

Table of Contents

1. INTRODUCTION.....	2
1.1 PURPOSE.....	2
1.2 SCOPE.....	2
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	2
1.4 REFERENCES.....	3
1.5 OVERVIEW.....	3
2. GENERAL DESCRIPTION	3
2.1 PRODUCT PERSPECTIVE.....	3
2.2 PRODUCT FUNCTIONS.....	3
2.3 USER CHARACTERISTICS	4
2.4 GENERAL CONSTRAINTS	4
3. SPECIFIC REQUIREMENTS	4
3.1 EXTERNAL INTERFACE REQUIREMENTS	4
3.1.1 <i>User Interfaces.</i>	4
3.1.2 <i>Hardware Interfaces.</i>	5
3.1.3 <i>Software Interfaces.</i>	5
3.2 FUNCTIONAL REQUIREMENTS	6
3.2.1 <i><Functional Requirement or Feature #1> The phone App: Sign-up.</i>	6
3.2.2 <i><Functional Requirement or Feature #2> The phone App: Sign-in</i>	7
3.2.3 <i><Functional Requirement or Feature #3> The phone App: Find appointments</i>	8
3.2.4 <i><Functional Requirement or Feature #4> The phone App: Book an appointment</i>	9
3.2.5 <i><Functional Requirement or Feature #5> The phone App: Customize account and profile information.</i>	10
3.2.6 <i><Functional Requirement or Feature #6> The facility program: Add appointments.</i>	11
3.2.7 <i><Functional Requirement or Feature #7> The facility program: Edit appointments.</i>	12
3.2.8 <i><Functional Requirement or Feature #8> The facility program: Add a new section.</i>	13
3.2.9 <i><Functional Requirement or Feature #9> The facility program: Edit section name.</i>	14
3.2.10 <i><Functional Requirement or Feature #10> The facility program: remove section appointments.</i>	15
3.2.11 <i><Functional Requirement or Feature #11> The facility program: Transfer appointments.</i>	16
3.2.12 <i><Functional Requirement or Feature #12> The facility program: Customize account and profile information.</i>	17
3.3 USE CASES DIAGRAM.....	18
3.4 NON-FUNCTIONAL REQUIREMENTS	19
4. TEAM MEMBERS CONTRIBUTIONS	20
5. CONCLUSION.....	20

1. Introduction

This section will be an introduction to the purpose of this SRS document and the scope, definitions, references and overview.

1.1 Purpose

The purpose of this Software Requirements Specification is to define the system clearly and give detailed information about Appointments System (AS), the intended audience are the owners of facilities who will use this system and other development teams such as design team, analysis team, requirements team, and other members of development team.

1.2 Scope

The owners of many different facilities face a lot of problems with appointments scheduling, also the majority of our community had these problems as well, as a result of these problems Appointments system (AS) solved most of them by allowing the user to book appointments online and the facilities to schedule it.

First part is the facility interface, The owner of the facility should provide the requirements and information which is needed to be registered in the Appointments system which will be a java based program which will provide a control panel that allow the facility owner to add different sections and its appointments to ease it to the user to find what they're looking for and to manage the appointments such as, changing dates, changing times, and edit their sections or the branch name of the facility, second part is the user interface (The customer), the user who is going to book an appointment using Appointments system (AS) need to have the phone app and should provide the information to be registered, after registering the user can search about facility which the appointment will be in and after finding it the user can browse in sections of this facility, after choosing the section the user can select the date and all of the available appointments will be shown in different times so the user can book the time needed.

1.3 Definitions, Acronyms, and Abbreviations

<i>Term</i>	<i>Definition</i>
<i>AS</i>	<i>Appointments System</i>
<i>App</i>	<i>Application</i>
<i>SRS</i>	<i>Software Requirements specification</i>
<i>GUI</i>	<i>Graphical User-Interface</i>
<i>SMS</i>	<i>Short message service</i>
<i>SDP</i>	<i>Software Development plan</i>

1.4 References

IEEE Guide to SRS.

1.5 Overview

In the second section of this SRS will be a general description about the system functionality which will be more specific and going to be about how the system interact with other systems and the constraints and general characteristics of the users.

The third section will be about the specific requirements and the functionality of the system, it will provide a list of all of the functional and non-functional requirements and different ways how the system and the interfaces will work.

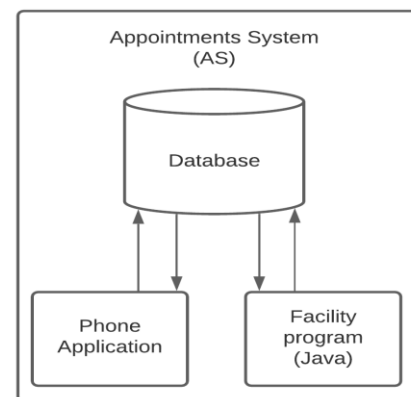
The fourth section will Identify and describe each team member contributions.

The fifth section will be a conclusion it will be about what has been achieved and what is left, and some future work.

2. General Description

2.1 Product Perspective

The AS spilt into two parts; the first part is a control panel java-based program which is connected to the main database server. The second part is a phone app for the user which is connected to the database server as well. The system provides to the user the ability to reach many different facilities and book appointments without direct contact which secures and save a lot of time to the user, and it secures the personal information and data of the user which brings tons of reliabilities.



2.2 Product Functions

The AS allows the user to register in the system using the phone App to browse for appointments within different facilities within different sections, the user can book appointments and select exactly the time and date which the user want, also the user can cancel, and change appointments time or date, also the user can change the account personal information and the login info and can customize the profile.

The facility will register in the system using the java-based program, which allows the facility to write a description to present itself, Also the facility can add, edit, and remove sections and can also add and remove and edit appointments and adjust the dates and time on them and monitor the user appointments and the user communications info.

After the user books an appointment AS will send a SMS message to check the user phone number and verifies it, then the AS will send a SMS message 1 day before the appointment to notify the user and also sends an email, also the AS provide the ability for the user to contact us using automated chat if there are any technical difficulties.

2.3 User Characteristics

The main two people are the customer and the facility user, the customer can only search for appointments and book them, the customer should have only the knowledge about how to use a phone App and knows how to browse and search.

The facility user can only add or remove or edit sections, and also add appointments and adjust their dates and time and also the facility user can monitor the customer appointments and user info, the facility user should know how to handle a control panel and large amount of data, also the facility user needs to know how to process the data and analyze it.

2.4 General Constraints

The user must use an IOS or Android operating system to run the phone App also it needs to be up-to-date, the user register info will be only by phone number and a password, also the user information and credentials will safe and secured.

The facility user needs a compatible computer to run the AS system program, the computer operating system needs to be either Windows OS or Mac OS and they need to be up-to-date so the program can work perfectly, the facility user needs to have a secured computer which is clear with viruses and bugs to maintain the proper security to the facility information and the customer information.

3. Specific Requirements

This section will be the detailed and the functional and non-functional requirements and the specific requirements for building this product.

3.1 External Interface Requirements

3.1.1 User Interfaces.

The customer who is going to use the phone App for the first time will see the front page of the App which is Sign-up/Sign-in. The user needs to sign-up if she/he don't have a registered account, if the user is already having a registered account they need to sign-in to continue, after the user sign-in the main page will pop up, the main page contains five elements, the first one is a textbox which allows the user to enter the name of the facility, the second element is a drop-down list which shows the sections of the facility the user searched for, the third element is a

textbox to enter the date of the appointment wanted which is year/month/day, the fourth element is a button which activate the searching process for the appointments, the fifth element is the settings button which shows a page that has two sections the first one is to customize the user profile the second one is to change the account information, after that when the user starts to search for appointments if there is no available appointments a pop-box will appear saying there is no appointments available on this date, if there is an available appointments it will open a new page that contains a list of the available appointments, when the user chooses the wanted appointment it will show a confirm button with the information of this appointment.

The first-time facility user who doesn't have an account registered need to sign-up in the system, if the user already have an account they need to sign-in, after the user signed-in in the system the first page will pop which is the main page, the main page contains three elements the first one is a settings button which opens up a new page that contain three elements the first one is a button to customize the facility profile, the second one is a button to change the account information, the third one is a button to contact the support, the second element in the main page is a button which brings up the appointments settings page that contains two elements the first one is a button to add new appointments the second one is to view/edit the added appointments, the third element in the main page is a button which brings up the sections settings page which contains two elements, the first one is a button to add new section, the second one is a button to view/edit already existing section.

3.1.2 Hardware Interfaces.

The AS is divided into a phone App and a java-based program both of these doesn't require any hardware interfaces because the phone App only a phone to run it, also the java-based program needs only an up-to-date computer the only hardware interface we may need is a server to run the database.

3.1.3 Software Interfaces.

The facility program interacts with the database server which stores the information of the facility, the information of the customer, the appointments, and the sections, the relation between the facility program and the database server is read or modify, the facility program can read the customer information and their appointments which is only retaliated to this facility, also the facility program can read the information about the added sections and appointments from the database server, the facility program can modify the existing appointments and the existing sections and the facility profile and account information which are all stored in the database server.

The phone App also interacts with the database server which also stores the customer information and their account information and the appointments booked, the phone App can read and modify the database information, the phone App can read the facilities which are exist

in the system and their added sections and also the appointments added in this facility, the two only things that the phone App can modify are the appointment the customer booked, it adds it to the customer information in the database, the second thing that the phone App can modify is the customer profile and account information.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 <Functional Requirement or Feature #1> The phone App: Sign-up

3.2.1.1 Introduction

This method allows the user to enter his information and creates a new account with their information.

3.2.1.2 Inputs

[1] Phone number (Integer)

[2] E-mail (String)

[3] Password (String)

3.2.1.3 source

Inputs entered from the user keyboard

3.2.1.4 destination

The database.

3.2.1.5 requirements

The user data.

3.2.1.6 pre-condition

Valid phone number and a valid email.

3.2.1.7 post-condition

The user data should be stored in the database

3.2.1.8 Processing

*The method will receive the inputs **check 3.2.1.2** then the method will check if all the inputs are correct then using an imported library the method will send this information to the database server to store it.*

3.2.1.9 Outputs

If all the inputs are true and every condition is true the method will send the information to the database to store it and output to the user that the sign-up is done successfully.

3.2.1.10 Error Handling

If the user entered a wrong format for any inputs the method will throw an exception telling the user to re-enter the credentials.

3.2.2 <Functional Requirement or Feature #2> The phone App: Sign-in

3.2.2.1 Introduction

This method allows the user who already have an existing account to enter their information to access their account and use the phone App.

3.2.2.2 Inputs

[1] Phone number (Integer).

[2] Password (String).

3.2.2.3 source

Inputs entered from the user keyboard.

3.2.2.4 destination

Main loop then database

3.2.2.5 requirements

The user account data.

3.2.2.6 pre-condition

The user must have an account.

3.2.2.7 post-condition

The user should have the access to the app.

3.2.2.8 Processing

The method will receive the inputs in 3.2.2.2 and using an imported library will check this information if they exist in the database server.

3.2.2.9 Outputs

If all the inputs were correct and every condition is true the method will tell the user that the sign-in is done successfully and grant access to the user to use the phone App.

3.2.2.10 Error Handling

If the user entered an incorrect phone number or password the method will throw an exception telling the user that the phone number or password is wrong.

3.2.3 <Functional Requirement or Feature #3> The phone App: Find appointments

3.2.3.1 Introduction

This method will search the database for the available appointments based on the inputs the user entered and view them to the user.

3.2.3.2 Inputs

[1] Name of the facility (String)

[2] Section (String)

[3] year, month and day of the appointment (all Integers)

3.2.3.3 source

Inputs entered from the user keyboard and mouse.

3.2.3.4 destination

The database.

3.2.3.5 requirements

The appointment information.

3.2.3.6 pre-condition

The information of the appointment should be valid and the facility should exist in the system.

3.2.3.7 post-condition

The user finds the appointment.

3.2.3.8 Processing

After the user enters all the inputs correctly, using an imported library the method will search through the database for available appointments added on the section on the facility on the date the user entered.

3.2.3.9 Outputs

If the method found available appointments based on the date and the section the user entered it will open a new page with the appointments found with their time.

3.2.3.10 Error Handling

If the user entered a name of a facility that doesn't exist the method will throw an exception telling the user that this facility doesn't exist.

If the user entered a wrong date format the method will throw an exception telling the user that the date format is incorrect.

If the method didn't find any appointments based on the date and the section the user entered it will throw an exception telling the user that there are no appointments available.

3.2.4 <Functional Requirement or Feature #4> The phone App: Book an appointment

3.2.4.1 Introduction

This method allows the user to book an appointment.

3.2.4.2 Inputs

[1] Appointment (Appointment).

3.2.4.3 source

Inputs from method 'find appointments in 3.2.3'.

3.2.4.4 destination

The database.

3.2.4.5 requirements

The appointment the user wants to book.

3.2.4.6 pre-condition

The appointment should already exist and be found in find appointments method

3.2.4.7 post-condition

The appointment gets booked.

3.2.4.8 Processing

The method will take the inputs and by using an imported library the method will modify the user information in the database and enters the new information that the user wants.

3.2.4.9 Outputs

If every condition is true the method will book the appointment by the user information and tell the user that the appointment has been booked successfully.

3.2.4.10 Error Handling

Errors should be handled in the 'find appointments in 3.2.3' method, because all the inputs come from there.

If the user doesn't have a verified e-mail address the method will throw an exception and a message will pop up telling the user they need to have a verified account to book this appointment.

3.2.5 <Functional Requirement or Feature #5> The phone App: Customize account and profile information.

3.2.5.1 Introduction

This method allows the user to edit the profile and account information of their account.

3.2.5.2 Inputs

[1] Phone number (Integer)

[2] Name (String)

[3] E-mail (String)

[4] Password (String)

3.2.5.3 source

Inputs entered from the user keyboard

3.2.5.4 destination

The database.

3.2.5.5 requirements

The new user information.

3.2.5.6 pre-condition

All of the information should be in the correct format.

3.2.5.7 post-condition

The user profile information must be updated.

3.2.5.8 Processing

The method will take the inputs from 'find appointments in 3.2.3', and by using an imported library the method will book the appointment with the user name in the database and hide it so it never shows in find appointments until the user cancel it and sets two reminders one of them will be a SMS and the other one will be an e-mail message.

3.2.5.9 Outputs

If the inputs are all entered correctly and every condition is true the method will modify the user information and shows the new one and tell the user that they have been edited successfully.

3.2.5.10 Error Handling

If the user entered an incorrect format for any input the method will throw an exception telling the user which input has been entered incorrectly and tell the user to enter all the inputs again.

3.2.6 <Functional Requirement or Feature #6> The facility program: Add appointments.

3.2.6.1 Introduction

This method allows the facility program user to enter the required inputs and choose which section for an appointment then adds it to the facility appointments which are stored in the database server.

3.2.6.2 Inputs

[1] Date: year/month/day (All integer)

[2] Time (Integer)

[3] Branch name or location (String)

[4] Section name (String)

3.2.6.3 source

Inputs entered from the user keyboard

3.2.6.4 destination

The database.

3.2.6.5 requirements

The information of the appointment to be added.

3.2.6.6 pre-condition

That there are no existing appointments in the data base with the same time and date of the appointment to added.

3.2.6.7 post-condition

The appointment should be added and stored in the database.

3.2.6.8 Processing

This method first takes the inputs in 3.2.6.2 then it creates an appointment object then using an imported library the method will interact with the database server, then it will send the data to the database and search for the facility and the section then it stores them there.

3.2.6.9 Outputs

If all the inputs were entered correctly and every condition is true, the method will store the appointment in the database then it will print to user that the appointments has been added successfully.

3.2.6.10 Error Handling

If the user entered an incorrect format for any inputs an exception will be thrown and a message will be shown telling the user to re-enter the inputs in the correct format.

If there is an already existing appointments in the data base with the same time and date and section an exception will be thrown and a message will be shown telling the user that there is an already existing appointment with the same time and date.

3.2.7 <Functional Requirement or Feature #7> The facility program: Edit appointments.

3.2.7.1 Introduction

This method allows the facility program user to edit the information of the already added appointments.

3.2.7.2 Inputs

[1] Date: year/month/day (All integer)

[2] Time (Integer)

[3] Branch name or location (String)

[4] Section name (String)

3.2.7.3 source

Inputs entered from the user keyboard and mouse.

3.2.7.4 destination

The database.

3.2.7.5 requirements

The new information that will be updated in the appointment.

3.2.7.6 pre-condition

The appointment should exist and the information should be in the correct format.

3.2.6.7 post-condition

The appointment information must be updated.

3.2.7.8 Processing

This is an overloaded method takes any inputs from 3.2.7.2 then using an imported library the method searches the database for the old input then replaces it with the new one the user entered.

3.2.7.9 Outputs

If the inputs have been entered correctly and every condition is true the method will replace the new data with the old one and print “the edit has been done successfully”.

3.2.7.10 Error Handling

If the user entered an incorrect format for any inputs an exception will be thrown and a message will be shown telling the user to re-enter the inputs in the correct format.

If there is an already existing appointments in the data base with the same time and section and date with the new edited appointment an exception will be thrown and a message will be shown telling the user that there is an already existing appointment with the same time and date.

3.2.8 <Functional Requirement or Feature #8> The facility program: Add a new section.

3.2.8.1 Introduction

This method creates a new section with a dynamic size in the facility to add appointments in.

3.2.8.2 Inputs

[1] Section Name (String)

3.2.8.3 source

Inputs entered from the user keyboard

3.2.8.4 destination

The database.

3.2.8.5 requirements

The name of the section to be added.

3.2.8.6 pre-condition

That there is no existing section in the database with the same name of the section to added.

3.2.8.7 post-condition

The section should be added and stored in the database.

3.2.8.8 Processing

The method creates a new section object with the name the user entered which has linked list of type appointment to store appointments in.

3.2.8.9 Outputs

If the name were entered correctly and every condition is true the method will process then it will print to the user “the section has been added successfully”.

3.2.8.10 Error Handling

If the name format were entered incorrectly an exception will be thrown and a message will be shown telling the user to re-enter the name with the correct format.

If there is an already existing section with the same name an exception will be thrown and a message will be shown telling the user “there is already a section with the same name”

3.2.9 <Functional Requirement or Feature #9> The facility program: Edit section name.

3.2.9.1 Introduction

This method allows the facility program user to edit the already added sections names.

3.2.9.2 Inputs

[1] Section Name (String)

3.2.9.3 source

Inputs entered from the user keyboard.

3.2.9.4 destination

The database.

3.2.9.5 requirements

The new name of the section to be edited.

3.2.9.6 pre-condition

The section should exist and the name should be not used by any other section.

3.2.9.7 post-condition

The section name must be updated and stored in the database.

3.2.9.8 Processing

This is a method that takes the name inputs then using an imported library the method searches the database for the old name input then replaces it with the new one the user entered.

3.2.9.9 Outputs

If the inputs have been entered correctly and every condition is true the method will replace the new name with the old one and print “the edit has been done successfully”.

3.2.9.10 Error Handling

If the user entered an incorrect format for the name input an exception will be thrown and a message will be shown telling the user to re-enter the name in the correct format.

If there is an already existing section in the data base with the same name with the new edited section an exception will be thrown and a message will be shown telling the user that there is an already existing section with the same name.

3.2.10 <Functional Requirement or Feature #10> The facility program: remove section appointments.

3.2.10.1 Introduction

This method allows the facility program user to remove appointments from the already added sections.

3.2.10.2 Inputs

[1] Date: year/month/day (All integer)

3.2.10.3 source

Inputs entered from the user keyboard or mouse.

3.2.10.4 destination

The database.

3.2.10.5 requirements

The date of the appointment to be removed.

3.2.10.6 pre-condition

The appointment must be existing to be removed.

3.2.10.7 post-condition

The appointment must be removed from the section and database.

3.2.10.8 Processing

This method takes the inputs in 3.2.10.2 and can access the linked list in the section and search for the appointment to be removed based on the input the user entered then removes it from the list and by using an imported library the method will search for the section then removes the appointment from it.

3.2.10.9 Outputs

If the inputs have been entered correctly and every condition is true the method will remove the appointment and print “the remove has been done successfully”.

3.2.10.10 Error Handling

If the user entered an incorrect format for the input an exception will be thrown and a message will be shown telling the user to re-enter the date in the correct format.

If there is no existing appointment in the data base with the same date an exception will be thrown and a message will be shown telling the user “there is no existing appointment with the same date to be removed”.

3.2.11 <Functional Requirement or Feature #11> The facility program: Transfer appointments.

3.2.11.1 Introduction

This method allows the facility program user to transfer the already added sections appointments.

3.2.11.2 Inputs

[1] The section name (String).

[2] Date of the appointment: year/month/day (All integer)

3.2.11.3 source

Inputs entered from the user keyboard.

3.2.11.4 destination

The database.

3.2.11.5 requirements

The section name that the appointment will be transferred to and the appointment date that will be transferred.

3.2.11.6 pre-condition

The section that the appointment will be transferred to should exist and also the appointment.

3.2.11.7 post-condition

The appointment should be successfully transferred to the section the user entered.

3.2.11.8 Processing

*This method takes the inputs in **3.2.11.2** and using an imported library the method searches the database for the section name that the appointment will be transferred to, if the method finds it will add this appointment to the new section and removes it from the old one.*

3.2.11.9 Outputs

If the name input has been entered correctly and every condition is true the method will transfer the appointment to the new section and it will print to the user “the transfer has been done successfully”.

3.2.11.10 Error Handling

If the user entered an incorrect format for the name input an exception will be thrown and a message will be shown telling the user to re-enter the name in the correct format.

If there is no existing section in the data base with the same name that the appointment will be transferred to an exception will be thrown and a message will be shown telling the user "there is no existing section with the same name to transfer to”.

3.2.12 <Functional Requirement or Feature #12> The facility program: Customize account and profile information.

3.2.12.1 Introduction

This method allows the facility program user to edit the profile and account information of their account.

3.2.12.2 Inputs

[1] Name (String)

[2] facility description (String)

[3] E-mail (String)

[4] Password (String)

3.2.12.3 source

Inputs entered from the user keyboard.

3.2.12.4 destination

The database.

3.2.12.5 requirements

The facility data and information.

3.2.12.6 pre-condition

The information should be in the correct form.

3.2.12.7 post-condition

The facility profile information must be updated in the program and the database.

3.2.12.8 Processing

*The method will take the inputs in **3.2.11.2** and by using an imported library the method will search the database for the old information and replace them with the new one.*

3.2.12.9 Outputs

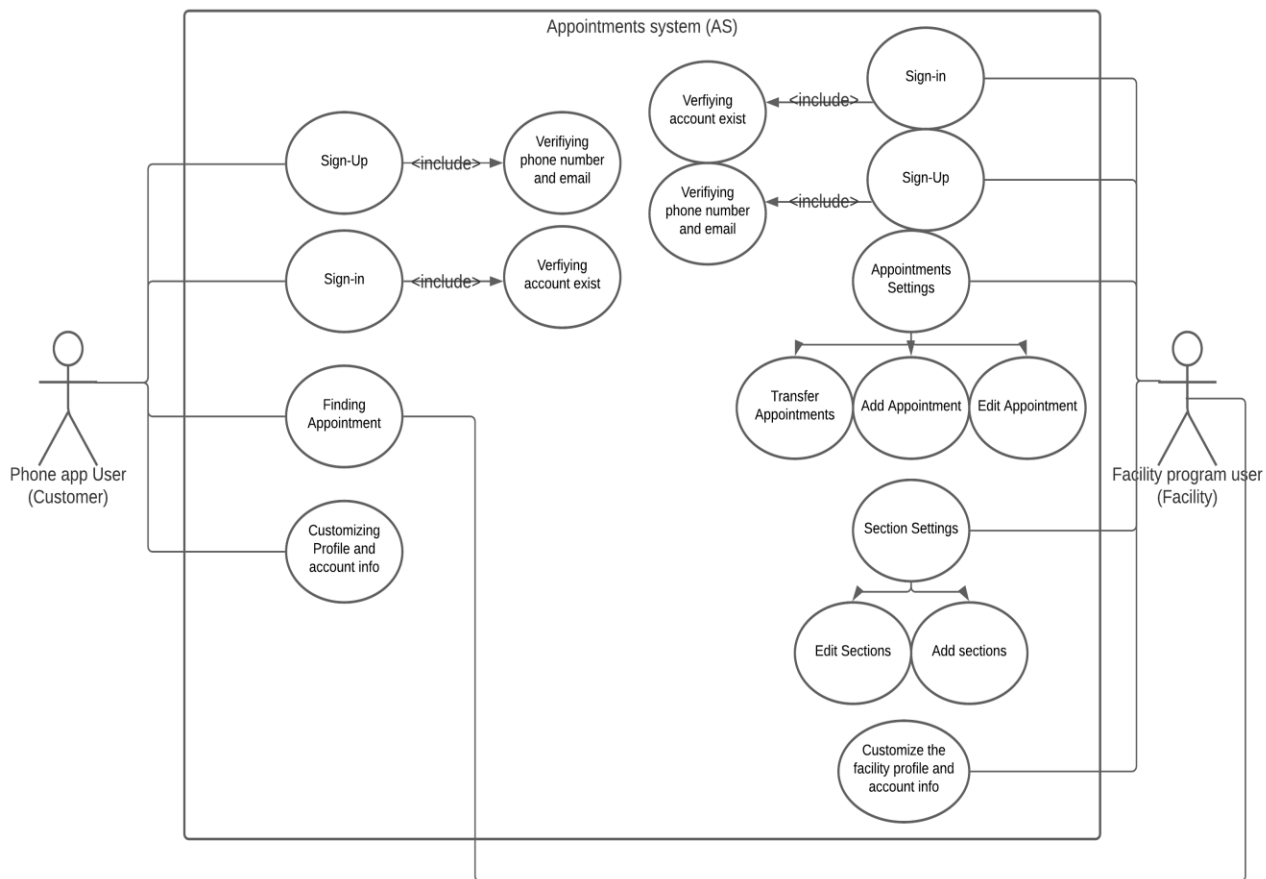
If the inputs are all entered correctly and every condition is true the method will modify the user information and shows the new one and print to the user “the edit has been edited successfully”.

3.2.12.10 Error Handling

If the user entered an incorrect format for any input an exception will be thrown and a message will be shown telling the user “re-enter the inputs in the correct format”.

3.3 Use cases diagram

This will be the use case diagram which is a representation of the user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



3.4 Non-Functional Requirements

Reliability: AS works for 98% of the time for the average number of users, for 24/7.

Usability: AS GUI is user-friendly, it is very simplistic and not complex at all, it is self-explanatory.

Availability: The primary actor in this system is the customer, the customer app is a phone app which is widely available because nowadays nearly everyone has a phone.

Manageability: When editing the code of the facility program the phone App stays up running without stopping.

Security: the facility program users are the only users who can view the customer information who booked and add sections and add appointments and transfer them.

Maintainability: if editing appointments or editing sections features became unavailable, they can be maintained and be back within 2-3 hours.

Data Integrity: The AS shall secure the data by keeping backups of all the records to the database every day.

4. Team Members Contributions

<i>Member</i>	<i>Roles/Activities</i>
<i>Saleh</i>	<i>Writing, researching, evaluating, brainstorming</i>
<i>Rayan</i>	<i>Researching, evaluating, brainstorming</i>
<i>Fahad</i>	<i>Researching, evaluating, brainstorming</i>
<i>Abdulelah</i>	<i>Researching, evaluating, brainstorming</i>
<i>Riyadh</i>	<i>Researching, evaluating, brainstorming</i>

5. Conclusion

In the end this SRS document explained the whole purpose and talked about the intended audience of this document, also the scope of the AS which was about the origin of need and a high-level description of the system functionality, then it explained its perspective which was about the relationship this product has and how it's functions and the user characteristics and the main constraints which include hardware, software and network on the system, then in detailed; we explained the specific requirements of the system and its functional requirements and non-functional ones, in the meantime we finished the first phase which is about the SDP and the second phase which is about the SRS document in the future our team intend to complete the design phase and the implementation phase as well.