CITYSERVICES

The Project submitted to

Sant Gadge Baba Amravati University, Amravati

towards partial fulfillment of the Degree of

Bachelor of Engineering

in

Information Technology

Guided by

Prof. Sagar Padiya

Submitted by

Mr. Rohitraj Deshmukh

Mr. Monish Sathe

Mr. Akhil Jamatkar

Mr. Vishal Kothalkar

DEPARTMENT OF INFORMATION TECHNOLOGY
SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON (M.S.)
2021-2022

CITYSERVICES

The Project submitted to

Sant Gadge Baba Amravati University, Amravati

towards partial fulfillment of the Degree of

Bachelor of Engineering

in

Information Technology

Guided by

Prof. Sagar Padiya

Submitted by

Mr. Rohitraj Deshmukh

Mr. Monish Sathe

Mr. Akhil Jamatkar

Mr. Vishal Kothalkar

DEPARTMENT OF INFORMATION TECHNOLOGY
SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON (M.S.)
2021-2022

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING
SHEGAON

2021-2022

CERTIFICATE

This is to certify that **Mr. Rohitraj Deshmukh, Mr. Monish Sathe, Mr. Akhil Jamatkar, Mr. Vishal Kothalkar** students of final year B.E. (Information Technology) in the year 2021-2022 of Information Technology Department of this institute has completed the project work entitled "**CityServices**" based on syllabus and has submitted a satisfactory account of his/her work in this report which is recommended for the partial fulfillment of degree of Bachelor of Engineering in Information Technology.

Prof. Sagar Padiya
(Project Guide)

Dr. A S Manekar
Head, Dept. of IT
S.S.G.M. College of Engineering,
Shegaon

Dr. S. B. Somani
Principal
S.S.G.M. College of Engineering,
Shegaon

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING SHEGAON

2021-2022

CERTIFICATE

This is to certify that the project work entitled "**CityServices**" submitted by **Mr. Rohitraj Deshmukh, Mr. Monish Sathe, Mr. Akhil Jamatkar, Mr. Vishal Kothalkar** name students of final year B.E. (Information Technology) in the year 2021-2022 of Information Technology Department of this institute, is a satisfactory account of his work based on syllabus which is approved for the award of degree of Bachelor of Engineering in Information Technology.

Internal Examiner

Date:

External Examiner

Date:

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this seminar in this form without their valuable help, cooperation and guidance.

First and foremost, I wish to record my sincere gratitude to Management of this college and to our beloved Principal, ***Dr. S B Somani****, for their constant support and encouragement in preparation of this project and for making available internet, library and laboratory facilities needed to prepare this project.*

Further my sincere thanks to ***Dr. A S Manekar****, Head of the Department, Information Technology, for his valuable suggestions and guidance throughout the period of this project.*

I express my sincere gratitude to my guide, ***Prof. Sagar Padiya*** *for guiding me in investigations for this seminar and in carrying out relevant work. Our numerous discussions were extremely helpful. I received his/her esteem guidance, encouragement and inspiration.*

Last but not the least, I wish to thank my parents for financing my studies in this college as well as for constantly encouraging me to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

Student Names

1 Mr. Rohitraj Deshmukh

2 Mr. Monish Sathe

3 Mr. Akhil Jamatkar

4 Mr. Vishal Kothalkar

ABSTRACT

In this covid pandemic situation, small-scale entrepreneurs suffered due to lockdown as well as common people were unable to get most of the services at their doorstep. In the current scenario, we people are so into technology that we don't want to make too many efforts to get things done, we always wanted an easy way to get things done in less time as well. Suppose any of user's appliances breaks down, let's say your cooler breaks down in hot summer then it would be a very tedious task to go out in summer and find a repairing person to overcome such situation our application **"CityService"** will came into the picture. So, this project is all about providing the household services at user's doorstep by finding the nearest professional at your fingertips. Similarly, the app would help small-scale entrepreneurs to grow faster and reach out to many customers in the city.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this fast-growing technology, the developer still has to take the appointment of a person who solves the problems related to our daily life like plumbing-related problems, mechanical problems, electrical problems, electronic problems, pest control, etc.

To take an appointment with the service provider developer have to call him or with a personal meeting developer can meet him, and it is not sure that the developer gets the appointment of the service provider at a time because there are many problems occur, like the service provider is busy at somewhere else or he is not present at his office when developer go there or he wants heavy cost for fix the problem, etc.

Developers are not getting any service on time and also not proper changes of services. It is also not secure in terms of safety concerns.

To overcome these types of problems developers are going to make our React Native application where the people get the appropriate result. This React native application is very dynamic and very easy to understand. The interface of the react-native application is very easy and anybody can easily work on it. This React Native application can provide all the descriptions and important information about the problem.

1.1 PREFACE

Imagine that user AC unexpectedly stops working in the summer as the clock ticks at 11 p.m. Now that you feel hot that you need a person to fix users' AC, you'll ask users' relatives if they know someone who can fix them. Does all reject what to do next as his night? Now, if an agency comes into motion where I will ask the individual and boom for AC repair! Now they're sending someone to fix and save the user from hot summer nights. Okay, "CityServices" does just that. This project will describe the implementation of the collaborative process of improving the consolidation of services through an Android app that also analyzes better results which the existing system was not doing. The App is more user friendly so that anyone can handle the App without having any complexity. CityServices is an Android app that works closely with Service Providers for Individuals (ISPs) and turns them into micro-entrepreneurs.

CityServices does so by offering market access, credit, acquisition, and a host of other services to ISPs. Developers give their job-seekers “leads.” By standardizing the end-user experience, pricing, and distributing the service under the brand name of CityServices, they drive their entire business.

1.2 STATEMENT OF PROBLEM

When someone needs 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 deliver consistently flawless service on instance. Our city services React Native application provides the most expedient and annoys-free way to get users domestic work done. Keeping that in sense our proposed system is a marketplace for services and it is the platform where the rates were standardized and there is no necessary haggling over prices. Several aspects like painting, pest control, home cleaning, plumbing, electrical works, and carpentry services are involved in a system to provide a happy and healthy home atmosphere to satisfy consumers Developer aims to help in providing optimal solutions to all user's household troubles with more efficiency, ease and majorly, a delicate touch. A single click system describes booking highly skilled in-house professionals and gets user's service done 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 users get” is the right thing to do.

1.3 OBJECTIVES OF THE PROJECT

- To develop CITY SERVICES Android Application for the USER with the help of the latest technology.
- Aggregation of service providers (housekeepers, electrician, Beautician, etc) details in one place.
- Book professional for services, conformation mail, etc
- Review and Rating of service providers by customers.
- Advertisement of Services (For the generation of revenue)

1.4 SCOPE AND LIMITATIONS OF THE PROJECT

The scope of our project is to design a complete environment to provide a safe and user-friendly environment for online service booking. The main aim of the project is to provide an easy-to-use application for services provided to the customer. Developers often get frustrated while taking the appointment of service provider because there the many problems occur like the service provider is busy art somewhere else or his not receiving our call or his cost is very high according to a problem. So in this project, the developer will remove this headache.

Developers have observed how limitations in the existing system:

- Existing system is offline.
- Difficult to manage records.
- No time limit for service to be provided.
- No guaranteed service.
- Difficult to find a paper service provider.
- 24 hours service is not available.
- No security.

So, our purpose is to overcome this limitation with the following features.

- Household services are easily available.
- To provide household services at any time.
- Easy online payment.
- Saving time.
- Make available household services through the developer site & application.

1.5 ORGANIZATION OF THE PROJECT

The organization of our project includes two Applications one is a USER Application and another is a PROVIDER Application. So, the developer has created two applications.

USER Applications: -

This application is used on the user end. In the user application, the user will get the all the screens like Login Screen, Register Screen, Home Screen, Categories, list of professionals, support, and a lot more. Also, the user developer gets all the information about professionals, all the categories, and all the services. Where users can book any service as per user's requirement. Users will get all the functionality that the developer has provided to do the user task easier.

PROVIDER Application: -

This application is used on the Provider end. In the provider application, the provider will get all the requests made by a user like upcoming requests, cancel requests, today's requests, etc. Also, he can make a call and do chatting with the user. Also, he will see all the reviews and ratings made by a user for the specific professional.

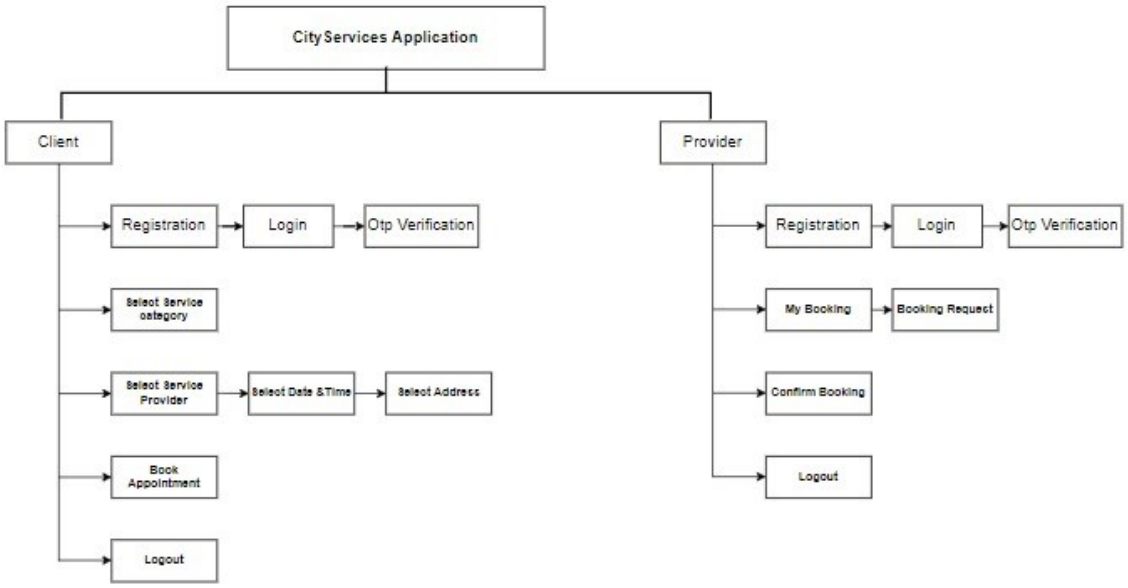


Figure 1: Organization of Project

2. LITERATURE SURVEY

When someone needs 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 deliver consistently flawless service on instance. Our online system for household services provides the most expedient and annoys-free way to get users domestic work done. Developers aim to help in providing optimal solutions to all user's household troubles with more efficiency, ease, and majorly, a delicate touch. A single click system describes booking highly skilled in-house professionals and gets user's service done 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 users get" is the right thing to do. Keeping that in sense our proposed system is a marketplace for household services and it is the platform where the rates were standardized and there is no necessary haggling over prices. Several aspects like painting, pest control, home cleaning, plumbing, electrical works, and carpentry services are involved in a system to provide a happy and healthy home atmosphere to satisfy consumers. Developers have observed so many problems which occur in our daily routine, for example, the problem of electricity, if the electricity goes there are many problems occurs in our work, like a developer cannot charge the phones, batteries, etc. the electric devices which are used in the kitchen cannot be work.

In the existing system, the whole process is offline and the customer can not get a proper solution for the problem. The existing system is not available for 24 hours, so the customer can not get a solution at any time.

The developer also observes the problem of plumbing at our home, there is one type of plumbing problem is occurs, In that situation what developer will do. Firstly developer calls the plumber and takes his appointment if available or the developer directly meets him for the appointment to fix the problem. This process is very time-consuming and it is not sure that the developer gets the appropriate solution for our problem.

The developer also noticed that if the developer is going somewhere and our vehicle is getting suddenly stopping. In this situation what the developer will do is that developer

call the mechanic to fix the problem which occurs in our vehicle, and at one moment if the developer block in some place where no network is available on our phone and at that place, the rush of traffic is very less so which kind of problem developer have to face. Firstly Developer has to find the mechanic and ask him to fix the problem. This process is very time-consuming and it is not sure that the developer gets the solution for the problem because many problems occur in that situation like a mechanic is busy with other work or his payment is very high according to the problem etc.

By observing these types of problems developer thought that how can the developer get the solution to these types of problems easily and appropriately, the developer get the idea to develop our android application and website where people can get any type of service like plumbing, electrical, electronics, paint, pest control, etc. In this stage of technology, our application is very simple and easy to understand.

URBAN CLAP

Urban Clap is an app-based service marketplace that connects the customer to service professionals. Their strategy is to connect a greater number of customers to use the platform of Urban Clap to make their life easier and more comfortable. With the rise in nuclear families, Dual Career couples, the focus of customers is to spend quality time with their families whenever possible. Services at the doorstep at one click of the mouse are welcoming change accepted by customers today, giving rise to a business model like Urban Clap, which is here to stay for a long time. However, the success of these businesses is well dependent on how successfully Urban Clap can meet the expectations of its customers, reduce their pain and provide overwhelming satisfaction to its customer base.

Findings

Technology Stack:

- Product and Design: Adobe Photoshop
- Programming language and Framework: Java, MySQL, and so on.

Problem Statements

1) Adobe Photoshop: Adobe Photoshop is an excellent method for creating and editing photographs and other types of media files. It is most likely the most common tool, with millions of users worldwide. And though it has a lot of features and a simple way to manipulate files, it has a lot of bugs. Each latest edition includes a slew of bug fixes. However, developers are discussing the shortcomings of the app in general and its use over other applications on the market that are close to Photoshop.

Even though Adobe Photoshop is almost flawless in every way, it also has a lot of flaws. Like any other piece of software, no matter how good it is, certain flaws wreak havoc.

Newcomers Can Become Perplexed

Photoshop is intended to be useful for a wide range of applications. That is, everyone can master it up to a certain degree with the aid of some tutorials. However, if the user is a complete novice with no prior experience with image editing or design software, the user can find Photoshop challenging to use. Specifically, to comprehend the toolset. It is primarily due to the vast amount of functions available on the app that users can get confused. That does not mean the user can never be a Photoshop pro. It just takes a short amount of time as compared to other simple resources available.

Support for vector graphics

Photoshop's native compatibility with vector graphics is also limited. And if the user uses additional plugins to extend the functionality and usability, the user can not edit any of the vector formats that need modifications. Many who deal with websites, in particular, can find it challenging to edit and create formats such as SVG. As a result, Photoshop is essentially a raster image editor.

Problems with performance

Even when displaying high-quality graphics on an intermediate PC with mediocre specifications and hardware, the user cannot see any performance problems. However, those using novice computers can experience intense lags during initialization, as well as difficulty rendering even tiny images or graphics. The app can be very resource-intensive at times due to the large range of features that not

everybody can use. In this case, older models are preferable because they seem to run more regularly on less hardware.

2) Java:

Poor UI Experience

- To make a program's graphical UI (GUI), designers utilize diverse language-explicit tools. In this way, for Android applications, there's Android Studio that makes applications that look and feel native. There are a couple of GUI manufacturers Java developers can browse: Swing, SWT, JavaFX, and JSF being the most well-known.
- SWT utilizes native elements however it's not appropriate for convoluted UI. JavaFX is spotless and current-looking, yet entirely it's not exceptionally full grown. By and large, picking a solid match for the user's GUI expanding on Java requires extra deliberation.

Readability Issues

- Many Java developers would attest to the fact that the Java code is verbose and involves a lot of lines of code. It means that the work that could be done in fewer coding syntaxes in other programming languages requires more effort in Java.
- There are complicated sentences and the code is difficult to understand sometimes. Developers need to comment specifically if there are any changes they want in their Java software development projects.

Objectives

For the development of the above-stated project our objectives are

1) To develop a portal that will be able to complete the Product and Design using Adobe XD

Discussion: The work in XD is more straightforward and quicker than in Photoshop. The main reason is XD is targeted primarily to designing and prototyping. Photoshop has extensive functionality — photo and illustration creation and editing, 3D graphics designing, etc. It makes it much more robust and therefore the work is not so quick and

smooth. The developer can save all colors and character styles with just one click and reuse them consistently on each screen. The developer can also edit colors and character styles. All the changes the user made are automatically applied to every object that uses the respective color or character style. In Photoshop developers have to save each color and character style separately and the developer can't edit it later, so this is a considerable advance of XD. The developer can save repeated GUI parts (for example buttons, text fields, navigation bars, etc.) as symbols and reuse them consistently on each screen. When the developer edits the symbols later, the change is automatically applied to every place the symbol is used. The developer can also edit the symbols separately (the change will not apply to other symbols) — just by clicking on Ungroup Symbol option in the context menu. In Photoshop developer can use Smart Objects but the developer will see the changes after saving the edited object, not in real-time like in XD.

2) To develop the entire application in React Native

- **To get the Cross-platform compatibility**

  Discussion: One of the significant advantages of the React Native framework is "the use of one codebase." As a result, cross-platform apps show the same level of performance as native apps.

  React Native uses React JS with a native UI library and native APIs to write the base code of one application that can run on both iOS and Android platforms.

  So, for a web app that uses React JS, the developer can re-use the state management, component logic, and React hooks to have a better mobile application with all the native functionalities. This reduces the development time and helps the user keep the overall app development cost low.

  Faster development means the user can launch the user's app as fast as possible and join the competition earlier than the user planned. Also, with its one codebase technique, users don't have to hire separate developers to build apps for iOS and Android devices.

- **To develop Reusable code and pre-built components**

  Discussion:

The concept of cross-platform mobile app development is that users can reuse the written code for both iOS and Android apps. React Native also follows the same principle. It follows the notion write once and launching everywhere.

This adds convenience to developers. They don't need to build separate designs for different platforms. Also, if a company has an existing web app, it can use most of its code to build a mobile app.

- **To perform Coding in a familiar programming language**

  Discussion:

  To develop a user's app using React Native, users don't need users developers to learn a different language and coding syntaxes if they are already familiar with JavaScript.React Native is easy to learn if the developer has a working knowledge of JavaScript or React. The important thing is they should know which mobile components correspond with which web components. Once this is done, the user's developer can start using React Native for the user's mobile app development. Also Read: How to Make An App for Android and iOS Using React Native?

- **Trustworthy**

  Discussion:

  Before React Native was developed and launched, Facebook made a mobile app to check the platform's viability. It helped their developers to build an amazing mobile app for the social media platform. Today, everyone uses the Facebook mobile app, and, indeed, it gives the user a feeling of using a native mobile app. Since, React Native has gone open-source, more companies have started using React Native to develop their mobile apps. Instagram, Skype, Walmart, etc., are a few examples.

- **To Build stable and high-performing apps**

  Discussion:

  React Native uses a simplified binding strategy for its codebase. Therefore, to change one object, the user has to modify its state before applying updates. That will help the user develop a more stable and reliable application than other cross-platform development methods. When users think of building a new mobile app

from scratch, React Native is a great option However, it also works well for adding a single view or user flow to existing native applications. With a few steps, the user can include new React Native-based features, views, screens, etc. Also, instead of Webview, React Native uses native APIs for rendering code. As a result, it helps users deliver better-performing apps.

- **To get the Support of third-party plugins**

  Discussion:

  React Native offers third-party plugin support. The two types of plugins applicable are Native Modules and JavaScript Modules. So, for instance, if the user wants to add Google Calendar or Google Maps to the user's basic app, then through React Native, the user can link to any plugin with a Native or third-party module.

- **Growing all the time**

  Discussion:

  Facebook and the massive community are constantly working on improvements to React Native. That means if React Native cannot offer the user a solution for a problem right now, it can't be the same all the time. A few months later, the situation might be different.

- **Hot and Live Reloading**

  Discussion:

  React Native has a hot and live reloading function at its core. With this feature, the developers can work with the code changes in real time and fix them when the application is loading. While hot reloading reloads only a particular area of the change, the live feature automatically reloads the complete application after making changes. This eases the development and testing process.

- **To develop a Rich UI**

  Discussion:

  React Native helps the user create unique and compelling UIs through pre-built declarative components, such as Button, Picker, Slider, Switch, and so on. Users can use TouchableNativeFeedback and TouchableOpacity to make users

components. There are many other Android and iOS-specific components available to make it work efficiently on iOS and Android mobile devices.

Examples:

iOS: SegmentedControliOS, ImagePickerControlleriOS,DataPickeriOS,etc.

Android: DrawerLayoutAndroid, DatePickerAndroid, ViewPageAndroid, etc.

FOREX

Forifix is an Integrated Pest Management venture, launched on September 5, Defense Day as a shield against the domestic terrorists (Pests and their pollutants) in and around the facilities. It addresses a significant gap between a serious pain point in every household and the horrid solutions available that one could ever risk. Forifix offers Pest Prevention with Safer, Odorless, and environment-friendly products approved by WHO/EPA and FDA. In addition to this, they provide Heat Proofing, Water Tank Cleaning, and Home Improvement with allied repair and maintenance services.

HANDYMAN

Handyman which was incorporated on 21st November 2014, The Handyman was the first Professional Handyman Service provider in Pakistan. The ‘Handyman’ is a term used to describe technicians such as Electricians, Plumbers, AC Mechanics, Carpenters, Wall Painters, etc. Starting initially in Islamabad, they have a team of qualified handymen available to help households and businesses with cleaning, electrical, plumbing, painting, and other jobs that require urgent attention. They also provide cleaning & janitorial services, repair works as well as tasks related to construction management (building or constructing a house, office, or a commercial plaza, or remodeling a user's property).

3. ANALYSIS

3.1 Detailed Statement of the Problem

Home Services app is the new trend in the market of on-Demand applications. With proper market research, the inclusion of vital features, followed by appropriate marketing can make the app successful. The demand for Home Services applications will be on the rise as developers imagine developers all want an Iron in our lives. The age-old canters are not going to cause the user any worries anymore. The trusted home services application with professional and qualified personnel can repair and fix everything around user's home in an efficient manner. Problems get accentuated with rapid urbanization, rising incomes, and an abundance of low-cost workers. People are constantly in a hurry and are willing to pay more to ensure a certain level of service. The only other option available when scouting for these handymen is to avail of the services of inept search directories and run the risk of being bombarded by incessant calls. Moreover, with the smartphone being the preferred gateway to these services for most people and with monumental growth in its penetration across the country. As the existing application "Urban Clap" is only available in metro cities, this application extends its reach to all the cities across India. This Application also aims to implement a contract-based model for flats around the city as an add-on service to their apartments at a reasonable amount.

3.2 Requirement Specifications

Hardware Requirements:

1. Processor: Intel Pentium IV and above
2. RAM: 2GB or more
3. Hard disk 500 GB and more

Software Requirements:

1. Microsoft Windows XP and above
2. Smart Phone: Oreo 2 GB RAM 8 GB storage and more**.**
3. VS Code.

Other Requirements:

1. Web Browser: Internet Explorer, Google Chrome, Mozilla Firefox.

Documentation:

1. Microsoft Word 2010 and above
2. Microsoft Excel 2010 and above
3. Microsoft Visio 2010 and above
4. Edraw Max 8.0

3.3 Technology

Front-End Development

Front-end developers build with the user in mind. Front-end development is a style of computer programming that focuses on the coding and creation of elements and features of a website that will then be seen by the user. It's about making sure the visual aspects of a website are functional. Users can also think of the front end as the "client-side" of an application. So let's say the user is a front-end developer. This means the user's job is to code and bring to life the visual elements of a website. The user would be more focused on what the user sees when they visit a website or app. And, users want to make sure the site is easy to interact with while also running smoothly.

These developers take the visual designs from UX and UI designers and bring the website to life, making sure it functions well for the user. One of the many ways a user could use front-end skills is by creating a static website, which is a website with fixed content that's delivered to a user's browser exactly as it's stored. Users might run into a static website if the user happens upon a simple landing page or a small business website that doesn't allow users to perform any interactive tasks.

Front end developers build elements like:

- Buttons
- Layouts
- Navigation
- Images
- Graphics
- Animations
- Content organization

- JavaScript

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

1. History

- Creation at Netscape

The first web browser with a graphical user interface, Mosaic, was released in 1993. Accessible to non-technical people, it played a prominent role in the rapid growth of the nascent World Wide Web. The lead developers of Mosaic then founded the Netscape corporation,which released a more polished browser, Netscape Navigator, in 1994. This quickly became the most used.

During these formative years of the Web, web pages could only be static, lacking the capability for dynamic behavior after the page was loaded in the browser. There was a desire in the burgeoning web development scene to remove this limitation, so in 1995,

Netscape decided to add a scripting language to Navigator. They pursued two routes to achieve this: collaborating with Sun Microsystems to embed the Java programming language, while also hiring Brendan Eich to embed the Scheme language.

Netscape management soon decided that the best option was for Eich to devise a new language, with syntax similar to Java and less like Scheme or other extant scripting languages. Although the new language and its interpreter implementation were called LiveScript when first shipped as part of a Navigator beta in September 1995, the name was changed to JavaScript for the official release in December.

The choice of the JavaScript name has confused, implying that it is directly related to Java. At the time, the dot-com boom had begun and Java was the hot new language, so Eich considered the JavaScript name a marketing ploy by Netscape.

- Adoption by Microsoft

Microsoft debuted Internet Explorer in 1995, leading to a browser war with Netscape. On the JavaScript front, Microsoft reverse-engineered the Navigator interpreter to create its own, called JScript.

JScript was first released in 1996, alongside initial support for CSS and extensions to HTML. Each of these implementations was noticeably different from their counterparts in Navigator. These differences made it difficult for developers to make their websites work well in both browsers, leading to widespread use of "best viewed in Netscape" and "best viewed in Internet Explorer" logos for several years.

- The rise of JScript

In November 1996, Netscape submitted JavaScript to Ecma International, as the starting point for a standard specification that all browser vendors could conform to. This led to the official release of the first ECMAScript language specification in June 1997.

The standards process continued for a few years, with the release of ECMAScript 2 in June 1998 and ECMAScript 3 in December 1999. Work on ECMAScript 4 began in 2000.

Meanwhile, Microsoft gained an increasingly dominant position in the browser market. By the early 2000s, Internet Explorer's market share reached 95%. This meant that JScript became the de facto standard for client-side scripting on the Web.

Microsoft initially participated in the standards process and implemented some proposals in its JScript language, but eventually, it stopped collaborating on Ecma work. Thus ECMAScript 4 was mothballed.

- Growth and standardization

During the period of Internet Explorer dominance in the early 2000s, client-side scripting was stagnant. This started to change in 2004, when the successor of Netscape, Mozilla, released the Firefox browser. Firefox was well received by many, taking significant market share from Internet Explorer.

In 2005, Mozilla joined ECMA International, and work started on the ECMAScript for XML (E4X) standard. This led to Mozilla working jointly with Macromedia (later acquired by Adobe Systems), who were implementing E4X in their ActionScript 3 language, which was based on an ECMAScript 4 draft. The goal became standardizing ActionScript 3 as the new ECMAScript 4. To this end, Adobe Systems released the Tamarin implementation as an open-source project. However, Tamarin and ActionScript 3 were too different from established client-side scripting, and without cooperation from Microsoft, ECMAScript 4 never reached fruition.

Meanwhile, very important developments were occurring in open-source communities not affiliated with ECMA work. In 2005, Jesse James Garrett released a white paper in which he coined the term Ajax and described a set of technologies, of which JavaScript was the backbone, to create web applications where data can be loaded in the background, avoiding the need for full page reloads. This sparked a renaissance period of JavaScript, spearheaded by open-source libraries and the communities that formed around them. Many new libraries were created, including jQuery, Prototype, Dojo Toolkit, and MooTools.

A major addition to the specification were event listeners, which date back to at least the early 2000s. However, Microsoft Internet Explorer only supported a proprietary

method named "attachEvent" before version 9, released in 2011, making "click" events preferred for compatibility.

Google debuted its Chrome browser in 2008, with the V8 JavaScript engine that was faster than its competition. The key innovation was just-in-time compilation (JIT), so other browser vendors needed to overhaul their engines for JIT.

In July 2008, these disparate parties came together for a conference in Oslo. This led to the eventual agreement in early 2009 to combine all relevant work and drive the language forward. The result was the ECMAScript 5 standard, released in December 2009.

- Reaching maturity

Ambitious work on the language continued for several years, culminating in an extensive collection of additions and refinements being formalized with the publication of ECMAScript 6 in 2015.

The creation of Node.js in 2009 by Ryan Dahl sparked a significant increase in the usage of JavaScript outside of web browsers. Node combines the V8 engine, an event loop, and I/O APIs, thereby providing a stand-alone JavaScript runtime system. As of 2018, Node had been used by millions of developers, and npm had the most modules of any package manager in the world.

The ECMAScript draft specification is currently maintained openly on GitHub, and editions are produced via regular annual snapshots. Potential revisions to the language are vetted through a comprehensive proposal process. Now, instead of edition numbers, developers check the status of upcoming features individually.

The current JavaScript ecosystem has many libraries and frameworks, established programming practices, and substantial usage of JavaScript outside of web browsers. Plus, with the rise of single-page applications and other JavaScript-heavy websites, several transpilers have been created to aid the development process.

2. Client-Side Usage

JavaScript is the dominant client-side scripting language of the Web, with 97% of websites using it for this purpose. Scripts are embedded in or included

in HTML documents and interact with the DOM. All major web browsers have a built-in JavaScript engine that executes the code on the user's device.

- Examples of Scripted Behavior
    - Loading new web page content without reloading the page, via Ajax or a WebSocket. For example, users of social media can send and receive messages without leaving the current page.
    - Web page animations, such as fading objects in and out, resizing, and moving them.
    - Playing browser games.
    - Controlling the playback of streaming media.
    - Generating pop-up ads.\
    - Validating input values of a web form before the data is sent to a web server.
    - Logging data about the user's behavior and then sending it to a server. The website owner can use this data for analytics, ad tracking, and personalization.
    - Redirecting a user to another page.
- Libraries and Frameworks

  Over 80% of websites use a third-party JavaScript library or web framework for their client-side scripting. jQuery is by far the most popular library, used by over 75% of websites. Facebook created the React library for its website and later released it as open-source; other sites, including Twitter, now use it. Likewise, the Angular framework created by Google for its websites, including YouTube and Gmail, is now an open-source project used by others. In contrast, the term "Vanilla JS" has been coined for websites not using any libraries or frameworks, instead of relying entirely on standard JavaScript functionality.
- Other Usage

  The use of JavaScript has expanded beyond its web browser roots. JavaScript engines are now embedded in a variety of other software systems, both for server-side website deployments and non-browser applications. Initial attempts at

promoting server-side JavaScript usage were Netscape Enterprise Server and Microsoft's Internet Information Services, but they were small niches. Server-side usage eventually started to grow in the late 2000s, with the creation of Node.js and other approaches. Electron, Cordova, React Native, and other application frameworks have been used to create many applications with behavior implemented in JavaScript. Other non-browser applications include Adobe Acrobat support for scripting PDF documents and GNOME Shell extensions written in JavaScript. JavaScript has recently begun to appear in some embedded systems, usually by leveraging Node.js.

3. Features

The following features are common to all conforming ECMAScript implementations unless explicitly specified otherwise.

- Imperative and Structured

  JavaScript supports much of the structured programming syntax from C (e.g., if statements, while loops, switch statements, do-while loops, etc.). One partial exception is scoping: originally JavaScript only had function scoping with var; then block scoping was added in ECMAScript 2015 with the keywords let and const. Like C, JavaScript makes a distinction between expressions and statements. One syntactic difference from C is automatic semicolon insertion, which allows semicolons (which terminate statements) to be omitted.

- Weakly Typed

  JavaScript is weakly typed, which means certain types are implicitly cast depending on the operation used.

  - The binary `+` operator casts both operands to a string unless both operands are numbers. This is because the addition operator doubles as a concatenation operator.
  - The binary `-` operator always casts both operands to a number.

- Both unary operators (`+`, `-`) always cast the operand to a number.

  Values are cast to strings like the following:
- Strings are left as-is
- Numbers are converted to their string representation
- Arrays have their elements cast to strings after which they are joined by commas (`,`)
- Other objects are converted to the string `[object Object]` where `Object` is the name of the constructor of the object

Values are cast to numbers by casting to strings and then casting the strings to numbers. These processes can be modified by defining `toString` and `valueOf` functions on the prototype for string and number casting respectively.

JavaScript has received criticism for the way it implements these conversions as the complexity of the rules can be mistaken for inconsistency. For example, when adding a number to a string, the number will be cast to a string before performing concatenation, but when subtracting a number from a string, the string is cast to a number before performing subtraction.

JavaScript type conversions			
Left Operand	**Operator**	**Right Operand**	**Result**
`[]` (empty array)	`+`	`[]` (empty array)	`""` (empty string)
`[]` (empty array)	`+`	`{}` (empty object)	`"[object Object]"` (string)
`false` (boolean)	`+`	`[]` (empty array)	`"false"` (string)
`"123"` (string)	`+`	`1` (number)	`"1231"` (string)
`"123"` (string)	`-`	`1` (number)	`122` (number)
`"123"` (string)	`-`	`"abc"` (string)	`NaN` (number)

Table 1: JavaScript Type Conversions

Often also mentioned is `{} + []` resulting in `0` (number). This is misleading: the `{}` is interpreted as an empty code block instead of an empty object, and the empty array is cast to a number by the remaining unary `+` operator. If the user wraps the expression in parentheses `({} + [])` the curly brackets are interpreted as an empty object and the result of the expression is `"[object Object]"` as expected.

- Dynamic

  Typing

  JavaScript is dynamically typed like most other scripting languages. A type is associated with a value rather than an expression. For example, a variable initially bound to a number may be reassigned to a string. JavaScript supports various ways to test the type of objects, including duck typing.

  Run-time evaluation

  JavaScript includes an `eval` function that can execute statements provided as strings at run-time.

  Object-orientation (prototype-based)

  Prototypal inheritance in JavaScript is described by Douglas Crockford as:

  Users make prototype objects, and then ... make new instances. Objects are mutable in JavaScript, so the developer can augment the new instances, giving them new fields and methods. These can then act as prototypes for even newer objects. Developers don't need classes to make lots of similar objects... Objects inherit from objects. What could be more object-oriented than that?

  In JavaScript, an object is an associative array, augmented with a prototype (see below); each key provides the name for an object property, and there are two syntactical ways to specify such a name: dot notation (`obj. x = 10`) and bracket notation (`obj['x'] = 10`). A property may be added, rebound, or deleted at run-time. Most properties of an object (and any property that belongs to an object's prototype inheritance chain) can be enumerated using a `for...in` the loop.

  Prototypes

JavaScript uses prototypes whereas many other object-oriented languages use classes for inheritance. It is possible to simulate many class-based features with prototypes in JavaScript.

Functions as an object constructor

Functions double as object constructors, along with their typical role. Prefixing a function call with *new* will create an instance of a prototype, inheriting properties and methods from the constructor (including properties from the `Object` prototype). ECMAScript 5 offers the `Object.create` method, allowing explicit creation of an instance without automatically inheriting from the `Object` prototype (older environments can assign the prototype to `null`). The constructor's `prototype` property determines the object used for the new object's internal prototype. New methods can be added by modifying the prototype of the function used as a constructor. JavaScript's built-in constructors, such as `Array` or `Object`, also have prototypes that can be modified. While it is possible to modify the `Object` prototype, it is generally considered bad practice because most objects in JavaScript will inherit methods and properties from the `Object` prototype, and they may not expect the prototype to be modified.

Functions as methods

Unlike many object-oriented languages, there is no distinction between a function definition and a method definition. Rather, the distinction occurs during function calling; when a function is called as a method of an object, the function's local keyword is bound to that object for that invocation.

Functional

JavaScript functions are first-class; a function is considered to be an object. As such, a function may have properties and methods, such a `.call()` and `.bind()`. A nested function is a function defined within another function. It is created each time the outer function is invoked. In addition, each nested function forms a lexical closure: the lexical scope of the outer function (including any constant,

local variable, or argument value) becomes part of the internal state of each inner function object, even after the execution of the outer function concludes. JavaScript also supports anonymous functions.

Delegative

JavaScript supports implicit and explicit delegation.

Functions as roles (Traits and Mixins)

JavaScript natively supports various function-based implementations of Role patterns like Traits and Mixins. Such a function defines additional behavior by at least one method bound to `this` keyword within its `function` body. A Role then has to be delegated explicitly via `call` or `apply` to objects that need to feature additional behavior that is not shared via the prototype chain.

Object composition and inheritance

Whereas explicit function-based delegation does cover composition in JavaScript, implicit delegation already happens every time the prototype chain is walked to, e.g., find a method that might be related to but is not directly owned by an object. Once the method is found it gets called within this object's context. Thus inheritance in JavaScript is covered by a delegation automatism that is bound to the prototype property of constructor functions.

Miscellaneous

JS is a zero-index language.

Run-time environment

JavaScript typically relies on a run-time environment (e.g., a web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a web page DOM). These environments are single-threaded. JavaScript also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML `<script>` elements). This is not a language feature per se, but it is common in most JavaScript implementations. JavaScript processes messages from a queue one at a time. JavaScript calls a function associated with each new message, creating a call stack frame with the function's arguments and local

variables. The call stack shrinks and grows based on the function's needs. When the call stack is empty upon function completion, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output is performed using events and callback functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information.

Variadic functions

An indefinite number of parameters can be passed to a function. The function can access them through formal parameters and also through the local `arguments` object. Variadic functions can also be created by using the `bind` method.

Array and object literals

Like many scripting languages, arrays and objects (associative arrays in other languages) can each be created with a succinct shortcut syntax. These literals form the basis of the JSON data format.

Regular expressions

JavaScript also supports regular expressions like Perl, which provides a concise and powerful syntax for text manipulation that is more sophisticated than the built-in string functions.

Promises and Async/await

JavaScript supports promises and Async/await for handling asynchronous operations. A built-in Promise object provides functionality for handling promises and associating handlers with an asynchronous action's eventual result. Recently, combinator methods were introduced in the JavaScript specification, which allows developers to combine multiple JavaScript promises and do operations based on different scenarios. The methods introduced are Promise. race, Promise. all, Promise. all settled and promised. any. Async/await allows an asynchronous, non-blocking function to be structured in a way similar to an ordinary synchronous

function. Asynchronous, non-blocking code can be written, with minimal overhead, structured similarly to traditional synchronous, blocking code.

-React Native

React Native is an open-source UI software framework created by Meta Platforms, Inc. It is used to develop applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows, and UWP by enabling developers to use the React framework along with native platform capabilities. It is also being used to develop virtual reality applications at Oculus.

1. History

In 2012 Mark Zuckerberg commented, "The biggest mistake developer made as a company was betting too much on HTML as opposed to native". Using HTML5 for Facebook's mobile version resulted in an unstable application that retrieved data slowly. He promised Facebook would soon deliver a better mobile experience.

Inside Facebook, Jordan Walke found a way to generate UI elements for iOS from a background JavaScript thread, which became the basis for the React web framework. They decided to organize an internal Hackathon to perfect this prototype to be able to build native apps with this technology.

In 2015, after months of development, Facebook released the first version of the React JavaScript Configuration. During a technical talk, Christopher Chedeau explained that Facebook was already using React Native in production for their Group App and their Ads Manager App.

2. Implementation

The working principles of React Native are virtually identical to React except that React Native does not manipulate the DOM via the Virtual DOM. It runs in a background process (which interprets the JavaScript written by the developers) directly on the end device and communicates with the native platform via serialized data over an asynchronous and batched bridge.

React components wrap existing native code and interact with native APIs via React's declarative UI paradigm and JavaScript.

While React Native styling has a similar syntax to CSS, it does not use HTML or CSS. Instead, messages from the JavaScript thread are used to manipulate native views. With React Native developers have to write native code in the languages of the aimed platform such as Java or Kotlin for Android, Objective-C or Swift for iOS, and C++/WinRT or C# for Windows 10.

Microsoft builds and maintains React Native for Windows and React Native for macOS

3. Introduction

Many different kinds of people use React Native: from advanced iOS developers to React beginners, to people getting started programming for the first time in their career. These docs were written for all learners, no matter their experience level or background. To work with React Native, the user will need to have an understanding of JavaScript fundamentals. If users are new to JavaScript or need a refresher, user can dive in or brush up at Mozilla Developer Network.

- Create native apps for Android and iOS using React

  React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces. Use a little—or a lot. Users can use React Native today in user's existing Android and iOS projects or users can create a whole new app from scratch.

- Written in JavaScript—rendered with native code

  React primitives render to native platform UI, meaning the user's app uses the same native platform APIs other apps do. Many platforms, one React. Create platform-specific versions of components so a single codebase can share code across platforms. With React Native, one team can maintain two platforms and share a common technology—React.

- Native Development For Everyone

  React Native lets users create truly native apps and don't compromise users users' experiences. It provides a core set of platform-agnostic native

components like View, Text, and Image that map directly to the platform's native UI building blocks.

- Seamless Cross-Platform

React components wrap existing native code and interact with native APIs via React's declarative UI paradigm and JavaScript. This enables native app development for whole new teams of developers and can let existing native teams work much faster.

- Fast Refresh

See user changes as soon as user saves. With the power of JavaScript, React Native lets users iterate at lightning speed. No more waiting for native builds to finish. Save, see, repeat.

Advantages of React Native

The fact that React Native renders using its host platform's standard rendering APIs enables it to stand out from most existing methods of cross-platform application development, like Cordova or Ionic. Existing methods of writing mobile applications using combinations of JavaScript, HTML, and CSS typically render using web views. While this approach can work, it also comes with drawbacks, especially around performance. Additionally, they do not usually have access to the host platform's set of native UI elements. When these frameworks do try to mimic native UI elements, the results usually "feel" just a little off; reverse-engineering all the fine details of things like animations takes an enormous amount of effort, and they can quickly become out of date.

In contrast, React Native translates users' markup to real, native UI elements, leveraging existing means of rendering views on whatever platform users are working with. Additionally, React works separately from the main UI thread, so the user's application can maintain high performance without sacrificing capability. The update cycle in React Native is the same as in React: when props or states change, React Native re-renders the views. The major difference between React Native and React in the browser is that React Native does this by leveraging the UI libraries of its host platform, rather than using HTML and CSS markup.

For developers accustomed to working on the Web with React, this means the user can write mobile apps with the performance and look and feel of a native application while using familiar tools. React Native also represents an improvement over normal mobile development in two other areas: the developer experience and cross-platform development potential.

Developer Experience

If users have ever developed for mobile before, the user might be surprised by how easy React Native is to work with. The React Native team has baked strong developer tools and meaningful error messages into the framework, so working with robust tools is a natural part of the user's development experience.

For instance, because React Native is “just” JavaScript, users don’t need to rebuild a user application to see the user's changes reflected; instead, the user can hit Command+R to refresh the user's application just as the user would any other web page. All of those minutes spent waiting for the user's application to build can add up, and in contrast, React Native’s quick iteration cycle feels like a godsend.

Additionally, React Native lets users take advantage of intelligent debugging tools and error reporting. If users are comfortable with Chrome or Safari’s developer tools (Figure 1-1), they user will be happy to know that the user can use them for mobile development, as well. Likewise, users can use whatever text editor user prefers for JavaScript editing: React Native does not force the user to work in Xcode to develop for iOS or Android Studio for Android development.

Besides the day-to-day improvements to the user's development experience, React Native also has the potential to positively impact the user's product release cycle. For instance, Apple permits JavaScript-based changes to an app’s behavior to be loaded over the air with no additional review cycle necessary.

All of these small perks add up to saving users and users fellow developers time and energy, allowing the user to focus on the more interesting parts of user work and be more productive overall.

Code Reuse and Knowledge Sharing

Working with React Native can dramatically shrink the resources required to build mobile applications. Any developer who knows how to write React code can now target the Web, iOS, and Android, all with the same skillset. By removing the need to "silo" developers based on their target platform, React Native lets the user's team iterate more quickly, and share knowledge and resources more effectively.

Besides shared knowledge, much of the user's code can be shared, too. Not all the code users write will be cross-platform, and depending on what functionality you need on a specific platform, the user may occasionally need to dip into Objective-C or Java. (Happily, this isn't too bad, and we'll cover how so-called native modules work in Chapter 7.) But reusing code across platforms is surprisingly easy with React Native. For example, the Facebook Ads Manager application for Android shares 87% of its codebase with the iOS version, as noted in the React Europe 2015 keynote. The final application we'll look at in this book, a flashcard app, has total code reuse between Android and iOS. It's hard to beat that!

Risks and Drawbacks

As with anything, using React Native is not without its downsides, and whether or not React Native is a good fit for the user's team really depends on the users' individual situation.

The largest risk is probably React Native's maturity, as the project is still relatively young. iOS support was released in March 2015, and Android support was released in September 2015. The documentation certainly has room for improvement, and continues to evolve. Some features on iOS and Android still aren't supported, and the community is still discovering best practices.

Because React Native introduces another layer to user's projects, it can also make debugging hairier, especially at the intersection of React and the host platform. We'll cover debugging for React Native in more depth in Chapter 8, and try to address some of the most common issues.

React Native is still young, and the usual caveats that go along with working with new technologies apply here. Still, on the whole, I think we'll see that the benefits outweigh the risks.

Back-End Development

It is also known as server-side development. Its works in the background and the users don't aware of the content. Developers can say it is behind-the-scene activity. This happens or occurs when the user clicks or performs any action on a website created by Frontend technology. It mainly focuses on databases, backend logic, APIs, and Servers. It is called backend development. In the backend development, the developer can use various programming languages for example JavaScript, python, PHP, Ruby, etc., and their framework.

In our project, the developer mainly focuses on JavaScript so developers is going to use JavaScript as a backend language. A database provides a place for user's web application's data to be stored.

A. Node.js

It is an open-source and backend JavaScript runtime environment. This is created in the chrome V8 engine. The use of Node.js is to run or execute the JavaScript code outside the web browser. Developers can aware of one thing developers use NodeJs as the backend but it helps to run the frontend as well as backend code. It is not a programming language. It is based on JavaScript. This technology is mostly used by those developers who use or prefer JavaScript as a backend language.

Benefits of Node.js for app development:

In a world where technology is evolving at a tremendous pace, there is an increase in demand for apps that interact with users in real time. Today, every app developer see has at least one real-time feature integrated into it such as messages or notifications. Real-time apps have many different functions in a short period and appear to happen in real time. Node.js is best for developing such apps where speed is the most important aspect along with app scalability. It is the technology of choice for real-time apps that are data-

intensive and connected to IoT devices. Let's explore some of the top benefits of Node.js for real-time application development.

- Event-driven server

Real-time applications have to interact with thousands of real-time users leveraging the app. Node.js helps in supporting the response based on the event-based server that aids in the non-blocking functionality of the app.

- Data sync

Node.js enables developers to effectively use the non-blocking I/O functionality. Transmission of data and syncing between the server and the client is faster by using Node.js.

- Scalable

As Node.js is based on JavaScript programming language, it can scale up applications faster. Applications that are Node.js powered and based on the single-threaded model, as well as the event loop, can effectively manage client scalability requests.

- Shareability

Node.js facilitates microservice architecture. It is a real-time programming language that enables developers to leverage library code packages more than once and share them across diverse projects which saves development time and increases productivity.

- SEO friendliness

SEO is imperative for any business to gain online visibility and customer reach. Node.js enhances the app engagement and offers more visibility to the app. Apps powered by Node.js not only are high on performance but also offer a great user experience which is imperative for high SEO rankings.

- Proxy server

It is a great programming language for projects where intermediary admins are required. A developer needs to use a twenty-line code to leverage Node.js as a proxy server. This makes the app most suited for streaming data from a range of sources.

B. Express.js

Express.js or developer can say Express, is a backend Node.js framework. It is free and open-source software. It is used for creating web applications and APIs (Application Programming Interface). It is one of the most used backend frameworks by the developer. It is based on JavaScript. The primary use of Express is creating and building APIs means it handles or accept the request and sent the appropriate response to the user. Express made it simpler and easier to write the back-end code and implement it in a structured format. Express helped us in designing the web applications and APIs required for our project like it.

Supports many middlewares which makes the code shorter and easier to write. Asynchronous programming and Single-threaded architecture are the biggest advantages of using Express in our application. For our application robust API Created a new folder to start our express project and the steps for it is, that developer had to add a command in the command prompt to initialize the package.json file. After that, a developer had to accept the default settings and continue. npm init is the command to start.

C. GitHub

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration. GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technically savvy and use of the command line. GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects –like writing books. Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects. As a company, GitHub makes money by selling hosted private code repositories, as well as other business-focused plans that make it easier for organizations to manage team members and security. Developers utilize Github extensively at Kinsta to manage and develop internal projects.

3.4 Functional Requirements

Application requirements Typically, a React Native application has two types, USER and Provider. Users can do login and then see all the services with all the categories. Also, see all the professionals where they can book the professionals as per their requirements. They can also see all the bookings and also cancel the bookings. They can do the reviews and ratings to the professionals. Also, they can do chatting and call. Based on that, a list of user stories is shown below to illustrate some of the required functionalities for this application:

- As a user, user can do Login
- As a user, user can see all the Services and categories
- As a user, user can see all the professionals with all the details like reviews and ratings
- As a user, the user can book the professionals as per their requirements.
- As a user, the user can see all the upcoming, past, and cancel bookings.
- As a user, the user can cancel the bookings.
- As a user, the user can chat with the professional and make a call to the professional.
- As a user, users can give reviews and ratings to the professionals.
- As a user, user can add to favorite to the professionals
- As a provider, a provider can get all the requests like today's bookings, and upcoming bookings.
- As a provider, a provider can see all the reviews and ratings given by the user.
- As a provider, a provider can chat and make a call with the user.
- As a provider, a provider can see all the details of bookings like id, date, time, and user information.

3.5 Non-Functional Requirement

Without correctly defining them, the users developer may meet all of the users functional requirements but still deliver an unsatisfactory product. However, if users have not specifically defined the NFRs, they cannot be held responsible for that.

Many e-commerce projects in the past have failed or become much more costly than expected due to a lack of well-defined NFRs. Without good NFRs users could end up with an e-commerce platform that meets its functional requirements but is slow, insecure, hard to manage, costly to maintain, of poor quality, and provides a poor experience to users.

There are many other types of NFRs, all of which can have a critical impact on the performance, scalability, and usability of user's e-commerce platforms. These include:

- **Security** – it is important to specify the level of security that should be met such as the OWASP top 10.
- **Privacy** – meeting basic requirements for GDPR.
- **Scalability and performance** – ensuring that the system can scale to meet expected traffic and order volume at normal and peak times.
- **Speed of key user journeys** – defining how long each step with key user journeys will take.
- **Speed of web services** – defining how long web services will take to respond.
- **Accessibility** – ensuring that the platform meets the basic accessibility standards throughout.
- **Documentation** – ensuring that the platform is sufficiently documented.
- **Quality** – even the best e-commerce platforms can be used badly so a user should insist that code is developed to a good quality standard.
- **Extensibility** – ensuring that the platform is extended in such a way as to make future development feasible.
- **Data integrity and retention** – defining how long data should be stored and how the integrity of data is maintained.
- **Testing** – defining how unit testing will be built into the solution.
- **Compatibility** – ensuring that the platform can be easily integrated with 3rd party systems.
- **Search** – defining how quickly the system will return search results.
- **Availability** – defining the agreed uptime of the platform under normal conditions.

- **Infrastructure** – defining the infrastructure performance thresholds (CPU and memory usage).
- Although this list is not exhaustive, it demonstrates the wide range of NFRs that can – and should – be defined when building out the list of requirements for, and specifications of, an e-commerce platform.

Why NFRs are so important

- NFRs essentially define the quality of users' platforms. A functional requirement will define its behavior, but the NFR will define how well it will exhibit that behavior.

A well-defined set of NFRs will have the following benefits:

- Better code quality
- Better security
- The reduced total cost of ownership
- Better user experience
- Fewer bugs
- Better speed and performance
- Better uptime

3.6 Feasibility Study

A feasibility study is carried out when there is a complex problem or opportunity. It is considered the primary investigation which emphasizes on “Look before User Loop” approach to any project. A Feasibility study is undertaken to determine the possibility of either improving the existing system or developing a completely new system.

Developers are going to develop the new system which is feasible as our application is very user-friendly and easy to understand.

Economic Feasibility

For proving that system developed is economical, the economical feasibility study takes place to check the cost of developing a system against the benefits that it provides. If the cost is less and the benefits are more then a developer can define our system to be

economically developed. User saves time in searching for a particular product to be purchased with simply a few clicks. The registration process is speedier than registering manually. The saving of papers as all data are stored computerized. The record is free of human errors as there is less chance of mistakes. The above benefits are in terms of saving time, minimizing errors, and providing efficiency in work done. In terms of economical feasibility, our application is very reasonable in cost. So the application is economically feasible.

Technical Feasibility

In this type of study, the current technology used in an organization is checked such as the existing software, hardware, and personnel staff to determine whether it will work for the proposed system or completely new ones are to be used. The technology that was important in developing a new system such as Development tools, and back-end database systems were available from within the organization. The proposed system is capable of adding, changing, and enhancing functionality, features, etc. The proposed system is capable of handling large storage of data. The back-end and front-end technology have greater importance for providing accurate, error-free, frequencies of data to be used.
Our project is technically feasible in terms of current technology. Our project will provide the latest platform like android technology.

Operational Feasibility

The operational feasibility is concerned with the operability of the system after it has been installed. That is, some programmers may not like changes in their routine method of work or fear that they will lose their peer group. The following areas will have the operational feasibility in the proposed project

3.7 Use Case Diagrams:

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

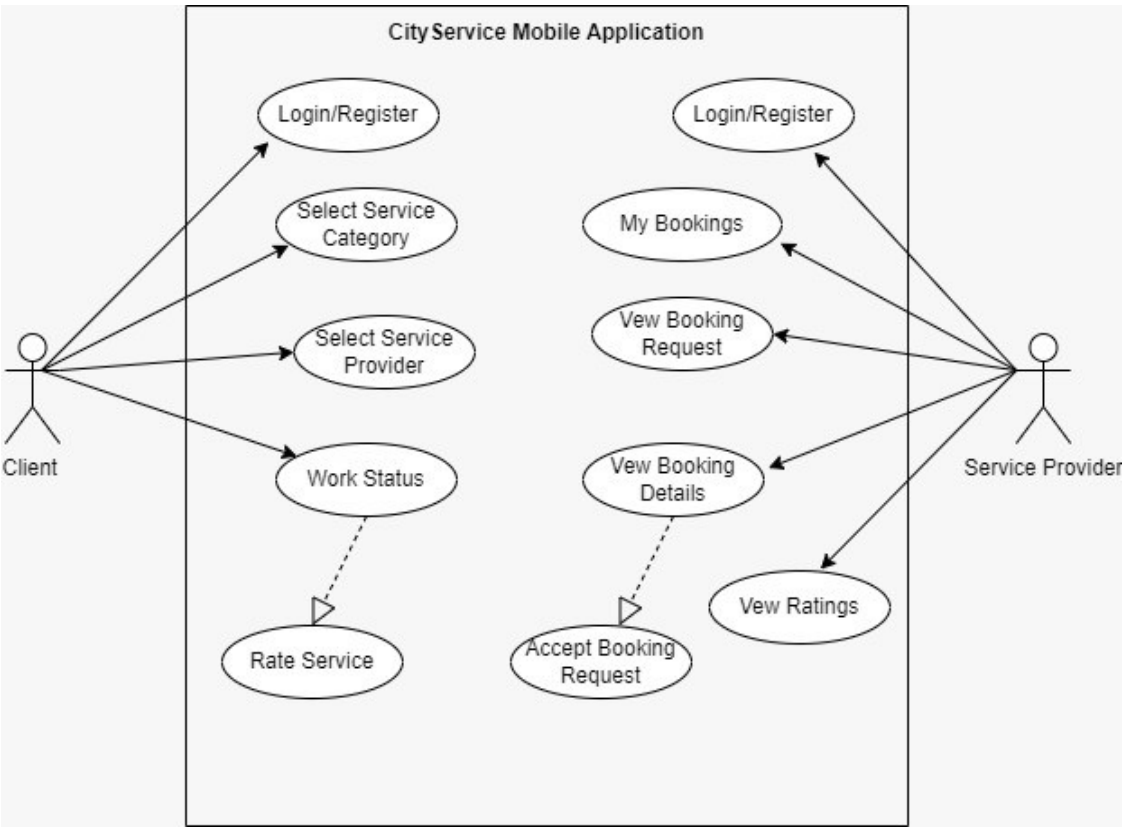


Figure 2: Use a Case Diagram

3.8 Use Case Specification:

The Brief Description of the use case diagram will be as follows:

Client:

- Login/Register
- Get All the Services
- Get All the Categories
- Get All the Professionals
- Select the Professionals
- Book the Professionals
- Booking Confirm
- Add Reviews and Ratings

Provider:

- Login
- Get All the Requests of Bookings
- Get All the reviews and ratings
- Chat and Calling Function
- Get All the information for booking

4. DESIGN

Before designing any system, the problem is identified, specified, and defined in detail. After identifying and specifying the problem, an analysis of the problem is done. Then according to the requirements, the basic architecture of the system is chosen or defined. However, before design, design goals should be defined and prioritized.

4.1 Design Goals

Certain design goals are identified and prioritized that may help or drive the design of the system. The design architecture of the system should achieve maximum design goals. Design goals identifies and prioritize to design this system have been listed below:-

- The most important design goals are to achieve all the functional requirements of the system.
- Flexibility to add or extend the design.
- The system should be modifiable or loosely coupled.
- The system must be secure.

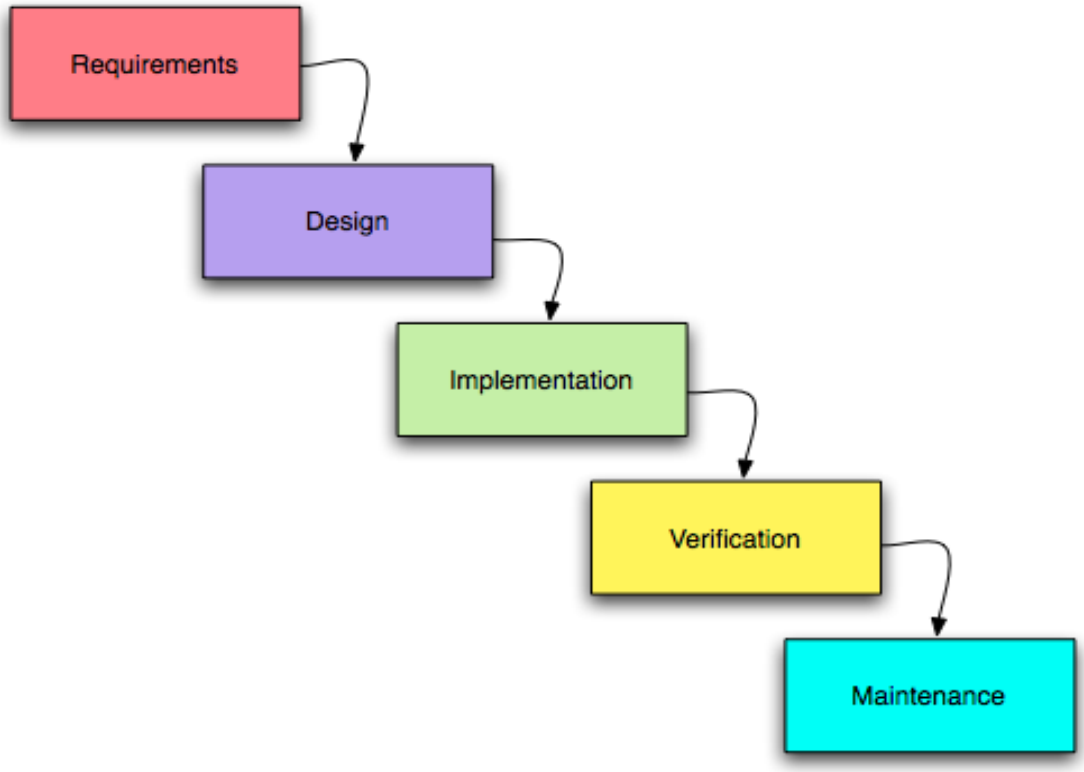
4.2 Design Strategy

Our project is developed using a specific software development lifecycle. The software development approach is best suited for the project depending on the requirement and other factors. A process model is a development strategy that is used to achieve a goal that satisfies the requirements abiding by the constraints.

Analysis: The requirement analysis aims to understand the exact requirements of the customer and to document them properly.

Requirement gathering and analysis: This activity consists of first gathering the requirement and then analyzing the gathered requirement.

Requirement Specification: The customer requirement identified during the requirement gathering and analysis activity are organized into a software requirements specification (SRS).



Design: The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming languages. In technical, during the design phase, the software architecture is derived from the SRS document.

Traditional Design Approach: The traditional design technique is based on the data flow-oriented design approach. While using this technique the design phase consists of two important activities: first, a structured analysis of the requirement specification is carried out where the detailed structure of the problem is examined.

Object-oriented design approach: Object-oriented design approach (OOD) is a relatively new technique. In this technique, various objects that occur in the problem domain and the solution domain are first identified and the different relationships that exist among these objects are identified.

Code: The purpose of the coding of software development is to translate the software design into source code. The coding phase is also sometimes called the implementation

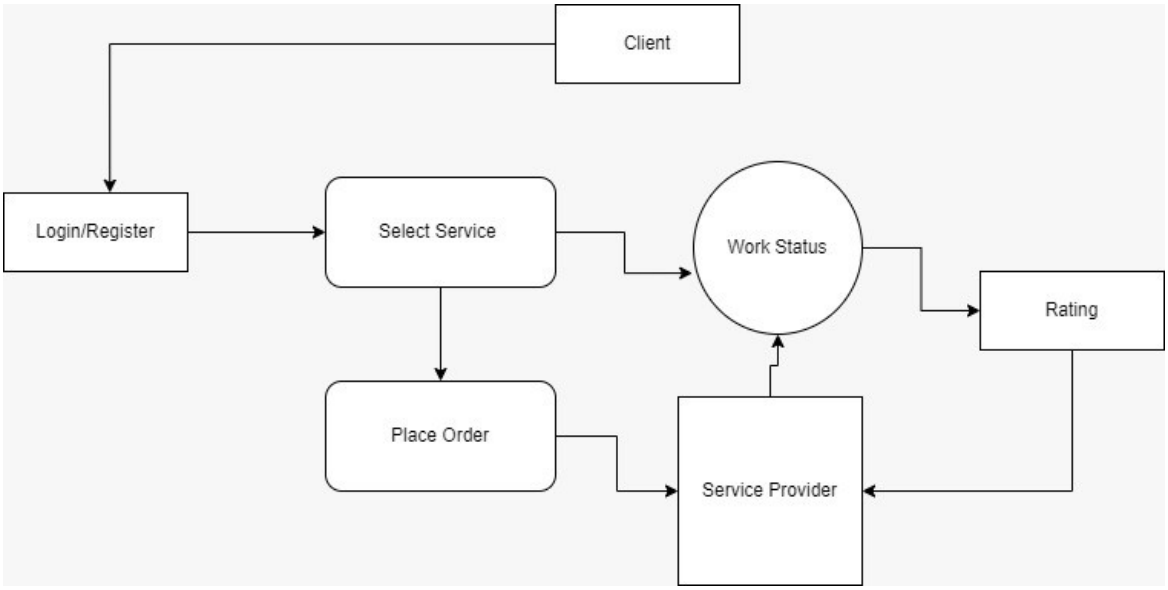
phase since the design is implemented into a workable solution in this phase. Each component of the design is implemented as a program module. The end product of this phase is a set of program modules that have been individually tested. To enable the engineers to write a good quality program, every software development organization normally formulates the coding standards that suit itself. A coding standard addresses issues such as the standard ways of laying out the program code, the template for laying out the function and module headers, commenting guidelines, variables, and function naming conventions, the maximum number of source lines permitted in each module, etc.

Testing: System testing is normally carried out in a planned manner according to a system test plan document.

Maintenance: Maintenance of a typical software product requires much more effort than the effort necessary to develop the product itself.

4.3 Module Diagram

The purpose of a component diagram is to show the relationship between different components in a system. For UML 2.0, the term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.



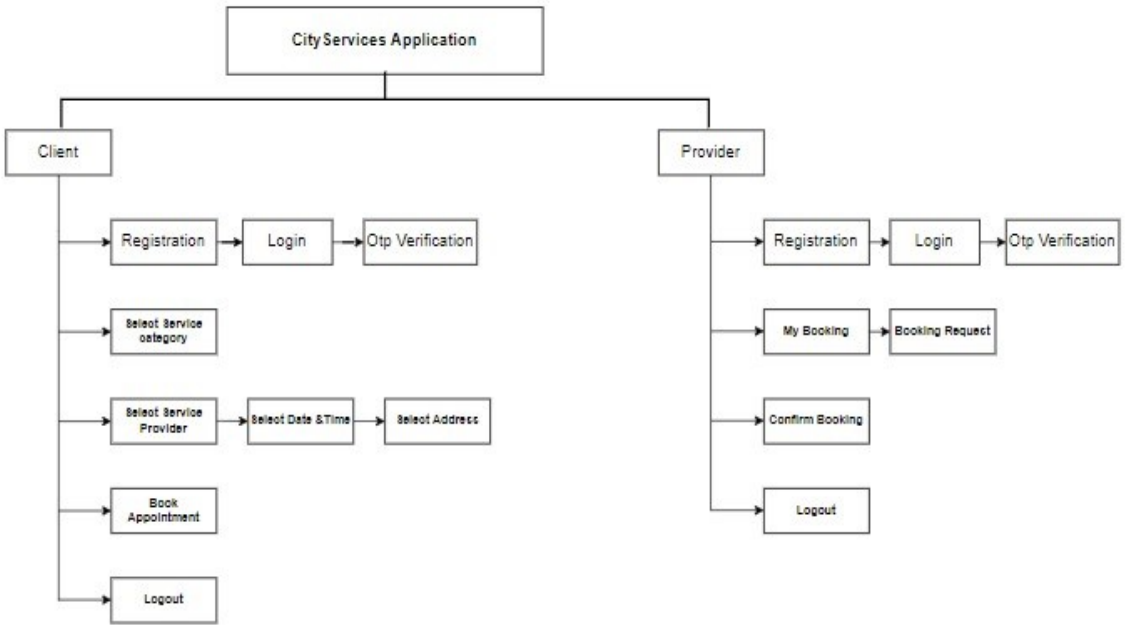
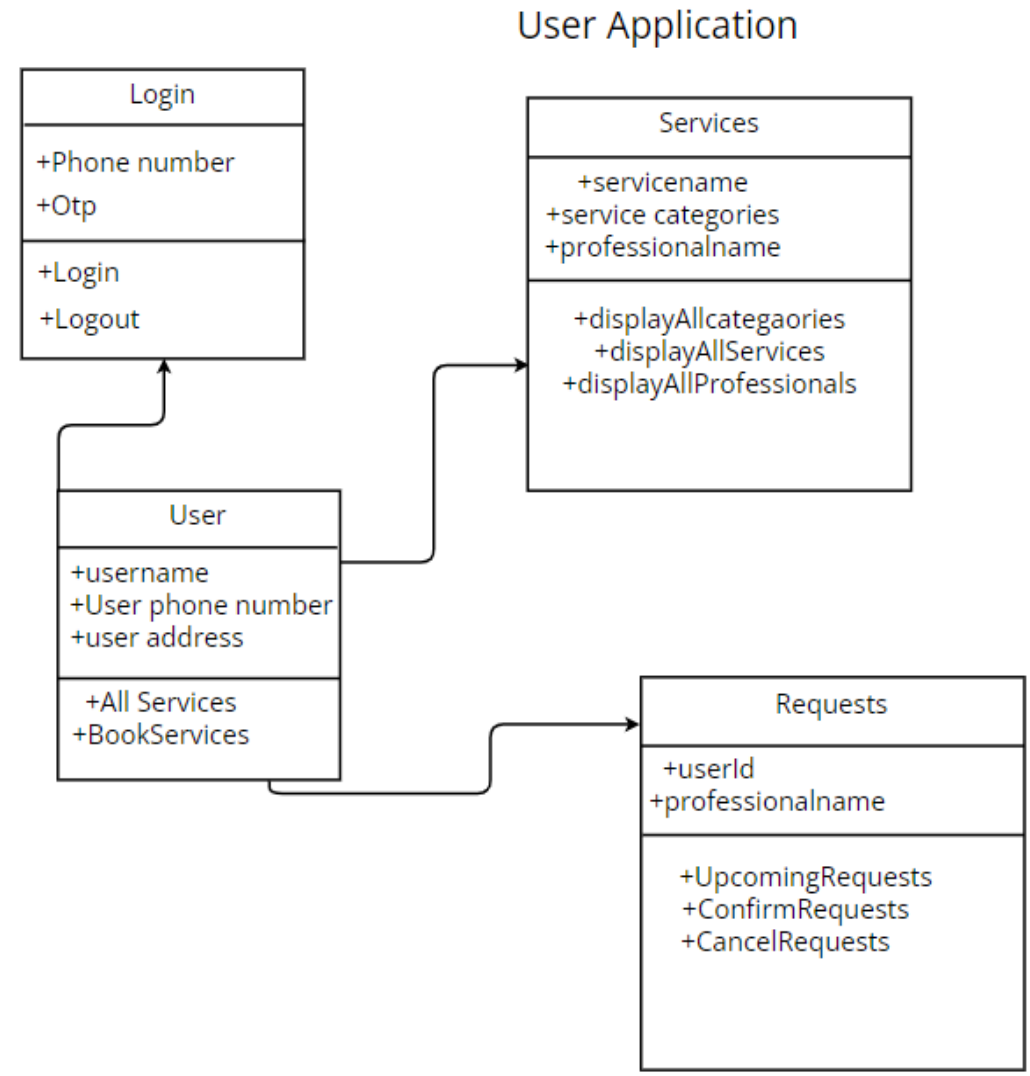


Figure 3: Architecture Diagram

4.4 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The class diagram is the main building block in object-oriented modeling. They are being used both for general conceptual modeling of the systematics of the application and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes that contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class, and
- The bottom part gives the methods or operations the class can take or undertake.


User Application
Login
+Phone number
+Otp
+Login
+Logout
Services
+servicename
+service categories
+professionalname
+displayAllcategaories
+displayAllServices
+displayAllProfessionals
User
+username
+User phone number
+user address
+All Services
+BookServices
Requests
+userId
+professionalname
+UpcomingRequests
+ConfirmRequests
+CancelRequests

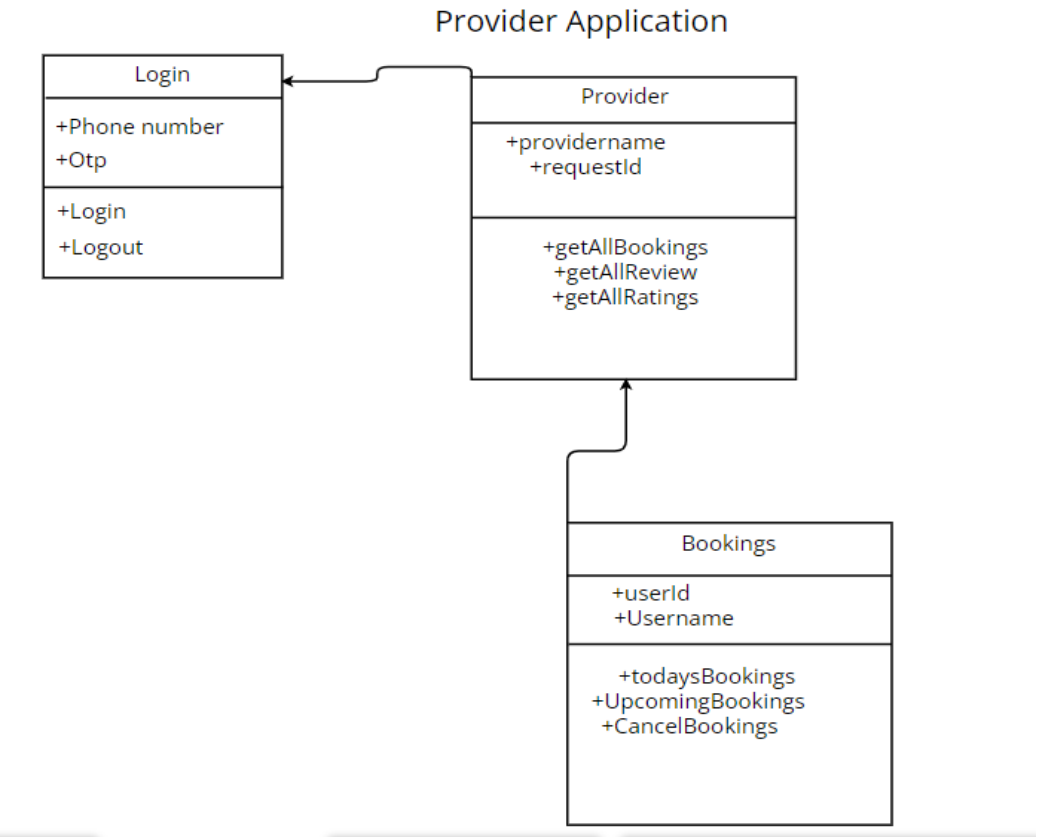


Figure 4: Class Diagram

4.5 Sequence Diagram

- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- Graphically, a sequence diagram is a table that shows objects arranged along the x-axis and messages, ordered in increasing time, along the y-axis

Contents:

Sequence diagrams commonly contain the following:

- Objects
- Links
- Messages

Like all other diagrams, sequence diagrams may contain notes and constrains

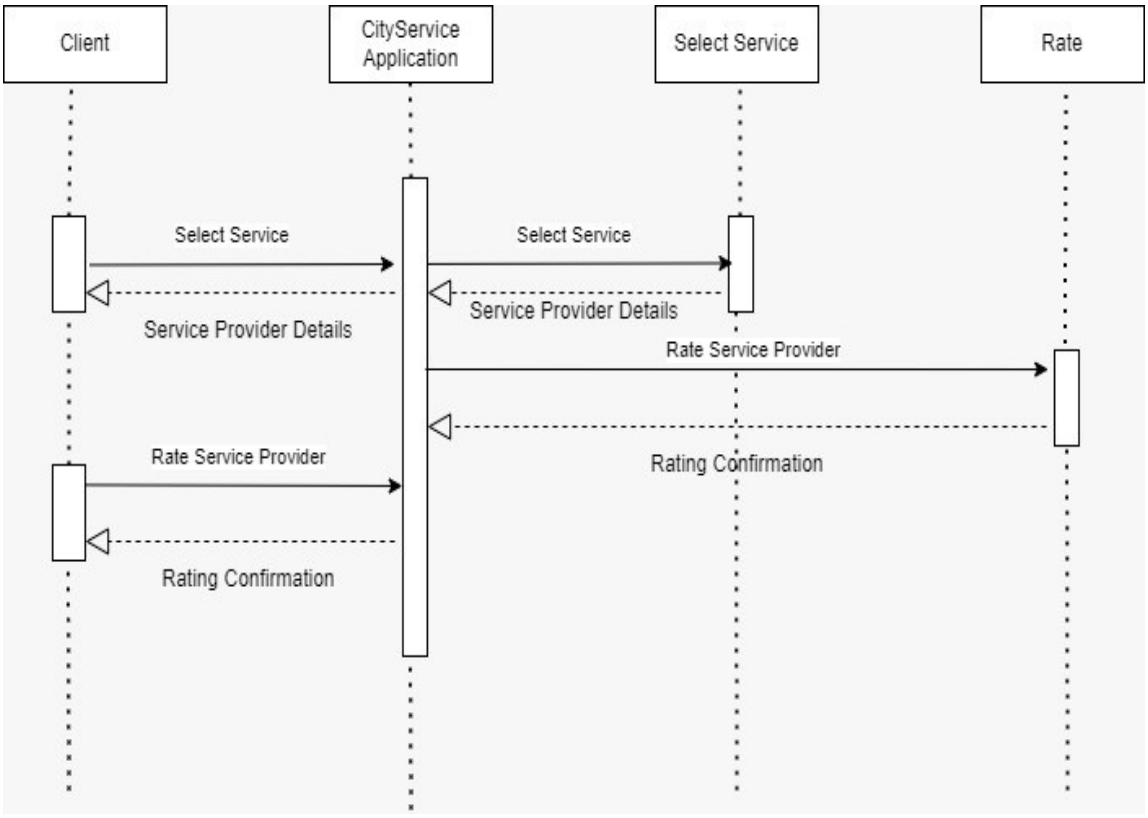


Figure 5: Sequence Diagram

4.6 Activity Diagram

An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine. Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in the state of the system or the return of a value.

Contents

Activity diagrams commonly contain

- Activity states and action states
- Transitions
- Objects

Like all other diagrams, activity diagrams may contain notes and constraints.

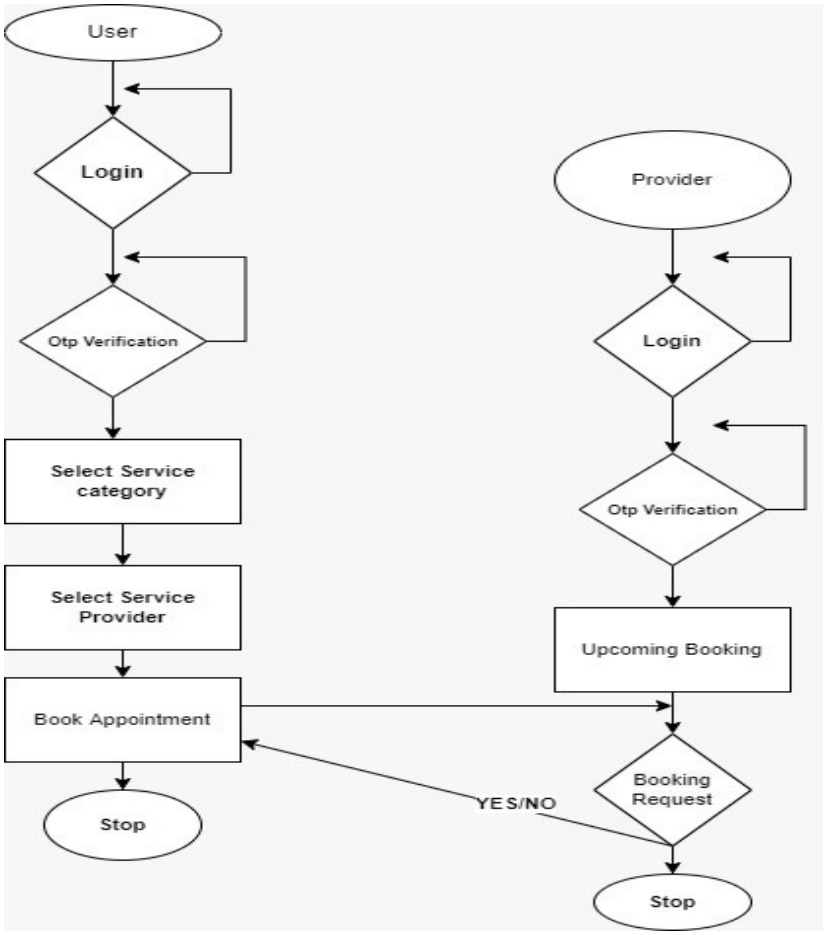


Figure 6: Activity Diagram

5. IMPLEMENTATION

5.1 Implementation Strategy

The implementation is the final and important phase. It involves User training, system testing, and successful running of the developed system. The users test the developed system when changes are made according to their needs. The testing phase involves the testing of the developed system using various kinds of data. Elaborate testing of data is prepared and the system is tested using the test data.

The Implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resource and additional equipment has to be acquired to implement the new system. In the network backup system, no additional resources are needed. Implementation is the stage where theoretical design is turned into a working system.

5.2 Software Requirements

- Operating System:
- Microsoft Windows XP and above
- Smart Phone: Oreo, 2 GB RAM, 8 GB Storage, and Above

Visual Studio Code:

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Android Studio:

Android Studio provides a unified environment where users can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules

allow the user to divide the user's project into units of functionality that the user can independently build, test, and debug.

Nodejs:

Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional websites and back-end API services but was designed with real-time, push-based architectures in mind.

Node Package Manager:

Npm is the package manager for the Node JavaScript platform. It puts modules in place so that nodes can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

React Native:

React Native is an open-source JavaScript framework, designed for building apps on multiple platforms like iOS, Android, and also web applications, utilizing the very same code base. It is based on React, and it brings all its glory to mobile app development.

5.3 Hardware requirements

1. External Hard Drive

Given that most things are done online these days it’s important to have some kind of hard drive to store information in. External hard drives keep users' business organized and user information in a secure place. Users must invest in this type of equipment to maintain a successful running business online and physically. This is especially beneficial for those who are afraid or don‟t like to store information in the Cloud or directly on their computer’s hard drive. These are also used as a backup for any accidents that may occur. Users want to make sure user invest in security for theft cases.

2. Desktop/Laptop Computer

Everyone has one these days and it is crucial for any business that wants to grow and thrive to have a designated computer system. Make sure to invest in the equipment that will last and that is reputable for businesses. Finding a computer or laptop won’t be hard since they are sold everywhere, along with much technical equipment.

3. Network Server

Network servers are highly important for fast and easy speeds and databases. Users will have access to more storage capacity and higher security. For the user's business, the user wants to stay away from using the user's computer's network server and get one that is specifically dedicated to the user's business only. There are many options out there for the user to choose from. Use the number of space users need, security, and options users want for backups as criteria for user decisions. Make sure to research thoroughly and work with specialists to help the user choose one that fits the user.

4. Mobile device

A mobile device is almost always necessary when running a business. There will be moments when the user needs to make professional calls and the user don‟t want to use the user's phone. Having a device that is specifically for the user's business helps with organization and professionality. From tablets to cellphones, it's good to have a mobile device that users can take anywhere and handle business situations easily and conveniently.

5.4 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. These two diagrams are special diagrams used to focus on software components and hardware components. Deployment diagrams are used by the system engineers.

A diagram that shows the configuration of run-time processing nodes and the components that live on them; a deployment diagram addresses the static deployment view of the system.

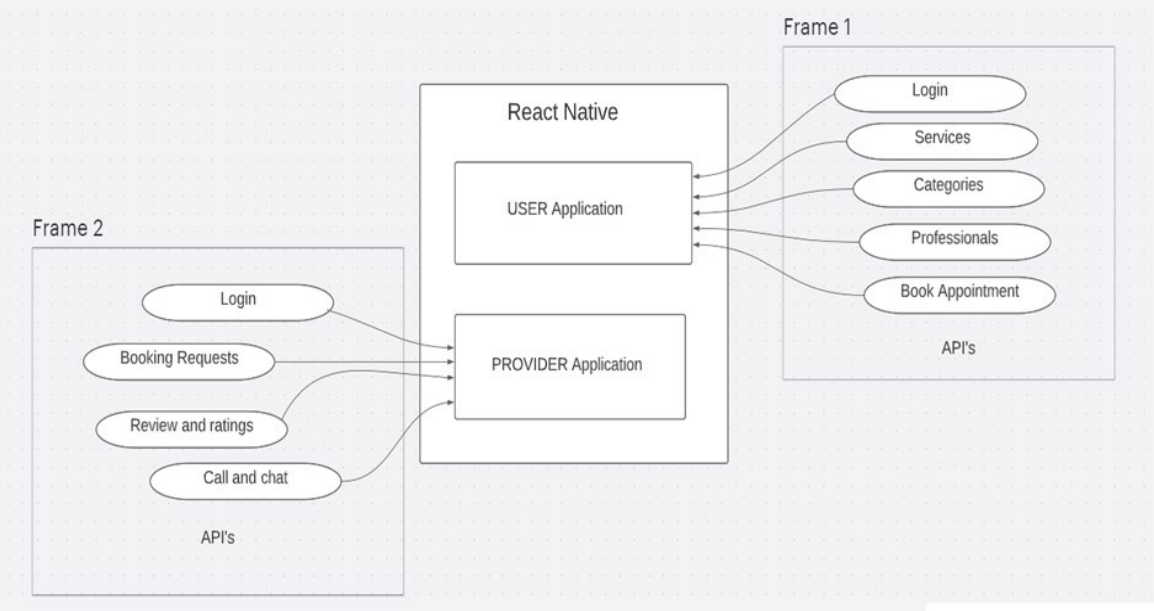
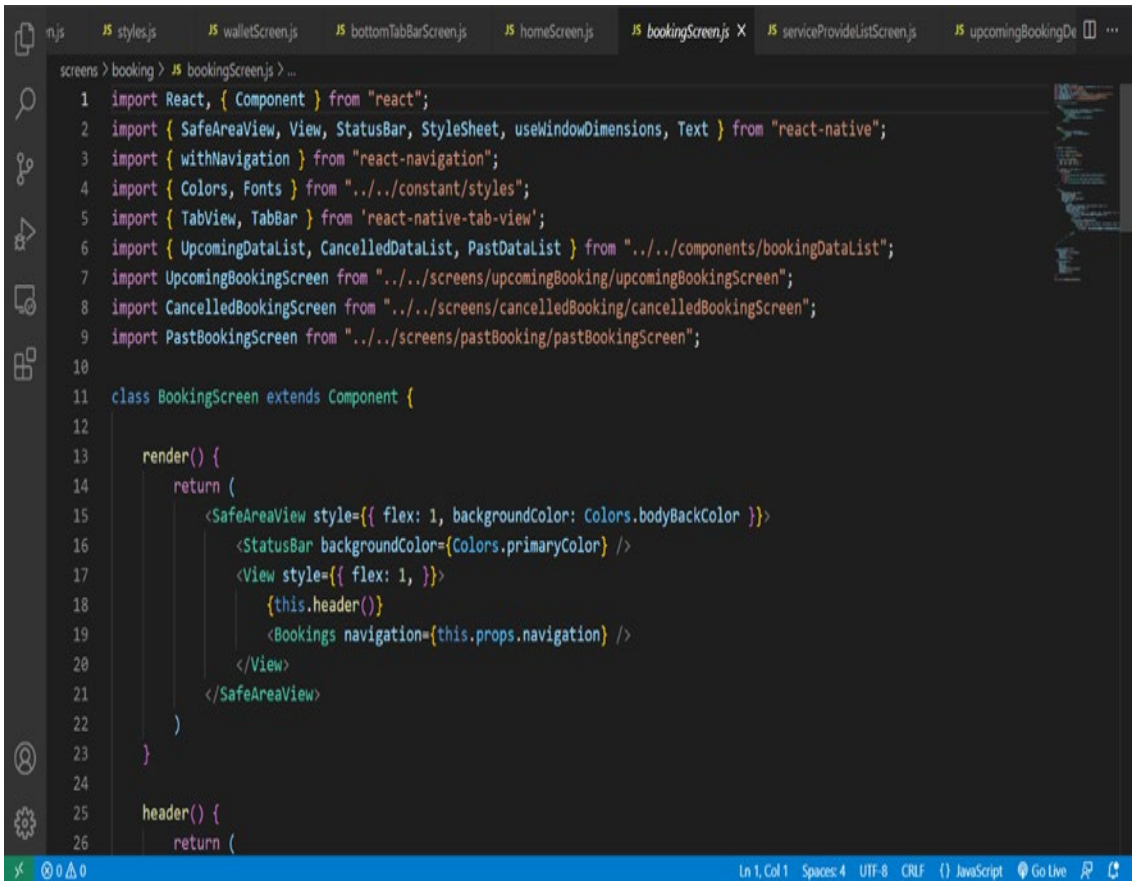


Figure 7: Deployment Diagram

5.5 Implementation Level Details (Algorithms)

Application development This section is dedicated to demonstrating the functionalities development process from the front-end of the React Native application. Due to the limited scope of this thesis, it is not able to mention all the files or describe every step in the project in detail, but it aims to discuss precisely all fundamental parts that are needed to implement the React Native application. Basic concepts of any third-party libraries or modules are also explained along the way. The project structure is divided into 2 parts USER Application and Provider Application which contain the source code and front-end and API respectively.

USER Application:



5.5.1 App.js file in user Application

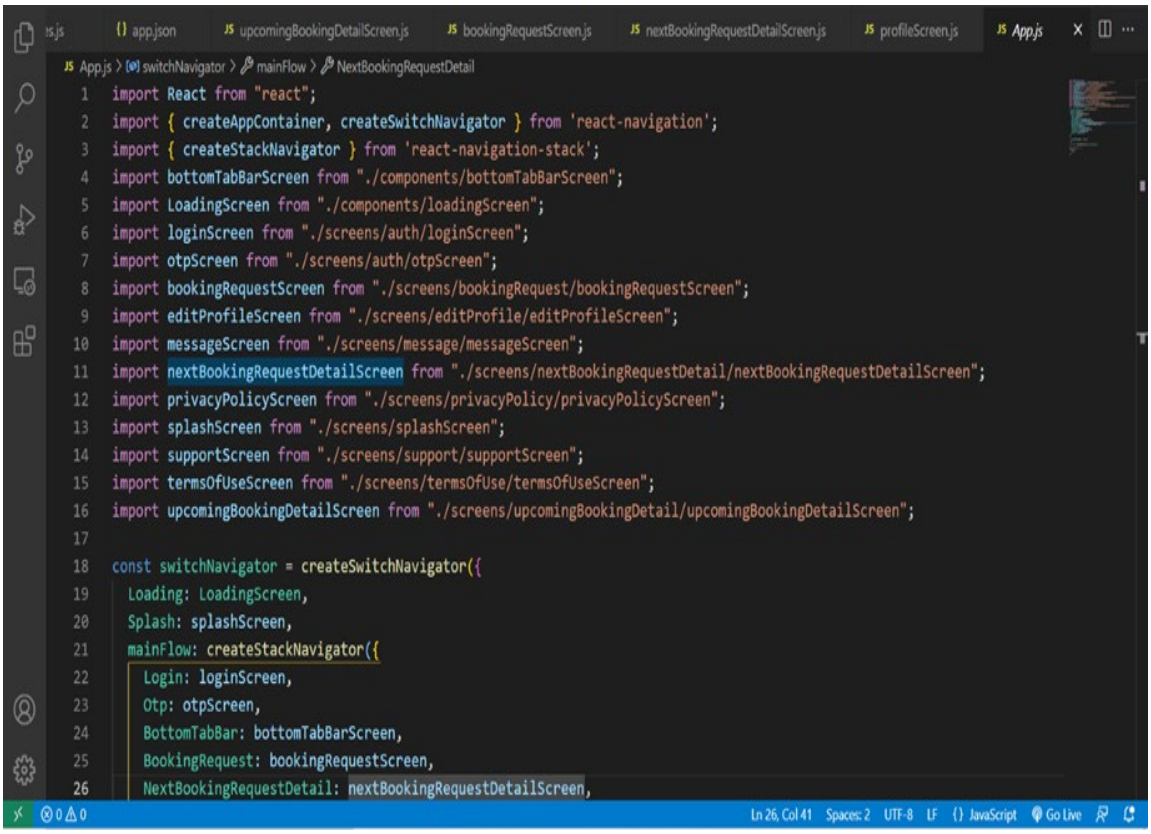
App.js is the main file in our USER Application. When the developer runs our application then this file will execute and all the required scripts will be generated with the help of npm. Npm is the package manager for the Node JavaScript platform. It puts modules in place so that nodes can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs. The developer has added all the screen js files that the developer has created in it and provided the navigation to all

5.5.2 Booking.js file in a user application

Due to the limited scope of this thesis, it is not able to mention all the files so specify only one file. This is the booking Screen js file. Developers use all the react-native style components, all the functionalities, features, and all the other things that developers are

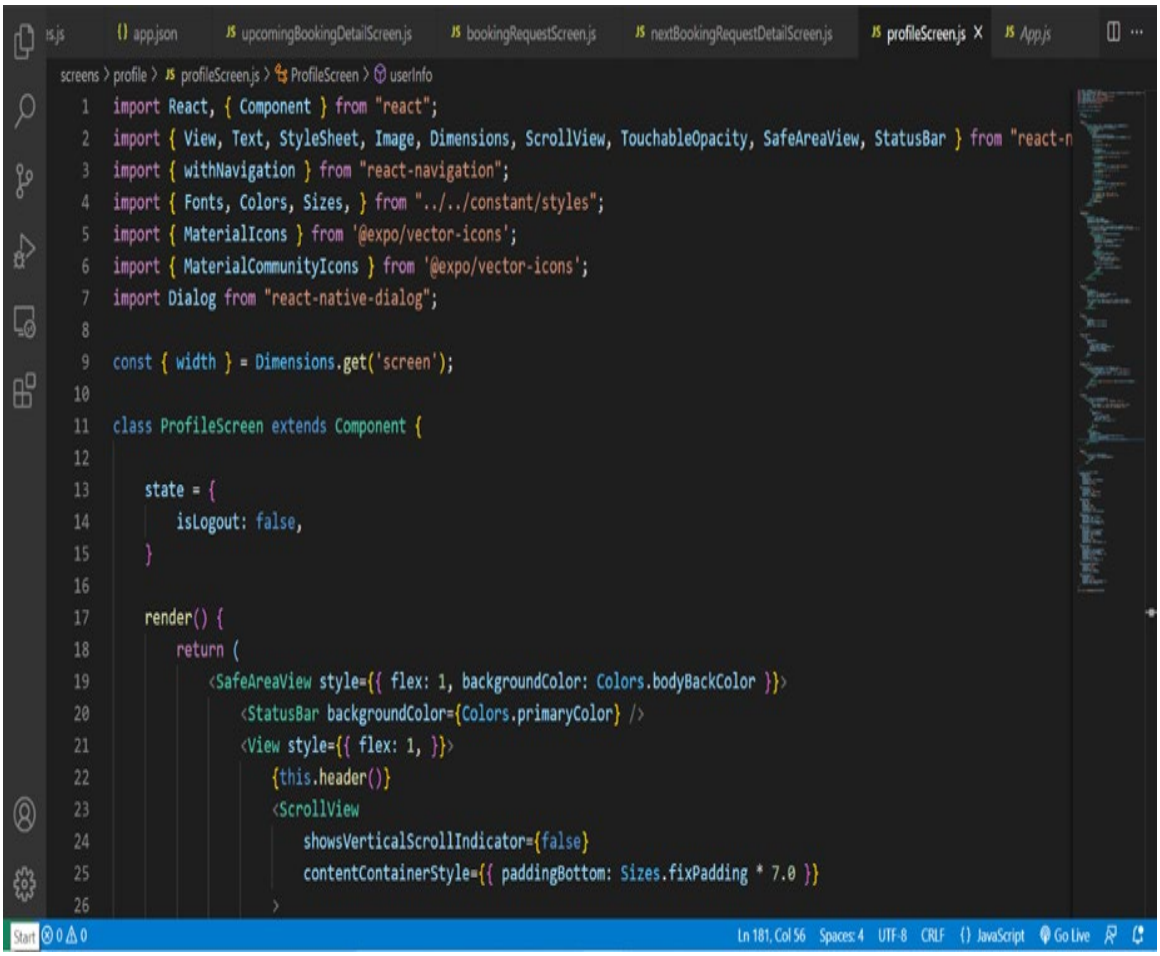
required to do. The developer uses the React-Native material.UI framework to design the UI.

PROVIDER Application:



5.5.3 App.js file in provider Application

App.js is the main file in our PROVIDER Application. When the developer runs our application then this file will execute and all the required scripts will be generated with the help of npm. Npm is the package manager for the Node JavaScript platform. It puts modules in place so that nodes can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs. The developer has added all the screen js files that the developer has created in it and provided the navigation to all.



5.5.4 Profile.js file in Provider Application

Due to the limited scope of this thesis, it is not able to mention all the files so specify only one file. This is the Profile Screen js file. Developers use all the react-native style components, all the functionalities, features, and all the other things that developers are required to do. The developer uses the React-Native material.UI framework to design the UI.

To RUN Both the Applications USER Application and Provider Application follow the given Statements:

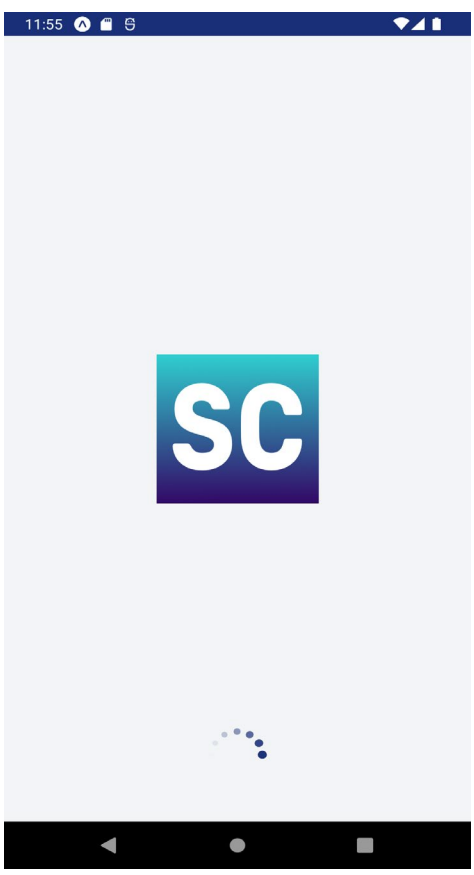
1. Firstly, Open the project folder in the **Visual Studio code.**
2. Then Open the **Terminal**.
3. Run the Android emulator (the user will get the android emulator in android studio).

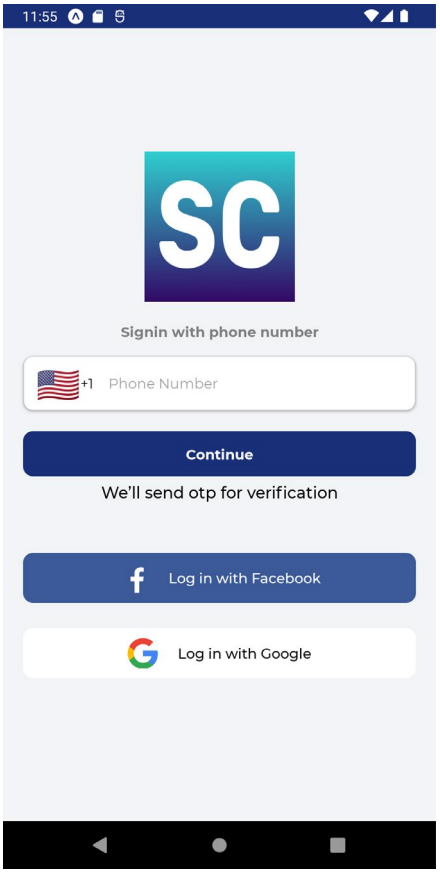
4. Type expo **start**
5. It will open a window in chrome with the port no: **exp://192.168.1.3:19000**
6. Then their user will get the option that runs in the android emulator.
7. Once the user clicks on **run android emulator** then the user's project starts building the javascript
8. Finally, the user's Application is run successfully.

Note: (If the user gets some error then simply just delete the **node_module** and p**ackage.lock.Jason** and Do **npm Install** in terminal)

5.6 Implementation of Various Dashboard

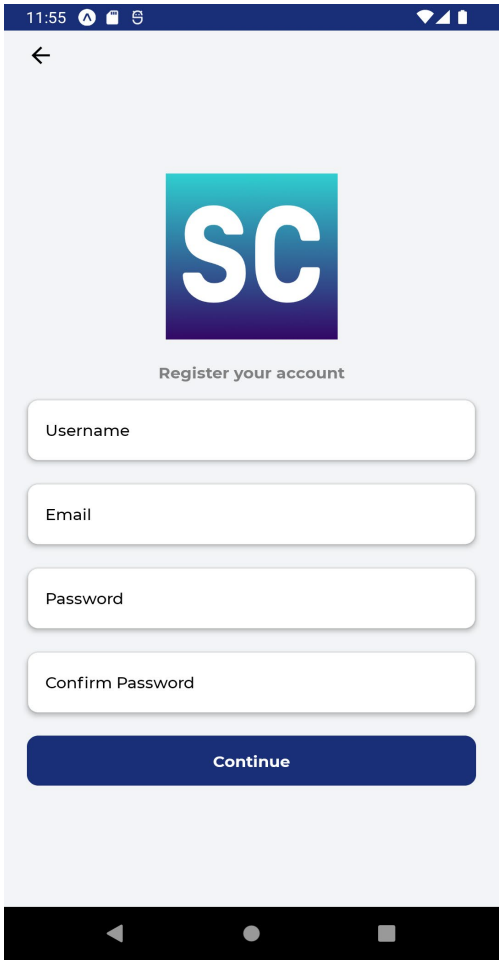
USER Application:

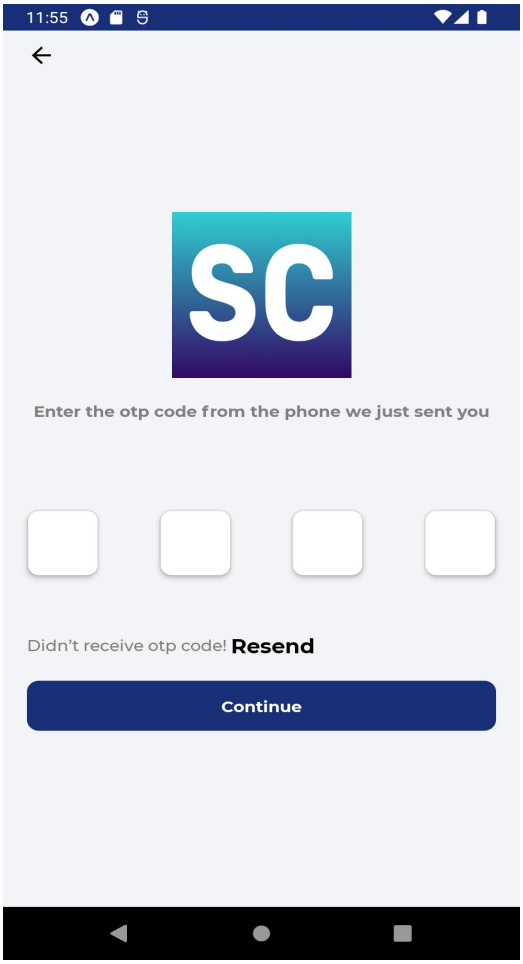




Screenshot 1: Indicates the Splash Screen of the user application.

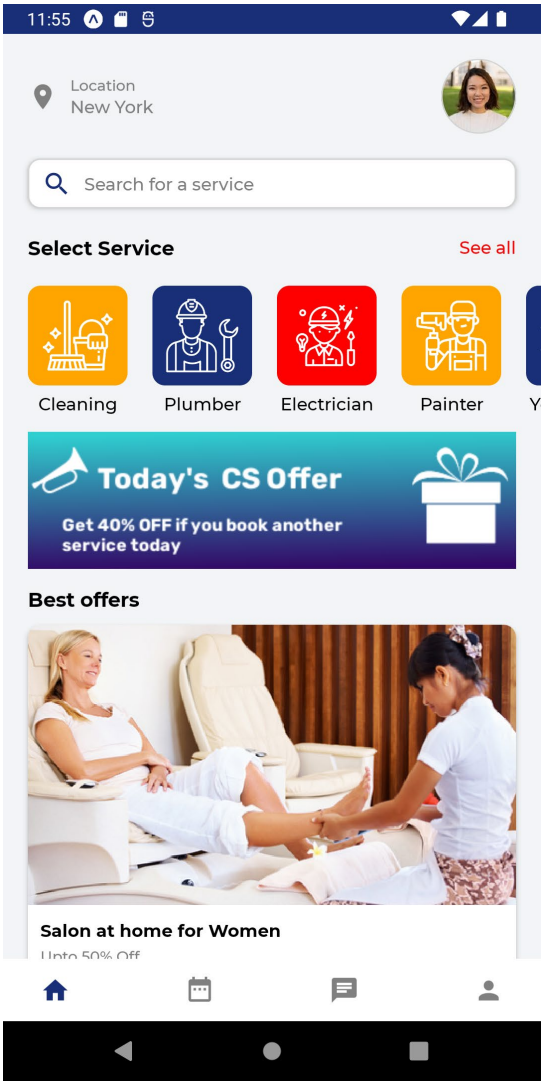
Screenshot 2: Indicates the Sign In Screen Of the user Application.

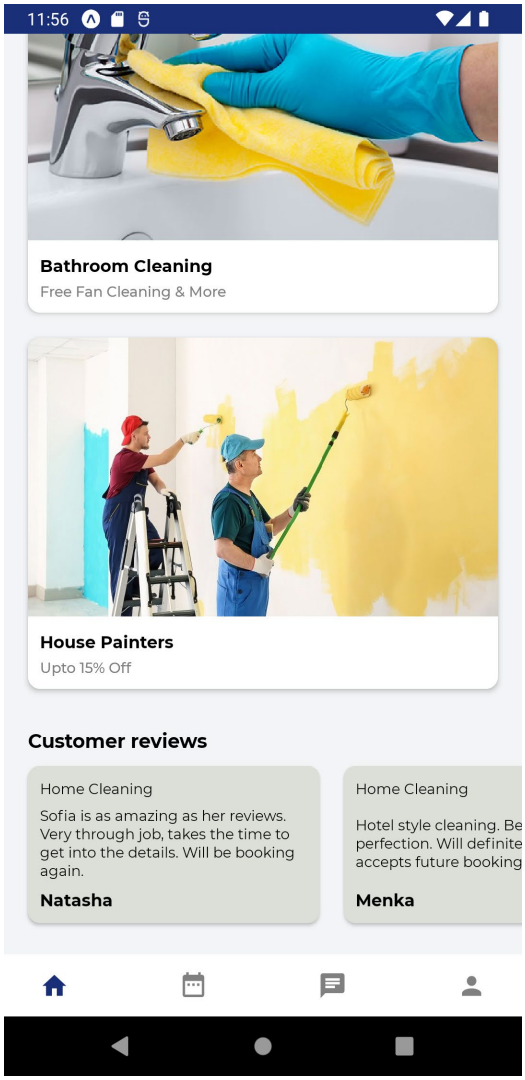




Screenshot 3: Indicates the Register Screen of the user application.

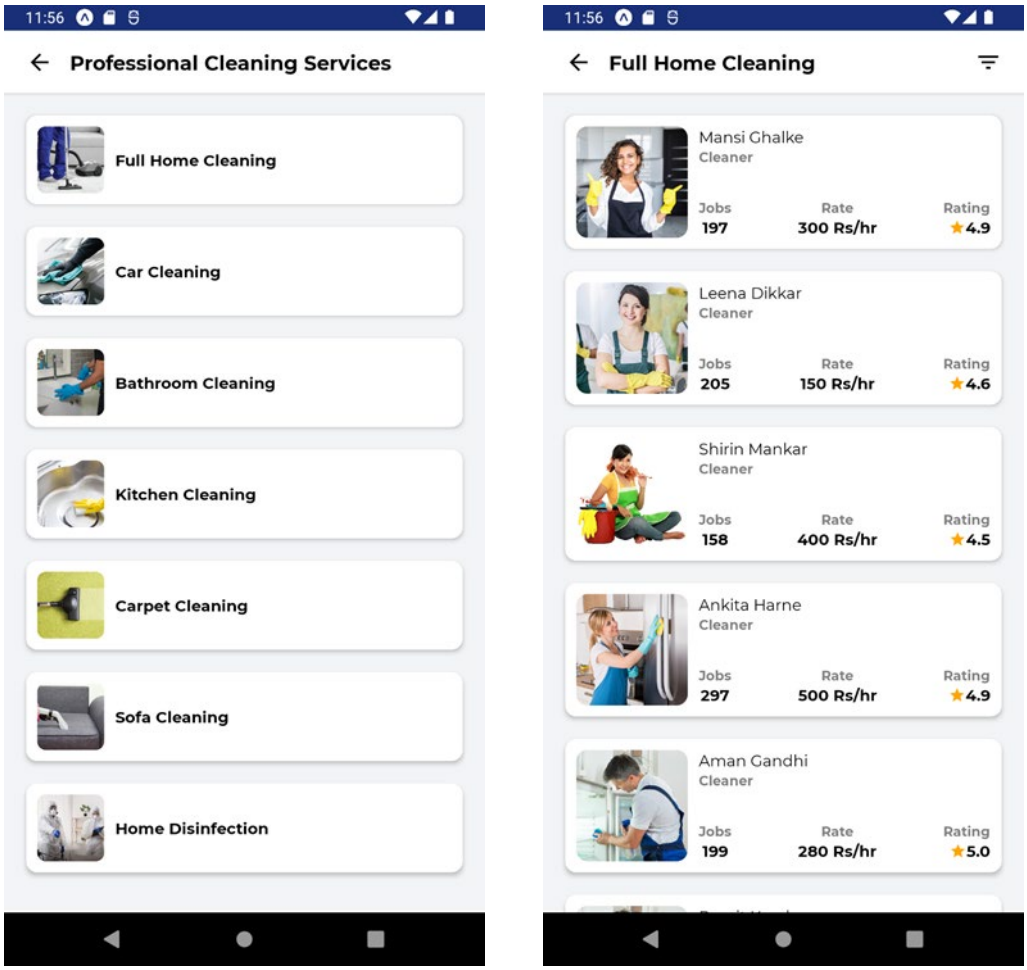
Screenshot 4: Indicates the OTP Screen Of the user Application.





Screenshot 5: and **Screenshot 6:** indicate the Home Screen of the User Application.

Users will see this nice UI developer made with the material in which a react native framework. In This, the user will get all the services And categories like cleaning, plumber, electrician, Beautician, etc. Given below users will see all the reviews of users. And below that, the user will see the icons Home icon, Bookings icon, Chat icon, and Profile icon.

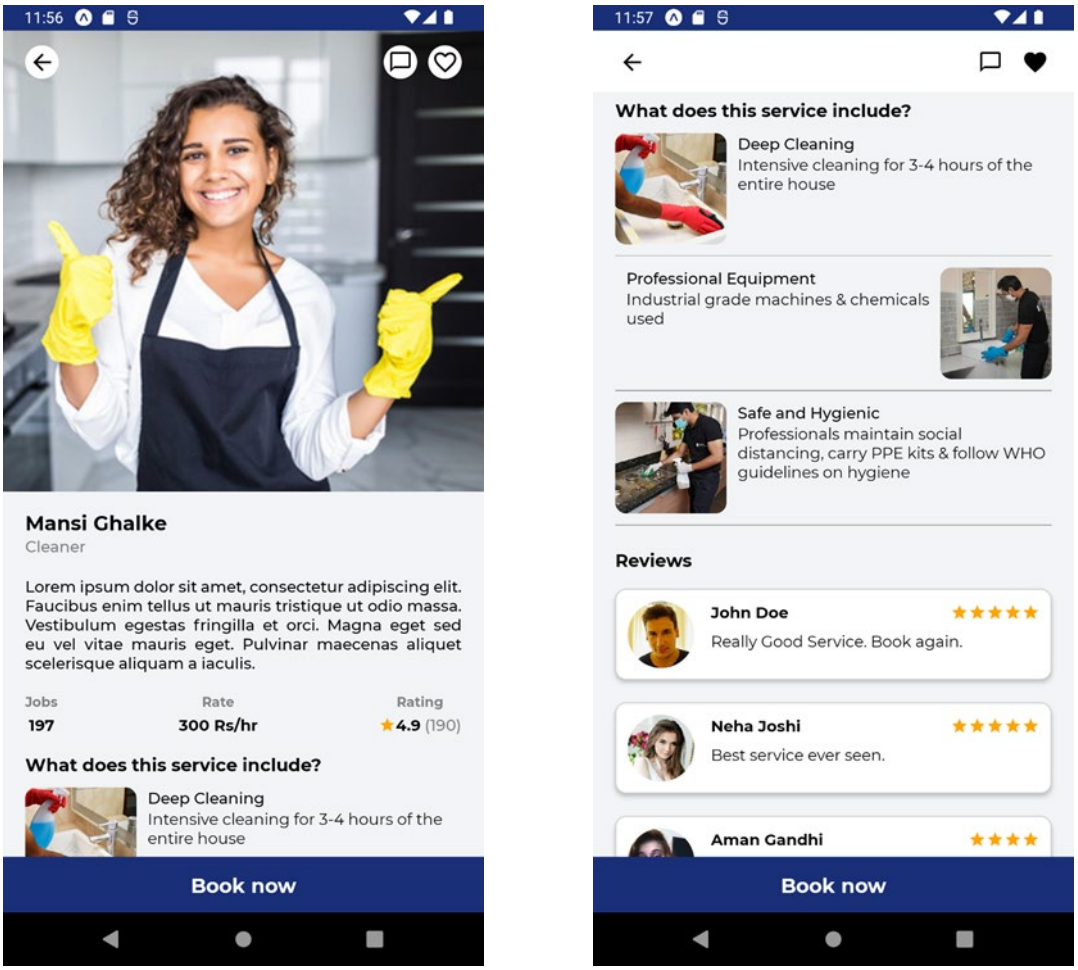


Screenshot 7: indicates Categories Screen in the User Application.

Screenshot 8: indicates Professionals list Screen in User Application.

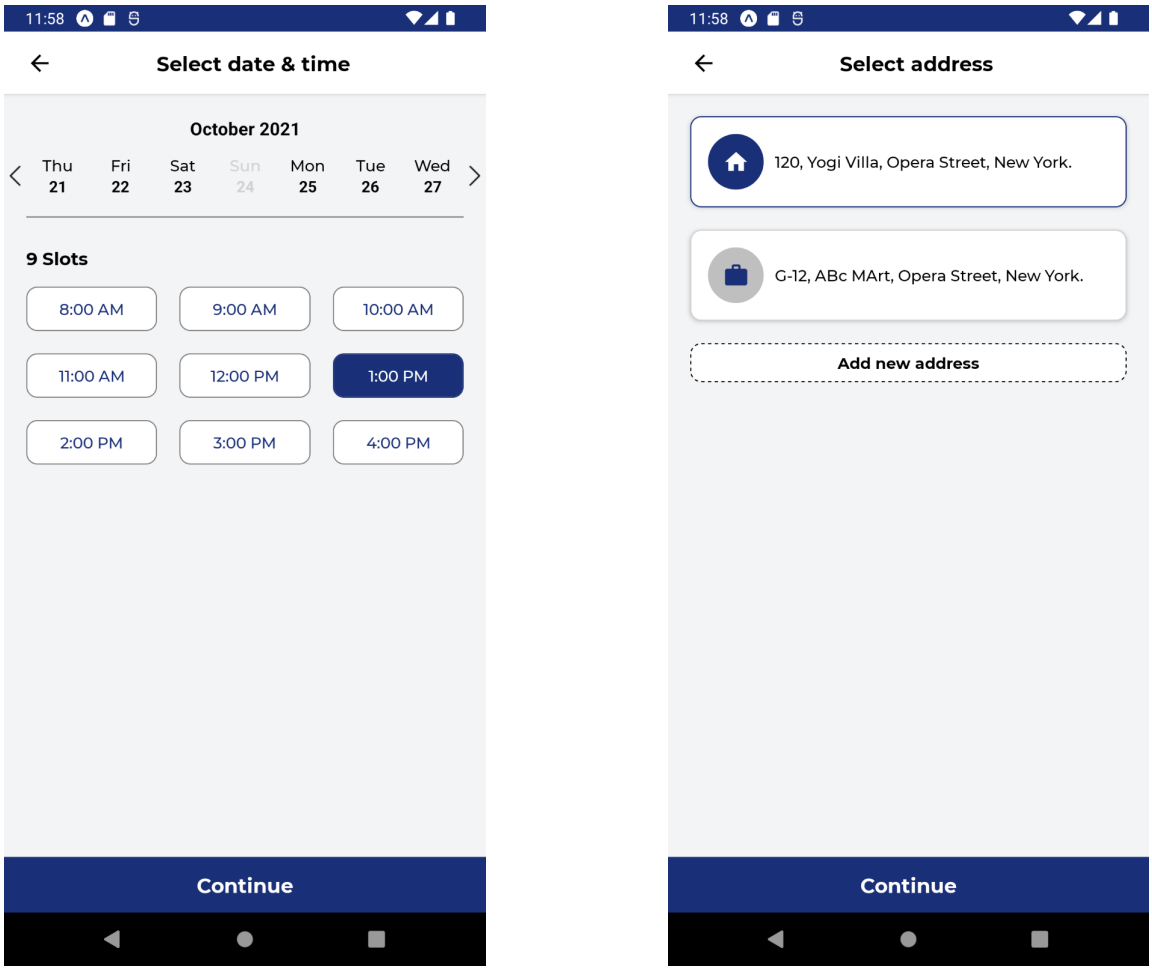
If the User clicks on services like Cleaning, then the user will get all the services in cleaning like Full Home Cleaning, Car Cleaning, Bathroom Cleaning, Kitchen Cleaning, Carpet cleaning, sofa cleaning, home disinfection, etc.

If the user clicks Full home cleaning, then the user will get the list of All the professionals available. Once the user clicks on any professional user will get all the information about that professional.



Screenshot 9: and **Screenshot 10:** indicate the Professional Info Screen in the User Application.

Once the user clicks on any professional user will get all the information about the Professional like what kind of professional, he/she is, what he or does, and what his reviews and ratings are based on the user's can book the professional. Also, the user will get the basic rate information of the Professionals. If a user wants to book the professional then the user has to click on the book now button.

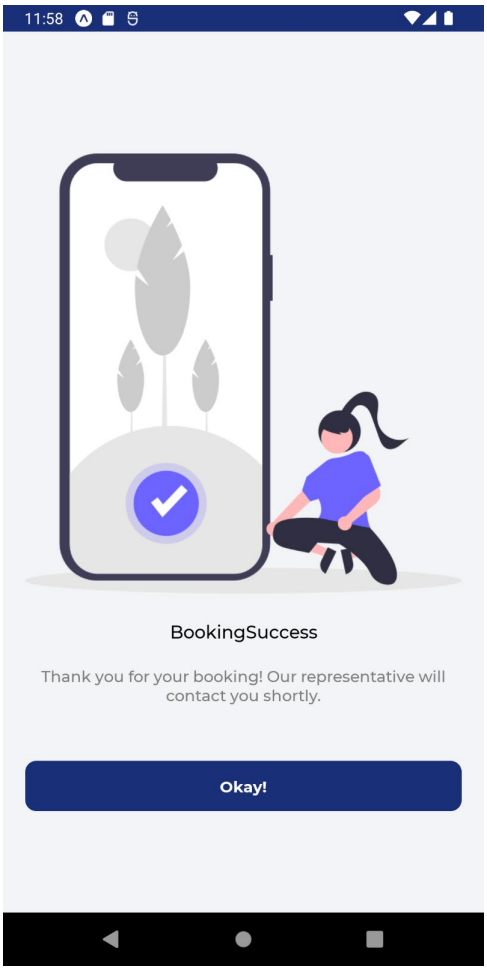


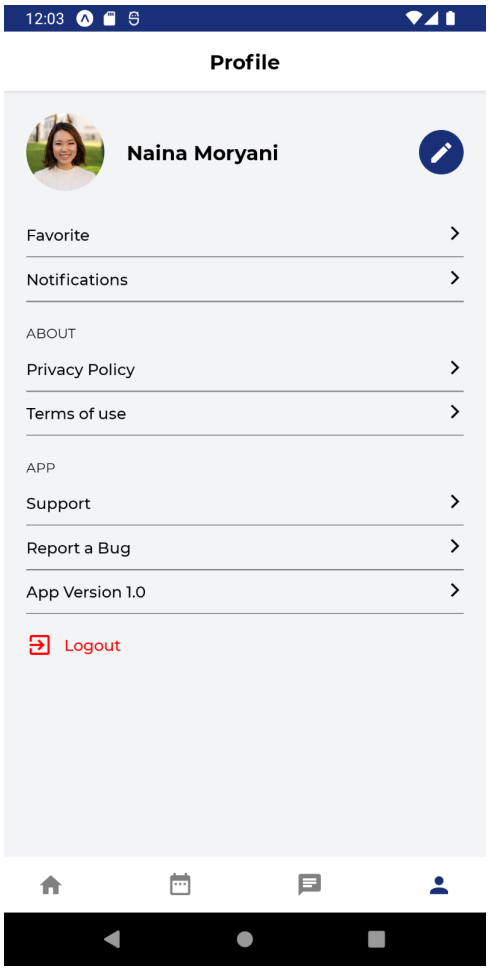
Screenshot 11: indicates the booking Screen in the User Application.

Screenshot 12: indicates the Address Confirmation Screen in the user Application.

Once the user clicks on a book user will get this screen where the user will see the calendar and the timing slots so as per the user's choice user can choose the date and time for the service.

Where the user has to confirm the user's address or the user can add the new address.

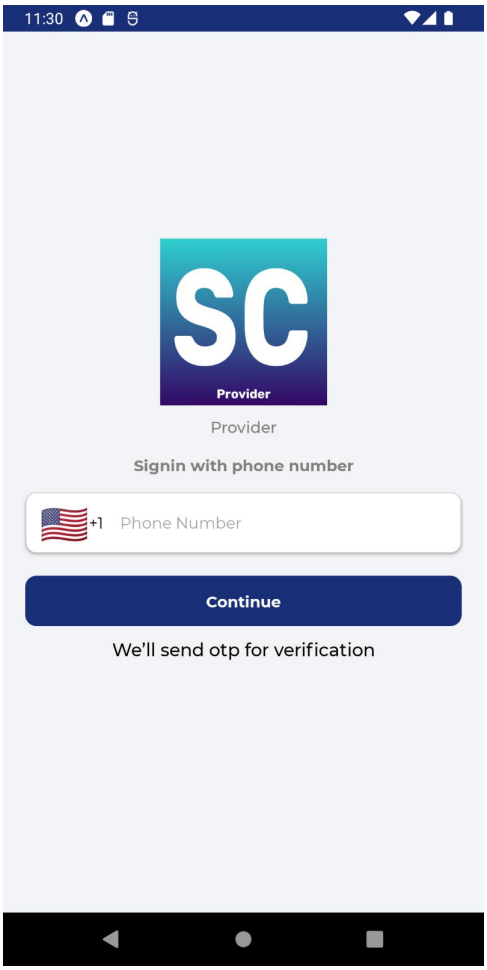


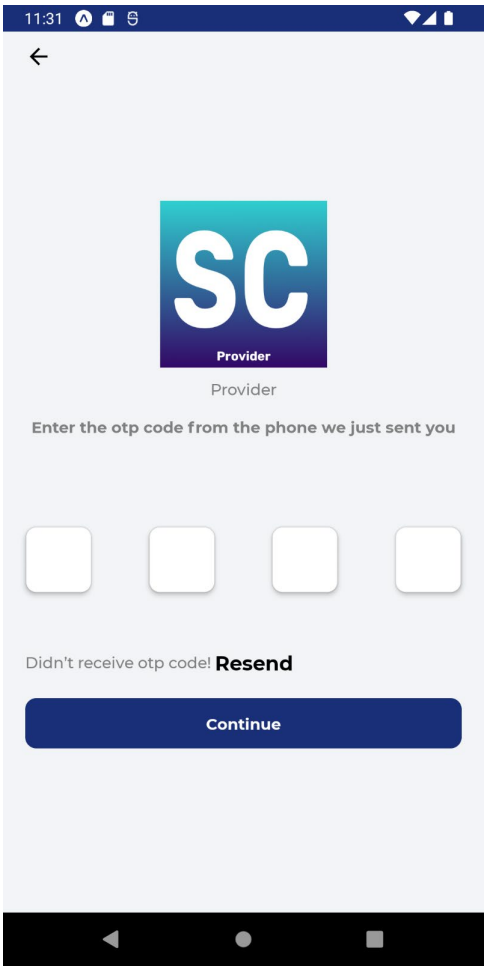


Screenshot 13: indicates the booking success screen. Once the user clicks on continue user will get this screen that the user's booking is successful.

Screenshot 14: indicates the profile screen.

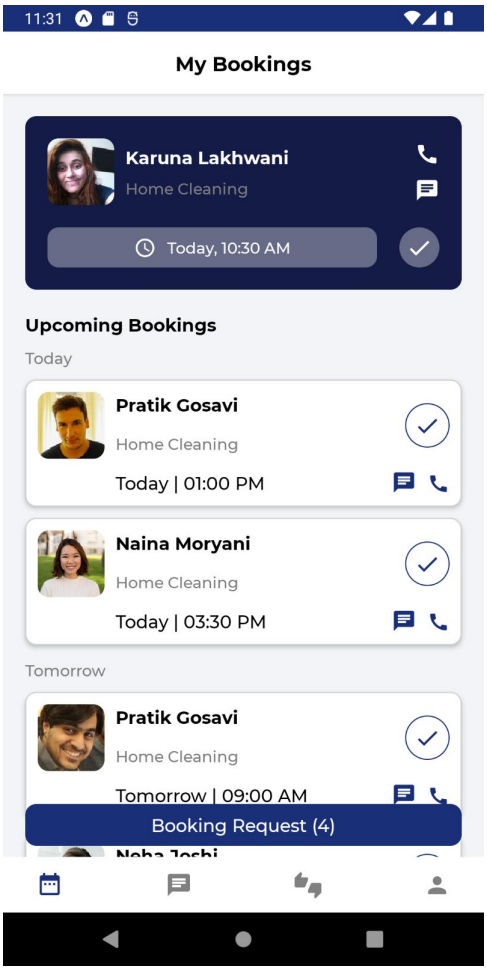
PROVIDER Application:

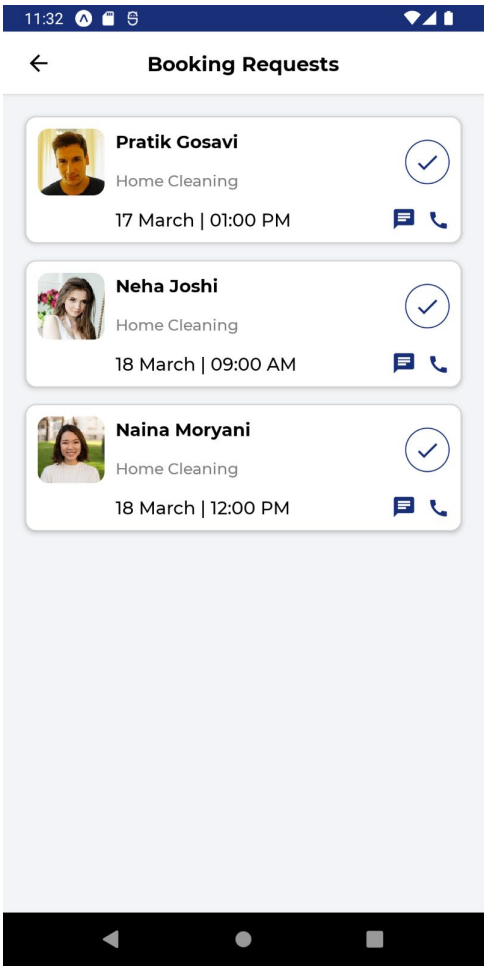




Screenshot 1: indicates the Sign In Screen in the Provider Application.

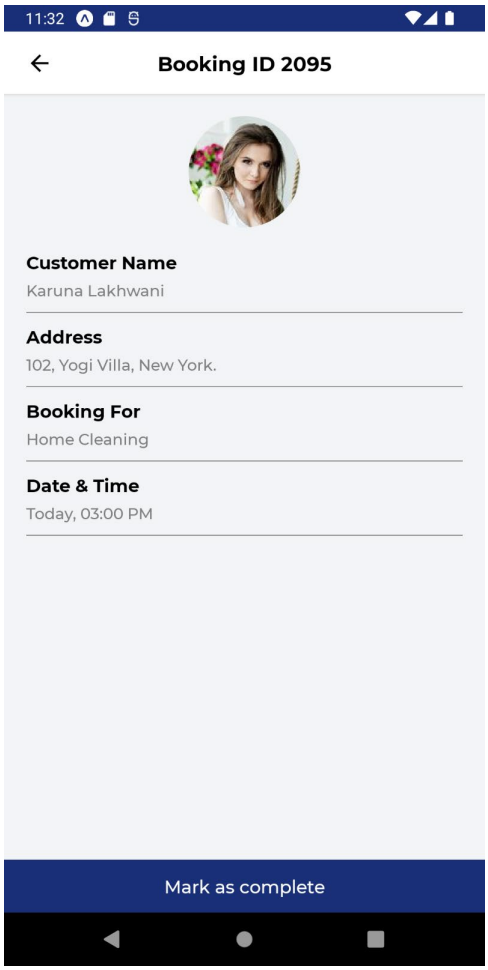
Screenshot 2: indicates the OTP Screen in Provider Application.

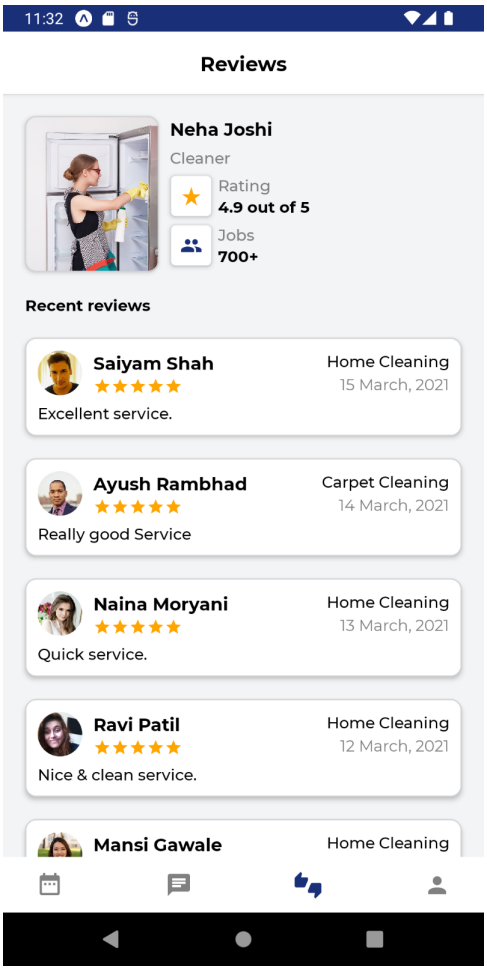




Screenshot 3: indicates the GetBookingHome Screen in the Provider Application. Here the provider will see all the bookings made by the user like Upcoming bookings, today's bookings, and all the booking requests, the date, and timings also.

Screenshot 4: indicates the Requests Screen in Provider Application.





Screenshot 5: indicates the Booking Details Screen in the provider Application.

Screenshot 6: indicates the Reviews and Ratings Screen in the provider Application.

Once the user clicks on any requests user will see all the information of that user like time, date, name of the user, and the address the user.

Where the provider can see all his/her reviews and ratings given by the different users.

5.7 Testing

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, for the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at

implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however, most test effort is employed after the requirements have been defined and the coding process has been completed.

5.7.1 Unit Testing:

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as the user expects. Each unit is tested separately before integrating into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use. Unit testing is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming, a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a class, which may belong to a base/super class, abstract class, or derived/child class. Ideally, each test case is independent of the others: substitutes like method stubs, mock objects, fakes, and test harnesses can be used to assist in testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation.

5.7.2 Integration Testing:

Integration testing, also known as integration and testing (I&T), is a software development process in which program units are combined and tested as groups in

multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integration testing is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to build a product through continual testing and revision. There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

5.7.3 Validation testing:

At the validation level, testing focuses on user-visible actions and user recognizable output from the system. Validations testing is said to be successful when software functions in a manner that can be reasonably expected by the customer. Two types of validation testing Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing before the software goes to beta testing. Beta testing comes after alpha testing. Versions of the software, known as the beta version, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

6. CONCLUSION

City services are needed and demand for services is increasing as people are so busy with their work that no one has time to do things. And Professionals (i.e. service providers) also need the right platform to be able to measure their business to reach as many customers as possible. That would allow them to grow quickly. According to literature research and interviews, home-based services and delivery can be supported by many features of interest to the property owner, engineer/builder, occupier, sender/seller, service provider, employer, and municipality. through the services found in the books and discussions present three interesting points of focus; saving on the cost of new service as well as influence on building image.

7. FUTURE SCOPE

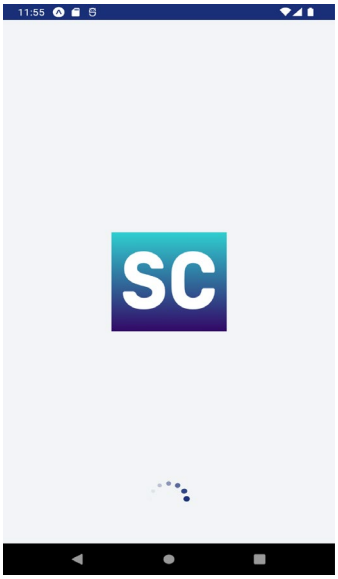
Although this application meets all the feature requirements as planned, there are still some aspects that need to improve.

- In the future developer developed a website and, in our application, the developer will put map navigation for the service provider to find the location of the user.
- And the advertisement of Professional where the Professional can serve his service.
- Also, developers are planning to add the equipment required for the services (For example: Suppose users AC needs any major/minor equipment so will provide this product in our application and user will easily buy on the same platform.

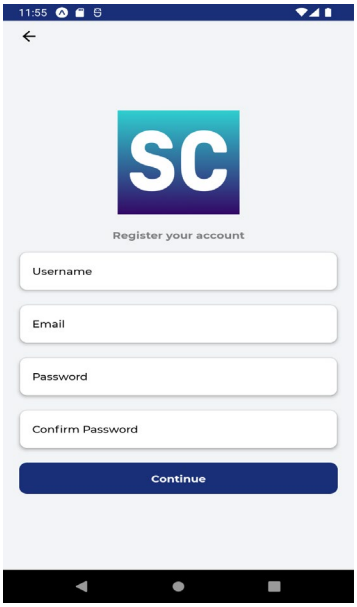
8. USER MANUAL

Follow the Given Figures from Left to Right One by One and up to so on.

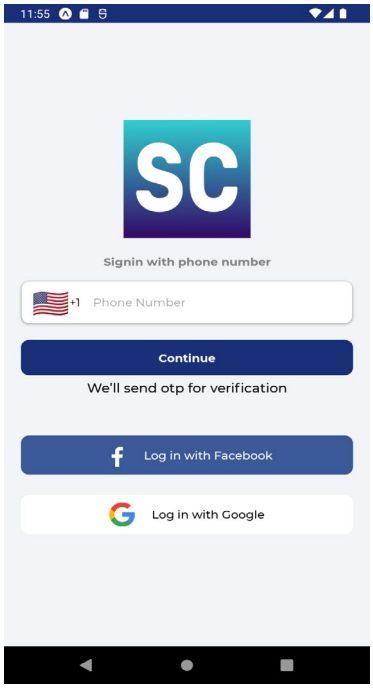
USER Application



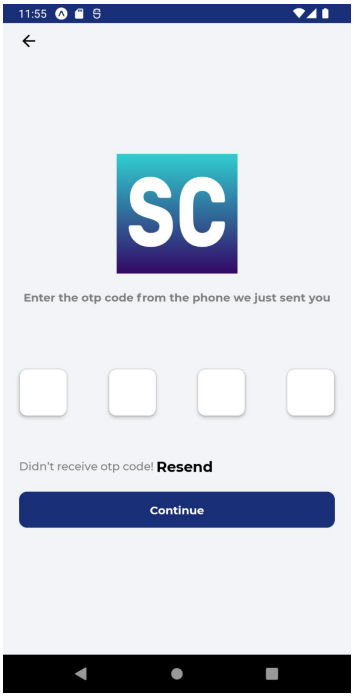
Screenshot 1: This is the splash screen of the CityServices Application.



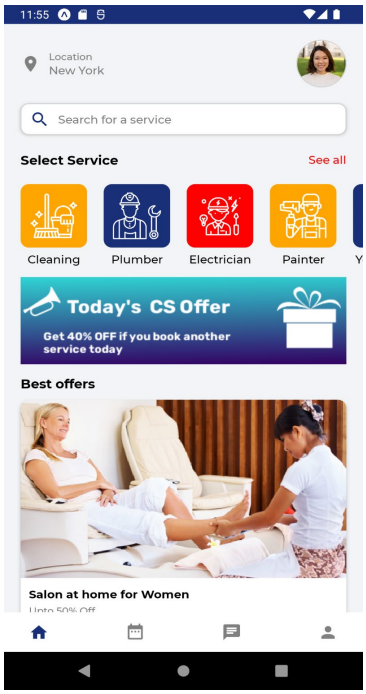
Screenshot 2: After the splash Screen User gets this screen where the user has to go through the registration process.



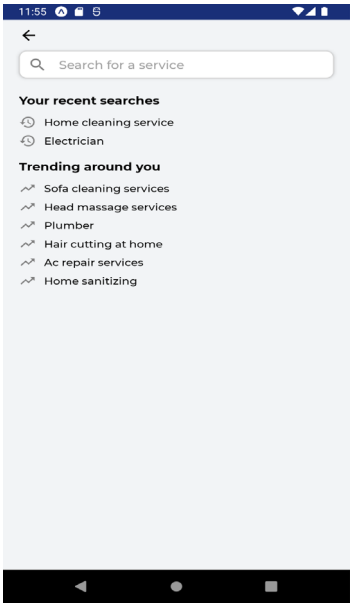
Screenshot 3: After that user has to enter the mobile number or the user can log in through the Facebook or google account.



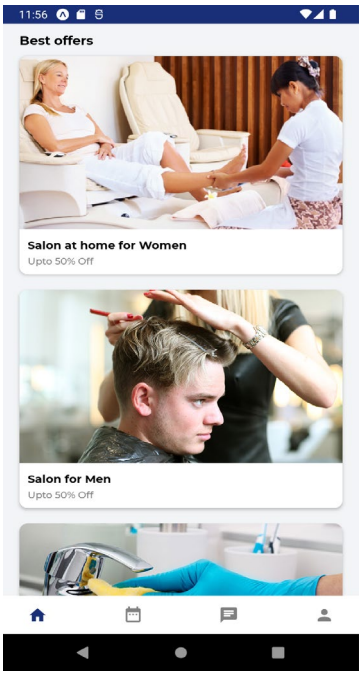
Screenshot 4: Here the user has to verify the one-time password to authenticate to the System.



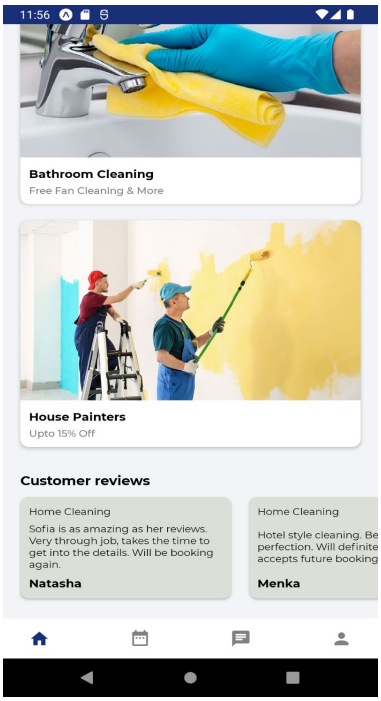
Screenshot 5: After the entire registration process once the user login to the system the home screen of the application is been displayed there. From there the user can see all the available services which are being served by the CityServices Application.



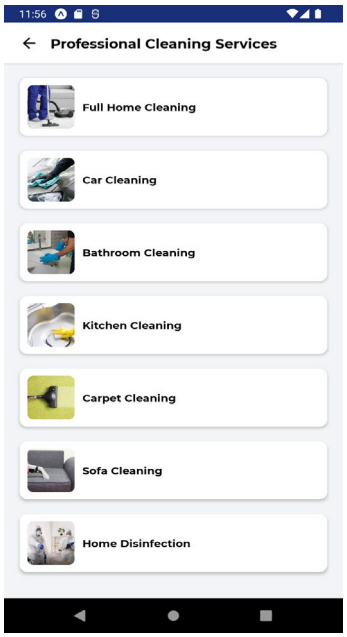
Screenshot 6: At the top of the home screen, there is a search bar where users can search for the services in the application.



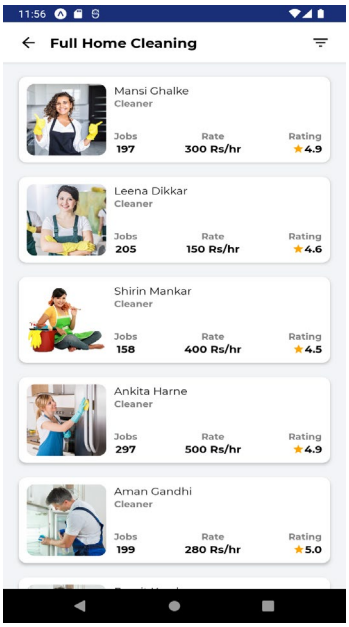
Screenshot 7: In the middle of the Home screen, there is a button for Today's offers. By clicking it user can see all the offers available at the respective Service.



Screenshot 8: At the bottom of the page, by scrolling down the user can see the customer reviews of the respective Services being Served.

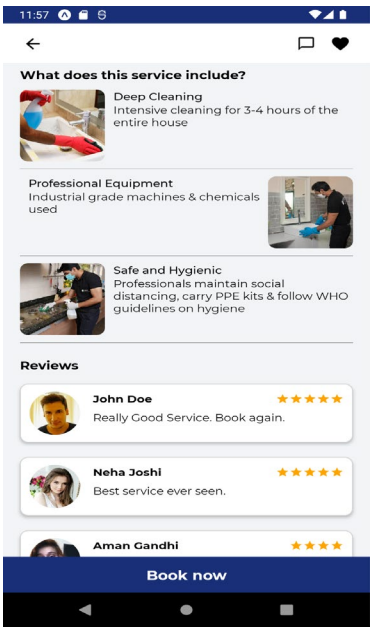


Screenshot 9: After clicking on one of the categories (e.g. Professional Cleaning Services) from the categories section then All the services under this category is being displayed to the User.

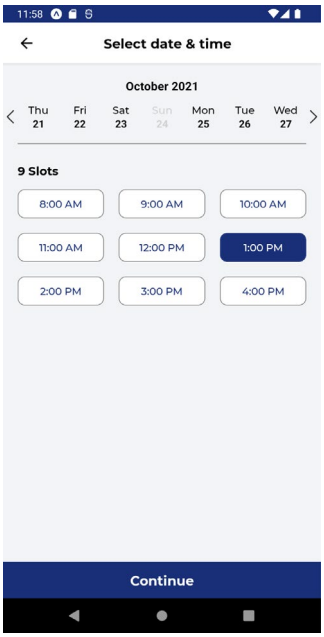


Screenshot 10: After clicking on one of the services (e.g. Full Home Cleaning) from the Professional Cleaning Services category section then All the service providers under this service are being enlisted to the User.

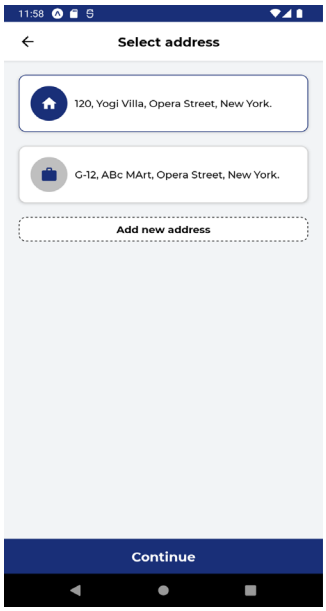
Screenshot 11: After clicking on one of the Service providers (e.g. Mansi Ghalke) from the Professional's list then the detailed information of that service provider is displayed to that User.



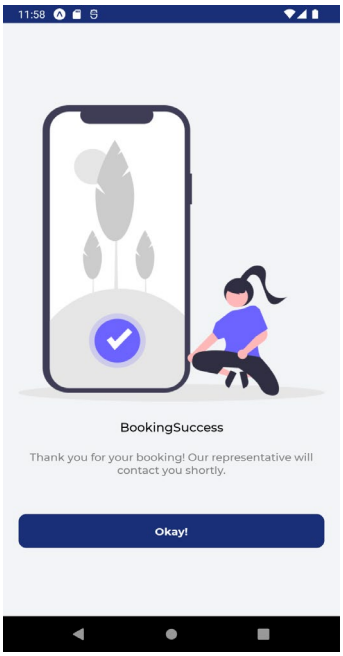
Screenshot 12: By scrolling down users can see "what does this service include?", Reviews given by customers to that service provider and the rating given by the customers to that service provider.



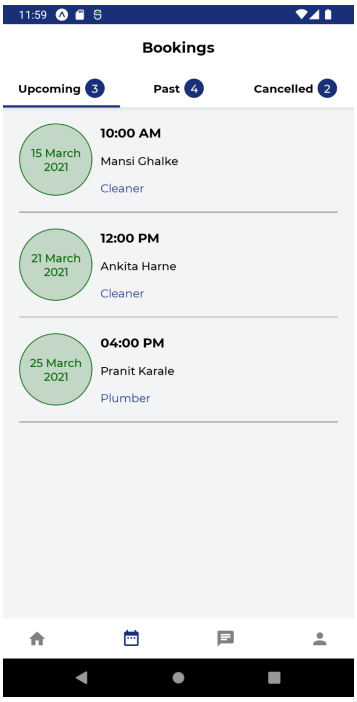
Screenshot 13: After clicking on the Book Now button the user has to select the date and the time slot so that the service provider would serve his services at the user's convenient date and time slot.



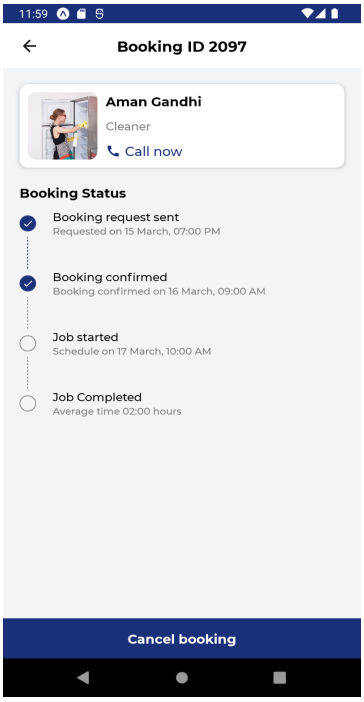
Screenshot 14: After selecting the date and time slot by clicking on the continue button the user has to select his address or the user can add a new address so that the service provider can serve the services at the doorstep.



Screenshot 15: After selecting the address the user has to click on the continue button then the Booking Success Message is Displayed on the Screen.

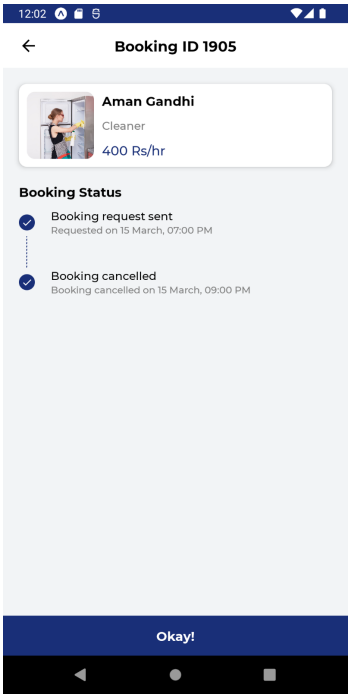


Screenshot 16: In the second tab, the user can see all the upcoming bookings that are being scheduled.

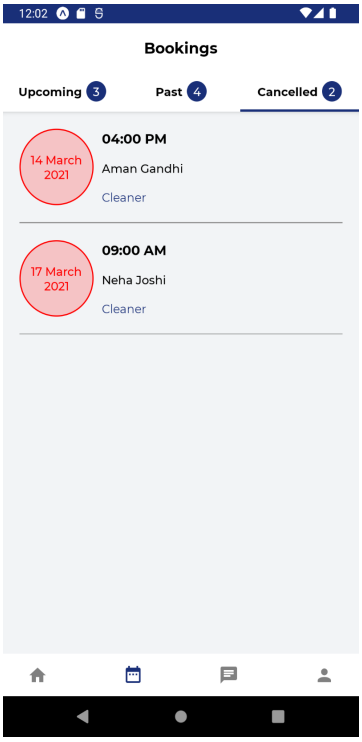


Screenshot 17: After clicking on one of the bookings, the user can see the booking Id, and the booking status and there is a call now option is shown as well.

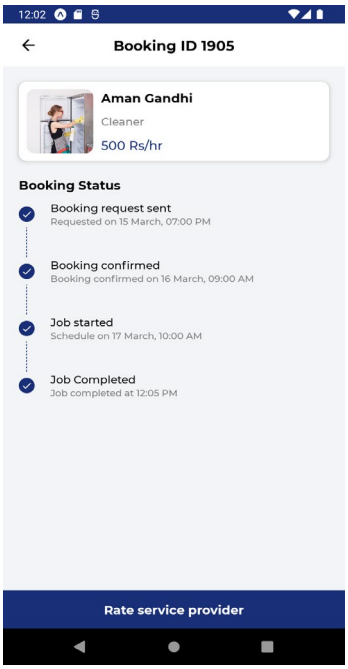
Screenshot 18: After that user can also see the past booking history.



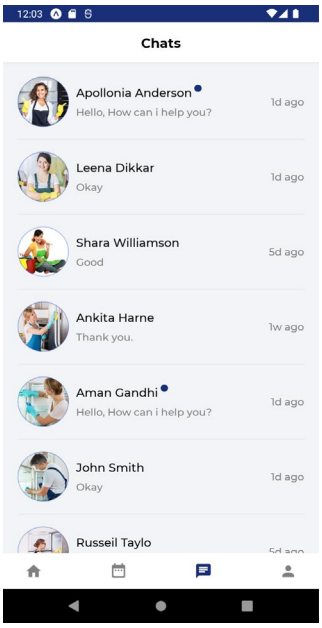
Screenshot 19: By clicking on past booked services the status and booking id have been displayed to the user.



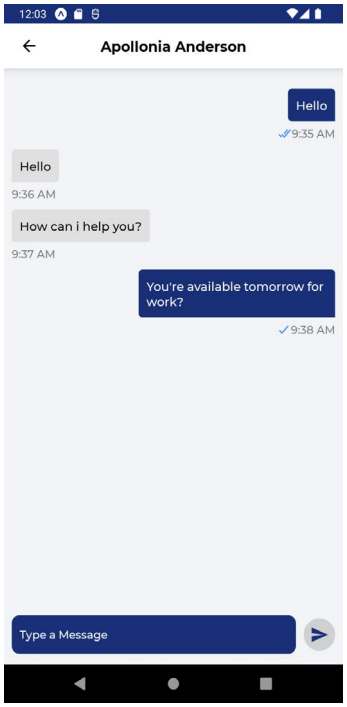
Screenshot 20: After that user can also see the services that were canceled before.



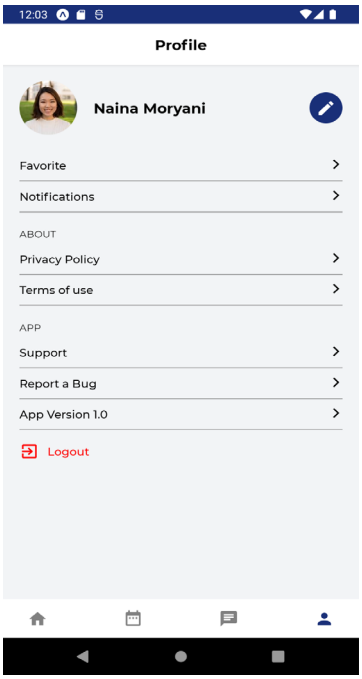
Screenshot 21: After the completion of service, at the bottom, there is a button to rate the service provider based on the service provider's service.



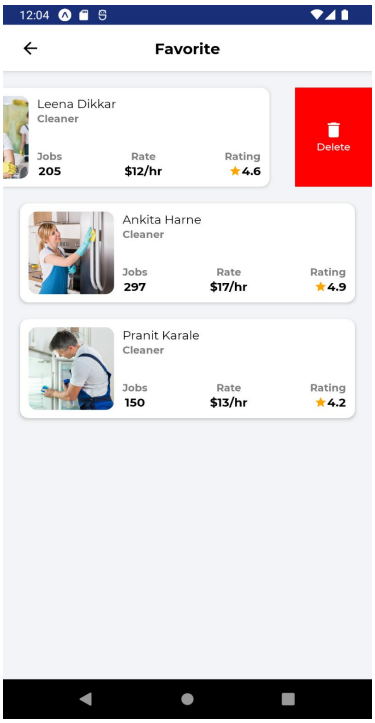
Screenshot 22: In the third tab, there is a facility to chat with the service provider so that the user can thoroughly explain the kind of service the user is expecting from the service provider.



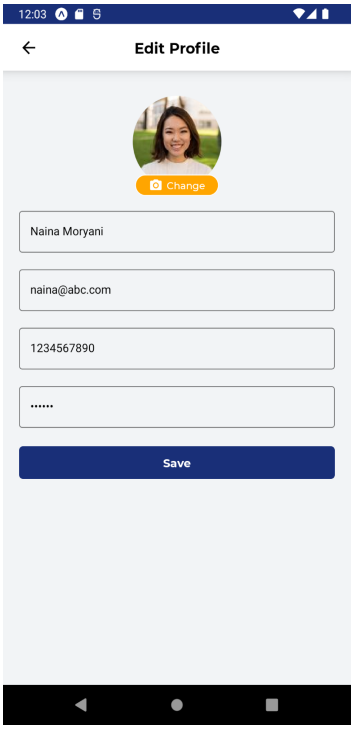
Screenshot 23: After selecting the service provider to chat this screen is displayed to the user from here the user can chat with the service provider.



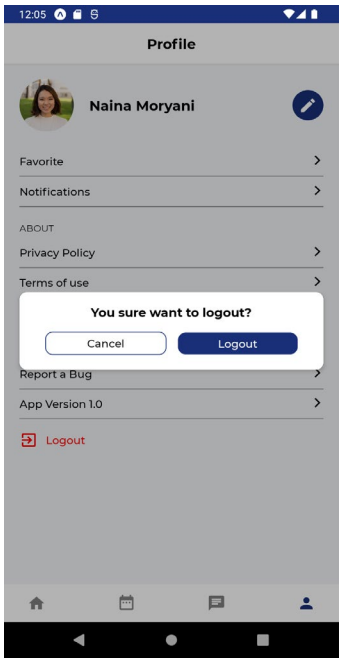
Screenshot 24: In the fourth tab, the User can see the profile page where different facilities are enlisted like favorites, Notification, edit profile, and so on.



Screenshot 25: In the favorite, the user can see the favorite service provider.



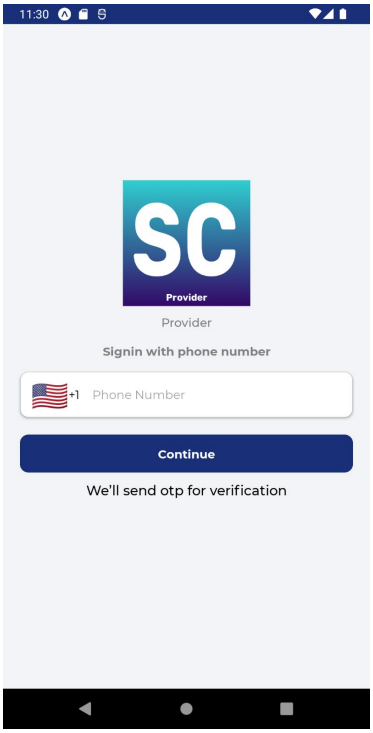
Screenshot 26: After clicking on the edit profile icon, the user is redirected to this screen where the user can edit the details about the user.



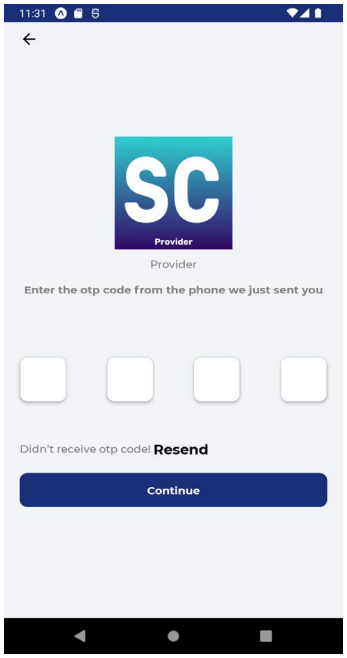
Screenshot 27: At last, there is a logout button, by clicking it user can sign off from the application.

PROVIDER Application

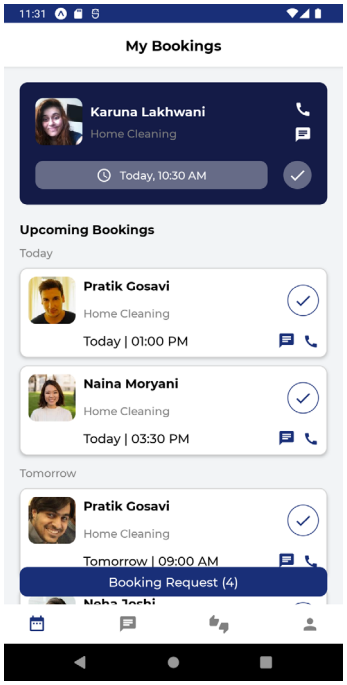
Screenshot 1: This is the splash screen of the CityServices Provider Application.



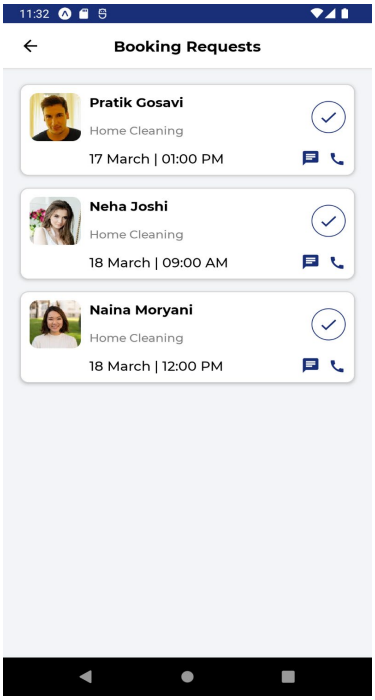
Screenshot 2: After that Service provider has to enter the mobile number and click on the continue button.



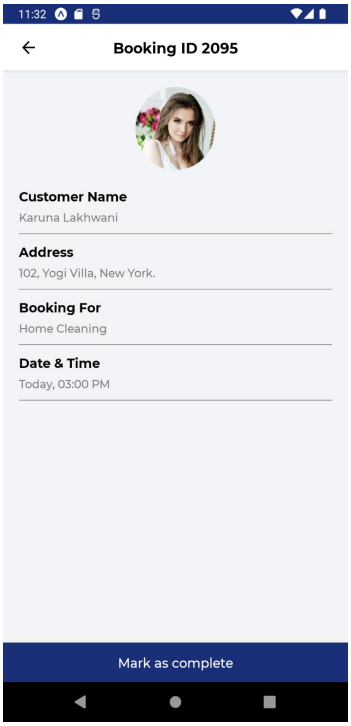
Screenshot 3: Here the Service provider has to verify the one-time password to authenticate to the Service provider application.



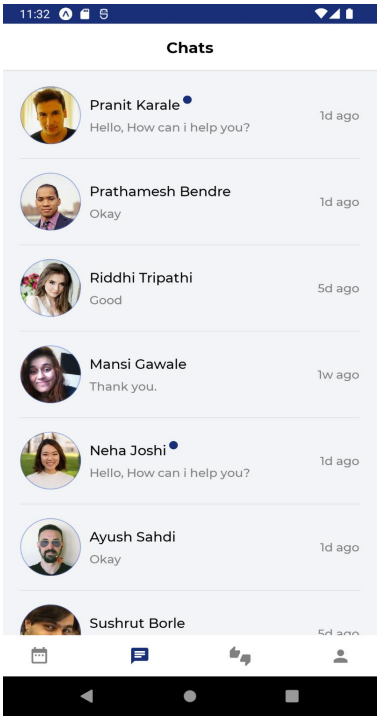
Screenshot 4: This is the home screen for the service provider. Here provider can see the upcoming services where the provider has to provide the services.



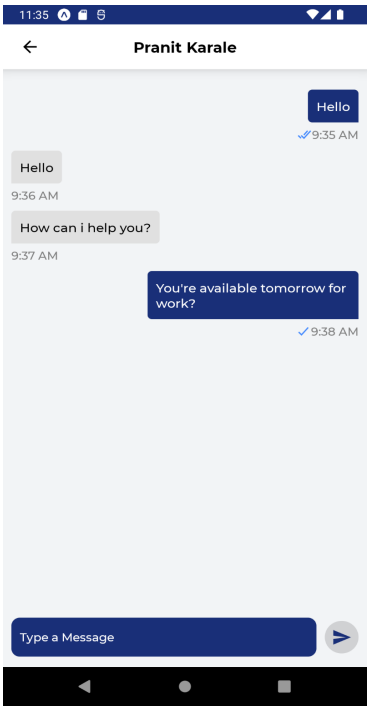
Screenshot 5: Here Service provider can see all the booking requests from here provider can accept the booking from the user.



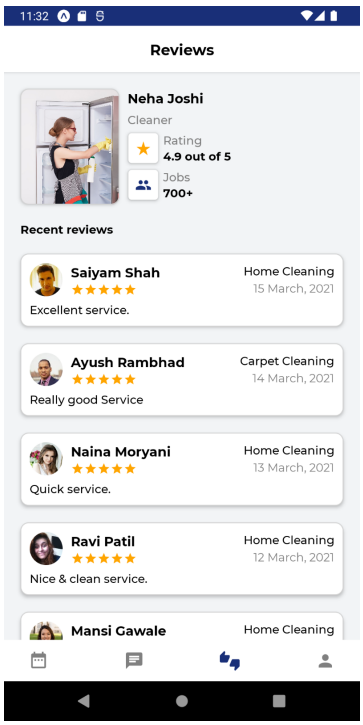
Screenshot 6: From here service provider can see the customer's name, booking id, address, booking for, and Date and Time.



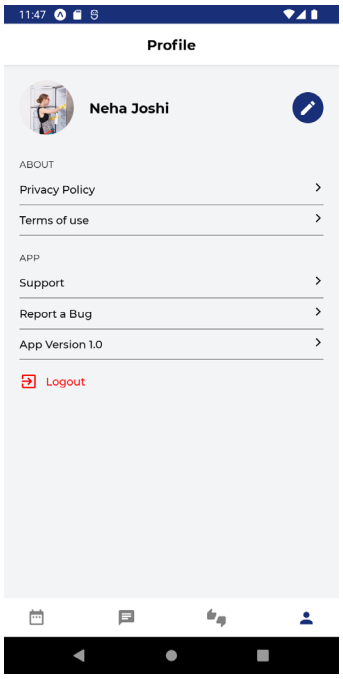
Screenshot 7: In the second tab, there is a facility to chat with the user.



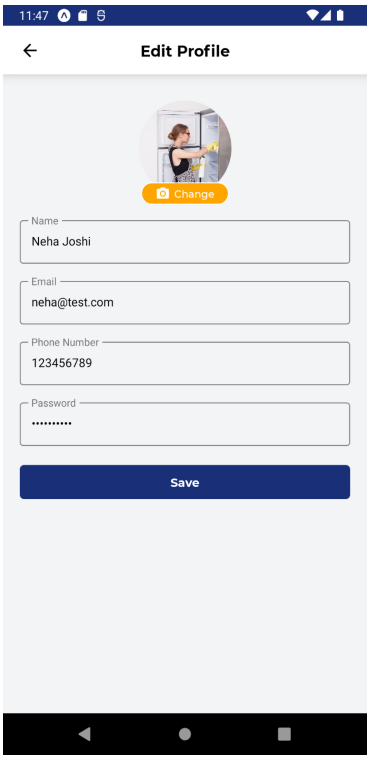
Screenshot 8: This is the chat screen for the service provider.



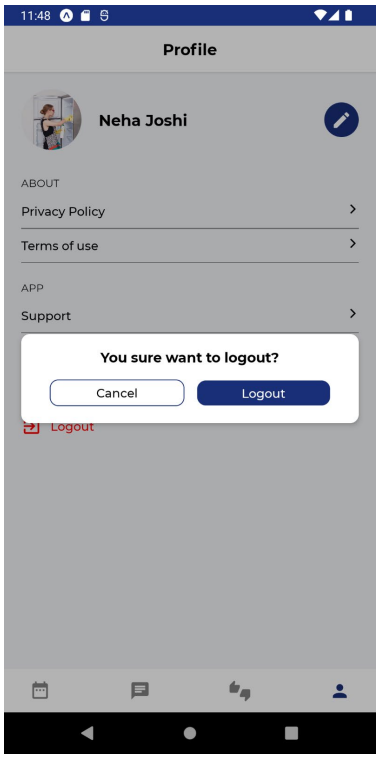
Screenshot 9: From here service provider can see the reviews and ratings that are given to him/her service.



Screenshot 10: In the fourth tab, the service provider can see his/her profile.



Screenshot 11: By clicking on the edit profile icon, the service provider can edit the name, email-id, phone number, and password and then click Save.



Screenshot 12: At last, there is a logout button, by clicking it service provider can sign off from the application.

REFERENCES

[1] Kamal Dharani, Sania Bhatti, Amrita Dewani; Renovate -It: A geo-based technical professional hiring system for repairing and maintenance services 2011.

[2] Cunlu Zhang, Yimeei Guo; A Knowledge Management System Solution Based on ASP Platform for Domestic Appliance Repair 2012.

[3] Denis Gikundi; A Mobile application for locating the available handyman services within a locality 2011.

[4] Reetesh V. Golhar, Prasann A. Vyawahare; Design and implementation of android based mobile app for an institute 2009.

[5] Tom Seymour, Jasmine Zakir Hussain; How to create an app 2014.

[6] Pradeep Kothari, Android Application Development, DreamTech Press 2011.

[7] Dr. Ashok Talukder, Ms. Roopa Yavagal, Mr. Hasan Ahmed: Mobile Computing, Technology, Applications, and Service Creation, 2nd Edition, Tata McGraw-Hill, 2010.

[8] Android SDK: A software development kit that enables the creation of applications for the Android platform. Available: http://www.webopedia.com /TERM/ A/ Android _SDK.html

[9] Eclipse: Is famous for its Java Integrated Development environment. Available: https://eclipse.org/ide/

[10] Kamal Dharani, Sania Bhatti, Amirita Dewani; Renovate -It: A geo-based technical professional hiring system for repairing and maintenance services 2011.

[11] Cunlu Zhang, Yimeei Guo; A Knowledge Management System Solution Based on ASP Platform for Domestic Appliance Repair 2012.

[12] Denis Gikundi; A Mobile application for locating the available handyman services within a locality 2011.

[13] Reetesh V. Golhar, Prasann A. Vyawahare; Design and implementation of android based mobile app for an institute 2009.

[14] Tom Seymour, Jasmine Zakir Hussain; How to create an app 2014.

[15] Pradeep Kothari, Android Application Development, DreamTech Press 2011.

SOURCE CODE LISTING

User Application

- **Login Screen:** The user can log in with the help of a phone number and get the OTP.
- **Home Screen:** Get All the categories of services and all the other details like offers, etc.
- **Services Screen:** Gey All the list of services
- **Professional Screen:** Get All the professional's details with their related work.
- **Booking Screen:** Users can book the professional as per their requirements.
- **BookingSucccess Screen:** Get the screen of booking is successful.
- **SeeAllBookings Screen:** Get All the bookings like upcoming bookings, confirm bookings, and cancel bookings.
- **Chat Screen:** Chat with the provider.
- **Review And Rating Screen:** Users can give reviews and ratings to the professionals.

Provider Application

- **Login Screen:** The provider can log in with the help of a phone number and get the OTP.
- **Home Screen:** Get All the Bookings like upcoming, today, and next.
- **BookingRequests Screen:** Gey All the list of booking requests.
- **Review And Rating Screen:** The provider can see all the reviews and ratings given by the different users.

RESEARCH PAPER PUBLICATION

Title - CityServices.

Authors - Vishal Kothalkar, Rohitraj Deshmukh, Akhil Jamatkar, Monish Sathe, Prof. Sagar Padiya

Publisher - International Journal of Advanced Research in Science, Communication, and Technology (IJARSCT)

ISSN- 2581-9429

Volume - 02

Issue - 4 May 2022

DOI - 10.48175/IJARSCT-3951

Page Numbers - 433 - 439

NAME OF ALL PROJECTEES

Rohitraj Sanjayrao Deshmukh
rohitrajdeshmukh9960@gmail.com
8805421366
Tapadiya nagar mahajani plot akola
444001

Monish Arun Sathe
monisharunsathe@gmail.com
7350428532
Arush Enclave Apartment, Ishwar point, Tapowan
Amravati 444603

Akhil Madhav Jamatkar
akhiljamatkar@gmail.com
8380915405
VHB Colony Ratanlal Plot Akola 444001

Vishal Arun Kothalkar
vishalkothalkar5@gmail.com
7218516417
At Post Dharangaon, Teh-Malkapur,Dist-
Buldhana,Bharambe Layout 443112