**Software Requirement and Design** **Specifications Report**

**HOSPITAL AND MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| Course Code | CS 3004 |
| Instructor | Sir Abdul Rehman |
| Project Team | Abdul Ahad Shaikh (20K-0390)  Ali Jodat (20K-0155)  Basil Ali Khan(20K-0477) |
| Submission Date | 05/Dec/2022 |

**[Instructions]**

* No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.
* All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.
* This’ Instruction’ section should also be removed in final document.

**Table of Contents**

1. **INTRODUCTION 5**
   1. Purpose of Document 5
   2. Intended Audience 5

**2. OVERALL SYSTEM DESCRIPTION 6**

* 1. Project Background 6
  2. Project Scope 6
  3. Not In Scope 6
  4. Project Objectives 6
  5. Stakeholders 6
  6. Operating Environment 6
  7. System Constraints 6
  8. Assumptions & Dependencies 6

**3. EXTERNAL INTERFACE REQUIREMENTS 7**

* 1. Hardware Interfaces 7
  2. Software Interfaces 7
  3. Communications Interfaces 7

**4. FUNCTIONAL REQUIREMENTS** 8

* 1. Use Cases 8

**5. NON-FUNCTIONAL REQUIREMENTS** 9

* 1. Performance Requirements 9
  2. Safety Requirements 9
  3. Security Requirements 9
  4. User Documentation 9 10

1. **SYSTEM ARCHITECTURE 11**
   1. SYSTEM LEVEL ARCHITECTURE 11
   2. SOFTWARE ARCHITECTURE 11
2. **DESIGN STRATEGY 12**
3. **DETAILED SYSTEM DESIGN 13**
   1. DATABASE DESIGN 13
4. APPLICATION DESIGN 15
5. REFERENCES 15

# 1. Introduction

### 1.1. Purpose of Document

The purpose of this document is to provide a detailed view of the project we have been working on.

### 1.2. Intended Audience

This application is built keeping in view the doctors, patients and random user.

#### [Definition of Terms, Acronyms and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]

|  |  |
| --- | --- |
| **Term** | **Description** |
| ASP | Active Server Pages |
| DD | Design Specification |
| UCD | USE CASE DIAGRAM |
| URL | UNIFORM RESOURCE LOCATOR |
|  |  |
|  |  |
|  |  |
|  |  |

##### 1.3 Document Convention

Font style: Times New Roman.

Font size: 12

For headings: (As given) Font style: Times new Roman.

Font size: 14

Font decoration: Italic, and Bold.

# 2. Overall System Description

### 2.1. Project Background

The idea behind this project is to provide an easiest and free way to get a person register and login as patient and they can easily book appointments and cancel them and buy medicines decreasing counter chaos just have to go to pharmacy and show token number their order will be given to them. Doctor can easily register through admin and can login easily update their data and can check appointments. Pharmacist can easily login and update stocks easily.

### 2.2. Project Scope

### This project is a Hospital Management System application built on Flutter. This will be used by three type of users;

### 1. Admins,

### 2. Doctors ,

### 3. Patients

### 4. Pharmacist

### There will be preset Admin, who will be allowed to sign-in with their own email and password, stored in the database, and no other email and password will be accepted. Admin can register and remove doctors and can search all the doctors with the help of this app. User can register and login as Patient with their own email and password. Their data will be stored in the database as well. Patient can calculate BMI and book and cancel appointment.

### Random user can only buy pharmacy buy generating token and their order will be given to them by pharmacist through then saved in database.

### 2.3. Not In Scope

Not Applicable.

### 2.4. Project Objectives

In today’s world, people have started to rely more on applications. They just need everything to be done within few taps on the screen. And people avoid crowded place and long lines just to see if doctor available or not so through this it people can easily book appointments and buy medicines avoid counter chaos.

This application is absolutely free, with easiest UI and interface possible so that people with least knowledge of technology can also operate it.

### 

### 2.5. Stakeholders

* The admin, who can perform insertion deletion and checking operations that no one else is authorized to do.
* Developer who developed the project, for further releases and new versions (Both front end, and back end).
* Internal database engineer who manages the database working.

### 2.6. Operating Environment

**Hardware platform:** It needs pretty basic hardware requirements. Any average mobile will do the work.

The hardware we used to run this is mentioned below:

* Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)
* RAM – 4 GB or more
* Screen Resolution – 720 x 1500
* Hard disk – 3 GB or more
* Touch Screen
* 8gb ram (hardly uses 1 or 2gb(s) of ram for all the softwares we used) - Less than 500mb of disk space for project, and related data.

**Network Environment:** Should have a decent internet connection.

**Applications:**

❏ • IDE – Visual Studio Code

• Emulator – Android Studio

• Android Version – 11.0

• Dart – 2.18.4

### 2.7. System Constraints

* **Software constraints:**

This system doesn’t require any software constraints so far.

* **Hardware constraints:**

It needs pretty basic hardware requirements. Any average mobile will do the work.

The hardware we used to run this is mentioned below:

* Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)
* RAM – 4 GB or more
* Screen Resolution – 720 x 1500
* Hard disk – 3 GB or more
* Touch Screen
* **Cultural constraints:**

The person should know how to read, and understand basic English language.

* **Legal constraints:**

Not applicable.

* **Environmental constraints:**
* There are no environmental constraints. The system can be used anywhere anytime.
* The project is developed for people of all ages and ethnicities
* **User constraints:**

Age above 15 mature ones so that don’t waste doctors time and useless medicine buy.

**2.8. Assumptions & Dependencies Assumptions**:

We assume that people with some knowledge of hospital and will use this application.

# 3. External Interface Requirements

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

### 3.1. Hardware Interfaces

The system is perfectly supported by tablets and mobile phones as well.

The type of data we are taking from the user are login credentials, all the basic information e.g. Name, contact, gender, email etc. which is stored in database.

### It needs pretty basic hardware requirements. Any average mobile will do the work.

### The hardware we used to run this is mentioned below:

### - Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)

### - RAM – 4 GB or more

### - Screen Resolution – 720 x 1500

### - Hard disk – 3 GB or more

### - Touch Screen

### 3.2. Software Interfaces

**Operating System:** Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)

**Database:** SupaBase

**Ide and tools:** Visual Studio Code, Android Studio, Flutter

#### Communications Interfaces

* User email for signup should be valid.
* No encryption usage, or data transfer or synchronization issues.

# 4. Non-functional Requirements

### 4.1. Performance Requirements

1. Performance wise, our project is built to be very responsive and fast. The transitions between the interfaces take least amount of time.
2. Capacity wise, our system is very storage friendly, hardly requires some Mbs of data.
3. Safety wise, the data of users, interacting with our system stays safe and can only be accessed by the system owner.
4. The software is reliable in a sense that it fulfills all the needs that it is promised to fulfill. It was tested for any sort of bugs/ issues and was fixed by the developers eventually.

### 4.2. Safety Requirements

1. We took extra care that our system must not cause any damage on the machine on which the user is running our system.
2. The only thing that the user should take care of is the entry of dummy (fake) results in the database. Dummy data must be deleted from the database.

### 4.3. Security Requirements

1. External users such as someone out of the organization must not be given access to the system’s Admin panel. Login ensures this. Only admin of that app will have access
2. Only the Stakeholders should have access to the system.
3. The data of the user stays safe and untouchable. So, privacy is maintained.

## 4.4. Software Quality Attributes

No additional quality attributes.

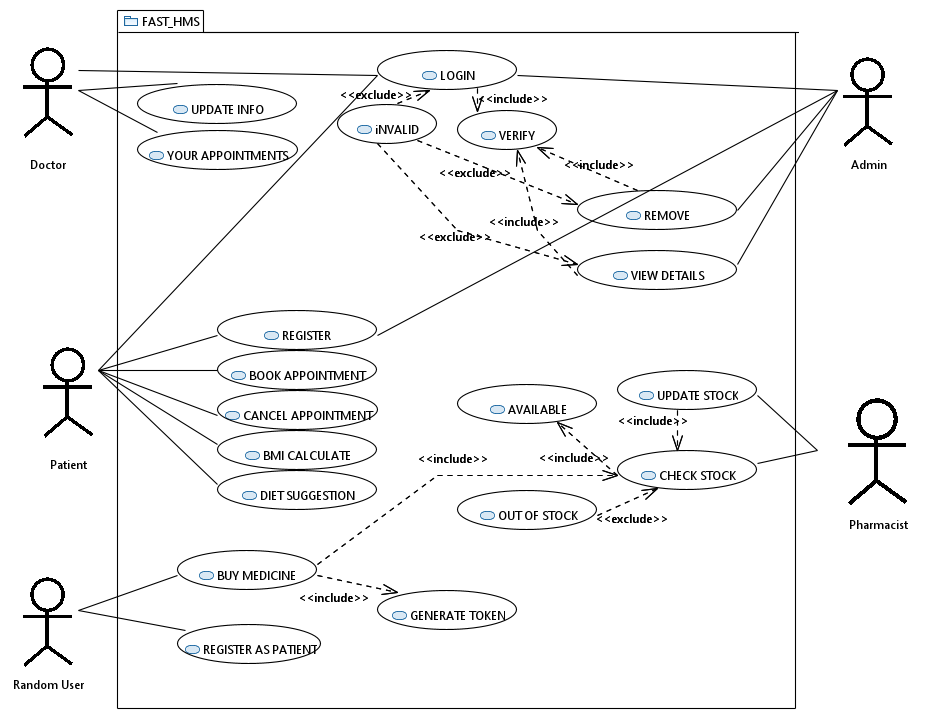
**Other Requirements:**

Not applicable.

# 5. Functional Requirements

##### 5.1. Use Cases

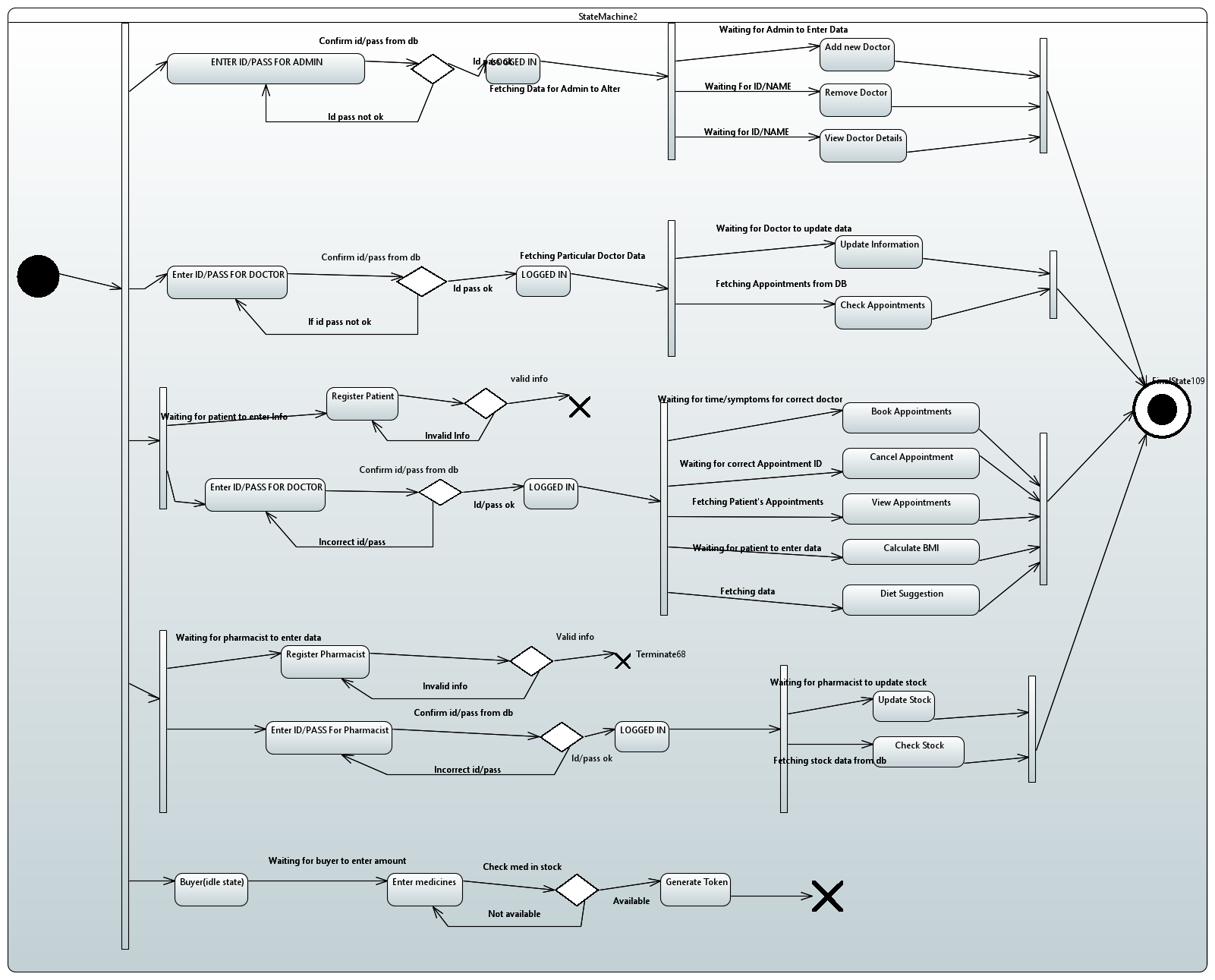
|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Hospital And Management System |
| **Use Case Description:** | |
| **Primary actor:** Admin | **Other actors:** Doctor, Patient, Pharmacist, Random User |
|  |  |
| **Pre-conditions:**  ▪ customer must have an adequate internet connection, and a mobile to run the app, and he should signup before moving ahead. | |
|  | |



#### 6. System Architecture

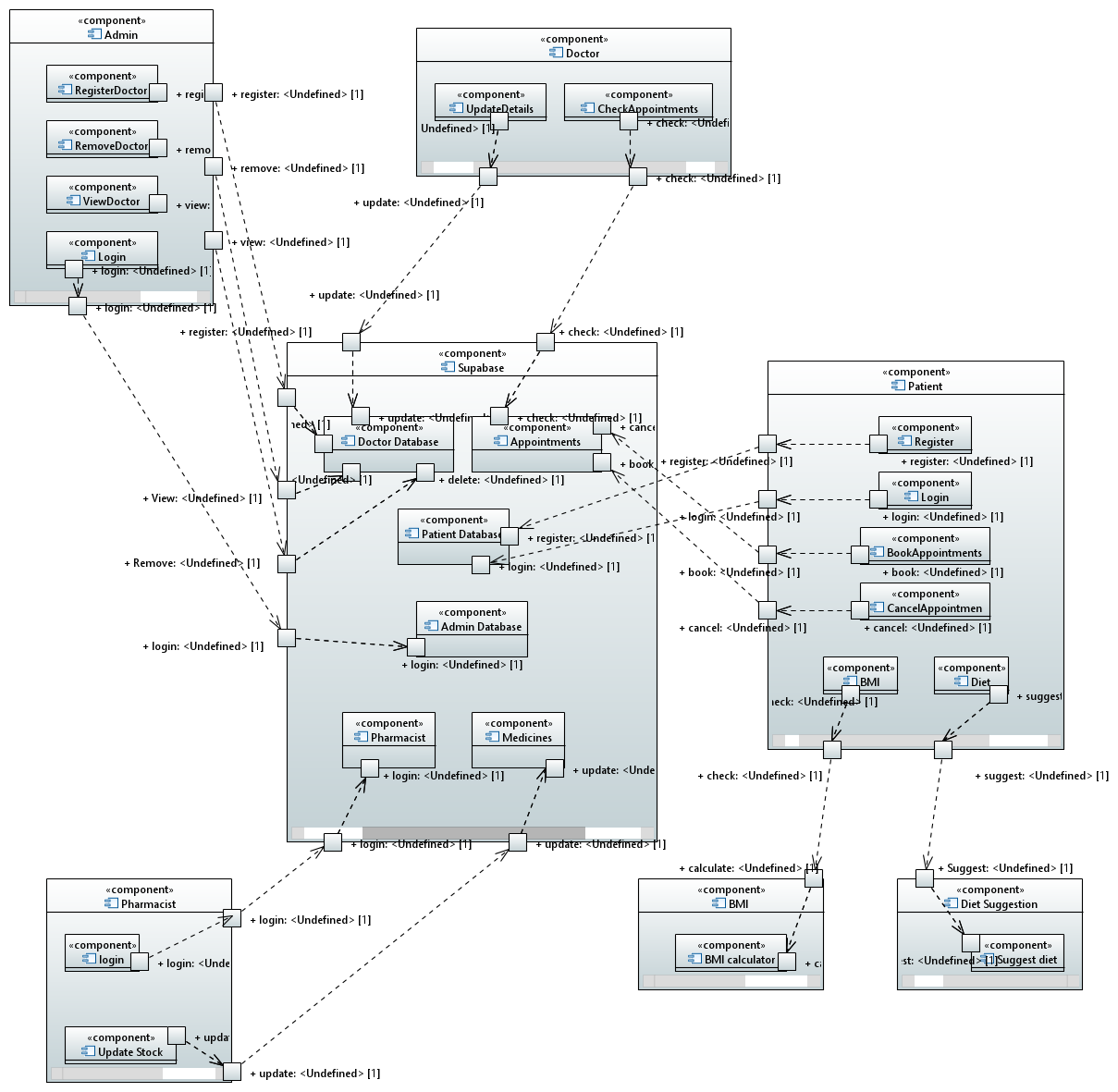
The architecture embodies the major static and dynamic aspects of a system. It is a view of the whole system highlighting the important characteristics and ignoring unnecessary details. In the context of our approach, architecture is primarily specified in terms of views of tier architecture which is a client-server architecture in which the presentation, the application processing and data management are logically separate processes.

**State Machine Diagrams:**

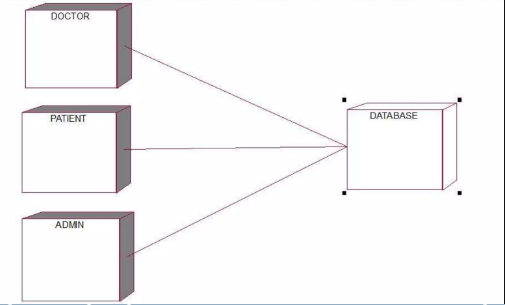


**6.1. System Level Architecture**

#### COMPONENT DIAGRAM



#### DEPLOYMENT DIAGRAM



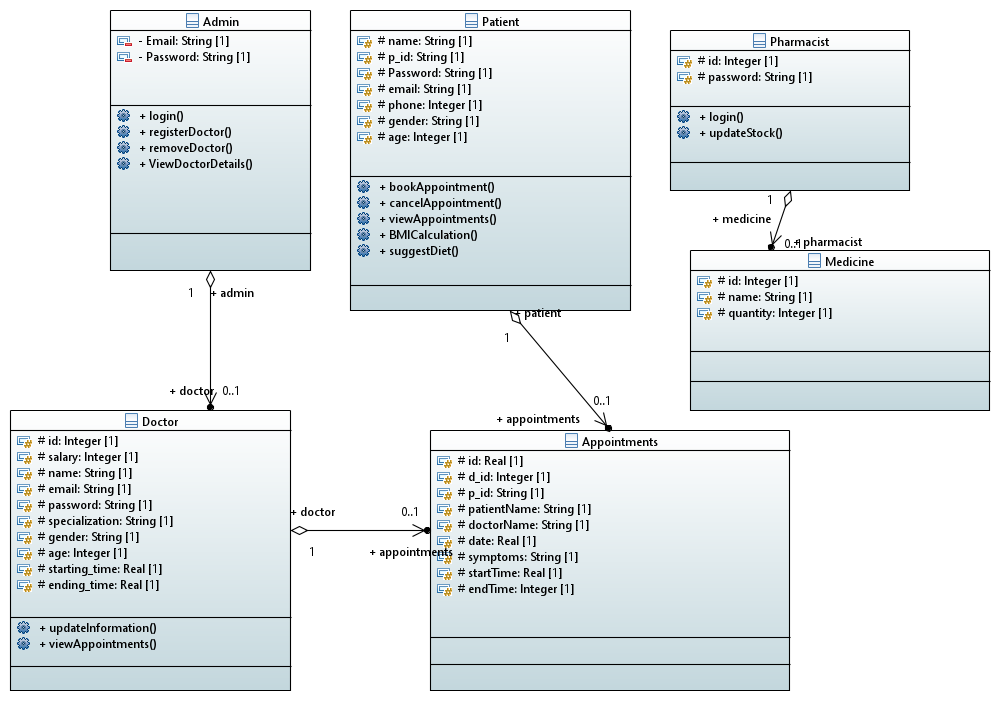
#### 7. Design Strategy

Future system extension or enhancement: we aim to add additional features to the app, such as adding chat bot using for assistance in case of emergencies if doctor not available, as well as checkout and payment and delivering medicines at home options

User interface paradigms: the program prompts the user to enter their information at multiple stages: while signing up, logging in, as well as booking appointment. Moreover, the UI also displays the user information at multiple pages, such as displaying the appointment details and user data and medicines available in pharmacy.

Multiple user can use app as synchronization and concurrency is there.

***8. System Design Class Diagram:***



**Functions:**

**Admin:**

* 1. Login - takes admin information, and verify the information from the database. Then give acces to functionalities.
  2. Register Doctor.
  3. Search Doctor details.
  4. Remove Doctor.

**Patient:**

1. Login - takes patient information, and verify the information from the database. Then give acces to functionalities.
2. Register new user as patient.
3. Book appointment
4. Cancel Appointment
5. Calculate BMI.
6. Get diet Suggestion in for certain symptoms and disease.

**Doctor:**

1. Login - takes admin information, and verify the information from the database. Then give access to functionalities.
2. Update his/her information from database.
3. See all hi appointment.

**Pharmacist:**

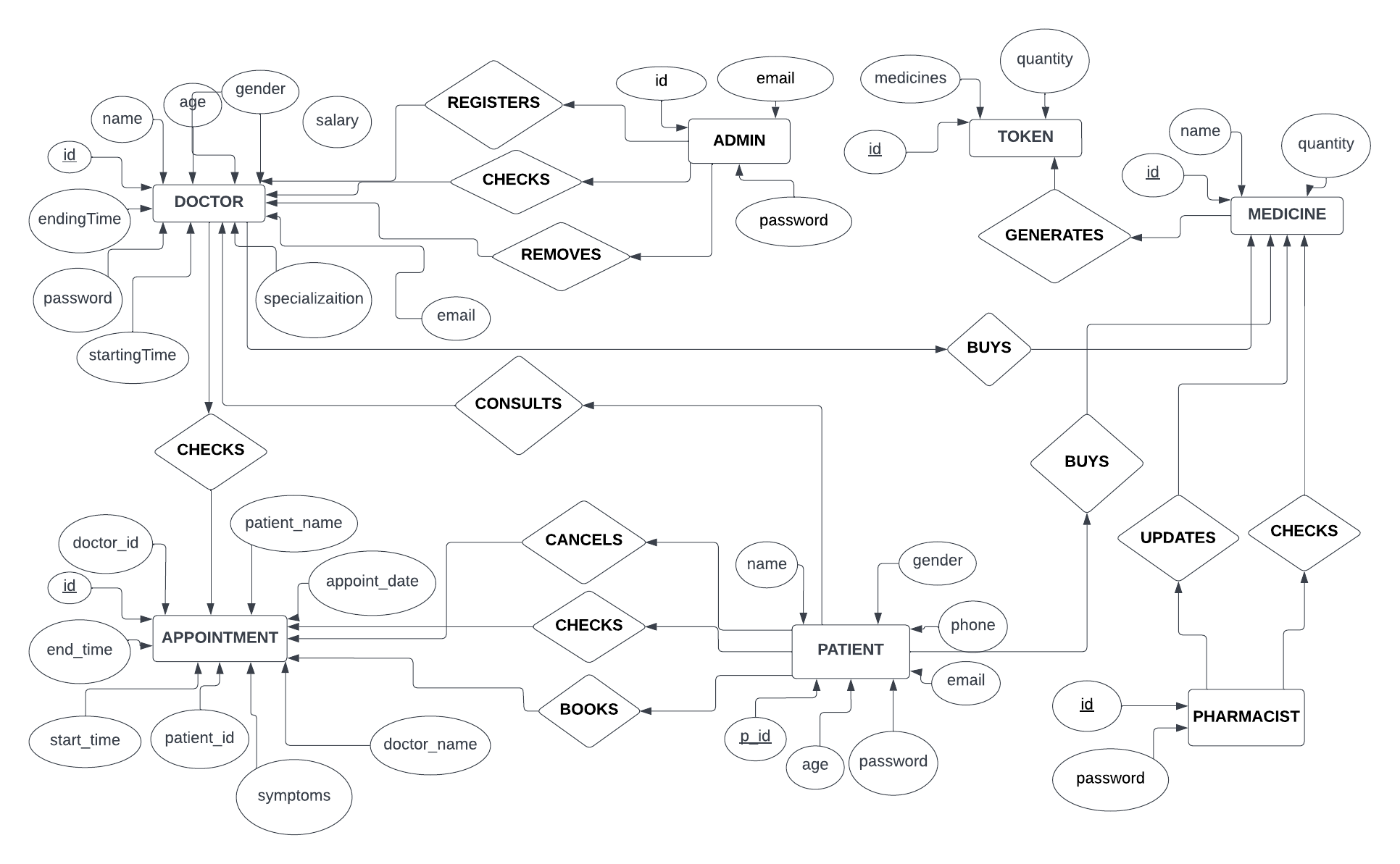
1. Login - takes admin information, and verify the information from the database. Then give access to functionalities.
2. Update stock.

**Random User (Not registered/Registered):**

1. Buy Medicine and get it from pharmacy with the generated token given to them through app.

## 8.1. Database Design

**8.1.1. ER Diagram**

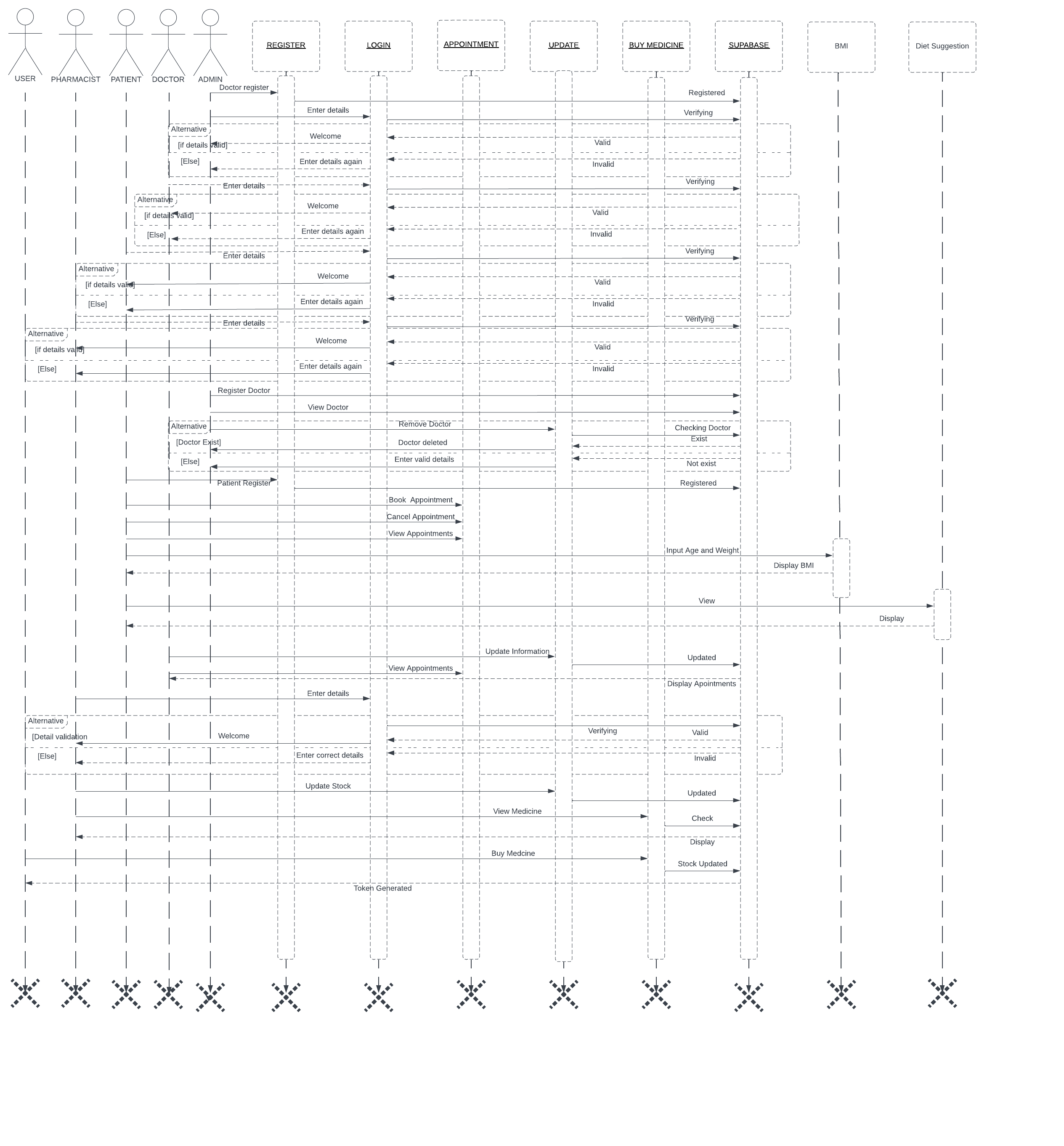


**8.1.2. Data Dictionary**

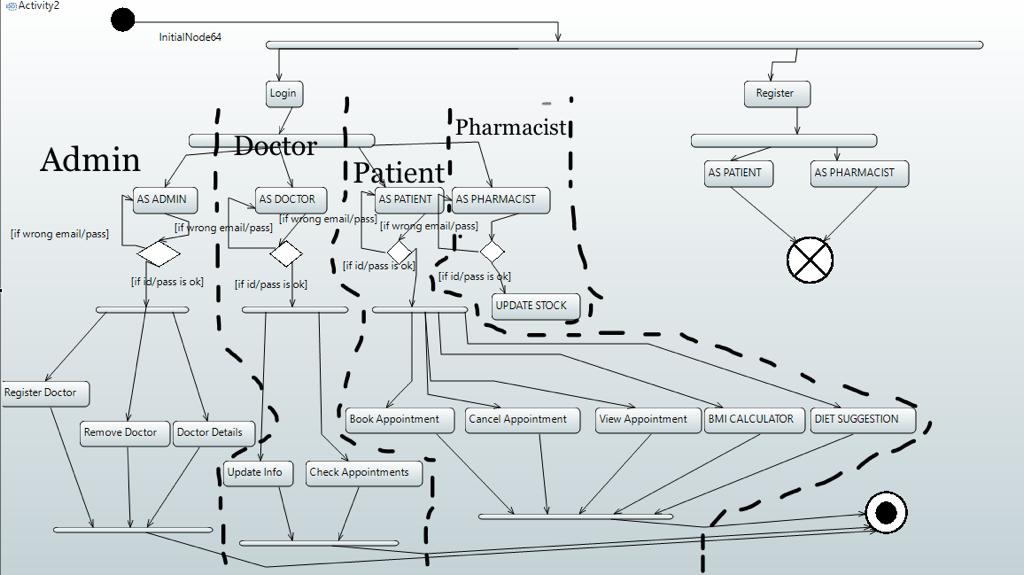
Not Applicable

# 9. Application Design

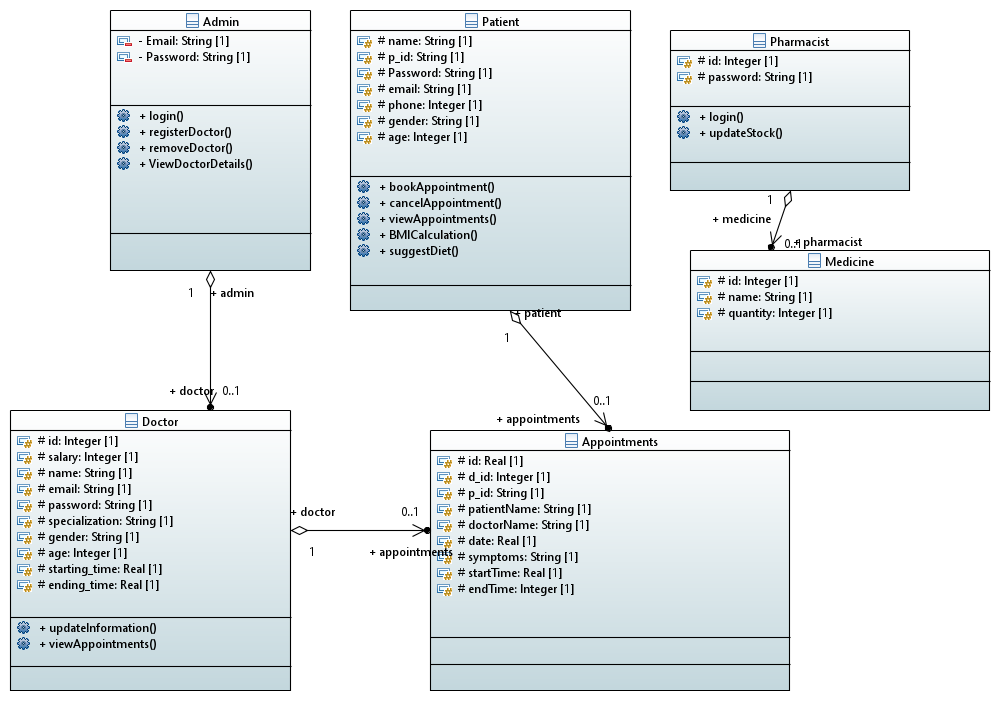
**9.1.2. Sequence Diagram**



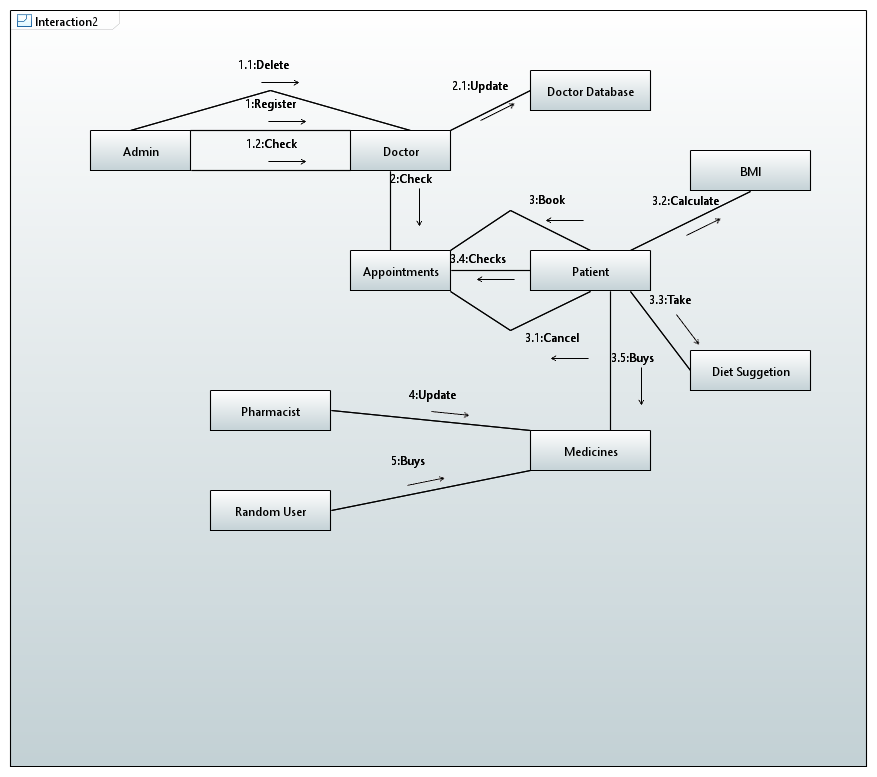
**9.1.3. Activity Diagram**



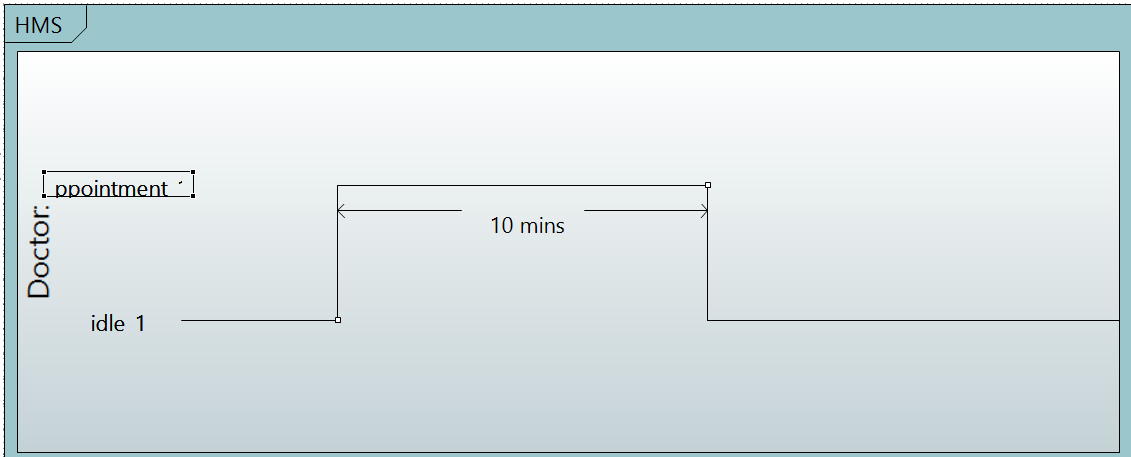
**Entity Class Diagram:**



**Collaboration Diagram:**



**Timing Diagram:**



#### 10. References

<https://supabase.com/docs/reference/dart/upgrade-guide>

<https://dart.dev/>

<https://mega.nz/folder/3WhSkBjJ#YSDBnDegckd9-xSQ04W0qA/folder/HXxyyDiQ>

<https://docs.flutter.dev/>