and even after finding them they have to wait for long appointments of the worker and even after the completion of work handymen desires fee according to his own will. There is no any pre-set standards for this.

## **1.3 Aims and Objectives**

Our main objective is to provide services to the people to save their time through the use of innovation by proving them layman services to solve their daily based house held problems through android application.

## **1.4 Scope of Project**

Through our services we hope to accomplish following major goals:

Customer's benefits:

- Time saving.
- Quality assurance.
- Consistent and timely service.
- Customer supportive.
- Customer comfort.
- User friendly.

Workers benefit:

- Employment.
- More earning.
- Better family support.

Service providers benefit:

- Business startup.
- Commission based profits.
- Research based learning activity.



## CHAPTER 2

### Software Requirement Specification

Location Tracking is becoming very important in this era, it is helping to manipulate the location of someone's home for this location tracking GPS(Global Positioning System) is used, GPS is a navigation system using worldwide, GPS provides outdoor locations with very high accuracy.<sup>[1]</sup> In Karigar-Hub GPS is used with google map API. Google Map helps to identify the location with custom markers, augmenting the map data with image overlays, embedding one or more maps as fragments, and many such implementations help the average Android user <sup>[2]</sup>. In Karigar-Hub customers and workers both have different interface that why google map is helping us to make it different for them.

#### 2.1 Description

##### 2.1.1 User Classes and Characteristics

There would be three types of user which will interact with the application Customers, Workers and Administrative. So, all of them would have different type of requirements. Customer will request for a worker from different categories. The request would be send to all nearest workers in some kilometres range. If the workers accept the customer request it will generate the live map tracking on both side. So, customer can see the worker is coming to his location. Customer side helps them for scheduling of worker to manage the multiple workers at a time. While on the worker side the request generates, and the worker



would have the authority to accept or not, if it is not available it can change his sign to offline. Alternatively, if the worker is linked with some other customer through Karigar-Hub so he will not get any further request until he freed from his work. Administration panel will be on the web portal to monitor each both worker and customer ends.

### **2.1.2 Operating Environment**

It will be android based app which must have android version 6.0 marshmallow or greater than this. Phone should have at least 1GB of RAM and 100MB of free space to enjoy the best feature of Karigar-Hub. For admin panel the minimum core 2 duo laptop or personal computer with having minimum 2MB internet speed. Webhosting services would be required a domain and a server to monitor both app and web portal.

### **2.1.3 Design and Implementation and Constraints**

The main constraint for our project implementation is integration. To integrate the whole application at the end generates too many bugs which was difficult to handle. Karigar-Hub is based on two tier android applications and one website, which needs proper internet connection at every time; otherwise, no one can use it.

### **2.1.4 Assumption and Dependencies**

It is assumed that the handymen in Pakistan would like to work through mobile phone they would be familiar with the applications and familiar with internet usage. The whole application is internet based so there is a need of internet every time. It will be expected that the worker and clients will have fair internet.

## **2.2 System Use Cases**

Use case is a procedure, which is used to determine, define and set up the system requirements. The use case should consist of all system activities that is related

to users. Use case should organize the functional requirements, actors (users) and Records paths from trigger events to goals.

Several system activities involved in our project, these activities and with their respective actors shown in the following diagram, which is explaining the working scenario of Karigar-Hub. Each of the system activity is further explained in their unique tables.

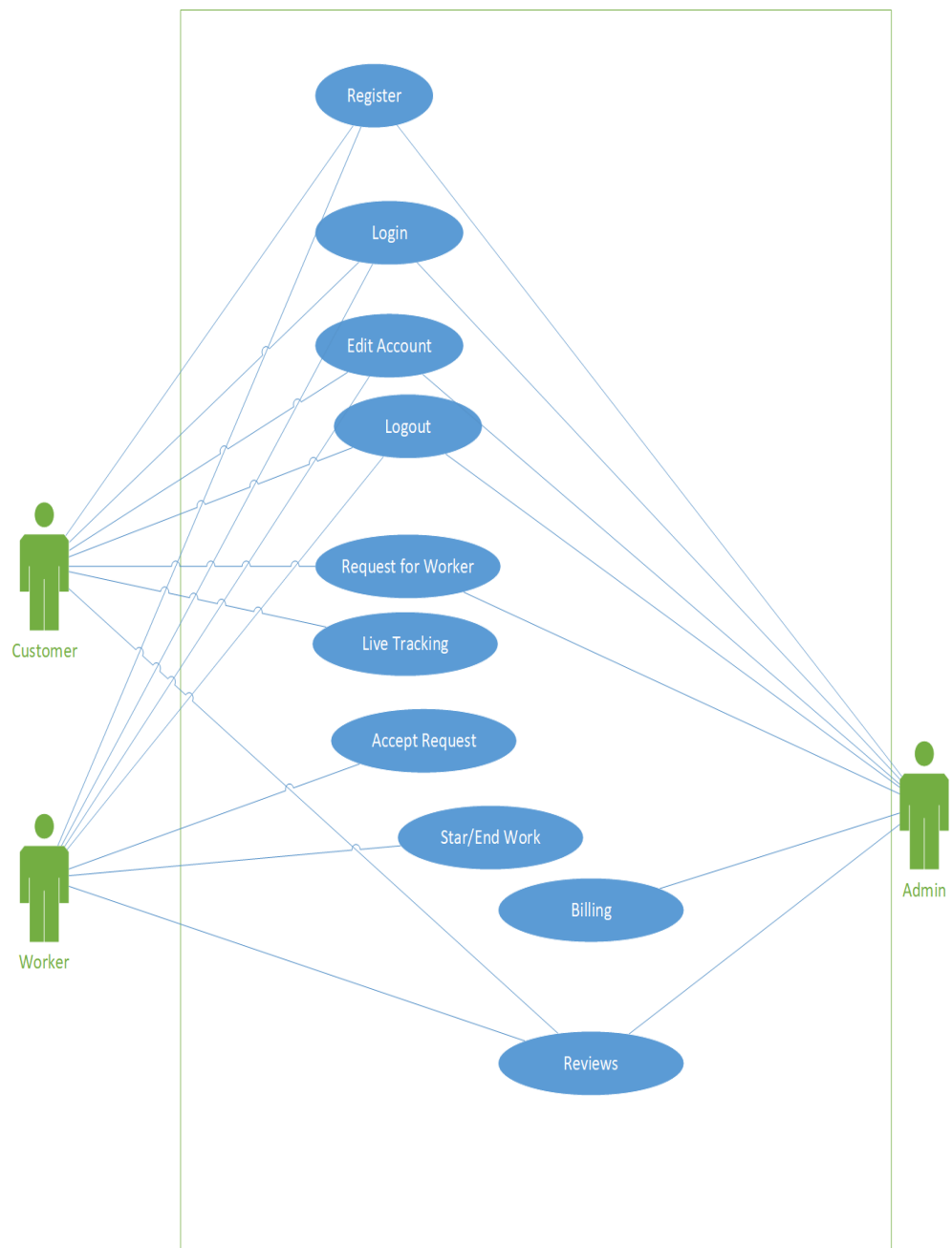


Figure 2-1 Use Case Diagram

### 2.2.1 Register (U1)

<b>Unique Identifier</b>	<b>U1</b>
<b>Objective</b>	To have customer/worker correct information and validated mobile number. So, we can maintain the record of each customer/worker. It will help us to manage users by their recent activities.
<b>Priority</b>	High
<b>Source</b>	Customer and Worker.
<b>Actors</b>	Customer and Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>Basic Flow</b></p> <ol style="list-style-type: none"> <li>1. User(Customer/Worker) fill the required fields.</li> <li>2. Validating data (User enter new number for registration).</li> <li>3. User redirected to main screen, ready to use application services.</li> </ol> <p style="text-align: center;"><b>Alternative Flow</b></p> <ol style="list-style-type: none"> <li>1. Registered user cannot register again with the same number.</li> <li>2. If user already exist, then app will move to login screen.</li> </ol>
<b>Includes</b>	None
<b>Pre-Conditions</b>	User must be connected with internet and active mobile number.
<b>Post-Conditions</b>	Then he can easily log in or log out from any android device.

Table 2-1 Register

### 2.2.2 Login (U2)

<b>Unique Identifier</b>	<b>U2</b>
<b>Objective</b>	To secure customer/worker information and to validate that the customer /worker login to their own account and not others
<b>Priority</b>	High
<b>Source</b>	Customer and Worker.
<b>Actors</b>	Customer and Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. User (Customer/Worker) enter the login credentials.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Validating data (checking whether the provided credentials are correct or not).</li> <li>3. User redirected to main screen, of their account.</li> </ol> <p style="text-align: center;"><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. User cannot use same credentials for another account</li> </ol>
<b>Includes</b>	U1
<b>Pre-Conditions</b>	User must be connected with internet and active mobile number.
<b>Post-Conditions</b>	User can easily log in or log out from any android device.

Table 2-2 Login

### 2.2.3 Edit Account (U3)

<b>Unique Identifier</b>	<b>U3</b>
<b>Objective</b>	To view users (Customer/Worker) info and to make changes to the users request user (customer/worker).
<b>Priority</b>	High
<b>Source</b>	Admin.
<b>Actors</b>	Customer and Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. Admin login using his or her credentials.</li> <li>2. Validating data (checking whether the provided credentials are correct or not).</li> <li>3. Views user's info</li> <li>4. Edits users (Customer/worker) request.</li> </ol> <p style="text-align: center;"><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. Admin can view customers data but cannot change their private data like their mobile number and age etc.</li> </ol>
<b>Includes</b>	U2
<b>Pre-Conditions</b>	Admin must be connected to internet.
<b>Post-Conditions</b>	Admin easily log in or log out from any device with an active internet connection

Table 2-3 Edit Account

## 2.2.4 Logout (U4)

<b>Unique Identifier</b>	<b>U4</b>
<b>Objective</b>	To logout of the system so it cannot be misused by any user (worker/customer).
<b>Priority</b>	High
<b>Source</b>	Admin.
<b>Actors</b>	Customer and Worker
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. Customer and Worker will click on the logout button their session would be finished.</li> <li>2. Customer will move to the front activity of the application where they have to login again.</li> <li>3. On that activity new user can also register but at one time only one user can use the application on single device.</li> </ol> <p style="text-align: center;"><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. Admin can view customers data but cannot change their private data like their mobile number and age etc.</li> </ol>
<b>Includes</b>	U2
<b>Pre-Conditions</b>	Customer must be connected with internet. Customer must be logged in.
<b>Post-Conditions</b>	User can easily log in or logout to any other device.

Table 2-4 Logout

## 2.2.5 Request for Worker (U5)

<b>Unique Identifier</b>	<b>U5</b>
<b>Objective</b>	To request for the right kind of worker for home based task using the application.
<b>Priority</b>	High

<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. Customer login using his or her credentials.</li> <li>2. Customer selects the worker type and generates a request</li> <li>3. The request goes to all nearby workers that fall in the mentioned category</li> </ol> <p style="text-align: center;"><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. Customers can request one or more than one workers at the same time.</li> </ol>
<b>Includes</b>	U2
<b>Pre-Conditions</b>	Customer must be connected with internet. Customer should know the type of work that is being demanded by the worker.
<b>Post-Conditions</b>	Customer will be moved to live tracking and worker will be linked with the customer. Both the customer and worker would have option to cancel customer's request, but cancellation would generate the penalty to those who will cancel it.

Table 2-5 Request for Worker

### 2.2.6 Live Tracking (U6)

<b>Unique Identifier</b>	<b>U6</b>
<b>Objective</b>	To let workers, attain location of customers and vice-versa using live tracking system.
<b>Priority</b>	High
<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. The application lets the customers and the workers locate the location of each other using the live tracking system</li> </ol>

	<ol style="list-style-type: none"> <li>2. The worker reaches the customers location</li> <li>3. The worker performs their desired tasks.</li> <li>4. The worker gets its amount after performing its work</li> </ol> <p><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. Live tracking of more than one workers cannot be done simultaneously</li> </ol>
<b>Includes</b>	U5
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. The user must have an active internet connection</li> <li>2. The user must be logged in.</li> </ol>
<b>Post-Conditions</b>	Admin easily log in or log out from any device with an active internet connection

Table 2-6 Live Tracking

### 2.2.7 Generated Request for Worker (U7)

<b>Unique Identifier</b>	<b>U7</b>
<b>Objective</b>	To let the customer, generate a request to the worker through the use of application.
<b>Priority</b>	High
<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. The customer log in</li> <li>2. customer checks the type of worker</li> <li>3. customer generates a request for a worker through the application</li> </ol> <p><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. Only a single worker can accept the customer's request at a time.</li> </ol>
<b>Includes</b>	U5
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. The user must have an active internet connection</li> <li>2. The user must be logged in.</li> </ol>

<b>Post-Conditions</b>	Customer easily log in or log out from any device with an active internet connection
------------------------	--

Table 2-7 Generated Request for Workers

### 2.2.8 Reviews (U8)

<b>Unique Identifier</b>	<b>U8</b>
<b>Objective</b>	To let admin, know about the satisfaction of work the worker has delivered to the customer.
<b>Priority</b>	High
<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. After the work is done the customer lets the admin know that how much satisfied he is from the work done by the worker.</li> <li>2. If a worker is unable to satisfy a specific number of customers regarding his work he can be blocked to use the application by the admin.</li> </ol> <p><b>ALTERNATIVE FLOW</b></p> <ol style="list-style-type: none"> <li>1. A customer can give only one review for a worker.</li> </ol>
<b>Includes</b>	U2 and U4
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. The customer must be logged in to write a review about the worker</li> <li>2. The worker must have completed the work before writing a review against his work satisfaction.</li> </ol>
<b>Post-Conditions</b>	The customer should be notified that the complaint has been received and is under consideration. The complaint no should be allotted to the customer.

Table 2-8 Reviews



## 2.2.9 Billing (U9)

Unique Identifier	U9
<b>Objective</b>	To let the application, generate bill according to the work type and billing type (day-based or time based) and charges per hour in case of time based billing.
<b>Priority</b>	High
<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. The customer login to his account.</li> <li>2. The customer requests worker.</li> <li>3. The worker arrives and performs the work</li> <li>4. System generates bill on the workers screen according to date based or time-based feature and the standard charges for the type of work. The notification is generated to both the customer and the admin.</li> </ol>
<b>Includes</b>	U10
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. The user(worker/customer) must have an active internet connection</li> <li>2. The user must be logged in.</li> </ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>1. Work will be completed.</li> </ol>

Table 2-9 Billing

## 2.2.10 Start/End of Work (U10)

<b>Unique Identifier</b>	<b>U10</b>
<b>Objective</b>	Let the user know when the work has started and ended.
<b>Priority</b>	High
<b>Source</b>	Customer/Worker.
<b>Actors</b>	Customer/Worker.
<b>Flow of Event</b>	<p style="text-align: center;"><b>BASIC FLOW</b></p> <ol style="list-style-type: none"> <li>1. The customer login to his account</li> <li>2. The customer requests worker</li> <li>3. The worker arrives at customers location</li> <li>4. The customer says worker to start the work</li> <li>5. The worker clicks on start working button on its screen and starts to work.</li> <li>6. The worker completes his work and clicks on finish work in order to generate payment based upon his work type and duration spent on work.</li> <li>7. The worker gets its amount after performing its work</li> </ol>
<b>Includes</b>	U5 and U6
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. The user must have an active internet connection</li> <li>2. The user must be logged in.</li> </ol>
<b>Post-Conditions</b>	<ol style="list-style-type: none"> <li>1. After clicking the start work button time of work will be started</li> <li>2. After the end of work payment will be generated.</li> </ol>

Table 2-10 Start/End of Work

## **2.3 Other Non-functional Requirements**

### **2.3.1 Performance Requirements**

Karigar-Hub application requires a GPS enabled android phone to work with at least version 6.0 marshmallow and a minimum of dual core CPU and 1GB of RAM and an active GPRS connection. GPS is top priority technology use for locating a device position accurately <sup>[1]</sup>.

### **2.3.2 Safety Requirements**

To ensure that the application is working effectively without any bugs the developer team updates the application after every two weeks. There is also a report bug feature available where users can report any bugs they have encountered while using the application so the developers can resolve the issue or issues.

### **2.3.3 Security Requirements**

Karigar-Hub does not have any security requirements and thus any type of user can use it without any additional privileges.

### **2.3.4 Software Quality Attributes**

Karigar-Hub provides the users (customers/workers) with some unique features like time based and day based feature. Due to its well designed and easy to use interface it can be used both by experts and typical users however users (workers) must have basic knowledge of English for customers and Urdu language for workers and know how to write in Urdu before using the application.

## **2.4 Other Requirements**

### **2.4.1 Response Time**

The response time should not be more than 5 seconds although the wish is that it should not be more than 1 percent 100 percent of the time.

### **2.4.2 Accuracy of Interpretation**

Application must maintain at least 70 percent accuracy although it should be 100 percent accuracy at most.

### **2.4.3 Battery Life**

The battery of the device should last for at least 50 minutes.

## 2.5 System Requirements Chart

Requirement No	Priority	Type	Source	Contained in Use Case	Description
SR1	High	Functional	Customer/Worker	U1, U2, U3	Customer and worker both will be register and then login to use the services of Karigar-Hub. They can easily change their details like mobile number name etc.,
SR2	High	Functional	Customer	U4	Customer Request for worker on the pointed location.
SR3	High	Functional	Customer/Worker	U5	Both the worker and customer will be linked through live mapping, it will help the worker to find the customer location easily and customer can view easily where the customer reached.
SR4	Medium	Functional	Customer	U7	Customer will provide the review about worker, it will help the administrator to upgrade or down gradation of worker.
SR5	High	Functional	Worker/Customer	U9	Worker will start and end the work, it will help to generate the payment of specific work.
SR6	High	Functional	Worker	U8, U9	Bill will be generated after the end of work.

Table 2-11 System Requirements

## CHAPTER3

### DESIGN AND METHODOLOGY

#### 2.1 Design

##### UsecaseDiagram

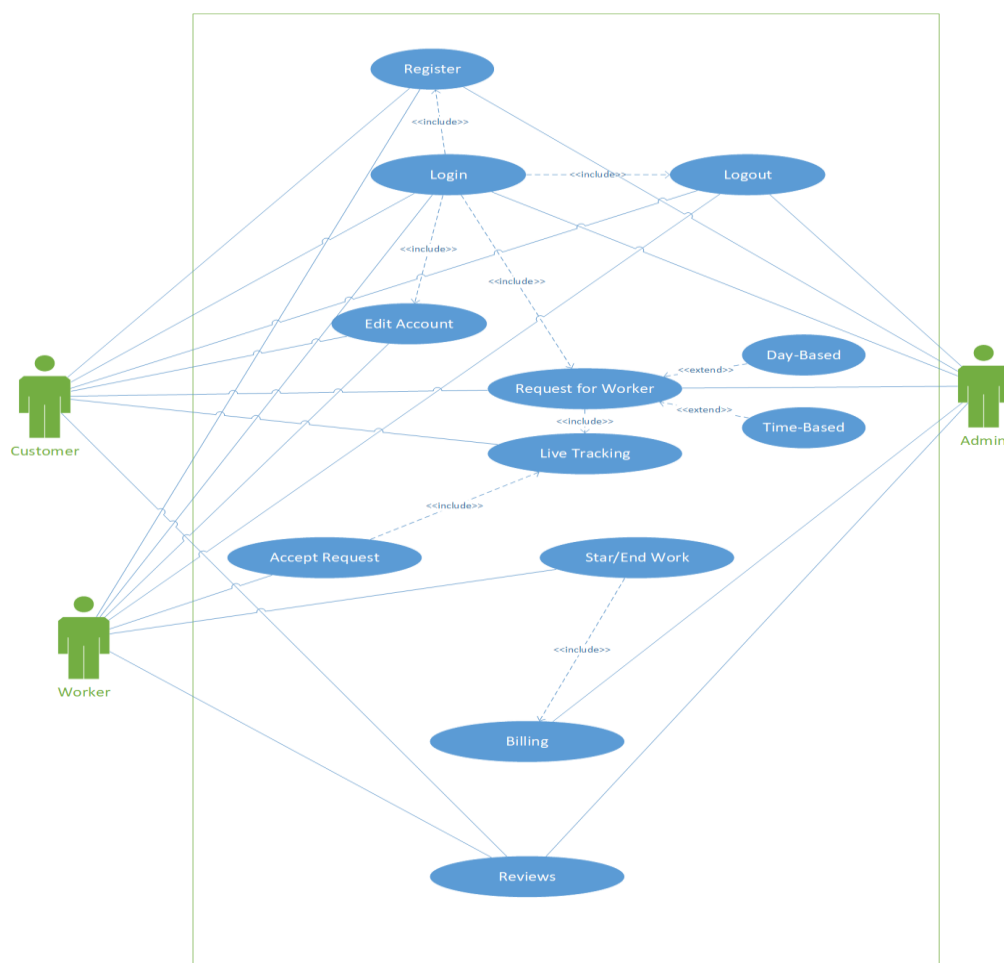


Figure 0-1 (Use Case Diagram)

**Include:** Include is a straight link between two use cases which shows that the action of one use case is added into the action of the base use case.

**Extend:** Extended use case is meaningful on its own, it is independent of the extending base use case.

### 2.1.1 Domain Model

Domain model is a conceptual model of the domain that incorporate both behaviour and data, domain model is used for lower level layer for persistent and publishes

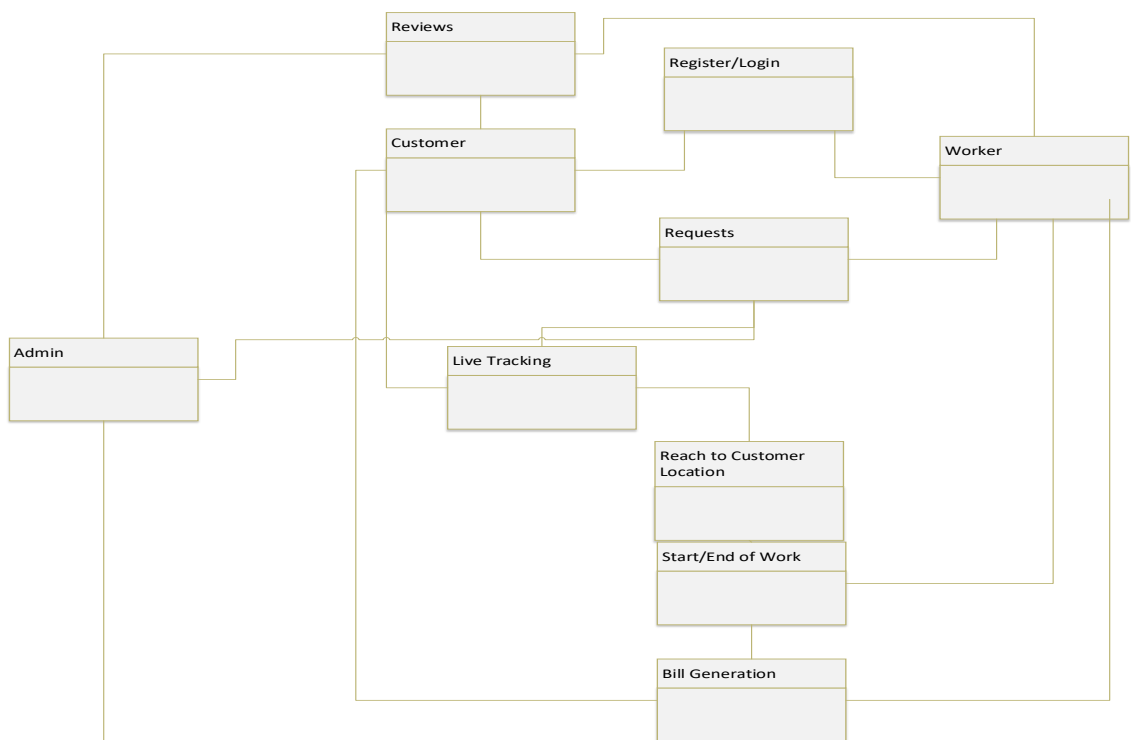


Figure 0-2(Domain Model)

### 2.1.2 Sequence Diagram

A sequence diagram shows interaction between objects arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence in which the messages are exchanged between the objects that need to carry out the functionality of the scenario.

### 2.1.2.1 Login:

The diagram (diagram number) depicts that firstly the mobile number and password are submitted in the login GUI then after that the username and password entered are matched with the entire Database if the login credentials match with the database credentials then login is a success otherwise login is a failure .If login is a success the user is provided with his Login Home Page and his Entire Credentials.

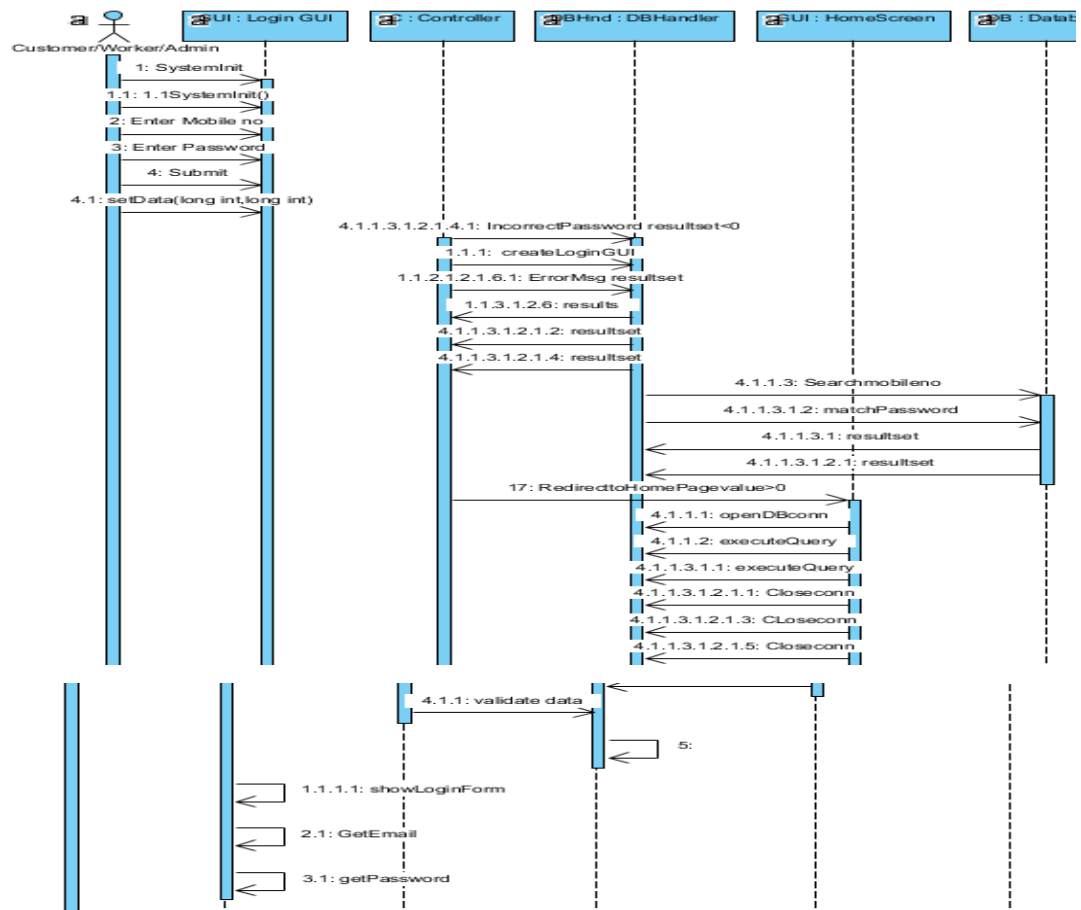


Figure 0-3 (Sequence Diagram: Login)



### 2.1.2.2 Register:

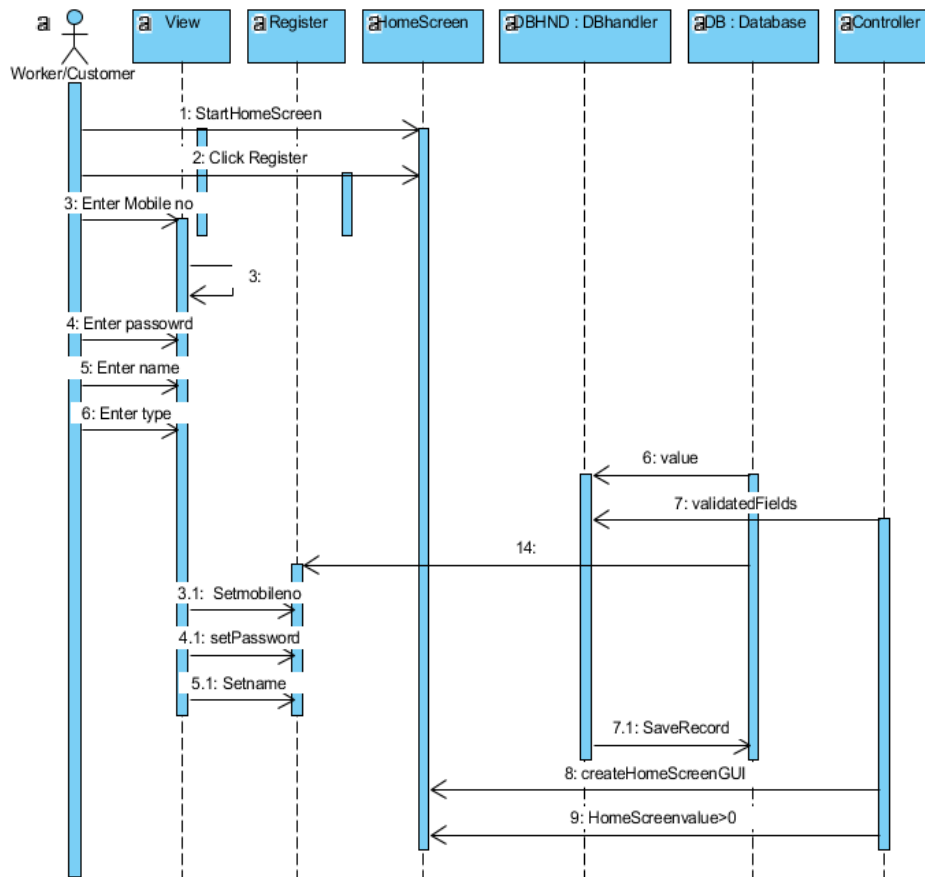


Figure 0-4(Sequence Diagram: Register)

The above picture depicts that firstly you have to click on the register button after that you have to enter the credentials like your mobile no, password, name and type and after that the values are set and values are stored in the database and from there the credentials are validated, if credentials are validated Home Screen GUI is created against that particular account.

### 3.1.3.3 Reviews

This picture depicts that customer has to be logged in in order to generate feedback. First of all we enter a mobile number the system validates the mobile number from user database after that database returns result set then you enter the password and password is again validated from the user database the result

set is returned if the result set is valid then you can generate feedback from the user feedback system after that the feedback is stored in the users database.

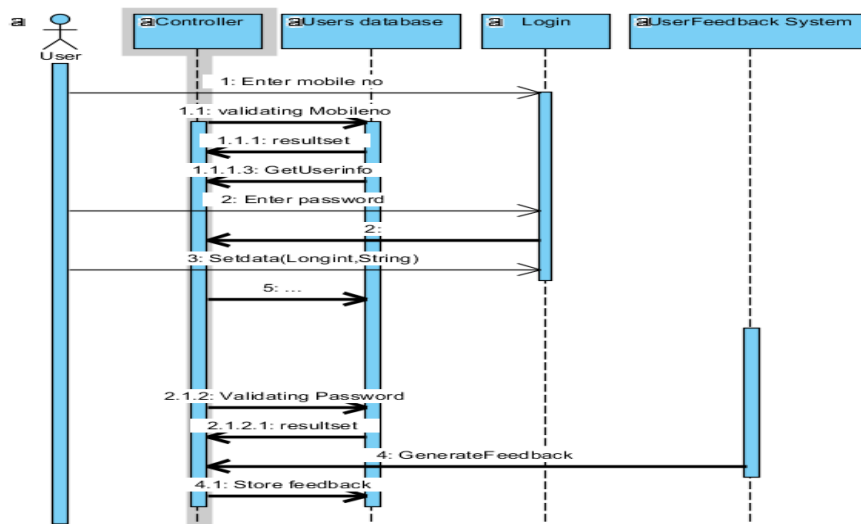


Figure 0-5(Sequence Diagram: Reviews)

### 3.1.3.4 Logout:

The Diagram depicts that when a customer/worker/Admin clicks on the logout button the controller moves the control to the login GUI where the logout function is called and the customer /worker/Admin is redirected to the Admin GUI Home Screen.

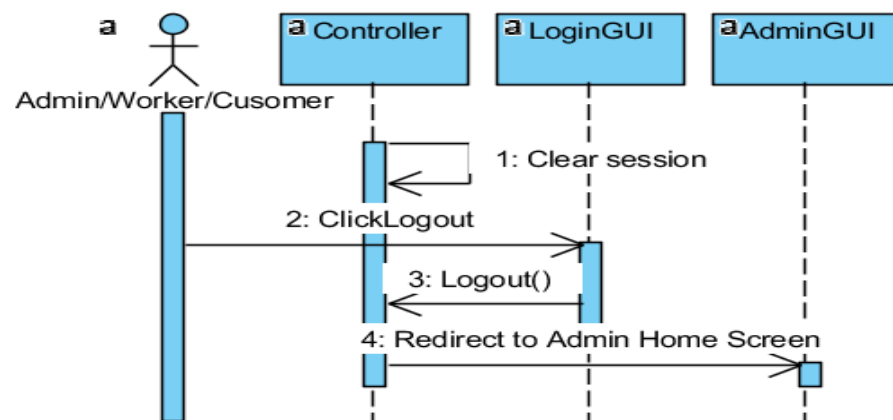


Figure 0-6 (Sequence Diagram: Logout)

### 3.1.4 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is a depiction of the relationships and interactions amongst the software objects in Unified Modelling Language (UML).

#### 3.1.4.1 Login:

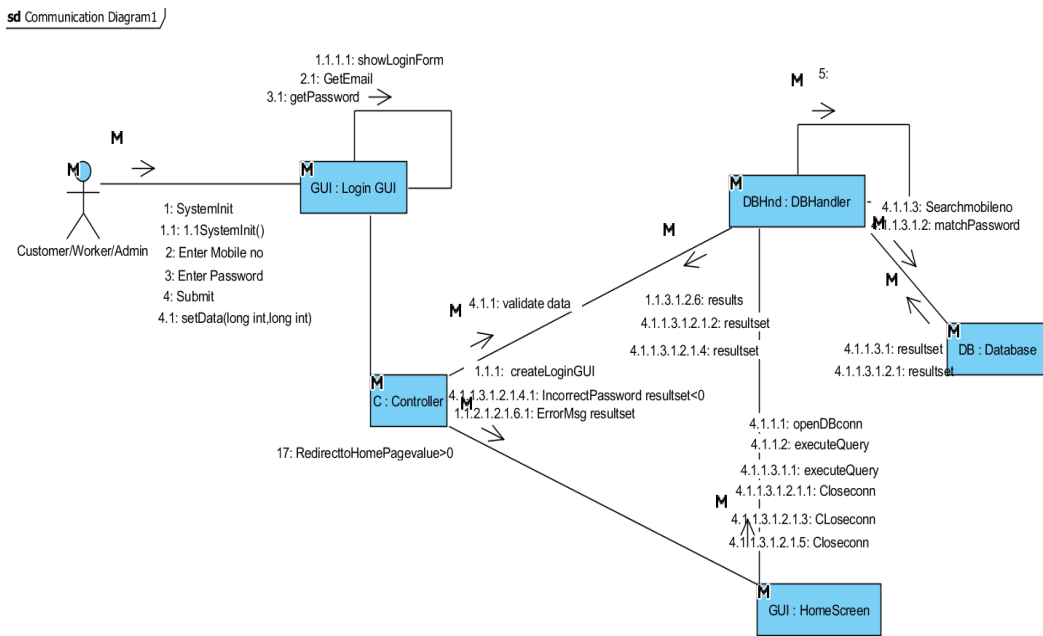


Figure 0-7(Collaboration Diagram: Login)

#### 3.1.4.2 Register:

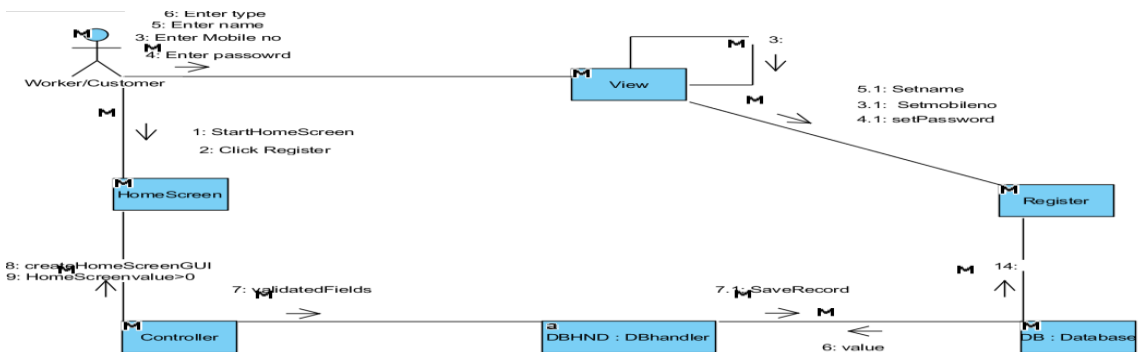


Figure 0-8(Collaboration Diagram: Register)

### 3.1.4.3 Reviews:

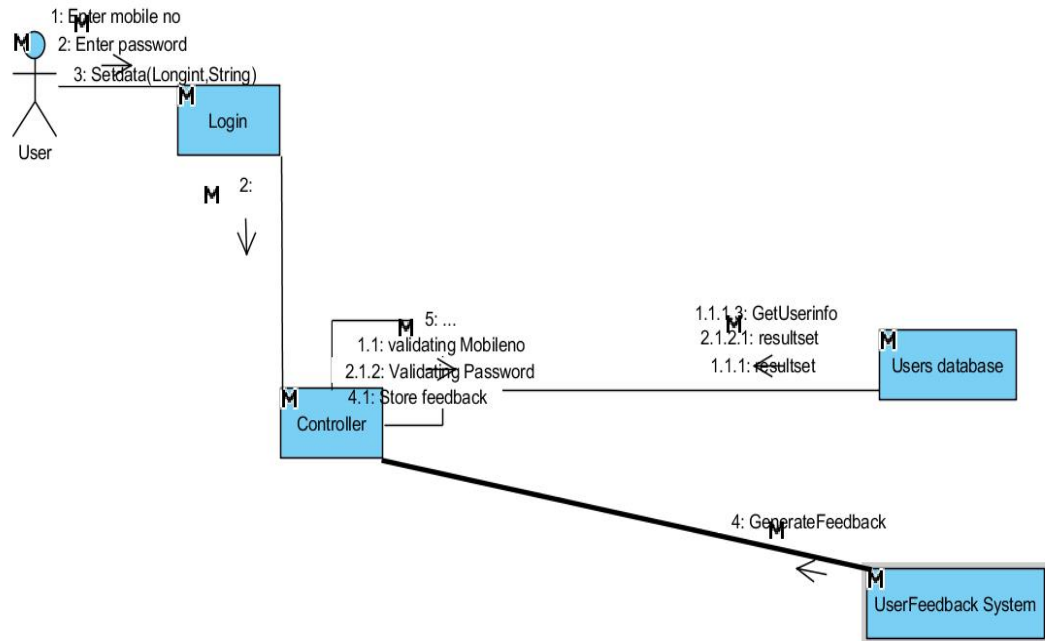


Figure 0-9(Collaboration Diagram: Reviews)

### 3.1.4.4 Logout:

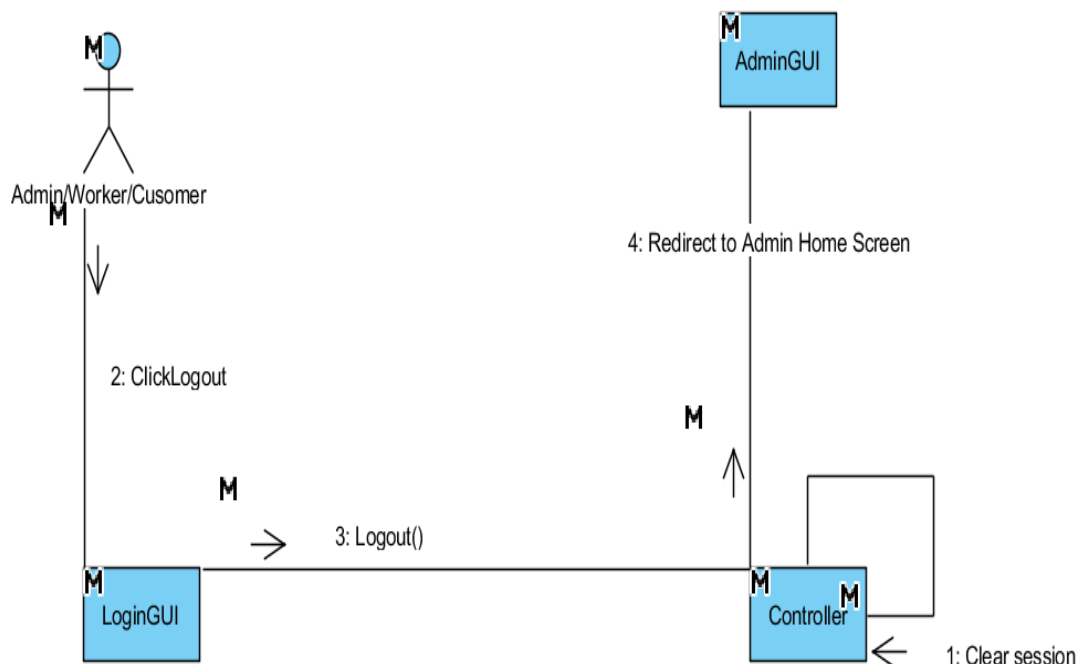


Figure 0-10(Collaboration Diagram: Logout)

### 3.1.5 Design Class Diagram

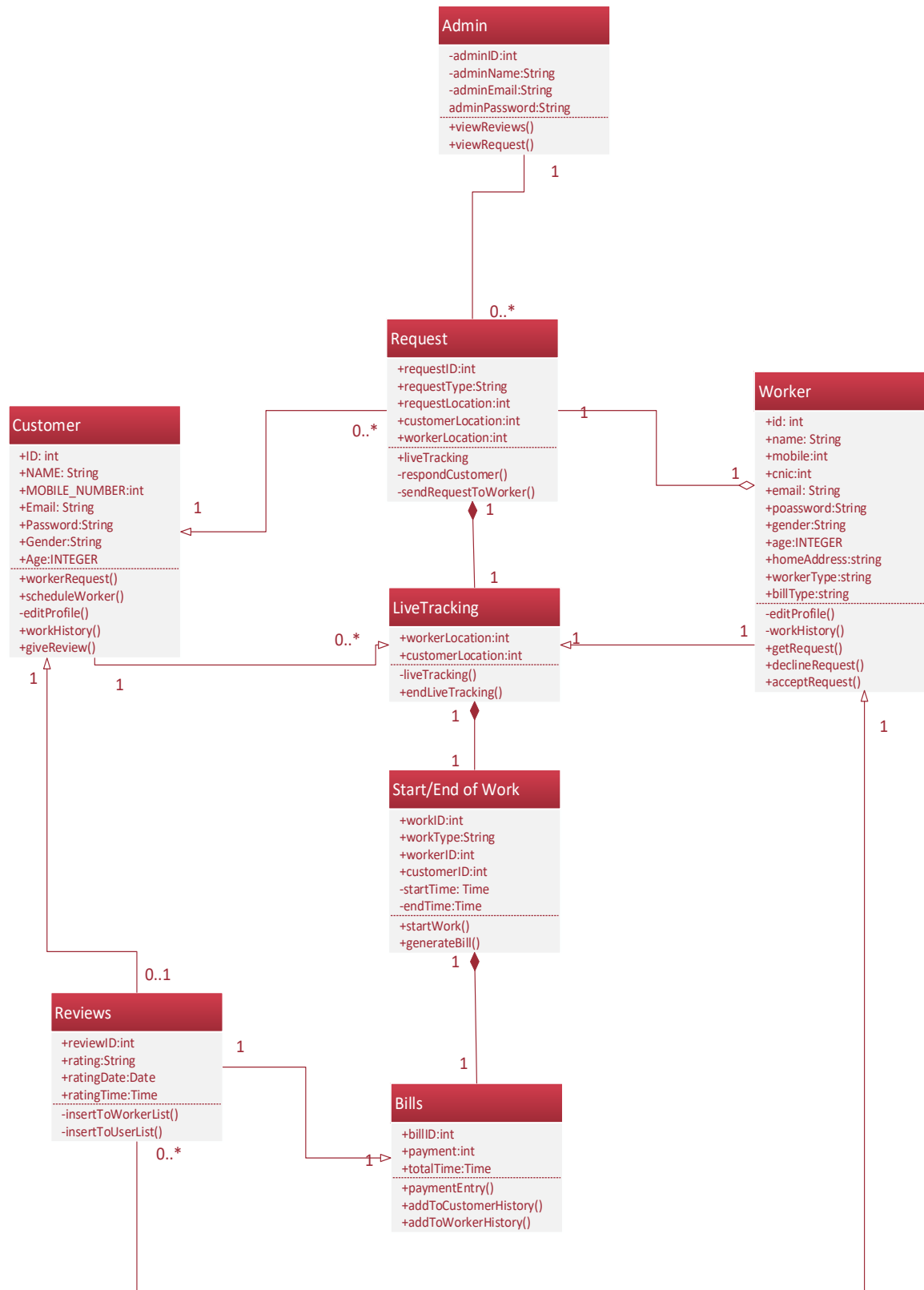


Figure 0-11 (Class Diagram)

## Description

1. In the figure above the Admin class has adminId of type id,adminName of type string ,adminEmail of type string and adminPassword of type string
2. Request class contains requestID of type int ,requestType of type string,requestLocation of type int,customerLocation of type int and workerLocation of type int.
3. Customer class contains ID of type int Name of type string ,Email of type string ,MOBILE\_NO of type string ,Password of type string,Gender of type string,Age of type int
4. Live tracking class contains customerLocation and workerLocation both of type int.
5. Worker class contains id of type int ,name of type string ,mobile of type int ,cnic of type int,email of type string ,password of type string,gender of type string age of type int ,homeAddress of type string,workerType of type string ,billType of type string.
6. Start/End of Work class contains workId of type int , workType of type string , workerID of type int ,customerId of type int ,startTime of type time ,endTime of type time.
7. Reviews class contains reviewID of type int,rating of type string ,ratingDate of type date and ratingTime of type time.
8. Bills class contains billID of type int,totalTime of type time and payment of type int.

### 3.1.6 Data Model

#### 3.1.6.1 Identify Entities

Customer, Worker, Request, Work, Bill, Admin and Reviews are the entities of data model.

### 3.1.6.2 Find Relationships

	Customer	Worker	Request	Work	Bill	Reviews	Admin
Customer			generate		pay	give	
Worker			need	Start/End		get	
Request	generated by	needs					
Work		Start/End			generate		
Bill	pay			work			
Reviews	give						check
Admin			view			check	

### 3.1.6.3 Fill in Cardinality:

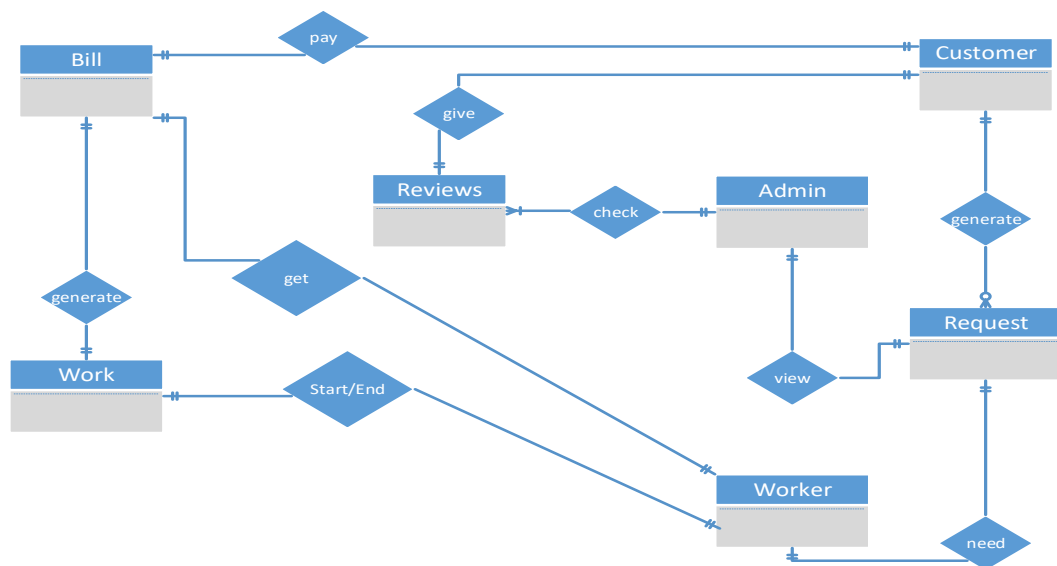


Figure 0-12 (Fill in Cardinality)

From the description of the problem we see that:

1. Each customer can pay bill generated against a single work.
2. Each customer can generate zero or many requests.
3. Each customer can give a review against a single Worker.
4. 1 Worker can work against a single request at a time.
5. 1 Admin can check multiple Reviews.
6. Admin can view request.

7. A worker can Start/End a single work at a time.
8. Each worker gets a single bill generated against a single task.

#### 3.1.6.4 Primary Keys

The primary keys are customerID, requestID, workerID, CNIC, billID, reviewID, workID.

#### 3.1.6.5 Identify Attributes:

The only attributes indicated are customerID, requestID, workerId, billId, workID, reviewID and adminID

#### 3.1.6.6 Map Attributes

Attribute	Entity	Attribute	Entity
customerID	Customer	requestID	Request
workerID	Worker	billID	Bill
workID	Work	reviewID	Review
adminID	Admin		



### 3.1.6.7 Draw Fully Attributed ERD

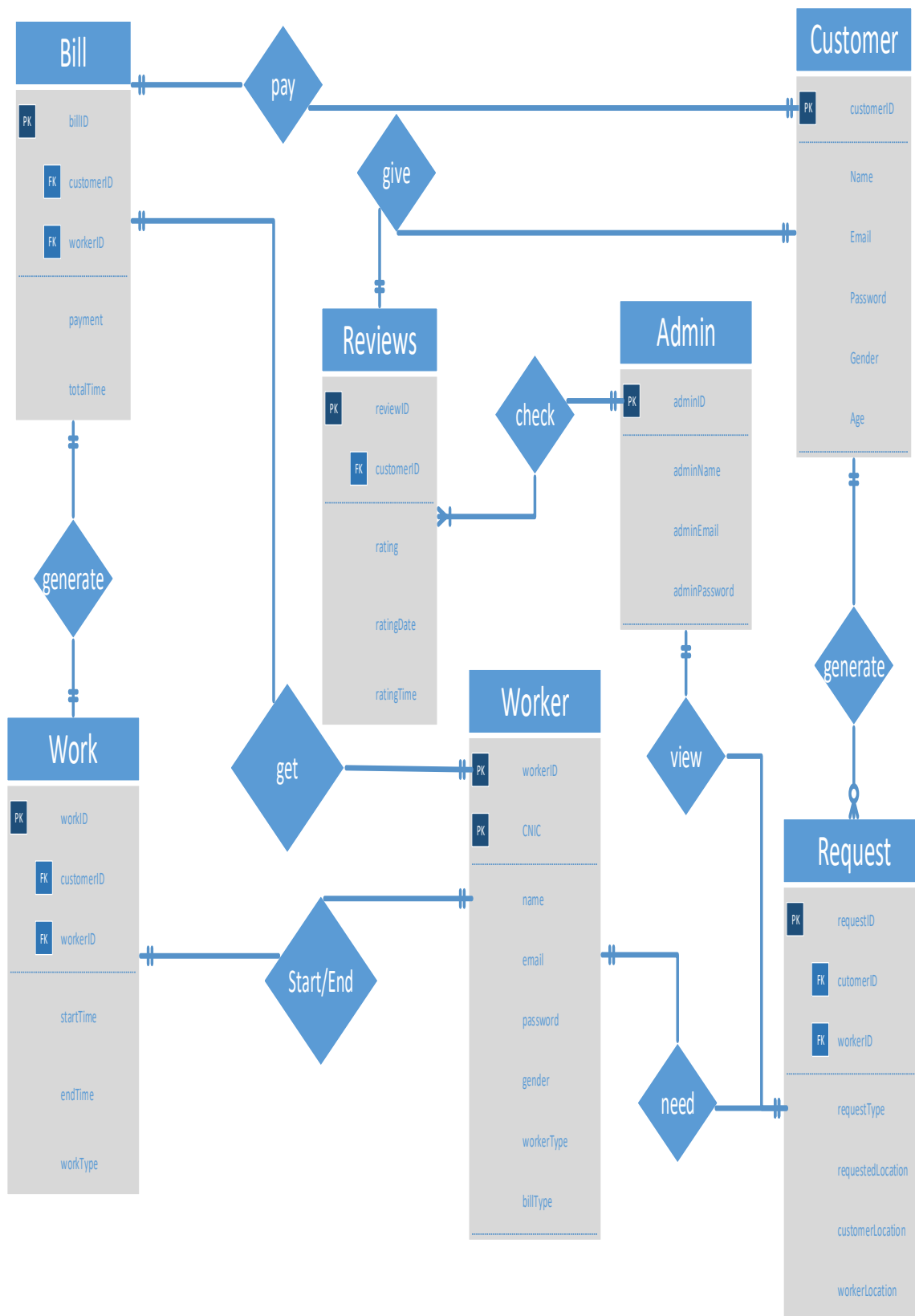


Figure 0-13 (Fully Attributed ERD)

### 3.1.7 METHODOLOGY

#### 3.1.7.1 RAD

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. In RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning involved in RAD development, it makes it easier to incorporate the changes within the development process [6].

Rapid Application Development (RAD) suits on our project because our requirements are completely defined, and project can be easily divided into modules which results the development of project in shorter time span. RAD also provide time boxed development cycles or multiple cycles can be developed at the same time.

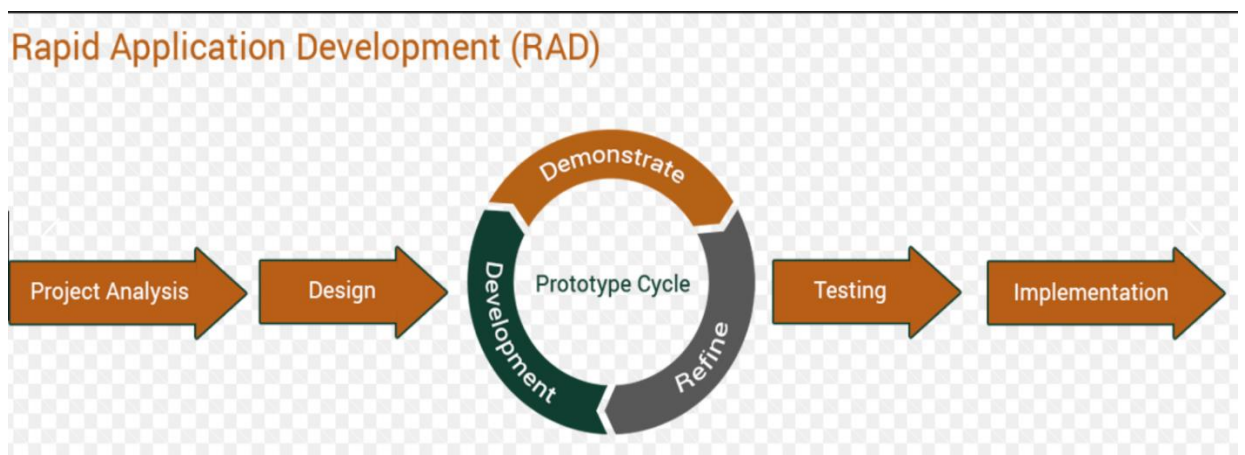


Figure 0-14(Rapid Application Development)

#### 3.1.7.2 ADVANTAGES:

- Development and delivery of software in shorter time.
- Changing requirements in our project can be accommodated easily
- Progress of project can be measured as project is divided in modules
- Productivity can be done with shorter number of people which is the case in our android application development.

### **3.1.7.3 CONS**

- Project should be broken down in modules in order to use the RAD model
- Management complexity of the project is more.
- Integration issues will occur as different modules are done on different machines.
- Requires skilled developers.

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 DATABASE

We have used firebase that provide us various features like Authentication, Messaging and Real-time Database [3]. Database in firebase is a cloud-based database and doesnot need SQL-based queries to store and retrieve data. Firebase database is highly reliable and superfast means that data is updated and synchronized in no time [4]. We also use the firebase authentication feature that provides us to authenticate user phone number, email, facebook and gmail validation [5].

There are six main roots

- Karigar
- User
- Karigar available
- Working Karigar
- Customer requests
- Work complete

##### 4.1.1 Karigar

In Karigar root we have multiple childs that represent different workers and each child contains data about single worker which contains the workers phone number used as a unique identifier. Unique identifiers contain further sub childs which contains the information of worker like its first name, last name, email address, phone no, password and its type

### **4.1.2 User**

In User root we have multiple childs that represent different users and each child contains data about single user which contains the user phone number used as a unique identifier. Unique identifiers further contain sub childs which has the information of user like its first name, last name, email address, phone no, password.

### **4.1.3 Karigar available**

Karigar root will be created on run time which contains the childs id of workers which are online at that time and that id contains a subchild which contains the location and the information of workers available.

### **4.1.4 Customer Requests**

When user requests for a worker the request child is generated on run time in customer request root which has its unique id. Child further contains user id and its location where the worker is requested.

### **4.1.5 Working Karigar**

When worker accepts request the worker is removed from the root "**Karigar available**" and then the code generates a child in working worker which shows that the worker is in working condition. This root contains the child of customer request id which further contains the id of user and worker which are connected and duration of continued work.

### **4.1.6 Work complete**

Work complete has childs which are unique identifiers of complete work done and identifier child further contains request id, user and worker id, work duration and work payment.

#### **4.1.7 Advantages for using Firebase.**

##### **4.1.7.1 Real-time:**

Instead of using the HTTP requests, Firebase real-time Database uses synchronization of data every time data changes, any device which are connected with the real-time database receives that update just within milliseconds. Firebase provides collaborative and immersive experiences without having to think about networking coding.

##### **4.1.7.2 Offline:**

A major benefit of using firebase database is that Firebase apps remain responsive even when it is used offline because the Firebase Real-time Database SDK persists your data to the disk. Once online connectivity with database is re-established, the client device receives changes it missed whether in any way and synchronizes it with current state of the server.

##### **4.1.7.3 Accessible from Client Devices:**

The Firebase Real-time Database can directly be accessed from a mobile device or web browser and there's no need to use an application server. Security of data and data validation are available through the Firebase real-time. Database Security Rules, expression-based rules that are executed when data is accessed for reading or writing data.

##### **4.1.7.4 Scale across multiple databases:**

With the use of Firebase Real-time Database, you can support your applications data needs by splitting your data across multiple database instances in the same Firebase project. Control access to entire data in each database with custom firebase real-time database rules for each database instance.

## **4.2 Software used for Android development:**

### **4.2.1 Android studio:**

Android studio is an integrated development environment used for the development of android applications.

#### **4.2.1.1 Advantages:**

- Android studio has a flexible Gradle-based build system
- A fast and feature-rich emulator with versatile functionalities
- A vast environment where you can develop for all the Android devices.
- Instant run your project so that you don't have to build APK every time.
- Importing sample codes from GitHub and other platforms becomes easy.
- Extensive testing tools and large variety frameworks.
- Built-in support for Google Cloud Platform, making it easy to integrate Google.
- Cloud Messaging and App Engine.

### **4.2.2 JSON**

JavaScript Object Notation is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. These properties make JSON an ideal data-interchange language.

## **4.3 Errors during Android Development:**

### **4.3.1 Firebase Connectivity Error:**

This error occurred because the old version of the firebase dependencies added were not being supported and synced by the latest version of the android studio that was android 3.0.1 at that moment.

#### **4.3.2 Firebase retrieval error:**

Inability to read the content of our Firebase real time data base. This error occurred because we did not set the rules of data base read to true in the Firebase Real time database rules.

#### **4.3.3 Module Integration Error:**

Performing module based coding on different machines and integrating them afterwards on a single machine caused errors because the project names were different as well as the project had different android versions whereas the coding layouts were different in certain conditions.

#### **4.3.4 Linking a java file with multiple Xml files:**

During the creation of a map activity the java file was not being linked with other xml files which resulted in the crash of the application.

#### **4.3.5 Auto Acceptation of Customers Request:**

During the coding of map Activity whenever we generated a request from the customers side the customer's request automatically got accepted by the nearby available Karigar whereas the objective was that the worker should accept the request based on his own personal choice



### **4.3.6 Extending an Activity:**

When applying multiple extends to the same java file in android studio the java file displayed errors which resulted in the inability to run the application.

## **4.4 Tools used for web development**

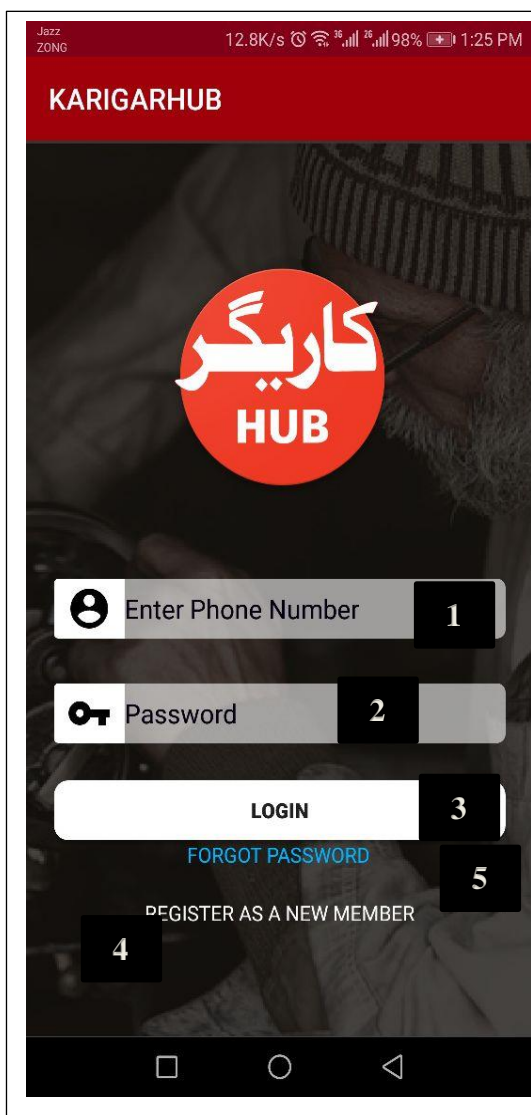
### **4.4.1 PHPStorm:**

PHPStorm is an editor that edits code and deeply understands the code structure, supporting all **PHP** language features for modern and old projects. It provides the best code completion, refactoring's, on-the-fly preventing errors, and more.

## CHAPTER 5

## USER MANUAL

## STEP 1



1: Use your verified phone number to login enter in the field area to proceed.

2: Enter that password which was provided by you at the time of registration.

3: Press login to use Karigar-Hub services.

4: If you are not register TAP on the “REGISTER AS A NEW MEMBER” to register in the application.

5: If you are registered and forgot your password then TAP on “FORGOT PASSWORD” it will take you to another page where you can recover your password using your mobile number.

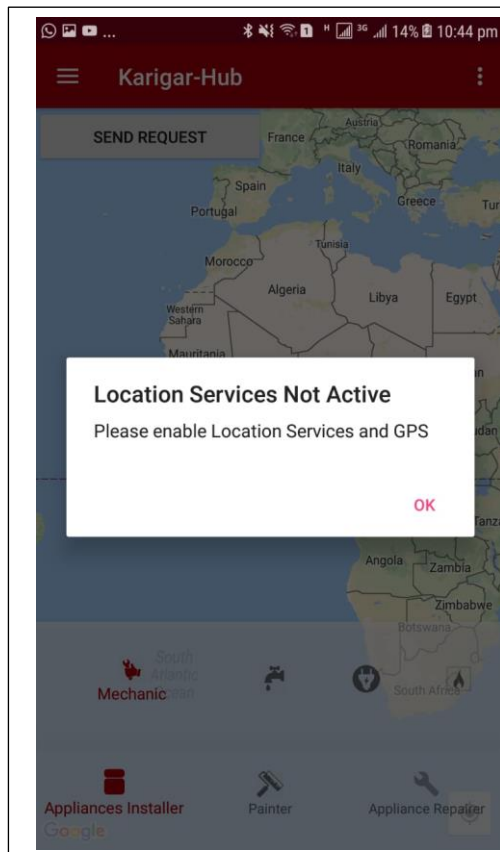
## STEP 2

- 1: Enter your first name properly.
- 2: Enter your last name properly.
- 3: Enter your valid email address
- 4: Type your password with minimum 6 letters
- 5: Enter your valid phone number.
- 6: Press Register

## STEP 3

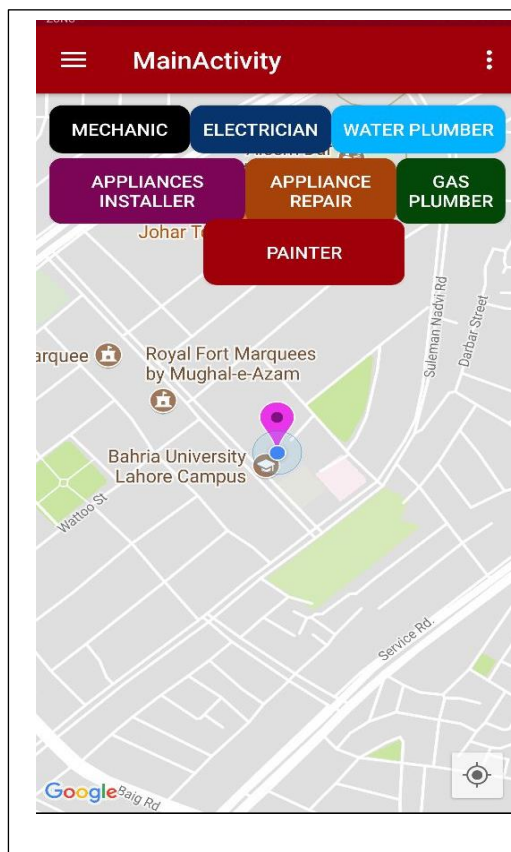
- 1: Enter the code you will receive on your mobile number
- 2: After entering the valid code that was received on your mobile number click verify.
- 3: If you don't receive the code click "Resend Code" button to get code and repeat the steps 1 and 2

## STEP 5



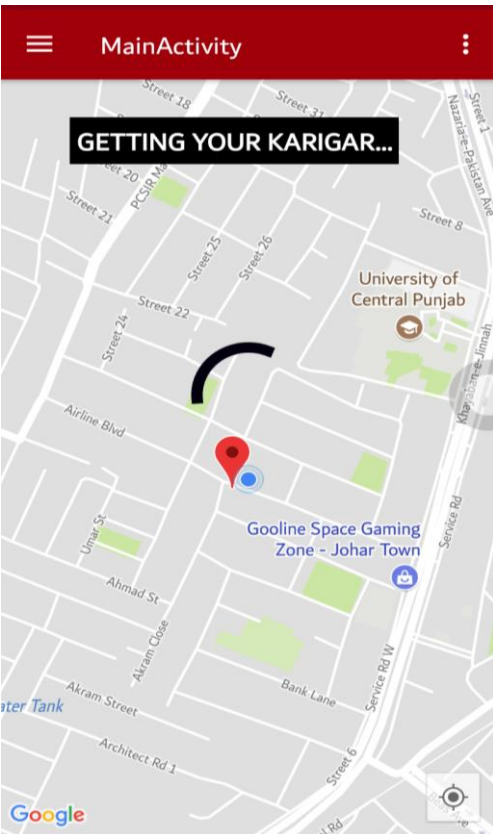
1: Click “OK” and turn on your location services from your phone settings.

## STEP 6



Now you can select type of worker you want and click then the request will be sends to all nearby workers.

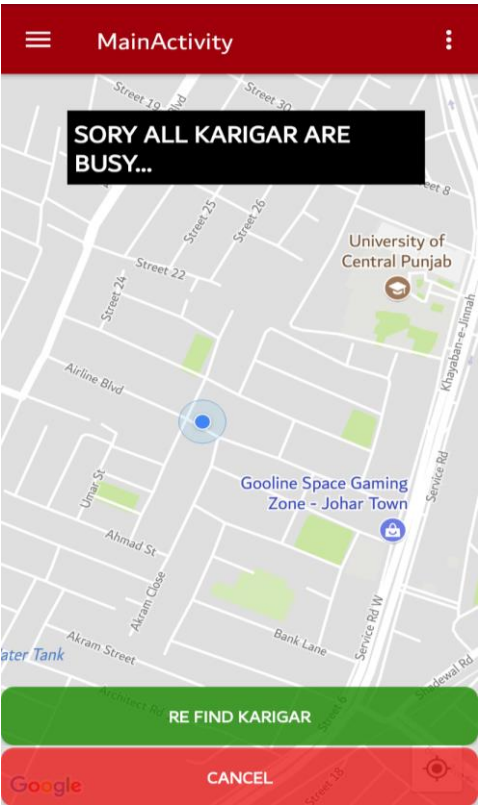
## STEP 7



The screenshot shows a mobile application interface with a red header bar containing a hamburger menu icon, the text "MainActivity", and a vertical ellipsis icon. The main content is a map from Google Maps showing a street grid in Johar Town. A red location pin is placed on the map, with a blue circle around it. A black dialog box with white text "GETTING YOUR KARIGAR..." is overlaid on the map. A black curved arrow points from the dialog box towards the location pin. Other labels on the map include "University of Central Punjab", "Gooline Space Gaming Zone - Johar Town", "Airline Blvd", "Ahmad St", "Akram Close", "Bank Lane", "Service Rd W", "Service Rd", "Khagabane-Jimah", "Street 18", "Street 20", "Street 21", "Street 22", "Street 24", "Street 25", "Street 26", "Street 27", "Street 28", "Street 29", "Street 30", "Street 31", "Street 32", "Street 33", "Street 34", "Street 35", "Street 36", "Street 37", "Street 38", "Street 39", "Street 40", "Street 41", "Street 42", "Street 43", "Street 44", "Street 45", "Street 46", "Street 47", "Street 48", "Street 49", "Street 50", "Street 51", "Street 52", "Street 53", "Street 54", "Street 55", "Street 56", "Street 57", "Street 58", "Street 59", "Street 60", "Street 61", "Street 62", "Street 63", "Street 64", "Street 65", "Street 66", "Street 67", "Street 68", "Street 69", "Street 70", "Street 71", "Street 72", "Street 73", "Street 74", "Street 75", "Street 76", "Street 77", "Street 78", "Street 79", "Street 80", "Street 81", "Street 82", "Street 83", "Street 84", "Street 85", "Street 86", "Street 87", "Street 88", "Street 89", "Street 90", "Street 91", "Street 92", "Street 93", "Street 94", "Street 95", "Street 96", "Street 97", "Street 98", "Street 99", "Street 100".

It will show this waiting dialog until worker receive customer request.

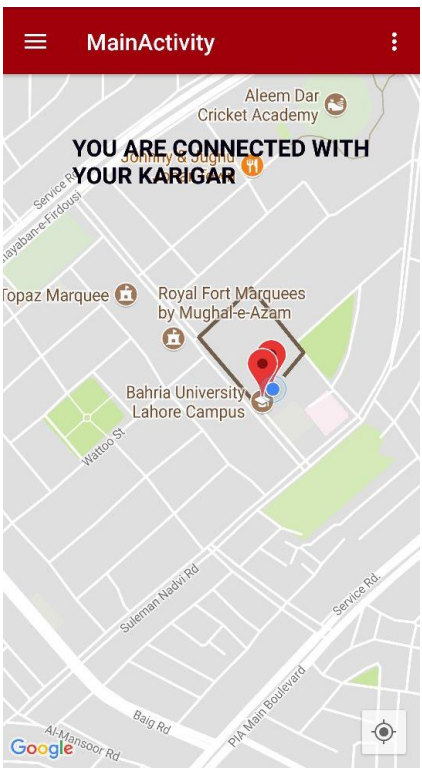
## STEP 8



The screenshot shows the same mobile application interface as in Step 7. The red header bar is still present. The map shows the same street grid. A blue location pin is now visible, indicating that no worker was found. A black dialog box with white text "SORRY ALL KARIGAR ARE BUSY..." is overlaid on the map. At the bottom of the screen, there are two buttons: a green button labeled "RE FIND KARIGAR" and a red button labeled "CANCEL".

If no any worker is found then this window will appear where customer can retry to find the worker or cancel to find the customer.

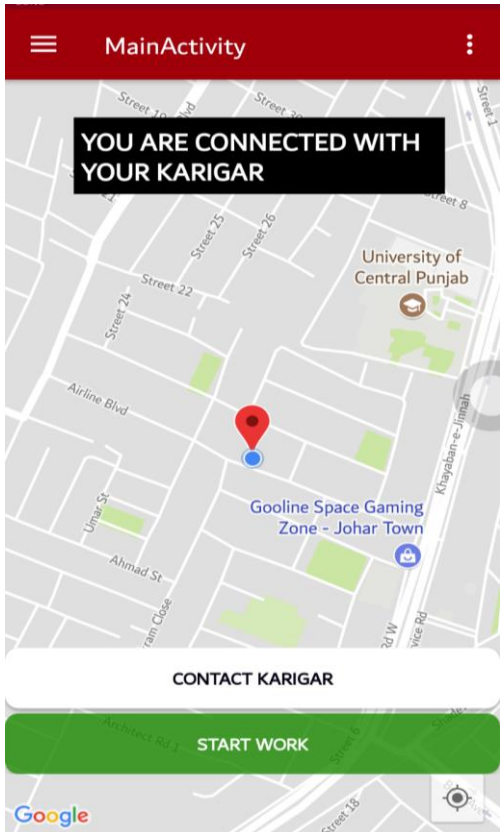
## STEP 9



The screenshot shows the 'MainActivity' interface of an application. At the top, there is a red header with a hamburger menu icon on the left and three vertical dots on the right. Below the header is a map view. A red location pin is placed on the map, with a blue dot below it. A black text box with white text is overlaid on the map, reading 'YOU ARE CONNECTED WITH YOUR KARIGAR'. The map shows various landmarks including 'Aleem Dar Cricket Academy', 'Royal Fort Marquees by Mughal-e-Azam', and 'Bahria University Lahore Campus'. Street names like 'Watoos St', 'Suleman Nazki Rd', and 'PK Main Boulevard' are visible. The Google logo is in the bottom left corner.

When the worker will accept the request live tracking will be started.

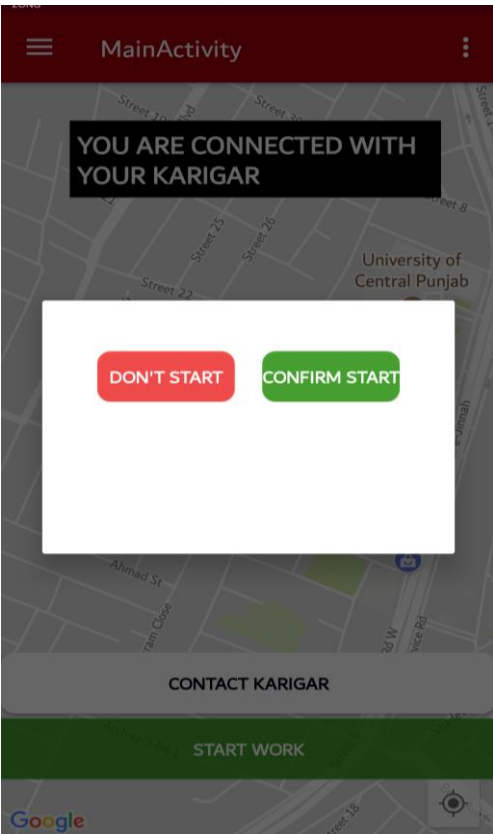
## STEP 10



The screenshot shows the 'MainActivity' interface of an application. At the top, there is a red header with a hamburger menu icon on the left and three vertical dots on the right. Below the header is a map view. A red location pin is placed on the map, with a blue dot below it. A black text box with white text is overlaid on the map, reading 'YOU ARE CONNECTED WITH YOUR KARIGAR'. Below the map, there are two buttons: a white button with black text that says 'CONTACT KARIGAR' and a green button with white text that says 'START WORK'. The map shows various landmarks including 'University of Central Punjab' and 'Gooline Space Gaming Zone - Johar Town'. Street names like 'Street 22', 'Street 24', 'Street 25', 'Street 26', 'Airline Blvd', 'Umar St', 'Ahmad St', and 'Khasrbane-Jinnah' are visible. The Google logo is in the bottom left corner.

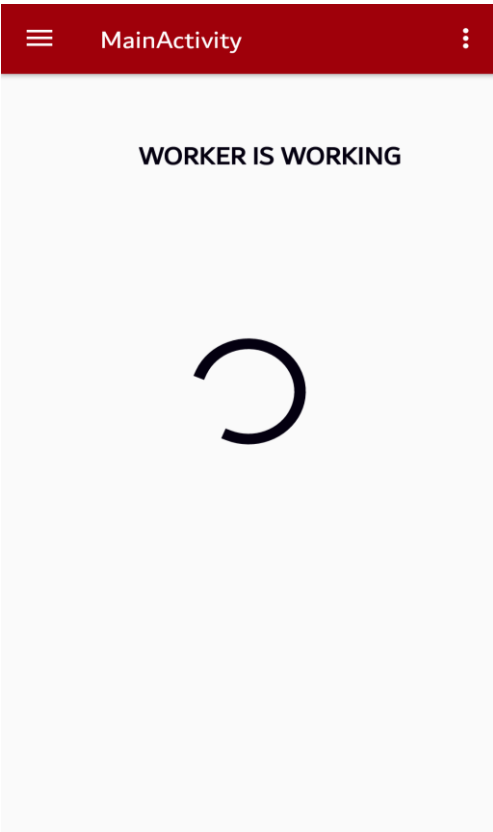
If the worker is found and arrived at customer location then this window will appear.

## STEP 11



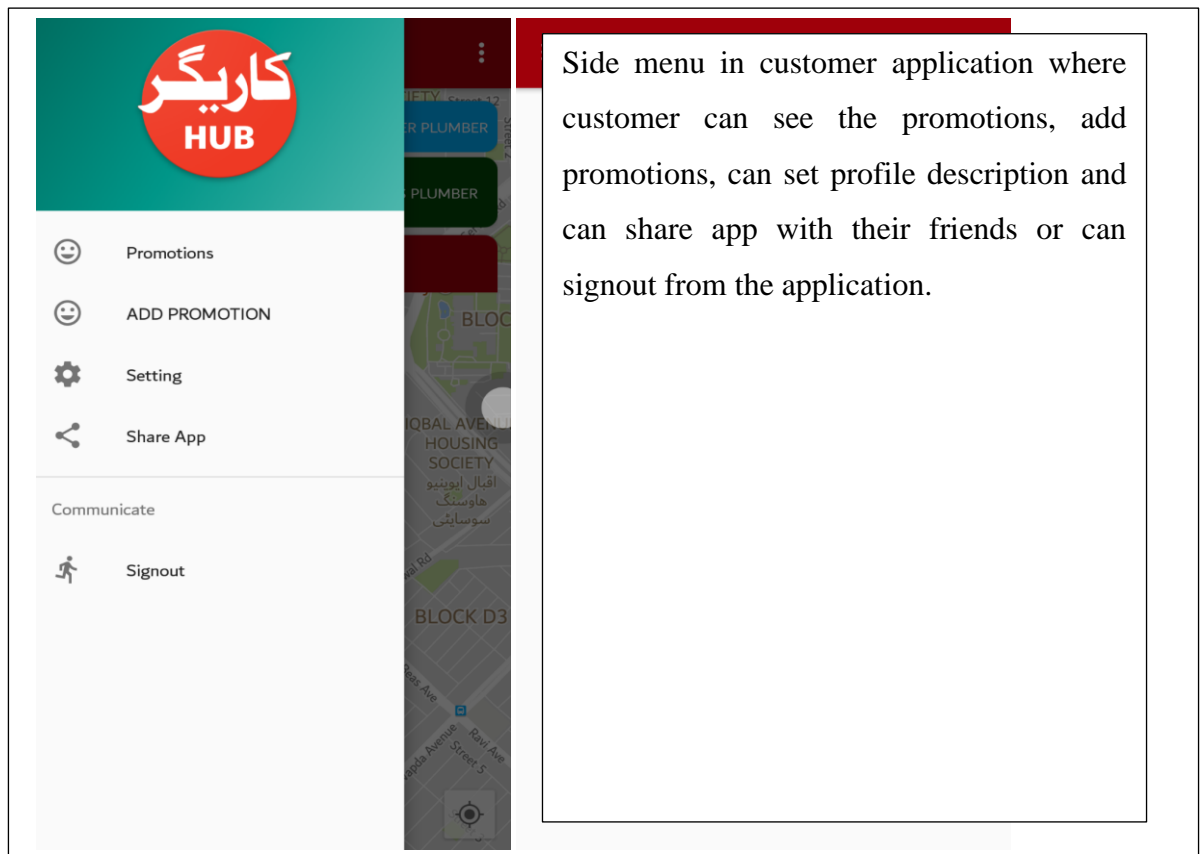
It is on the customer side start or don't start the work.

## STEP 12

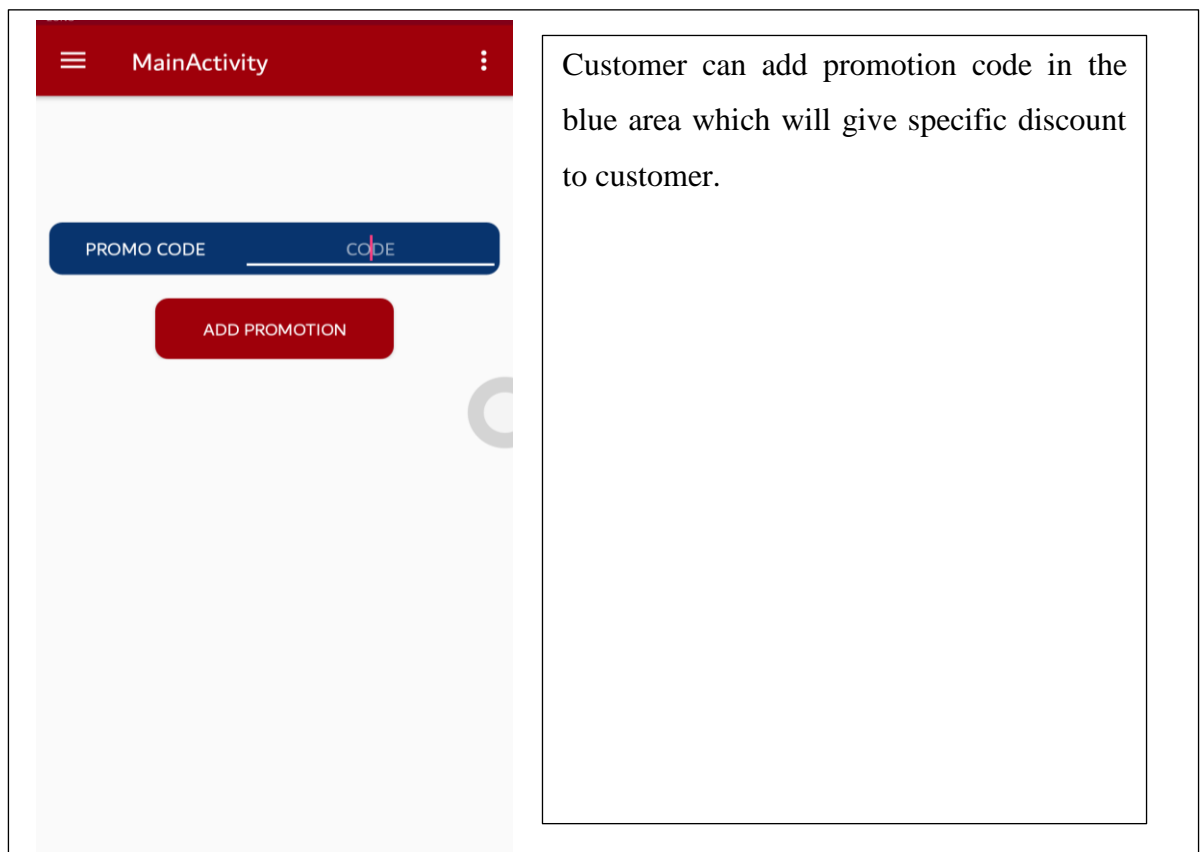


After the confirm start this window will appear where customer can only see the worker is working.

## STEP 12

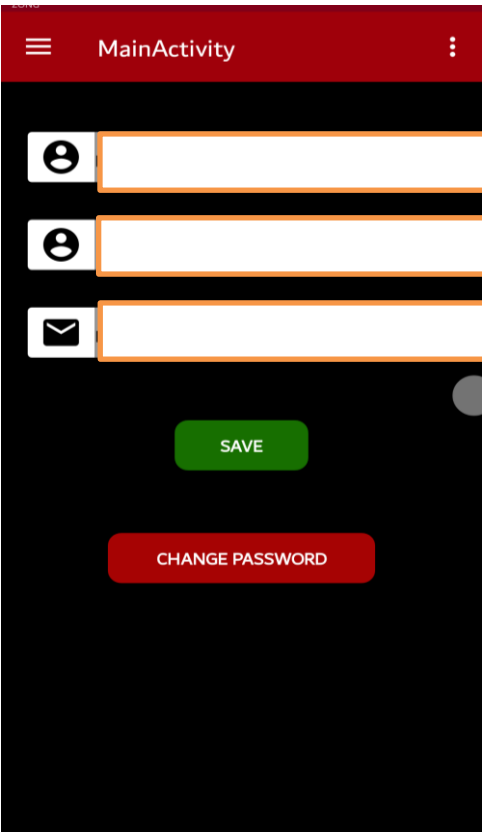


## STEP 13





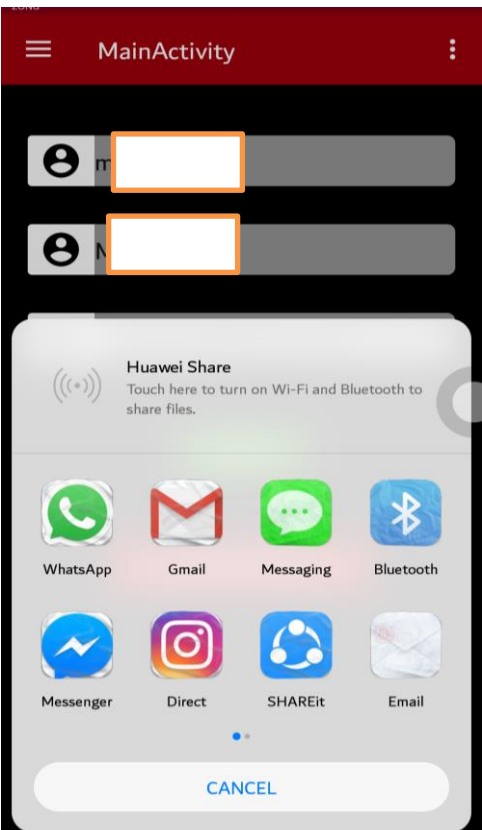
## STEP 14



The screenshot shows the MainActivity interface with a red header bar containing a hamburger menu icon, the text "MainActivity", and a vertical ellipsis icon. Below the header, there are three input fields, each preceded by an icon: a person icon for the first field, another person icon for the second, and an envelope icon for the third. Each input field is highlighted with an orange border. Below the input fields, there is a green "SAVE" button and a red "CHANGE PASSWORD" button.

After click on setting customer can change their first name, last name email and change their password.

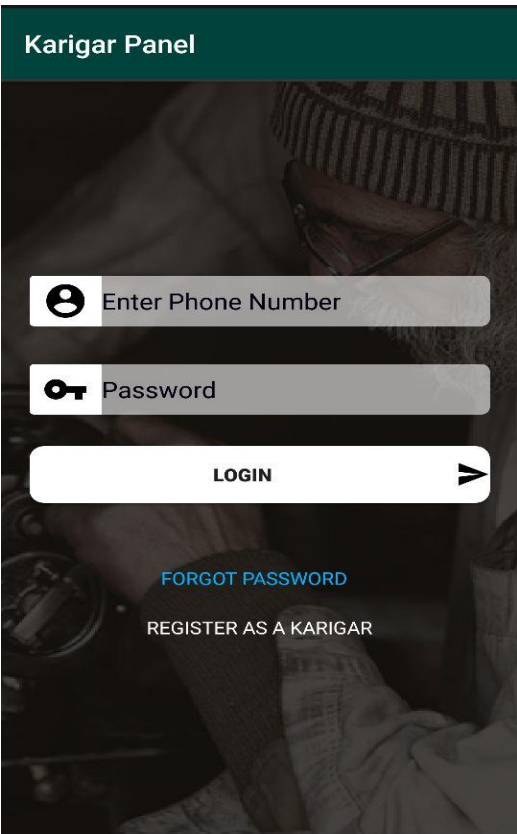
## STEP 15



The screenshot shows the MainActivity interface with a red header bar containing a hamburger menu icon, the text "MainActivity", and a vertical ellipsis icon. Below the header, there are two input fields, each preceded by a person icon. Each input field is highlighted with an orange border. A sharing sheet is displayed over the bottom half of the screen, titled "Huawei Share" with a Wi-Fi and Bluetooth icon. Below the title, it says "Touch here to turn on Wi-Fi and Bluetooth to share files." The sharing sheet contains eight sharing options: WhatsApp, Gmail, Messaging, Bluetooth, Messenger, Direct, SHAREit, and Email. A "CANCEL" button is at the bottom of the sharing sheet.

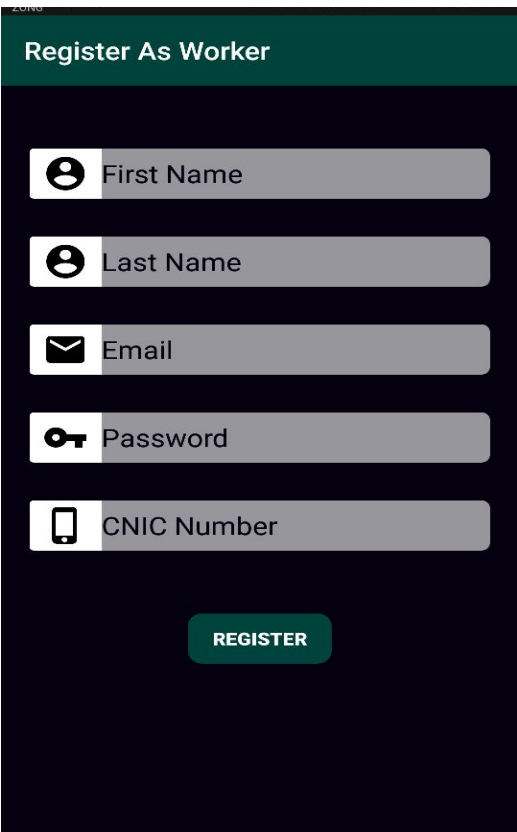
After clicking the share button customer can choose between the various options to share the application.

## STEP 16



Worker Login screen where customer can login with phone number and password can log in. Worker can click the forgot password to change their password by phone number verification. New Worker can also register by using application.

## STEP 17



Register Form Number 1 for Worker  
Worker should fill this form properly to continue.

## STEP 18

### Register As Worker

Phone Number

**VERIFY PHONE NUMBER**

1 2 3 ( ) ,  
4 5 6 + - ;  
7 8 9 / N ✕  
\* 0 # . ↩

Register Form Number 2 for where the worker can register their mobile number by verifying.

## STEP 19

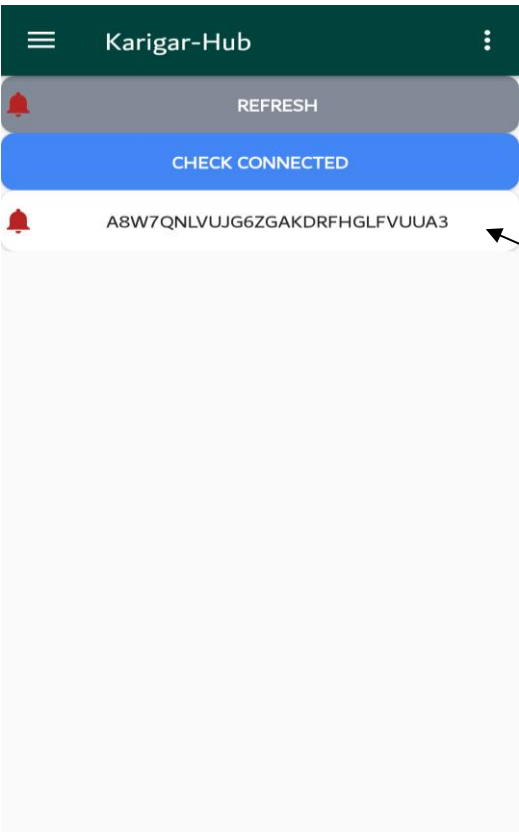
### Register As Worker

Mechanic       Electrician  
 GasPlumber       WaterPlumber  
 Appliance Installer       Appliance Repair  
 Painter

**SAVE**

Register Form Number 3 for where the worker can select type of work in which they can work.

## STEP 20

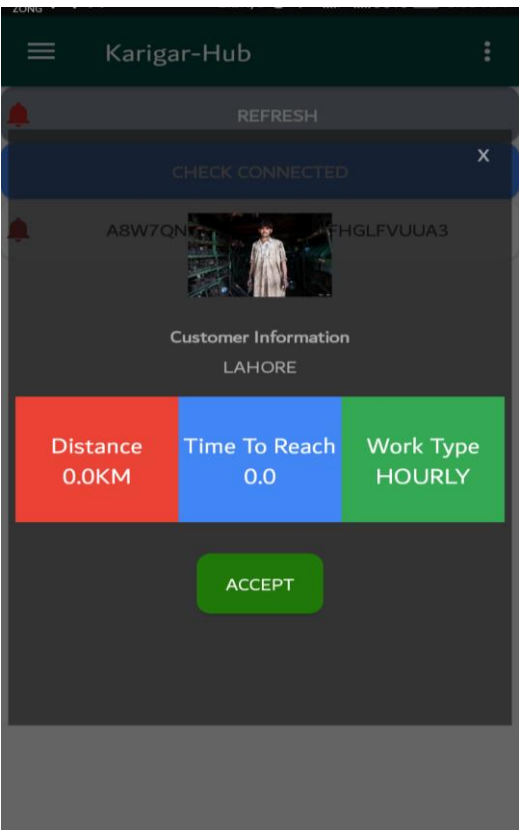


The screenshot shows the main interface of the 'Karigar-Hub' app. At the top, there is a dark green header with a hamburger menu icon on the left, the text 'Karigar-Hub' in the center, and a vertical ellipsis menu icon on the right. Below the header is a grey bar with a red notification bell icon and the text 'REFRESH'. Underneath is a blue bar with the text 'CHECK CONNECTED'. A white notification card is visible, featuring a red bell icon on the left and the alphanumeric string 'A8W7QNLVUJG6ZGAKDRFHGLFVUUA3' in the center. An arrow points from the text on the right to this notification card.

Main screen will arrive when the worker successfully logged.

Request will arrive there customer can click to accept or decline.

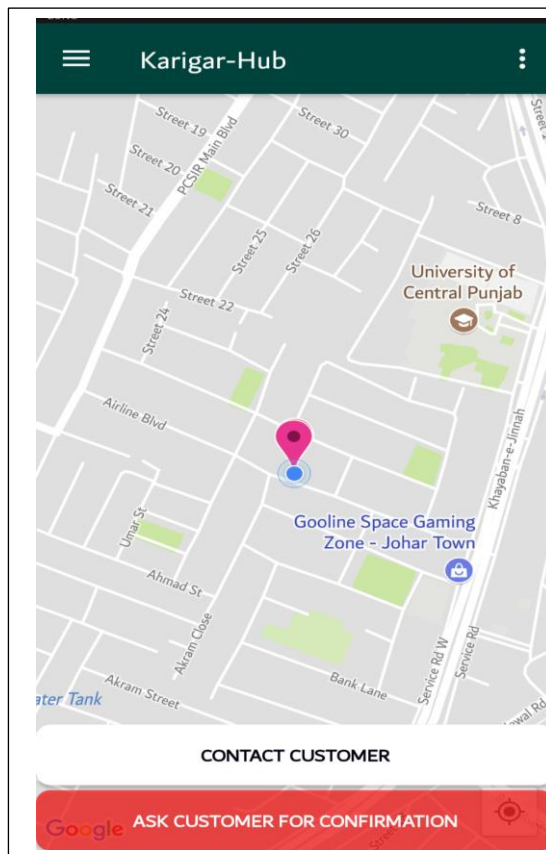
## STEP 21



The screenshot displays a detailed job request window within the 'Karigar-Hub' app. The window has a dark grey background and contains the following elements: a red notification bell icon on the left, the alphanumeric string 'A8W7QNLVUJG6ZGAKDRFHGLFVUUA3' in the center, and a small image of a person in a white uniform. Below the image, the text 'Customer Information' and 'LAHORE' is displayed. At the bottom, there are three colored boxes: a red box with 'Distance 0.0KM', a blue box with 'Time To Reach 0.0', and a green box with 'Work Type HOURLY'. A green 'ACCEPT' button is positioned at the bottom center of the window.

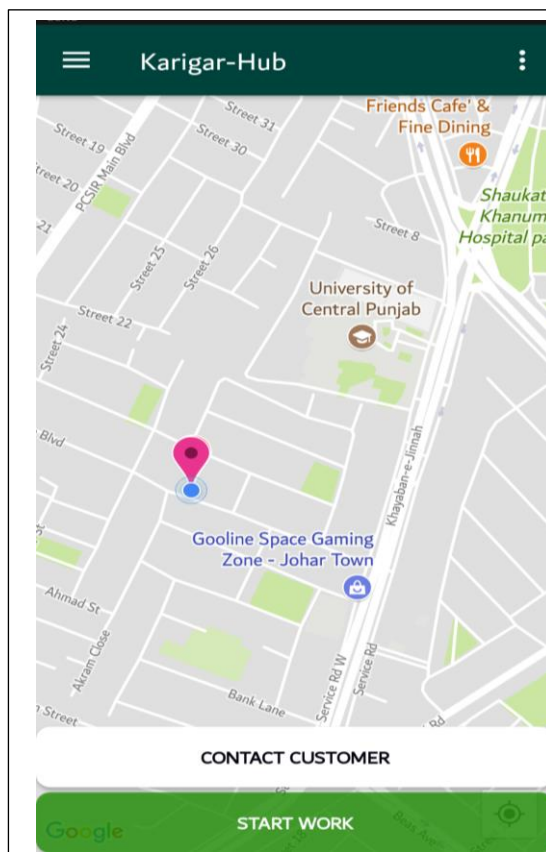
This window will appear where the complete detail of work and can accept the work.

## STEP 22



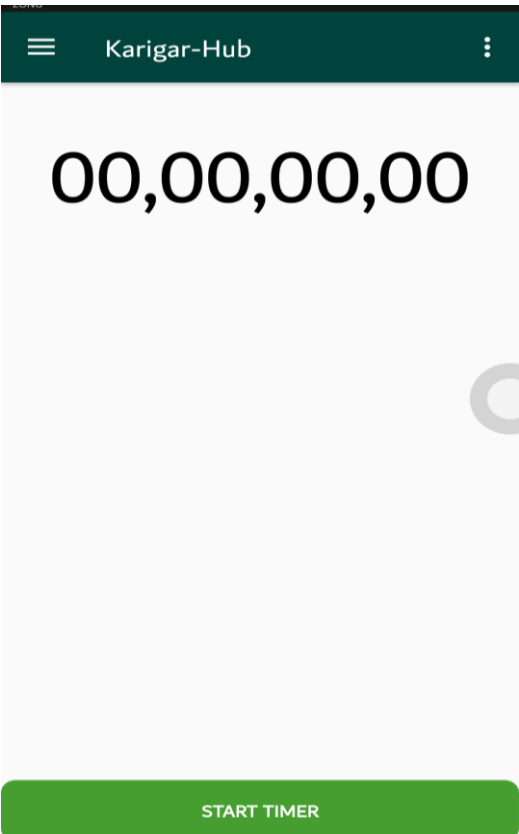
After accepting the request live tracking will be started, worker can contact to the customer and can ask customer for confirmation.

## STEP 23



When customer confirm start, Start work button will appear.


## STEP 24



The screenshot shows the Karigar-Hub app interface. At the top, there is a dark green header with a hamburger menu icon on the left, the text "Karigar-Hub" in the center, and a vertical ellipsis icon on the right. Below the header, the main display area shows a large digital timer with the value "00,00,00,00". At the bottom of the screen, there is a green button with the text "START TIMER".

After clicking the start work, start timer will appear, where worker can start the timer.


## STEP 25



The screenshot shows the Karigar-Hub app interface. At the top, there is a dark green header with a hamburger menu icon on the left, the text "Karigar-Hub" in the center, and a vertical ellipsis icon on the right. Below the header, the main display area shows a large digital timer with the value "00,01,22,37". At the bottom of the screen, there is a red button with the text "END WORK".

Worker can end the work which will generate the bill.

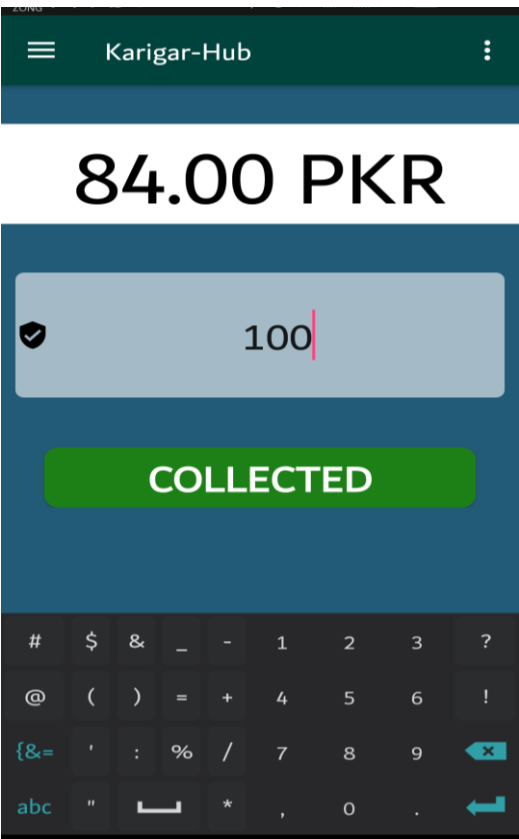
## STEP 26



The screenshot shows the Karigar-Hub app interface. At the top, there is a dark green header with a hamburger menu icon on the left, the text "Karigar-Hub" in the center, and a vertical ellipsis icon on the right. Below the header, the amount "84.00 PKR" is displayed in large white text. A large dark blue rectangular area occupies the middle of the screen. At the bottom of this area, there is a grey button with the text "COLLECT CASH" in white capital letters.

Worker can collect cash bill.

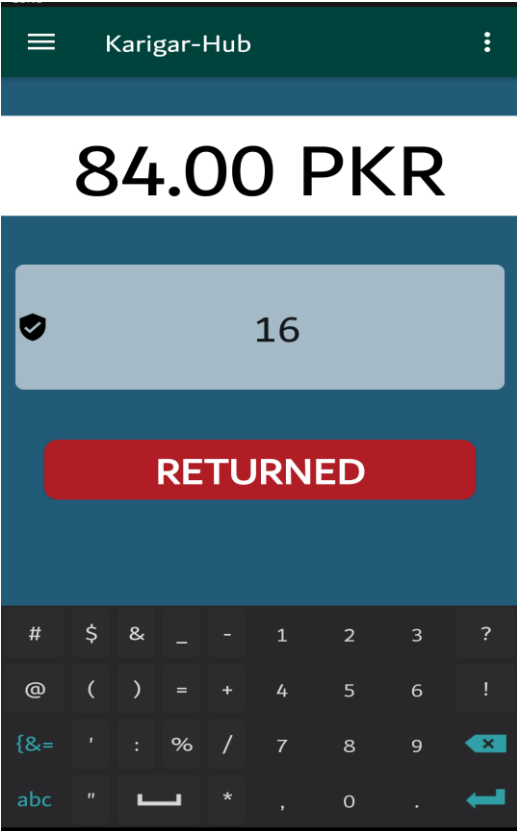
## STEP 27



The screenshot shows the Karigar-Hub app interface. At the top, there is a dark green header with a hamburger menu icon on the left, the text "Karigar-Hub" in the center, and a vertical ellipsis icon on the right. Below the header, the amount "84.00 PKR" is displayed in large white text. Below this, there is a light blue input field containing the number "100" with a red cursor. A green button with the text "COLLECTED" in white capital letters is positioned below the input field. At the bottom of the screen, a numeric keypad is visible, featuring symbols like #, \$, &, \_, -, 1, 2, 3, ?, @, (, ), =, +, 4, 5, 6, !, {&=, ', :, %, /, 7, 8, 9, a backspace arrow, abc, ", \_ (underscore), \*, , 0, ., and a return arrow.

Worker can add the amount of collected cash.

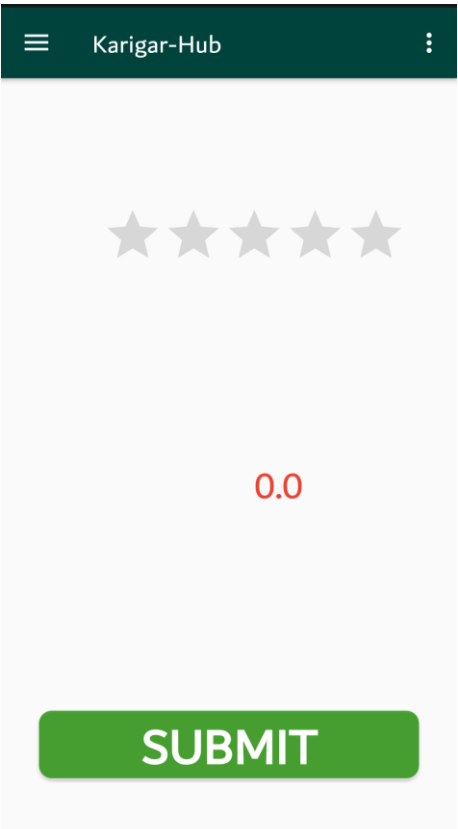
## STEP 28



The screenshot shows the Karigar-Hub app interface. At the top, there is a dark green header with a menu icon, the text "Karigar-Hub", and a vertical ellipsis icon. Below the header, the amount "84.00 PKR" is displayed in large black text. Underneath, there is a light blue input field containing the number "16" with a checkmark icon on the left. A red button with the text "RETURNED" is positioned below the input field. At the bottom, a dark grey calculator keypad is visible, featuring various symbols and numbers.

Worker should enter the returned amount to continue.

## STEP 29

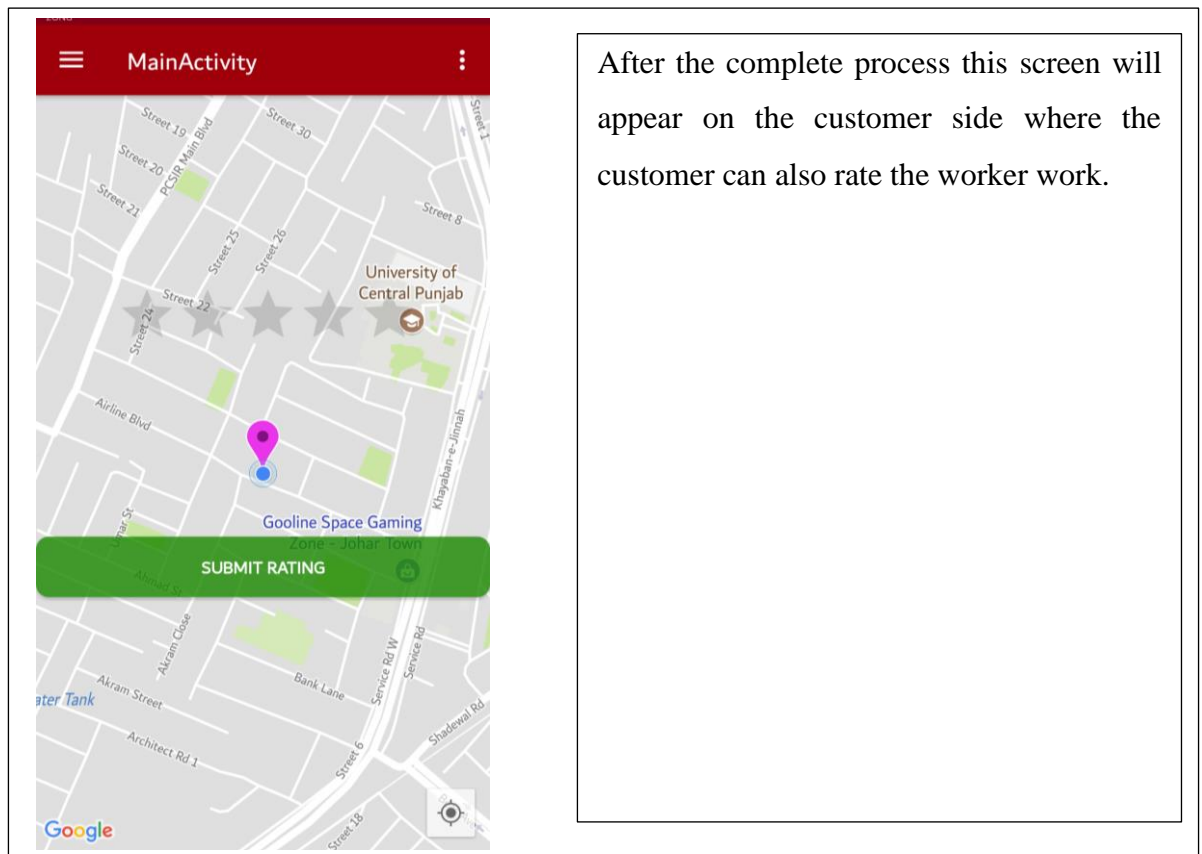


The screenshot shows the Karigar-Hub app interface for customer rating. At the top, there is a dark green header with a menu icon, the text "Karigar-Hub", and a vertical ellipsis icon. Below the header, there are five grey stars arranged horizontally. Below the stars, the score "0.0" is displayed in red text. At the bottom, there is a green button with the text "SUBMIT" in white.

Worker can rate the customer.



## STEP 30



## ADMIN PANEL



localhost/KarigarWeb/index.php

## KARGIAR HUB

ADMIN Online

Search...

MAIN NAVIGATION

- Dashboard
- CONTROLS
- Mailbox 5 16 12

Dashboard REQUESTS

Home Dashboard

**3**

New Request

[More info](#)

**33.33%**

Bounce Rate

[More info](#)

**9**

User Registrations

[More info](#)

**2**

Karigars

[More info](#)

**Click on more info to view each of them in detail**

**REFRESH**

Requests

Show 10 entries Search:

Worker ID	User ID	Worker Location	User Location	PRICE
Trident	Internet Explorer 4.0	Win 95+	4	X

Showing 1 to 1 of 1 entries Previous 1 Next

localhost/KarigarWeb/BounceRate.php

## KARGIAR HUB

ADMIN Online

Search...

MAIN NAVIGATION

- Dashboard
- CONTROLS
- Mailbox 5 16 12

REQUEST BOUNCES

Home Dashboard

**HOME**

**33.33%**

Bounce Rate

**REFRESH**

Bounce Request Information

Show 10 entries Search:

Worker ID	User ID	Worker Location	User Location	PRICE
UMAR	Alli Moeid	Baldia Chowk	Noor Masjid	500
03006595899	03120470244	WorkerLocation	UserLocation	Price
03214888900	03064333758	WorkerLocation	UserLocation	PRICE

Showing 1 to 1 of 1 entries Previous 1 Next

localhost/KarigarWeb/register.php

## ADMINREGISTER

ADD NEW ADMIN FOR ADMIN PANEL

Full name

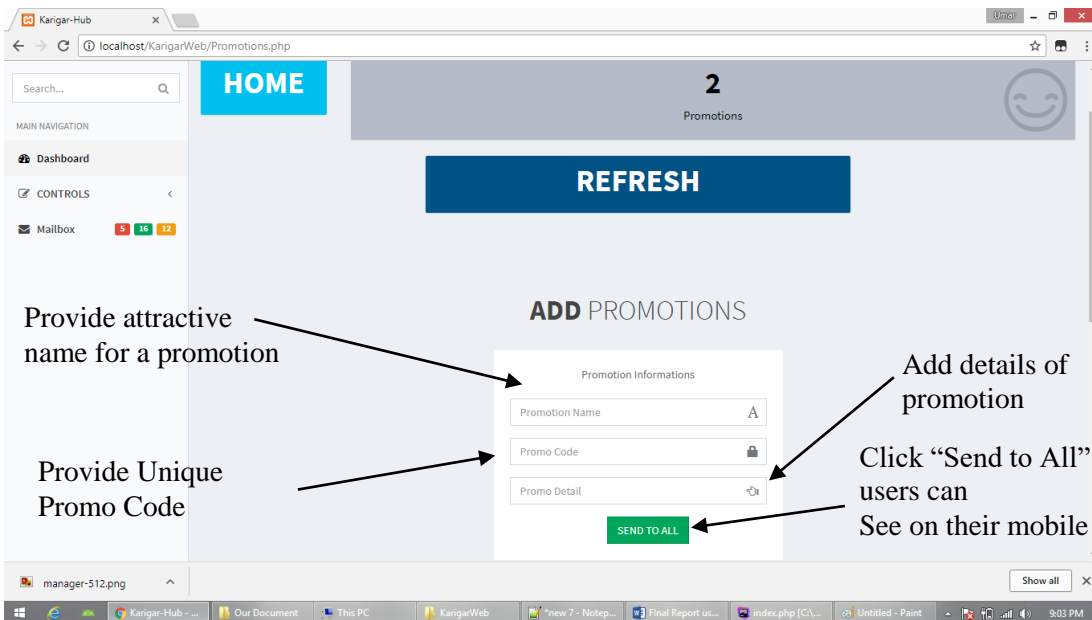
Email

Password

Retype password

I agree to the terms

**Register**



Provide attractive name for a promotion

Provide Unique Promo Code

### ADD PROMOTIONS

Promotion Informations

Promotion Name

Promo Code

Promo Detail

SEND TO ALL

Add details of promotion

Click "Send to All" so users can See on their mobile apps

## CHAPTER 6

### CONCLUSION AND RECOMMENDATIONS

#### **6.1 Conclusion:**

Karigar-Hub is committed to provide excellence in its customer service. Customer service is an important aspect in the provision of our services using our application. Perfection in customer benefit can only be fully accomplish by having a value system known as feedback which places the customer at the heart of all in all. Karigar-Hub has fixed this fashion of customer benefit by giving high arrangement to a customer.

#### **6.2 Recommendations:**

##### **6.2.1 Strong internet connection:**

The workers should use a 3G or 4G facility on their mobiles otherwise the application might lag while getting the location of the customer.

##### **6.2.2 Free mobile storage:**

The worker and customer should at least have 1GB of free space on their phone so that the application can run on the mobile phone without affecting the phones speed

**6.2.3 Running out of battery:**

The worker should have a power bank along with him if in case the battery of mobile phone runs out.

**6.2.4 Work satisfaction level:**

The customer should ensure that the work satisfaction level has been achieved to and that the work has been completed before giving worker his payment

## REFERENCES

### Journal Papers:

- [1]. “A New Approach for Location based Tracking” Shaveta Bhatia<sup>1</sup>, Saba Hilal,  
,Manav Rachna International University 2013.
- [2] “GOOGLE MAPS AND RSS INTEGRATION IN ANDROID”, Sabyasachi  
Patra<sup>1</sup>, Karishma Velisetty<sup>2</sup> and Prathamesh Patel<sup>3</sup> 2013.
- [3] Study of Google Firebase API for Android “Navdeep Singh” Bachelor of  
Technology Student, Department of Computer Science Engineering, Seth Jai  
Parkash Mukand Lal Institute of Engineering and Technology, Radaur,  
Yamunanagar, India 2016.
- [4] Real Time Databases for Applications “Sonam Khedkar<sup>1</sup>, Swapnil Thube<sup>2</sup>, MCA  
(IMCOST), Mumbai University C-4, Wagle Industrial Estate, Near Mulund (W)”  
2017.
- [5] Go Mumbai “NehaSaxena , Suhasini Maurya, Ankit Rai , Santosh Yadav, Jinesh  
Malavia” 2017.
- [6] Rapid Applications Development Techniques: A Critical Review “Riffat Naz and  
M. N. A. Khan, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology  
(SZABIST) Islamabad, Pakistan” 2015.