**Doctor Appointments Reservation**

V 0.1

By John Shahla

4th year 1st subclass

**1.0 Introduction:**

***1.1. Purpose***

The purpose of this document is to provide a detailed description of the Doctor appointments reservation system, it will explain the details, features and advanced features of the system and how to operate it

***1.2. Scope of project***

The purpose of this system is to provide reservation service for patients, which the patient and reserve, confirm and cancel an appointment, which otherwise performed manually by the doctor or by their assistant, the first version of the app requires interference to prevent spam, although some spam measures could be applied but requires more expensive procedures which will be explained later.

The system also allows the doctors or assistants to handle appointments easier and more efficiently.

The system will contain a non-relational database system (nosql)

***1.3. References***

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

***1.4. Further Development***

**WILL NOT BE INCLUDED IN V 1.0**

**AntiSpam:**The system will be able to send a verification SMS to the patient to confirm the reservation and to lock it in the database

***1.5. Overview of Document***

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

**2.0. Overall Description**

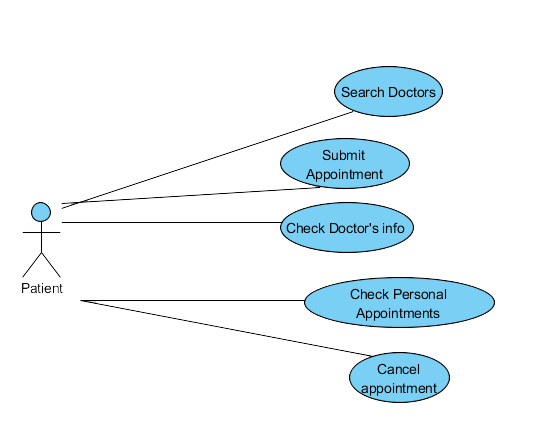
***2.1 System Environment***

The

**2.2. Functional Requirements Specification**

**Knowing that the system has been installed on the user's machine**

**2.2.1** Patient Use cases



Use Case: **Search doctors**

**Diagram**



**Brief Description**

The patient can search for doctors based on name, specialization, location and payments methods

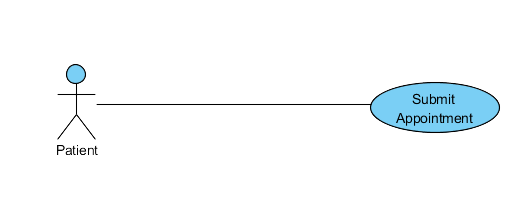
**Initial Step-By-Step Description**

1. The patient chooses to search for a doctor based on name, specialization, location and payments methods
2. The system displays the results
3. The patient can select a specific doctor
4. The system will provide the schedule for this doctor, and brief information about this doctor

**Xref:** Section 3.2.1, Search Doctors

Use case: **Submit an appointment request**

**Diagram:**



**Brief description:**

The patient can request an appointment

**Initial Step by Step Description:**

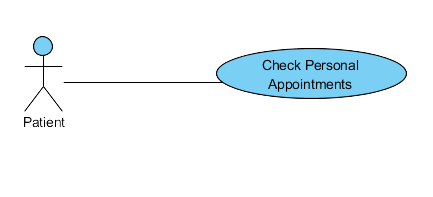
Before this step can be initiated the patient has already chosen a doctor and they are on the schedule screen

1. The patient can browse available time slots
2. The patient chooses the wanted time slot
3. The patient submits an appointment request

**Xref:** Section 3.2.2, Submit reservation

Use Case: **Check personal reservations**

**Diagram**



**Brief Description**

The patient checks their own reservations

**Initial Step-By-Step Description**

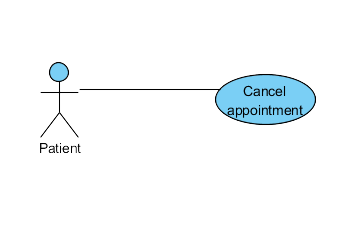
No earlier steps needed, it can be empty

1. The patient chooses the My reservation button
2. The system displays the personal reservation information

**Xref:** Section 3.2.3, Check personal Pateint

Use Case: **Cancel Reservation**

**Diagram:**



**Brief Description**

The patient cancels a requested or a confirmed reservation

**Initial Step-By-Step Description**

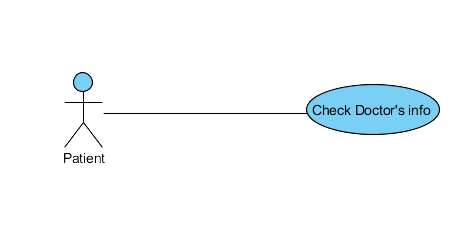
Before this step the patient must already had requested an appointment, regardless of the confirmation status and had navigated to the My appointment section

1. The patient chooses the specific reservation
2. The patient can cancel the reservation

**Xref:** Section 3.2.4 Cancel reservation Patient

Use Case: **Check Doctor’s info**

**Diagram**



**Brief Discretion:**

The Patient can check doctor’s info

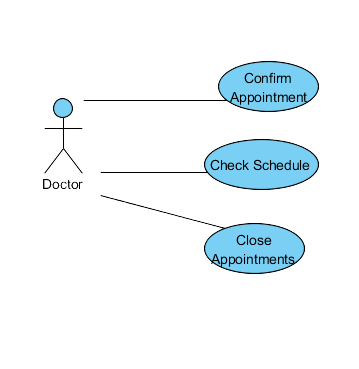
**Initial Step-By-Step Description**

Before this step, the patient must already had searched and chose for a specific doctor

1. The patient chooses to check the doctor’s information
2. The system displays the requested information

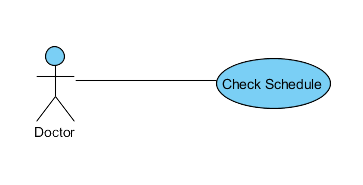
**Xref:** Section 3.2.5, Check info

**2.2.2 Doctor Use Cases**



Use Case: **Check Doctor’s own schedule**

**Diagram**



**Brief Description**

The doctor is able to check their own schedule

**Initial Step-By-Step**

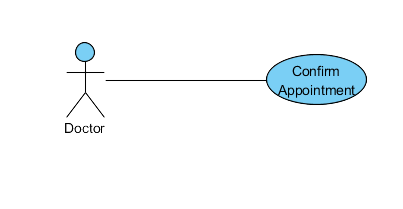
Must have doctor’s privilege

1. The doctor requests the schedule from the system
2. The system displays their schedule

**Xref:** Section 3.2.6, Check Schedule Doctor

Use case: **confirm Appointment**

**Diagram**



**Brief Description**

The doctor can confirm a requested reservation

**Initial Step-By-Step Description**

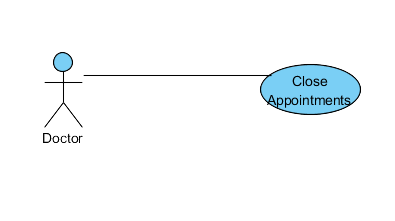
The patient must had been requested an appointment

1. The doctor checks their own schedule
2. They choose to confirm or deny appointment
3. The system will reserve this time slot as reserved and can’t be reserved again

**ْXref:** Section 3.2.7, Confirm

Use case: **Close Appointments**

**Diagram**



**Brief Description**

The doctor can close appointments for a certain time slots or days in which these time slots will not be valid to reserve by patients

**Initial Step-By-Step Description**

Must have doctor’s privilege

1. The doctor checks their own Schedule
2. The system displays the schedule
3. The doctor chooses what time slots or days to be closed
4. The system will not display these slots as open for patients
5. If there were reserved appointments in those slots the system will display the cancelled patients contact info

**Xref:** Section 3.2.8, Close Appointments

***2.3 User Charactaristics***

The users should Internet and application literate

***2.4 Non-Functional Requirements***

1. The system will be installed on a high-speed server and will have a dedicated backend structure
2. Applied antispam measures
3. Easy access
4. Simple Ui
5. Interactive

**3.0. Requirements Specification**

***3.1 External Interface Requirements***

No external Interface Required

***3.2 Functional Requirements***

**3.2.1** Search Doctors

|  |  |
| --- | --- |
| **Use Case Name** | Search Doctors |
| **XRef** | Section 3.2.1, Search Article  SDD, -Not Set- |
| **Trigger** | The patient accesses the app |
| **Precondition** | The search menu is displayed with advanced search |
| **Basic Path** | * The patient chooses how to search according to their desired choices being by name, location and specialization * Filters can be applied such as payment methods * The system displays a list of doctors * The patient selects a doctor * The system displays brief information about the doctor and their schedule |
| **Alternative Paths** | -- |
| **Postcondition** | The system displays brief information about the doctor and their schedule |
| **Exception Paths** | The patient may abandon the search at any time. |
| **Other** | -- |

**3.2.2 Submit Appointment**

|  |  |
| --- | --- |
| **Use Case Name** | Submit Appointment |
| **XRef** | Section 3.2.2, Submit appointment  SDD –NOT SET- |
| **Trigger** | The patient clicks on the submit button |
| **Precondition** | The patient must already had been chosen a time slot |
| **Basic Path** | * The patient clicks the submit button to confirm a submission |
| **Alternative Paths** | - |
| **Postcondition** | The system registers the submissions, and displays that their submissions is registered |
| **Exception Paths** | - |
| **Other** | - |

**3.2.3** Check Personal Appointments

|  |  |
| --- | --- |
| **Use Case Name** | Check Personal Appointments |
| **XRef** | Section 3.2.2  SDD -Not Set- |
| **Trigger** | The Patient selects the My Appointments tab |
| **Precondition** | The patient is using the app and they are on the main menu |
| **Basic Path** | The system displays the personal appointment schedule |
| **Alternative Paths** | - |
| **Postcondition** | The system displays the personal appoinbtment schedule |
| **Exception Paths** | The schedule can be empty |
| **Other** |  |

**3.2.4** Cancel Appointment patient

|  |  |
| --- | --- |
| **Use Case Name** | Cancel Appointment patient |
| **XRef** | Section 3.2.4  SDD –NOT SET- |
| **Trigger** | The patient clicks on an appointment in their personal schedule  Or they can click on it from the doctor’s screen |
| **Precondition** | An appointment must have been submitted or confirmed |
| **Basic Path** | * The Patient clicks on the appointment they wish to cancel * Another window pops out asking if they want to commit to this cancelation * The patient confirms |
| **Alternative Paths** | - |
| **Postcondition** | The appointment is cancelled and no longer visible on the doctor’s schedule |
| **Exception Paths** | - |
| **Other** | - |

***4.0 API Reference***

Domain Name: -NOT SET-

**Namese** is for selecting to search doctors by name

And using the **Name** key for the doctors' name

Ex: Domain/Namese=1&Name=Ahmad

**Advse** is for selecting the advanced search feature

**Cloc** is for choosing country location

**Sloc** is for choosing State location

**Payinfo** is for choosing payment methods

**Specilize** is for choosing the specialization of the doctor

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** | **Required?** |
| Namese | 1,0 | Yes |
| Name | String | No |
| Advse | 1,0 | Yes |
| CLoc | String | No |
| Sloc | String | No |
| Payinfo | String | No |
| Specialize | String | No |

The response will be a json for each page of results

For search:

Response = { “pages” : 3 //total number of page results

“page” : 1

“ResultsCount” : 10 //Can be maximum 10

//Can be changed

“Doctors” : {

1: {

“Name” : “Ahmad”

“Cloc” : “Syria”

“Sloc” : Homs

“Specilize” : “somemedicalword”

}}}

The Response for a doctor :

Res = {

“Name” : “Ahmad”

“Cloc” : “Syria”

“Sloc” : Homs

“Specilize” : “somemedicalword”

“PhoneNum” : 0389450845

“SlotDuration” : 30

“startday” : “9:00”

“endday” : “14:00”

“Week1” : {

“Sun” : {

“slot1” : 1 // 1 for taken

“slot2” : 0

…..}}}

Response for Doctor’s info

Res = {“Name” : “Ahmad”

“Cloc” : “Syria”

“Sloc” : Homs

“Specilize” : “somemedicalword”

“PhoneNum” : 0389450845

“StudyLoc” : “Syria”

“Uni” : “Albaath”

“yearsinpro” : 6

}

Response for personal schedule

Res = {

“App1” : {

“Date” : “14,1,2021”

“Day” : “Monday”

“time” : “13:30”

“Doctor” : “Ahamd”

“Location” : {

“Country” : “Syria”

“Gov” : “Homs”

“Street” : “AlHamra”

“Otherifno” : “In front of the fire station”}}}