

"HIRE ME"



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER, 2019

**DEVELOPMENT AN ANDROID APPLICATION NAMED
"HIRE ME"**

BY

Sajeeb Ahamed

ID: 161-15-667

Samiul Alam Shourov

ID: 161-15-1010

and

Shamim Ahammed Nayam

ID: 161-15-653

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Taslima Ferdaus Shuva

Lecturer (Senior Scale)

Department of Computer Science & Engineering

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

APPROVAL

This Project titled “**HIRE ME**” submitted by **Sajeeb Ahamed ID: 161-15-667, Samiul Alam ID: 161-15-1010, And Shamim Ahammed ID: 161-15-653** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 10 December, 2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

Dr. Sheak Rashed Haider Noori
Associate Professor & Associate Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Zahid Hasan
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Mohammad Shorif Uddin
Professor

Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

ACKNOWLEDGEMENT

First of all, the author expresses heartiest thanks and gratefulness to Almighty Allah to give them the capability to complete the final work successfully.

We are grateful and wish our profound indebtedness to our Supervisor Ms. **Taslima Ferdous Shuva**, Lecturer (Senior Scale), Department of Computer Science & Engineering, Daffodil International University, Dhaka. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project. The author would like to express the heartiest gratitude to all faculty member of Uttara campus and Dr. Syed Akhter Hossain, Professor, and Head, Department of Computer Science and Engineering, for their kind help to finish the project and also to other faculty member and the staff of CSE department of Daffodil International University. Finally, we would like to express our profound gratitude to our parents and our team for their inspiration, sacrifice, and extension of their hands during the progress of work.

DECLARATION

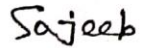
We hereby declare that, this project has been done by us under the supervision of **Ms. Taslima Ferdous Shuva, Lecturer (Senior Scale), Department of Computer Science & Engineering** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

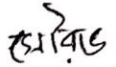


Ms. Taslima Ferdous Shuva
Lecturer (Senior Scale)
Department of Computer Science & Engineering
Daffodil International University


Submitted by:



Sajeep Ahamed
ID: 161-15-667
Department of Computer Science & Engineering
Daffodil International University



Md Samiul Alam Shourov
ID: 161-15-1010
Department of Computer Science & Engineering
Daffodil International University



Shamim Ahammed Nayam
ID: 161-15-653
Department of Computer Science & Engineering
Daffodil International University

ABSTRACT

Today we are utilizing UBER, Pathao and numerous ride-sharing applications for transportation. We are utilizing numerous sorts of shopping applications for shopping items from on the web. All that we are utilizing a direct result of our life's effortlessness and more agreeableness. From a similar idea, Hire Me has been created. HIRE ME is the mobile application which will help any user to find or get his/her desired service. This project is to some extent similar to an existing app which is "Sheba.xyz - Online Service Market". But this project has more feature than sheba.xyz and it is more significant as we've included real time location tracker. In HIRE ME the new feature is Online hiring system. Experts can be found for any problem. For example if an employer posts some job in the app, that post will be visible to all the similar job seekers who are expert in it. And one or more experts may handle the project. Job seeker and employer can also negotiate via email or verbal conversation via call. This is the key component that makes my application unique. Here I've built up the undertaking with online connection facilities, login method, posting method, call and mailing method.

TABLE OF CONTENTS

CONTENTS	PAGE
Title Page	i
Approval	ii
Acknowledgements	iii
Declaration	iv
Abstract	v
Table of Contents	vi-viii
List of Figures	ix - x

CHAPTER

CHAPTER 1: INTRODUCTION 1-2

1.1 Introduction	1
1.2 Project Objective	1
1.3 Methodology	1
1.4 Organization of the Report	2
1.5 Expected Justification	2

CHAPTER 2: Background 3-7

2.1 Introduction	3
2.2 System Development Life cycle	3-4

2.3 Phases of software development life cycle	5
2.4 System study/initial idea	5-6
2.5 Feasibility study	6
2.6 Apps Comparison	6-7

CHAPTER 3: Requirement Specification **8-12**

3.1 Introduction	8
3.2 Permission requirement	8
3.3 Threats to system security	8-9
3.4 Admin/Web Area	9-11
3.5 Database security	11
3.6 Conclusion	12

CHAPTER 4: Design Specification **13-41**

4.1 Design & Prototyping	13
4.2 User interface Prototyping	14
4.3 ER Diagram	15
4.4 DFD Diagram	16
4.5 Technology used	17
4.6 App Start	18
4.7 Works	19
4.8 Post	20

4.9 Search	21
4.10 Skills	22
4.11 Profile	23
4.11 Final output	24-41

CHAPTER 5: Implementation & Testing **42-44**

5.1 Introduction	42
5.2 How it works	42
5.3 Goals of Hire Me	42
5.4 Testing	43
5.5 Automatic User Interface Test	43
5.6 Test UI Performance	44
5.7 Security Testing	44
5.8 System Testing	44

CHAPTER 6: Conclusion & Feature Work **45**

6.1 Conclusion	45
6.2 Feature Implement	45

REFERENCES **46-47**

APPENDIX	48 - 57
Plagirism Result	58 - 59
LIST OF FIGURES	
Figure: 2.1: Process of System Development Life Cycle	4
Figure: 2.2: Different Phase of Software Development Life Cycle	5
Figure: 3.1: Client Message Database	9
Figure: 3.2: Admin Panel	10
Figure: 3.3: Admin Database	11
Figure: 3.4: Firebase Real Time Database Rules	11
Figure: 4.1: User Class Diagram	13
Figure: 4.2: User Interface Prototyping	14
Figure: 4.3: ER Diagram	15
Figure: 4.4: DFD Diagram	16
Figure: 4.5: How Android App Start	18
Figure: 4.6: Post Database	19
Figure: 4.7: post Fragment	20
Figure 4.8: Search Database	21
Figure 4.9: Skill Page	22
Figure 4.10: User Profile	23
Figure 4.10.1: Splash Screen (Mobile View)	24
Figure 4.10.2: Sign in page	25
Figure: 4.10.3: Register Page [1]	26

Figure: 4.10.4: Forget Password	27
Figure: 4.10.5: Works Page	28
Figure: 4.10.6: Post Page	29
Figure: 4.10.7: Search Page	30
Figure: 4.10.8: Search Fragment	31
Figure: 4.10.9: Skills page	32
Figure: 4.10.10: User Profile Page	33
Figure: 4.10.11: Post update page	34
Figure: 4.10.12: User profile update page	35
Figure: 4.10.13: Home page	36
Figure: 4.10.14: Features	37
Figure: 4.10.15: Preview Page	38
Figure: 4.10.16: Contact Page	39
Figure: 4.10.17: Admin Panel	39
Figure: 4.10.18: Real Time Search	40
Figure: 4.10.19: Admin Registration	41
Figure: 4.10.20: Admin Sign In	41

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

We aimed to develop an app that can create a part-time income scope for people. So I have taken the project named Hire Me. People around us need service in their daily life. We face different types of problems. But there are also the experts around us who can solve the problems.

Our app will create communication between the user and the experts. We are also providing income scope to the students too. Our app has a quick search and post system. For a particular problem, the user can post or search for the experts. By seeing your post experts will call you or by search feature you also can search the expert's then contract with them. These things are the core functionalities of our app [1].

1.2 OBJECTIVE

Unemployment [2] is a big problem in our country. The job fields are not creating as the way students are graduating. Even many experts in different task can't promote their ability to others as there is no common platform. We want to help them to interact with the Users. Our app will help the experts to find works and ensure themselves an income scope.

1.3 METHODOLOGY

Our project is an android based app. You have to run it via an android operated system mobile. There must need an internet connection. We have used firebase as our database. It can provide real-time data. In the search feature of our app user can quickly find the experts list according to their real-time alive. Our database is secure and only the registered user can access the data. While registration user has to provide their NID. Users can't use multiple NID while opening an account. For Security, we can access the exact location of the users and can take the help of his NID and profile picture. For the users the interface is developed in a very simple may as the UI can be more user-friendly and easy to use.

1.4 ORGANIZATION OF THE REPORT

The report contains 6 chapters. In chapter 1 the intro, methodology and project objective has described. In chapter 2 we have described about the background of the system. How the whole system is actually gone through in this chapter we have described. Chapter 3 contains the requirement and specification. It also contains the feature of firebase and the admin panel web sites information. This chapter also describes the web landing page of our project. In chapter 4 the whole project structure and the final output has been described. Chapter 5 contains the system test and the result of the project.

1.5 EXPECTED JUSTIFICATION

Our aim was to make services easy and quick. User can solve their daily life problem easily by the respective experts around. This app will make a bridge between the experts and the users. This app is in the initial stage so much more development is actually needed to ensure the security.

CHAPTER 2

BACKGROUND

2.1 Introduction

Our app UI has some functionality. We have used a bottom navigation with 5 buttons. All are in a form of fragment in an activity named Homepage. We also added others activities, via fragment we can reach those activity. Some features are listed below:

- I. Online Login/Registration
- II. Posting
- III. Mailing
- IV. Searching any expert
- V. View distance of the experts and the post
- VI. Communicate with expert via app
- VII. Sending email and make call to the expert

2.2 System Development life cycle

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system. An effective System Development Life Cycle should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

System Development Life Cycle (SDLC) means combination of various activities for developing a system. In other words, we can say that various activities put together to develop a system are referred to as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means Software Development.

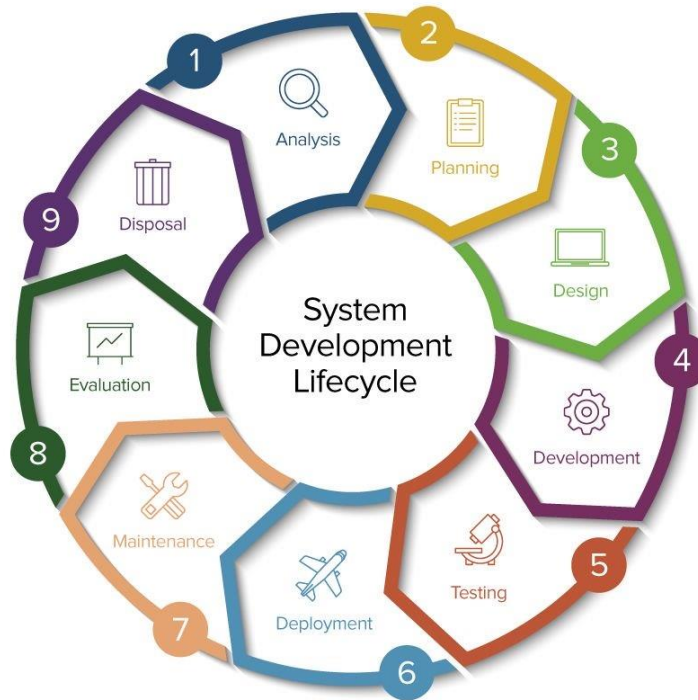


Figure 2.1: Process of System Development Life Cycle [3]

Following are the different phases of software development cycle:

- I. Preliminary study
- II. Feasibility study
- III. Detailed system study
- IV. System analysis
- V. System design
- VI. Coding
- VII. Testing
- VIII. Implementation
- IX. Maintenance
- X. Review

2.3 Phases of Software Development Life Cycle

This section describes the different phases and related activities of system development life cycle in details. Here the figure 2.3 that will show the system development life cycle process



Figure 2.2: Different phase of Software Development Process [4]

2.4 System Study/Initial Idea

This study is the first stage of System Development Life Cycle. This stage gives a clear picture of what actually the physical system is? After completing the system study, a system proposal is prepared by the system analyst (who studies the system) and placed before the user. All projects must start with an initial idea. Usually, this consists of a brief definition on what is the project all

about, what is its purpose and what the project aims to accomplish. How will the success of the project be measured.

To describe the system study phases more analytically, we would say that system study phases through the following steps:

- I. Problem identification & project initiation
- II. Background analysis
- III. Interface of findings

2.5 Feasibility Study

Expanding on the Initial Idea, the Feasibility Study involves drawing up the terms of reference, which state the objectives and scope of the project, how long it should take and how the results should be presented. The terms of reference are usually drawn up by senior management. The feasibility study must determine if the development of the project is justified in terms of economic and organizational terms.

The main role of the analyst in the feasibility study is to analyze the current system at a high level. Data Flow Diagrams (DFD) are used to describe how the current system performs and to illustrate known problems.

The system I was trying to build a lot of potentials. The application system for helping the user called Hire Me with many unknown person and also helping an unemployed person to earn money all over the world. The platforms I have chosen are Android Studio, Java programming language and Firebase to stored data.

2.6 Apps Comparison

From a particular feature, this project is similar like an existing app title is “Sheba.xyz - Online Service Market”. But this project has more feature than sheba.xys and it is more significant. In Hire Me the new feature is online experts searching system. All experts can contribute in any problem. Suppose one general user given a post in the app. And after that this post will go to all the similar experts. And one or more experts may handle the project. That’s why this feature make our app more significant. .

There also another similar project named is “Digital Manush”. Also that project is little bit similar with Hire Me. But there have many limitations. They made it with some fixed data. And user have to must use these data. And also that solution is for particular location. Only for Dhaka city. But Hire Me is for world wide. Anyone can open account from any part of world and also he/she may find expert from his location around. So this feature makes my app more significant than ‘Digital Manush’.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Introduction

Every software system must need some requirements. Our app has also some requirement specification [5]. Our app must need an android to run. It only works in android OS. Not yet developed for iPhone operating system. We need permission to use firebase database. Initially firebase is providing 1 GB storage and 30 GB of data transfer limit free. If the limitation cross then it will be paid. But as testing phase it is enough but later if user increases then the paid version can be used.

3.2 Permission Requirement

To use any android application, user must need provide some permission. One thing is very important here and that is, Android has launched total 16 versions. Android (A-P). From android M (Marshmallow) there a new option added which is called android runtime permission. But the lower version from android 'M' this option was not available. When any user install an app the all permission was automatically granted. But from android Mobile user must provide all permission individually. And all the state has followed perfectly in Hire Me.

The set of permissions are need for using Hire Me is listed below:

- I. Internet Permission
- II. Access user's COARSE Location
- III. Access user's FINE Location
- IV. Call Permission

If any user reject any permission then the particular feature will not be available for that user.

3.3 Threats to System Security

The most important things for a system is security. Our app has authentication system. User also need to register first and provide their NID to open an account. User can't use same NID multiple.

Time while registering. Our database is strongly secured as it is maintained by Google. We have an admin panel to observe the database. The admin panel can access all the information of the users. It will be easy to monitor the activities of the users.

3.4 Admin and Web Area

We design landing page for our apps promotion it's also called a promotional page. These landing page has two development part frontend and backend. For frontend we use HTML, CSS, JS and BOOTSTRAP [6] and for backend we use PHP and php framework LARAVEL [7]. In these web page user can got lots of benefit. Like they can contact us directly with us via contact us section. We will share our information here, so that user can gets us easily. These web page has four following section.

User satisfaction and security first priority always. So that we give an opportunity to folks directly contact with us. So that we create a contact us section where any user can leave us message. For these section we create MYSQL database [8]. So that user message store safely on our database and directly converted to our admin panel.

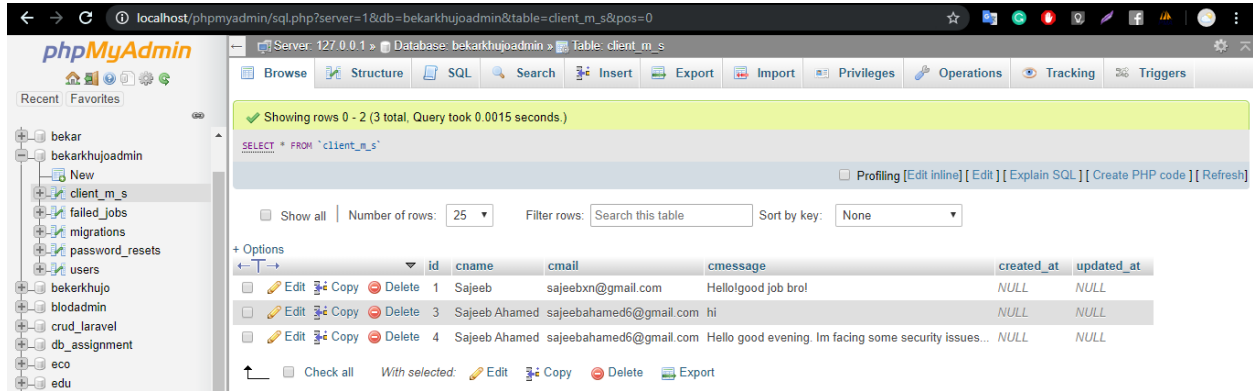


Figure 3.1: Client Message Database

Admin panel was developed by a PHP and php framework named Laravel [7]. In the website based admin panel we have connected the firebase database [9] of our app with it. The website can be monitored by the admin. All kind of 3rd party verification and security issue will be observed here. This access will be given only to the persons who will be act as admin. This Dashboard is easy to monitor and quick to fetch the information of the users. Any harmful posts will be instantly deleted by using this admin panel.

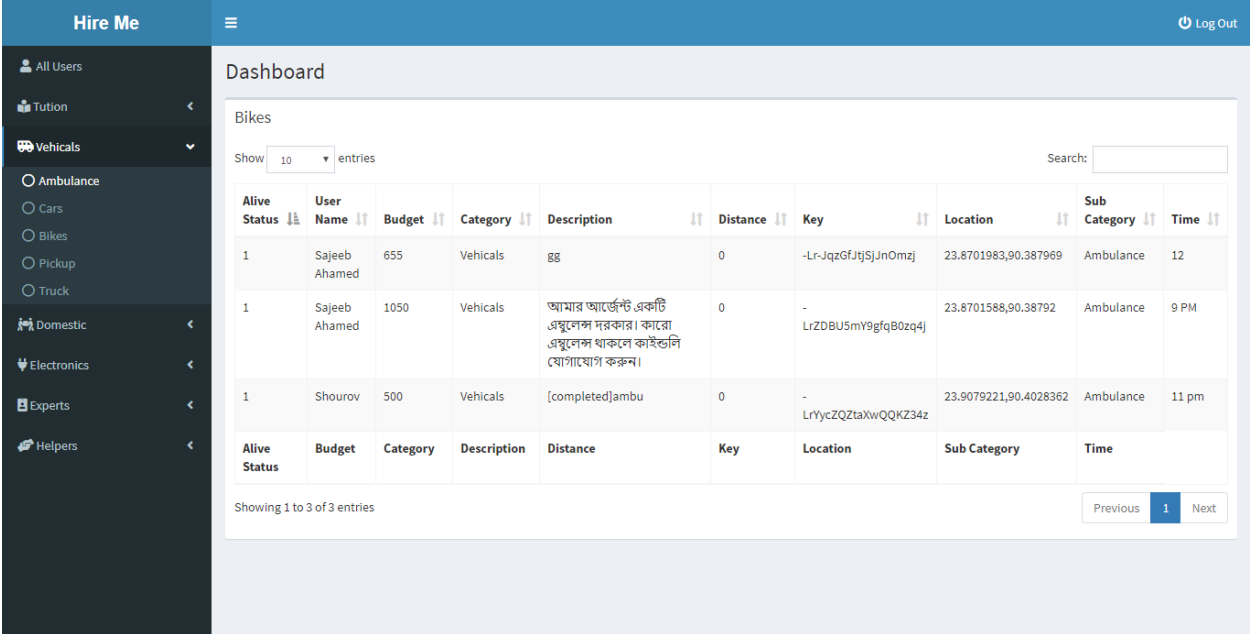


Figure 3.2: Admin Panel

For admin panel we strictly followed some rules. Like they must have verified email address what we officially provided. And then they must be completed their signup and sign in process.

And for the membership process we create separate database in MYSQL

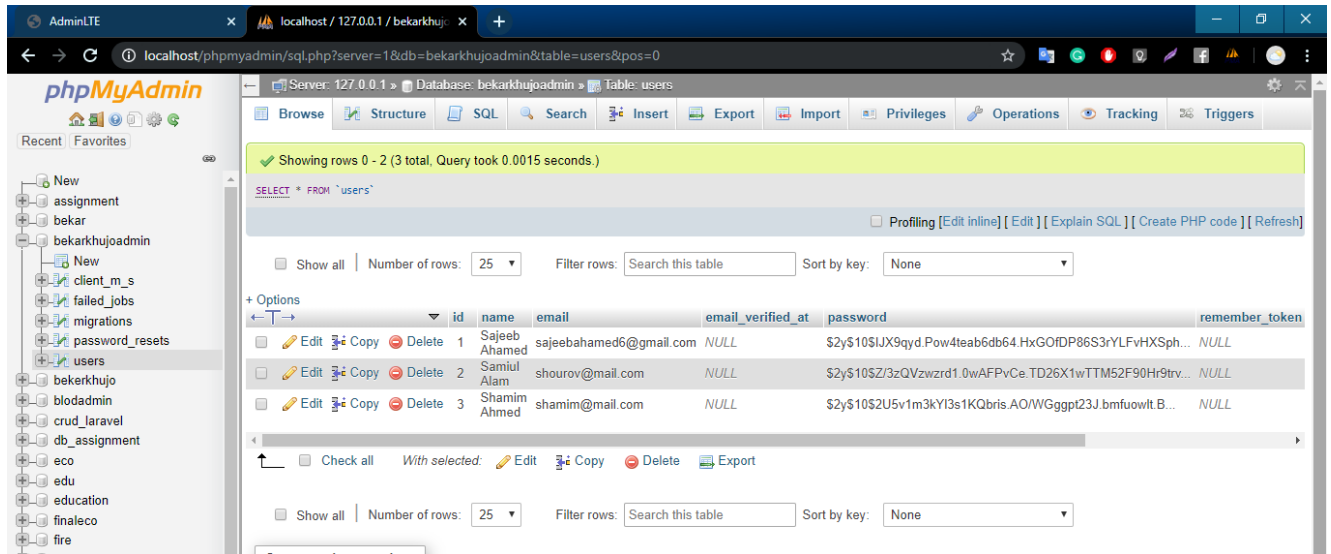


Figure 3.3: Admin Database [8]

3.5 Database Security

In this online database has been used. For online database Firebase Real time database has used. While using database there are an issue of giving security of these database. While using Firebase we don't have any headaches. Because this is Google's service and Google will provide the security. So there are no need extra security. And there are another option in firebase to protest data. We may turn off all write and read data from firebase.



Figure: 3.4 Firebase Real time Database Rules [9]

3.6 Conclusion

After completing full requirement specification Hire Me will work perfectly. Also we are trying to reduce the unemployment problem in Bangladesh though the amount is too less. And the person will find their desire service and the expert may find their earning source. If a person really be serious in this platform or work sincerely then he/she must gain from this system.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Design & Prototyping

In every application designing and prototyping is a very important phase. The design concept is important as the most user friendly interface can be more efficient and reliable for the users. A good designing can have a good feedback and user experience. Besides prototyping can provide advantage to the developer that how much improvement and designing is actually need.

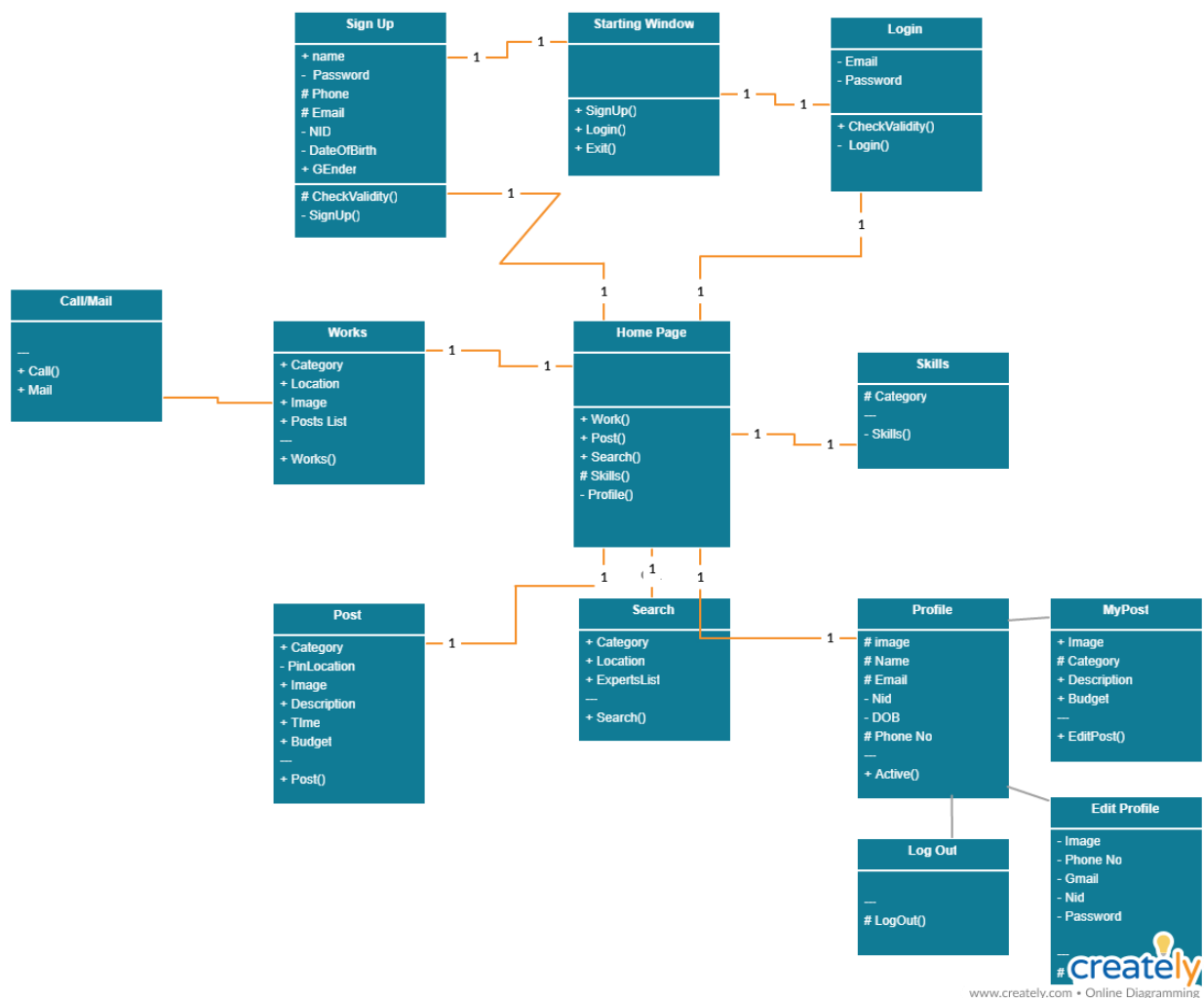


Figure: 4.1: Class Diagram [10]

4.2 User Interface Prototyping

Prototype is an initial sample of an application. Prototype is actually considered as an initial phase of an application. By the prototyping developer can understand the farther requirements and facilities which is desired by the users. So prototyping is really very much important to build a successful application. Prototyping is also uses for testing purpose. The test output can provide a point of view that how efficient and useable an application really is.

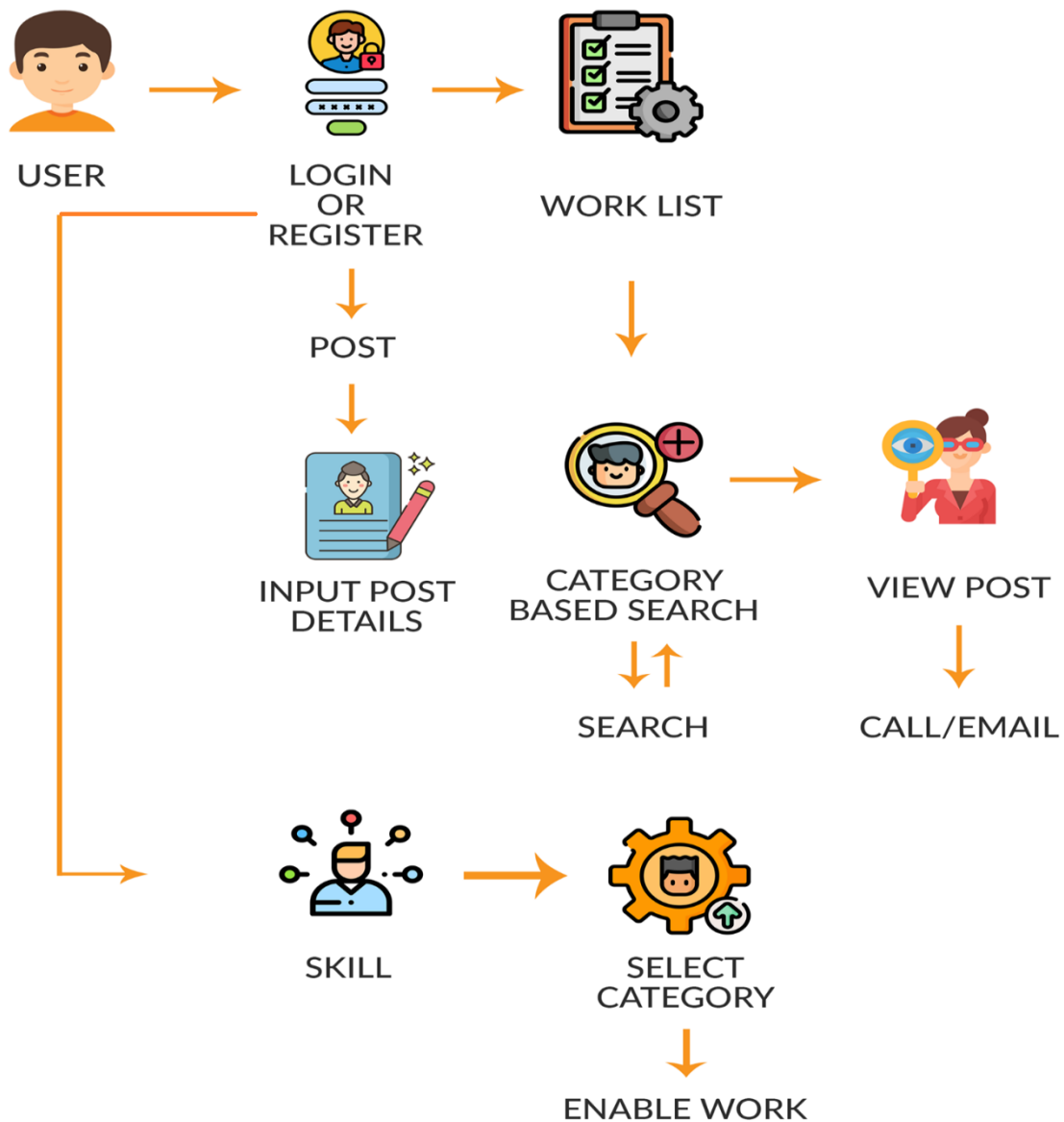


Figure 4.2: User Interface Prototyping

4.3 ER Diagram

Here is the ER diagram of our Database. For checking authentication user must need to provide email and password in the database. User can post their problem in the database to show in the works field. User need to provide alive state in the database to make himself visible in the search field. Profile database retrieve data of user in the app.

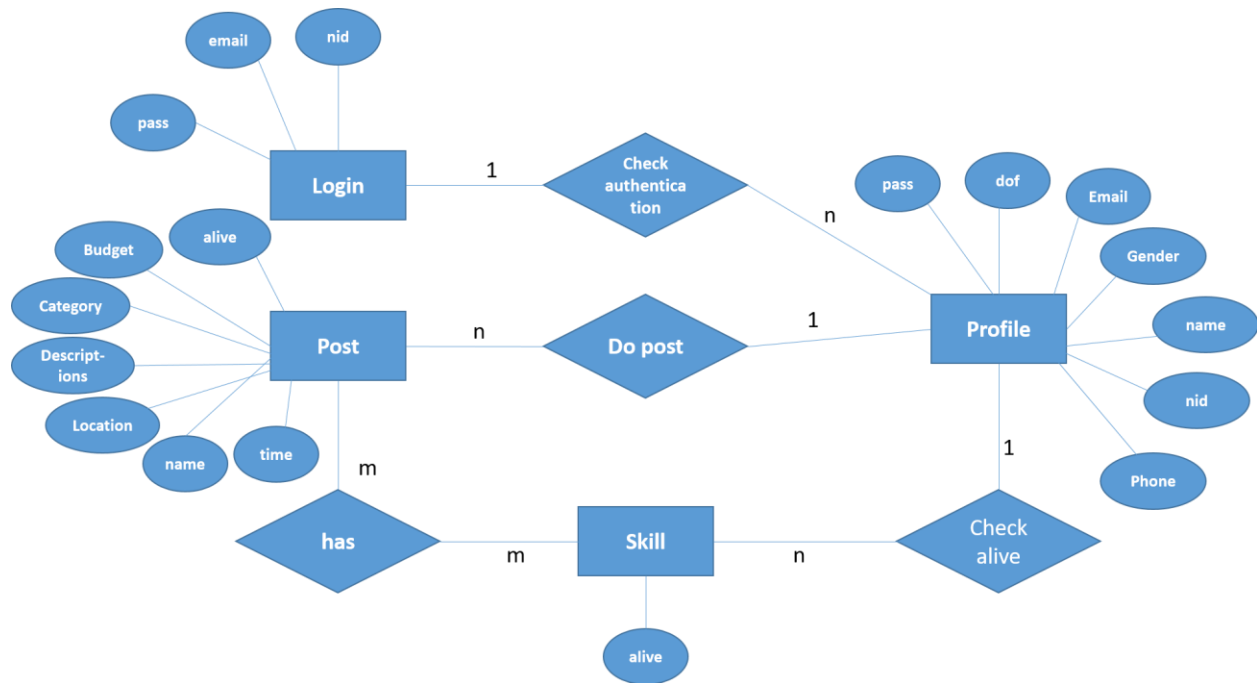


Figure 4.3: ER Diagram

4.4 DFD Diagram

Here is the DFD diagram of our project. Our system initially request for authentication. After authentication user can access post, search and skills data. Post data is responsible for posting. Skill data is to determiner a user skills which he can perform. Profile data is used to provide information of user.

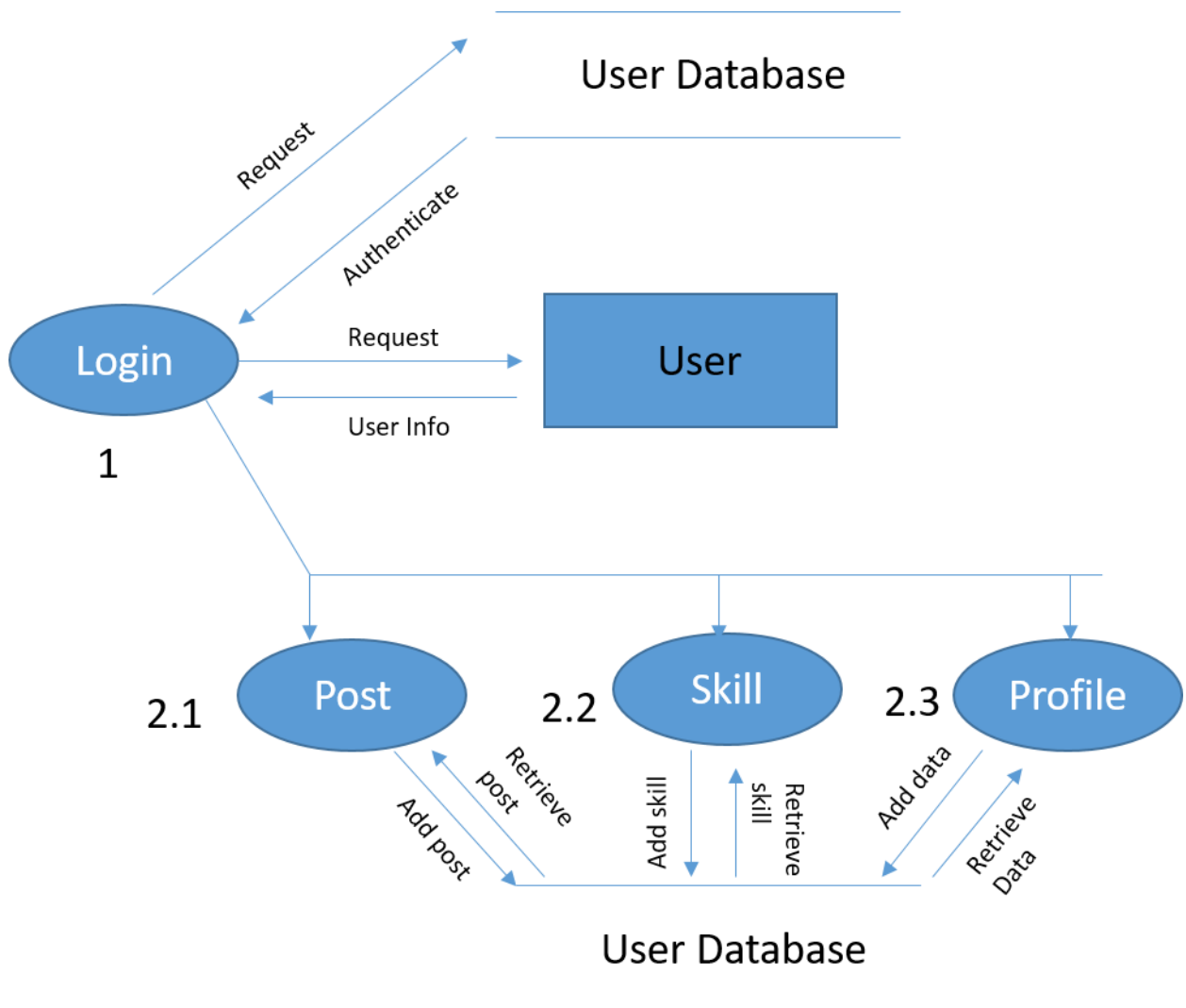


Figure 4.4: DFD Diagram

4.5 Technology Used

In this app we have used android studio, android virtual device and java programming language.

Android Studio: Android studio [11] is an IDE for developing android app. It is a product of google. It has some facilities like debugging and testing. All of those facilities are free. Android studio contains designing tools with preview. We can see our designing in preview or can run in the virtual device. It also can use different libraries for the UI update and modification. We also can generate the signed apk by a click and also help to upload app in play store.

Android Studio contains SDK tools [13]. It is platform independent. SDK provides build tools, debugging tools and so on. Android studio has android SDK manager. It provides components, tools, resources and platform needed to develop a complete android app.

ADV Manager: ADV manager [12] is a testing device that runs on android emulator. Here we can install an apk file for testing. It runs on an emulator and provides a virtual environment. The virtual device can be configured by the user given specification.

Java: The whole project is developed in java programming language. Java is a strong programming language which can be run in different platform. Java program converted into byte code which is not human and machine readable. The code is converted by DVM then it can be readable by machine. It is secured and can run in different OS like windows, Linux and Mac OS. So java is more reliable and fast programming language. Java needs a software development kit which is actually an interpreter needs to make a complete application.

Firebase: Firebase is an online database system [9]. It is a product of Google. It provides the real time data management system which is very important. It makes the app management system easy and efficient. We can integrate multiple project in our firebase console. Initially in a given range it is free to use. It is giving facilities of Database, Analytics, Storage, Authentication, Hosting, crash report and AdMob etc.

4.6 App Start

Android applications are written in Java programming language. First of all, it is converted into byte code from the Java file. Then it is only readable by DVM. Then DVM converts it into a .dex file. There also exists Android Manifest and resources. Then a combination of all this generates an APK file which can be run in an Android device.

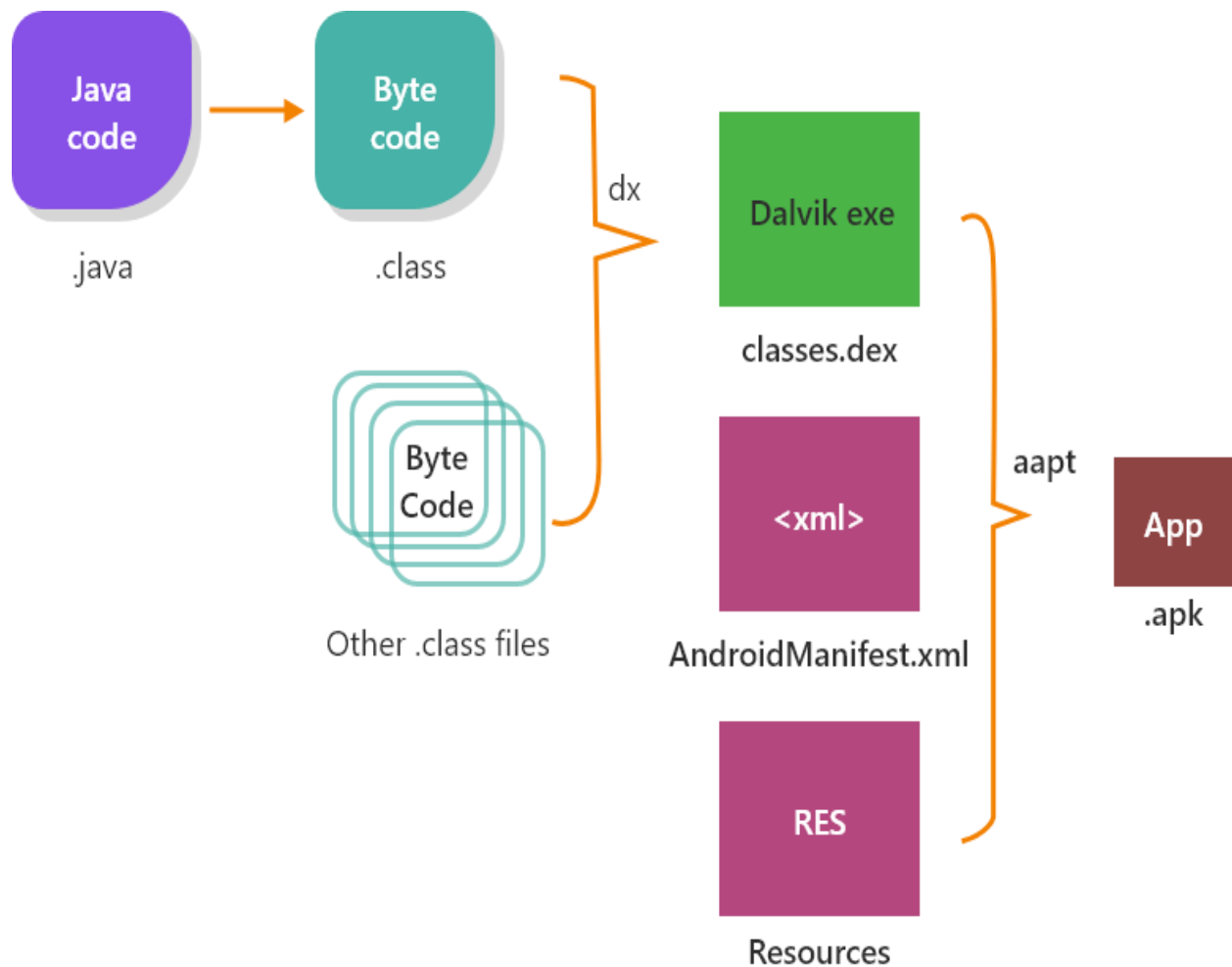


Figure 4.5: How android app start [11]

4.8 Post

In the post fragment User can post their required tasks according to category. User have to pin location to integrate his location with his post. User also can select picture for post. User can input description, time duration and budget for the work.

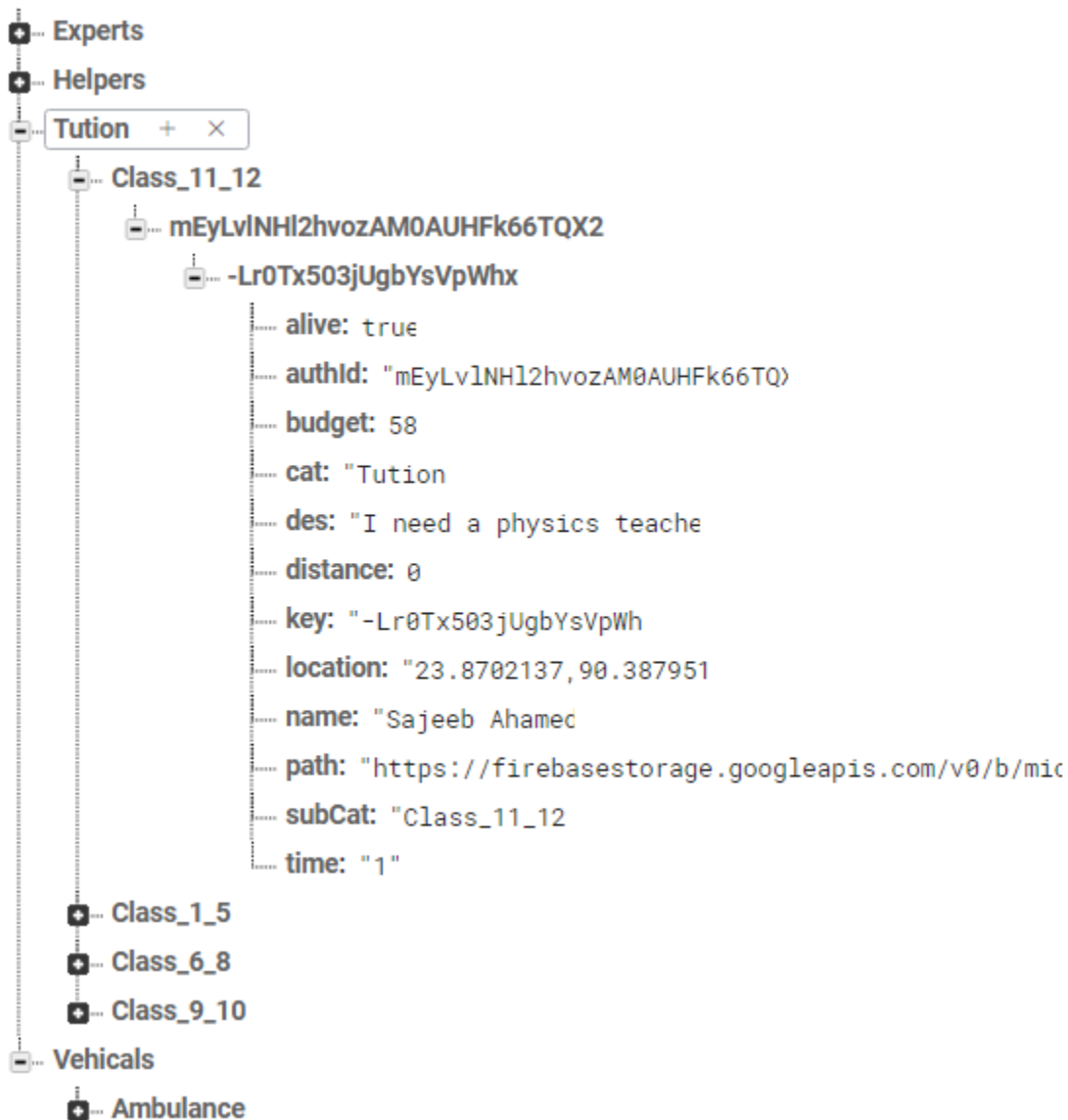


Figure 4.7: Post Fragment

4.9 Search

In the search fragment user can find the experts list according to category. User can call or mail his desired expert. User can notice the location and distance of the experts. Then they can find the nearest expert easily for a quick service. If an expert make himself active by enable the switch button in a particular skill then he will show in the list for work.

```
..... doctor: "live"  
..... driver: "live"  
..... electrician: "not"  
..... email: "alam@gmail.com"  
..... fridge: "not" X  
..... gas_Repairer: "not"  
..... image_url: "https://firebasestorage.googleapis.com/v0/b/mic  
..... lawyer: "live"  
..... live: true  
..... location: "23.9078974, 90.402841"  
..... locksmith: "not"  
..... maid: "not"  
..... mobile: "not"  
..... name: "alam"  
..... oven: "not"  
..... phone: "01521106776"  
..... photographer: "live"  
..... pickup: "not"  
..... plumber: "not"  
..... taxi: "not"  
..... television: "not"  
..... truck: "not"  
..... vhankti: "live"
```

Figure 4.8: Search Database

4.10 Skills

In the skills fragment Experts can enable the skills via switch button. Experts will just enable the skills which actually they want to perform. Experts will show in the list according to the enabled category.

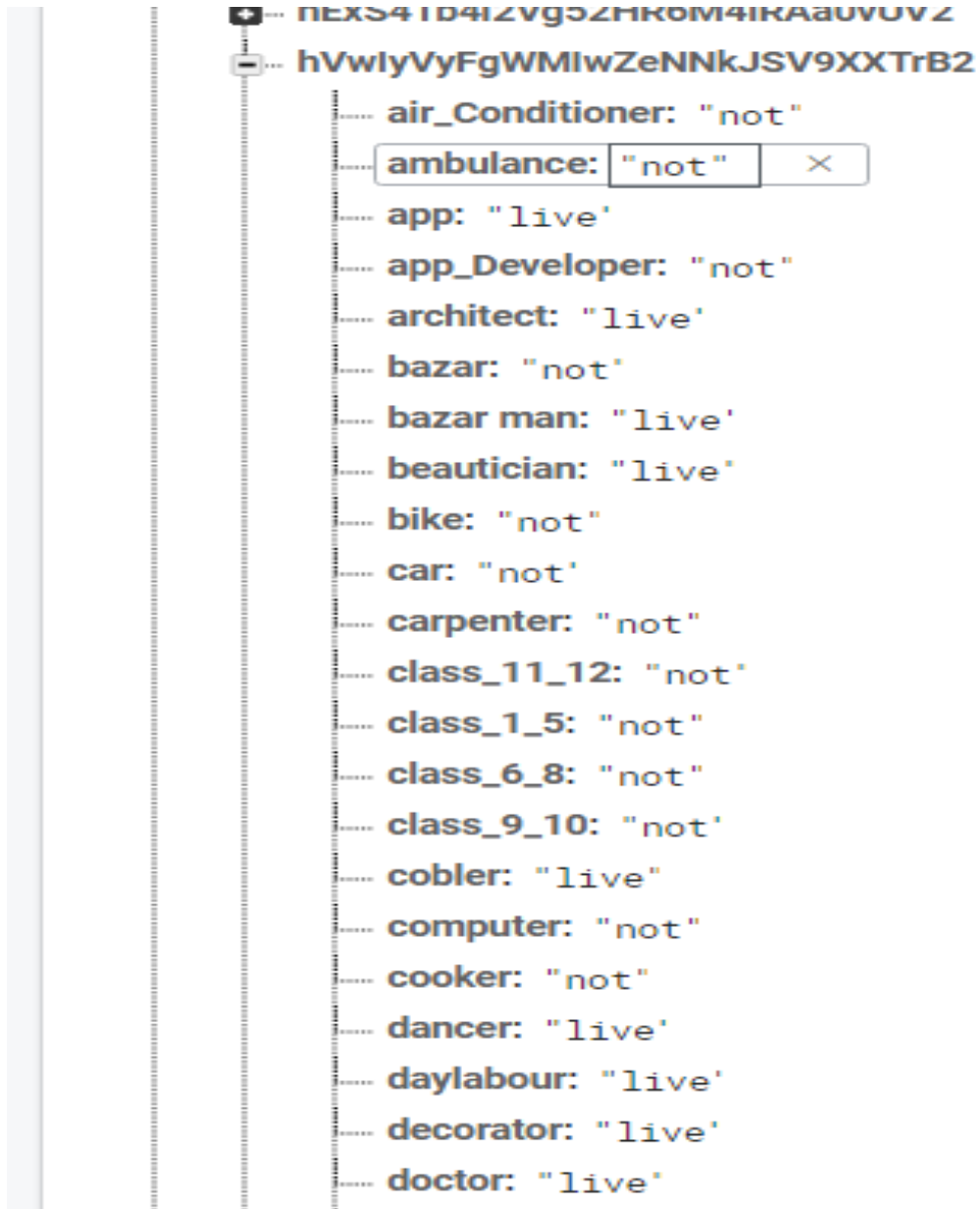


Figure 4.9: Skill Page

4.10.1 Profile

In the profile fragment User can see his details. The fragment contains an important feature named alive. If the user enable the alive option then he will be shown in the search list according to the enabled skills. If the user make himself disabled in the alive option then he will not be shown in the search list for all kind of skills. The fragment also contains my posts, logout and edit profile option. In my post activity user can find all his posts according to the category and user can edit his posts. Edit profile option is to edit user details. The logout option will make the user out of the app system.

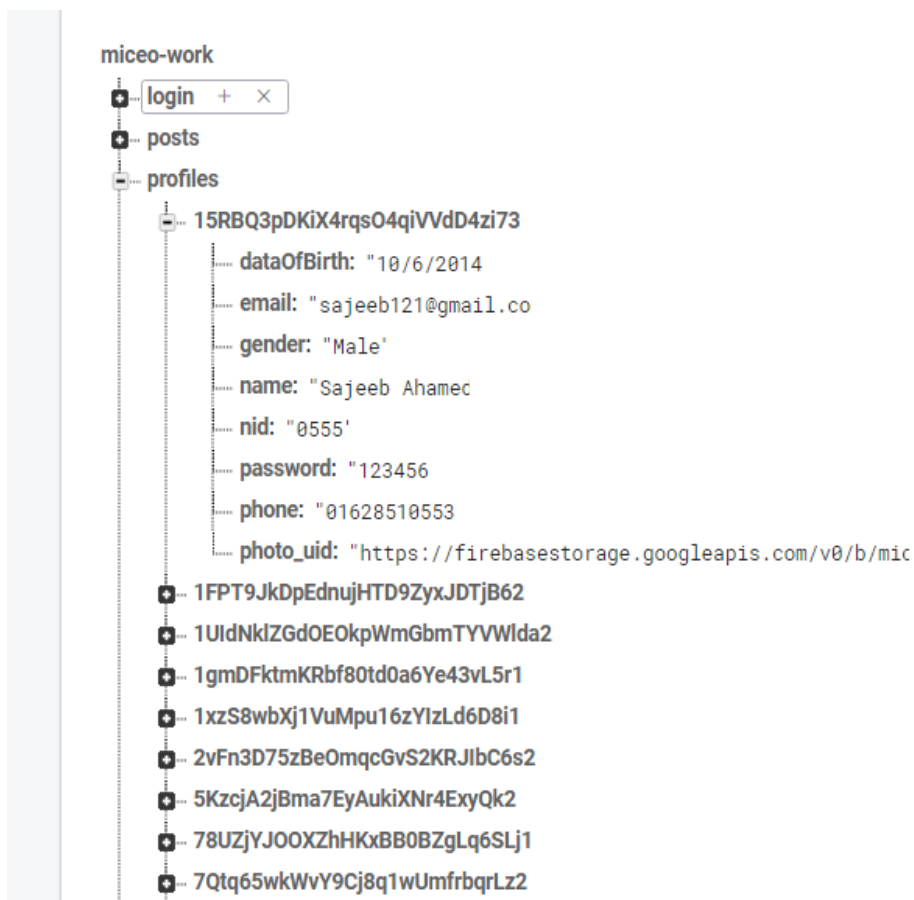


Figure 4.10: User Profile

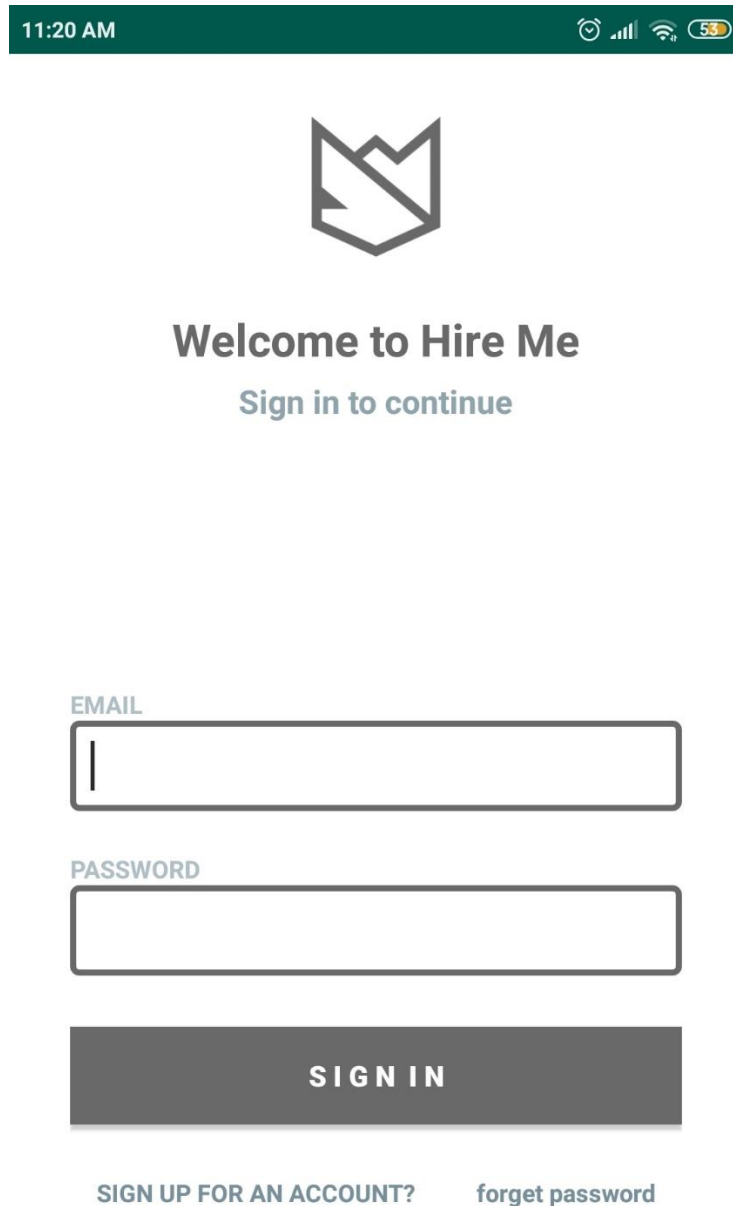
4.10.2 Final Output

The Figure 4.10.1 is the splash screen. This activity will remain for 3 seconds then it will disappear. Then it will enter in the login page.



Figure 4.10.1: Splash Screen (Mobile View)

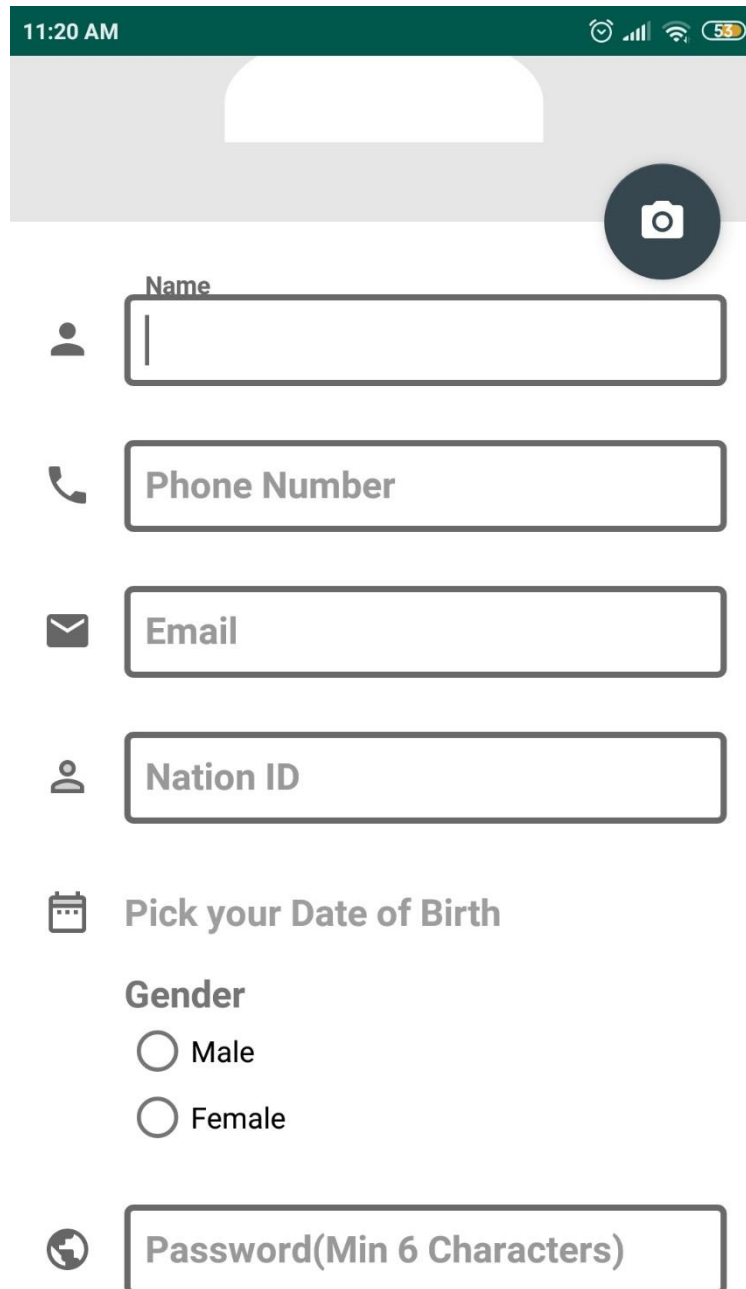
The Figure 4.10.2 is the login page. This page is for user authentication. We will only allow the registered user to enter in the app. The login system is based on firebase email and password authentication system.



The image shows a mobile application login screen. At the top, there is a dark green status bar with the time '11:20 AM' and icons for alarm, cellular signal, Wi-Fi, and 5G battery. Below the status bar is a stylized logo consisting of a square with a diagonal line and a smaller square inside. The main heading is 'Welcome to Hire Me' in a bold, dark font, followed by the sub-heading 'Sign in to continue' in a lighter blue font. There are two input fields: the first is labeled 'EMAIL' and contains a vertical cursor; the second is labeled 'PASSWORD'. Below the input fields is a dark grey button with the text 'SIGN IN' in white. At the bottom, there are two links: 'SIGN UP FOR AN ACCOUNT?' and 'forget password'.

Figure 4.10.2: Sign in page

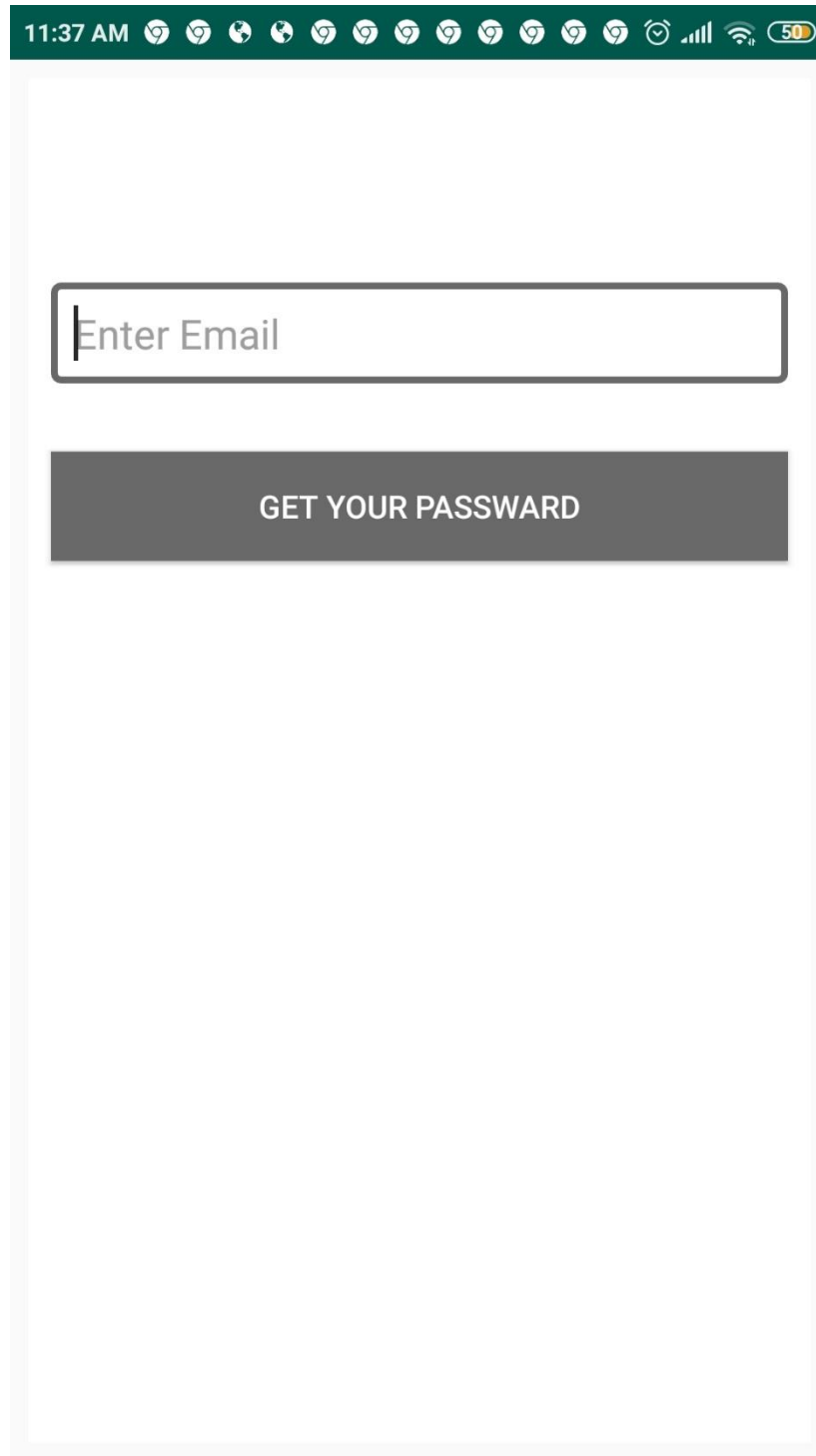
The Figure: 4.10.3 is the signup page. Here the registration process take place. User must need to fill up all the field .Then if user agree with the term and condition then he/she will be allowed to register for the app.



The image shows a mobile application registration screen. At the top, there is a dark green status bar with the time '11:20 AM' and icons for alarm, signal strength, Wi-Fi, and battery (53%). Below the status bar is a light gray header area with a white rounded rectangle and a circular camera icon on the right. The main form consists of several input fields, each with an icon to its left: a person icon for 'Name', a phone icon for 'Phone Number', an envelope icon for 'Email', another person icon for 'Nation ID', a calendar icon for 'Pick your Date of Birth', radio buttons for 'Gender' (Male and Female), and a globe icon for 'Password (Min 6 Characters)'.

Figure: 4.10.3: Register Page

The Figure: 4.10.4 is forget password field. In case of forgetting password user have to enter his email and NID for get back his/her password.



The image shows a mobile application interface for a password recovery feature. At the top, there is a dark green status bar with the time '11:37 AM' and various system icons. Below this, the main content area is white and contains a single text input field with the placeholder text 'Enter Email'. A dark grey button with the text 'GET YOUR PASSWORD' is positioned directly below the input field.

Figure: 4.10.4: Forget PasswordThe Figure: 4.10.5 is the works fragment situated in the homepage. This fragment contains all the works list posted by the users. Experts can call or mail the user for work.

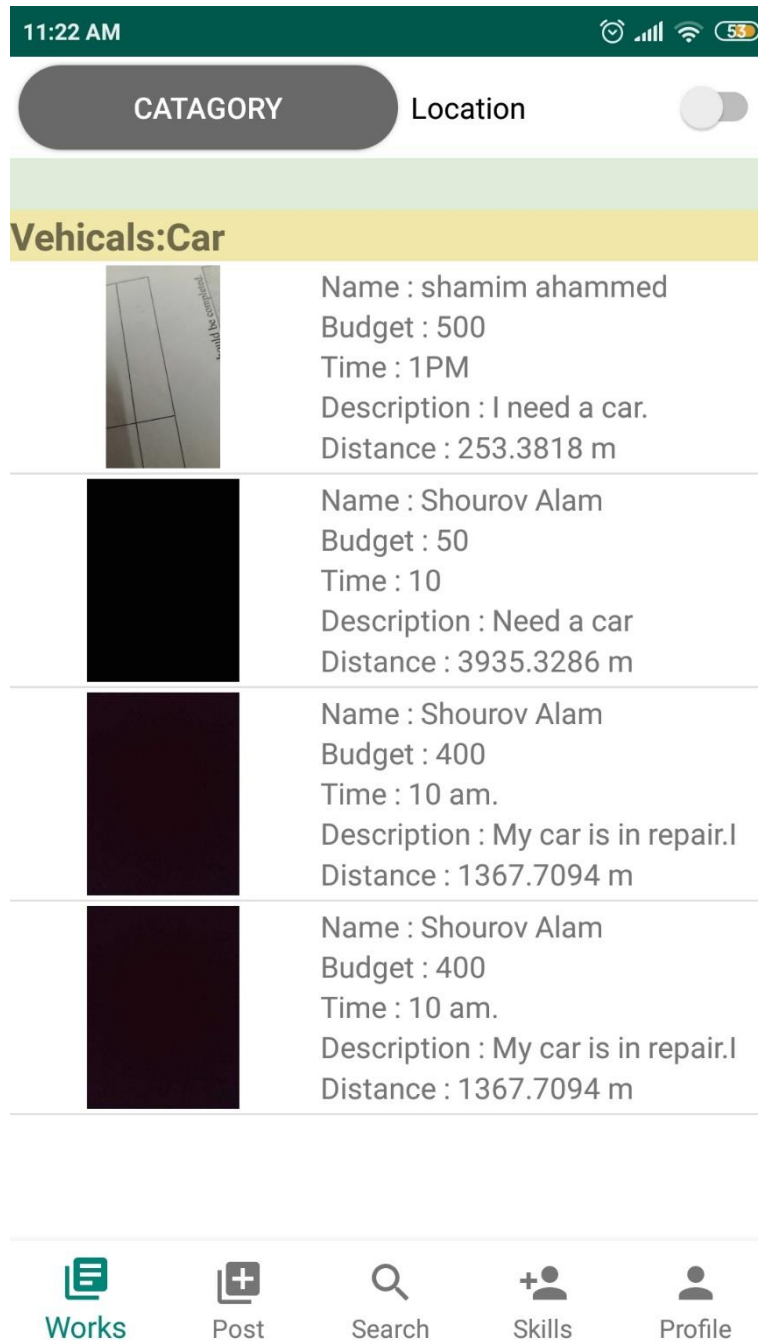


Figure: 4.10.5: Works Page

The Figure: 4.10.6 is the post fragment. Here user can post for his required task. User also can select his required task by category.

11:24 AM

CATAGOTY

Pin Location

PATH/pic.jpg

SELECT IMAGE

Write a short description

Enter Time or Duration

Enter Your Budget Here

POST

Works Post Search Skills Profile

Figure: 4.10.6: Post Page

The Figure: 4.10.7 is search fragment. In this fragment user can search the experts quickly according to category.

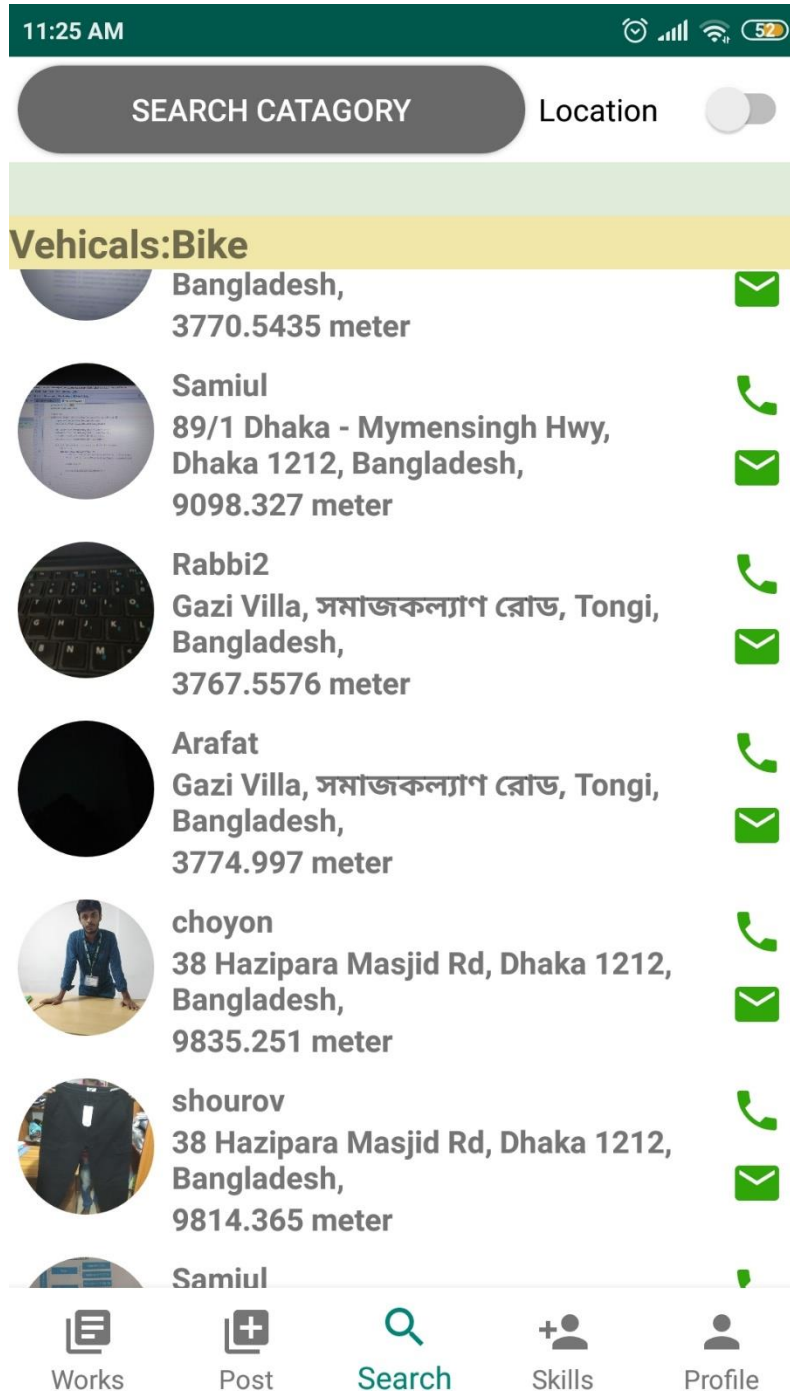


Figure: 4.10.7: Search Page

The Figure: 4.10.8 is also the search fragment. But it is little bit different. If user check the switch button then the experts in the 5 kilometers radius will be shown in the list.

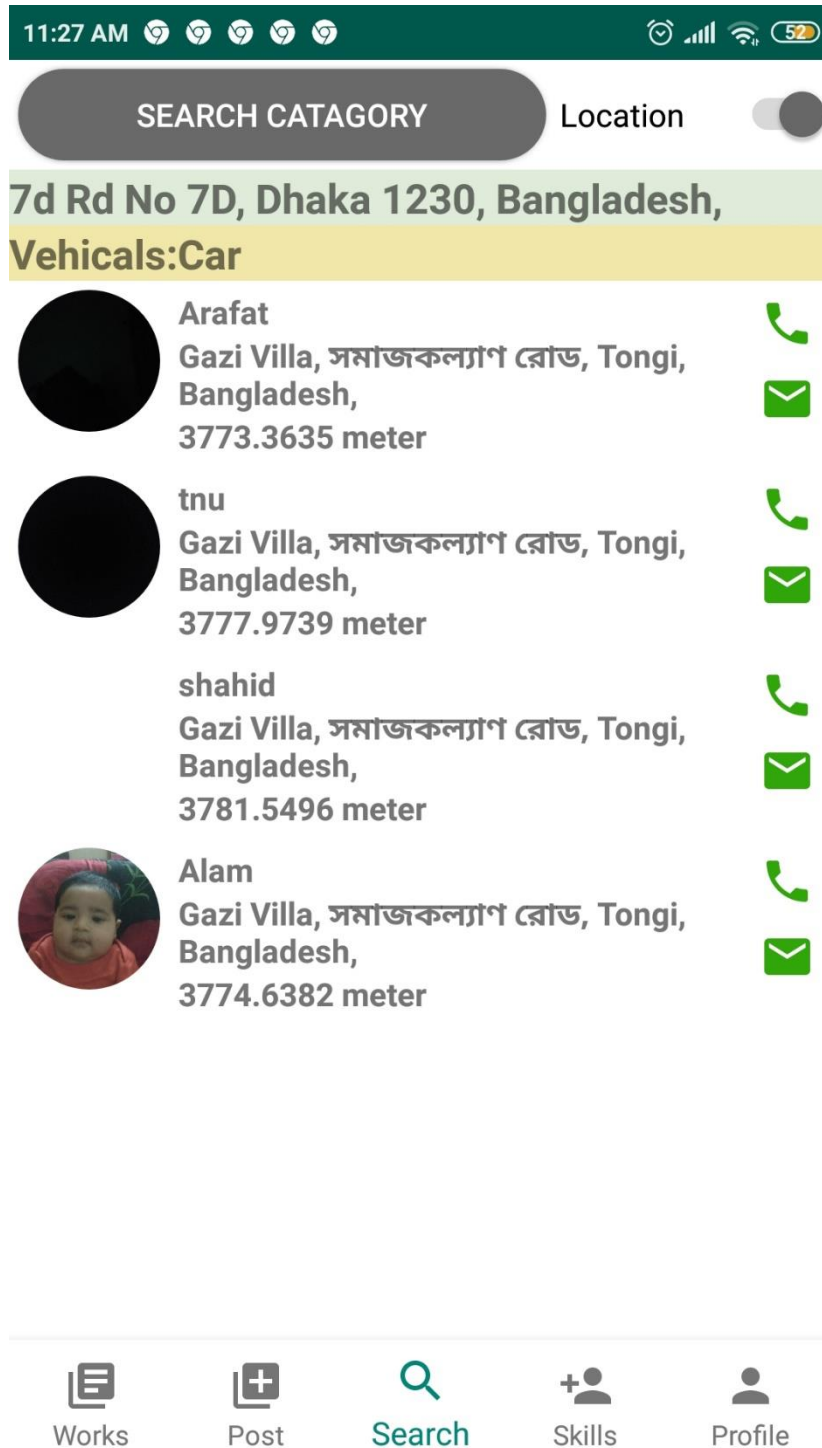


Figure: 4.10.8: Search Fragment

The Figure: 4.10.9 is the skills fragment. This fragment helps the experts to enable the category of work in which he/she is willing to do.

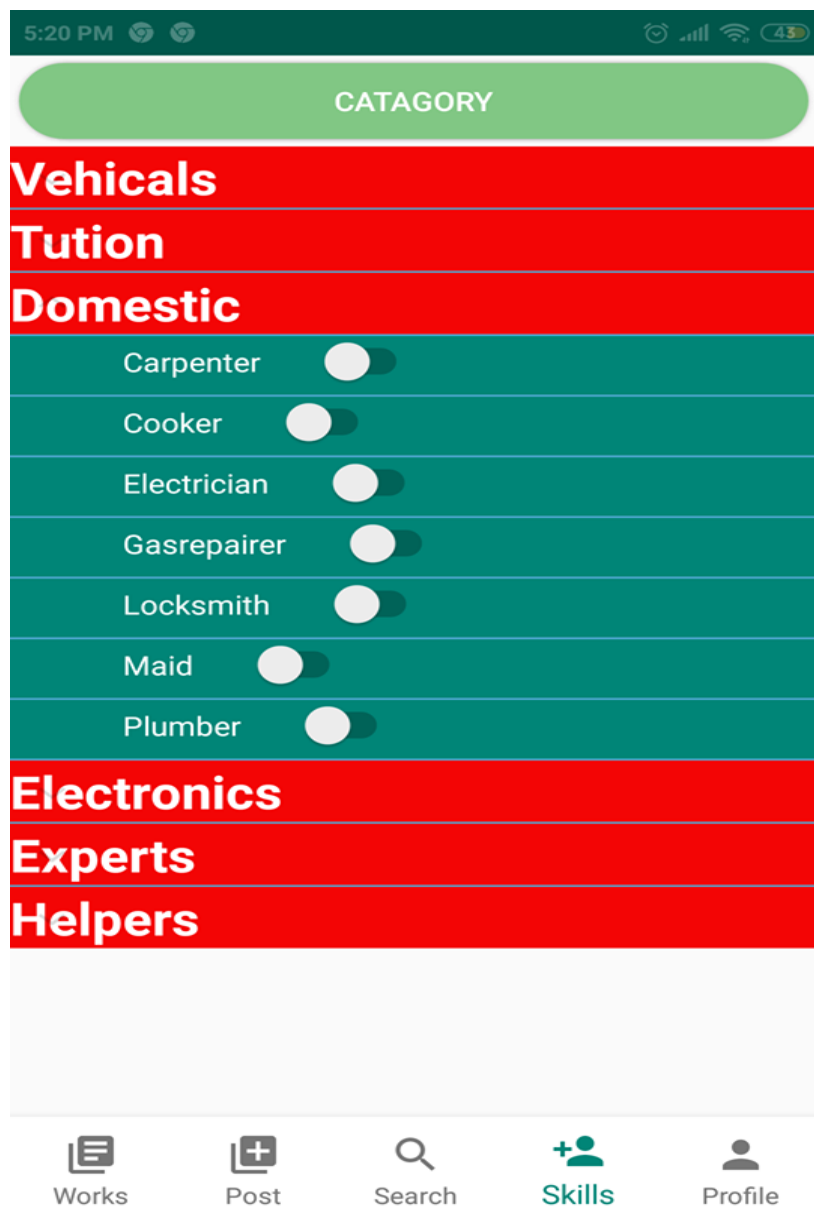


Figure: 4.10.9: Skills page

The Figure: 4.10.10 is the profile fragment. Here user can view his/her details. Even user can make himself active for the work he want to perform. Here are also some other option like logout, edit profile and my posts.



Figure: 4.10.10: User Profile Page

The Figure: 4.10.11 is the post edit page. Here user can edit his/her post which have already posted.

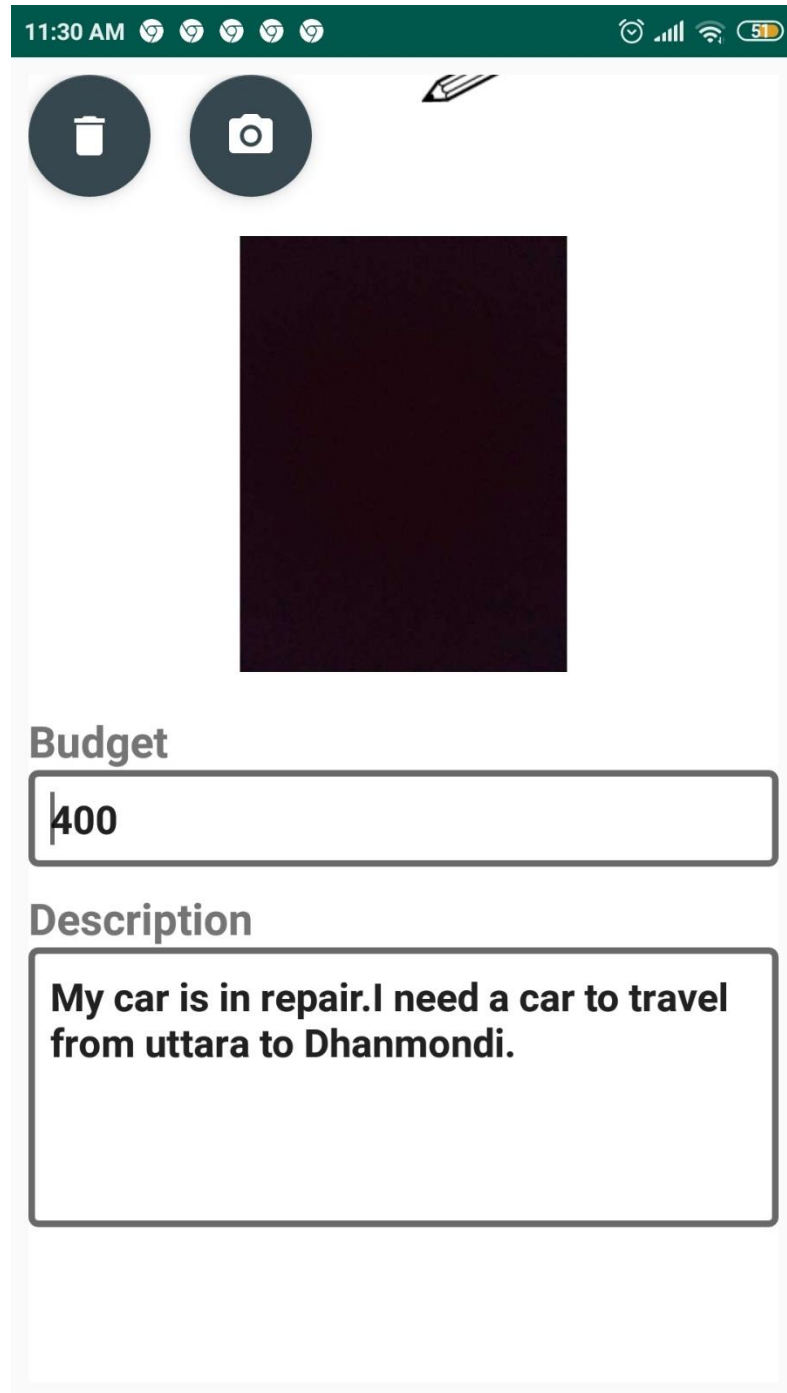
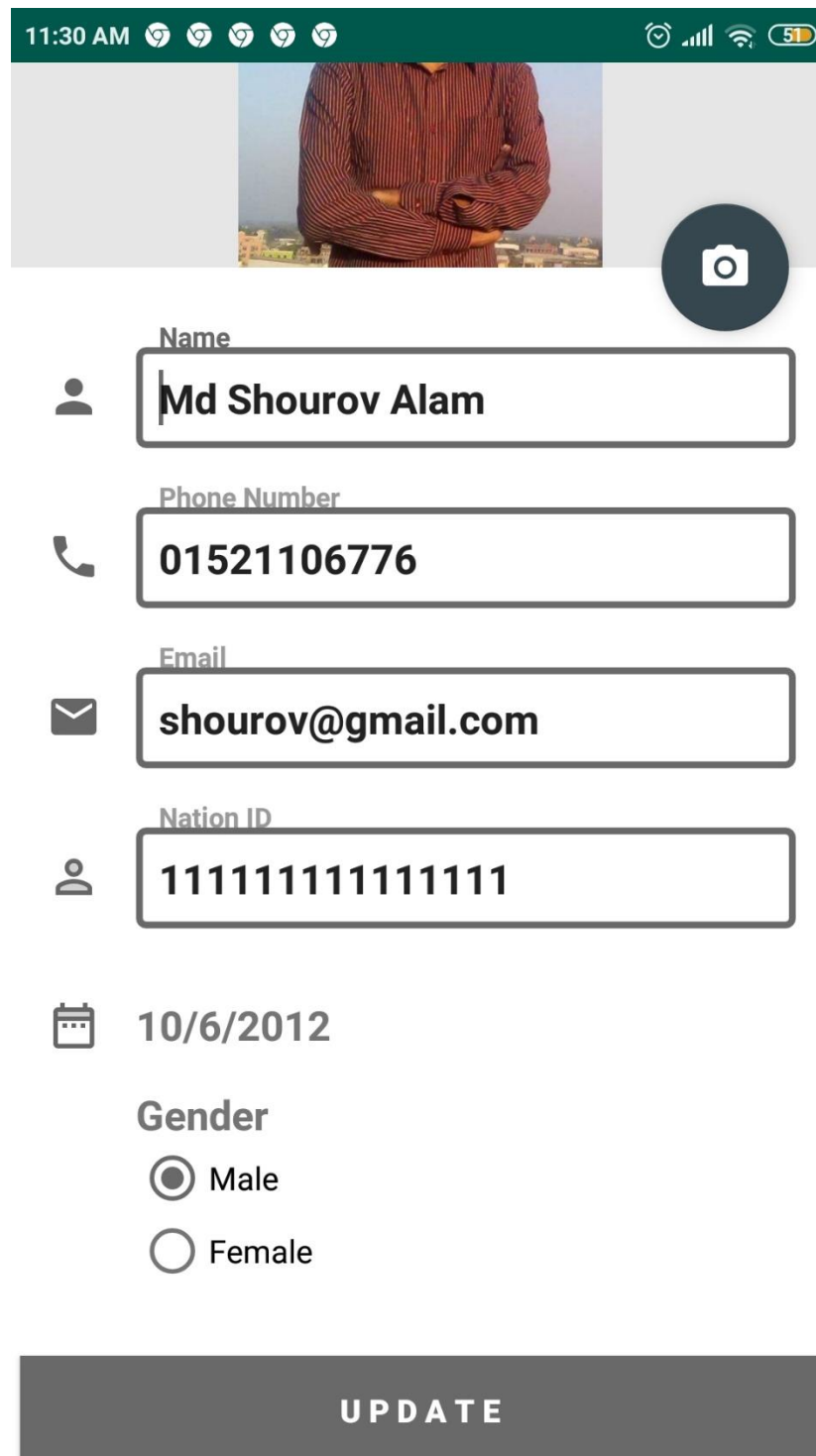


Figure: 4.10.11: Post update page

The Figure: 4.10.12 is the profile update page. Here user can update his/her profile.



The image shows a mobile application interface for updating a user profile. At the top, there is a status bar with the time 11:30 AM, signal strength, Wi-Fi, and battery icons. Below the status bar is a profile picture of a person in a red and white striped shirt, with a camera icon in the bottom right corner. The form contains the following fields:

- Name:** Md Shourov Alam
- Phone Number:** 01521106776
- Email:** shourov@gmail.com
- Nation ID:** 1111111111111111
- Date of Birth:** 10/6/2012
- Gender:** Male (selected), Female

At the bottom of the form is a large grey button labeled **UPDATE**.

Figure: 4.10.12: User profile update page

Web Part (Admin/Web)

HOME

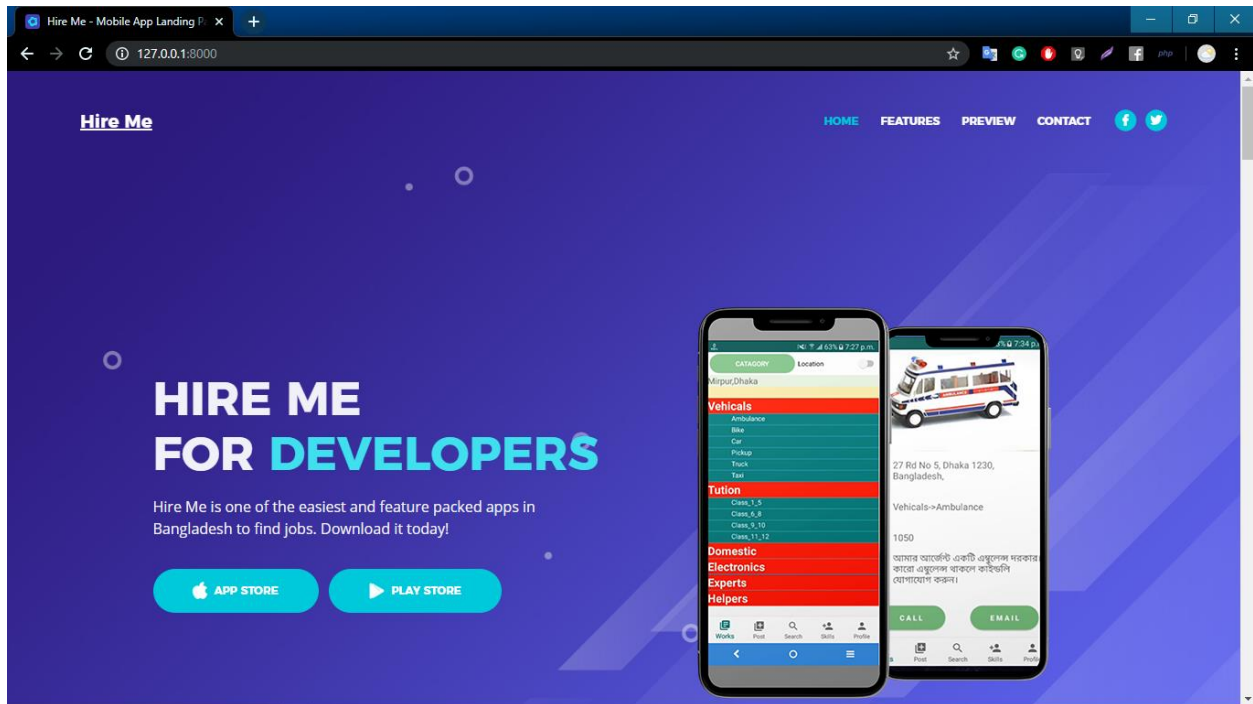


Figure 4.10.13: Home Page

Home sections includes app download section and an attractive slider. This is mainly for user attraction with apps download link so that user can easily get our application.

FEATURES

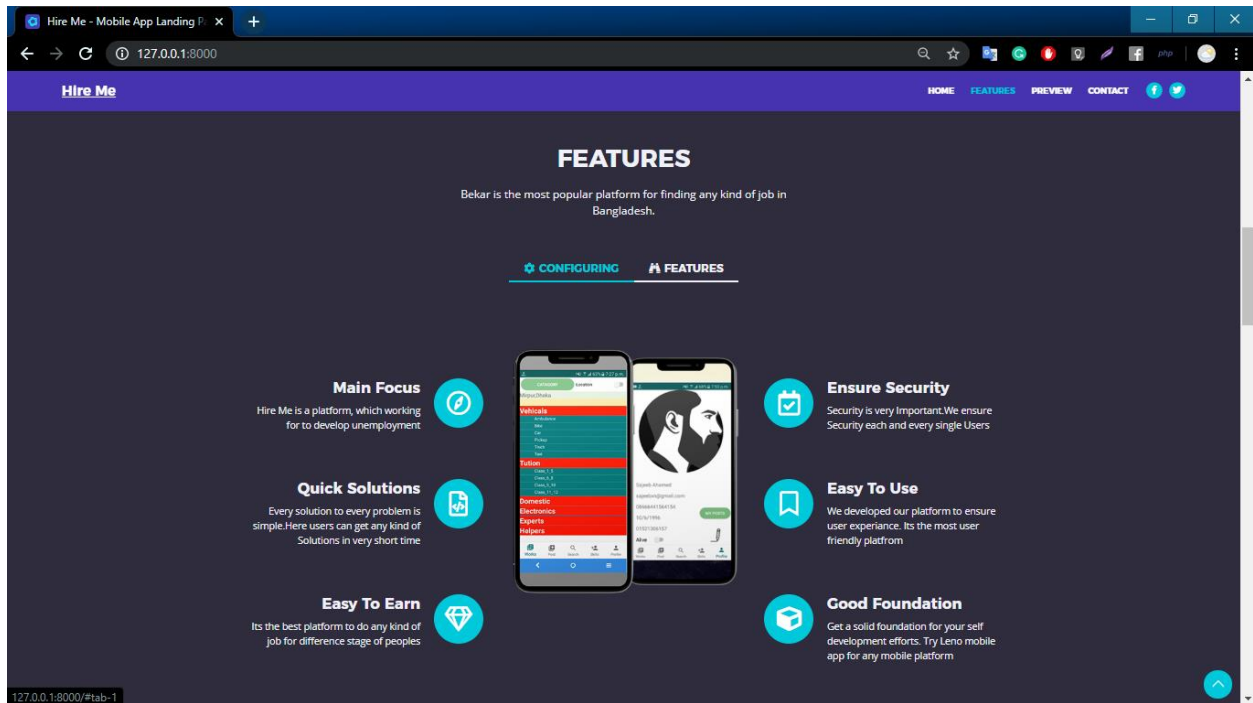


Figure: 4.10.14: Features

Features section represent our apps functionality and service what we provide for our users. In features section we try to ensure our main goal. Like what is our main focus, quick solution, security, earning ration etc.

PREVIEW

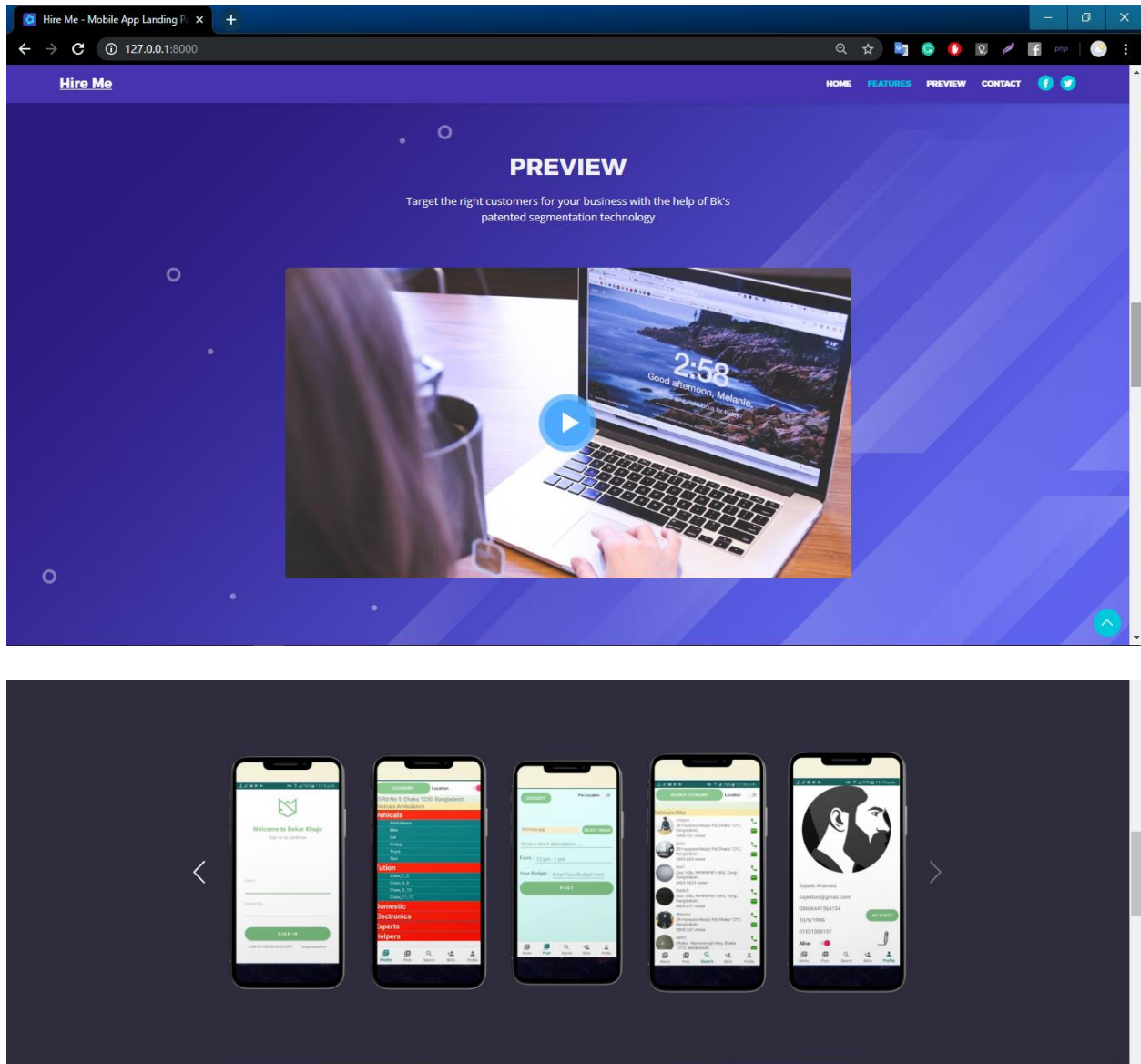


Figure 4.10.15: Preview Page

In preview section we try to represent our promotional video. So that any user can get idea easily about our plan so that they can compare our service to others. And we includes all of our screen so that any user can get idea about our application and service before installing our app.

CONTACT

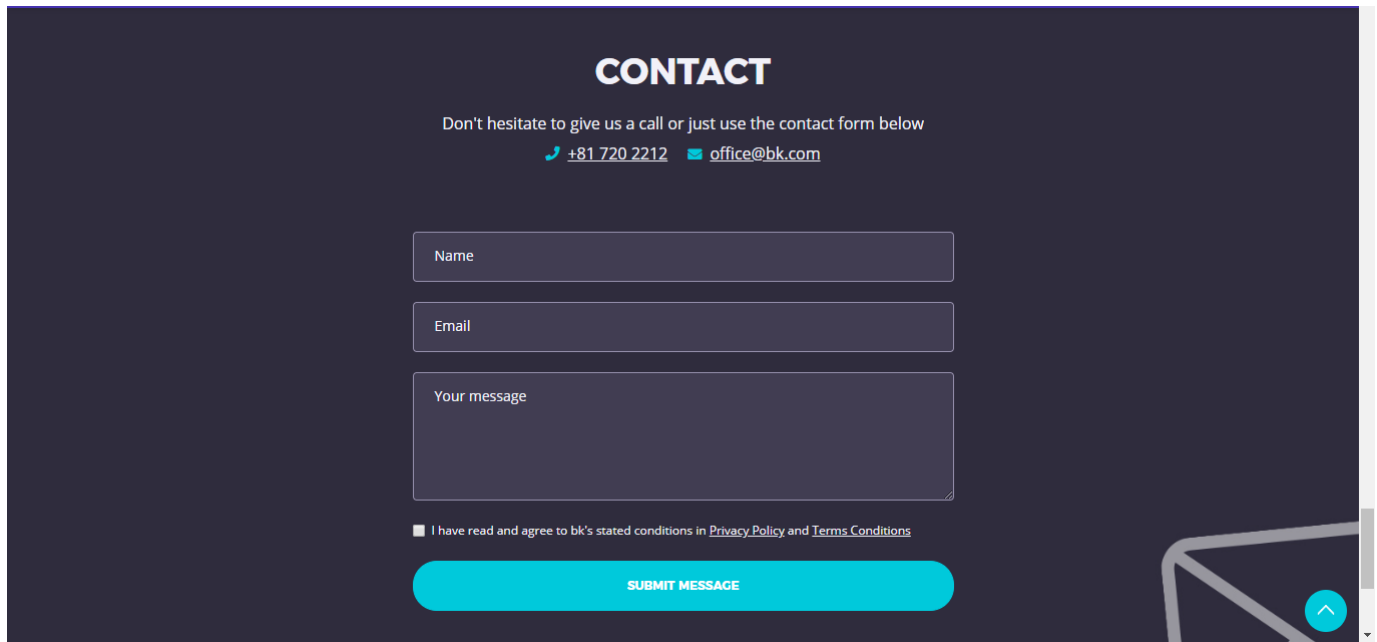


Figure 4.10.16: Contact Page

User satisfaction and security first priority always. So that we give an opportunity to folks directly contact with us. So that we create a contact us section where any user can leave us message.

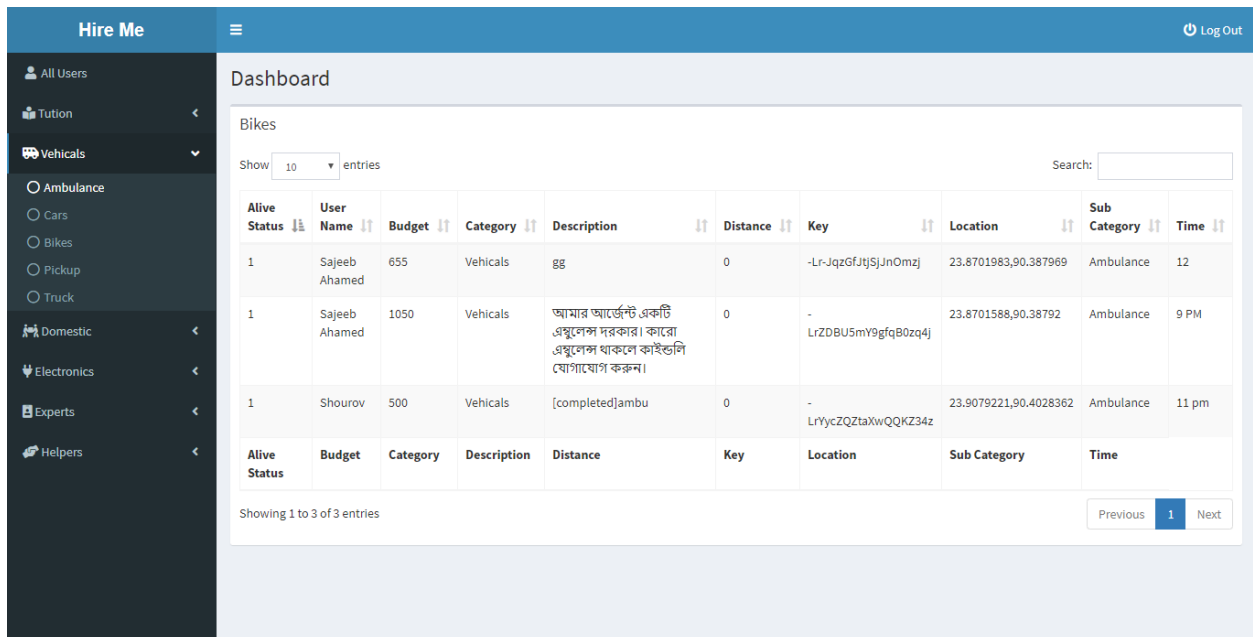


Figure 4.10.17: Admin Panel

Each and every section we used real-time search so that we can easily catch any unethical post or spammers.

The figure consists of two screenshots of a web application dashboard, both titled 'Hire Me' and featuring a 'Log Out' button in the top right corner. The left sidebar contains a navigation menu with categories like 'All Users', 'Tuition', 'Vehicals', 'Domestic', 'Electronics', 'Experts', and 'Helpers'. The main content area displays a 'Dashboard' with a 'Data Table With Full Features'.

Top Screenshot: User Search

The search bar contains the text 'sajeeb'. The table below shows the results:

Name	Gender	Phone	Nid	Email	Date of Birth	Picture
sajeeb	Male	019968235968	123455679	sa@mail.com	10/6/2013	
Sajeeb Ahamed	Male	01628510553	0555	sajeeb121@gmail.com	10/6/2014	
Sajeeb Ahamed	Male	01521306157	08666441564154	sajeebxn@gmail.com	10/6/1996	
Sajeeb1	Male	58465	49315	saj@gmail.com	10/6/2013	

Bottom Screenshot: Car Search

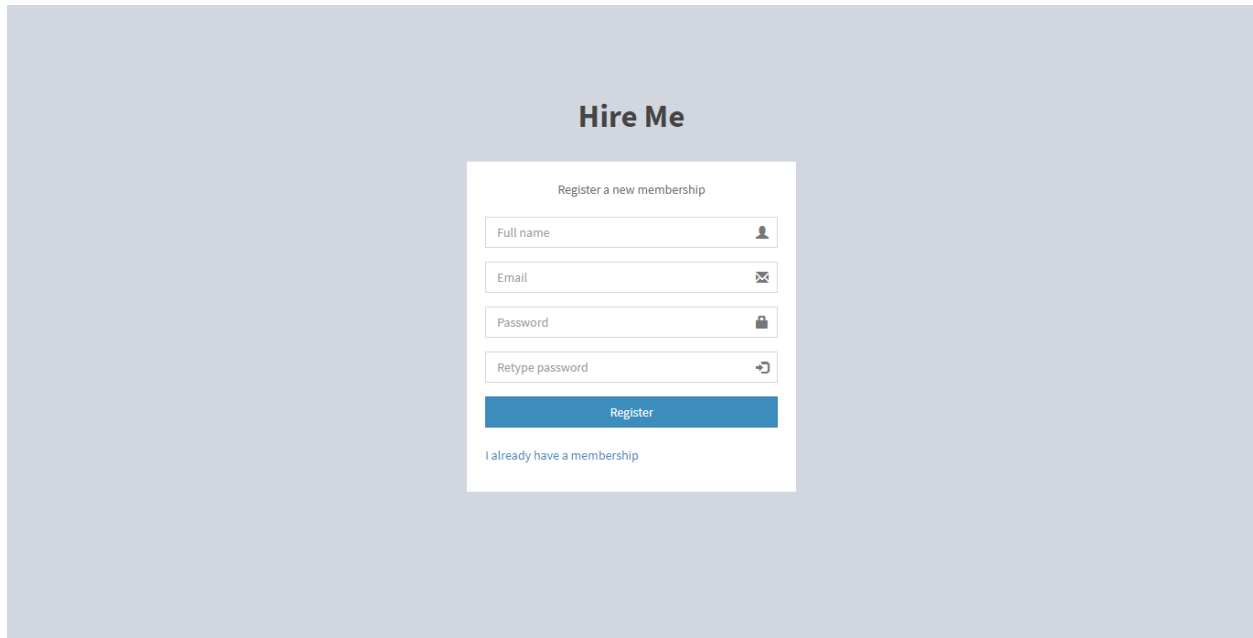
The search bar contains the text '10 am'. The table below shows the results:

Alive Status	Name	Budget	Category	Description	Distance	Key	Location	Sub Category	Time
	Shourov Alam	50	Vehicals	Need a car	0	-LsWsP5BRpZs3UCGIX9Y	23.9078972,90.4028536	Car	10
	Shourov Alam	400	Vehicals	My car is in repair.I need a car to travel from uttara to Dhanmondi.	0	-Lu1ydZFadhtK6stQ4kR	23.87018,90.387882	Car	10 am.
	Shourov Alam	400	Vehicals	My car is in repair.I need a car to travel from uttara to Dhanmondi.	0	-Lu2-JwDQJ5u4t9kNCBr	23.87018,90.387882	Car	10 am.

At the bottom of the second screenshot, it says 'Showing 1 to 3 of 3 entries (filtered from 4 total entries)' and includes 'Previous', '1', and 'Next' navigation buttons.

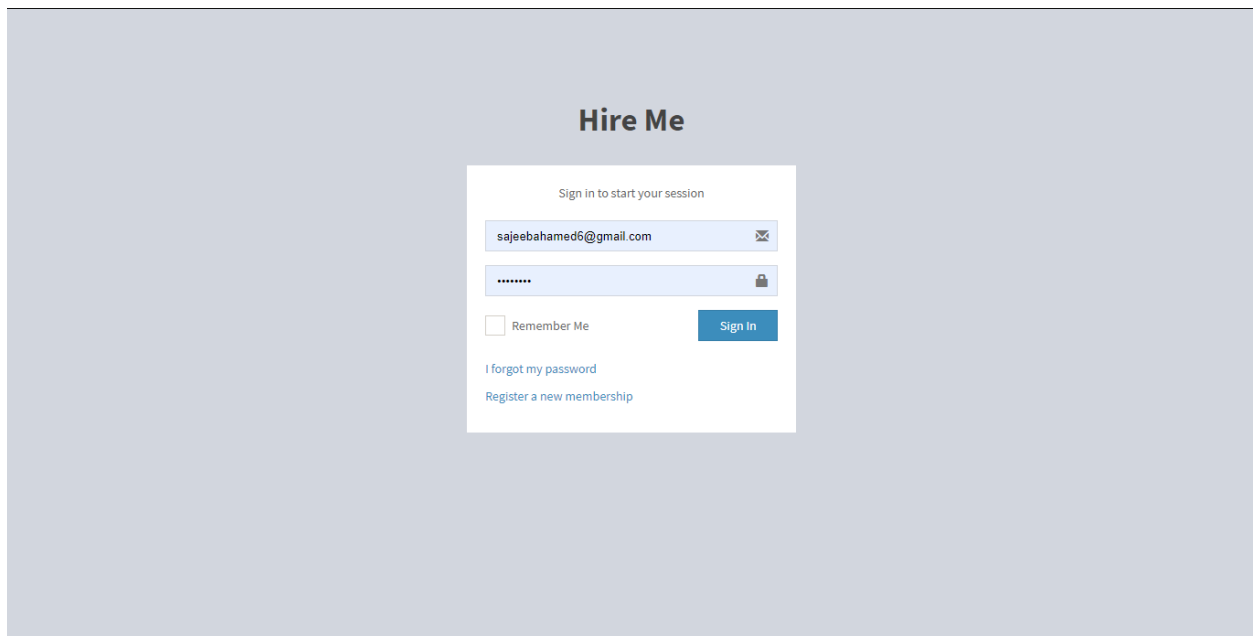
Figure: 4.10.18: Real Time Search

For admin panel we strictly followed some rules. Like they must have verified email address what we officially provided. And then they must be completed their signup and sign in process.



The screenshot shows a registration form titled "Hire Me" with the subtitle "Register a new membership". The form contains four input fields: "Full name" with a person icon, "Email" with an envelope icon, "Password" with a lock icon, and "Retype password" with a refresh icon. Below the fields is a blue "Register" button and a link "I already have a membership".

Figure 4.10.19: Admin Registration



The screenshot shows a sign-in form titled "Hire Me" with the subtitle "Sign in to start your session". The form contains two input fields: "Email" with the value "sajeebahamed6@gmail.com" and an envelope icon, and "Password" with masked characters "*****" and a lock icon. Below the fields is a "Remember Me" checkbox and a blue "Sign In" button. At the bottom, there are links for "I forgot my password" and "Register a new membership".

Figure 4.10.20: Admin Sign In

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

Implementation and testing is very important for an android Application. Another important part of an android application is developing, but testing is more important than developing. After developing an android application testing and implementation is needed. After completing implementation and testing part we can published our application. When a project is accomplished, it must have an intention of future implementation. In this sector there must be an option to implement HIRE ME in future. At present it's all about narrow features. Nevertheless many feature will be included in future. On the other hand implementation never mean that all time include new feature. In future we will fix some weak side and then promote the quality of android application. So in future there are some option to implement the feature. On the other hand if any user have any complain about the feature or any unusable or some feature is less important they may dismiss in future. At last user really found a proper and better android application for android device.

5.2 How it works

After completing the implementation the testing period will start. During the period of developing many system need to be tested. For HIRE ME when we came to develop any feature I have tested the feature in the initial phase. Usually I have tested till the amusement output didn't come. Although it wasted much time from me but since doing this accurately today I have develop the system. In generally testing and implementation is vice versa. If any case the testing period is failed then further implementation is started. Many times when I didn't get my expected output. Then I start thinking and researched with this issue, I start gathered information about this issue, and then try to solve this from own. But if any time I fail to resolve any problem I went to my eldest, or Google or any types of android group. When the testing rate is 100% then it would accept.

5.3 Goals of Hire Me

The main goal of HIRE ME is to helping the people by getting service (example: bazarman, tuition, rent ambulance car or bike, hire nearby experts) with the help of any expert. Now a days we know that many people in our country may have great experience but for many reason they may unemployed. So our main goal to help them to get a per time job and earn money for reducing their unemployment problem. Many people in our country who want to done her/his work by the helping of third party. They can also find third party to complete his work. On the other hand some people in our country don't have enough time to done their work that's why their need a service provider to done their work. Student can also find tuition by the helping of our application. Motor Bike, ambulance or any kinds of car owner can rent their vehicles for short time or long time. So after completing my project I am sure that it can fulfil my goals.

5.4 Testing

Testing is very important for an android application. Before publishing an android application testing is needed. Frist we must have checked all the feature is working or not, all data are has been checked so that user may not deliver any unnecessary data. And then checked the expected output has come or not. If the expected output has come than ok, but otherwise we need to detect bug and start testing again. Verification and validation is another two very important in testing an android project. In our application verification and validation this two feature has been very much concentrated.

5.5 Automate user interface test

User interface confirm that our android application meets its functional requirement and attain a high standard of quality. After that user use our application very friendly. We know that around the world user use a lot of android device that has different version. So we need to keep in mind that our building application should perform same in all the device. On the other hand there many android tab user around the world so our application also support this device. And the big screen this application support also. Before building our application we keep in mind about it and tested properly. In our application we use latest version of android so that user can use it friendly.

5.6 Test UI Performance

UI is very important than any other component of android application. Backend service is less important than User interface. Firstly user see the user interface carefully if user he/she think that this applications user interface is very good than he/she will go to the backend of this application. That's why In **HIRE ME** we use all the update version of the technology to enhance user interface performance. We hope that it will be an excellent application.

5.7 Security Testing

At present security plays a very important role for an android application. So security is most important for an application. Without higher security user never want to use an android application. So before building an android application we should keep in mind about the security issue. If security is loss than everything is lose. In **Hire Me** we used firebase database. All the data are stored in firebase. Firebase database is a product of Google. If data is lost or stolen than firebase is responsible for this event. At present we all are know that firebase is more secure than others. So no need for extra security for our application. So keep in mind and always checked that if no one can steel data or access data without permission.

5.8 System Testing

A fully integrated system needs a system testing. This will figure out the lacking of a system. The app may not install in every device and it may found some bug. So the testing approach can solve all those issues. The system testing of our project gone through 3 steps .UI testing, Integration testing and the Unit testing.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Hire me is such kind of app which actually makes our daily life's problem easier by finding the experts. This app gives a freelancing opportunities to all kinds of skilled people. It connects the user and the skilled people. We are in a middle income county. So people can use this app for their income scope. This app includes different skills category. The person having respective skills can make himself active in a particular area. Then the user can easily find him. In this way the communication system works. This app makes a particular problem solved and also makes an income scope for the user. Here user can post his problem and also can find the active experts list. This app is easy to use. The UI of this app is designed such a way so that the user can easily control the app.

6.2 Future Implementation or Limitation

The app is now in the primary stage. There need a lot of implementation to make it a business standard app. Some future plane is given below:

- I. A payment gateway will be added like Hire Bkash and Rocket
- II. A Website that will work same like our app
- III. The admin will be improved
- IV. User's 5% revenue will be charged for our business

REFERENCE

[1] Business idea from Uber, Shebaxyz, Patao

<< <https://www.cleveroad.com/blog/how-to-develop-your-own-uber-like-business> >>, last accessed on 20-10-2019 at 1.00 am

[2] Unemployment Problem analysis

<<https://www.researchgate.net/publication/272241966_Determinants_of_Unemployment_in_Bangladesh_A_Case_Study>>, last accessed on 23-10-2019 at 11.00 pm

[3] System Development life cycle

<<<https://www.smartsheet.com/system-development-life-cycle-guide>>>, last accessed on 25-10-2019 at 11.00 pm

[4] Phases of Software Development Life Cycle

<<<https://www.am7s.com/7-stages-of-system-development-life-cycle/>>>, last accessed on 1-11-2019 at 2.00 am

[5] Requirement specification

<<<https://www.softwaretestingclass.com/software-requirement-specification-srs/>>>, last accessed on 3-11-2019 at 9.00 pm

[6] Frontend Technologies

<<<https://www.w3schools.com/whatis/>>>, last accessed on 5-11-2019 at 8.00 pm

[7] Backend Admin Technologies (Php/Laravel)

<<<https://laravel.com/docs/6.x>>>, last accessed on 15-11-2019 at 2.00 am

[8] Local host MySQL Database

<<<https://dev.mysql.com/doc/mysql-getting-started/en/>>>, last accessed on 15-11-2019 at 2.00 am

[9] Firebase Real-time Database

<<<https://firebase.google.com/docs/database/?authuser=6>>>, last accessed on 15-11-2019 at 2.05 am

[10] Design & Prototyping

<<<https://creately.com/>>>, last accessed on 17-11-2019 at 8.00 pm

[11] How to start android app

<<<https://www.thedroidsonroids.com/blog/how-to-start-your-android-development-journey-5-basic-steps>>>, last accessed on 17-11-2019 at 8.05 pm

[12] AVD Manager

<<<https://developer.android.com/studio/run/managing-avds>>>, last accessed on 17-11-2019 at 10.00 pm

[13] Android SDK

<https://developer.android.com/studio/command-line/sdkmanager>>>, last accessed on 17-11-2019 at 1.00 am

APPENDIX

APPENDIX A: Activity

Home Page

```
Splash_activity.java x Sign_up_activity.java x Login_activity.java x Post.java x Work.java x Search.java x HomePaze.java x
1 package com.example.moon.microwork.activities;
2
3 import ...
59
60 public class HomePaze extends AppCompatActivity implements Work.OnFragmentInteractionListener
61 , Post.OnFragmentInteractionListener, Search.OnFragmentInteractionListener,
62 Profile_frag.OnFragmentInteractionListener {
63
64     FrameLayout container;
65     BottomNavigationView bottomNavigationView;
66     LocationRequest locationRequest;
67     private static final String TAG = "HomePaze";
68     private Stack<Integer> frag_collection;
69     public static Stack<Fragment> workItem;
70     private static final int REQUEST_LOCATION_CODE = 99;
71     private static final int REQUEST_CHECK_SETTINGS = 1001;
72     GoogleApiClient googleApiClient;
73
74
75     @Override
76     protected void onCreate(Bundle savedInstanceState) {
77         super.onCreate(savedInstanceState);
78         setContentView(R.layout.activity_home_paze);
79         container = findViewById(R.id.container_layout);
80         bottomNavigationView = findViewById(R.id.nav_view);
81         frag_collection = new Stack<>();
82         workItem = new Stack<>();
83         // if(googleApiClient==null)
HomePaze > onBackPressed() > new OnClickListener > onClick()
```

Login activity

```
Splash_activity.java x Sign_up_activity.java x Login_activity.java x Post.java x Work.java x Search.java x HomePaze.java x
1 package com.example.moon.microwork.activities;
2
3 import ...
31
32
33 public class Login_activity extends AppCompatActivity {
34
35     TextView tvSignUp;
36     TextView tvForgetPass;
37     AppCompatActivity etEmailIn, etPassIn;
38     Button btnSignIn;
39     ProgressDialog progressDialog;
40     private static final String TAG = "Login_activity";
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_login_activity);
46
47         init_views();
48         init_variables();
49         init_functions();
50         init_listeners();
51
52     }
53
54     private void init_views() {
55         tvSignIn = findViewById(R.id.tv_signin);
Login_activity > init_listeners()
```

My Posts

```
Splash_activity.java × Sign_up_activity.java × Login_activity.java × MyPosts.java × Post.java × Work.java × Search.java ×
1 package com.example.moon.microwork.activities;
2
3 import ...
46
47 public class MyPosts extends AppCompatActivity {
48
49     Button btnOpenCatagory;
50     // RecyclerView recyclerView_all_posts;
51     private static final String TAG = "Search";
52     private ArrayList<String> header;
53     private HashMap<String, List<String>> child;
54     private ExpandableListView expandableListView;
55     private MutableLiveData<Location> mutableLiveData;
56     private TextView textView_show_direction;
57     private static Location location;
58     private ProgressDialog progressDialog;
59     private ArrayList<Post_models> post_modelsArrayList;
60     private ListView listViewPosts;
61     public static String completeAddressString;
62     public static String catSubCat;
63     public static MutableLiveData<Post_models> post_modelsMutableLiveData;
64
65
66     @Override
67     protected void onCreate(Bundle savedInstanceState) {
68         super.onCreate(savedInstanceState);
69         setContentView(R.layout.activity_my_posts);
70         textView_show_direction = findViewById(R.id.textViewDirection_mn);
71
72         MyPosts > loadData() > new ValueEventListener > onDataChange()
```

Show Post

```
Splash_activity.java × Sign_up_activity.java × Login_activity.java × MyPosts.java × ShowPost.java × Post.java × Work.java ×
1 package com.example.moon.microwork.activities;
2
3 import ...
54
55 public class ShowPost extends AppCompatActivity {
56
57     private ImageView imageView_product;
58     private EditText etPrice;
59     private EditText etdes;
60     private LottieAnimationView lottieAnimationView_sw;
61     private FloatingActionButton fab_sw, fabledsw;
62     private Button btn_update;
63     private static Post_models post;
64
65     private static final String TAG = "ShowPost";
66
67
68     private File file;
69     private File compressor;
70     private Uri final_path = null;
71     private boolean FLAG_IMAGE_SET = false;
72     private StorageReference ref ;
73     private String imageUrl = "";
74     private ProgressDialog progressDialog;
75     private static String imageUrl = "";
76     DatabaseReference child_ref;
77     String[] str_ary;
78
79     ShowPost > onCreate()
```

APPENDIX B: Adapter

CustomExpandableAdapter

```
Sign_up_activity.java x Login_activity.java x MyPosts.java x ShowPost.java x CustomExpandableAdapter.java x Post.java x Work.java x
1 package com.example.moon.microwork.adapters;
2
3 import android.content.Context;
4 import android.util.Log;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.BaseExpandableListAdapter;
9 import android.widget.CompoundButton;
10 import android.widget.Switch;
11 import android.widget.TextView;
12 import com.example.moon.microwork.R;
13 import com.example.moon.microwork.constants.FirebaseConstants;
14 import com.example.moon.microwork.models.CustomChildObject;
15 import com.example.moon.microwork.utils.FirebaseHelper;
16
17 import java.util.ArrayList;
18 import java.util.HashMap;
19 import java.util.List;
20 public class CustomExpandableAdapter extends BaseExpandableListAdapter {
21     ArrayList<String> group_header;
22     HashMap<String, List<CustomChildObject>> group_child;
23     Context context;
24     private static final String TAG = "CustomExpandableAdapter";
25
26     public CustomExpandableAdapter(ArrayList<String> group_header, HashMap<String, List<CustomChildObject>> group_child, Context context) {
27         this.group_header = group_header;
28         this.group_child = group_child;
29     }
30 }
CustomExpandableAdapter > context
```

ExtendAdapterForWorkFrag

```
Login_activity.java x MyPosts.java x ShowPost.java x CustomExpandableAdapter.java x ExtendAdapterForWorkFrag.java x Post.java x Work.java x
1 package com.example.moon.microwork.adapters;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 public class ExtendAdapterForWorkFrag extends BaseExpandableListAdapter {
19
20     private ArrayList<String> group_header;
21     private HashMap<String, List<String>> group_child;
22     private Context context;
23     private String from;
24
25
26
27     public ExtendAdapterForWorkFrag(ArrayList<String> group_header, HashMap<String, List<String>> group_child, Context context , String from) {
28         this.group_header = group_header;
29         this.group_child = group_child;
30         this.context = context;
31         this.from = from;
32     }
33
34     @Override
35     public int getGroupCount() { return group_header.size(); }
36
37
38     @Override
39     public int getChildrenCount(int groupPosition) {
40         return group_child.get(group_header.get(groupPosition)).size();
41     }
42
43     @Override
44 }
ExtendAdapterForWorkFrag
```

ListAdapterForPost

```
1 package com.example.moon.microwork.adapters;
2
3 import ...
4
19
20 public class ListAdapterForPost extends BaseAdapter {
21     private ArrayList<Post_models> arrayList;
22     private Context context;
23     private LatLng currentLatLng;
24     private LayoutInflater inflater;
25     private ImageView imageView;
26     private TextView tvBudget;
27     private TextView tvService;
28     private TextView tvDescrip;
29     private TextView tvDistance;
30     private TextView tvName;
31
32     private static final String TAG = "ListAdapterForPost";
33     boolean isLocationOn = false;
34
35
36 @
37     public ListAdapterForPost(ArrayList<Post_models> arrayList, Context context) {
38         this.arrayList = arrayList;
39         this.context = context;
40         this.currentLatLng = currentLatLng;
41         inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
42         // isLocationOn = AppConst.isIfLocationOn();
43     }
44 }
```

RecyclerViewAdapter

```
1 package com.example.moon.microwork.adapters;
2
3 import ...
4
21
22 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.MyViewHolder> {
23     private ArrayList<Post_models> arrayList;
24     private Context context;
25     private LatLng currentLatLng;
26
27     private static final String TAG = "RecyclerViewAdapter";
28
29
30     public RecyclerViewAdapter(LatLng currentLatLng, ArrayList<Post_models> arrayList, Context context) {
31         this.arrayList = arrayList;
32         this.context = context;
33         this.currentLatLng = currentLatLng;
34     }
35
36     @NonNull
37     @Override
38     public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
39         View view = LayoutInflater.from(this.context).inflate(R.layout.post_recycler_item, parent, attachToRoot: false);
40         return new MyViewHolder(view);
41     }
42
43     @Override
44     public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
45         Post_models post_models = arrayList.get(position);
46     }
47 }
```

APPENDIX C: Fragment

Post

```
Post.java × CustomExpandableAdapter.java × ExtendAdapterForWorkFrag.java × ListAdapterForPost.java × RecyclerViewAdap
1 package com.example.moon.microwork.fragments;
2
3 import ...
54
55
56 public class Post extends Fragment {
57
58     ArrayList<String> header;
59     HashMap<String,List<String>> child;
60     private ExpandableListView expandableList;
61     private static final String TAG = "Post";
62     private TextView tvCat;
63     private File file;
64     private File compressor;
65     private Uri final_path = null;
66     private boolean FLAG_IMAGE_SET = false;
67     private TextView tvPath;
68     private Button btnPost;
69     private Post_models post_models;
70     private Switch aSwitch;
71     private EditText etDescription,etBudget,etTime;
72     private ProgressDialog progressDialog;
73     private StorageReference ref ;
74     private String imageUri = "";
75
76
```

Search

```
t.java × CustomExpandableAdapter.java × ExtendAdapterForWorkFrag.java × ListAdapterForPost.java × RecyclerViewAdapter.java × Search.java
66 import java.util.Arrays;
67 import java.util.HashMap;
68 import java.util.HashSet;
69 import java.util.List;
70 import java.util.Locale;
71 import java.util.Objects;
72
73
74 /**
75  * A simple {@Link Fragment} subclass.
76  */
77 public class Search extends Fragment {
78
79     Button btnOpenCatagory;
80     Switch btnPinLocation;
81     // RecyclerView recyclerView_all_posts;
82     private static final String TAG = "Search";
83     private ArrayList<String> header;
84     private HashMap<String, List<String>> child;
85     private ExpandableListView expandableListView;
86     private MutableLiveData<Location> mutableLiveData;
87     private TextView textView_show_direction,tvLocation;
88     private static Location location;
89     private ProgressDialog progressDialog;
90     private ArrayList<Post_models> post_modelsArrayList;
91     // private ListView listViewSearch;
92     private RecyclerView recyclerView;
93     public static String completeAddressString;
```

Skills_fragment

```
expandableAdapter.java x ExtendAdapterForWorkFrag.java x ListAdapterForPost.java x RecyclerViewAdapter.java x Search.java x Skills_frag.java >
1 package com.example.moon.microwork.fragments;
2
3
4 import ...
35
36
37
38 /**
39  * A simple (@link Fragment) subclass.
40  */
41 public class Skills_frag extends Fragment {
42
43     skillAdapter sAdap;
44     Button btnOpenExpandableListView;
45     ExpandableListView expandableListViewSkills;
46     private ArrayList<String> header;
47     private HashMap<String, List<CustomChildObject>> child;
48     private static final String TAG = "Skills_frag";
49     int[] childrensArrayList = {R.array.vehicals,R.array.tutions,R.array.domestic,R.array.electronics,R.array.exparts,R.array
50     private static String liveStatus;
51     private List<CustomChildObject> temp;
52     private List<String> strings;
53     MutableLiveData<Boolean> isCompleted;
54     CustomExpandableAdapter customExpandableAdapter;
55
56     public Skills_frag() {
57
```

Work

```
Work.java x CustomExpandableAdapter.java x ExtendAdapterForWorkFrag.java x ListAdapterForPost.java x RecyclerViewAdapter.java x
67 import java.util.HashMap;
68 import java.util.List;
69 import java.util.Locale;
70
71
72 public class Work extends Fragment {
73
74     Button btnOpenCatagory;
75     Switch btnPinLocation;
76     // RecyclerView recyclerView_all_posts;
77     private static final String TAG = "Work";
78     ArrayList<String> header;
79     HashMap<String,List<String>> child;
80     ExpandableListView expandableListView;
81     MutableLiveData<Location> mutableLiveData;
82     TextView textView_show_direction,tvLocation;
83     public static Location location;
84     private ProgressDialog progressDialog;
85     ArrayList<Post_models> post_modelsArrayList;
86     ListView listViewPosts;
87     public static String completeAddressString;
88     public static String catSubCat;
89     public static MutableLiveData<Post_models> post_modelsMutableLiveData;
90
91
92
93 @ public static String getCompleteAddressString() { return completeAddressString; }
94
95
96
Work
```


APPENDIX D: Model

Login

```
Work.java × Login.java × CustomExpandableAdapter.java × ExtendAdapterForWorkFrag.java × ListAdapterForPost.java ×
1 package com.example.moon.microwork.models;
2
3 public class Login {
4     private String email;
5     private String password;
6     private String nid;
7
8     public Login() {
9     }
10
11    public Login(String email, String password, String nid) {
12        this.email = email;
13        this.password = password;
14        this.nid = nid;
15    }
16
17    public String getEmail() { return email; }
20
21    public void setEmail(String email) { this.email = email; }
24
25    public String getPassword() { return password; }
28
29    public void setPassword(String password) { this.password = password; }
32
33    public String getNid() { return nid; }
36
37    public void setNid(String nid) { this.nid = nid; }
40
Login
```

Post_models

```
Login.java × Post_models.java × CustomExpandableAdapter.java × ExtendAdapterForWorkFrag.java × ListAdapterForPost.java ×
1 package com.example.moon.microwork.models;
2
3
4 public class Post_models {
5     private String cat;
6     private String subCat;
7     private String location;
8     private String path;
9     private String des;
10    private String time;
11    private int budget;
12    private boolean isAlive;
13    private String key;
14    private String authId;
15    private float distance;
16    private String name;
17
18    public Post_models() {
19    }
20
21
22    public Post_models(String cat, String subCat, String location, String path, String des, String time, int budget,
23        this.cat = cat;
24        this.subCat = subCat;
25        this.location = location;
26        this.path = path;
27        this.des = des;
28        this.time = time;
```

Profile

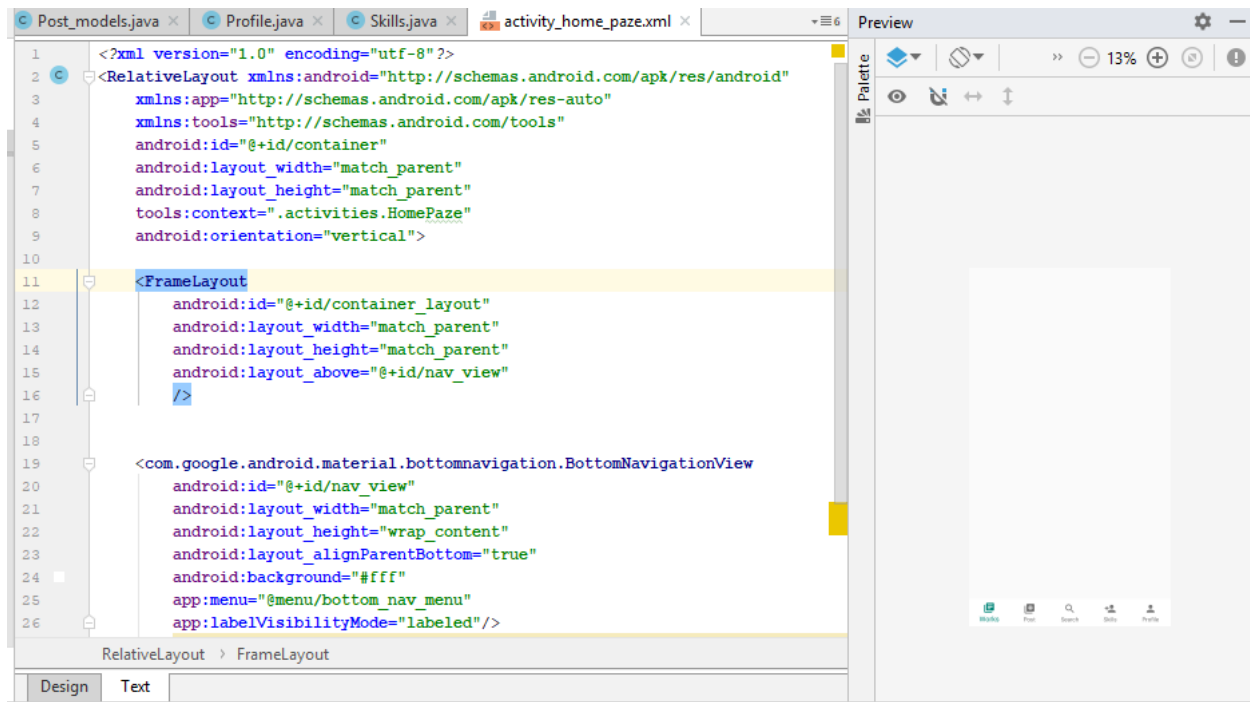
```
Login.java x Post_models.java x Profile.java x ExtendAdapterForWorkFrag.java x ListAdapterForPost.java x RecyclerViewAdapter.java x
1 package com.example.moon.microwork.models;
2
3 public class Profile {
4     private String photo_uid;
5     private String name;
6     private String phone;
7     private String dataOfBirth;
8     private String email;
9     private String password;
10    private String nid;
11    private String gender;
12
13
14    public Profile() {
15    }
16
17    public Profile(String photo_uid, String name, String phone, String dataOfBirth, String email,String password,String nid, S
18        this.photo_uid = photo_uid;
19        this.name = name;
20        this.phone = phone;
21        this.dataOfBirth = dataOfBirth;
22        this.email = email;
23        this.password = password;
24        this.nid = nid;
25        this.gender = gender;
26    }
27
28    public String getPhoto_uid() { return photo_uid; }
```

Skills

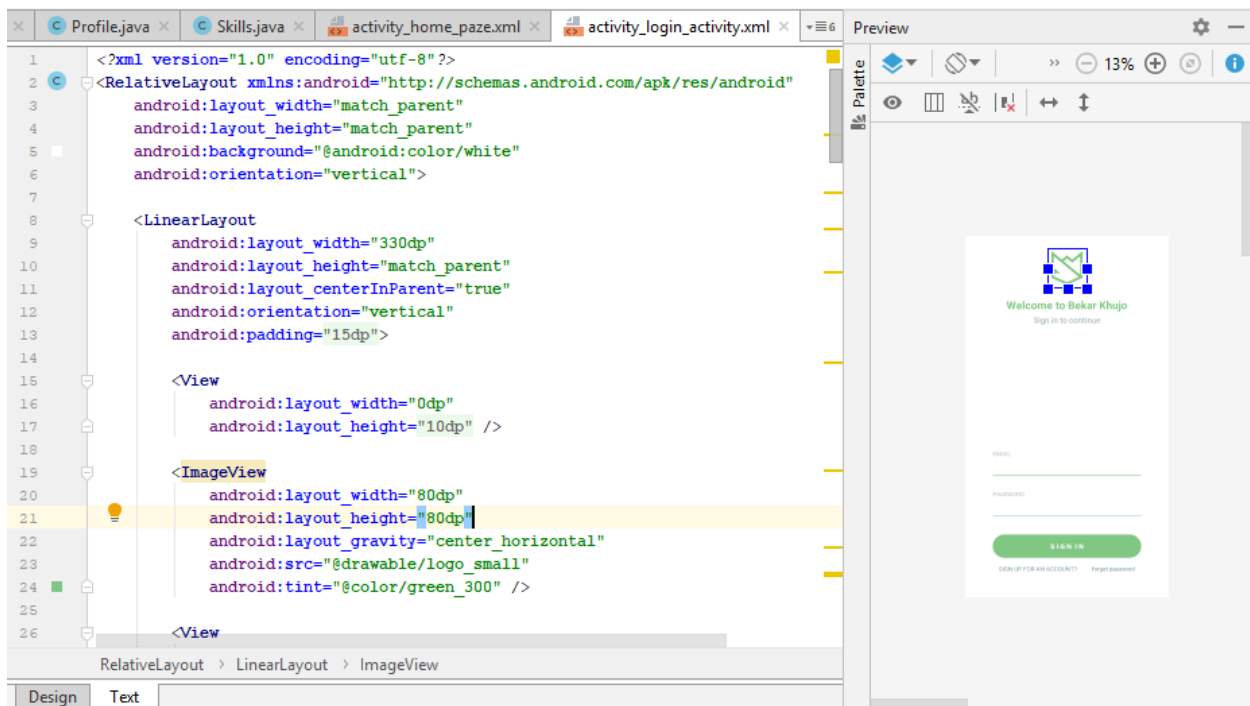
```
Login.java x Post_models.java x Profile.java x Skills.java x ListAdapterForPost.java x RecyclerViewAdapter.java x
1 package com.example.moon.microwork.models;
2
3 import com.google.android.gms.maps.model.LatLng;
4
5 import java.util.Map;
6
7 public class Skills {
8     private boolean isLive;
9     private String Ambulance;
10    private String Bike;
11    private String Car;
12    private String Pickup;
13    private String Truck;
14    private String Taxi;
15    //private String Other_Vehicals;
16
17
18    private String Class_1_5;
19    private String Class_6_8;
20    private String Class_9_10;
21    private String Class_11_12;
22    //private String Other_tution;
23
24    private String Carpenter;
25    private String Cooker;
26    private String Electrician;
27    private String Gasrepairer;
28    private String Locksmith;
```

APPENDIX E: Xml

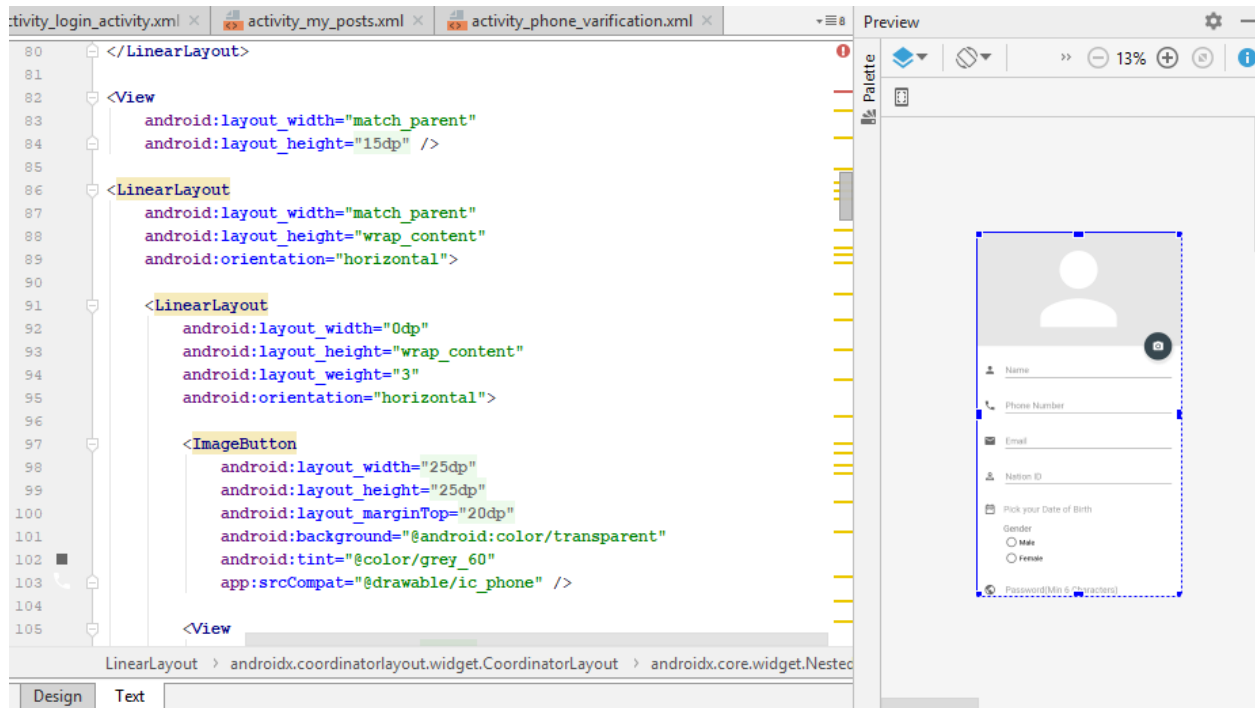
activity_home_paze



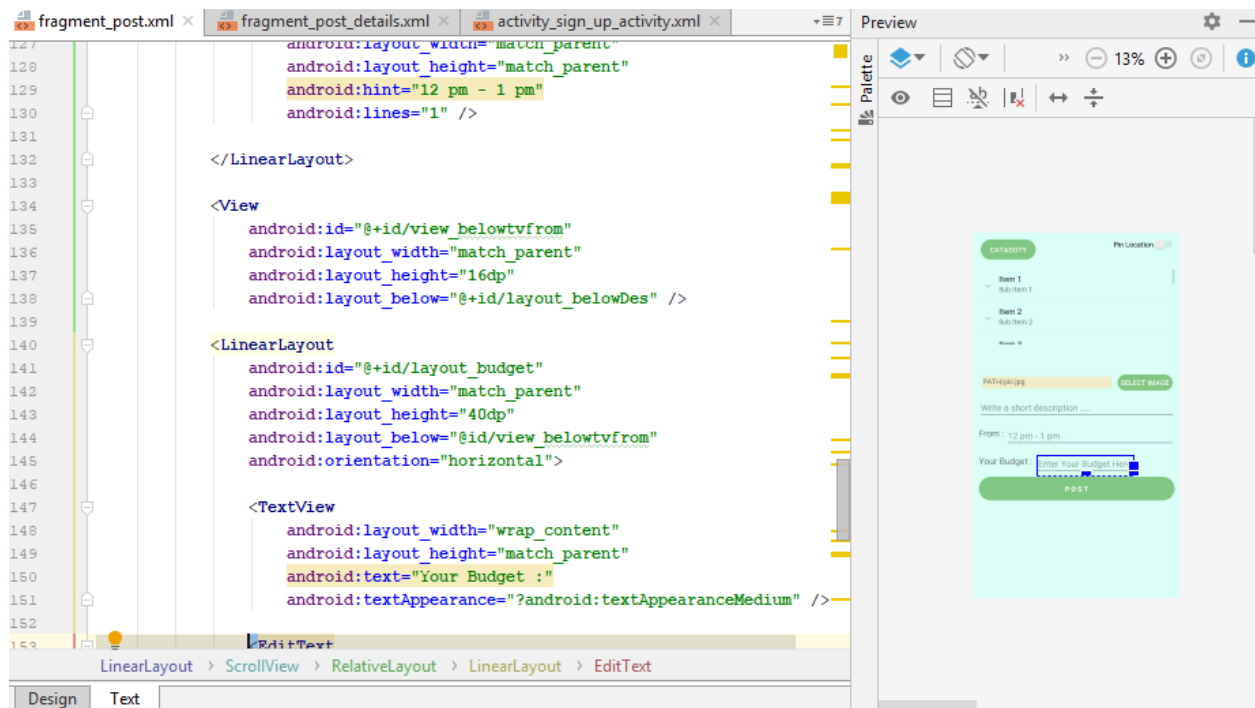
activity_login_activity



activity_sign_up_activity



activity_post



Plagiarism Result

161-15-1010

ORIGINALITY REPORT

13%

SIMILARITY INDEX

12%

INTERNET SOURCES

0%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	lonewolfonline.net Internet Source	2%
3	oer.nios.ac.in Internet Source	1%
4	Submitted to London School of Commerce and IT Student Paper	1%
5	Submitted to South Birmingham College Student Paper	1%
6	Submitted to University of Greenwich Student Paper	1%
7	baadalsg.inflibnet.ac.in Internet Source	1%
8	talonibarret.com Internet Source	1%
9	www.managementparadise.com	

Internet Source

<1%

10

dspace.daffodilvarsity.edu.bd:8080

Internet Source

<1%

11

Submitted to EThames Graduate School

Student Paper

<1%

12

dspace.library.daffodilvarsity.edu.bd:8080

Internet Source

<1%

13

actainformaticamalaysia.com

Internet Source

<1%

14

ethesis.nitrkl.ac.in

Internet Source

<1%

Exclude quotes On

Exclude matches < 10 words

Exclude bibliography On