,hsffh sdifnA/i

by assignmentslabs 1

Submission date: 13-Sep-2022 08:50PM (UTC-0700)
Submission ID: 1791813745
File name: MAD.docx (786.22K)
Word count: 4624
Character count: 24592

Contents

Executive Summary

We have to develop a mobile application that enables consumers to schedule house cleanings. Customers will register for the service and provide details about their residences, such as the number of rooms, bathrooms, and flooring kinds, among other things. In a way similar to that of a freelancer, when a customer posts on the app that they need someone to clean their house, the cleaners may access it and accept the task; however, the app will determine the price.

You must compile data on every customer and take pictures of their residences. There are two separate access and sign-up forms since the application is necessary to register the contractors. After the job is done, the client and the cleaner get a chance to evaluate the experience. The client discusses the property's advantages and disadvantages, and the cleaner details if the client was happy with the service. Standard data design tools and techniques were used to complete the system design process, and the relevant UML diagrams and flowcharts are supplied to offer a visual representation of the system.

Then, Android Studio was used as the primary development tool for the system development process. An internal database that was created using Firebase used to modify data.

The system includes both functional and non-functional aspects to provide users the greatest amount of functionality and a user-friendly setting to produce the desired results anticipated from the system development.

After that, a highly appealing backdrop is utilized for the system's implementation, and various font and color combinations are employed to create a very user-friendly interface that the users can quickly get used to and navigate with ease.

In order to provide the consumers high security features, the system is additionally outfitted with security needs such fundamental data input validations and username & password confirmations.

Finally, system documentation was created to facilitate the creation of mobile applications. This documentation contains a test plan covering a set of test cases, UML flow chart diagrams, and a final conclusion.

Introduction.

The process of creating software for mobile devices such as smartphones and personal digital assistants, most typically for the operating systems Android and iOS, is referred to as mobile application development. The software may be preloaded on the device, it may be downloaded from an app store designed specifically for mobile devices, or it may be accessible through a mobile web browser. Java, Swift, C#, and HTML5 are some of the programming and markup languages that are used during the creation of applications in this manner. So, at this location,

Within this context, the program facilitates communication between the consumer and the cleaner. Customers have the ability to publish information on the residence that needs to be cleaned. After that, the cleaner is able to read those specifics and decide whether or not to accept the task. After the job is done, the client and the cleaner may each provide an evaluation for the other.

Standard data design techniques are used throughout the process of carrying out the data design. The UML diagrams and flow chart diagrams are used in order to offer a pictorial representation of the system that has been built, and the database schema is utilized in order to provide the connections that exist between the items in the database.

The construction of the library management system was carried out in accordance with the notions of object-oriented programming based upon Java. The primary tool for development was Android Studio, which was used throughout the process.

SQLite, which is an internally driven database, is used in order to successfully carry out the database operations that are necessary for the data activities.

In order to ensure that the mobile application had all of the necessary capabilities, the development process was broken up into many stages, each of which was predicated on a distinct set of activities. Additionally, the system was put through several kinds of tests.

When using the mobile application system, the users are given the option to choose from a variety of user levels, each of which imposes a unique set of criteria, both functional and non-functional, on the user.

As a result, users are given a preliminary validation via the use of user authentication levels so that the data integrity may be confirmed. Users of the mobile application system are afforded an increased level of protection by virtue of the system's implementation of user validations, password-protected logins, and password confirmations.

The mobile application provides the users with an extremely user-friendly interface by using appealing colors and fonts to do so. As a result, the users are able to effortlessly control the program without the need for instruction.

Additionally, in order to give users with a more satisfying experience while interacting with the mobile application, the "long click" button characteristic has been included in the mobile application's design.

While the testing of the test plan is the last testing that is done to certify that the development process was successful, other kinds of testing techniques are also employed to guarantee that the right operating ability of the mobile application system can be achieved.

Task 01.

- Critically compare mobile operating system, development tools and technologies for the above mention application.

Mobile Operating System.

The Mobile Operating System (also known as MOS) is a kind of software that is installed on mobile devices that enables these devices to execute apps and programs. It is the same software that is used for operating computers, but mobile operating systems are more lighter and easier to use.

There is a wide variety of well-known computer operating systems available around the globe.

• Android OS
• iOS
• Windows
• Operating System for Blackberries
• Web OS

Apple's iOS, Google's Android, and Microsoft's Windows are the three most popular mobile operating systems. **[CITATION web18 \l 1033]**

Android vs iOS vs Windows.

Android
• the most widely used operating system in the world
• Based on Linux kernel
• Software with an Open Source License
• Created in-house by Google
• first release in the year 2008
• The most recent version is 12

iOS
• the second most widely used operating system worldwide
• exclusive to use on iPhones
• A high level of discretion and protection
• Designed and created by Apple Inc. Windows

• A Windows mobile device is sometimes referred to as a pocket pc.
• Created by the company Microsoft.
• Lightweight version of the Windows operating system for personal computers.

What are the mobile development tools?

Mobile development tools are pieces of software that have been developed expressly to aid in the process of building mobile applications. These tools may be downloaded from the internet and used by developers. This may be performed via a multitude of methods, one example of which is using native mobile development tools as well as cross-platform mobile development tools. These are just two examples of the many possible methods.

Using native mobile development tools will make the process of creating customized mobile apps for mobile devices much simpler for you. These applications will provide a more streamlined user experience, will have a better general level of quality, and will be able to take use of all of the features made available by the native platforms on which they run. Cross-platform mobile development tools, on the other hand, offer the capacity to design a standard app for many platforms all at once. However, despite the fact that this considerably reduces the amount of time and money required to submit an application, this choice is not available. The overall quality of specialized non-platform applications that were produced in a cross-platform environment is worse, and the number of errors that they include is greater, in contrast to the quality of locally developed programs. [CITATION gur18 \l 1033]

I. Native mobile app development.
Building applications for a single operating system is the literal definition of "native mobile app development."
Ex: android applications, iOS apps

Native applications are constructed using specific programming languages. Java and kotlin for Android
The iOS programming languages Swift and Objective-c

Developers working on Android and iOS platforms need a specific development

environment to create native apps.

Integrated Development Environment (IDE) and Software Development Kit (SDK) are both names for the same toolkit (IDE).

Frameworks for the creation of native mobile applications

Let's take a more in-depth look at the development frameworks that allow native applications for Android and iOS after we've gone over the overall overview that was just presented.

Android: Android Studio.

Integrated development environments were discussed in the prior section, when the concept of native mobile development was presented. Integrated development environments are one of the tools that are required to construct a native application.

Android Studio is the official integrated development environment (IDE) for Android development, and it is also the IDE that developers use the most. When you use Android Studio to build Android applications, you not only gain access to a great code editor that makes use of amazing development tools, but you also get access to certain complex capabilities that may make your experience as a developer more difficult.

Among these traits are some of the following:

The performance of an emulator with all of its features enabled is superb. a building method that is both flexible and based on grids.

Comprehensive collection of frameworks and validation tools.

Support for importing sample code and app features that may be reused, provided through a variety of code formats and interaction with GitHub.

Integration with the Google Cloud Platform.

Provision of support for C++ as well as the Native Development Kit (NDK).

Building native applications on any operating system, whether it an integrated development environment (IDE), Windows, Mac, or Linux, is one of the advantages of using Android Studio for software development.

Having said that, in order to do this, you will need to install a Java Development Kit (JDK).

In addition, there are development tools such as the Code Converter (Java), the Compiler, and the Java Runtime Environment (JRE) (Java).

After the Java Development Kit (JDK) has been set up, you can now install Android Studio and begin developing your first Android application.

Android Studio is a popular option among developers of Android software when it comes to selecting a framework. When you think about all of its benefits, it's not hard to see why.

Native applications such as WhatsApp Messenger, LinkedIn, Netflix, Evernote, Uber, and many more are some of the most popular apps that have been developed using

Android Studio. In conclusion, we are going to discuss the advantages and disadvantages of using Android Studio.

iOS: XCode

XCode, which is found on the iOS side of the aisle, is comparable to Android Studio.

It is the official integrated development environment (IDE) for Apple and has a lengthy history in the realm of iOS programming. Due to the fact that it is native to iOS programming, it supports both Swift and Objective-C.

Compilers, debuggers, storyboard editors, and XML interface builders are just some of the tools that can be found in XCode, which is used to create iOS applications (XIB).

Some parts of iOS app design are only accessible through X code because to the exclusive nature of Apple's default style.

Even on non-native platforms such as Flutter or React Native, for instance, storyboards cannot be changed without XCode.

In addition, several alternative integrated development environments (IDEs) still call for the installation of XCode.

characteristics,

The following is a list of some of XCode's important features:

A wide range of programming languages are supported, including C, C++, Objective-C, Java, Python, and Apple Script, among others.

Utility for the creation of huge binary files, including codes that are compatible with different architectures of the Mac-O executable file format.

iOS application compilation and debugging on ARM architecture.

Graphical User Interface (GUI) tool that eliminates the need for coding and makes it simple to create and prototype user interfaces.

Filtering of code, version editing, an encoding assistance, and several forms of source control are all included.

Because each of these capabilities comes with a sizable number of ohms, XCode has become the universal programming language for iOS app developers.

iOS developers of all skill levels consider this to be a very helpful solution since it is a decent solution.

The source code checker, which calls attention to any typos that may have been typed in, is sure to be helpful to novice programmers.

Taking use of this functionality is likewise quite straightforward.

In a similar vein, you may reduce the amount of time spent encoding by making use of code snippets and templates that are often used. XCode will make it easy for you to upload your software to the App Store as you make progress with it.

Additionally, if you've ever found yourself in a bind, the documentation that comes with XCode is as comprehensive as it gets. The developer will find in the assistance library a variety of resources, such as documentation for the software development kit (SDK), references to APIs, code samples, and encoding instructions, as well as

everything else necessary to successfully and quickly address issues.

The viewer for the document has all of this information. Everything is updated on its own, as you would think, and XCode is used to develop the vast majority of native iOS applications.

Through XCode, popular applications such as Firefox, LinkedIn, SlideShare, Asana, Lift, and WordPress were ported to the iOS platform.

II. Cross- platform mobile development.

Third-party suppliers are responsible for the development of cross-platform mobile technology.

These providers provide an integrated application programming interface (API) on top of the native software development kit (SDK). Using this built-in application programming interface (API) and a single integrated development environment (IDE), developers can now construct a single code base that is compatible with numerous operating systems.

The most effective application frameworks available across platforms

There are a few mobile frameworks that are compatible with several platforms, but let's take a look at the top five providers in this space.

Flutter.

Even though it's been around for just a short while, Flutter has already built a name for itself in the industry and is continuing to get more and more followers.

2015 was the year that Google first began working on the framework, but it wasn't until late 2018 that Flutter took it out of beta.

Since then, developers have been using it to construct mobile applications that provide users with a native experience.

Flutter is a multitasking mobile framework that is built on Dart, which is Google's programming language. In general, Flutter is open-source and free. This programming language is an essential component of the framework that Flutter is built on.

Pre-time native code, often known as AOT, has been compiled for Dart on several different systems.

Flutter has great space for improvement in both its boot time and its overall speed as a result of its ability to remove the need for a JavaScript bridge (as in native response).

Characteristics

The following is a list of some of the great features that are available to you when you use Flutter:

This functionality, known as "Hot Reload," refreshes your application as soon as you make any changes to the code. In other words, you are able to correct faults in a matter of seconds. You are not obligated to try out new features or upgrade.

They are a very crucial component in pipe development, and their name is widget. Each widget gives you the ability to customize a different component of the overall user experience. Performance is improved with Flutter's widget hierarchy since there is no

longer a need for continuous migration of code between Flutter and the platform.

Integration of Code: Flutter makes use of a few different ways that make it easier to communicate not just with code from outside sources but also with the API. On Android, for instance, you are able to reuse native languages such as Kotlin, which is local to Android, or Swift, which is native to iOS.

International Float Libraries Flutter's internationalized widgets and classes make it simple and hassle-free to install your app in many locations.

The Flutter app creator is another useful feature of Flutter.

It's essentially a drag-and-drop editor that doesn't need any coding, and you can use it to create UI designs that look fantastic. Flutter, despite its age, has already been used to develop interesting applications such as the new Google Ads and Alibaba apps. These apps were created using Flutter. Therefore, let's briefly discuss the benefits and drawbacks of it.

Benefits-:

Flutter applications give outstanding performance (compared to other cross-platform frameworks).

During the development process, code changes may be executed extremely rapidly thanks to Hot Reload. Both pre-built and user-created widgets make the process of developing applications far more efficient.

Beneficial for the creation of elegant and cutting-edge user interface designs.

A fairly gradual and manageable learning curve for novices. Because Flutter is expanding at such a quick rate, there is a growing community of software developers.

Disadvantages-:

a lack of libraries provided by other parties. Additionally, the Flutter library has its limitations. Flutter applications often use a significant amount of storage space.

Compatibility with iOS remains a significant challenge for software developers.

When compared to other common forms of technology, using Dart could need a little more skill.

It will take some time (several weeks) for the cross-platform, including Flutter, to obtain the most recent changes to the operating system.

Croslithics is ineffective when used on floats. The management of push notifications requires some strategy.

Native applications run ten times more quickly than Flutter applications.

Xamarin.

Although Xamarin had some level of popularity in 2019, it is currently steadily falling behind in the race for the top spot in the Statista.com statistics.

Since its inception in 2011, when Xamarin was first released, the platform has amassed

a community of more than 1.4 million developers.

The response is native, the same as Xamarin, which is dependent on C#, an established programming language.

Because it is a component of the.NET framework, you will be able to utilize your code alongside other.NET features while taking advantage of the platform's remarkable reliability. Lambdas and Language Integration Query are two examples of available functionalities (LINQ).

It builds applications that are compatible with all mobile devices by combining Xamarin C# and specific local libraries, and then wrapping the whole thing in a.NET layer.

In a nutshell, the code for Xamarin is compiled on a local machine. By using an API-based strategy, the framework encourages the development of platform-specific functionality.

As a result of this process, the final product has performance and a user interface that are on par with those of native applications designed for iOS and Android.

Xamarin provides two distinct products—named Xamarin—to support the two most popular mobile operating systems. iOS and Xamarin. Android.

For iOS, the AOT is compiled using Xamarin source code, and then it is translated straight into native ARM assembly code.

On the other hand, Xamarin Android applications use a technique called Just-in-Time compilation to build native assembly code while the application is actively running.

First, the source code must be turned into an intermediate language so that this may be accomplished.

When developing for several mobile platforms, developers are able to reuse a significant portion of their code when they utilize Xamarin.

Xamarin. One product called Forms is able to accomplish just that. It is a library that contains over 40 different controllers that are compatible with several platforms and is aimed to assist in the construction of prototypes and apps.

It uses React Native.

Facebook's response to Google's front is an alternative known as the nature of the reply.

Since it was released in 2015, many people consider it to be a revolutionary step in the field of cross-platform software development.

Prior to the year 2015, Facebook has been working on the project for its internal hackathon from 2013. The primary objective was to enable web creation on mobile phones while preserving the program's interoperability across several platforms.

In a nutshell, the behavior of the response is determined by a very capable JavaScript package that goes by the name Reaction.

To construct a user interface for a fully complicated mobile application, the framework makes use of the functionality provided by smaller bits of code that are referred to as "components."

Because these reactive components may be reused, the development process can be sped up, which in turn decreases stress. It is open-source software that enables

applications designed for both Android and iOS, in addition to online apps.

One of the most significant benefits, according to the programming language that is often used in JavaScript, is that the response is a local giveaway.

Along with HTML and CSS, JavaScript is considered to be one of the most important web technologies in the world. Because of this, it is often quite simple for many developers who have just a little amount of expertise in web programming to

Make the most of your remarks as a new talent.

Applications such as Arty, Bloomberg, and Delivery.com all have the capability of activating mobile apps from inside a responsive native environment.

The architecture that underpins local applications is comprised of three primary threads, and they are as follows:

The user interface thread is sometimes referred to as the main thread. The app designed specifically for Android or iOS may be used here. Additionally, it is the only thread that may alter the user interface of the application, making it unique in that regard. All user interface threads are shared by all mobile applications.

Shadow Threads: In addition, the native reaction is dependent upon the background thread in order to compute the layout that should be used for the reaction library. This function is carried out via ghost threads.

Your JavaScript Code Is Activated And Executed By The JavaScript Thread Finally, the JS thread acts as a JavaScript engine that is responsible for activating and executing your JavaScript code. The UI feed is responsible for regulating the display of UI elements and processes, and it is also where the structure and functionality of the UI are specified.

The native response could employ additional threads, such as native module threads and renderer threads, depending on the operating system and the kind of project that is being worked on. With that being said, here is a rundown of the advantages and disadvantages of using the native reaction.

Benefits-:

There is a critical need for much shorter developmental periods. The quicker the refresh rate, the faster the debugging process will be.

Due of its dependence on JavaScript, React Native has access to a large community of developers. Implementation of the user interface that is both simple and attractive.

Performance findings for applications that are almost native

Disadvantages-:

There are restrictions on custom modules and components.

For the development of some platform-specific modules, a native developer is still

necessary. Problems with compatibility and bug fixing are rather prevalent.
A significant amount of supplementary resources is required on top of the development environment.

Task 02.

- **Provide the UML diagrams/ flowcharts for the given problem with clear explanations on the design decisions. (Use case, class, activity).**

Use case diagram.

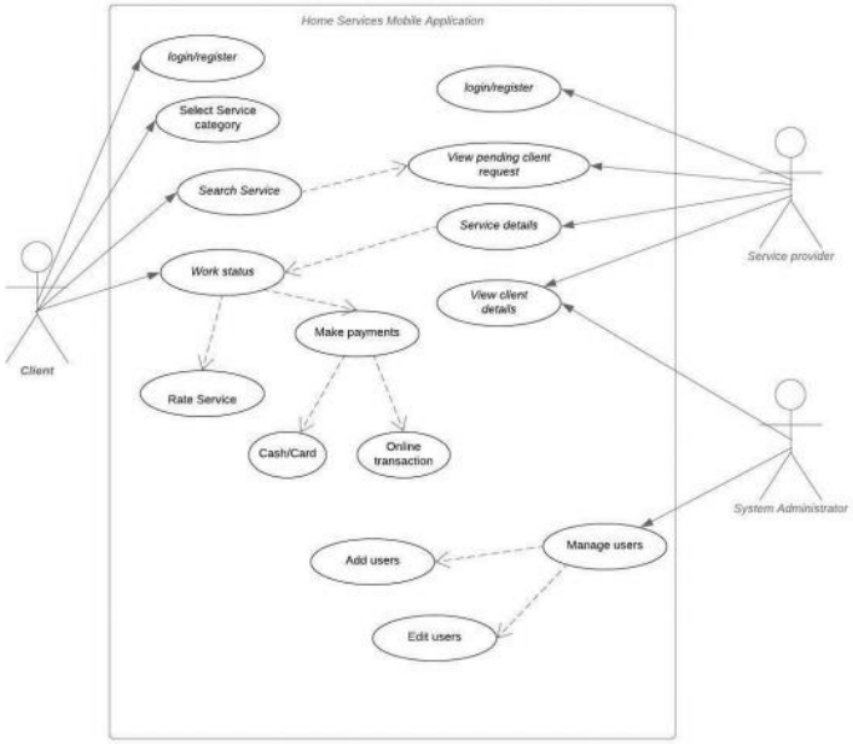

Figure 1 Use case Diagra

Sequence diagram.

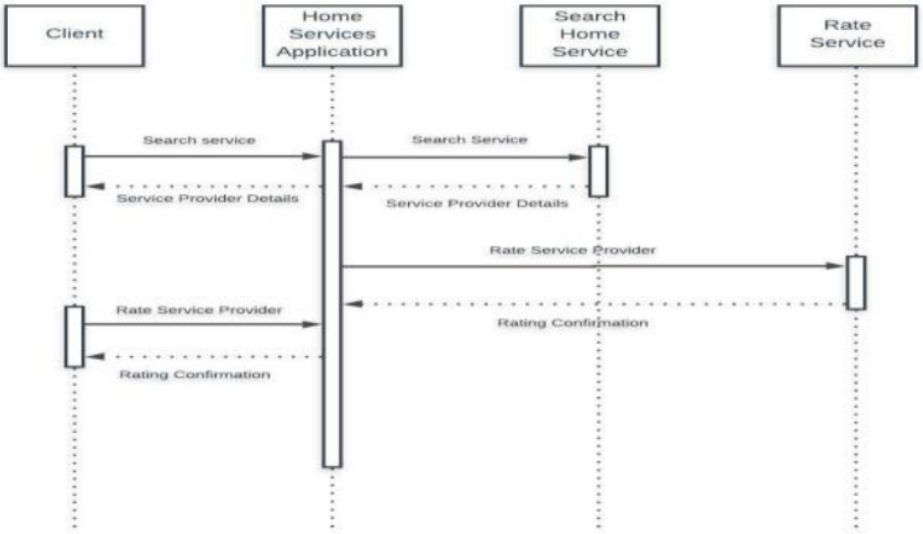

Figure 2 Sequence Diagram

Control flow diagram.

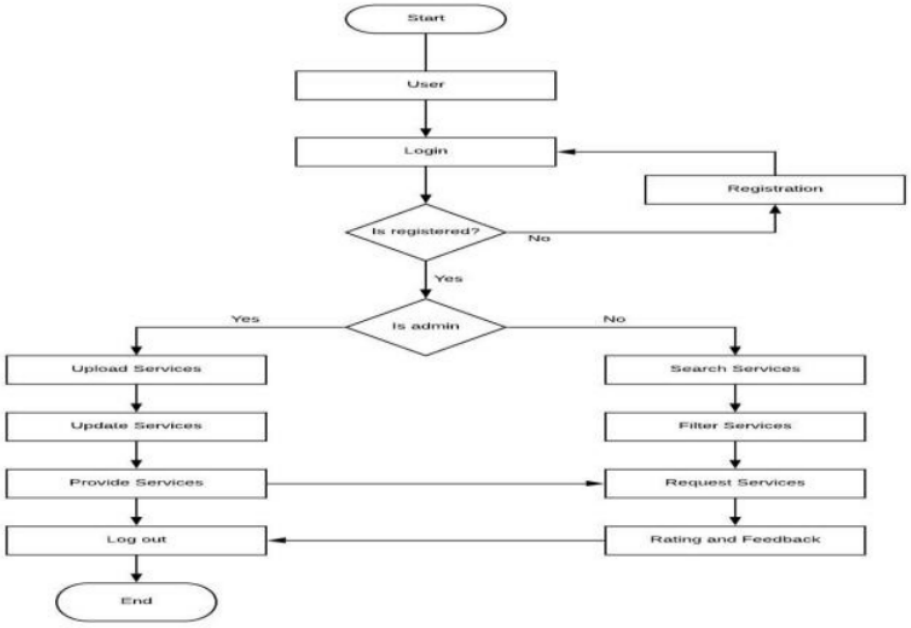

Figure 3 Control flow diagram

Activity diagrams.

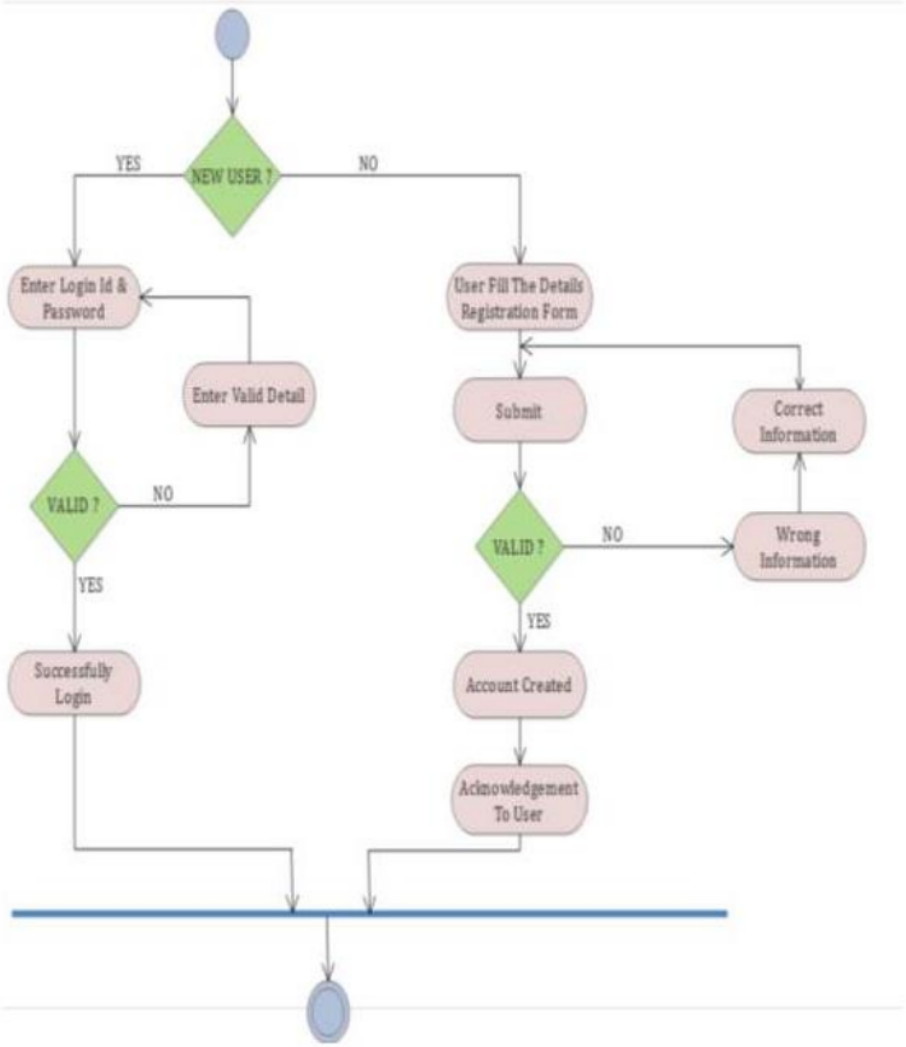

Figure 4 Login activity

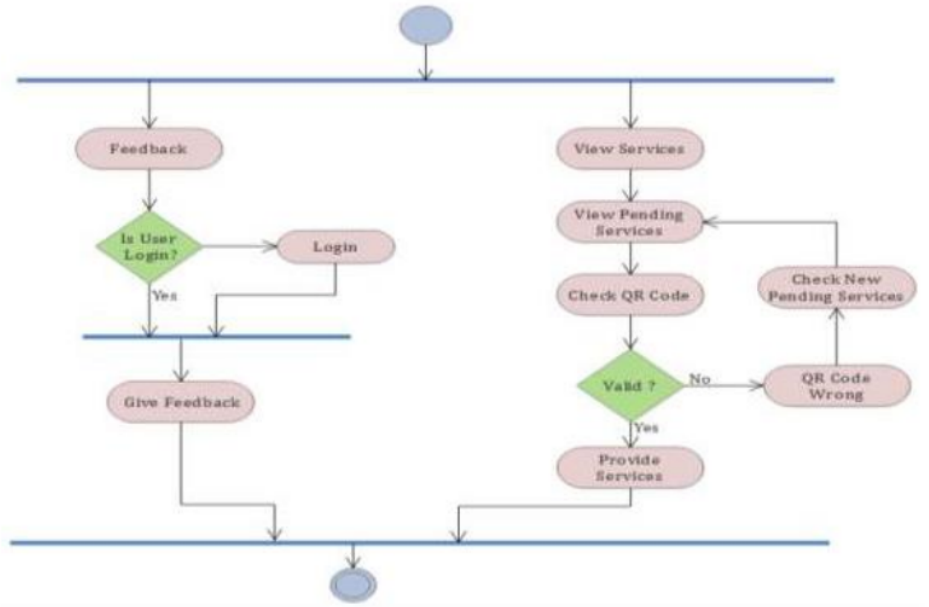

Figure 5 Service provider activity

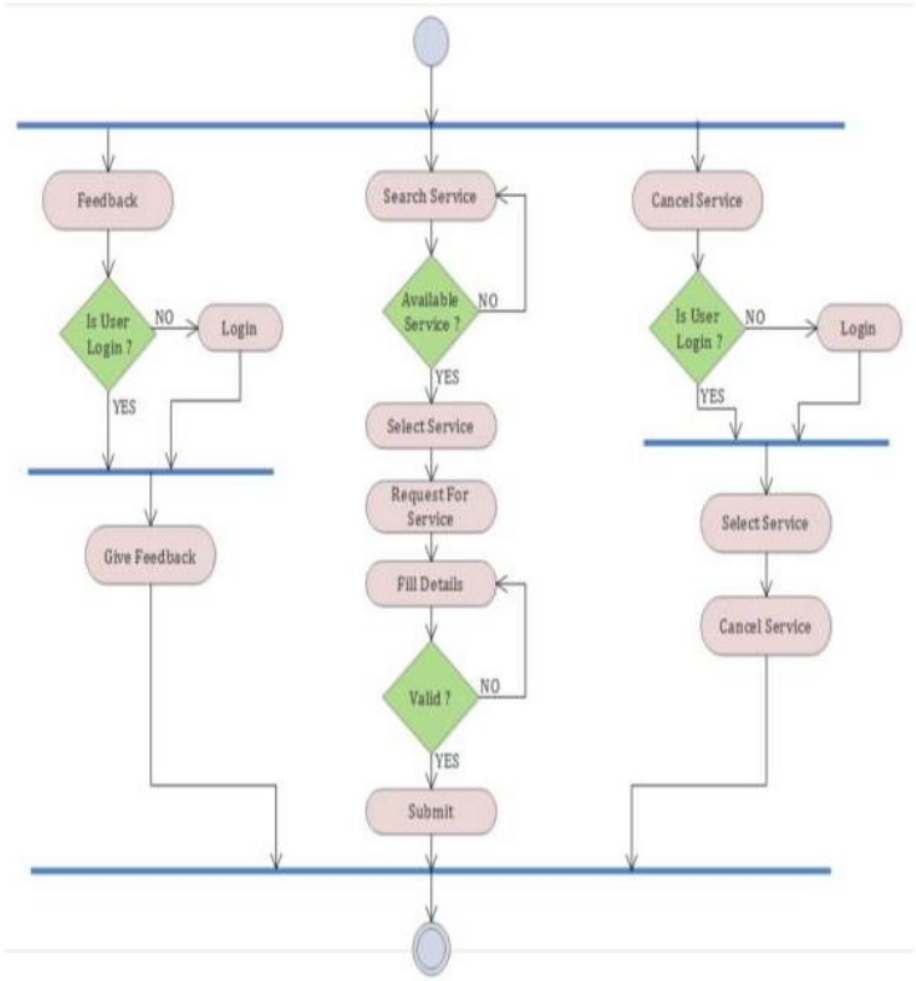

Figure 6 Customer activity

Task 03.

Design attractive user interfaces for the given scenario.

Administrative functions:

- User registration and login.

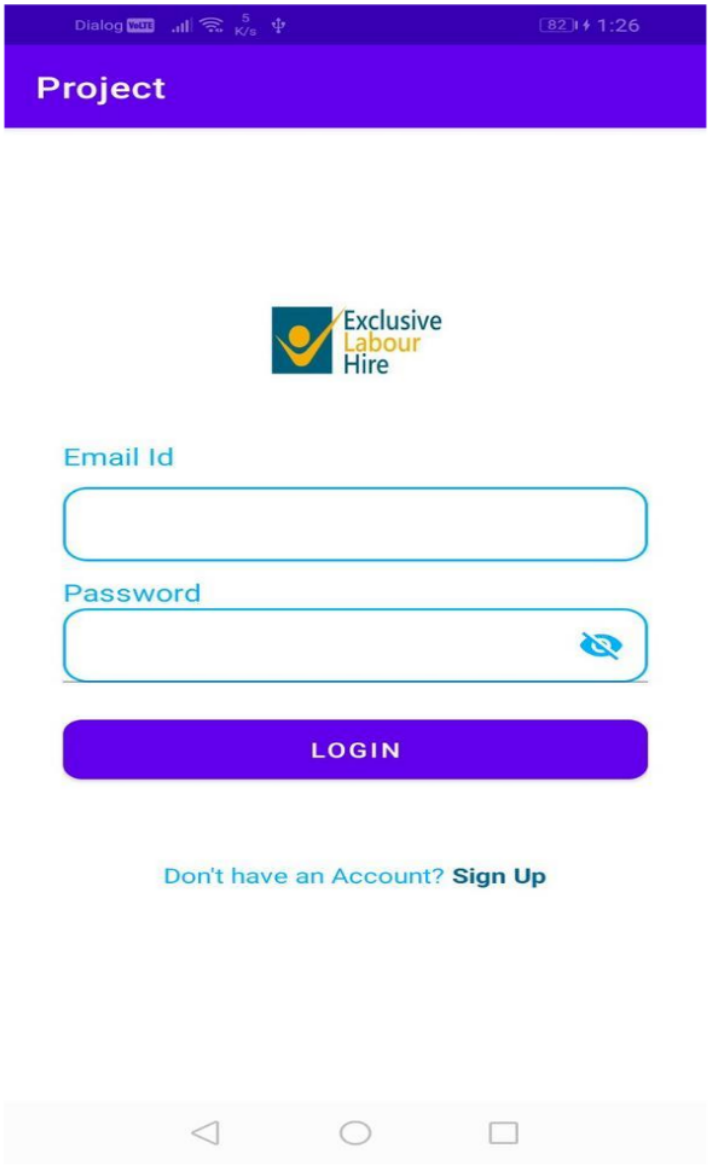

Figure 7 Login

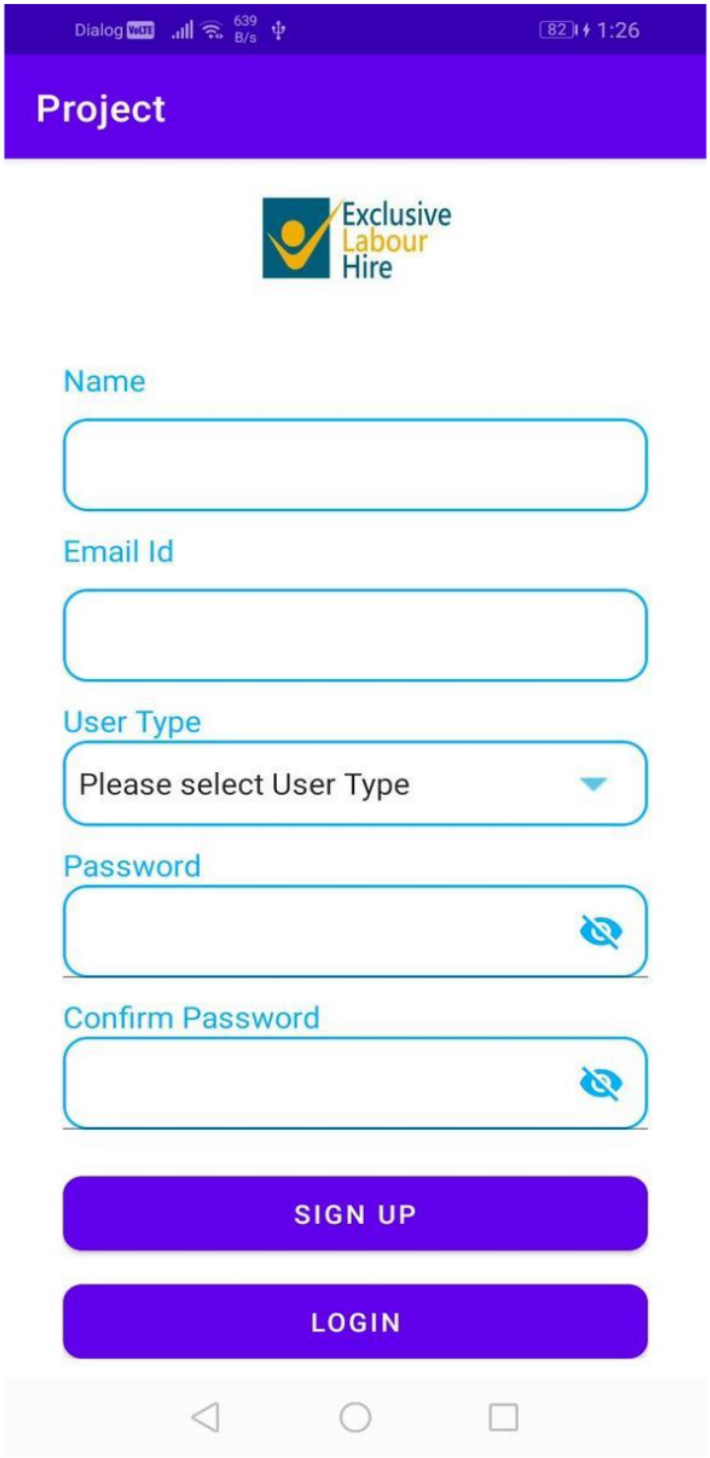

Figure 8 Registration

In the event that the user does not already have an account, he needs to create one. To do it, follow the steps that are listed below.

- On the screen where you log in, look for a button labeled "Sign Up."
- Please fill out the form, and then choose the option that says "client."
- Click the button labeled "Sign Up."

After the user registration process is finished, the user may log in using the email address and password that they registered with. There are two types of users in the system. One is customer, and the other one is cleaners. They can select their role from the drop down and register.

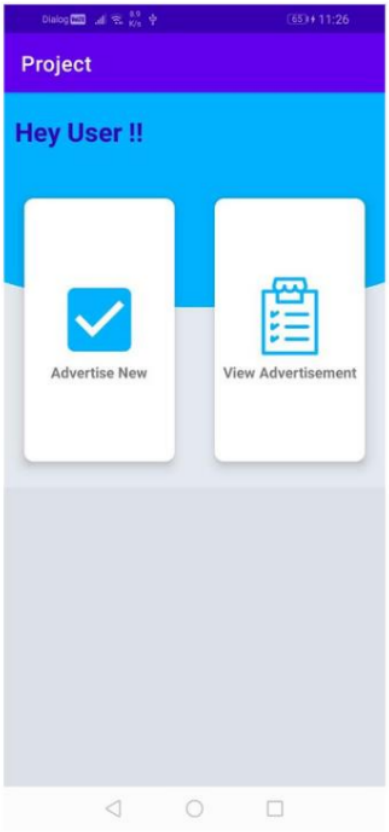

Figure 9Home Page

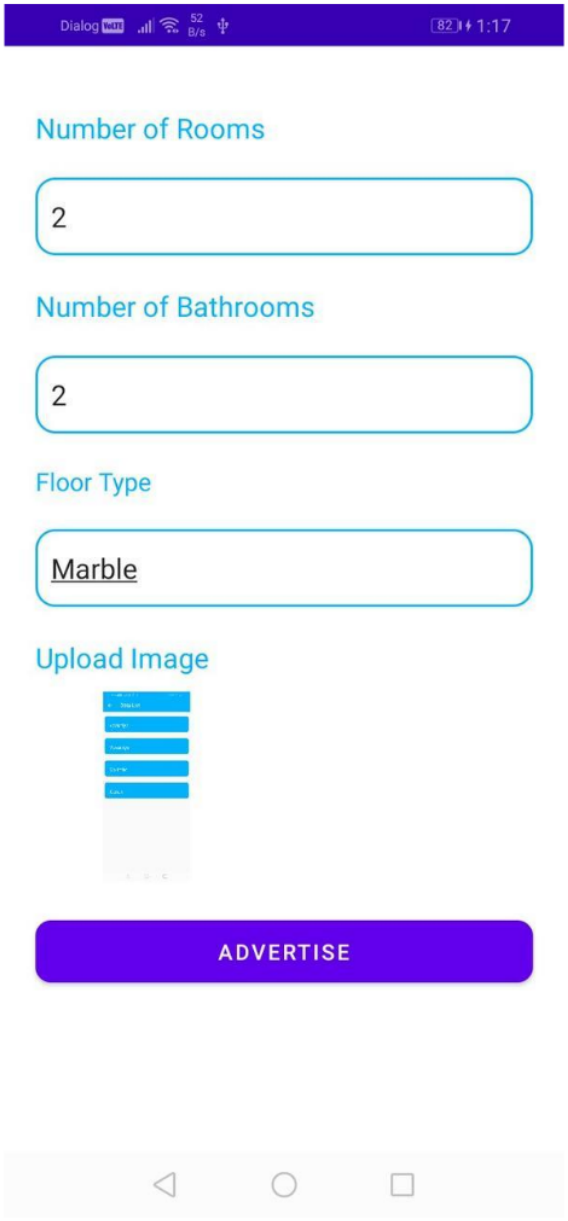

Figure 10 Advertise by Customer

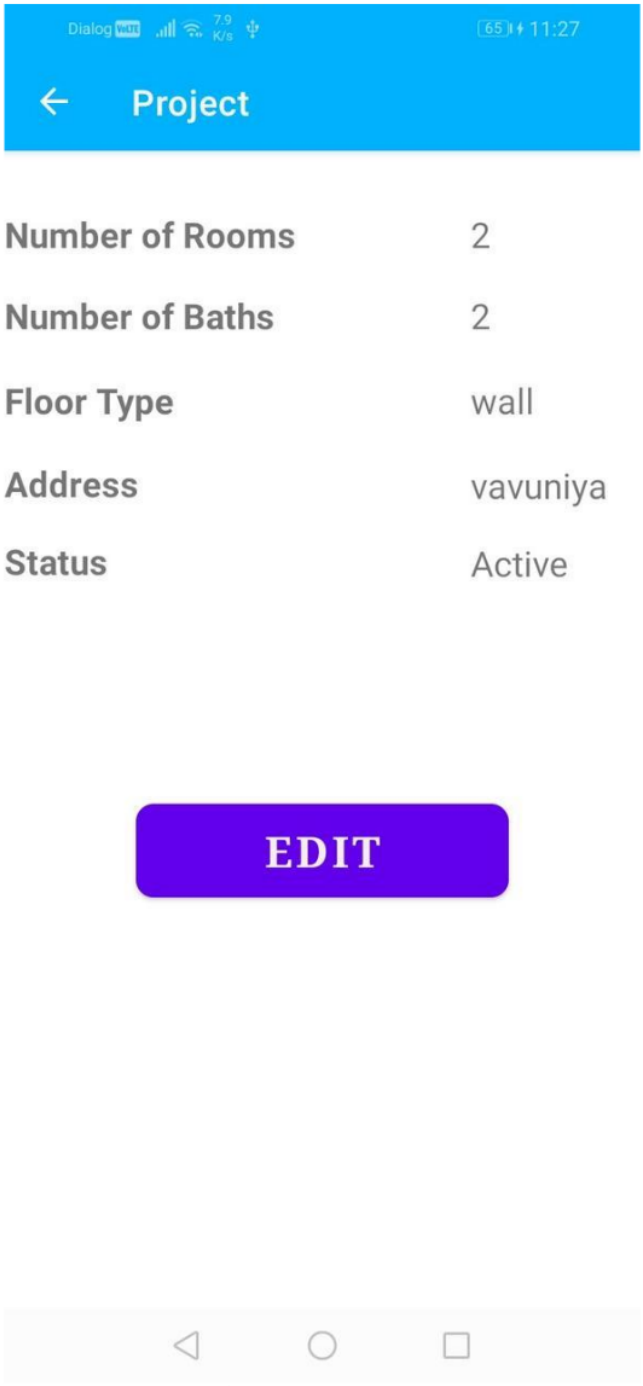

Figure 11 Customer view advertisement status

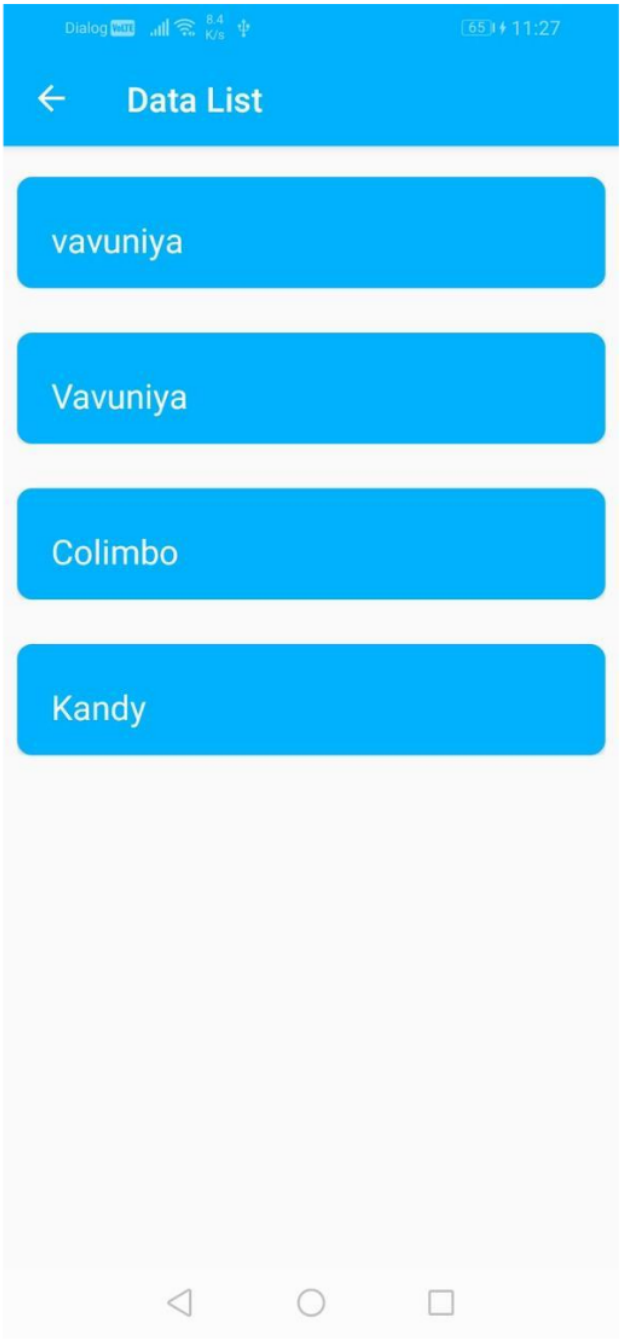

Figure 12 Cleaner view existing advertisements

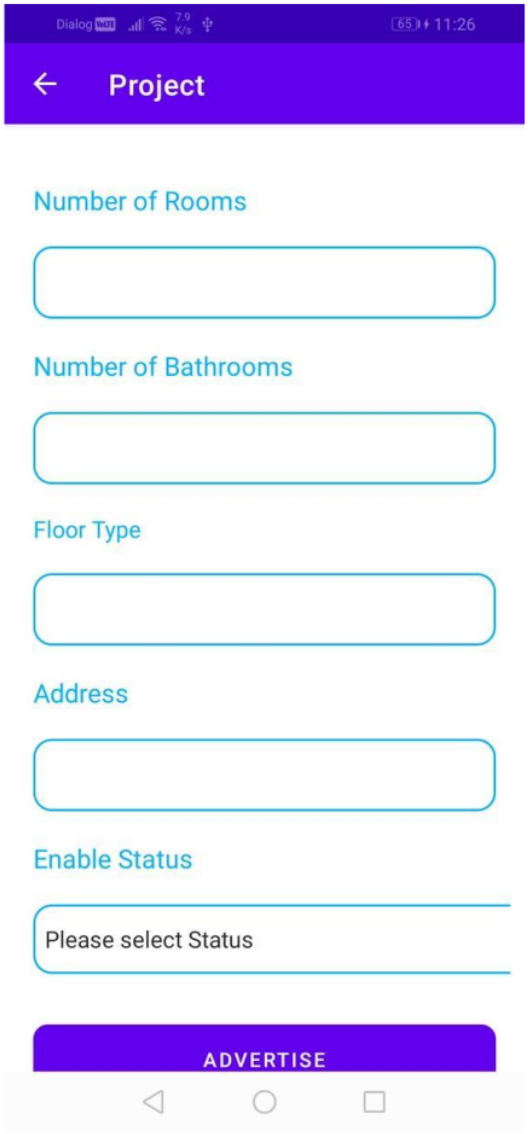

Figure 13 Advertise accept or reject update by cleaner

Task 04.

-Include test plan, test data and proper application of the test plan. Test your mobile application according to the test plan.

Test plan for the House clean application.

Test case	Expected result	Actual result	Test passed / Failed
01	Able to register as a customer	Able to register as expected	Pass
02	Able to register as a cleaner	Able to register as expected	Pass
03	Customer add advertisement	Customer able to add advertisement	Pass
04	Cleaner could be able to view the advertisement and change the update to active or not.	Cleaner can select the work and update the status	Pass
05	Able to login as Cleaner and Customer	Able to login into different dashboards	Pass
06	Customer and cleaner could be able to review each other	Both are able to give feedbacks	Pass

Conclusion.

The management of House Clean has made the decision to develop a mobile application in order to provide consumers with an easier way to book cleaners.

The system was developed using Object-Oriented Programming principles, with Java as the underlying language, and Android Studio as the primary tool for software creation and modification. Firebase is used to do the database tasks that are necessary for the data modification.

The implementation of the system was carried out based on Activities at each stage of development, and the generation of the necessary functional outputs was carried out appropriately. The development process was highly dependent on time, and the design of layouts required a significant amount of time while also requiring management of the necessary operations.

In order to provide much more user-friendly and interactive features, the database makes use of a number of different "Button click" choices, such as short press and long press. In addition, the system supports a variety of input kinds, including text and visual data, so that it may provide users the opportunity to interact with the contents of images.

For the purpose of providing users with the highest possible degree of protection, the system has been outfitted with several user authentication levels, in addition to having basic validations, password confirmations, and password encryptions enabled.

The graphical layout, which is supplied by means of colors and fonts, gives users an easy-to-use interface to work with. In addition, the system is endowed with certain specific features, such as notification messages, to offer users with the useful features of mobile apps.

The testing of the system was carried out both in the course of its creation and at the conclusion of the process. This was accomplished by distributing the system to a variety of users and collecting their comments on its performance as well as its friendliness to users.

In conclusion, it is possible to draw the conclusion that the cleaning management system that was built for the House clean has the potential to be effectively used to increase the activities of the users in a manner that is both efficient and productive.

Reference.

guru99, 2018. *guru99*. [Online]
Available at: https://www.guru99.com/java-tutorial.html

webopedia, 2018. *webopedia*. [Online]
Available at: https://www.webopedia.com/insights/mobile-os-and-different-types/