Al-Imam University
Computer Science Department
College of Computer and Information Sciences

CS290: Software Engineering
2nd Semester 1442H / Spring 2021

SRS Document
For The Smart Platform For AWQAFS Project

Prepared By:

Students Names:	IDs:
Shoug Ali Alsuhaibani	440022732
Raghad Abdulkreem Alateeq	440021987
Manar Mofid Yabrak	440027466
Reema Ali Alomran	440019606
Sarah Abdullah Alsarami	440020811

Instructor: Jawza Almutairi

TABLE OF CONTENTS

1 Introduction

1.1 Purpose

The Software Requirement Specification (SRS) document provides an overview of the SRS. It contains all of the information needed to adequately design and implement the "The Smart Platform For AWQAF`s " such as the purpose of the platform, it's scope, requirements, use case diagram use cases description and analysis models.

1.2 Scope

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements, goals and to provide a detailed overview of our software product.

1.3 Definitions, Acronyms, and Abbreviations

- **User:** anyone works under/for the general authority for AWQAF. (includes Employee, Manager, System Administrator)
- **Employee:** is a person works for the authority with minimum privileges.
- **Manager:** is a person responsible for controlling or administering all or parts of the authority.
- **System Administrator:** is a person responsible for managing and maintaining Users and Users' information in the system.
- **Central Database:** it is a Database formed of Data collected from different sources both inside and outside the Authority boundaries.
- **Information platform:** Contains dashboard and Authority key performance indicators.
- **key performance indicator (KPI):** is a type of performance measurement, KPIs evaluate the success of an organization or of a particular activity (such as projects, programs, products, and other initiatives) in which it engages.
- **Dashboard:** is a type of graphical user interface that often provides at-a-glance views of key performance indicators (KPIs) relevant to a particular objective or business process.
- **Cloud services:** refers to a wide range of services delivered on-demand to companies and customers over the internet such as Microsoft Office, antivirus.
- **SA:** stands for System Administrator**.**
- **SPA:** stands for The Smart Platform For AWQAF.

1.4 References

- https://www.lucidchart.com
- https://www.microsoft.com/microsoft 365/excel

1.5 Overview

For the rest of the (SRS) document, part (2) will show General Description of the system, Platform Function, User Characteristics, and Assumptions and Dependencies.

The next part (3) will specify the Specific Requirements of the system, Functional and Non-Functional, Design Constraints, Use Cases Description and Diagram.

2 General Description

This section will give an overview of the whole system. It will describe it's functionality, types of stakeholders that will use the system, the constraints, and assumptions for the system.

2.1 Platform Function

The "SPA" helps decision-makers connect people, information, and ideas to make quick and reliable decisions based on comprehensive insights and important information that is easily accessible.

2.1.1 The main functions of the platform include:

- **Get comprehensive insights**
  Through this function, the pace of growth can be accelerated by automatically discovering the factors behind past results to simulate future scenarios and predict outcomes using artificial intelligence and machine training.
- **Streamline analytics processes**
  Integrate existing data platforms to simplify the infrastructure for analytical solutions, even if the data is in several different sources.
- **Display decisions**
  Through the platform, it is possible to create dynamic and responsive visual stories that enable you to navigate to obtain detailed information and answer business questions quickly.
- **The comprehensive visions come into effect**
  Operational, financial, and strategic planning can be coordinated using interactive tools for budgeting, forecasting, and analysis to achieve better business outcomes.

2.2 User Characteristics

2.2.1 The Data Analyst

This type of user continuously collects, organizes, analyses, and presents data, also looks at statistics and demands arguments for every decision. The role includes documenting all your business data, identifying patterns and creating reports and dashboards that will support the decision-making process.

2.2.2 The Executive

CEO, is at the helm of the business, driving the company's success by improving operational analytics efficiency and constantly looking for ways to reduce costs. Business intelligence gives an organizational overview that allows the CEO to spot trends across the entire structure of the business – insights that can support:

- Business growth
- Innovation
- Operational efficiency

2.2.3 The Business User

Business intelligence users can come from across the organization. We often talk about two types of business users, the casual business intelligence user and the power user. The difference is that a power user has the capability of working with complex data sets, while the casual user will make use of dashboards to analyse predefined sets of data.

Generally, the business user is often a manager, who is looking for ways to help a department operate more efficiently and more effectively. This type of business user is often untrained in using BI tools but is capable of picking up the basics of reporting and can use business intelligence successfully to report on business activities. Often they'll undergo further training and become fully equipped to perform more in-depth analysis.

2.2.4 The IT Team

IT is another key player in the BI process. They are responsible for the infrastructure and available tools in the company and they manage the rights and roles of employees. they will also try to close the gap between IT and business operations to help increasing BI adoption across the board.

It's their job to ensure that business users are getting the most from data analytics, and they're also a key part of data governance and BI security.

2.3 Assumptions and Dependencies

- All programs necessary for the work must be compatible and consistent with the relevant programs on the sites of the General Authority.
- All solutions and systems necessary for the work must be compatible and approved by the companies providing the software with proof of that.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 *The user shall be able to use cloud services.*

3.1.2 *The user shall be able to send his problem to the system.*

3.1.3 *The user shall be able to log in to the system.*

3.1.4 *The user shall be able to search the data system.*

3.1.5 *The user shall be able to enter Waqf's name.*

3.1.6 *The user shall be able to enter Waqf's type.*

3.1.7 *The user shall be able to enter Waqf's budget.*

3.1.8 *The user shall be able to enter financials return percentage.*

3.1.9 *The user shall be able to display data, by any of the following formats: Diagram, Chart, Table or Standard Report format.*

3.1.10 *The user shall be able to export data, by any of the following formats: PDF and xlsx format.*

3.1.11 *The user shall be able to log out from the system.*

3.1.12 *The manager shall be able to give permission to establish new Waqf.*

3.1.13 *The manager and SA shall be able to request "Central database workflow" report, by any of the following formats: Diagram, Chart, Table or Standard Report format*

3.1.14 The manager and SA shall be able to request "Central database security" report, by any of the following format: Diagram, Chart, Table or Standard Report format.

3.1.15 The manager and SA shall be able to request "Data viewing record" report, by any of the following format: Diagram, Chart, Table or Standard Report format

3.1.16 The system shall be able to fetch data and link with its various sources, then store it in the central database.

3.1.17 The System shall be able to Extract into the information from central database.

3.1.18 The system shall be able display information as dashboards.

3.1.19 Through the available in Through the available information, the system shall be producing the information in the form of Key performance indicators on the platform.

3.1.20 The system shall be able to identify the data access boundary of every user by their IDs.

3.1.21 The system shall be able to identify the appropriate features to be presented to every user by their IDs.

3.1.22 The system shall be able to grant easy reachability of data, by allowing the user to search in the system.

3.1.23 The system shall be able to provide all data related to the Authority activities and fields of interest, by integrating AI tools.

3.1.24 The system shall be able to data analysis and modelling by using the BI Publisher tool.

3.1.25 The system shall able to generate periodic reports by using the **e** BI Publisher tool.

3.1.26 The system shall be able to receive user problems.

3.1.27 The system shall be able to determine the source of the problem.

3.1.28 The system shall be able to fix the problem by using AI tool.

3.1.29 The system shall be able to provide space to store data on the cloud.

3.1.30 *The system shall be able to provide the cloud services.*

3.1.31 *The system shall be able to store new waqf information.*

3.1.32 *The system shall be able to pay bills and taxes.*

3.1.33 *The system shall be able to distribute the return money to the beneficiaries at the request of the waqf owner.*

3.1.34 *The system shall be able to predict the yearly (return money).*

3.1.35 *The system shall be able to choose suitable location for building new waqf.*

3.1.36 *The system shall be able to predict how long the Waqf will take.*

3.2 Use Cases Description

3.2.1 Use Case #1

<table>
<tr><th colspan="2">Use Case Description</th></tr>
<tr><td colspan="2">System: The smart platform system.</td></tr>
<tr><td>Use Case Name: Log in</td><td>Use Case Number: 01</td></tr>
<tr><td colspan="2">Actor(s): User</td></tr>
<tr><td colspan="2">Description: This use case allows the user to login into the platform using created account by the system administrator.</td></tr>
<tr><td colspan="2">Relationships
▪ Includes: Check credentials.
▪ Extends: None.</td></tr>
<tr><td colspan="2">Pre-Condition(s):
▪ The user opens the platform.
▪ The user has the account.</td></tr>
<tr><th colspan="2">Basic Flows:</th></tr>
<tr><th>Primary Actor
(User)</th><th>System</th></tr>
<tr><td>1. User opens the platform.

3. User types his ID and password.

4. User clicks submit button.</td><td>2. System asks the user to enter his ID and password.

5. System checks ID and password.

6. System Checks credentials of the ID.

7. System displays main front page.</td></tr>
<tr><td colspan="2">Alternate flows and/or exceptional flows:
1. If in step 3 of the basic flow the user typed invalid ID and/or password:
1. Steps 1-5 are repeated.
2. The system displays Invalid message like “The ID and/or password is/are invalid”.
3. Steps 2-7 are repeated.</td></tr>
<tr><td colspan="2">Post Condition: Display the main platform page.</td></tr>
</table>

3.2.2 Use Case #2

<table>
<tr><th colspan="2">Use Case Description</th></tr>
<tr><td colspan="2">System: The smart platform system.</td></tr>
<tr><td>Use Case Name: Search</td><td>Use Case Number: 02</td></tr>
<tr><td colspan="2">Actor(s): User</td></tr>
<tr><td colspan="2">Description: This use case allows the user to enter in the search bar the name of the required data.</td></tr>
<tr><td colspan="2">Relationships
▪ Includes: Display Data.
▪ Extends: User information.</td></tr>
<tr><td colspan="2">Pre-Condition(s):
▪ User has logged into System successfully.
▪ Main front page of the platform is loaded.</td></tr>
<tr><th colspan="2">Basic Flows:</th></tr>
<tr><th>Primary Actor
(User)</th><th>System</th></tr>
<tr><td>1. User Clicks on the search bar.

3.User types new data to search.

4. User clicks search button.

7. User chooses the required data.</td><td>2. System Displays the previous search operations.

5. System Display the related Data.

8. Display Data use case is performed.</td></tr>
</table>

Alternate flows and exceptional flows: **1. If in step 3 of the basic flow the user selects one of the previous searched data:** 1. steps 1-2are repeated. 2. User selects one of the previously searched data. 3. steps 4-7 are repeated. **2. Unauthorized search**: If in step 3 of the basic flow the user typed data he/she has no access to, then: 1. step 1-4 are repeated. 2. System displays a message “You are not Authorized to Access this Data”. 3. System returns to main front page. 4. **Not Found search**: If in step 3 of the basic flow the user typed data that does not exists, then: 3. step 1-4 is repeated. 4. System displays a message “The Required Data Not Found”. 3.System returns to main front page.
Post Condition: None.

3.2.3 Use Case #3

Use Case Description	
System: The smart platform system.	
Use Case Name: Display data	**Use Case Number: 03**
Actor(s): user	
Description: This use case allows the user to display the analysed data.	
Relationships ▪ **Includes:** None. ▪ **Extends:** Diagram format, Chart format, Report format, Table format, Export Data, Users information.	
Pre-Condition(s): ▪ User has logged into System successfully.	
Basic Flows:	
Primary Actor (User)	**System**

1. User clicks on the data to be displayed. 3. User selects Diagram format.	2. System displays four display format options “Diagram format”, “Chart format”, “Report format”, “Table format". 4. Display data as Diagram format option is created. 5. System displays the data in the selected format.
Alternate flows and/or exceptional flows: **1. If in step 3 of the basic flow the user selects Chart format:** 1. Steps 1-2 are repeated. 2. The user selects Chart format. 3. Display data as Diagram format option is created. **2. If in step 3 of the basic flow the user selects Report format:** 1. Steps 1-2 are repeated. 2. The user selects Report format. 3. Display data as Diagram Report format option is created. **3. If in step 3 of the basic flow the user selects Table format:** 1. Steps 1-2 are repeated. 2. The user selects Table format. 3. Display data as Table format option is created. **4. if the user wants to export data in the basic flow or Alternate flows 1,2,3:** 1. Steps 1-5 are repeated. 2. User clicks on export button. 3. Use case Export data is performed.	
Post Condition: Display data page.	

3.2.4 Use Case #4

Use Case Description	
System: The smart platform system.	
Use Case Name: Export Data	**Use Case Number: 04**
Actor(s): User	
Description: This use case allows the user to Export the displayed.	
Relationships ▪ **Includes:** None. ▪ **Extends:** PDF format, xlsx format.	

Pre-Condition(s): ▪ User has logged into System successfully. ▪ Data to be export is displayed.	
Basic Flows:	
Primary Actor (User)	**System**
1. User clicks on “Export Data” option. 3. User selects “PDF format” option. 6.User clicks Export button.	2. System displays two export options ”PDF format”, ”xlsx format”. 4. Export data as PDF format option is created. 5. System displays the final format. 7. System Exports the data with the final format. 8.returns to the displayed data.
Alternate flows: **1. If in step 3 of the basic flow the user selects xlsx format option:** 1. steps 1-2 are repeated. 2. User selects “xlsx format” option. 3. Export data as xlsx format option is created. 4. steps 5-8 are repeated. **2. If in step 6 of the basic flow or Alternate flow the user clicks cancel button:** 1. steps 1-5 are repeated. 2. User clicks cancel button. 3. step 8 is repeated.	
Post Condition: a copy of the selected format is saved in the computer in the case of basic flow and alternate flow 1.	

3.2.5 Use Case #5

Use Case Description	
System: The smart platform system.	
Use Case Name: Cloud Services	**Use Case Number: 05**
Actor(s): User	
Description: This use case allows the user to benefit from the cloud services.	

Relationships

- **Includes:** None.
- **Extends:** None.

Pre-Condition(s):

- User has logged into System successfully.

Basic Flows:

Primary Actor (User)	System
1. User clicks on “Cloud Services” section.	2. System asks the User to enter his/her work mail address and password for accessing the account with two options ”Next”, ”Cancel”.
3. User enters the work mail address and password.	
4. User clicks “Next” option.	5. System checks user mail address and password.
	6. System displays User’s account on the Cloud.

Alternate flows and/or exceptional flows:

1. If in step 3 of the basic flow the User fills invalid work mail address and/or password:

1. steps 1-4 are repeated.
2. System displays message “Invalid work mail address and/or password”.
3. steps 2-6 are repeated.

3. If in step 4 of the basic flow the User clicks “Cancel” option:

1. steps 1-3 are repeated.
2. User clicks “Cancel” option.
3. System returns to main front page.

Post Condition: User has accessed his/her Cloud Services account successfully.

3.2.6 Use Case #6

<table>
<tr><th colspan="2">Use Case Description</th></tr>
<tr><td colspan="2">System: The smart platform system.</td></tr>
<tr><td>Use Case Name: Report problem.</td><td>Use Case Number: 06</td></tr>
<tr><td colspan="2">Actor(s): User.</td></tr>
<tr><td colspan="2">Description: This use case allows the user to send his problem.</td></tr>
<tr><td colspan="2">Relationships
▪ Includes: None.
▪ Extends: None.</td></tr>
<tr><td colspan="2">Pre-Condition(s):
▪ User has logged into System successfully.</td></tr>
<tr><th colspan="2">Basic Flows:</th></tr>
<tr><th>Primary Actor
(User)</th><th>System</th></tr>
<tr><td>1 .User clicks on the problem reporting section.

3. User clicks “Data problems” option.

5. User fills in the problem details.

6. User clicks “Submit” option.</td><td>2. System displays two possible options for a problem, "Data problems" and "User problems".

4.System displays a page to fill in the problem details with two options "Cancel", "Submit".

7.System receives the problem. 8.System

returns to the home page.</td></tr>
<tr><td colspan="2">Alternate flows and exceptional flows:
1. If in step 3 of the basic flow the user clicks on User problems:
1. Steps 1-2 repeated.
2. User clicks “User problems” option
3. Steps 4-8 repeated.
2. If in step 6 of the basic flow the user clicks on "Cancel" option:
1. Steps 1-5repeated.
2. User clicks “Cancel” option.
3. Steps 8 is repeated.</td></tr>
<tr><td colspan="2">Post Condition: User problem has been sending to the system.</td></tr>
</table>

3.2.7 Use Case #7

Use Case Description	
System: The smart platform system.	
Use Case Name: Request Monitoring Reports	**Use Case Number: 07**
Actor(s): System Administrator - Manager	
Description: This use case shows the reports requested by users who have access to them.	
Relationships ▪ **Includes:** Display Data. ▪ **Extends:** Central Database security report, Data viewing record, Central Database workflow report.	
Pre-Condition(s): ▪ System administrator/manager has logged into System successfully.	
Basic Flows:	
Primary Actor (System Administrator/ Manager)	**System**
1. SA/Manager selects "Request Monitoring Reports" section. 3. SA/Manager selects " Central Database security report" option.	2. System displays 3 request report options "Central Database security report”,” Data viewing record”," Central Database workflow". 4. Central Database security report option is created. 5. Display Data use case is performed.
Alternate flows: **1. If in step 3 of the basic flow the SA/Manager selects Data viewing record:** 1. steps 1-2 are repeated. 2. SA/Manager selects “Data viewing record” option. 3 Data viewing record option is created. 4. step 5 is repeated. **2. If in step 3 of the basic flow the SA/Manager selects Central Database workflow:** 1. steps 1-2 are repeated. 2. SA/Manager selects “Central Database workflow” option. 3 Central Database workflow option is created. 4. step 5 is repeated.	
Post Condition: None.	

3.2.8 Use Case #8

<table>
<tr><th colspan="2">Use Case Description</th></tr>
<tr><td colspan="2">System: The smart platform system.</td></tr>
<tr><td>Use Case Name: Register User</td><td>Use Case Number: 08</td></tr>
<tr><td colspan="2">Actor(s): System Administrator</td></tr>
<tr><td colspan="2">Description: This use case create a new account for a user to access the system.</td></tr>
<tr><td colspan="2">Relationships
▪ Includes: Define Access Level.
▪ Extends: None.</td></tr>
<tr><td colspan="2">Pre-Condition(s):
▪ System administrator has logged into System successfully.
▪ The user must work under the Authority.</td></tr>
<tr><th colspan="2">Basic Flows:</th></tr>
<tr><th>Primary Actor (System Administrator)</th><th>System</th></tr>
<tr><td>1. SA clicks on "Add New User" Section.

3. SA fills the form with the following information:
- Name.
- Employment title.
- Social security numbers.
- Mail Address.
- User ID.
- Password.

4. SA submits the form.

8. SA clicks “Confirm”.</td><td>2. System displays User Registration form.

5. System checks the user information.

6. Define Access Level use case is performed.

7. System displays a confirmation message.

9. System Adds the User.

10. System returns to the main page.</td></tr>
</table>

Alternate flows and exceptional flows: **1. If in step 3 of the basic flow the administrator fills invalid user information:** 1. steps 1-5 are repeated. 2.System displays a message “Invalid User information”. 3. steps 2-10 are repeated. **2. If in step 3 of the basic flow the administrator fills existent user information:** 1. steps 1-5 are repeated. 2.System displays a message “The User is in the System”. 3. step 10 is repeated.
Post Condition: User has been successfully added to the system.

3.2.9 Use Case #9

Use Case Description	
System: The smart platform system.	
Use Case Name: Define Access Level	**Use Case Number: 09**
Actor(s): System Administrator	
Description This use case allows Determine how much data each user can access.	
Relationships ▪ **Includes:** None. ▪ **Extends:** None.	
Pre-Condition(s): ▪ System administrator has logged into System successfully. ▪ The user must already have an account inside the system.	
Basic Flows:	
Primary Actor **(System Administrator)**	**System**
1. SA selects "Define access level for user"section. 3. SA enter User ID. 6. SA selects the permissions for the user. 7.SA clicks on "Submit" option.	2.System asks the SA to enter User ID. 4.System check the User ID. 5.System displays all info about User and list of User Permissions. 8-System return to the main page.

Alternate flows and exceptional flows: **1. If in step 3 of the basic flow System Administrator enters invalid ID:** 1. steps 1-3 are repeated. 2. System displays a message “Invalid User ID”. steps 2-8 are repeated.
Post Condition: The user's permissions on the system have been defined.

3.2.10 Use Case #10

Use Case Description	
System: The smart platform system.	
Use Case Name: User Information	**Use Case Number: 10**
Actor(s): System Administrator	
Description: This use case allows System Administrator to display all Users’ information or search for a specific user.	
Relationships ▪ **Includes:** Search, Display Data. ▪ **Extends:** None.	
Pre-Condition(s): ▪ System is ready. ▪ System Administrator has logged into her/his account successfully.	
Basic Flows:	
Primary Actor (System Administrator)	**System**
1. SA selects “User Information” section. 3. SA selected a “Display Specific User information” option.	2. System displays two display options “Display all Users’ information”, “Display Specific User information”. 4. Display Specific User information option is created. 5. Search Use Case is performed.
Alternate flows: **1. If in step 3 SA selected Display all Users’ information option:** 1. Steps 1-2 are repeated. 2. Display all Users’ information option is created. 3. Display Data Use case is performed.	

Post Condition: None.

3.2.11 Use Case #11

Use Case Description	
System: The smart platform system.	
Use Case Name: Remove User	**Use Case Number: 11**
Actor(s): System Administrator	
Description: This use case allows System Administrator to delete a user from the system.	
Relationships ▪ **Includes:** None. ▪ **Extends:** None.	
Pre-Condition(s): ▪ System administrator has logged into System successfully. ▪ The user must already have an account inside the system.	
Basic Flows:	
Primary Actor **(System Administrator)**	**System**
1. SA clicks on "Remove User" section. 3. SA enters the User ID. 6. SA clicks “Confirm” option.	2. System asks the SA to enter User ID. 4. System checks the validity of User ID. 5. System displays a confirmation message. 7. System deletes the User. 8. System returns to main front page.

Alternate flows and exceptional flows: **1. If in step 3 of the basic flow the Administrator enters invalid ID:** 1. steps 1-4 are repeated. 2. System displays a message “Invalid User ID”. 3. steps 2-8 are repeated. **2. If in step 6 of the basic flow the Administrator clicks “Cancel”:** 1. steps 1-5 are repeated. 2. SA clicks “Cancel” option. 3. step 8 is repeated.
Post Condition: User has been successfully deleted from the system.

3.2.12 Use Case #12

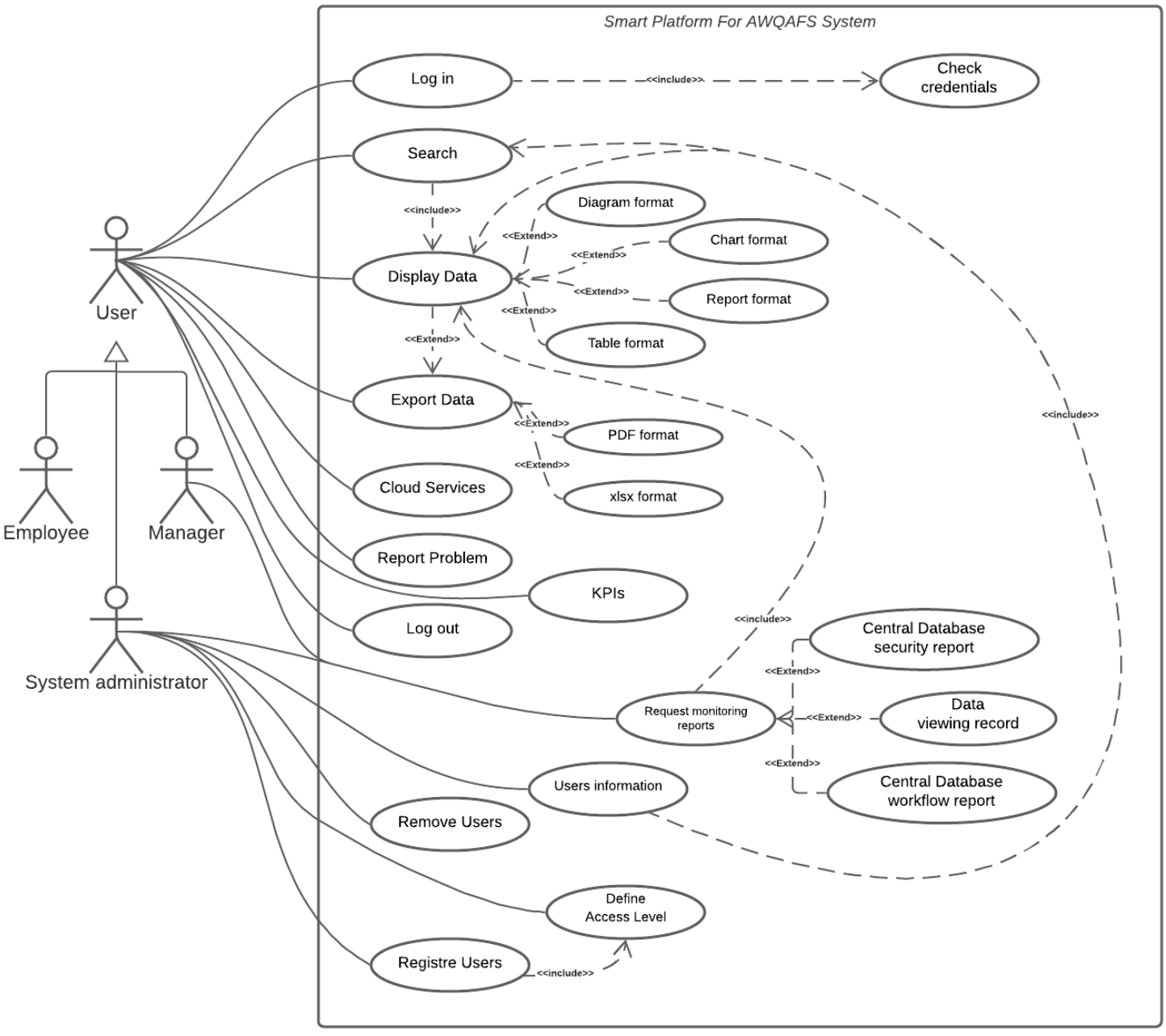
Use Case Description	
System: The smart platform system.	
Use Case Name: Log out	**Use Case Number: 12**
Actor(s): user	
Description: This use case allows the user to logout from the platform.	
Relationships ▪ **Includes:** None. ▪ **Extends:** None.	
Pre-Condition(s): ▪ The user already logged onto the platform.	
Basic Flows:	
Primary Actor (User)	**System**
1. The user clicks the logout button.	2. The system displays the message to the user to confirm logout. 3. The system logs out from the platform.

Alternate flows and/or exceptional flows:

1. **If in step 2 of the basic flow the Logout message not confirmed:**
    1. The platform stays on the current page.
2. **if the user wasn't active for 10 minutes continuously:**
    1. The system logs out from the platform automatically.
    2. The system displays the message "you weren't active for 10 minutes".

Post Condition: The platform displays login page.

3.3 Use Case Diagram



3.4 Non-Functional Requirements

Non-Functional Requirement is requirements that are specific criteria of the developed system. this can be used to judge the operation of the system, rather than specific behaviours. this is contrasted with a functional requirement that define the specific behaviours or function of the system.

3.4.1 Performance

Requirement ID	Requirement description
P1	The system shall be responded to user interactions and implement them quickly in less than 3 seconds.

3.4.2 Availability

Requirement ID	Requirement description
A1	The platform shall be available 99% of the time, downtimes due to maintenance shall not exceed one hour at a time, once a week.

3.4.3 Security

Requirement ID	Requirement description
S1	Permissions for accessing system data shall only be changed by a system administrator.
S2	The system shall correspond with authority protection systems.
S3	Since the platform deals with multiple systems, the transactions between them shall be safe.

3.4.4 Efficiency

Requirement ID	Requirement description
E1	The platform shall be able to serve up to 300 users at the same time.
E2	The system shall be able directly integrated with the data sources to facilitate better updating and extraction of data for the user.
E3	The employee that will be using the system shall be trained on it before.

3.4.5 Usability

Requirement ID	Requirement description
U1	The platform shall be user-friendly and easy to use.

3.4.6 Reliability

Requirement ID	Requirement description
R1	To avoid crashing the system, the system shall be catch and handle the exceptions.

3.5 Design Constraints

3.5.1 The system shall be able to access information stored in the several existing systems the authority have then store it in Central Database.
3.5.2 The system shall have a Windows-based desktop interface.
3.5.3 The system shall be run on individual employee desktops.

4 Team Members Contributions

The team held ZOOM meetings to write this SRS document.

5 Conclusion

In this SRS document we wrote the introduction, general description and specific requirements which will make it simpler for the developers to develop the software.

Design
For The Smart Platform For AWQAFS Project (SPA)

TABLE OF CONTENTS

Abstract

This document shows the architectural design of SPA. It contains graphical description about classes and functions in a simple way, obviously that structure of the system is now more understandable and the representation of requirements is shown in a way that reflects the purpose of the system also it gives a brief of how the software will be like.

1 Introduction

The following document is to present a detailed description of the SPA It will explain the purpose of the platform, its features, and interfaces of the platform, the constraints which should work in the system, how it interacts with the Internal and external systems, and various data sources.

The "SPA" is a platform that contains information and indicators performance to provide the possibility of discovering different knowledge and time-related information, and to extract and present this information in a way that suits the nature of the commission's work and the requirements of its users and Helping decision makers.

This document has been divided into four main sections which are design ,detailed design ,user interfaces design, conclusion the first section contains are System Context Diagram that defines the boundary between the system or subsystem, System Architecture Pattern that expresses a fundamental structural organization schema for software, Class Diagram That describes the structure of a system by showing the system's: classes, their attributes, operations (or methods), and the relationships, Class Method Description describes the contents of the objects that belong to system, the second section contains a Flowchart diagram that shows the process of selected methods, Sequence Diagram is an interaction diagram that details how operations are carried out in the system the third section is User Interface that is snapshots of user interaction with the interface, with detailed explanation of each figure, finally is the Conclusion, which is a summary of the whole document.

2 Design

2.1 System Context Diagram

The system architecture is composed of several systems the authority has and a set of subsystems working together to from a complete system as shown in Fig.1, Each subsystem contains classes that are logically similar in functionality.

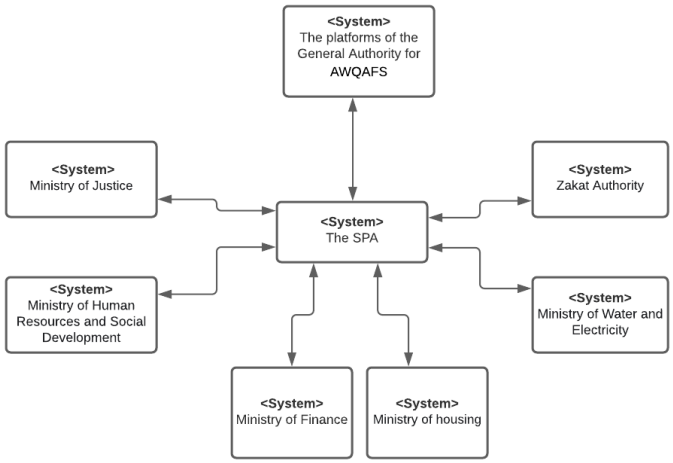


Fig. 1 SPA System Architecture

2.2 System Architecture Pattern

With the increasing demand of market, we need to store the information so that its retrieval will be simpler for processing management's decisions. The value of information is drastically changing in past years and ultimately all organizations have moved to Quality work. Quality and appropriateness of the information becomes a key challenge for most of the organizations in past decades. The problem associated with business intelligence architecture is the storage of heterogeneous data sources which might leads to inconsistency. The business intelligence information should also be presentable in different format for analyst's decision making. Model-View-Controller (MVC) architecture pattern resolve this issues. See Fig.2

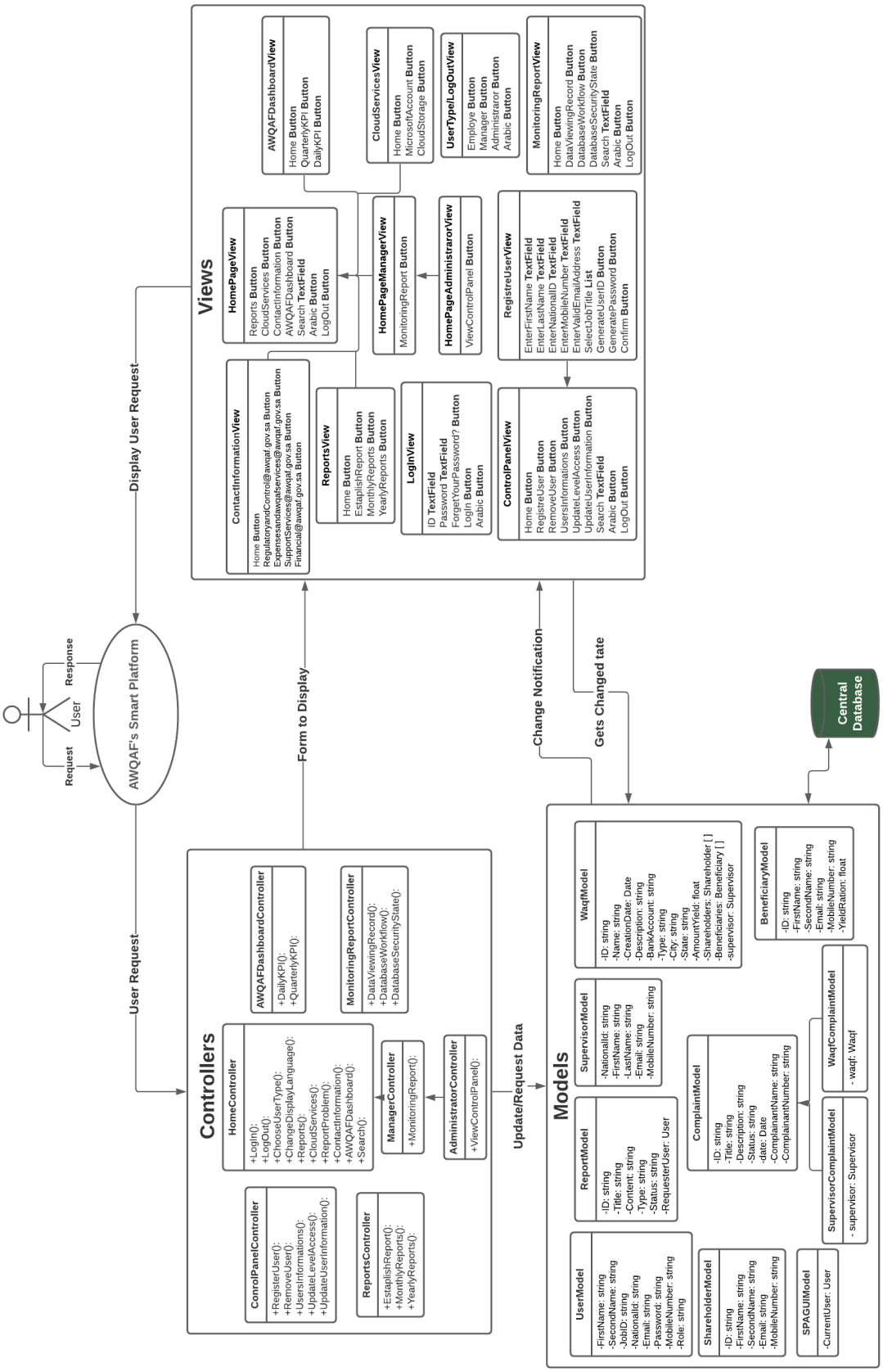
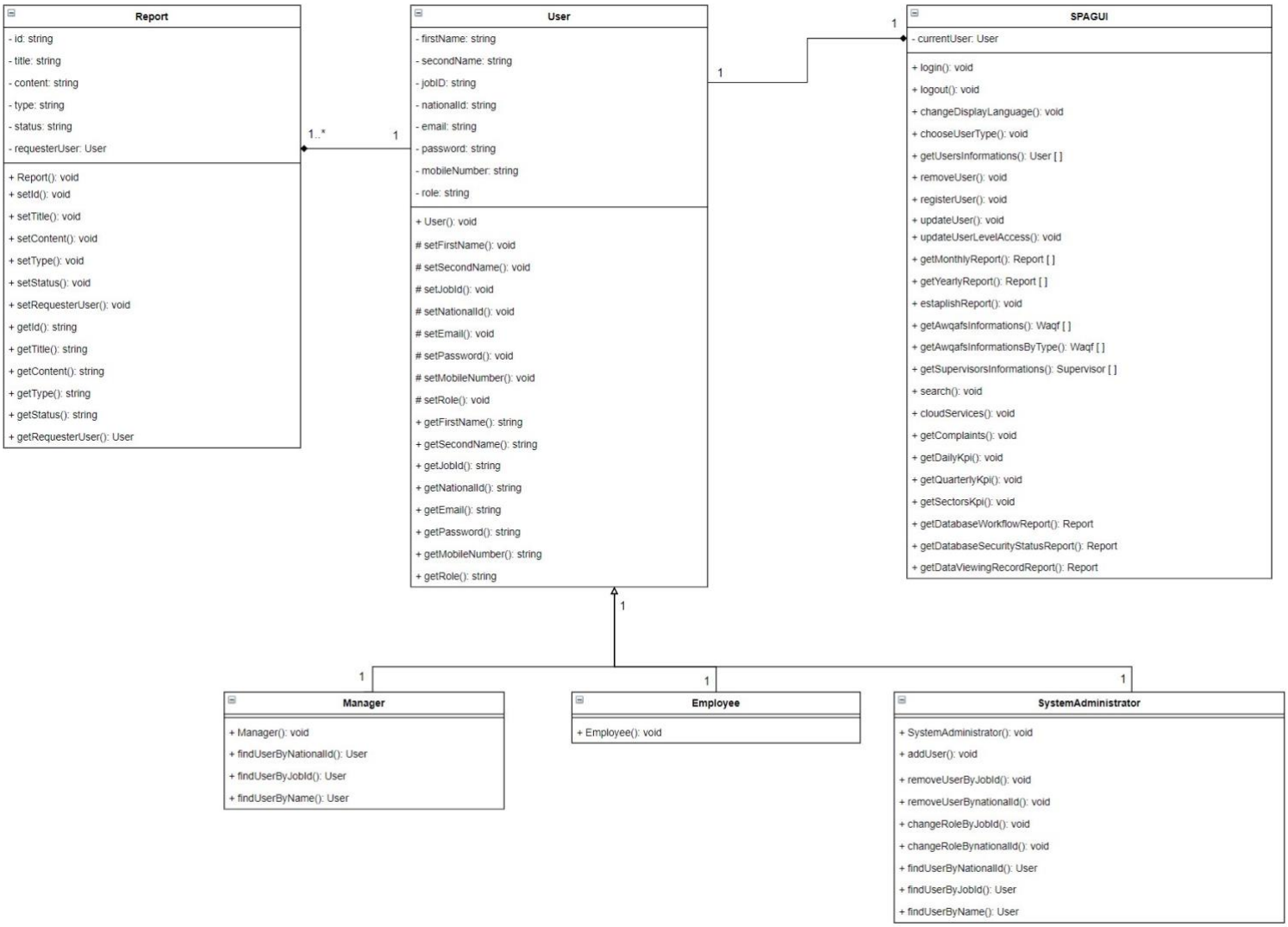
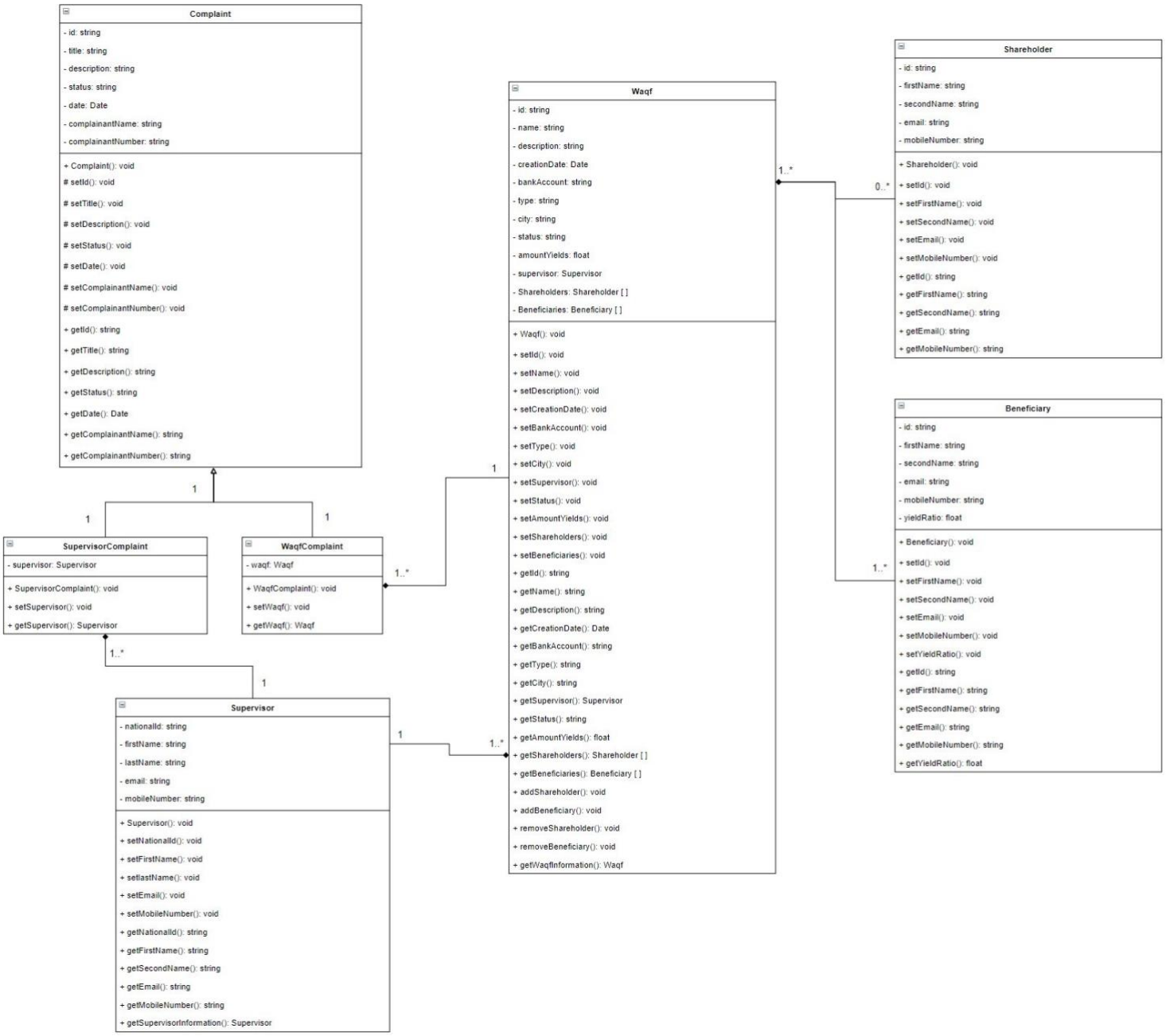


Fig. 2 SPA System Architecture Pattern

2.3 Class Diagram





There is a more clear copy of the Class Diagram, please find it on the below URL.
(please download the file then open it).
https://drive.google.com/file/d/view?usp=sharing

2.4 Class Method Description

- **Manager Class**

The Manager Class contain how can the manager find user by his Nationality Id and his Job Id.

Class	Manager
Method	findUserByNationalId
Visibility	Public
Return type	User
Parameters, types	nationalId: String
Description	This method allows to find the user by his National Id

Class	Manager
Method	findUserByJobId
Visibility	Public
Return type	User
Parameters, types	jobId: String
Description	This method allows to find the user by his job Id

Class	Manager
Method	findUserByName
Visibility	Public
Return type	User
Parameters, types	name: String
Description	This method allows to find the user by his name

▪ System Administrator Class

This class contain what can the SA do.

Class	System Administrator
Method	addUser
Visibility	Public
Return type	Void
Parameters, types	firstName: String, secondName: String, jobID: String, nationalId: String, Email: String, password: String, role: String
Description	This method allows to add User into system

Class	System Administrator
Method	removeUserByNationalId
Visibility	Public
Return type	Void
Parameters, types	nationalId: String
Description	This method allows to remove user by his Id' National

Class	System Administrator
Method	changeRoleByJobId
Visibility	Public
Return type	Void
Parameters, types	jobId: String, role: String
Description	This method to change the validity of user by his Id' job

Class	System Administrator
Method	changeRoleByNationalId
Visibility	Public
Return type	Void
Parameters, types	nationalId: String, role: String
Description	This method to change the validity of user by his Id' job

Class	System Administrator
Method	findUserByNationalId
Visibility	Public
Return type	User
Parameters, types	nationalId: String
Description	This method allows to find the user by his National Id

Class	System Administrator
Method	findUserByName
Visibility	Public
Return type	User
Parameters, types	Name: String
Description	This method allows to find the user by his name

Class	System Administrator
Method	findUserByJobId
Visibility	Public
Return type	User
Parameters, types	jobId: String
Description	This method allows to find the user by his job Id

Class	System Administrator
Method	removeUserByJobId
Visibility	Public
Return type	Void
Parameters, types	jobId: String
Description	This method allows to remove user by his Id' job

- **SPAGUI Class**

This class contain the GUI's functions of SPA.

Class	SPAGUI
Method	chooseUserType
Visibility	Public
Return type	Void
Parameters, types	userType: String
Description	This method allows to choose user type such as: Employee – Manager – Administrator

Class	SPAGUI
Method	changeDisplayLanguage
Visibility	Public
Return type	Void
Parameters, types	none
Description	This method allows to change language in GUI between English-Arabic.

Class	SPAGUI
Method	updateUser
Visibility	Public
Return type	Void
Parameters, types	firstName: String, secondName: String, jobID: String, nationalId: String, Email: String, password: String, role: String
Description	This method allows to update some information of user for example if the user change his email this method make update.

Class	SPAGUI
Method	getSupervisorsInformation
Visibility	Public
Return type	Supervisor[]
Parameters, types	none
Description	This method will return the information of the super visor who is responsible for monitoring the Waqf

Class	SPAGUI
Method	cloudServices
Visibility	Public
Return type	Void
Parameters, types	none
Description	This method allows to display cloud service functions for the authority sector employees.

Class	SPAGUI
Method	getDailyKpi
Visibility	Public
Return type	Void
Parameters, types	none
Description	This method allows to display the daily KPI of the authority

Class	SPAGUI
Method	getYearlyReport
Visibility	Public
Return type	Report[]
Parameters, types	none
Description	This method contains array the that represents the number of complaints

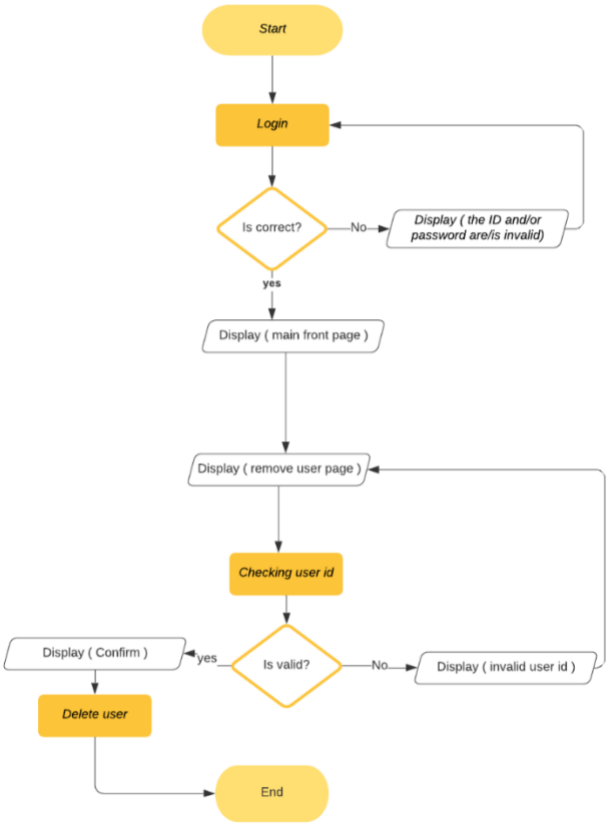
Class	SPAGUI
Method	getComplaints
Visibility	Public
Return type	Complaint[]
Parameters, types	none
Description	Represents the annual reports of the authority

3 Detailed Design

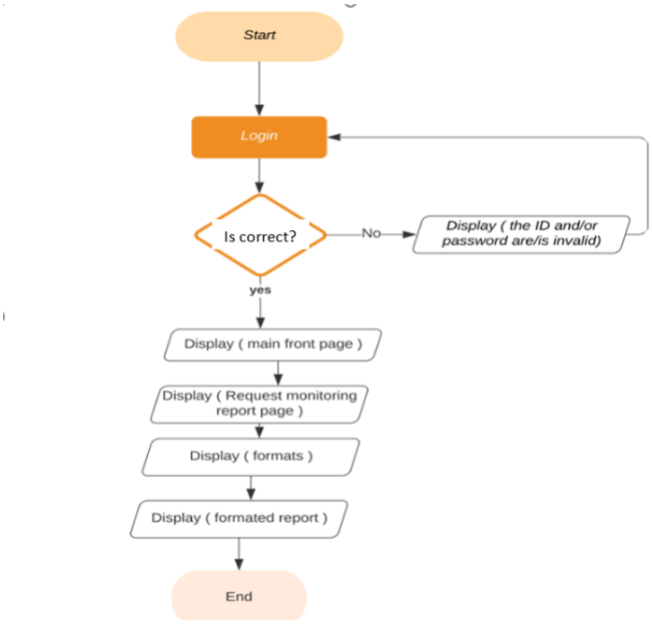
This section provides detailed diagrams for some functions.
The first three diagrams are flowcharts, and the rest are sequence diagrams.

3.1 flowchart diagram

1) *Remove User: Delete a user from the system.* 2) *Register user: Create new account for a user.*

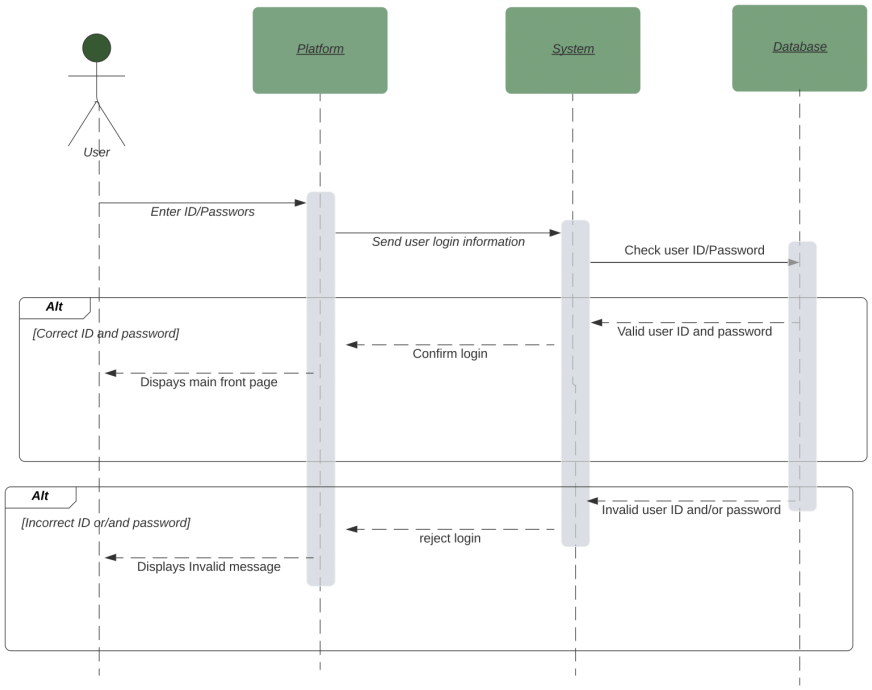


3*) Request monitoring report: Shows the reports by users who have access to them.*

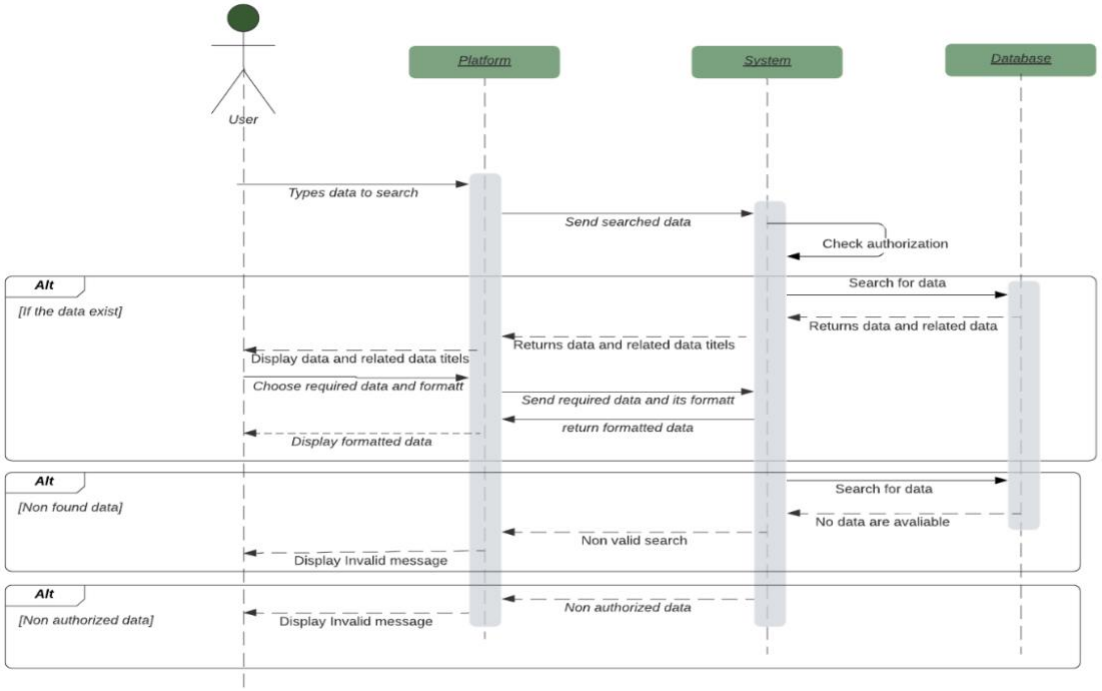


3.2 Sequence Diagram

1) Log in: Allows The user to log in into the system.



2) Search: Allows the user to search data in the database.



4 User Interface Design

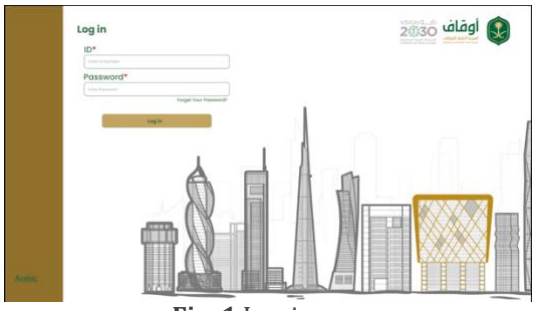


Fig. 1 Log in page

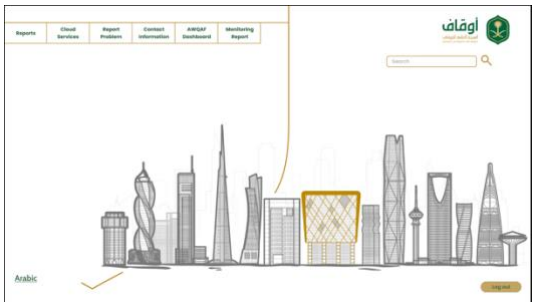


Fig. 2 Employee Home page

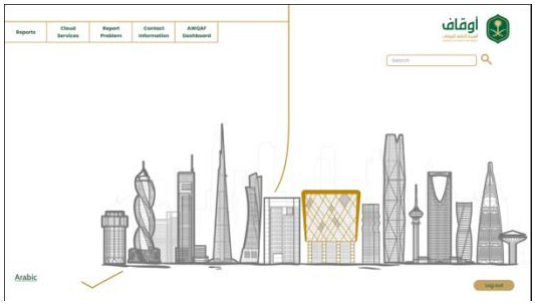


Fig. 3 Manager Home page

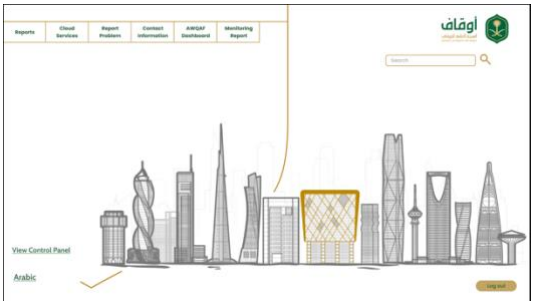


Fig. 4 Administrator Home page

You can see the rest of the Prototype in here
https://www.figma.com/proto/CS290-Project-Prototype

5 Conclusion

In the second phase, the design phase, the team has done to cover the design requirements appropriate to the project, starting from the context diagram to the user interface design to make it easier, understandable and it is of high efficiency and then the prototype for an initial evaluation the process was not easy but the team did their best to make it more understandable and clear.

6 References

- https://www.lucidchart.com
-https://www.figma.com