Chapter No. 02

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.

Hardware Interface Requirement

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

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.

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.

Software Interface Requirement

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

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.

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.

Javascript/React JS

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

Google Maps

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

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.

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.

Firebase

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

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.

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.

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.

Register/Manage Service

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

Verified/Pending Service

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

Search Service

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

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.

Chat Bridge

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

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.

Usability

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

Scalability

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

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.

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.

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.

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.

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 portfolio. | | |
| 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 portfolio 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 portfolio. | | |
| 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 portfolio 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 portfolio. | | |
| Brief Description | The public and private businesses, marketplaces, institutes and organizations watch their verified and pending services portfolios. | | |
| Pre-Condition | User must have an account in an application and submit the service portfolio. | | |
| Post-Condition | The service portfolio 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 portfolio 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 portfolios under their specific categories. | The system will show the services portfolios 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 portfolios by write down the category, apply category filter or nearby feature. | The system will filter the required services portfolios 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 portfolio. | | |
| 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 portfolio 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 portfolios by clicking on each of them. | The system show the pending or verified services portfolios. |
| 03 | The owners verified or rejected the pending service portfolios. | The system will verify or reject the services portfolios 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

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.