**Final Project report**

On

**Application to Book an Engineer in your Local Area**

**A PROJECT REPORT**

***Submitted by***

**Priyanshu Sharma (20bcs2402)**

***in partial fulfilment for the award of degree of***

# BACHELOR OF ENGINEERING

(Computer Science & Engineering)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI**

May, 2022

**Final Project report**



## BONAFIDE CERTIFICATE

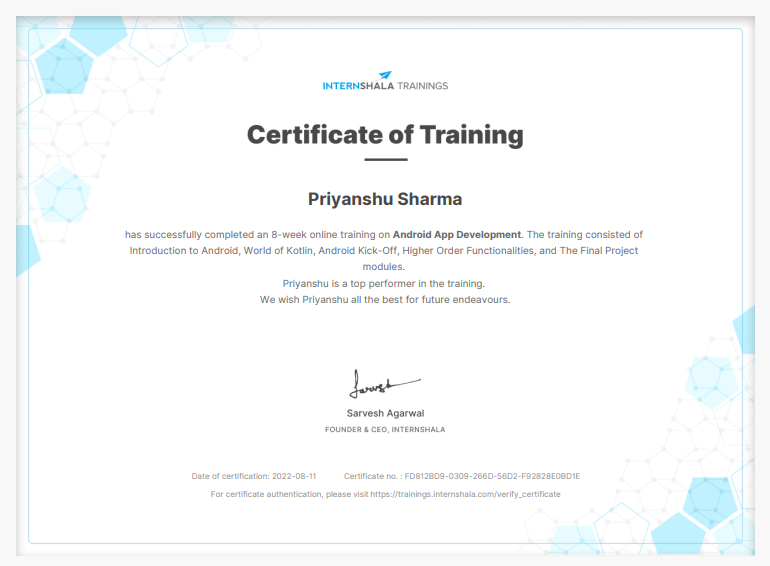
This is to certify that the project entitled **“Application to Book AN ENGINEERS in your Local Area”** has been submitted by Priyanshu Sharma (20BCS2402) for the partial fulfilment of the requirement for the award of degree Bachelor of Engineering in “Computer Science & Engineering” discipline in “Chandigarh University” during the 5

**SIGNATURE SIGNATURE**

**SUPERVISOR HEAD OF THE DEPARTMENT**

**INTERNAL EXAMINAR EXTERNAL EXAMINAR**

**Certification**

****

# ACKNOWLEDGEMENT

I would like to express our deep and sincere gratitude to our project Supervisor Er Deepika Man (E-id: E11809) and Co Supervisor Dr. Tajinder Kaur Man (E-id: E12242) for giving us the opportunity to do the project and providing valuable guidance throughout this research. Their dynamism, vision and exquisite efforts have deeply inspired me. They taught me the methodology to carry out the research and to present the research work as clearly as possible. It was a great privilege for me to study and work under their guidance.

I would also like to express my gratitude to for their continuous help and monitoring during the project work.

I am also thankful to our whole class most of all parents who have inspired us to face all the challenges and will all the hurdled in life.

In the last I would like to thank the management of Chandigarh University for providing us such an opportunity to learn these experiences.

**List of Figures Page No.**

Figure 1. User Case Diagram- Actor 16

Figure 2. User Case Diagram- System 16

Figure 3. Book Recommendation System 17

Figure 4. Flowchart 18

**List of Tables**

Table 1. Software Requirements 15

Table 2. Hardware Requirements 15

# Contents

**Topic Page No.**

*Certificate 2*

*Acknowledgement 3*

*List of Figures 4*

*List of Tables 4*

*Abstract 6*

*Definitions, Acronyms and Abbreviations 7*

**CHAPTER 1 INTRODUCTION 9-14**

**1.1 Introduction 8**

**1.2 Benefits 8**

**1.3 Objective 9**

**1.4 System Requirements 10**

**1.4.1 Software Requirements 10**

**1.4.2 Hardware Requirement 11**

**CHAPTER 2 LITERATURE SURVEY 15-16**

**2.1 Literature Review 14**

**2.2 Objective 15**

**CHAPTER 3 DESIGN FLOW/PROCESS 17-22**

**3.1 User Case Diagram 15**

**3.2 Book Recommendation System 16**

**3.3 SRS diagram 18**

**3.4 E-R diagram 19**

**3.5 Overall Desription 19-20**

**CHAPTER 4 RESULT AND VALIDATION 23-31**

**4.1 Result 21**

**4.2 Java Classes 22**

**4.3 Java MainActivity Code 23**

**4.4 XML MainActivity Code 27**

**4.5 SessionManager Java Class 28**

**4.6 Snapshots 29**

**4.7 Code Link 30**

**CHAPTER 5 CONCLUTION AND FUTURE WORK 31-33**

**5.1 Conclusion 33**

**5.2 Future Scope 33**

**5.3 Possible Future Scope 33-34**

**REFERENCES 35**

# ABSTRACT

The purpose of this project is that each day, houseowners encounter numbers of problems associated with their electrical appliances, wooden structures and water. These may seem minor issues but they lead to major problems it left untreated. Sometimes, people ignore them because they need to spend time to look for experienced professionals. It does not mean they don’t want to get rid of these household issues.

On- demand we made this project for carpenter, electrical and plumbing service and provide with quick help to resolve household issues associated with electrical appliances, wooden items, and water. This project will help the people to search and book experienced professionals at ease and hire them quickly and resolve their problem very easily.

On the other hand, plumbers, electricians, and carpenters can use this applition to get hired easily and grow their business. This Application will allow people to book professionals. In addition, this Application can be accessed on any kind operating system.

# DEFINITIONS, ACRONYMS AND ABBREVIATIONS

**DFD (DATA FLOW DIAGRAM)-** A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

## ER DIAGRAM (ENTITY RELATIONSHIP DIAGRAM)- An entity–relationship model

describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

## UML DIAGRAM (UNIFIED MODELLING LANGUAGE)- A UML diagram is a diagram

based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

# CHAPTER – 1

# 1.1 INTRODUCTION

When someone need aid with small but major household tasks, the trouble arises when service skilled persons are unavailable or the trusted providers are impossible to find, who delivers consistently flawless service on instance. Our online system for household services provides the most expedient and annoys free wat to get your domestic work done. We aim to help in providing optimal solutions to all your household troubles with more efficiency, ease and majorly, a delicate touch. A single click system describes booking highly skilled in-house professionals and gets your service on time. Customers’ overall willingness to pay is significantly and positively correlated with the expectation that fee-based services would be better, and with the belief that “pay for what you get” is the right thing to do. Keeping that I sense our proposed system is basically a marketplace for household services and it is the platform where the rate was standardized and there is not necessitate haggling over prices. Several aspects like electrical work, carpentry work and plumbing services are involved in a system to provide happy and healthy home atmosphere in order to satisfy consumers.

## 1.2 Benefits of “Application to Book an Engineers in your local area”

* This Application meet the requirements of electrical, plumbing and carpenter services in your local area.
* This Application is fully functional and flexible.
* It is very easy to use.
* It saves a lot of time and money.
* This portal opens 24/7.
* It increases the efficiency of booking of engineer services to the customers.
* To reduce burden in finding in-house solutions for the services, the proposed system provides several services by providing service specialists at your doorstep in one click.
* Customers who want to avail our services are invited to register for a free account in our portal with few simple steps.
* It provides custom features development and support with the software.

With the help of this Application carpenter, electrical and plumbing service and provide with quick help to resolve household issues associated with electrical appliances, wooden items, and water. This application will help the people to search and book experienced professionals at ease and hire them quickly and resolve their problem very easily.

On the other hand, plumbers, electricians, and carpenters can use this application to get hired easily and grow their business. This application will allow people to book professionals. In addition, this application can be accessed on any kind operating system.

# 1.3 Objectives

The primary objective of the online system for household services is about delivering the home services is about delivering the home services at the door step just by one click. This paper discusses about main theme of the online home services, numerous services provided and how the ordering and delivery services can be used by any authorized user intending to seek for household services through an ingenious web-based system or a mobile application. To provide an authenticated and authorized login module for the users such as service seekers, service providers and the admin, by providing appropriate credentials at the time of registration. To develop a web based online system for opting household services and to develop an identical mobile application for opting the services.

To design a interactive User Interface for seeking services on the go. To acknowledge the conformation of services opted by the users.

**1.4 SYSTEM REQIREMENTS**

**1.4.1 SOFTWARE REQUIREMENTS**

## • Android Studio

Android Studio is the official IDE for android app development. Works based on IntelliJ IDEA, You can download the latest version of the android studio from Android Studio 2.2 Download, If you are new to installing Android Studio on windows, you will find a file, named android-studio-bundle-143.3101438 -windows.exe.So already download and use the windows machine according to the android studio wizard guidance.

If you are installing Android Studio on Mac or Linux, you can download the latest version from Android Studio Mac Download, or Android Studio Linux Download, check the instructions provided and the downloaded file for Mac OS and Linux. This tutorial will consider setting up your location on a Windows operating system with Windows 8.1 operating system.

Android Studio provides an integrated environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to segment your project into functional units that you can independently create, test, and customize.

## XML

XML stands for Extensible Language Language. XML is a HTML markup language used to interpret data. Based on Standard Markup Language (SMGL). Basically, XML tags are not defined in XML.

We need to use and define tags in XML. XML tags describe data and are used to store and organize data. It is easily scalable and easy to upgrade. For Android, XML is used to use UI-related data, and is a simple language that does not make the structure difficult. XML contains only tags, while in use they only need to be requested. Basically on Android XML is used to use UI-related data.

So understanding the core part of the UI interaction with regard to XML is important. The user interface of the Android app is built as a series of basic layouts, widgets. Properties are ViewGroup objects or containers that control how a child’s view should be displayed on the screen. Widgets here are visual objects, such as buttons and text boxes. Consider the following simple example of the activity\_main.xml file.

## Google Fire Base

Firebase is a platform built by Google for creating mobile and web applications. It was originally a private company founded in 2011. In 2014, Google acquired a platform and is now their outstanding contribution to app development. Firebase originated from Envolve, a predecessor founded by James Tamplin and Andrew Lee in 2011. Promote developers provided with an API that allows integration of online chat functionality into their application. After releasing the chat service, Tamplin and Lee discovered that it was used to transfer app data that were not chat messages. Developers were using Envolve to synchronize application-like game data in real-time for all users. Tamplin and Lee decided to separate the chat system and the real-time structure that empowers us. They founded Firebase as a separate company in 2011 and were launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that syncs app data to all iOS, Android, and web devices, and stores it in the Firebase cloud. The product assists software developers in building real-time, integrated applications.

## Java For Android Development:

Java is a programming language and platform computer first released by Sun Microsystems in 1995. It has evolved from the beginning to empower a large part of today's digital world, by providing a reliable platform on which many resources and applications are built. New, innovative products and digital services designed for the future continue to rely on Java, too.

Although most modern Java applications integrate Java operating time and operating system together, there are still many applications and some that will not work without installing desktop Java. Java.com, this website, is intended for users who may still need Java for their desktop applications - especially applications that run Java 8. Developers and users who would like to learn how to configure Java. Android Gradle plugin 3.0.0 and later supports all Java 7 language features as well as a subset of Java 8 language features that vary by platform version.

If you are building your app using Android Gradle plugin 4.0.0 and above, you can use Java 8 Language APIs without the need for a low-level API for your application. This page explains the features of Java 8 language that you can use, how to customize your project for use, and any known problems you may encounter. See also the following video for an overview.

Table 1

Software Requirements

|  |  |  |
| --- | --- | --- |
| **SERVER SIDE** | Operator System | Android |
| Application Software | PHP My Admin |
| Software Tools | Android Studio , Java , Xml |
| Database | Google Fire Base |
| **CLIENT SIDE** | Mobile | Android base |
| Application | Servise on time |

Table. 2

Hardware Requirements

|  |  |  |
| --- | --- | --- |
| **SERVER SIDE**  **Minimum Hardware**  **Requirement** | Processor | Core i3 |
| Main Memory | 4 GB Ram |
| HDD | 80 GB |
| **Clint Side**  **Minimum Hardware**  **Requirement** | Processor | Pentium IV |
| Main Memory | 1 GB Ram |
| HDD | 4 GB |

**CHAPTER 2**

## 2.1 Literature survey

In recent years, advances in mobile technology have brought about significant changes in people's daily lives. Smartphones / cell phones abound in every aspect of human life. This has led to an urgent need to develop software that works on mobile devices. Engineers must comply with this basic requirement and deliver a high-quality application within and within the budget. In this case, the limitations of the development and testing of applications play an important role. In this paper, Systematic Review (SLR) is done to highlight the software development and evaluation process. The goal of the current book survey is to identify and compare existing measurement methods for standard software (desktop / laptop) and software / mobile application. Features that make mobile software / application different from normal software are identified in this book survey. In addition, the software development process is geared towards aging, so this study also identifies and compares the measurement methods used in the rapid software development of mobile applications. A review of the literature review proposes to fill the research gap to introduce official models of mobile app evaluation by considering specific aspects of mobile software. Our online in-house service system provides the most convenient and annoying way to get free what your homework done. We aim to help provide the right solutions for your entire family problems with extra efficiency, charge and especially, soft touch. One-click system explains book with highly skilled professionals in-house and get your service on time. Total for customers Payment willingness is highly correlated with payment-based expectations Services can be better, and with the belief that "pay what you get" is the right thing to do. As I feel our proposed system is actually a home services market and it Is a platform where the standard is set and there is no need to negotiate prices.

Several factors such as electrical work, carpentry and plumbing services are involved to provide a happy and clean home environment to the satisfaction of consumers.

Our online system for household services provides the most expedient and annoys free wat to get your domestic work done. We aim to help in providing optimal solutions to all your household troubles with more efficiency, ease and majorly, a delicate touch. A single click system describes booking highly skilled in-house professionals and gets your service on time. Customers’ overall willingness to pay is significantly and positively correlated with the expectation that fee-based services would be better, and with the belief that “pay for what you get” is the right thing to do. Keeping that I sense our proposed system is basically a marketplace for household services and it is the platform where the rate was standardized and there is not necessitate haggling over prices. Several aspects like electrical work, carpentry work and plumbing services are involved in a system to provide happy and healthy home atmosphere in order to satisfy consumers.

**2.2 Objective**

The primary objective of the online system for household services is about delivering the home services is about delivering the home services at the door step just by one click. This paper discusses about main theme of the online home services, numerous services provided and how the ordering and delivery services can be used by any authorized user intending to seek for household services through an ingenious web-based system or a mobile application. To provide an authenticated and authorized login module for the users such as service seekers, service providers and the admin, by providing appropriate credentials at the time of registration. To develop a web based online system for opting household services and to develop an identical mobile application for opting the services.

To design a interactive User Interface for seeking services on the go. To acknowledge the conformation of services opted by the users.

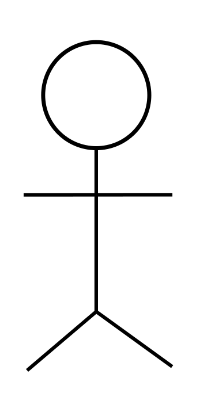
# Chapter 3:

# Design flow/Process

## 3.1 Use Case Diagram

A use case diagram at its simplest is representation of use’s interaction with the system that shows the relationship between the user and the different use case in which the use is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

**Actor** An actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an

instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may Figure 1. play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

**Association** An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

**System** If a subject (or system boundary) is displayed, the use case ellipse is visually located

inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained use cases, but merely that the use

Figure 2. case applies to that classified.

# 3.2 BOOK RECOMMENDATION SYSTEM

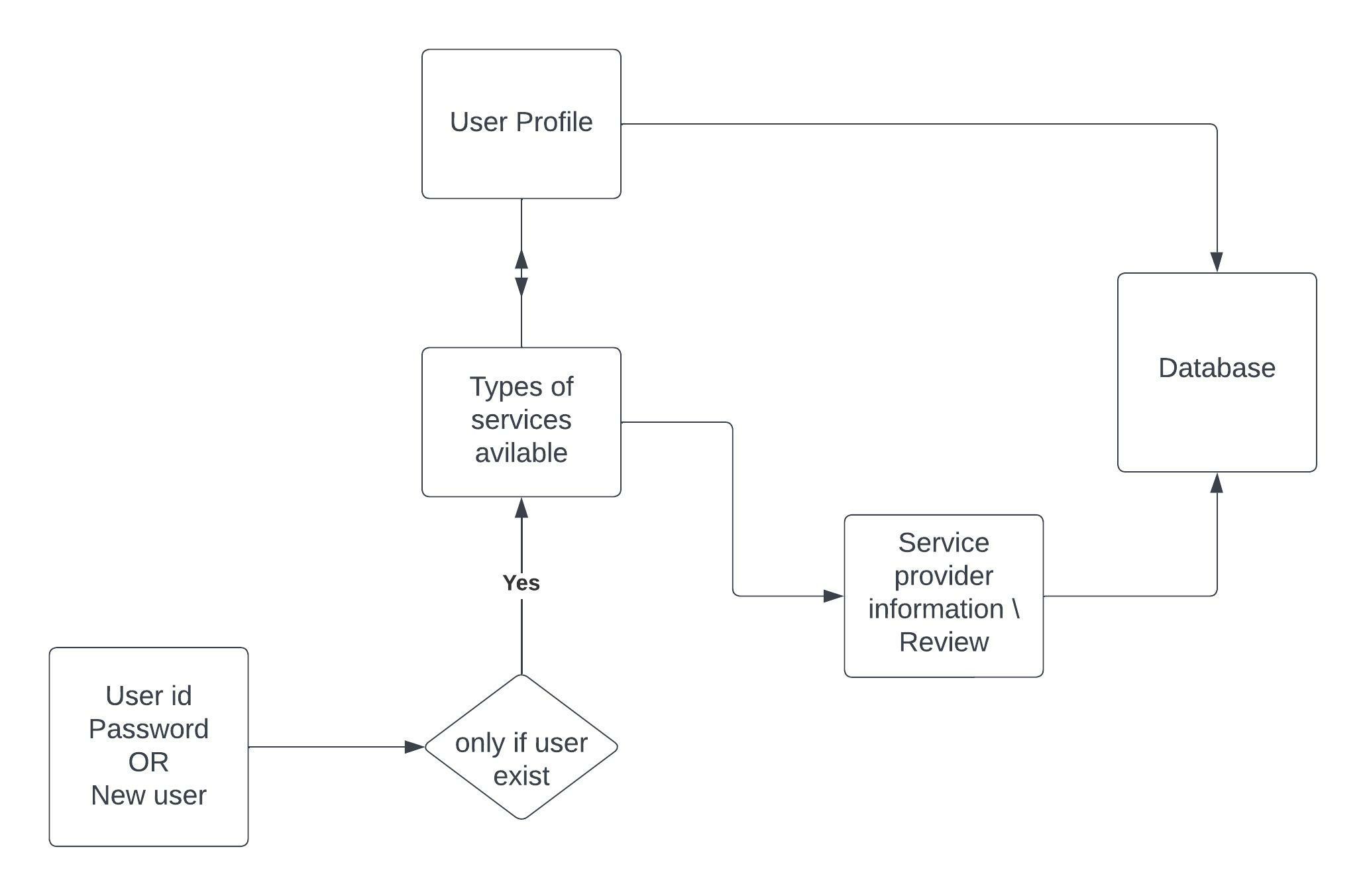


Figure 3

ACTOR

DATABASE

REGISTER/SIGNUP

LOGIN

VISIT HOME PAGE

SEEL SERVICES

SEE ENGINEERS

MAK BOOKIGS

LOGOUT

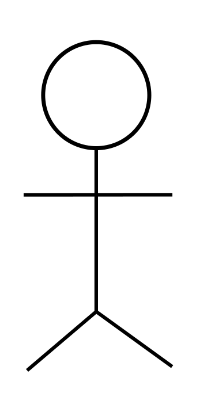
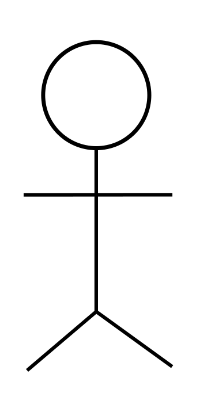


Figure 4

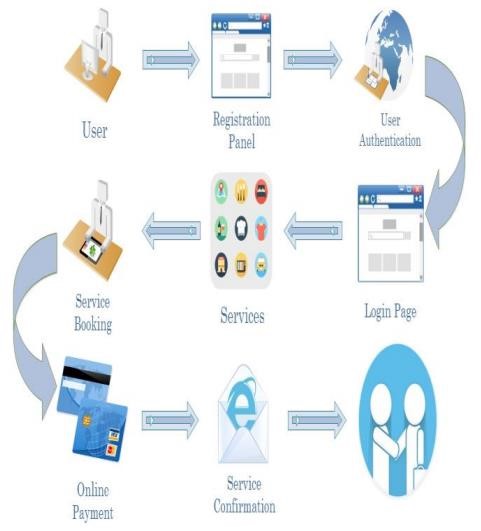


Figure 5

# 3.3 SRS (SOFTWARE ARE REQUIREMENT SPECIFICATION)

A software requirements specification (SRS document) describes how a software system should be developed. Simply put, an SRS provides everyone involved with a roadmap for that project. It offers high-grade definitions for the functional and non-functional specifications of the software, and can also include use cases that illustrate how a user would interact with the system upon completion. An SRS should have enough information for developers to complete the software described. It not only lays out the description of the software under development but also the purpose it will serve what the software is supported to do and how it should perform.

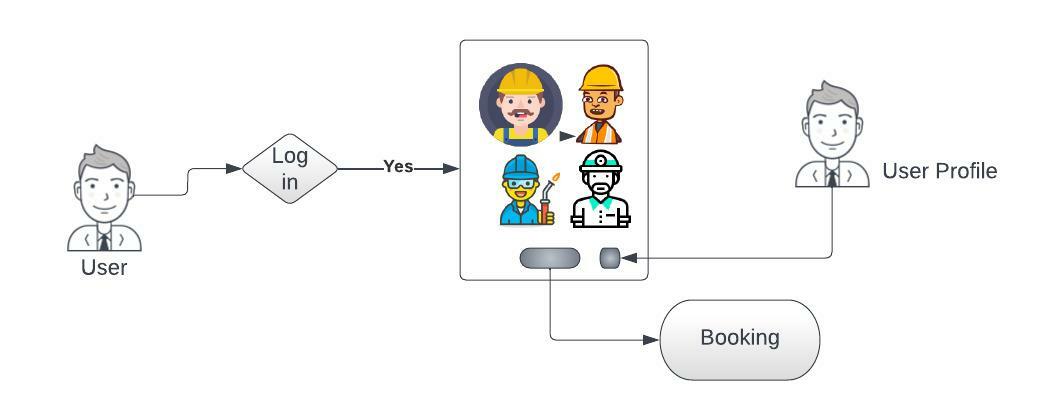
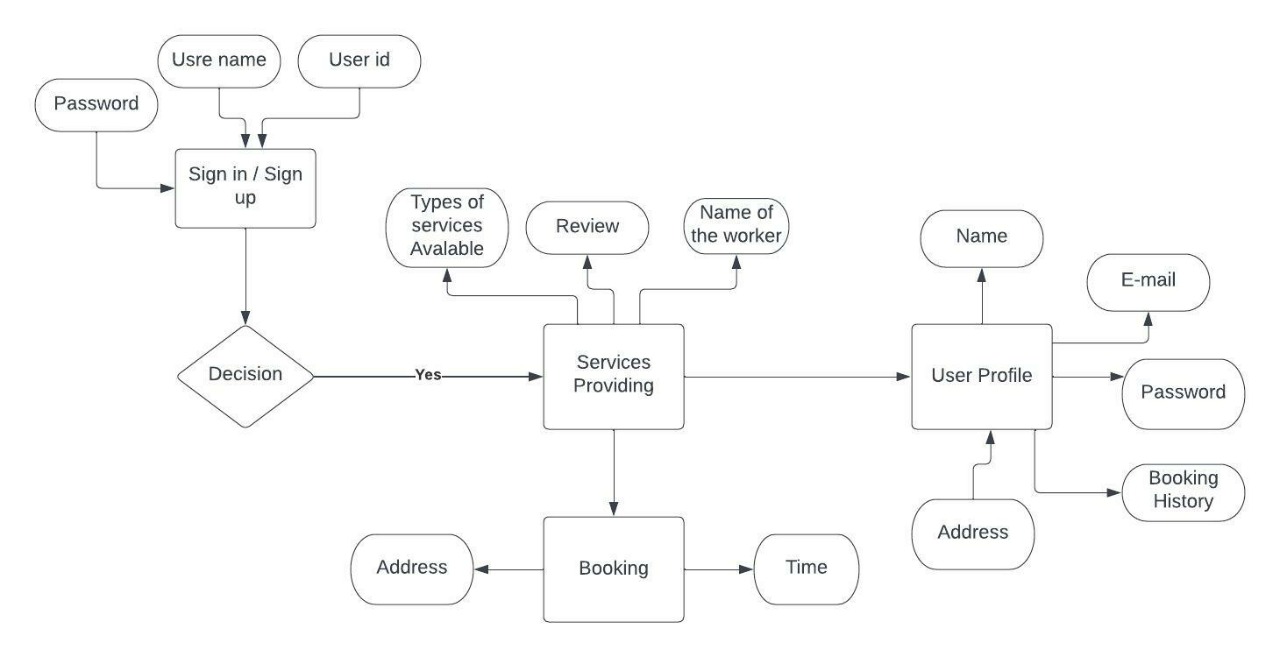


Figure 6

**3.4 E-R Diagram**



# 3.5 Overall Description

Our online system for household services provides the most expedient and annoys free wat to get your domestic work done. We aim to help in providing optimal solutions to all your household troubles with more efficiency, ease and majorly, a delicate touch. A single click system describes booking highly skilled in-house professionals and gets your service on time. Customers’ overall willingness to pay is significantly and positively correlated with the expectation that fee-based services would be better, and with the belief that “pay for what you get” is the right thing to do. Keeping that I sense our proposed system is basically a marketplace for household services and it is the platform where the rate was standardized and there is not necessitate haggling over prices. Several aspects like electrical work, carpentry work and plumbing services are involved in a system to provide happy and healthy home atmosphere in order to satisfy consumers.

# CHAPTER 4

# RESULT ANALYSIS AND VALIDATION

## 4.1 Result

Therefore, after implementing all the mentioned libraries as well as software tools, we finally full fledge application where users can make an account, log in as, visit our application and book their service they want.

## 4.2 Java Classes and Code

Java classes we used:

1. BookingAdapter
2. BookingForm
3. BookingHelperClass
4. BookingHistory
5. BookingModel
6. HomePage
7. HomeScreen
8. Login
9. MainActivity
10. Model
11. ModelGrid
12. Myadapter
13. myadapterGrid
14. myviewHolder
15. myviewHolder1
16. navigationActivity
17. proAdapter
18. ProfessionalsView
19. proHolder
20. proModel
21. SessionManager
22. UserHelperClass
23. UserProfile

Code: -

**4.3 Java MainActivity class code: -**

*package* com.example.serviceontime;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
  
*import* android.app.ActivityOptions;  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.os.Handler;  
*import* android.util.Pair;  
*import* android.view.View;  
*import* android.view.*WindowManager*;  
*import* android.view.animation.Animation;  
*import* android.view.animation.AnimationUtils;  
*import* android.widget.ImageView;  
*import* android.widget.TextView;  
  
*import* java.util.Objects;  
  
*public class* MainActivity *extends* AppCompatActivity {  
  
 *//variables* Animation topAnim, bottomAnim;  
 ImageView image;  
 TextView logo, slogan;  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
*// getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,WindowManager.LayoutParams.FLAG\_FULLSCREEN);* Objects.*requireNonNull*(getSupportActionBar()).hide();  
 setContentView(R.layout.***activity\_main***);  
  
 *//Animations* topAnim = AnimationUtils.*loadAnimation*(*this*, R.anim.***top\_animation***);  
 bottomAnim = AnimationUtils.*loadAnimation*(*this*,R.anim.***bottom\_animation***);  
  
 *//Hooks* image= (ImageView) findViewById(R.id.***imageView2***);  
*// image= (TextView) findViewById(R.id.imageView2);* logo= (TextView) findViewById(R.id.***textView***);  
 slogan= (TextView) findViewById(R.id.***textView3***);  
  
 *//Assign Animation* image.setAnimation(topAnim);  
 logo.setAnimation(bottomAnim);  
 slogan.setAnimation(bottomAnim);  
  
 *int* SPLASH\_SCREEN = 2000;  
 *new* Handler().postDelayed(*new* Runnable() {  
 @Override  
 *public void* run() {  
 Intent intent = *new* Intent(MainActivity.*this*,Login.*class*);  
 startActivity(intent);  
 finish();  
*// For Animation* Pair[] pairs;  
 pairs = *new* Pair[3];  
 pairs[0] = *new* Pair<View, String>(image, "logo\_image");  
 pairs[1] = *new* Pair<View, String>(logo, "logo\_text");  
 pairs[2] = *new* Pair<View, String>(slogan, "logo\_dec");  
  
 *if* (android.os.Build.VERSION.***SDK\_INT*** >= android.os.Build.VERSION\_CODES.***LOLLIPOP***) {  
 ActivityOptions options = ActivityOptions.*makeSceneTransitionAnimation*(MainActivity.*this*, pairs);  
 startActivity(intent,options.toBundle());  
 }  
 }  
 }, SPLASH\_SCREEN);  
 }  
}

**4.4 MainActivity XML Code: -**

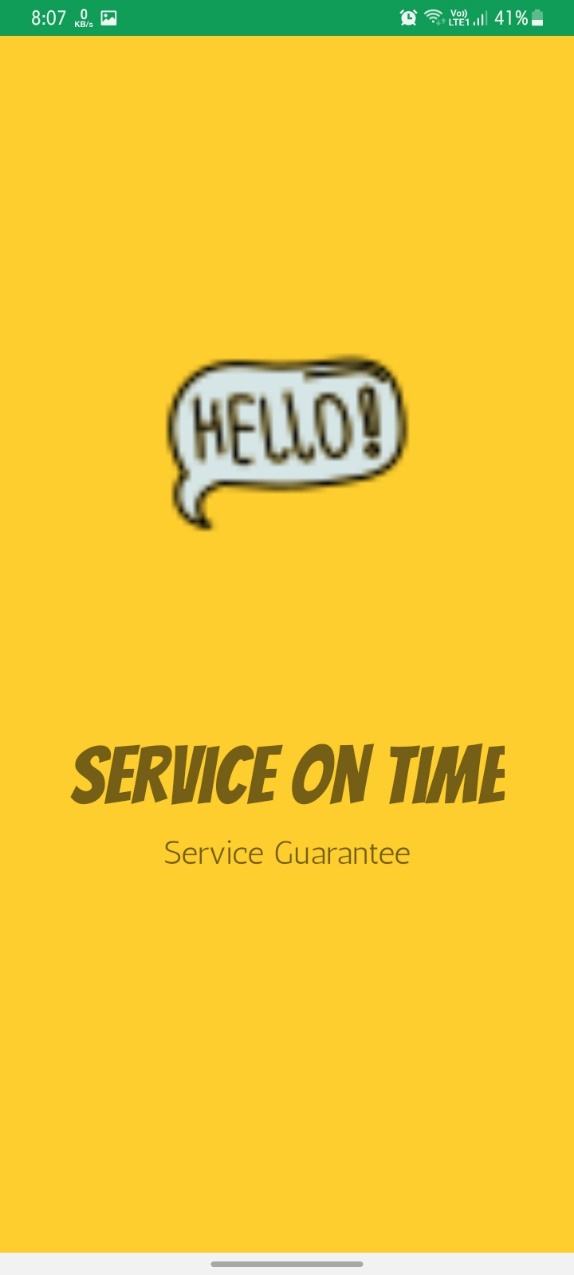
<?*xml version*="1.0" *encoding*="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout *xmlns:android*="http://schemas.android.com/apk/res/android"  
 *xmlns:app*="http://schemas.android.com/apk/res-auto"  
 *xmlns:tools*="http://schemas.android.com/tools"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent"  
 *tools:context*=".MainActivity"  
 *android:background*="#fece2f">  
  
 <ImageView  
 *android:id*="@+id/imageView2"  
 *android:layout\_width*="314dp"  
 *android:layout\_height*="175dp"  
 *android:src*="@drawable/hello"  
 *android:transitionName*="logo\_image"  
 *app:layout\_constraintBottom\_toBottomOf*="parent"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *app:layout\_constraintVertical\_bias*="0.298"  
 *android:contentDescription*="TODO"  
 *tools:ignore*="ContentDescription" /><TextView  
 *android:id*="@+id/textView"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:fontFamily*="@font/bangers"  
 *android:text*="Service On Time"  
 *android:textAlignment*="center"  
 *android:textSize*="60sp"  
 *android:transitionName*="logo\_text"  
 *app:layout\_constraintBottom\_toBottomOf*="parent"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/imageView2"  
 *app:layout\_constraintVertical\_bias*="0.207" />  
  
 <TextView  
 *android:id*="@+id/textView3"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:fontFamily*="@font/antic"  
 *android:text*="Service Guarantee"  
 *android:textAlignment*="center"  
 *android:textSize*="24sp"  
 *app:layout\_constraintBottom\_toBottomOf*="parent"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/textView"  
 *app:layout\_constraintVertical\_bias*="0.0" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

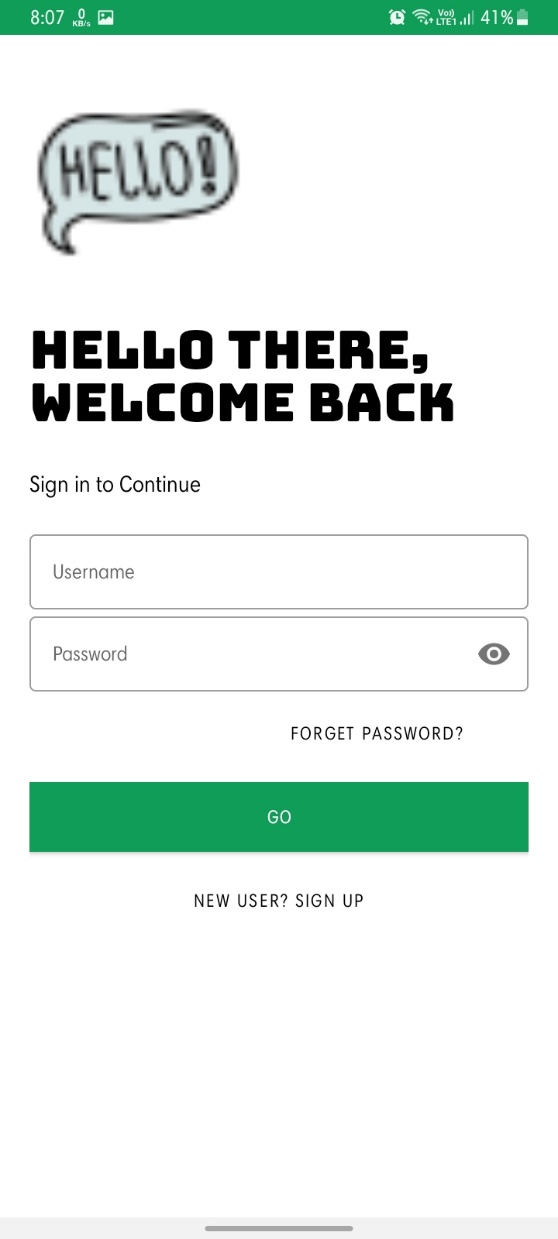
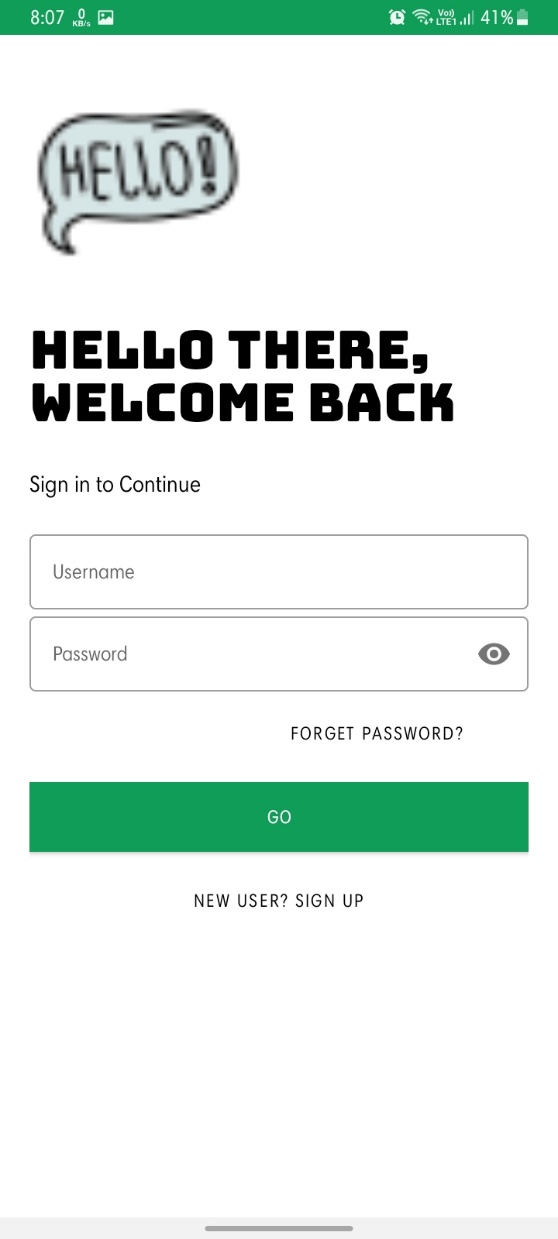
**4.5 SessionManager Java Class: -**

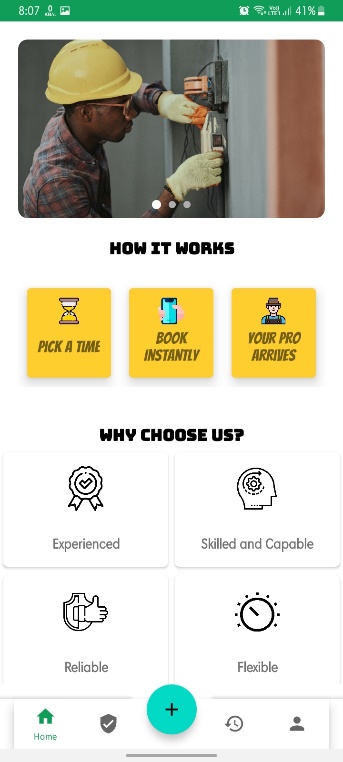
This java class is used to get all the information from the database and save in the local storage.

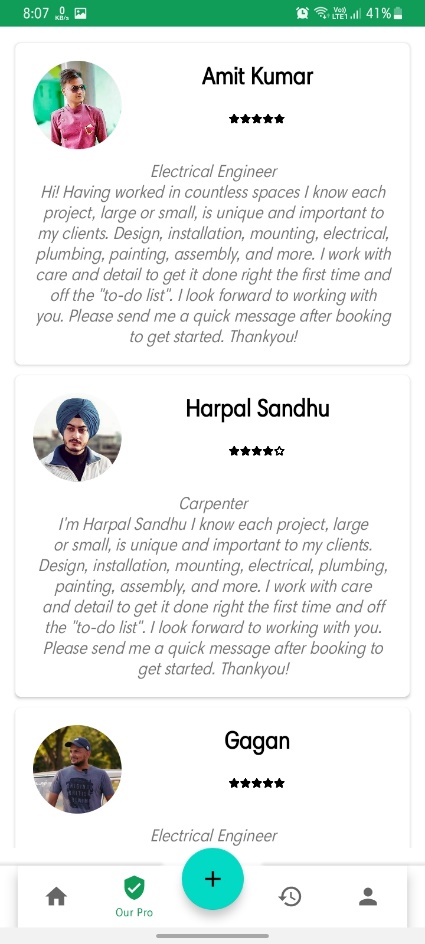
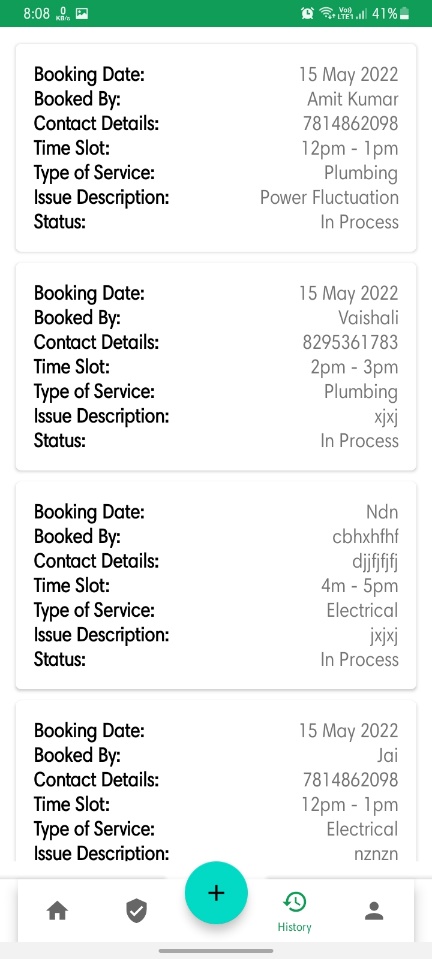
*package* com.example.serviceontime;  
  
*import* android.content.Context;  
*import* android.content.*SharedPreferences*;  
*import* android.se.omapi.Session;  
  
*import* java.util.HashMap;  
  
*public class* SessionManager {  
  
 *//Variables  
 SharedPreferences* userSession;  
 *SharedPreferences*.*Editor* editor;  
 Context context;  
  
 *private static final* String ***IS\_LOGIN*** = "IsLoggedIn";  
 *public static final* String ***KEY\_FULLNAME*** = "fullName";  
 *public static final* String ***KEY\_USERNAME*** = "username";  
 *public static final* String ***KEY\_EMAIL*** = "email";  
 *public static final* String ***KEY\_PHONENO*** = "phoneNo";  
 *public static final* String ***KEY\_PASSWORD*** = "password";  
  
 *public* SessionManager(Context \_context) {  
 context = \_context;  
 userSession = context.getSharedPreferences("userLoginSession", Context.***MODE\_PRIVATE***);  
 editor = userSession.edit();  
 }  
  
 *public void* createLoginSession(String fullName, String username, String email, String phoneNo, String password) {  
 editor.putBoolean(***IS\_LOGIN***, *true*);  
  
 editor.putString(***KEY\_FULLNAME***, fullName);  
 editor.putString(***KEY\_EMAIL***, email);  
 editor.putString(***KEY\_PHONENO***, phoneNo);  
 editor.putString(***KEY\_PASSWORD***, password);  
 editor.putString(***KEY\_USERNAME***, username);  
  
 editor.commit();  
 }  
  
 *public* HashMap<String, String> getUserDetailFromSession() {  
 HashMap<String, String> userData = *new* HashMap<String, String>();  
  
 userData.put(***KEY\_FULLNAME***, userSession.getString(***KEY\_FULLNAME***, *null*));  
 userData.put(***KEY\_USERNAME***, userSession.getString(***KEY\_USERNAME***, *null*));  
 userData.put(***KEY\_EMAIL***, userSession.getString(***KEY\_EMAIL***, *null*));  
 userData.put(***KEY\_PHONENO***, userSession.getString(***KEY\_PHONENO***, *null*));  
 userData.put(***KEY\_PASSWORD***, userSession.getString(***KEY\_PASSWORD***, *null*));  
  
 *return* userData;  
 }  
  
 *public boolean* checkLogin() {  
 *if* (userSession.getBoolean(***IS\_LOGIN***, *true*)) {  
 *return true*;  
 } *else* {  
 *return false*;  
 }  
 }  
  
 *public void* logoutUserFromSession() {  
 editor.clear();  
 editor.commit();  
 }  
}

## 4.6 Snapshot

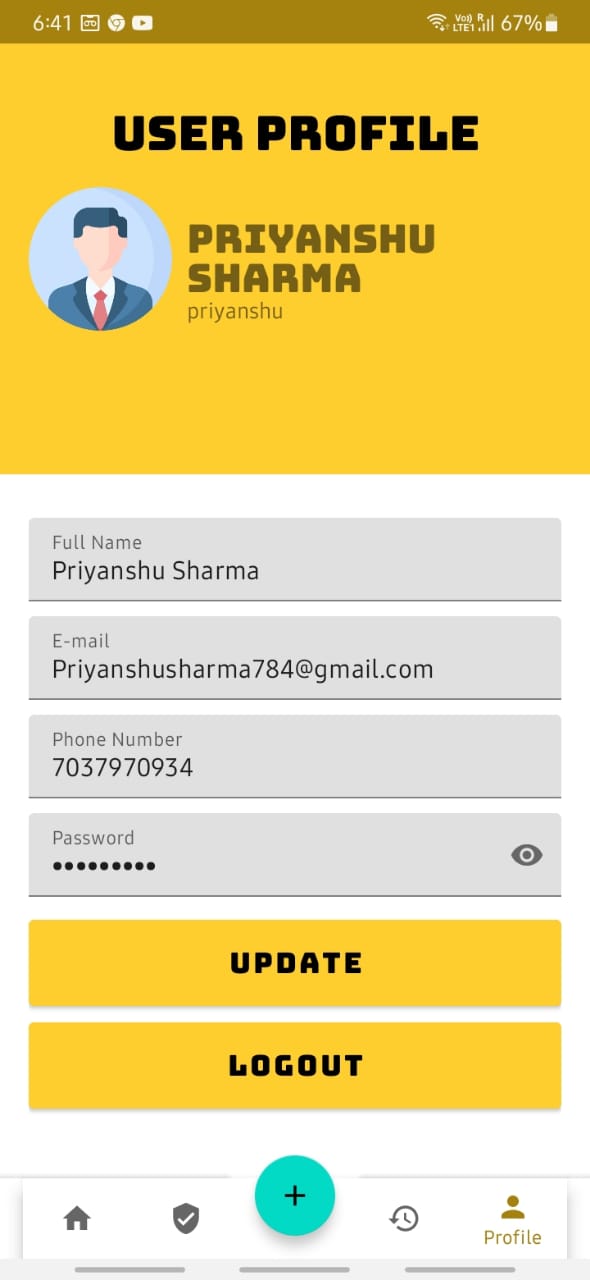
Splash Screen

******Signup Screen**

**Home Screen:** -

Professional Profile and booking History: -

User Profile Screen



**4.7 Code link:-**

All the contents and Code are in the link.

<https://drive.google.com/drive/folders/1QfOTJ4TvNaV2z2QrGYN-56HwUOKyEFvX?usp=sharing>

# CHAPTER 5

# CONCLUSION AND FUTURE Work

## 5.1 Conclusion

Our project is only a humble venture to satisfy the needs to manage their project work. Several userfriendly coding has also been adopted. The objective of the software planning is to provide a framework with a limited project completion time frame at the beginning of the project and should be updated on a regular basis.

## 5.2 Future scope

The online household services application provides some of the home services which are most frequently used. This system accommodates the changing needs of the end user. The overall system can be designed so that its capacity can be increased in response to the further requirements for which the application provides an appropriate service overseas. Further this application can be prolonged by merely adding up the required services and additional payment systems. For example, the current system provides the following services such as home painting, home cleaning, packers and movers, plumber repair and service further the system can be extended as per the requirements of the user. The system can have prolonged by adding the services such as mobile and computer repair, laundry services, catering services and many more. The discussion payment methods our system has, for example currently system has no online payment option, further it can be extended by adding the payment services.

## 5.3 Possible Future Work

* We can give more advance software for online car rental systems and add more facilities.
* We can add printer in the future.
* Integrate multiple load balancers to distribute the load of system.
* Create a backup mechanism for backing up data and information.
* We will host the platform on online servers, to make it accessible worldwide.
* We can add online payment option.
* We can add toll free customer care option.
* We can do two step verification method.

## REFERENCES

* Barrell, Dylan. Agile Accessibility Explained: A practical guide to sustainable accessible software development, Amazon Digital Services, 2019.
* Blanck, Peter. eQuality: The Struggle for Web Accessibility by Persons with Cognitive

Disabilities, Cambridge Disability Law and Policy Series, 2015.

* Burgstahle, Sheryl. Creating Inclusive Learning Opportunities in Higher Education, Harvard

Education Press, 2020.

* Burgstahle, Sheryl. Universal Design in Higher Education: From Principles to Practice, Harvard

Education Press, 2008.

* Byrne, Jim. 60 hot to touch Accessible Web Design tips – the tips no web developer can live without! Jim Byrne, 2006, (ISBN: 978-1-4116-6729-7).
* Chisholm, and May. Universal Design for Web Applications: Web Applications That Reach

Everyone, O'Reilly Media, 2008.

* Anderson, Erin, et al. Interact with Web Standards: A holistic approach to web design, New

Riders, 2010.

* Allsopp , John. Developing with Web Standards, New Riders, 2010.
* Budd, Andy, et al. Web Standards Creativity: Innovations in Web Design with XHTML, CSS, and DOM Scripting, Friends of ED, 2007.
* Budd, Andy, et al. CSS Mastery: Advanced Web Standards Solutions, Friends of ED, 2006.

## ONLINE SOURCES