**FAISALABAD SERVICES PORTAL APPLICIATION**

The project report is submitted for the partial fulfilment of the Bachelor of Science in Software Engineering Degree to the faculty of Software Engineering.



**Supervised by:**

Dr. Muhammad Yahya Saeed

**Submitted by:**

Muhammad Hammad

Muhammad Azhar

Hassan Shahid

Ali Hassan

**DEPARTMENT OF SOFTWARE ENGINEERING GOVERNMENT COLLEGE UNIVERSITY FAISALABAD**

**JULY 2023**

**ACKNOWLEDGEMENT**

Up and above anything else, all praise to Almighty Allah alone, The Omnipresent & the Most Merciful & Compassionate. Allah is the Center of Knowledge. Allah gives a reward for every effort & he never spoils our efforts. We owe a debt of thanks to our Supervisor, **Dr. Muhammad Yahya Saeed** for his supervisor contributions, in guiding us and motivating us, at every step in the completion of or project. He helped us out each time, whenever we faced any kind of problem.

We are really into all the respondents who spared their precious time and provided the required resources helpful for our project. We also want to give SPECIAL THANKS to our family members who supported us in every way to carry out this project smoothly. Last but not least, we again thank all whom we have mentioned & whom we have not mentioned for helping us in the completion of this work.

**DECELARTION**

We hereby declare that the project entitled **“Faisalabad Services Portal Application”** being submitted by us towards the partial fulfilment of the Bachelor of Science in Software Engineering degree to the faculty of Software Engineering at Government College University, Main Campus, Faisalabad is carried by us under the supervision of **Dr. Muhammad Yahya Saeed** and have not been submitted anywhere else or for the award of any other degree and Diploma. We will be solely responsible and the university may take an action, if the information inside this report is against the above deceleration statement.

**Signature of the Candidates:**

|  |  |
| --- | --- |
| Muhammad Hammad  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Muhammad Azhar  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Hassan Shahid  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ali Hassan  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

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

**CERTIFICATE OF SUBMISSION**

This is to certify that the following candidates mentioned below have successfully completed the project entitled “Faisalabad Services Portal Application” and has been submitted to the faculty of Software Engineering, Government College University, Main Campus, Faisalabad to fulfill the significant requirement of the Bachelor of Science in Software Engineering degree.

**Project Candidates Registration Number Roll Number**

Muhammad Hammad 2019-GCUF-08036 8983

Muhammad Azhar 2019-GCUF-06661 8939

Hassan Shahid 2019-GCUF-06088 8953

Ali Hassan 2019-GCUF-08069 8973

**Project Supervisor Head of Department**

Dr. Muhammad Yahya Saeed Dr. Muhammad Awais

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

Involvement in mobile technology over multiple decades currently rule on a large market segment of 90.72% world-wide users. Instead of sitting in front of large screens, people know prefer powerful pocket screens that are almost capable of completing the such same tasks even while they walking outside. By considering this great advantage, we developed an android based service application called Faisalabad Services Portal (FSP) targeting Manchester City of Pakistan with a population of almost 3.7 million, through which excellent local service solutions will become possible. By downloading our application, the users will easily find the best solution providers (i.e, public and private businesses, marketplaces, institutes and organizations) according to their taste and needs by visiting their profiles (which include their personal details and working credentials, etc) instead of finding physically appropriate people for the task from different locations might be from different marketplaces or from different references, which consume much time as well as not any knowing about how well they work? Our application will work as a bridge between different service providers and related clients. For the development of the application we used the latest technologies presently available in the market on various sides (i.e. front-end and back-end) named Google framework called flutter, Javascript library React JS and firebase for the database. The application is available on the Play Store and providing the realtime experience to their users.

***Chapter No. 01***

***Introduction***

* 1. **Introduction**

Involvement in mobile technology over multiple decades currently rule on a large market segment of 90.72% world-wide users. Instead of sitting in front of large screens, people know prefer powerful pocket screens that are almost capable of completing the such same tasks even while they walking outside. By considering this great advantage, we developed an android based service application called Faisalabad Services Portal (FSP) targeting Manchester City of Pakistan with a population of almost 3.7 million, through which excellent local service solutions will become possible. By downloading our application, the users will easily find the best solution providers (i.e. public and private businesses, marketplaces, institutes and organizations) according to their taste and needs by visiting their profiles (which include their personal details and working credentials, etc) instead of finding physically appropriate people for the task from different locations might be from different marketplaces or from different references, which consume much time as well as not any knowing about how well they work? Our application will work as a bridge between different service providers and related clients.

* 1. **Scope of the Project**

Our application covered public to private businesses, marketplaces, institutes and organizations and their working credentials under the domain of Faisalabad City.

* 1. **Problem Statement**

There is not a piece of digitize solution is available in Faisalabad City that covers both public and private businesses, marketplaces, institutes and organizations with their working credentials, that makes a bridge between these sectors and ordinary users.

* 1. **Problem Description**

Faisalabad Services Portal (FSP) is an android based application that connects local service providers with client who want to avail best services without physically visiting different locations (just need an internet connection). Our application targets public and private businesses, marketplaces, institutes and organizations that covered the categories like pharmacy, wood workshop, real estate, emergency services to public hospital etc. Additionally, there are many more features that enhance the users experience. The importance of developing this android based application is to solve these main currently facing problems by people of Faisalabad City.

* Finding local services are headache and time consuming in busy life routine, without knowing about how well they work.
* No system is currently covering all the public and private businesses, marketplaces, institutes and organizations working credentials with their location.
  1. **Objective**

Digitizing public and private businesses, marketplaces, institutes and organizations for the people of Manchester City to communicate with each other and find the best possible solutions of their real world problems.

* 1. **Features**

Faisalabad Services Portal (FSP) provides the following features to their registered ordinary users and service providers.

* All private and public marketplaces, businesses, institutes and organizations of Manchester City under single umbrella.
* Known about the credentials of each public or private marketplaces, businesses, institutes and organizations easily at one profile click.
* Discover nearby building and places with the help of the maps and nearby filter.
* Opportunity for businesses (etc.) to provide their services.
  1. **Nature of the Project**

The project is an android-based application registered on Google Play Store and developed by using various modern technologies.

* 1. **Tools/Technology**

Faisalabad Services Portal (FSP) are developed under the following hardware and software technologies and tools.

* + 1. **Figma**

The application interfaces are designed and prototyped under a collaborative web application called Figma.

* + 1. **Dart/Flutter**

The logical functionalities of Faisalabad Services Portal (FSP) are written in dart developed by Google and the front-end of the application is written in its framework called flutter.

* + 1. **Javascript/React JS**

For admin side, to show various statistics and for services control the Javascript and its library React JS is used in an application.

* + 1. **Google Maps**

The google maps technology and its application programming interface are used in location and nearby feature of Faisalabad Services Portal (FSP).

* + 1. **Firebase**

For back-end database operations, the application used cloud-hosted NoSQL realtime database called firebase.

* 1. **Process Model**

Faisalabad Services Portal (FSP) is developed by using iterative model. In which at initial development work is conducted based on initial requirements that are clearly defined, and subsequent features are added through iterations until the final system is completed.

* 1. **Summary**

This chapter covered Faisalabad Services Portal (FSP) an android based application brief introduction, the scope of this project, why was this application necessary, its objective, the various features this application provided and nature of the project. After explaining this, the following chapter also discuss tools and technologies under which application was developed and process model through which it completed.

***Chapter No. 02***

***Requirement Specification***

* 1. **Interface Requirement**

Interface requirements are defined as those that the system must accomplish. These include the both hardware and software interface requirements. Faisalabad Services Portal (FSP) is an android based application and mostly be made up of software components, which highlights the importance of designing the interface components in such a way that are ease in interaction and well in performance. These are the following hardware and software interface requirements of our application.

* + 1. **Hardware Interface Requirement**

These are the various hardware interface requirements that are necessary for development and deployment of an application.

* + - 1. **Desktop Computer/Laptop**

Desktop computer or laptop having high end specifications are required due to modern technologies and tools (like flutter and react, etc) used in development process of an application. The minimum required specifications are 6 gigabytes of RAM, Core i5 Processor and 512 gigabytes of SSD.

* + - 1. **Android Smart Phone**

Faisalabad Services Portal (FSP) is an android based application that must deploy and run on an android based smart phone. So, an android smart phone having minimum 2 gigabytes of RAM is required with an internet connection.

* + 1. **Software Interface Requirement**

These are the following software interface requirements that are essential for the development of an application.

* + - 1. **Figma**

A collaborative web application Figma is required for designing and prototyping of Faisalabad Services Portal (FSP) because it provides various advanced features and plugins.

* + - 1. **Dart/Flutter**

An object-oriented and class-based language Dart is required to write logical functionalities of an application and its open source user interface development kit or framework flutter is to write down the front-end.

* + - 1. **Javascript/React JS**

For the development of admin side, various services control and statistics view Javascript and its powerful library React JS is required.

* + - 1. **Google Maps**

For the development of following nearby and location features, google maps and its application programing interface is required.

* + - 1. **Visual Studio Code**

As a code editor, visual studio code is required because an excellent editor makes the work one-hundred times easier and visual studio code has all the capabilities that declare it an excellent editor.

* 1. **Database Requirement**

Database is a necessary segment of an application, which provide an ability to perform various data operations at back-end. The following database is required in an application.

* + 1. **Firebase**

For back-end database operations, Firebase is required because the application used cloud-hosted NoSQL realtime database.

* 1. **Functional Requirement**

Functional requirements are actual product features or functions that developers must implement. These are the various functional requirements of Faisalabad Services Portal (FSP), an android based application.

* + 1. **Signup/Login**

The application must have signup and login panel with various required fields like email address and password for login authentication and full name, email address and password for application registration. The application must provide the facility of login and signup through google authentication.

* + 1. **Forgot Password**

The application must have forgot password panel for the case when user is not able to recall password through email address authentication process.

* + 1. **Register/Manage Service**

The application must have services dashboard, where public and private businesses, marketplaces, institutes and organizations easily register and manage their services.

* + 1. **Verified/Pending Service**

The application must have verified and pending service feature, where the service providers watch the status of their recently registered services.

* + 1. **Search Service**

The application must have search service feature from which the ordinary user easily finds the required service nearby.

* + 1. **Recommend Service**

The application must have a recommend service feature through which the users swipe down to look at the recommend services generated by an algorithm.

* + 1. **Chat Bridge**

The application must have a chatting feature, from which the ordinary users communicate with their required service providers.

* 1. **Non Functional Requirement**

Non-Functional requirements specify the quality attributes of a product and how well it will operate. These are the following non-functional requirements of an application.

* + 1. **Usability**

Faisalabad Services Portal (FSP) provide effectiveness, efficiency and the overall satisfaction of the user, when they interact with the application.

* + 1. **Scalability**

The application is scalable because of an iterative approach used while development. In future, any addition or detached of feature is easily possible.

* + 1. **Maintainability**

The application is written by using write-clean code approach. The code is well-organized according to in order structure inside different files and directories.

* + 1. **Performance**

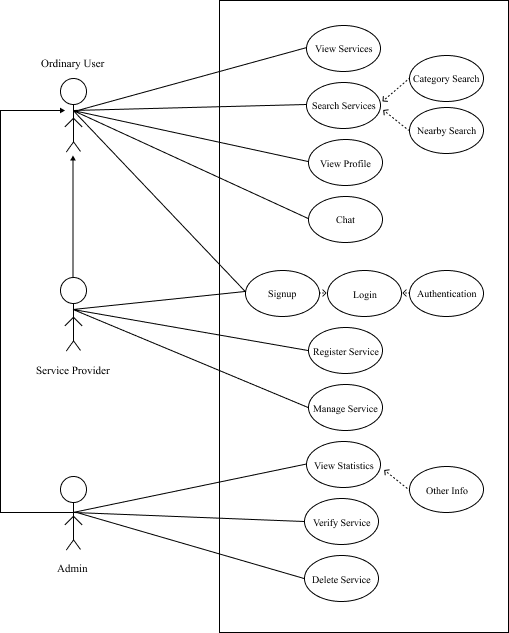
The performance of an application is smooth due to modern front-end and back-end technologies. The users observe the experience of an application overall excellent.

* + 1. **Security**

The application obeys both confidentiality, integrity and availability CIA triad. Only owner or admin can edit and delete the registered services. The application is available 24-hours, whenever the users want to use it.

* 1. **Use Case Diagram**

When a system is examined to gather its functionalities, use cases are created. Below is the following illustrative representation of application use cases.



* 1. **Use Cases Description Tables**

These are the following use cases (illustrate in an above use case diagram) description tables of Faisalabad Services Portal (FSP).

|  |  |  |  |
| --- | --- | --- | --- |
| **Signup Use Case Description Table** | | | |
| Name | Signup | | |
| ID | UC-01 | | |
| Objective | This use case defines the signup or registration process on the application. | | |
| Brief Description | The user will put the required credentials inside the fields or use authentication through Google process. | | |
| Pre-Condition | User must have to download and run application. | | |
| Post-Condition | The account is registered successfully. | | |
| Failed Condition | When required fields are empty or user not follows the conditions correctly. | | |
| Primary Actor | User | | |
| Dependency | None | | |
| Basic Flow | Steps | Action | Response |
| 01 | User download and open the application. | The system will show the panel screen. |
| 02 | The user select the signup panel. | The system will open the signup panel. |
| 03 | The user put the required credentials inside the fields and click on the signup button or just click on signup with Google. | The system authenticates all the putted information and then successfully registered the new account. |
| Alternative Flow | Errors | Action | Response |
| 01 | The required fields are not fill according to the required conditions. | The system will throw the message to fill required fields according to the required conditions. |
| 02 | The user want to register a new account through signup with Google feature and not login the Google account first in the device. | The system will throw the message to first login your Google account into the device. |

Use Case Description Table 01: Signup

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Use Case Description Table** | | | |
| Name | Login | | |
| ID | UC-02 | | |
| Objective | This use case defines the login process on the application. | | |
| Brief Description | The user write down the email address and password or use Google authentication to login in an application. | | |
| Pre-Condition | User must have to an account in an application or Google. | | |
| Post-Condition | The account is login successfully. | | |
| Failed Condition | When required fields are empty or user not write down the correct credentials. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the panel screen. |
| 02 | The user select the login panel. | The system will open the login panel. |
| 03 | The user put the email address and password inside the fields and click on the login button or just click on login with Google. | The system authenticates all the putted information from the database and then login the user into an application. |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credentials are wrong. | The system will throw the message to write down the correct credentials. |
| 02 | The user want to login through login with Google feature and not login the Google account first in the device. | The system will throw the message to first login your Google account into the device. |

Use Case Description Table 02: Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Forgot Password Use Case Description Table** | | | |
| Name | Forgot Password | | |
| ID | UC-03 | | |
| Objective | This use case defines the process when the user forget the password. | | |
| Brief Description | The user write down the email address and authentication process is done automatically through database. | | |
| Pre-Condition | User must have an account in an application. | | |
| Post-Condition | The password is forgot successfully. | | |
| Failed Condition | When required email address field is empty or user not write down the correct credential. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the panel screen. |
| 02 | The user select the forgot password panel. | The system will open the forgot password panel. |
| 03 | The user put the email address and press the forgot password button. | The system authenticates from the database. |
| 04 | The user write down the new password, confirm it and then click on the done button. | The system will save new password and login the user in application |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credential is wrong. | The system will throw the message of invalid credential and cancel the process. |

Use Case Description Table 03: Forgot Password

|  |  |  |  |
| --- | --- | --- | --- |
| **Register Service Use Case Description Table** | | | |
| Name | Register Service | | |
| ID | UC-04 | | |
| Objective | This use case defines the registration process of public and private businesses, marketplaces, institutes and organizations. | | |
| Brief Description | The public and private businesses, marketplaces, institutes and organizations write down their credentials for registration of their services. | | |
| Pre-Condition | User must have an account in an application. | | |
| Post-Condition | The service is send successfully to the application owners for further verification. | | |
| Failed Condition | When required field is empty or not obey the pre-defined conditions. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The user select the services form option at bottom navigation bar. | The system will open the services form page. |
| 04 | The user write down the required credentials and press the submit button. | The system will send the service form to the owners of the application for further verification and the service will show inside pending tab. |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credentials are missing. | The system will throw the message to must fill the required fields. |
| 02 | The credentials are not according to the pre-defined conditions | The system will tell user to fill the fields correctly, |

Use Case Description Table 04: Register Service

|  |  |  |  |
| --- | --- | --- | --- |
| **Update Service Use Case Description Table** | | | |
| Name | Update Service | | |
| ID | UC-05 | | |
| Objective | This use case defines the credentials updating process of public and private businesses, marketplaces, institutes and organizations. | | |
| Brief Description | The public and private businesses, marketplaces, institutes and organizations update their credentials after registration of their services. | | |
| Pre-Condition | User must have an account in an application with successfully verified service profile. | | |
| Post-Condition | The service credentials are successfully updated. | | |
| Failed Condition | When required field is empty or not obey the pre-defined conditions. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case with Register Service Use Case and Verified or Pending Service Use Case. | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The user select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | The user select the service profile wants to update. | The system will open the services form. |
| 05 | The user change the credentials want to update and press the update button. | The system will successfully updates the changed information. |
| Alternative Flow | Errors | Action | Response |
| 01 | The required credentials are missing. | The system will throw the message to must fill the required fields. |
| 02 | The credentials are not according to the pre-defined conditions | The system will tell user to fill the fields correctly, |

Use Case Description Table 05: Update Service

|  |  |  |  |
| --- | --- | --- | --- |
| **Delete Service Use Case Description Table** | | | |
| Name | Delete Service | | |
| ID | UC-06 | | |
| Objective | This use case defines the public and private businesses, marketplaces, institutes and organizations deletion process. | | |
| Brief Description | The public and private businesses, marketplaces, institutes and organizations delete their services after successfully registration. | | |
| Pre-Condition | User must have an account in an application with successfully verified service profile. | | |
| Post-Condition | The service credentials are successfully deleted. | | |
| Failed Condition | When user undo the process within 14 days. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case with Register Service Use Case and Verified or Pending Service Use Case. | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The user select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | The user long press the service profile wants to delete. | The system will open the delete service popup box. |
| 05 | The user select the yes button for deletion. | The system will successfully deletes the service. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 06: Delete Service

|  |  |  |  |
| --- | --- | --- | --- |
| **Verified or Pending Service Use Case Description Table** | | | |
| Name | Verified or Pending Service | | |
| ID | UC-07 | | |
| Objective | This use case defines the verified and pending process of service profile. | | |
| Brief Description | The public and private businesses, marketplaces, institutes and organizations watch their verified and pending services. | | |
| Pre-Condition | User must have an account in an application and submit the service profile. | | |
| Post-Condition | The service profile is under pending, verified or rejected. | | |
| Failed Condition | When user cancel the request. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case with Register Service Use Case. | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The user select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | Here user watch their verified or pending service profile by click on each of the tab. | The system will open the verified or pending tab according to the user click. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 07: Verified or Pending Service

|  |  |  |  |
| --- | --- | --- | --- |
| **View Service Use Case Description Table** | | | |
| Name | View Service | | |
| ID | UC-08 | | |
| Objective | This use case defines the view service process. | | |
| Brief Description | The user views the public and private businesses, marketplaces, institutes and organizations. | | |
| Pre-Condition | User must have an account and open the application | | |
| Post-Condition | The services are displayed successfully. | | |
| Failed Condition | No internet connection. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user views all the services profiles under their specific categories. | The system will show the services profiles under their specific categories. |
| Alternative Flow | Error | Action | Response |
| 01 | The user opens the application without internet connection. | The system will throw the message, no internet connection. |

Use Case Description Table 08: View Service

|  |  |  |  |
| --- | --- | --- | --- |
| **Search Service Use Case Description Table** | | | |
| Name | Search Service | | |
| ID | UC-09 | | |
| Objective | This use case defines the search service process of public and private businesses, marketplaces, institutes and organizations. | | |
| Brief Description | The user will search the public and private businesses, marketplaces, institutes and organizations according to their categories or nearby feature. | | |
| Pre-Condition | User must have an account and open the application. | | |
| Post-Condition | The filtered results shown successfully. | | |
| Failed Condition | When required service is not available. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user searches the require services profiles by write down the category, apply category filter or nearby feature. | The system will filter the required services profiles according to the feature and show them. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 09: Search Service

|  |  |  |  |
| --- | --- | --- | --- |
| **Recommended Services Use Case Description Table** | | | |
| Name | Recommended Services | | |
| ID | UC-10 | | |
| Objective | This use case defines the recommended service process of an application. | | |
| Brief Description | The user will swipe through services the application’s algorithm recommended at home page. | | |
| Pre-Condition | User must have an account and open the application. | | |
| Post-Condition | The recommend services are shown successfully. | | |
| Failed Condition | No internet connection. | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen with the recommend services feature through an algorithm. |
| Alternative Flow | Errors | Action | Response |
| 01 | When user open the application without internet connection. | The system will throw the message, no internet connection. |

Use Case Description Table 10: Recommended Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Chat Bridge Use Case Description Table** | | | |
| Name | Chat Bridge | | |
| ID | UC-11 | | |
| Objective | This use case defines the text communication process between the ordinary user and service provider. | | |
| Brief Description | The ordinary user will communicate through chat bridge with required service provider. | | |
| Pre-Condition | User must have an account and open the application with required service provider profile. | | |
| Post-Condition | The message send successfully. | | |
| Failed Condition | None | | |
| Primary Actor | User | | |
| Dependency | Signup Use Case or Login Use Case with Register Service Use Case and Verified or Pending Service Use Case. | | |
| Basic Flow | Steps | Action | Response |
| 01 | The user open the application. | The system will show the home screen. |
| 02 | The user select the required service profile and click on the chat button. | The system will establish a connection between the ordinary user and service provider. |
| 03 | The user send the message on the other side. | The system will show the message to the service provider. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 11: Chat Bridge

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin Panel Use Case Description Table** | | | |
| Name | Admin Panel | | |
| ID | UC-12 | | |
| Objective | This use case defines the admin panel process. | | |
| Brief Description | The owners of the application see the statistics, verify or reject the submitted service as well as delete the verified service. | | |
| Pre-Condition | The owners must login into the admin panel. | | |
| Post-Condition | The required action is performed successfully. | | |
| Failed Condition | None | | |
| Primary Actor | Owner | | |
| Dependency | None | | |
| Basic Flow | Steps | Action | Response |
| 01 | The owners of the application open the admin panel. | The system will show the home screen of the admin panel. |
| 02 | The owners review the pending or verified services profiles by clicking on each of them. | The system show the pending or verified services profiles. |
| 03 | The owners verified or rejected the pending service profiles. | The system will verify or reject the services profiles according to what the owners want. |
| 04 | The owners also review the violation and delete the service. | The system will successfully delete the service. |
| Alternative Flow | Errors | Action | Response |
| None | None | None |

Use Case Description Table 12: Admin Panel

* 1. **Summary**

This chapter discussed the requirement specification of the application, which includes interface requirements (hardware and software interface requirements), database requirements, functional requirements and non-functional requirements. It also talks over use case diagram and description table of each use case.

***Chapter No. 03***

***System Design and UML Diagrams***

* 1. **System Design**

System design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the requirement as described in software requirement specification (SRS).[1] Before development and implementation of Faisalabad Services Portal (FSP), we select the methodology, design application interface prototype using Figma. Furthermore, make its activity, state machine, sequence and deployment architectures by illustrated various diagrams that discussed below in detail.

* 1. **Design Methodology**

The application is designed using bottom up methodology, in which we solved smaller problems and integrate it as complete. This methodology is really suitable for object oriented programming approach and as we known earlier our application is build by using Flutter Framework that supported Dart an object oriented programming language.

* 1. **Application Interface Design**

In designing, application interface is a significant segment. The interaction of the user with an application is important and a non-conformed design leave a bad impression. Taking everything in mind, we designed the interface of Faisalabad Services Portal (FSP) as much interactive possible and user-friendly by using Figma, a collaborative web application for interface design. These are the following prototype images of the application.

* 1. **Unified Modeling Language Diagrams**

These are the following unified modeling language (UML) diagrams used to visually represent the architecture design of an android based application called Faisalabad Services Portal (FSP).

* + 1. **Activity Diagram**

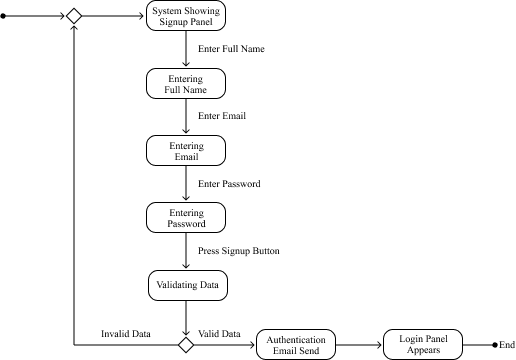
This is essentially a flowchart that represent the activities performed by the application. The illustration of the following diagram is given below.

* + 1. **State Machine Diagram**

This is almost similar to the activity diagram but describes in-depth all the states and transitions for a single object. It represents application behavior upon the user and other external actions. These are the following illustrative state machine diagrams of the application.

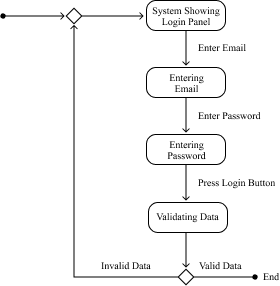
* + - 1. **Signup State Machine Diagram**

Given below is the following illustrative representation of signup panel state machine diagram.



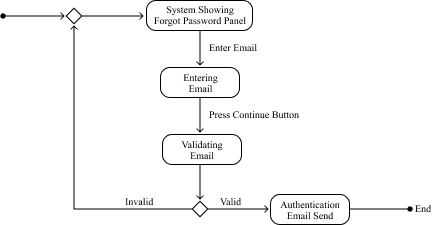
* + - 1. **Login State Machine Diagram**

Given below is the following illustrative representation of login panel state machine diagram.



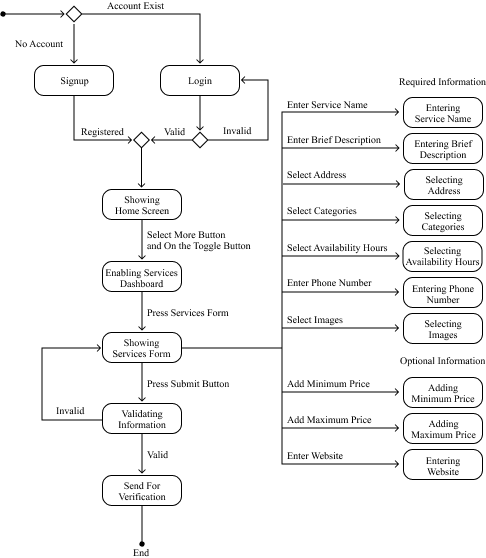
* + - 1. **Forgot Password State Machine Diagram**

Given below is the following illustrative representation of forgot password state machine diagram.



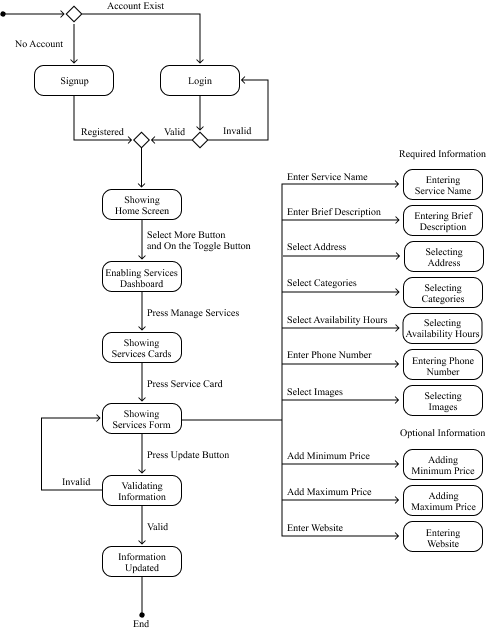
* + - 1. **Register Service State Machine Diagram**

Given below is the following illustrative representation of register service state machine diagram.



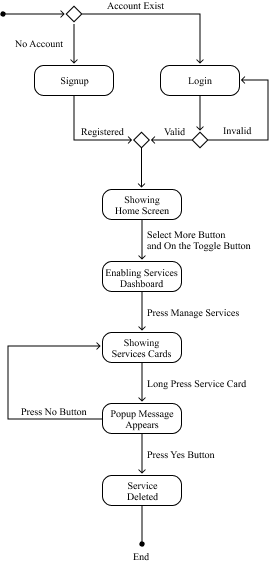
* + - 1. **Update Service State Machine Diagram**

Given below is the following illustrative representation of update service state machine diagram.



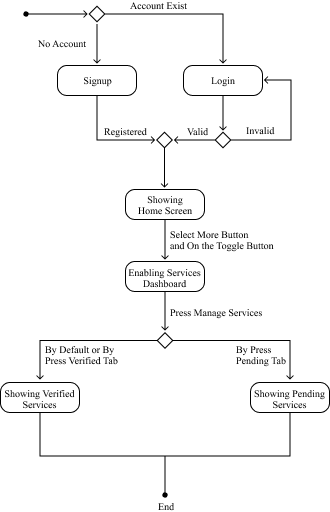
* + - 1. **Delete Service State Machine Diagram**

Given below is the following illustrative representation of delete service state machine diagram.



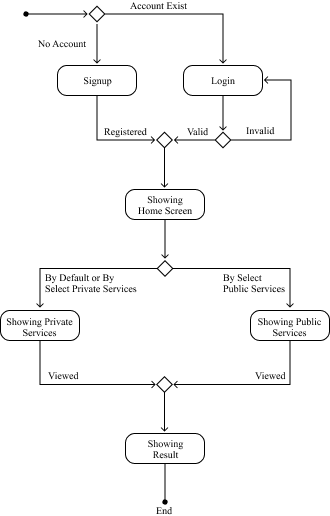
* + - 1. **Verified or Pending Service State Machine Diagram**

Given below is the following illustrative representation of verified or pending service state machine diagram.



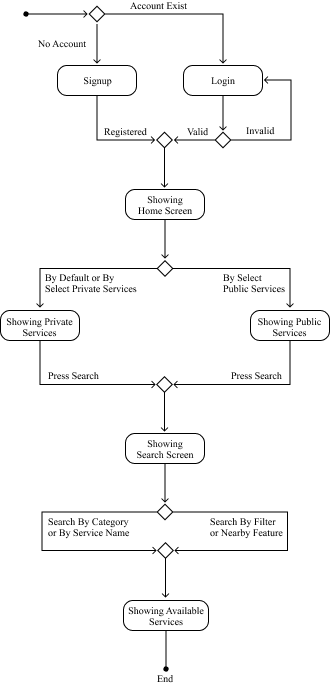
* + - 1. **View Service State Machine Diagram**

Given below is the following illustrative representation of view service state machine diagram.



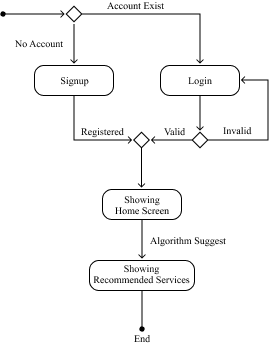
* + - 1. **Search Service State Machine Diagram**

Given below is the following illustrative representation of search service state machine diagram.



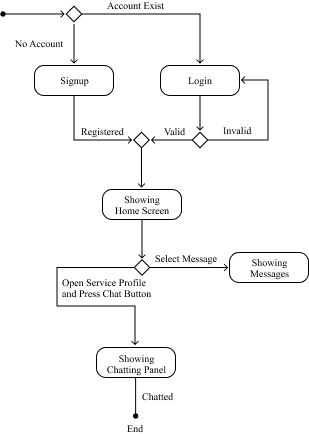
* + - 1. **Recommended Services State Machine Diagram**

Given below is the following illustrative representation of recommended services state machine diagram.



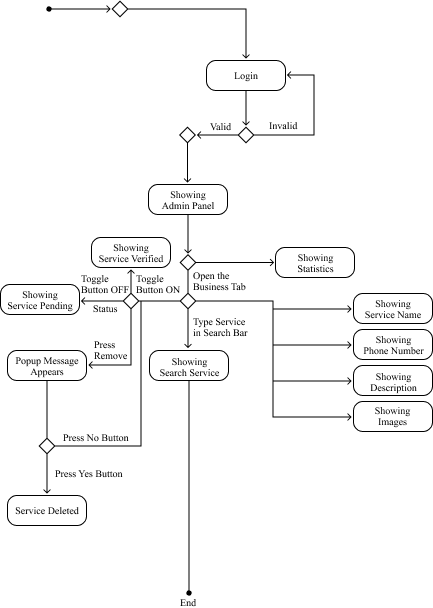
* + - 1. **Chat Bridge State Machine Diagram**

Given below is the following illustrative representation of chat bridge state machine diagram.



* + - 1. **Admin Panel State Machine Diagram**

Given below is the following illustrative representation of admin panel state machine diagram.

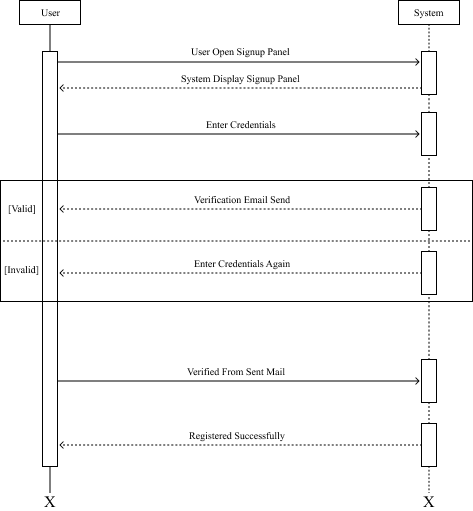


* + 1. **Sequence Diagram**

This is an interaction diagram that represents the order collection of objects working together as well as visually representation of the requirement uses cases previously discussed. These are the various sequence diagrams illustrate below.

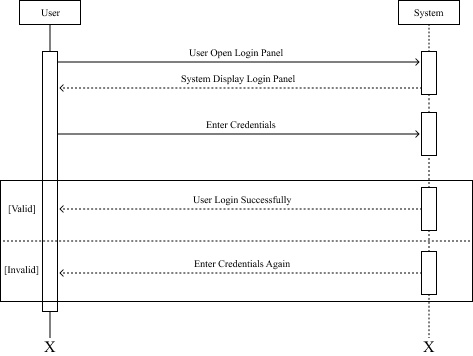
* + - 1. **Signup Sequence Diagram**

Given below is the following illustrative representation of signup panel sequence diagram.



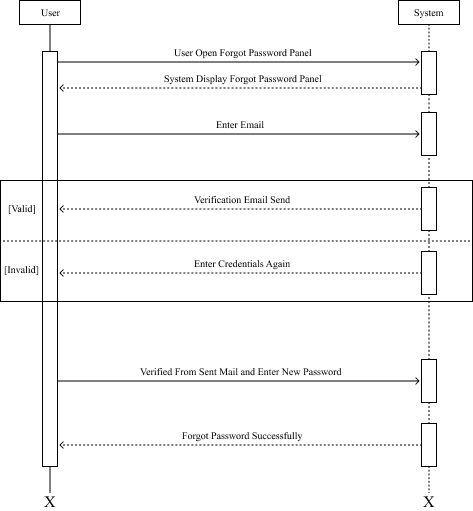
* + - 1. **Login Sequence Diagram**

Given below is the following illustrative representation of login panel sequence diagram.



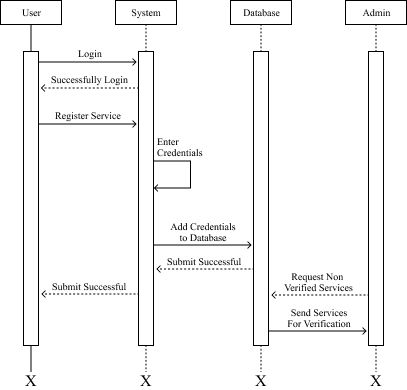
* + - 1. **Forgot Password Sequence Diagram**

Given below is the following illustrative representation of forgot password sequence diagram.



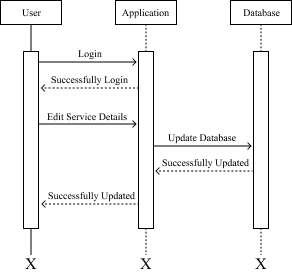
* + - 1. **Register Service Sequence Diagram**

Given below is the following illustrative representation of register service sequence diagram.



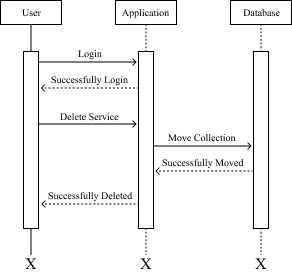
* + - 1. **Update Service Sequence Diagram**

Given below is the following illustrative representation of update service sequence diagram.



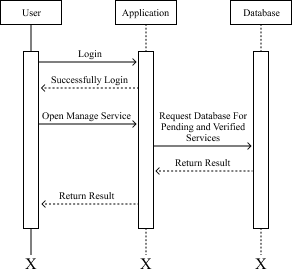
* + - 1. **Delete Service Sequence Diagram**

Given below is the following illustrative representation of delete service sequence diagram.



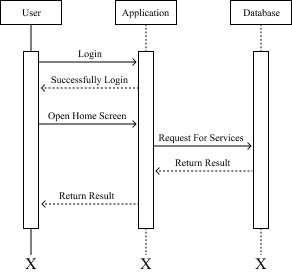
* + - 1. **Verified or Pending Service Sequence Diagram**

Given below is the following illustrative representation of verified or pending service sequence diagram.



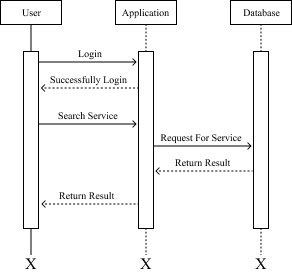
* + - 1. **View Service Sequence Diagram**

Given below is the following illustrative representation of view service sequence diagram.



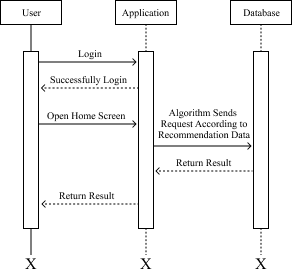
* + - 1. **Search Service Sequence Diagram**

Given below is the following illustrative representation of search service sequence diagram.



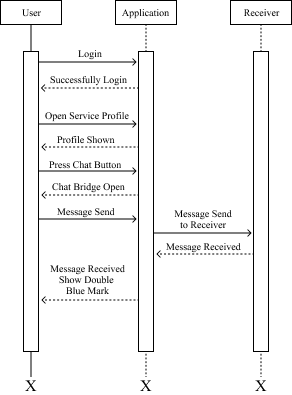
* + - 1. **Recommended Services Sequence Diagram**

Given below is the following illustrative representation of recommended services sequence diagram.



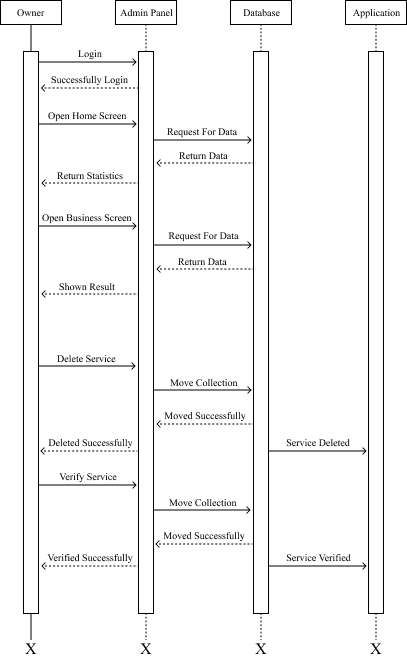
* + - 1. **Chat Bridge Sequence Diagram**

Given below is the following illustrative representation of chat bridge sequence diagram.



* + - 1. **Admin Panel Sequence Diagram**

Given below is the following illustrative representation of admin panel sequence diagram.



* 1. **System Design View Perspective Models**

There are two main types of system design models in the perspective of the view. The logical design view and the physical design view. The further explanation of these view with respect to the application is discussed below.

* + 1. **Logical Design View**

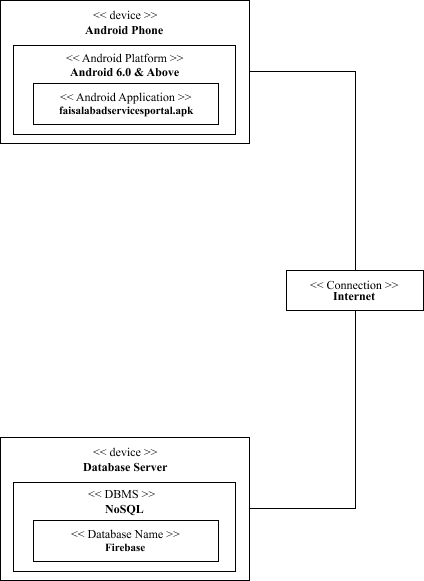
The application is used Firebase, a cloud-hosted NoSQL realtime database that stored data in the form of Javascript Object Notation (JSON). That’s why the logical design view is not possible because it consists of class and entity relationship diagrams.

* + 1. **Physical Design View**

This design view represents the end deployment of the application. It helps the engineers as a visual guide to deployed the system accordingly. In this the deployment diagram is used which shows the execution architecture with software and hardware of the application.

* + - 1. **Deployment Diagram**

Given below is the following deployment illustration of the Faisalabad Services Portal (FSP), an android based application.



* 1. **Summary**

This chapter talks about the system design, the methodology used in the designing process, the application interface prototype and various unified modeling language diagrams like activity, state machine and sequence diagrams. Furthermore, the chapter also discussed about the system design view perspective models like logical view and physical view models with deployment diagram.

***Chapter No. 04***

***Implementation***

* 1. **Module**

This segment shows the integration of the modules with each other to execute the functionalities and workflow of the application. These are the following module wise description.

* + 1. **Signup Module**

The account registration process is done through this module, where the users write their credentials required for the registration or use signup with Google feature.

* + 1. **Login Module**

The account authentication process is done through this module, where user writes email address and password of already registered account or use login with Google feature.

* + 1. **Forgot Password Module**

The forgot password authentication process is done through this module, where user writes email address for automatically authentication to write the new password.

* + 1. **Home Service Module**

The complete application is called through this module, where all services according to their categories and recommendations are shown.

* + 1. **Register Service Module**

The registration of public and private businesses, marketplaces, institutes and organizations is done through this module, where user writes the details of their service and submit it for further verification.

* + 1. **Update Service Module**

The public and private businesses, marketplaces, institutes and organizations details are updating through this module, where user can update the credentials of already registered service.

* + 1. **Delete Service Module**

The deletion process of already registered services are done through this module, where users delete their service by long press on their existing service.

* + 1. **Pending or Verified Service Module**

The pending or verified services are shown through this module, where users can watch their pending or verified services.

* + 1. **View Service Module**

All services portfolios according to their categories are control through this module, where user can view required service portfolio with ease.

* + 1. **Search Service Module**

The concatenation of services and complete searching process are down through this module, where user can use category or nearby feature to search the required service.

* + 1. **Recommend Service Module**

The recommendation services are control through this module, where recommended services with the use of an algorithm are shown on the home screen.

* + 1. **Chat Bridge Module**

The communication between ordinary user and service provider are done and control through this module, which attach with view service module.

* + 1. **Admin Panel Module**

The complete application is control through this module, where the owners saw the statistical information of their application as well as delete and verified the services.

* 1. **Framework**

Below was the given framework description that take part in the building of the application.

* + 1. **Flutter**

This is an open source framework build by Google for the development of cross platform applications from a single codebase. The front-end and logical operations of the application is developed through this user interface software development kit.

* 1. **Library**

This was the following library description that participate in the workflow of the application.

* + 1. **React JS**

This is an open source Javascript library used for constructing interactive interfaces and web applications. In the building of an application this library is used in admin panel, where all statistical information and further controls are given to the owners.

* 1. **Database**

This was the following database used at the back-end of the application, Faisalabad Services Portal (FSP).

* + 1. **Firebase**

This is a cloud based realtime no structure query language (NoSQL) database provided by Google, that stored data in Javascript Object Notation (JSON) format and synchronize it between the users realtime. This was used in the application as back-end database.

* 1. **External Application Programming Interface (EAPI)**

This was the following external application programing interface (API) used in the application.

* + 1. **Google Maps API**

This embed application programming interface (API) enables to portray and customize interactive maps on the platforms. This was used in the nearby and location feature of the application.

* 1. **Summary**

This chapter explained the modules that integrates with each other to functional the application. Furthermore, it discussed the library, framework, database and external application programming interface (EAPI) that participate in the workflow of the application.

***Chapter No. 05***

***System Testing***

* 1. **Testing**

Essential part of the application development is testing before release it in the public. These are the following subject of testing, we used after the development of an android based application called Faisalabad Services Portal (FSP).

* + 1. **Functional Testing**

In this subject, we test the application according to does its meet the exact need, what it’s supposed to do? The application main objective and all functional requirements are fulfilled after this testing.

* + 1. **Performance Testing**

In this subject, we test the efficiency and quickness of the application. After the testing process, we conclude the overall efficiency and timely response of the application is very good.

* + 1. **Compatibility Testing**

In this subject, we test the compatibility of the application on android based smartphones currently available in the market. The result is positive and also work well on the application programming interface level.

* 1. **Testing Methodology**

For android based application called Faisalabad Services Portal (FSP), we used the instrumented testing methodology as post-development. According to this methodology, the application is run on an android device either physical or emulated as test application. We launched an application and interact with it for user interface testing and also testing the logical operations are working well. Furthermore, we also used unit, integration and end-to-end testing methodologies under development or sometime at post-development phases (when more features are integrated).

* 1. **Test Cases**

In software engineering, a test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular [software testing](https://en.wikipedia.org/wiki/Software_testing) objective, such as to exercise a particular program path or to verify compliance with a specific requirement. [2] These are the following test cases given below to test the functional working of the application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Signup Test Case** | | | | | |
| Name | Signup | | | | |
| ID | TC-01 | | | | |
| Use Case Reference | UC-01 | | | | |
| Objective | Check signup module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on signup panel. | | | | |
| Pre-Requisite | The user download and open the application. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press signup button | N/A | Input required | Input required |
| Invalid syntax | Enter credential with valid syntax | Enter credential with valid syntax |
| Valid input | Home screen | Home screen |
| 02 | Press signup with Google | No Google account login in device | Redirect to the Google account login page | Redirect to the Google account login page |
| Google account already login in device | Home Screen | Home Screen |
| Status | Approved | | | | |

Table 01: Signup Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Login Test Case** | | | | | |
| Name | Login | | | | |
| ID | TC-02 | | | | |
| Use Case Reference | UC-02 | | | | |
| Objective | Check login module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on login panel. | | | | |
| Pre-Requisite | The user must be registered already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press login button | N/A | Input required | Input required |
| Invalid syntax | Enter credential with valid syntax | Enter credential with valid syntax |
| Invalid Input | Email or password invalid | Email or password invalid |
| Valid input | Home screen | Home screen |
| 02 | Press login with Google | No Google account login in device | Redirect to the Google account login page | Redirect to the Google account login page |
| Google account already login in device | Home Screen | Home Screen |
| Status | Approved | | | | |

Table 02: Login Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Forgot Password Test Case** | | | | | |
| Name | Forgot Password | | | | |
| ID | TC-03 | | | | |
| Use Case Reference | UC-03 | | | | |
| Objective | Check forgot password module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on forgot password panel. | | | | |
| Pre-Requisite | The user must be registered already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press next button | N/A | Input required | Input required |
| Invalid syntax | Enter credential with valid syntax | Enter credential with valid syntax |
| Valid input | Enter new password screen | Enter new password screen |
| 02 | Press done button | N/A | Input required | Input required |
| Valid input | Home Screen | Home Screen |
| Status | Approved | | | | |

Table 03: Forgot Password Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Register Service Test Case** | | | | | |
| Name | Register Service | | | | |
| ID | TC-04 | | | | |
| Use Case Reference | UC-04 | | | | |
| Objective | Check register service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on services form. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press submit button | N/A | Input required | Input required |
| Invalid syntax | Enter credential with valid syntax | Enter credential with valid syntax |
| Valid input | Service submitted and show in the pending tab | Service submitted and show in the pending tab |
| Status | Approved | | | | |

Table 04: Register Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Update Service Test Case** | | | | | |
| Name | Update Service | | | | |
| ID | TC-05 | | | | |
| Use Case Reference | UC-05 | | | | |
| Objective | Check update service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on update service form. | | | | |
| Pre-Requisite | The user must have a registered and verified service. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press update button | Invalid syntax | Enter credential with valid syntax | Enter credential with valid syntax |
| Valid input | Credentials updated successfully | Credentials updated  successfully |
| Status | Approved | | | | |

Table 05: Update Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delete Service Test Case** | | | | | |
| Name | Delete Service | | | | |
| ID | TC-06 | | | | |
| Use Case Reference | UC-06 | | | | |
| Objective | Check delete service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user open manage service panel. | | | | |
| Pre-Requisite | The user must have a registered and verified service. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Long press the service card | Yes | Service deleted successfully | Service deleted  successfully |
| No | Cancel | Cancel |
| Status | Approved | | | | |

Table 06: Delete Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Verified or Pending Service Test Case** | | | | | |
| Name | Verified or Pending Service | | | | |
| ID | TC-07 | | | | |
| Use Case Reference | UC-07 | | | | |
| Objective | Check verified or pending service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on manage service panel. | | | | |
| Pre-Requisite | The user must submit the service. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Click on the verified tab | None | Show verified service | Show verified service |
| 02 | Click on the pending tab | None | Show pending service | Show pending service |
| Status | Approved | | | | |

Table 07: Verified or Pending Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **View Service Test Case** | | | | | |
| Name | View Service | | | | |
| ID | TC-08 | | | | |
| Use Case Reference | UC-08 | | | | |
| Objective | Check view service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on the home screen. | | | | |
| Pre-Requisite | The user must have a registered account already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open the home screen | None | Show the services | Show the services |
| Status | Approved | | | | |

Table 08: View Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Search Service Test Case** | | | | | |
| Name | Search Service | | | | |
| ID | TC-09 | | | | |
| Use Case Reference | UC-09 | | | | |
| Objective | Check search service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on the home screen. | | | | |
| Pre-Requisite | The user must be registered already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press on the search | N/A | Input required | Input required |
| Invalid input | Relevant service is not available | Relevant service is not available |
| Valid input | Service is shown | Service is shown |
| Status | Approved | | | | |

Table 09: Search Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Recommended Service Test Case** | | | | | |
| Name | Recommended Service | | | | |
| ID | TC-10 | | | | |
| Use Case Reference | UC-10 | | | | |
| Objective | Check recommended service module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on the home screen. | | | | |
| Pre-Requisite | The user must be registered already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open the home screen | None | Show the recommended service according to the algorithm | Show the recommended service according to the algorithm |
| Status | Approved | | | | |

Table 10: Recommended Service Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chat Bridge Test Case** | | | | | |
| Name | Chat Bridge | | | | |
| ID | TC-11 | | | | |
| Use Case Reference | UC-11 | | | | |
| Objective | Check chat bridge module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The user is on the required service profile. | | | | |
| Pre-Requisite | The user must be registered already. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Press on the chat button | None | Open message panel | Open message panel |
| 02 | Click on the send message button | N/A | Text required | Text required |
| Input text or emoji | Message is send successfully | Message is send successfully |
| 03 | Single tick on the sent message | None | Message is sent successfully but not received by the required service provider | Message is sent successfully but not received by the required service provider |
| 04 | Double tick on the sent message | None | Message is received by the required service provider | Message is received by the required service provider |
| 05 | Double blue tick on the sent message | None | Message is seen by the required service provider | Message is seen by the required service provider |
| Status | Approved | | | | |

Table 11: Chat Bridge Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Admin Panel Test Case** | | | | | |
| Name | Admin Panel | | | | |
| ID | TC-12 | | | | |
| Use Case Reference | UC-12 | | | | |
| Objective | Check admin panel module is working properly. | | | | |
| Environment | Android Device | | | | |
| Assumption | The owners are on the admin panel. | | | | |
| Pre-Requisite | The owners must be login into the admin panel and some services are submit for verification or verified. | | | | |
| Test Procedure | Steps | Action | Input | Expected Result | Actual Result |
| 01 | Open the admin panel home screen | None | All the information is show | All the Information is show |
| 02 | Click on the verify service button | Verify service button clicked | Service is verified and shown in the user verified tab. | Service is verified and shown in the user verified tab |
| 03 | Click on the delete service button | Delete service button clicked | Service is deleted successfully | Service is deleted successfully |
| Status | Approved | | | | |

Table 12: Admin Panel Test Case

* 1. **Summary**

This chapter discussed the testing with its subjects as well as testing methodologies used in the testing of an application. Furthermore, we talk about the test cases and portray each of the test case in the form of table according to the functionality.

***Chapter No. 06***

***Conclusion and Future Work***

* 1. **Application Overview**

Third largest city of Pakistan according to the population growth rate, the Manchester City where the industrial backbone of the country exists, have not a single product that connects the ordinary people with the public and private businesses, marketplaces, institutes and organizations. Our android based application Faisalabad Services Portal (FSP) is the solution of this realtime problem, where public and private businesses, marketplaces, institutes and organizations register their services and ordinary users access those services with nearby and other interesting features. Additionally, the application as well allow the ordinary users to communicate with their required service providers using Chat Bridge. Faisalabad Services Portal (FSP) is presently available free on the Play Store and you can download it to add element of ease in your routine lifestyle.

* 1. **Milestones Achieved**
* These are the following milestones achieved by an android based application, Faisalabad Services Portal (FSP).
* The public and private businesses, marketplaces, institutes and organizations are able to register, update and delete their services.
* The ordinary users are able to access these services by search service and nearby feature.
* The ordinary users are able to see the service portfolio of each public and private businesses, marketplaces, institutes and organizations.
* One-to-one communication is established between the ordinary users and service providers.
* The recommended service feature able the ordinary users to choose the better option.
  1. **Limitation**

Every product has some limitations and Faisalabad Services Portal (FSP) has these following given below.

* + 1. **Domain**

The application is built by remembering the people of only one city. That makes the domain of this application limited to the Faisalabad City, which is one of the limitation the application faced.

* + 1. **Transaction System**

The application doesn’t provide the facility of online transaction because of some reasons. One of the main reason is, the transaction process demands high security, which is presently not possible due to the limitation of budget.

* + 1. **Single-Platform**

The application presently marks the community of only android users and not available for the users of other operating systems.

* 1. **Future Position**

The future position with respect to the application is to terminates the limitations as much as possible and must available the application on hybrid platforms with transaction system and much more.

* 1. **Summary**

This chapter talks about the overview of the application called Faisalabad Services Portal (FSP) and the milestones that application achieved within the duration. Furthermore, the chapter also discussed in detail about the limitations application faced and its future position.

***Report References***

1. *JavaTpoint. (n.d.). Software Design in Software Engineering. JavaTpoint. https://www.javatpoint.com/software-engineering-software-design*
2. *Systems and software engineering – Vocabulary. Iso/Iec/IEEE 24765:2010(E). 2010-12-01. pp. 1–418. doi:10.1109/IEEESTD.2010.5733835. ISBN 978-0-7381-6205-8.*