1. GENERAL DESCRIPTION

Hospitals are very important as they provide medical treatment to the patients with a range of diseases. Thus, it is essential that the data related to hospitals shall be stored safely.

## 1.1 Purpose

The main purpose of this system is to minimize the paperwork of the hospital records and to fetch the details quickly and accurately.

## 1.2 Scope

The scope of Hospital Management System is global. It will be a web based and android application that will allow multiple users to enter the system using their CNIC and password for making appointments or to access the appointment history.

## 1.3 Abbreviation

HMS Hospital Management System

## 1.4 Overview

HMS will have a user interface that will make the communication easy between user and the system. The record of appointments will be stored and accepted, declined or deleted appointments will be shown to the patients and can also be updated. The data of the patients will be maintained. The main objective is to provide a quick access to the information. The GUI will be made in a way that anyone could understand what is going on.

# 2. OVERALL DESCRIPTION

## 2.1 Product Perspective

Our product HMS will be an efficient system that manages activities of the hospital. Back end of the system will be implemented using Cloud Firestore for storing all the data of the hospital.

## 2.2 Product Functions

The following are the product functions of the HMS:

* Multi user account system
* Monitoring the whole hospital system
* Appointment History
* Responsive User Interfaces
* Appointments will be managed and recorded
* Medicine Inventory where the data of all the drugs is stored

## 2.3 User Characteristics

The person using the system will have a CNIC and password and only then the user’s appointments can be stored; should be registered first. This is to maintain the integrity of all the hospital data as privacy is a priority.

## 2.4 General Constraints

The user must have a valid CNIC and password. Certain user must log in into the system to access any data if they are authorized. When the server is updating the system would not work. The system shall use Google Cloud Firestore as database. The system will be a web-based and android application.

## 2.5 Assumptions and Dependencies

The system should be able to work when the user traffic is high i.e when multiple users are on the system. The system will only work as long as the server is working. The admin would be the only one who can delete the database.

# 3. SPECIFIC REQUIREMENTS

## 3.1 External Interface Required

### 3.1.1 User Interfaces

The external users of this system are the doctors, patients, the admin, and pharmacists. The admin has a control over the system and can monitor all the activities going on in the hospital. The doctor can make, cancel and delete appointments, can view the details of the patient. The patient can make appointments with the available doctors. The pharmacist will manage all the medicines in the hospital and can change the quantity of the medicines.

### 3.1.2 Hardware Interfaces

The external hardware used for accessing the HMS will be any device of the patients, doctors, etc. having a wireless LAN connection with web browsers supporting HTML 5. An application will also be available for the android users.

### 3.1.3 Software Interfaces

The Operating Systems can be any version of Windows, Linux, Unix or Mac which supports TCP/IP protocols, web accessing and writing. For the android application users, the minimum API level is 15.

### 3.1.4 Communication Interfaces

The communication interface is a local area network through wireless network routers.

## 3.2 Performance Requirements

The PCs used must be at least Pentium 4 machines so that they can give optimum performance of the product. For the android application users, the minimum API level is 15.

## 3.3 Design Constraints

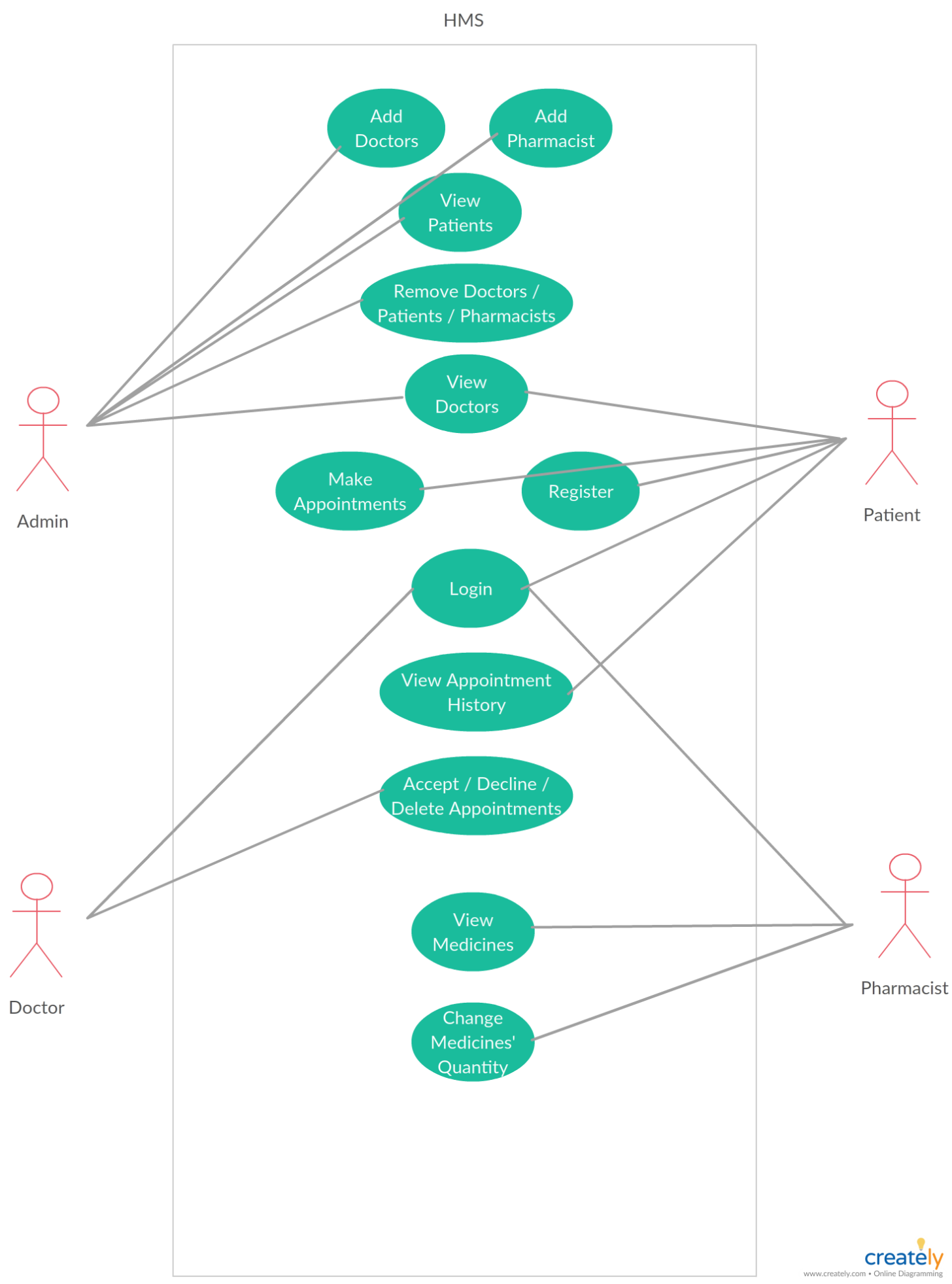
The constraints at the designing time are that many patients may be asking for an appointment at the same time so the designers must keep this in view and design the product in this way that it is easily updatable. There should also be real-time accessing of the database.

## 3.4 Software Quality Attributes

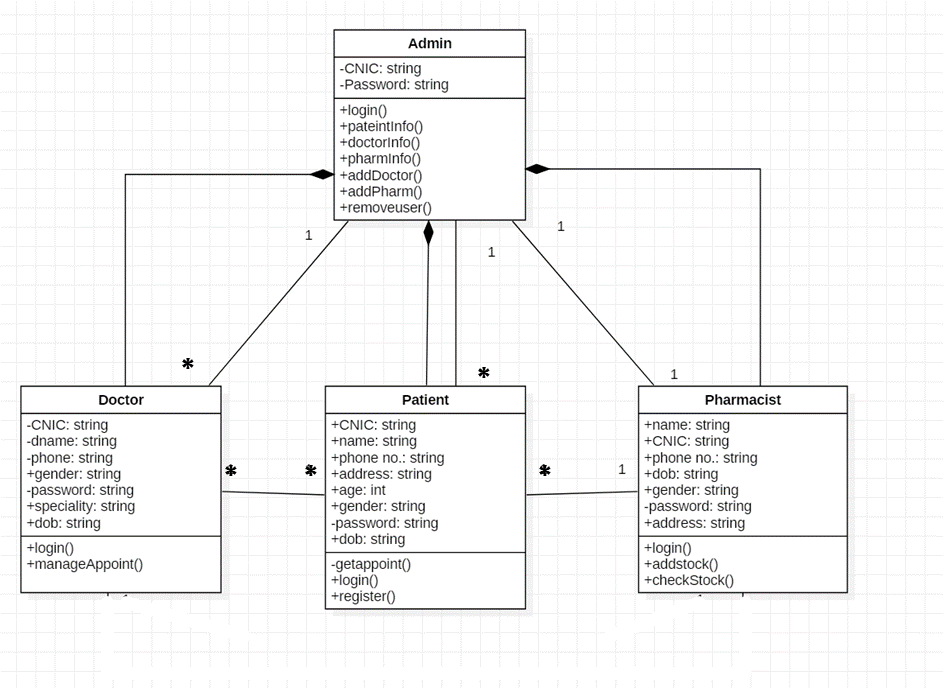
The system should be implemented in such a way that it is user friendly, reliable, secure and manageable. Also, the records should be updated easily and the system should be fast.

# 4. UML DIAGRAMS:

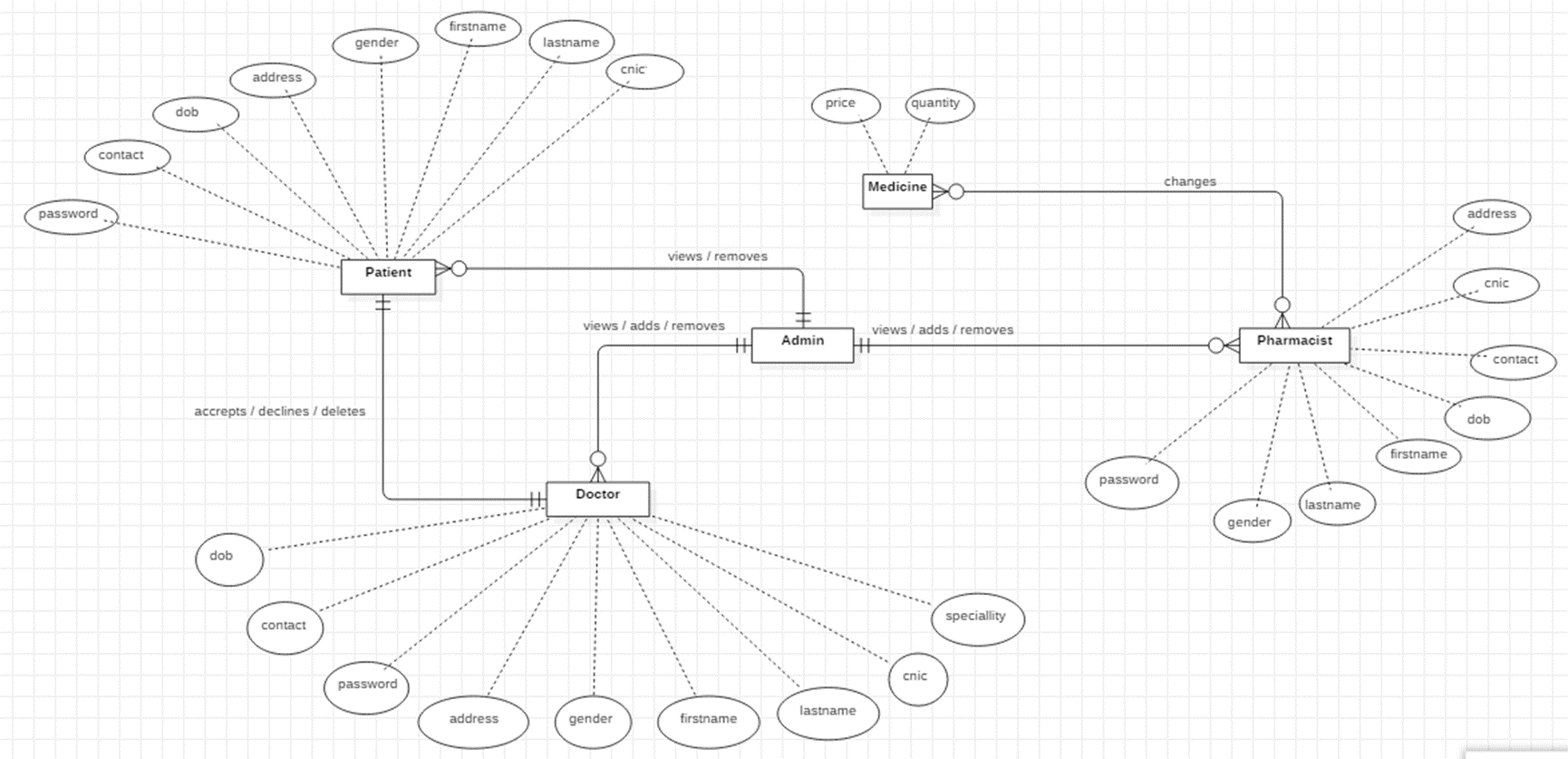
## 4.1 Use Case:



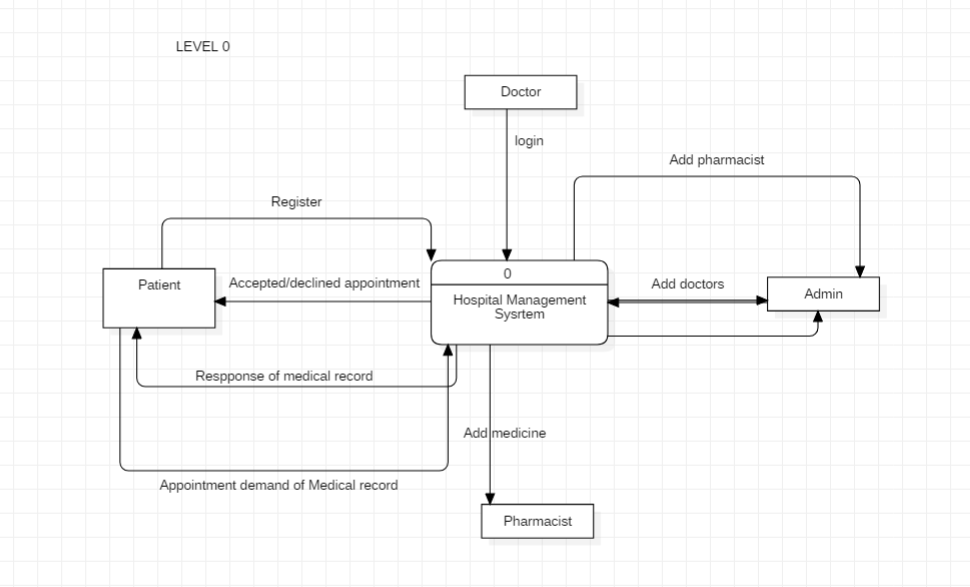
## 4.2 Class Diagram:

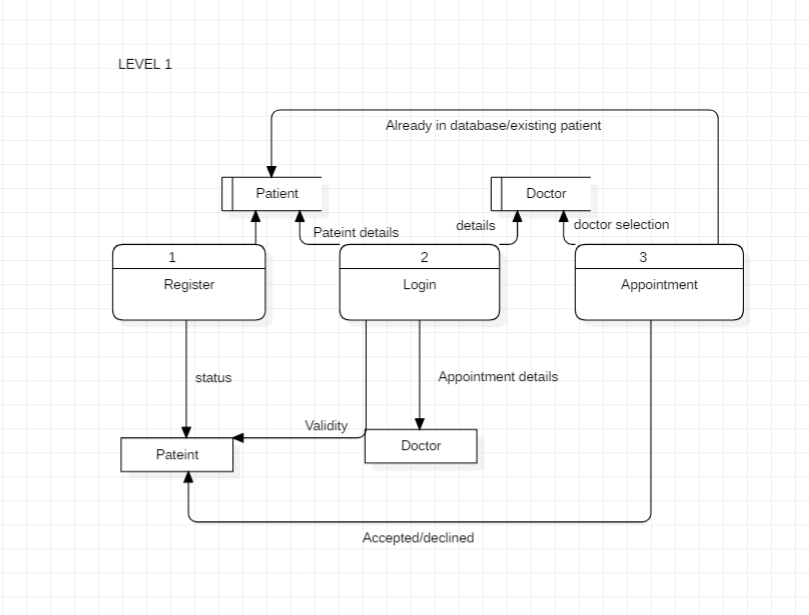


## 4.3 ER Diagram:

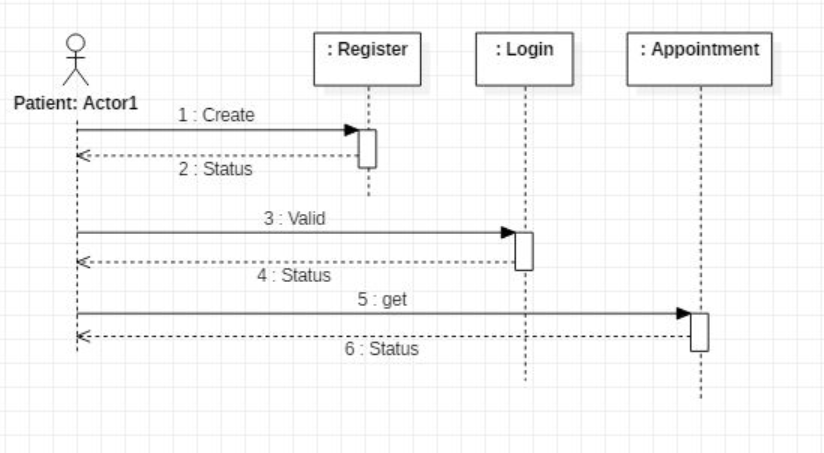


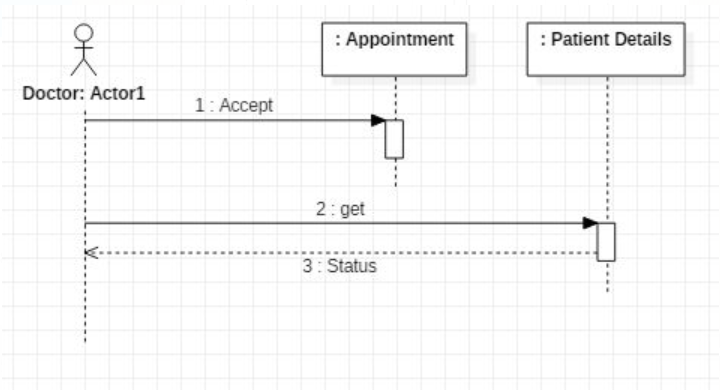
## 4.4 Data Flow Diagram:

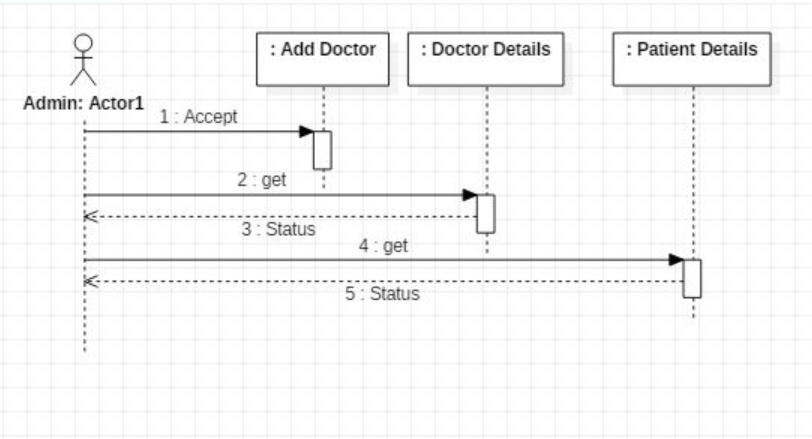




## 4.5 Sequential Diagram:







## 

# 5. TEST CASES:

|  |  |
| --- | --- |
| **Test Case 1** | |
| **Test Case ID:** PatientRegistration | **Test Designed by:** |
| **Test Priority (Low/Medium/High):** High | **Test Designed date:** 15/12/2018 |
| **Module Name:** RegisterPatient | **Test Executed by:** |
| **Test Title:** Verify Patients’ Registration | **Test Execution date:** 15/12/2018 |
| **Description:** Checks whether the patient has entered the right cnic length |  |
| **Pre-conditions:** Working internet connection, Web browser supports JavaScript / Android SDK >= 15 | |
| **Dependencies:** null | |

**Positive Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to patient registration page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\*\*\*\*\* |  |  |  |
| 4 | Provide valid cnic | CNIC: |  |  |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address, password |  |  |  |
| 6 | Click on register button |  | User is registered | Confirmation message shows up, user is registered and the information is stored in database. | Pass |

**Post-condition:**

The person is now a verified user and can access the further facilities of HMS.

**Negative Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to patient registration page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\* |  |  |  |
| 4 | Provide valid password | CNIC: /  3 |  | Error message: cnic length is not 13 |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address |  | Error message: Provide all details |  |
| 6 | Click on register button |  | User not registered | Confirmation message shows up. User not registered | Pass |

**Post Condition:**

The person is not registered and cannot access hospital facilities.

|  |  |
| --- | --- |
| **Test Case 2** | |
| **Test Case ID:** AddDoctorPharmacist | **Test Designed by:** |
| **Test Priority (Low/Medium/High):** High | **Test Designed date:** 15/12/2018 |
| **Module Name:** UsersAdd | **Test Executed by:** |
| **Test Title:** Verify Doctor’s Registration | **Test Execution date:** 15/12/2018 |
| **Description:** Checks whether the admin has entered the right cnic length of the doctor/patient |  |
| **Pre-conditions:** Admin logged in | |
| **Dependencies:** Test Case: PatientRegistration (positive input) | |

**Positive Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to add user page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\*\*\*\*\* |  |  |  |
| 4 | Provide valid cnic | CNIC: (length=13) |  |  |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address, password, specialty (for doctor) |  |  |  |
| 6 | Click on register button |  | User is registered | Confirmation message shows up, user is registered and the information is stored in cloud. | Pass |

**Post-conditions:**

The person is now a verified user and can access the further facilities of HMS.

**Negative Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to add user page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\* |  |  |  |
| 4 | Provide valid password | CNIC: (length=12) / CNIC: (length=14) |  | Error message: cnic length is not 13 |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address |  | Error message: Provide all details |  |
| 6 | Click on register button |  | User not registered | Error message shows up. User not registered | Pass |

**Post Condition:**

The user is not registered and cannot access hospital facilities.

|  |  |
| --- | --- |
| **Test Case 3** | |
| **Test Case ID:** UserLogin | **Test Designed by:** |
| **Test Priority (Low/Medium/High):** Med | **Test Designed date:** 15/12/2018 |
| **Module Name:** Login | **Test Executed by:** |
| **Test Title:** Verify login details | **Test Execution date:** 15/12/2018 |
| **Description:** |  |
| **Pre-conditions:** User registered, Working internet connection | |
| **Dependencies:** Test Cases: PatientRegistration, AddDoctorPharmacist (positive inputs) | |

**Positive Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to register page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\*\*\*\*\* |  |  |  |
| 4 | Provide valid cnic | CNIC: |  |  |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address, password |  |  |  |
| 6 | Click on signup button |  | User is registered | Confirmation message shows up, user is registered and the information is stored in database. | Pass |

**Post-conditions:**

The person is now a verified user and can access the further facilities of HMS.

**Negative Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to register page |  |  |  |  |
| 2 | Provide valid Name | Name: |  |  |  |
| 3 | Provide valid password | Password: \*\* |  |  |  |
| 4 | Provide valid password | CNIC: /  3 |  | Error message: cnic length is not 13 |  |
| 5 | Provide all details | First Name, Last Name, Gender, dob, contact, cnic, address |  | Error message: Provide all details |  |
| 6 | Click on signup button |  | User not registered | Confirmation message shows up. User not registered | Pass |

**Post Condition:**

The person is not registered and cannot access hospital facilities.

|  |  |
| --- | --- |
| **Test Case 4** | |
| **Test Case ID:** MakeAppointment | **Test Designed by:** |
| **Test Priority (Low/Medium/High):** Med | **Test Designed date:** 15/12/2018 |
| **Module Name:** Get Appointment | **Test Executed by:** |
| **Test Title:** Request Appointment | **Test Execution date:** 16/12/2018 |
| **Description:** Patient selects doctor and appointment time. Sends this request to the selected doctor |  |
| **Pre-conditions:** The user must be registered first | |
| **Dependencies:** Test Cases: PatientRegistration, UserLogin (positive inputs) | |

**Positive Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to appointment page |  |  |  |  |
| 2 | Choose valid Specialty | Specialty: Dermatology |  |  |  |
| 3 | Provide valid appointment date range and time | Date: 2018-12-16 to 2018-12-20  Time: 11:30:00 | Valid Date | Valid Date |  |
| 4 | Click on Make an Appointment button |  | Appointment request is sent. | Request sent for appointment. | Pass |

**Post-conditions:**

The user will then have an appointment scheduled with the chosen doctor.

**Negative Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigate to appointment page |  |  |  |  |
| 2 | Choose valid Specialty | Specialty: Dermatology |  |  |  |
| 3 | Provide valid appointment time | Date: 2018-11-16 to 2018-12-20  Time: 11:30:00 | Error: Dates not in same month | Error: Dates not in same month |  |
| 4 | Click on Make an Appointment button |  | Appointment request failed | Request not sent for appointment. | Pass |

**Post-conditions:**

The user will have to again put the right date to request appointment.

|  |  |
| --- | --- |
| **Test Case 5** | |
| **Test Case ID:** ConfirmAppointment | **Test Designed by:** |
| **Test Priority (Low/Medium/High):** Medium | **Test Designed date:** 15/12/2018 |
| **Module Name:** DocAppointment | **Test Executed by:** |
| **Test Title:** Appointment Confirmation | **Test Execution date:** 15/12/2018 |
| **Description:** Doctor accepts, declines or deletes the appointments and patient is notified. (Must touch the button 2 times for android application users) |  |
| **Pre-conditions:** The doctor must be registered, logged in and appointment must be successful. Patient must request an appointment first. | |
| **Dependencies:** Test Cases: AddDoctorPharmacist, UserLogin (positive inputs) | |

**Positive Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Login as Doctor |  | Opens doctor’s portal | Opens doctor’s portal |  |
| 2 | Accept/Decline/Delete request | Touches Accept/Decline/Delete button twice | Appointment is accepted/declined/deleted | Appointment is scheduled. Patient is notified | Pass |

**Post-conditions:**

The patient can now meet the doctor at the specified date and time.

**Negative Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Login as Doctor |  | Opens doctor’s portal | Opens doctor’s portal |  |
| 2 | Accept/Decline/Delete request | Touches Accept/Decline/Delete button once | Appointment is not accepted/declined/deleted | Appointment is not scheduled. Patient is not notified | Pass |

**Post-conditions:**

The appointment is neither accepted nor declined. Wait for doctor’s response.

# 6.ADVANTAGES AND LIMITATIONS:

## 6.1 Advantages

* Reduces manual labour
* Works on almost every device
* Easy Registration
* Less registration errors
* Fast data access
* Computerization
* Real time database
* Fast status check
* Easy to use GUI
* Appointment history
* Responsive User Interfaces

## 6.2 Limitations

Some limitations of this system are:

* Doctors / Pharmacist cannot update their details
* Patients can update their details if and only if they register again with the same CNIC
* Pharmacist cannot change quantity / view medicines on the webpage
* Appointment does not hide for the doctor when he deletes it. Only the status changes (For the website)
* Pharmacist cannot delete the medicine records
* When the admin adds a doctor, it does not automatically updates on the appointment form (For the website)

## 6.3 Future Uses

* Connect to NADRA using CNIC
* Biometric verification
* Get all details directly from the NADRA Database
* Login using PIN code generated by the system
* Notifications about appointments directly by sending an SMS
* Medical Report generation