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. Smart Access is an android based application & Web App 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

Smart Access 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.

React Native

Hybrid Platform that combines the best parts of native development with React, a best-in-class , Function JavaScript library for building operator interfaces.it also used some other libraries like Code Push.

JavaScript/React JS

For Web Side Panel, Various JavaScript and its powerful library React JS 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 Both Android and Web Sides.

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 Smart Access ,an **android based application & Web Based Panel**.

**App** **Functions :**

* Signup/Login

Smart Acess is very secured and not available for public usage so everyone cannot allow to login into the app handle it from firebase manually that which operator can login and track records.

* Sales Man Records

This Feature Show Real Time Graph Representation into the app which shows Salman recovery remain are other details such as loan empty recovery balance.

* Employee

In this page it include all the records of an employee which includes loan, recovery, remain as well as details is in which includes Credit Debit details and balance details as well managed by dates

* Advance Features

In advanced feature page you can select delete the bank as well as you can see all the records of the bank in which day includes (date of transaction amount as well as Credit Debit record from to to record etc) and insecurities on session you can add a operator who can access website panel are login into that parallel to manage over distribution section and it also has a feature of security to check password using fingerprint authentication.

**Website Functions :**

* Stocks

In the stock management section, we can effectively handle the inventory of all available categories. Currently, we have RGB 500 ml, 1000 ml, 1500 ml, and other variations. We can submit stock orders to the warehouse and utilize the warehouse function to access comprehensive stock data and calculate the current stock levels available.

* Daily Sale

In the daily sales module, we oversee the management of inventory and stocks that are dispatched by drivers and subsequently returned. Our goal is to ensure efficient handling of these transactions.

* Empty

This module facilitates the management of empty stocks that are returned to the company. Whether it involves pallets returned by our own company or third-party entities, we maintain real-time records of empty stocks within the warehouse.

* Sales Man Recovery:

.Salesperson recovery encompasses all records associated with individual salespeople, including loan recovery, closing balances, current balances, and the ability to delete records. This module also supports credit and debit functionalities.

* Sales Man (Record Book):

The Salesperson Record Book contains a comprehensive log of credit and debit transactions, allowing us to track their history by date and effectively monitor their sales performance.

* Other Staff (Record Book):

Similar to the Salesperson Record Book, the Other Staff Record Book contains detailed credit and debit transaction records for all staff members, enabling efficient tracking of their performance and financial activities.

* Employee Recovery:

The Employee Recovery Book consolidates all data related to employee loan recoveries. It provides insights into loan recovery records, categorized by months and dates, facilitating effective management and analysis.

* Daily Expenses and Closing:

Within the Daily Expenses and Closings module, we can record various expenses that occur during distribution, including bills, vehicle expenses, miscellaneous expenses, and expenditures related to salespeople, employees, and investors. This module provides real-time tracking of total cash, bank transfers, and daily closing records.

* Vehicle Expenses

This section records all vehicle-related expenses, allowing us to track monthly expenditures and apply relevant filters to streamline the data.

* Miscellaneous Expenses

Miscellaneous expenses encompass various expenditure categories, such as distribution, construction, stationery, and food expenses.

* company Credit Debit

The Company Credit and Debit module enables the addition of debit transactions, such as incentives or discounts, as well as credit transactions, such as uniform expenses or other relevant items.

* Tax

This section maintains records of monthly tax payments and closing balances, with the ability to filter data based on specific months.

* Billls

In the bill management section, we handle all types of distribution-related bills and maintain their records. This includes bills from service providers like PTCL, WAPDA, and gas companies.

* **Net Profit:**

The net profit module allows us to calculate the overall profit by considering total expenses, discounts, company expenses, total sales, current stock levels, total incentives, and net profit.

* **Rate Manager:**

In the rate manager module, we can efficiently manage the prices of various stocks. For example, we can set rates for RGB, 500 ml, and 1500 ml stocks. These rates will be utilized for future calculations within the web portal.

* **Promo Manager:**

The promo manager section enables us to add promotions or discounts for different stock items, providing the flexibility to modify their current prices separately.

* **Investor:**

Within the investor section, we maintain a list of all investors associated with the company or distribution. This includes their details, credit and debit information, as well as transaction history.

* **Balance Sheet:**

The balance sheet module allows us to generate comprehensive balance sheets that include records of both debit and credit transactions.

* **Daily Activity:**

The daily activity section provides access to all records on a daily basis. It includes the starting time of work, a detailed table of records, and the option to print out relevant information.

* **Profit & Loss:**

The Profit and Loss module offers a detailed overview of our current financial status, including profit or loss figures, closing balances, and credit and debit records.

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

Smart Access provide effectiveness, efficiency and the overall satisfaction of the operator, 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 operators observe the experience of an application overall excellent.

Security

The application obeys both confidentiality, integrity and availability CIA triad.Only Admin Can Handle Main Functionalities or Accessibility of Apps. The application is available 24-hours, whenever the operators 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 Smart Access.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 operator will put the required credentials inside the fields or use authentication through Google process. | | |
| Pre-Condition | Operator must have to download and run application. | | |
| Post-Condition | The account is registered successfully. | | |
| Failed Condition | When required fields are empty or operator not follows the conditions correctly. | | |
| Primary Actor | Operator | | |
| Dependency | None | | |
| Basic Flow | Steps | Action | Response |
| 01 | Operator download and open the application. | The system will show the panel screen. |
| 02 | The operator select the signup panel. | The system will open the signup panel. |
| 03 | The operator 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 operator 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 operator write down the email address and password or use Google authentication to login in an application. | | |
| Pre-Condition | Operator 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 operator not write down the correct credentials. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the panel screen. |
| 02 | The operator select the login panel. | The system will open the login panel. |
| 03 | The operator 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 operator 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 operator 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 operator forget the password. | | |
| Brief Description | The operator write down the email address and authentication process is done automatically through database. | | |
| Pre-Condition | Operator must have an account in an application. | | |
| Post-Condition | The password is forgot successfully. | | |
| Failed Condition | When required email address field is empty or operator not write down the correct credential. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the panel screen. |
| 02 | The operator select the forgot password panel. | The system will open the forgot password panel. |
| 03 | The operator put the email address and press the forgot password button. | The system authenticates from the database. |
| 04 | The operator write down the new password, confirm it and then click on the done button. | The system will save new password and login the operator 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 | Operator 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 | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The operator select the services form option at bottom navigation bar. | The system will open the services form page. |
| 04 | The operator 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 operator 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 | Operator 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 | Operator | | |
| 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 operator open the application. | The system will show the home screen. |
| 02 | The operator on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The operator select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | The operator select the service portfolio wants to update. | The system will open the services form. |
| 05 | The operator 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 operator 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 | Operator must have an account in an application with successfully verified service portfolio. | | |
| Post-Condition | The service credentials are successfully deleted. | | |
| Failed Condition | When operator undo the process within 14 days. | | |
| Primary Actor | Operator | | |
| 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 operator open the application. | The system will show the home screen. |
| 02 | The operator on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The operator select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | The operator long press the service portfolio wants to delete. | The system will open the delete service popup box. |
| 05 | The operator 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 | Operator 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 operator cancel the request. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case or Login Use Case with Register Service Use Case. | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator on the toggle button of services dashboard inside the more button. | The system will change the bottom navigation bar. |
| 03 | The operator select the manage services option at bottom navigation bar. | The system will open the manage services page. |
| 04 | Here operator 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 operator 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 operator views the public and private businesses, marketplaces, institutes and organizations. | | |
| Pre-Condition | Operator must have an account and open the application | | |
| Post-Condition | The services are displayed successfully. | | |
| Failed Condition | No internet connection. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator 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 operator 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 operator will search the public and private businesses, marketplaces, institutes and organizations according to their categories or nearby feature. | | |
| Pre-Condition | Operator 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 | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator open the application. | The system will show the home screen. |
| 02 | The operator 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 operator will swipe through services the application’s algorithm recommended at home page. | | |
| Pre-Condition | Operator must have an account and open the application. | | |
| Post-Condition | The recommend services are shown successfully. | | |
| Failed Condition | No internet connection. | | |
| Primary Actor | Operator | | |
| Dependency | Signup Use Case or Login Use Case | | |
| Basic Flow | Steps | Action | Response |
| 01 | The operator 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 operator 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 operator and service provider. | | |
| Brief Description | The ordinary operator will communicate through chat bridge with required service provider. | | |
| Pre-Condition | Operator must have an account and open the application with required service provider portfolio. | | |
| Post-Condition | The message send successfully. | | |
| Failed Condition | None | | |
| Primary Actor | Operator | | |
| 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 operator open the application. | The system will show the home screen. |
| 02 | The operator select the required service portfolio and click on the chat button. | The system will establish a connection between the ordinary operator and service provider. |
| 03 | The operator 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.