Software Requirements Specification

for

Doctor Appointment System

Version 1.0 approved

Prepared by

Somiha Tasnim (170128)

Tanvir Istiak Anik (170119)

Jashore University of Science and Technology

Date: 01.10.2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Somiha Tasnim	30/09/2019		1.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of a Doctor Appointment system application for the android. It explains the purpose and features of the application, the interface through which a user can access a personal account and the constraints that must be satisfied for security purposes. This document is intended for both the customer and the project development.

1.2 Document Conventions

This document will use IEEE format. The document uses the following conventions.

Short Form	Abbreviation
DB	Database
CRC	Class Relationship Collaboration
DFD	Data Flow Diagram
ER	Entity Relationship

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the Doctor Appointment System and it is restricted within the university premises. This has been implemented under the guidance of MD Shafiuzzaman (lecturer of Jashore University of Science of Technology). This project is useful for doctors and patients in hospitals and clinics.

1.4 Product Scope

The system has been facing problems due to its paper-based appointment system. With the increase in the number of patients visiting, it has become difficult to manage the appointment system manually. Recording of appointments and creating registers by pen and paper has become a tedious task. And also its difficult to manage huge number of patient database.

So, the purpose of the doctor appointment system is to ease appointment system and to create a convenient and easy-to-use application for doctors and patients in hospitals and clinics. It is very easy to make an appointment. Doctor and Patient also watch their appointment time. Moreover, doctor and patient can also edit or cancel appointment.

1.5 References

https://nevonprojects.com/doctor-appointment-booking-system/

https://www.slideshare.net/MdShafiuzzamanHira/srs-of-library-circulation-system

2. Overall Description

2.1 Domain Analysis

- **Technical Research:** From past experience we saw that in our country doctor appointment system being performed in a manual way. In manual system when people is sick they go to hospital/clinic or make a call for making appointment. Then the staff of hospital or clinic give a serial number to the patient according to patient serial list. After that patient goes to hospital or clinic and waits for doctor's appointment according to the serial number. Patient doesn't know the exact time of doctor appointment. If the doctor cancels his appointment for any emergency reason, the patient doesn't know about the cancellation of the appointment until he or she visits the hospital or clinic. Sometime, the record of patient is lost. So, many kinds of problem arise in manual doctor appointment system.

- **Existing Application:** Such kind of application are available in google play store. They are Doctor Appointment, Ibn Sina Doctor Appointment, Doctorola, Patient Medical Records & Appointment etc. The features of these applications are:
    1. Patient, Doctor, Admin authentication.
    2. View doctors details.
    3. Make appointment.
    4. Patient medical record.
    5. Pay off fees.
    6. Find blood information.
    7. Find hospital location.

  In our proposed system we want to make such an application which is used for getting information about doctor from doctor list and making appointment. The features of our application are:
    1. Patient, Doctor, Admin authentication.
    2. View doctors details.
    3. Make appointment.

- **Potential Customer Identify:** The customer of this application is doctor and patient. They can use it easily. Patient can book appointment easily and doctor can also see the appointment schedule easily.

- **Future Customer Identify:** In future any clinic and hospital can use this application in large scale. So, hospitals and clinics may be the future customer of this application.

2.2 Product Features and Prioritization

1. **Normal Requirement:** Normal requirement consists of objectives, goals, minimal function and performance that are stated during the meeting with customer. The normal requirement of our project are:

    - Doctor, patient, admin authentication.
    - Allow admin to check patient information.
    - Allow admin to check doctor information.
    - Allow admin to update appointment schedule.
    - Allow admin add, edit and delete appointment.
    - Allow patient to make an appointment.
    - Allow patient to request for editing an appointment schedule.
    - Allow patient to request for canceling an appointment schedule.
    - Allow doctor to request for editing an appointment schedule.
    - Allow doctor to request for canceling an appointment schedule.

2. **Expected Requirement:** Expected requirement consists of implicit requirements and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause of dissatisfaction. The expected requirement of our project are:
    - Accessible via internet.
    - Doctor, patient, admin register.
    - Doctor, patient, admin log in.
    - Doctor, patient, admin log out.
    - Maintain a database of doctor appointment system.
    - Change user password.
    - Configure the particular doctor appointment schedule.
    - Search available doctor.
    - Make and cancel appointment option.
    - Provide appropriate error message.

3. **Exciting Requirement:** These requirements are for features that go beyond the customer's expectation and prove to be very satisfying when present. The exciting requirement of our project are:
    - Offer log in with mobile phone.
    - Offer to find the hospital or clinic location.
    - Offer to get notification for confirmation message.

2.3 Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. The possible internal stakeholders of doctor appointment application are:

- **Hospitals/ Clinics:** Hospitals or Clinics can be the final authority over our budget. The can be our project sponsor. They can take any decision for the project.
- **Doctor:** Doctor can use this application easily. They will a big part of our system.
- **System Operator:** He will be able to interact with the software directly.
- **Developers:** We can take developers as stakeholder because the system will be updated day by day and if occurs any system interruption, they will find the problem and try to solve it.

The possible external stakeholders of doctor appointment application:

- **Patient:** Mainly the software will be made for the patient. They can easily book their appointment using this application.

2.4 User Classes and Characteristics

User of the system should be able to know appointment information of a hospital or clinic with the given date/time of appointment from the database. The system will support three types of user privileges, patient, doctor and admin. Patients will have access to make a new appointment, watch doctor details. Doctors will have access to watch appointment details. Admin will have access to manage the whole system.

The patient should be able to do the following functions:

- Make a new appointment.
- Cancel or Edit appointment.
- Watch doctor details.

The doctor should be able to do the following functions:

- Cancel or Edit appointment.
- Watch appointment details.

The admin should be able to do the following functions:

- Approve a new appointment.
- Cancel or Edit appointment.
- Update appointment.

2.5 User Story

As a patient user authenticate in our system is creating an account. If he or she already has an account then he or she will log in the system with his or her username and password. He or she search the doctor list and he or she can see details about the doctor and if doctor is available in that schedule patient user can make that appointment. If the patient user cannot find his or her available doctor, then he or she can exit from the application.

As a doctor user authenticate in our system is creating an account. If he or she already has an account then he or she will log in the system with his/her username and password. He or she can see all of his or her appointment schedule.

As a admin user he can log in his/her account and see all the information of doctor, patient and appointment schedule. Admin also can update the appointment.

2.6 Operating Environment

Operating environment for Doctor Appointment System is as listed below:

- Operating System: Android
- Database: Firebase
- Platform: Android

2.7 Design and Implementation Constraints

The Doctor Appointment system shall be an android based application system. The system shall be developed using Android and Firebase server.

A person who has no knowledge of computers will find it difficult to understand the system. But with a little knowledge it will be very easy to handle the project. This will be a mobile based application. So, it is not easy to implement a web based application.

2.8 Assumptions and Dependencies

- Each user must have a valid user id and password.
- Users must log in to the system to access any record.
- Only the Administrator can delete records.

2.9 User Documentation

- Help pages will be providing document with screen shots.
- If the user has more queries regarding this website then he/she can contact with the administrator through contact us page.

3. External Interface Requirements

3.1 User Interfaces

The user interface is designed in android. The developer will have to study the designing of the product. The use of the controls and the component from the Add items feature of the android. The user of the product will get very user friendly environment which will be very easy to work with.

3.2 Hardware Interfaces

- Android Phone

3.3 Software Interfaces

Following are the software used for the Doctor Appointment System.

Software used	**Description**
Operating system	We have chosen Android operating system for its best support and user-friendliness.
Database	To save the appointment records, doctor records we have chosen Firebase database.
Android Studio	To implement the project we have chosen java language for its more interactive support.

3.4 Communications Interfaces

This project will support in all android phones. The communication is done through internet.

4. System Features

4.1. Patient :

1. Authentication
2. Search and select Doctor.
3. Request for making Appointment.
4. Request for editing Appointment.
5. Request for canceling Appointment.

4.2. Doctor :

1. Authentication
2. Search Doctor.
3. Request for editing Appointment.

4. Request for canceling Appointment.
5. Watch Appointment schedule.

4.3. Admin :

1. Authentication
2. Search Doctor.
3. Confirm Appointment.
4. Editing Appointment.
5. Deleting Appointment.
6. Watch Appointment schedule.

Use Case Description: In this section use case scenarios of the above features are described elaborately.

Use Case Name: Sign up

Primary Actor: Patient, Doctor, Admin.

Goal in Context: To register in the system.

Precondition:

- Appointment system has been designed to add admin, doctor and patient.
- It has an interfaces for registration.

Triggers: The patient, doctor and admin have to register.

Scenario:

- Visit the register page.
- Input required information
- Check availability for username and check validity of password.
- Email send to user e mail address.
- User confirm from his/her e-mail address.
- Confirmation message are shown.

Exception:

- Invalid Input.
- Not valid for Registration
- Authentication failed.

Priority: Must be implemented.

When Available: At the first time of registration.

Use Case Name: Sign In

Primary Actor: Patient, Doctor, Admin.

Goal in context: When user want to enter the system.

Precondition: Must be registered.

Triggers: Need to log in the system.

Scenario:

- Visit the login page.
- Input Username and Password.
- Proceed the next activity.

Exception:

- Username/Email address invalid.
- Wrong Password.

Priority: Must be implemented.

When Available: First implemented and after log out for the system.

Use Case Name: Sign out

Primary Actor: Patient, Doctor, Admin.

Goal in Context: To exit from the system.

Precondition: Must be logged in.

Triggers: The patient, doctor and admin have to logout.

Scenario: Tap the log out button.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Update account

Primary Actor: Patient, Doctor, Admin.

Goal in Context: To update the system.

Precondition: The system must be programmed for password and other information

Triggers: The patient, doctor and admin have to update information and change the existing password to a new one.

Scenario:

- Visit the log in page and log in.
- Tap Edit button.
- Edit required information.
- Change password.

Exception:

- Weak password or password length is too short
- Invalid mobile number.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Confirm appointment

Primary Actor: Admin.

Goal in Context: To confirm new appointment.

Precondition:

- The system must be programmed for confirming new appointment.
- Admin must be logged in.

Triggers: The admin has to confirm new appointment.

Scenario:

- Visit the log in page and log in.
- Tap Maintain Appointment button.
- Tap Confirm button to confirm new appointment.
- Enter the new item data and confirm changes.
- Proceed to the next activity.

Exception: Already exist: Requested appointment already exist in the database.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Edit appointment

Primary Actor: Admin.

Goal in Context: To edit new appointment.

Precondition:

- The system must be programmed for editing new appointment.
- Admin must be logged in.

Triggers: The admin has to edit new appointment.

Scenario:

- Visit the log in page and log in.
- Tap Maintain Appointment button.
- Tap Edit button to add new appointment.
- Enter the new item data and confirm changes.
- Proceed to the next activity.

Exception: Does not exist: Requested appointment does not exist in the database.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Delete appointment

Primary Actor: Admin.

Goal in Context: To delete new appointment.

Precondition:

- The system must be programmed for deleting new appointment.
- Admin must be logged in.

Triggers: The admin has to delete new appointment.

Scenario:

- Visit the log in page and log in.
- Tap Maintain Appointment button.
- Tap Delete button to add new appointment.
- Enter the new item data and confirm changes.
- Proceed to the next activity.

Exception: Does not exist: Requested appointment does not exist in the database.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Search doctors and view doctor details.

Primary Actor: Patient, Doctor, Admin.

Goal in Context: To search doctor and view doctor details.

Precondition:

- The system must have database for available doctor.
- Patient, admin must be logged in.

Triggers: The admin and patient has to search for available doctor and view details about the doctor.

Scenario:

- Visit the log in page and log in.
- Tap Search Doctor button.
- Tap Select Doctor button.
- Proceed to the next activity.

Exception: Does not exist: Searching doctor which does not exist in the database.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: View patient details**.**

Primary Actor: Doctor, Admin.

Goal in Context: To view patient details.

Precondition:

- The system must have database for patient record.
- Doctor, admin must be logged in.

Triggers: The admin and doctor has to view details about the patient.

Scenario:

- Visit the log in page and log in.
- Tap View Patient Details button.
- See patient details.
- Proceed to the next activity.

Exception: Does not exist: Patient record which does not exist in the database.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: View all appointment schedule

Primary Actor: Doctors, Admin.

Goal in Context: To view all appointment schedule

Precondition:

- The system must have database for all appointment schedule record.
- Doctor, admin must be logged in.

Triggers: The admin and doctor has to view all appointment schedule.

Scenario:

- Visit the log in page and log in.
- Tap Maintain Appointment button.
- See all appointment schedule.
- Proceed to the next activity.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Request for making appointment

Primary Actor: Patient.

Goal in Context: To request for making appointment.

Precondition:

- The system must have programmed for making appointment option.
- Patient must be logged in.

Triggers: The patient has to book appointment.

Scenario:

- Visit the log in page and log in.
- Tap Making Appointment button.
- Proceed to the next activity.

Priority: Must be implemented.

When Available: First increment.

Use Case Name: Request for editing appointment

Primary Actor: Patient, Doctor.

Goal in Context: To request for editing appointment.

Precondition:

- The system must have programmed for editing option.
- Patient, Doctor must be logged in.

Triggers: The patient, doctor has to request for edit appointment.

Scenario:

- Visit the log in page and log in.
- Tap Edit Appointment button.
- Proceed to the next activity.

Priority: Must be implemented.

Exception: Does not exist: Requested editing appointment does not exist in the appointment.

When Available: Second increment.

Use Case Name: Request for canceling appointment

Primary Actor: Patient, Doctor.

Goal in Context: To request for canceling appointment.

Precondition:

- The system must have programmed for canceling option.
- Patient, Doctor must be logged in.

Triggers: The patient, doctor has to request for cancel appointment.

Scenario:

- Visit the log in page and log in.
- Tap Cancel Appointment button.

- Proceed to the next activity.

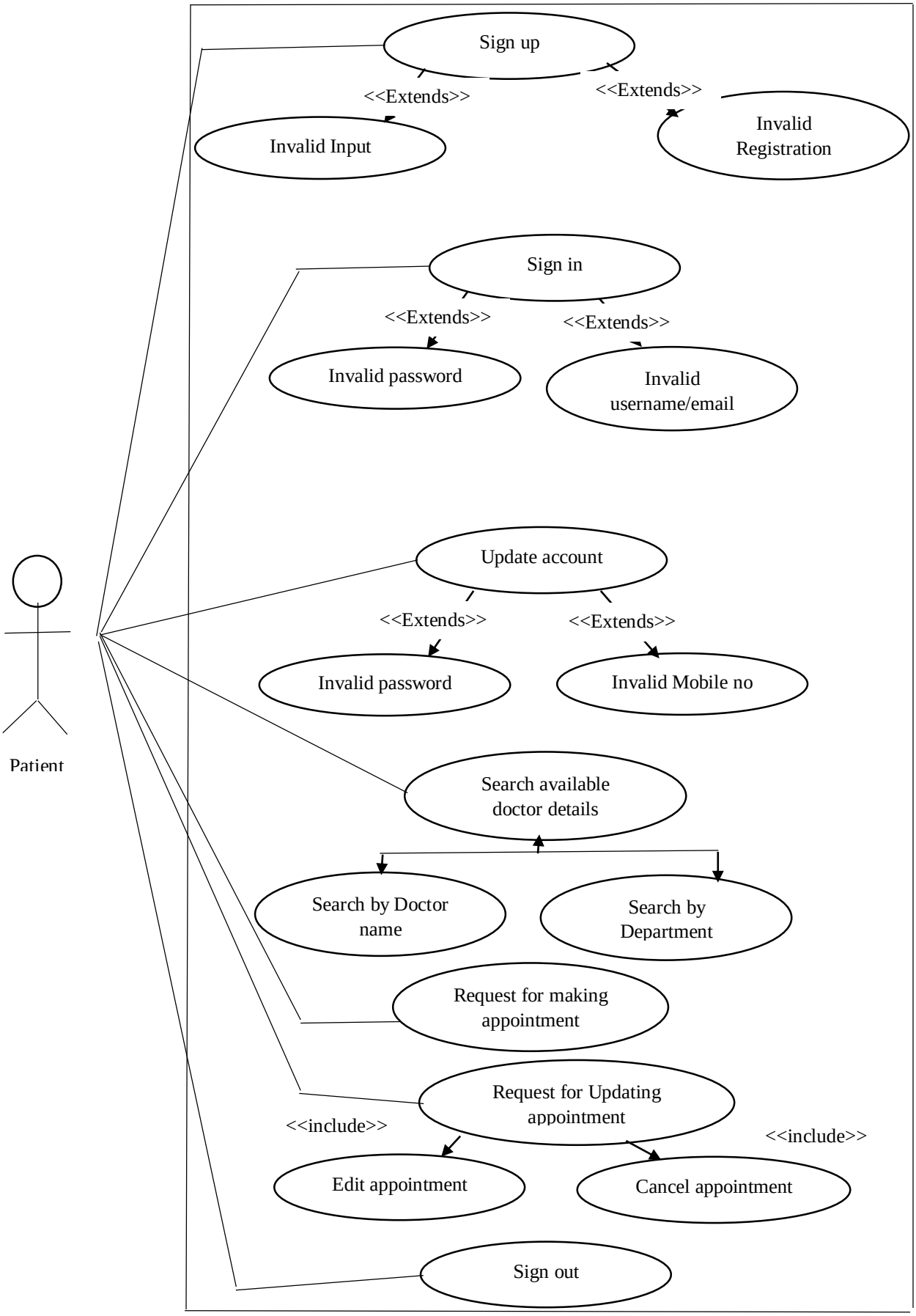
Priority: Must be implemented.

Exception: Does not exist: Requested canceling appointment does not exist in the appointment.

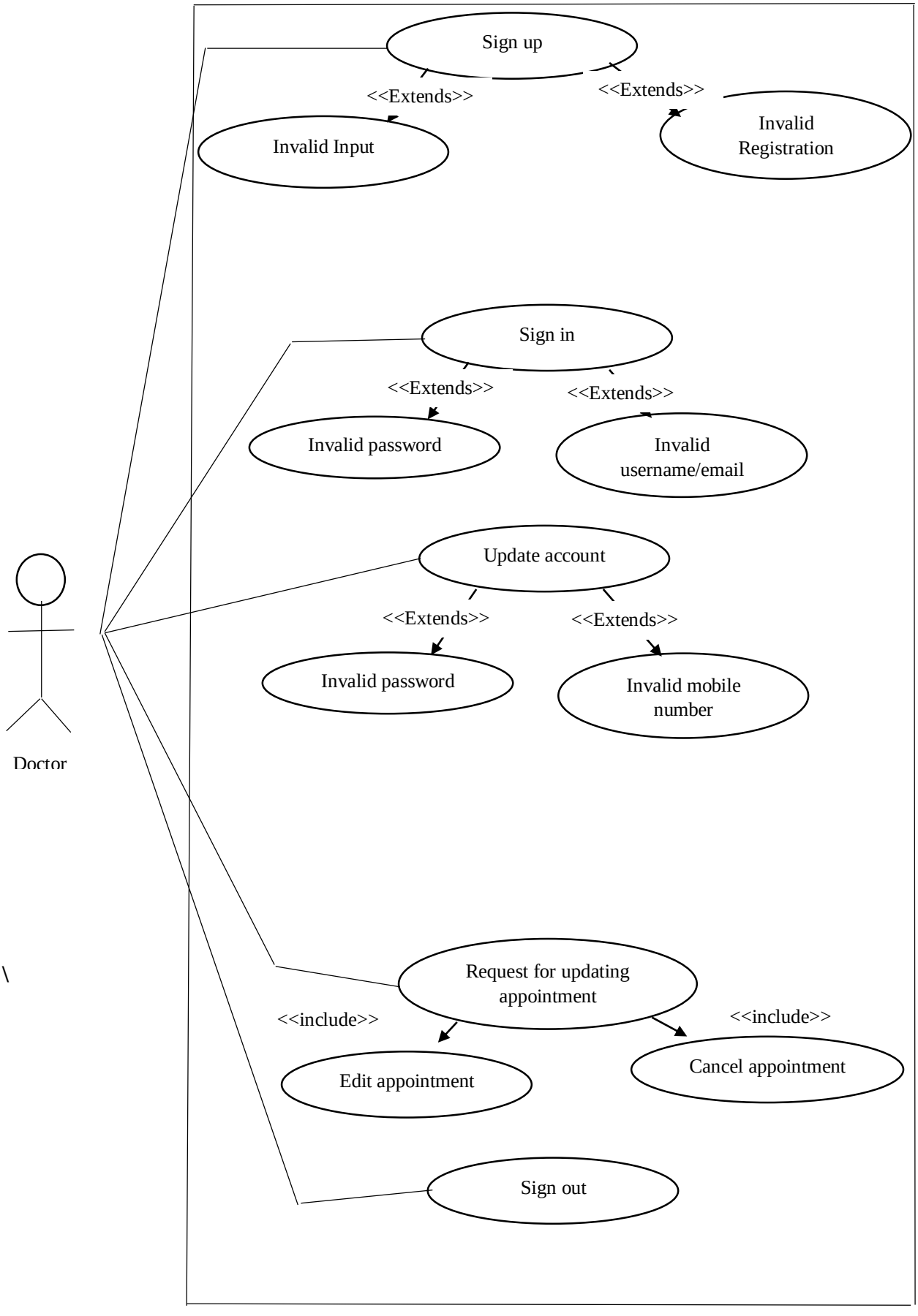
When Available: Second increment.

Use Case Diagram

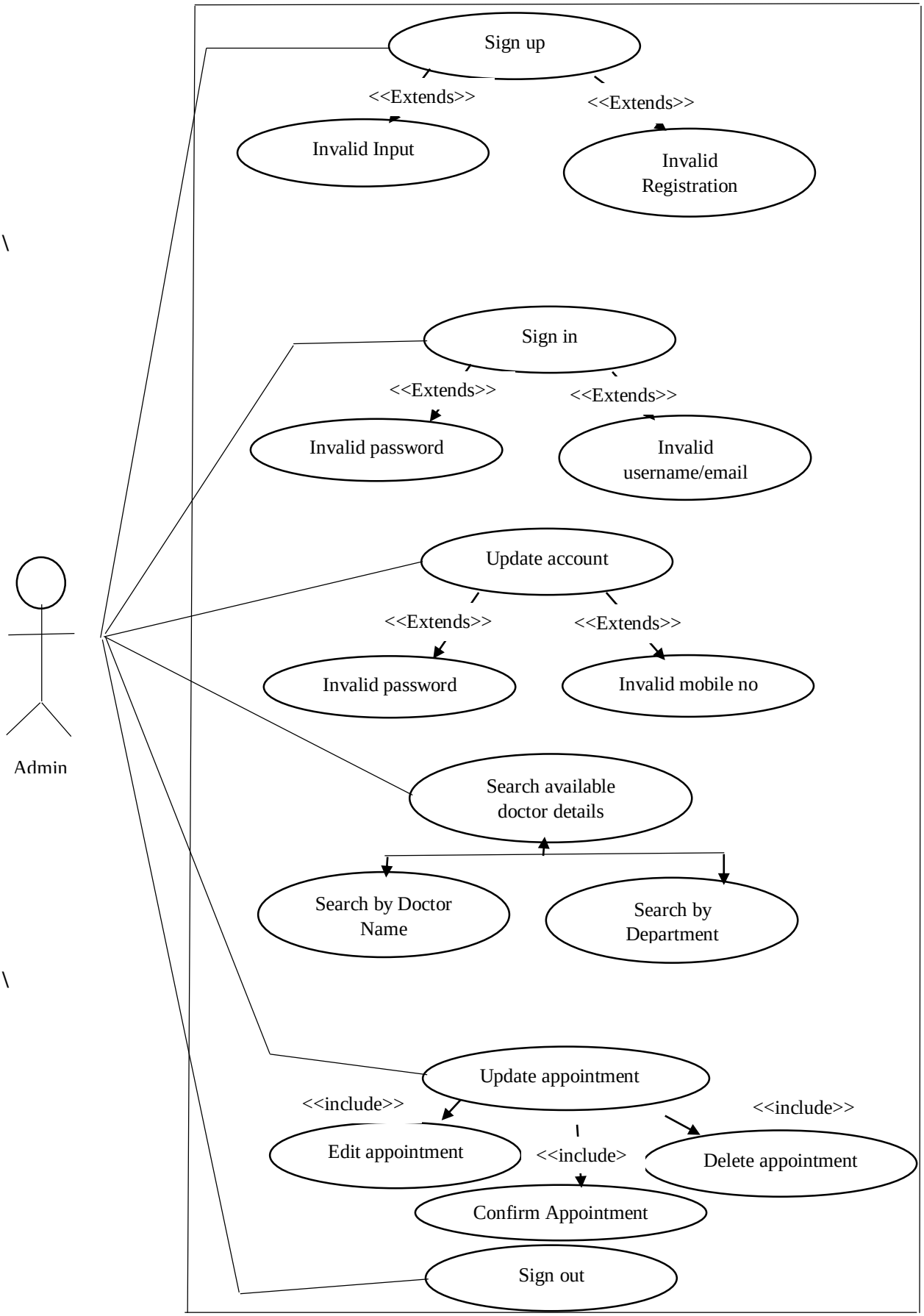
Use Case Diagram for Patient:



Use Case Diagram for Doctor:

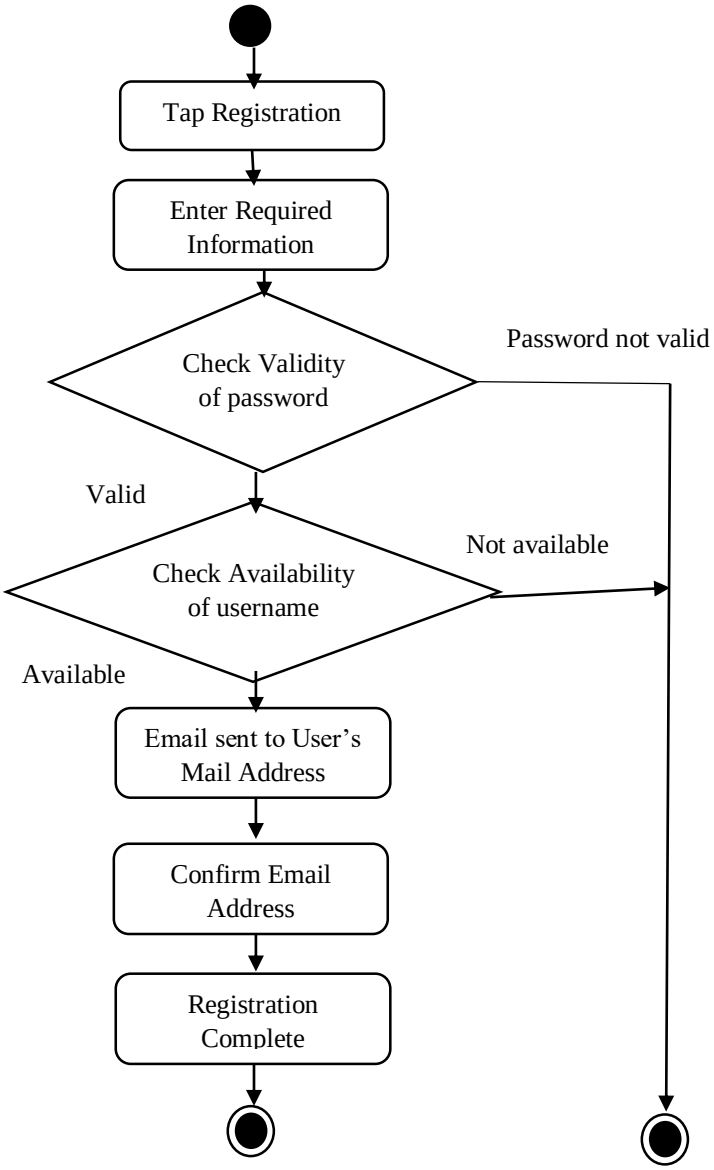


Use Case Diagram for Admin:



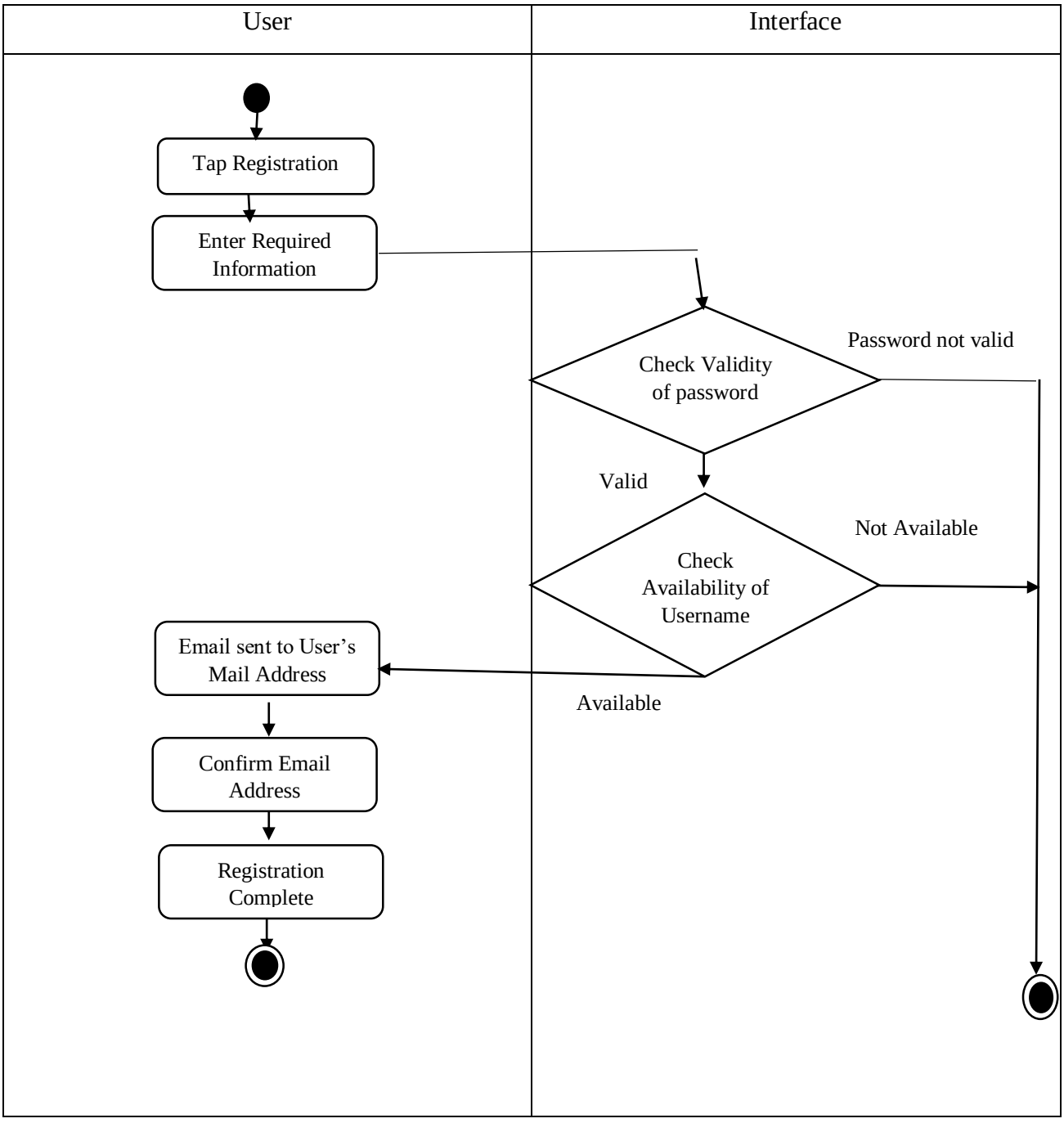
Activity Diagram and Swimlane Diagram:

Sign up:



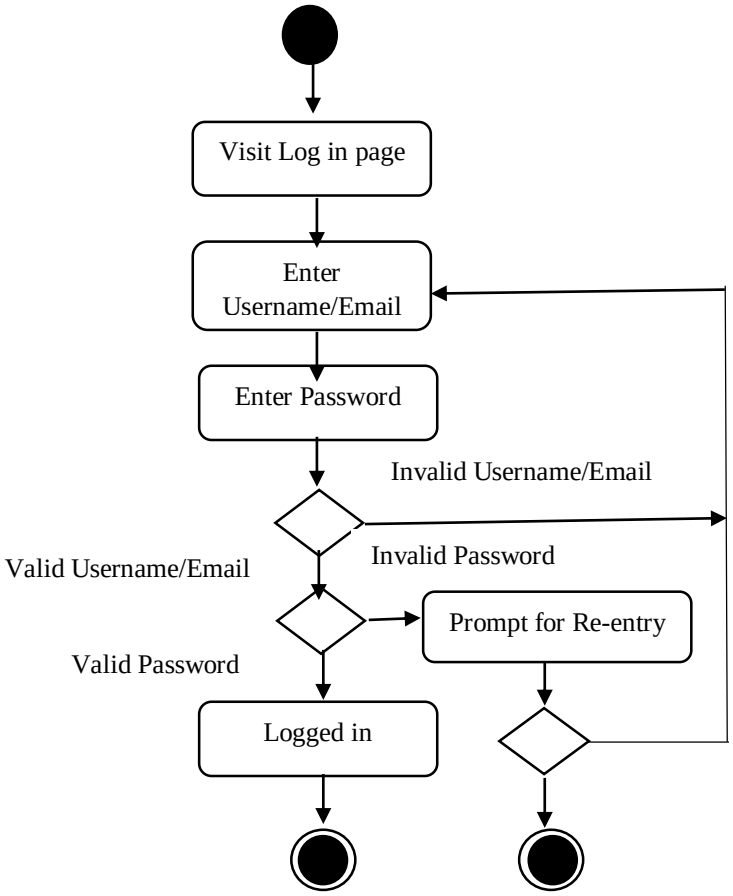
Swimlane Diagram:

Sign Up:



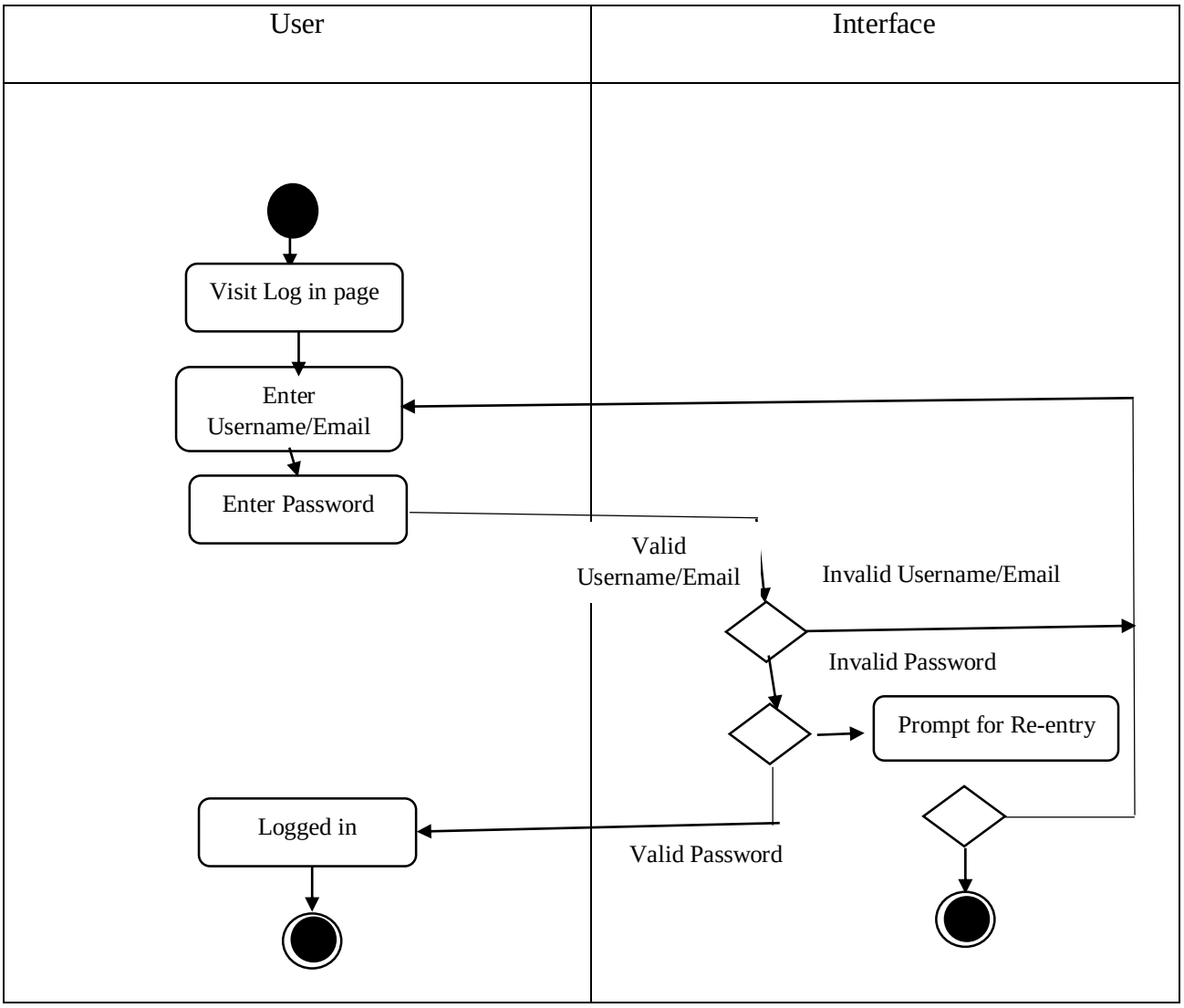
Activity diagram:

Sign In:



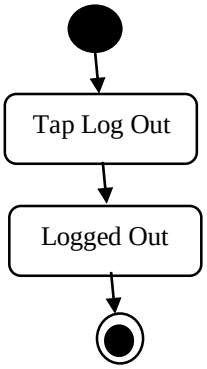
Swimlane Diagram:

Sign In:



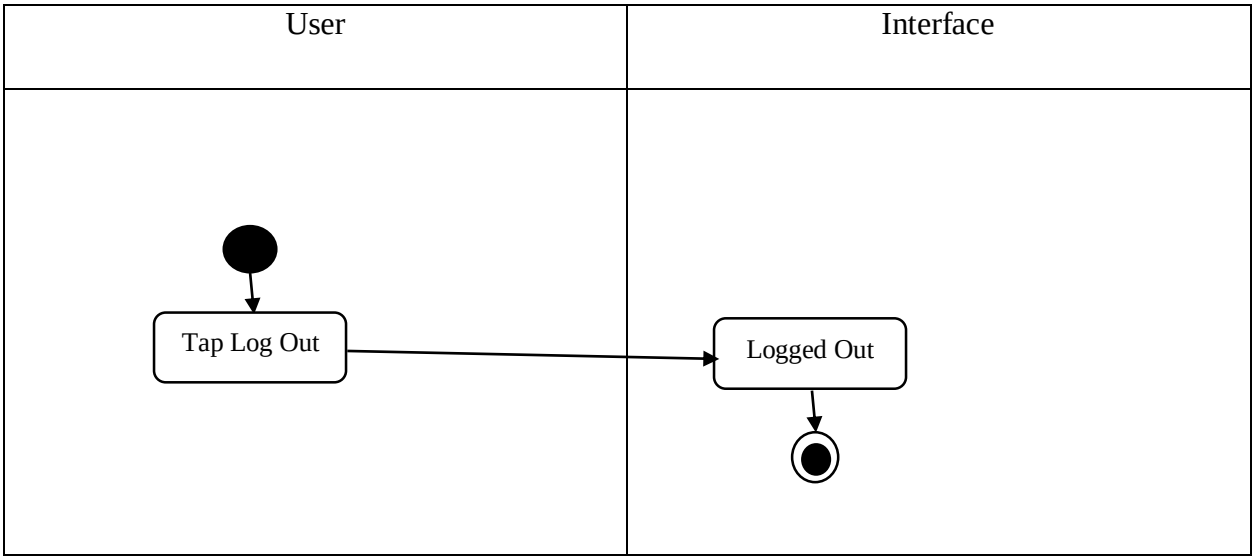
Activity Diagram:

Sign Out:



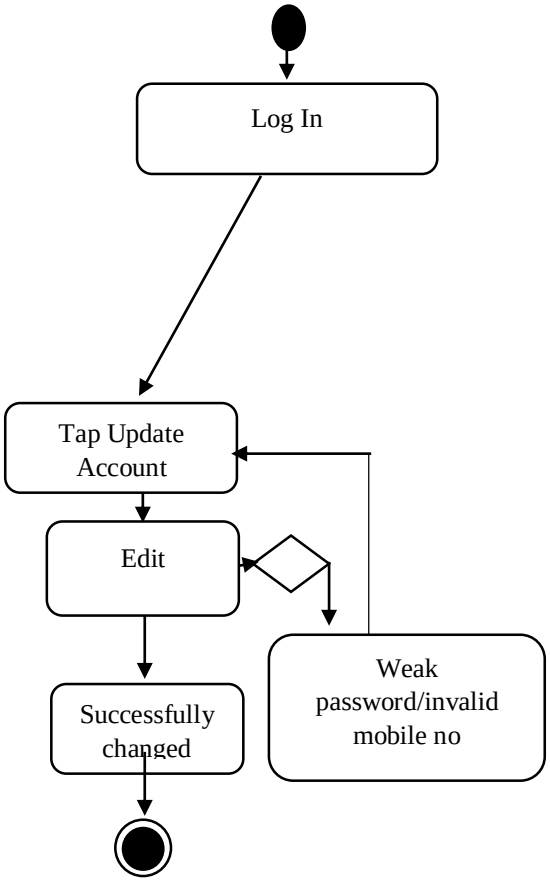
Swimlane Diagram:

Sign Out:



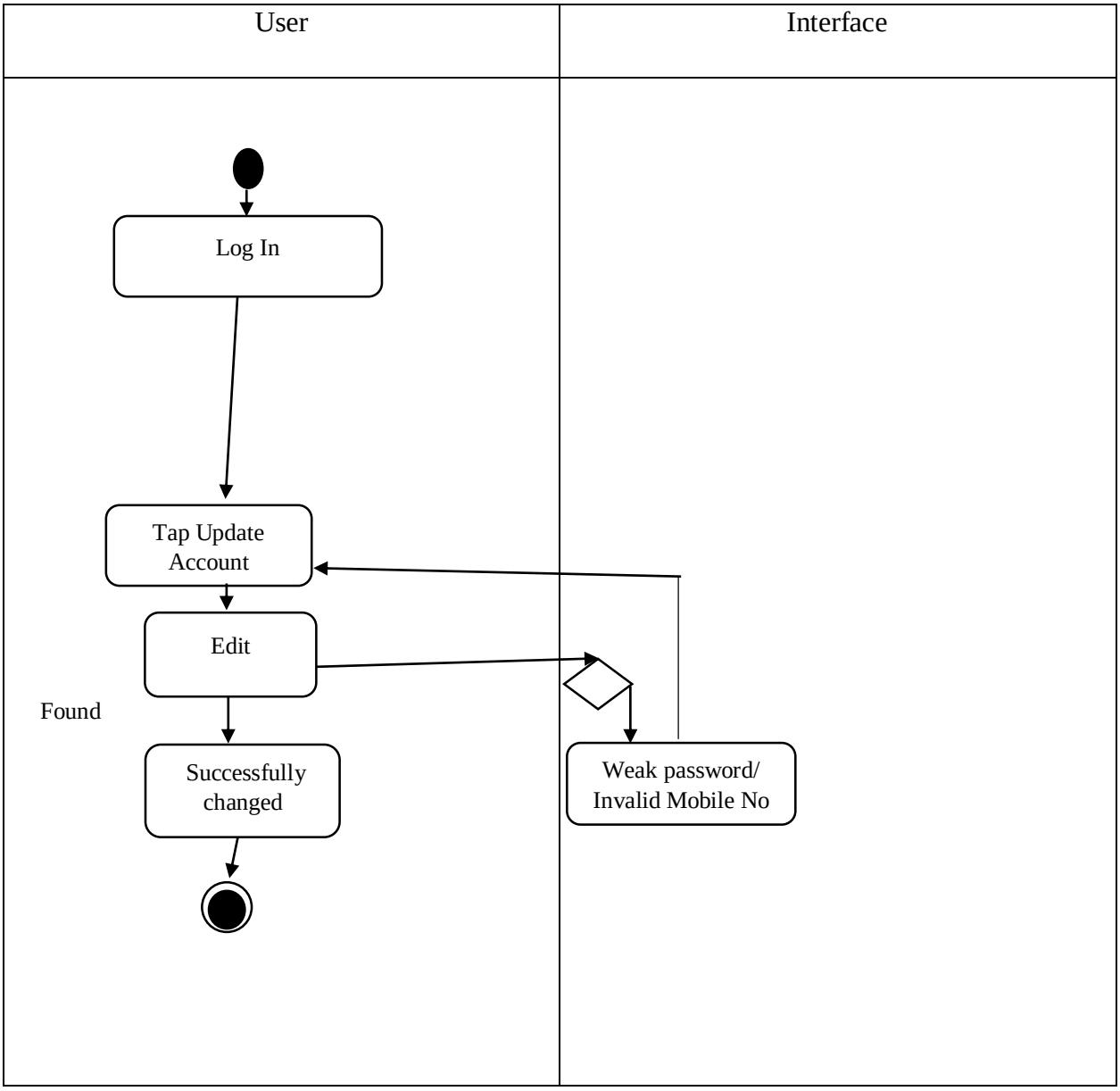
Activity Diagram:

Update Account:



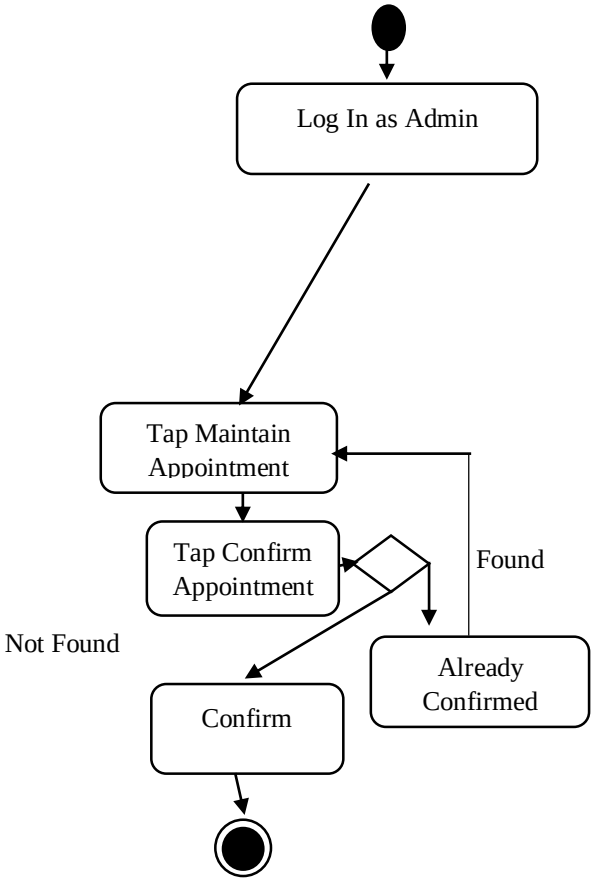
Swimlane Diagram:

Update Account:



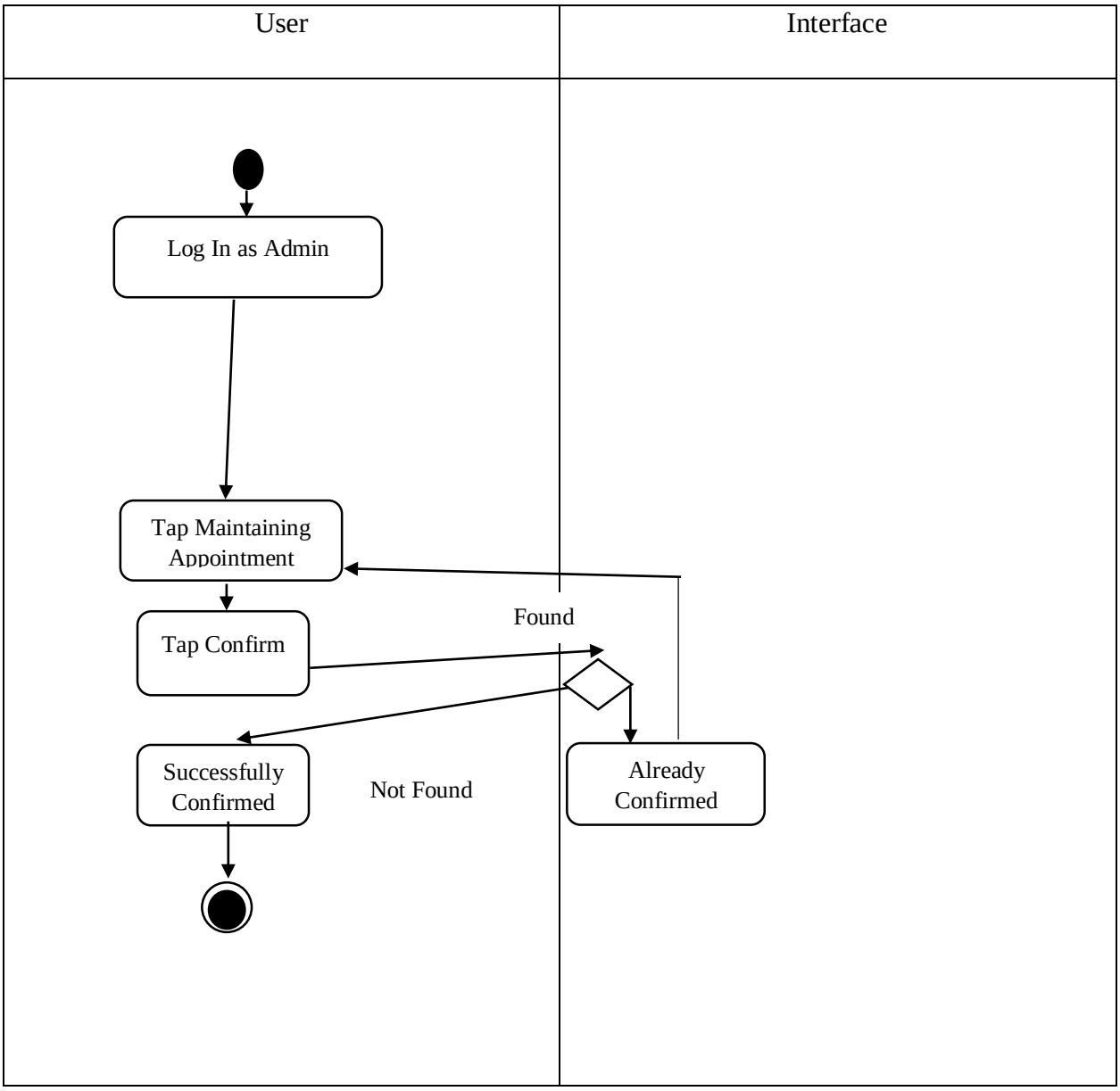
Activity Diagram:

Confirm Appointment:



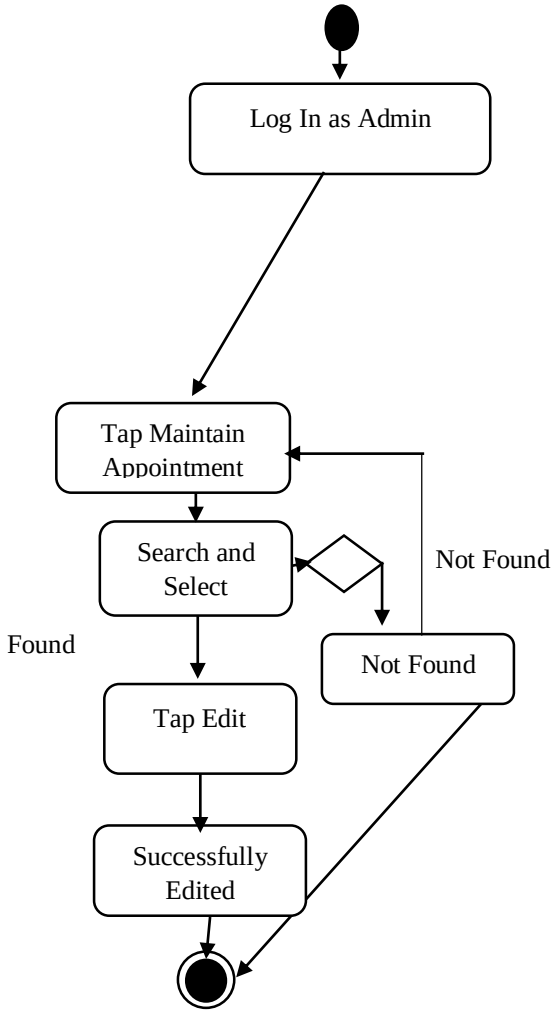
Swimlane Diagram:

Confirm Appointment:



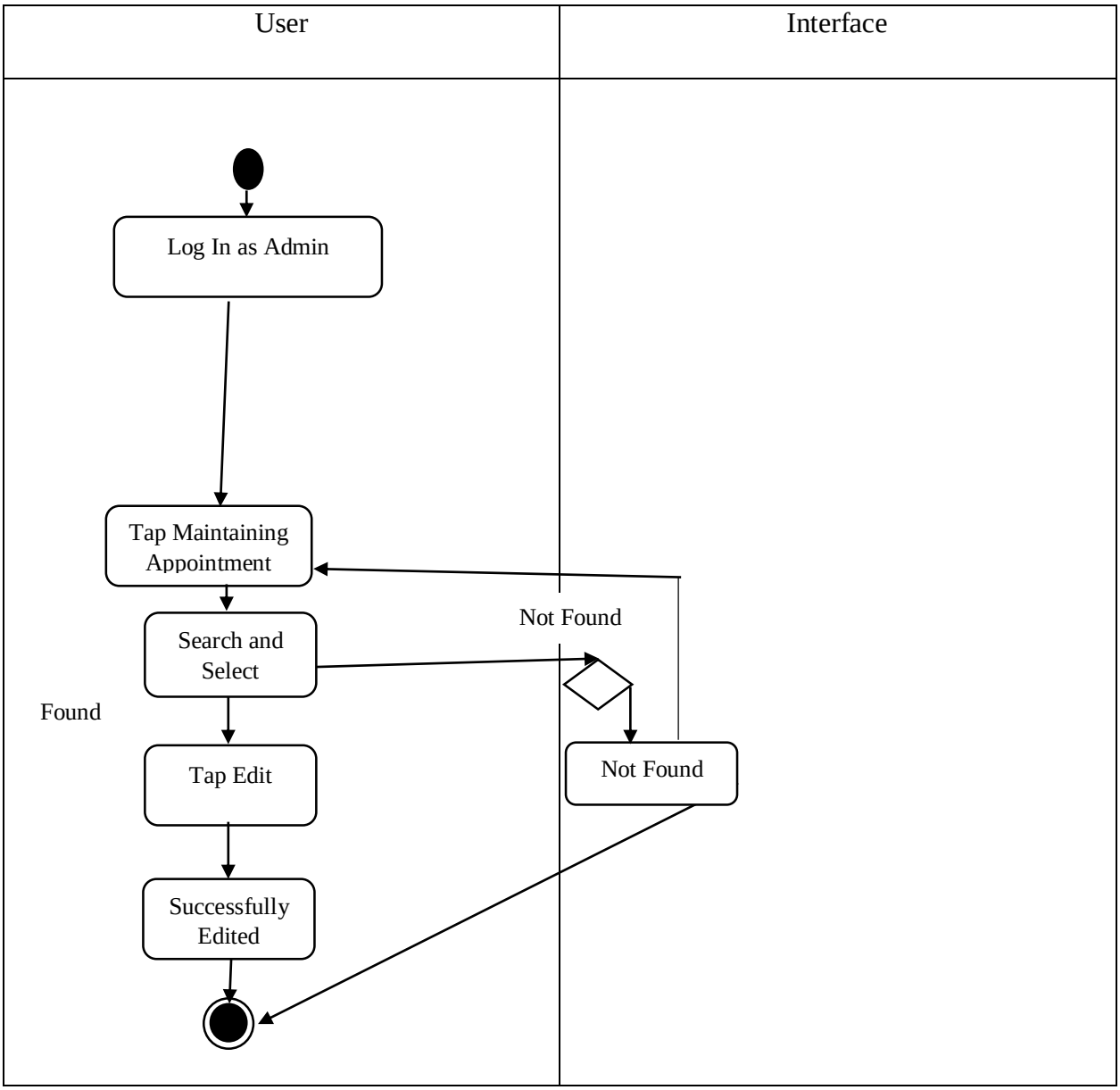
Activity Diagram:

Edit Appointment:



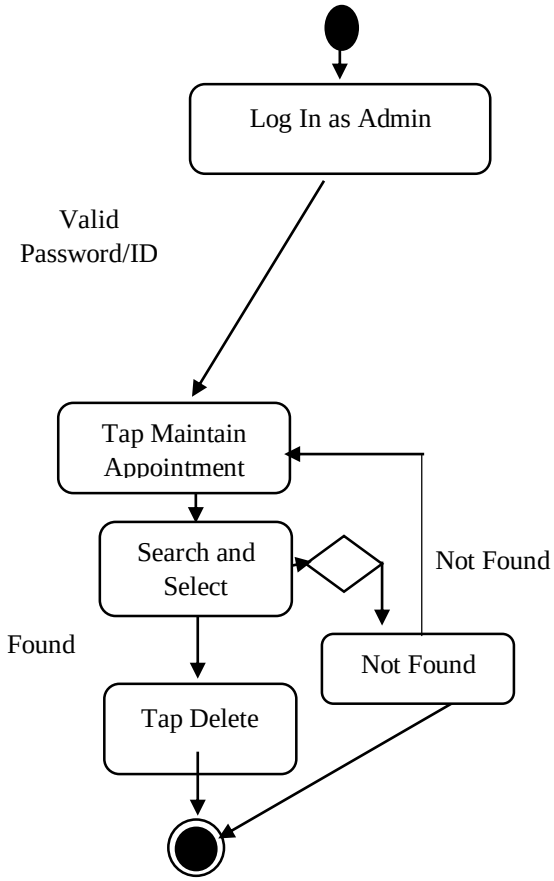
Swimlane Diagram:

Edit Appointment:



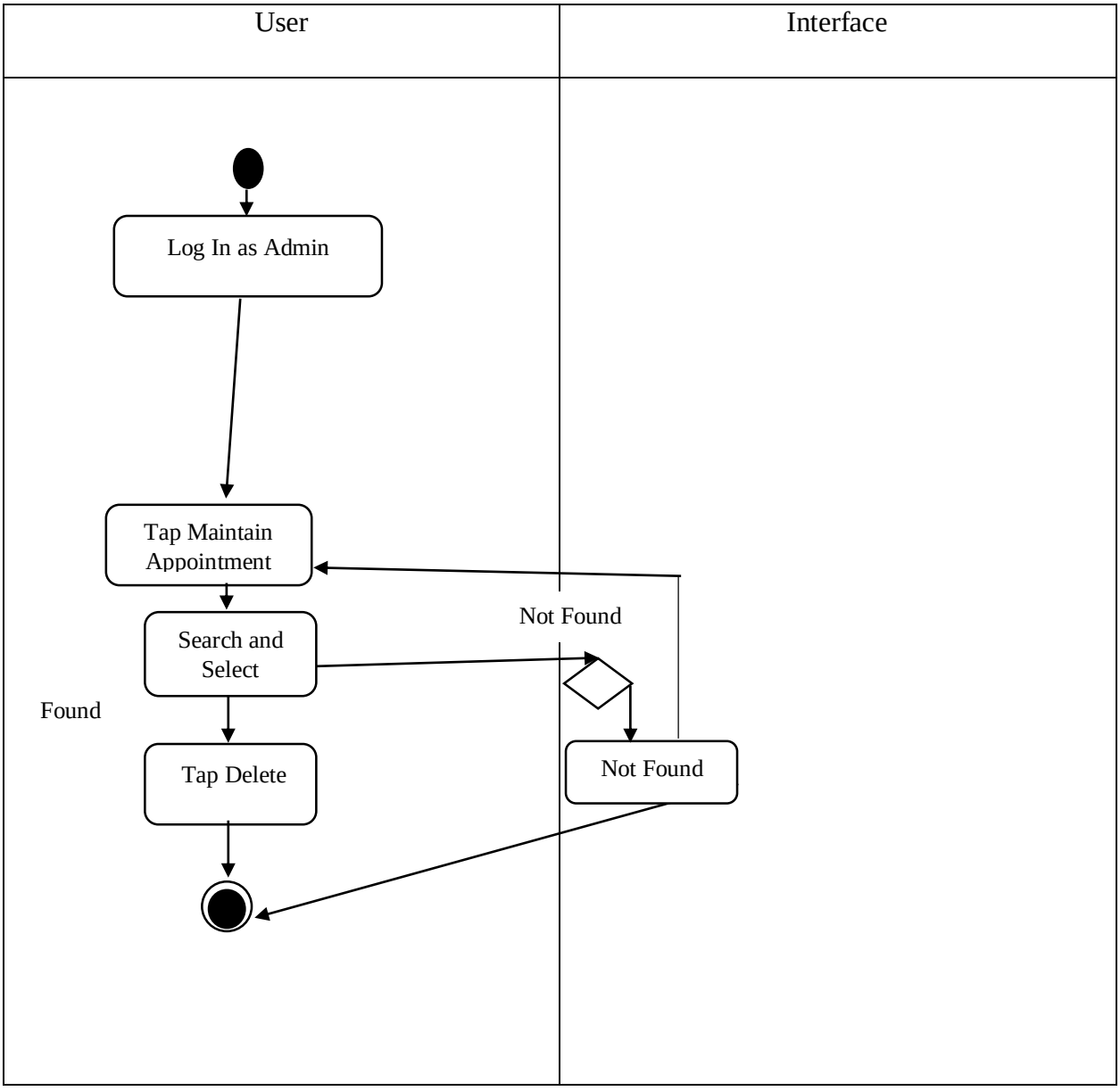
Activity Diagram:

Delete Appointment:



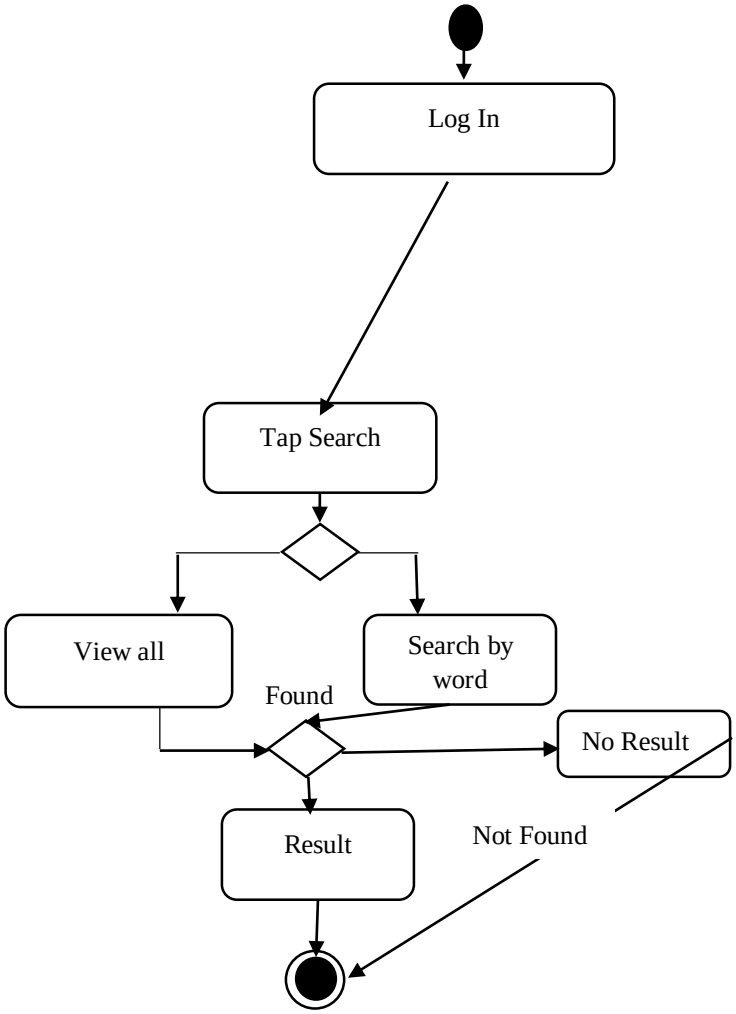
Swimlane Diagram:

Delete Appointment:



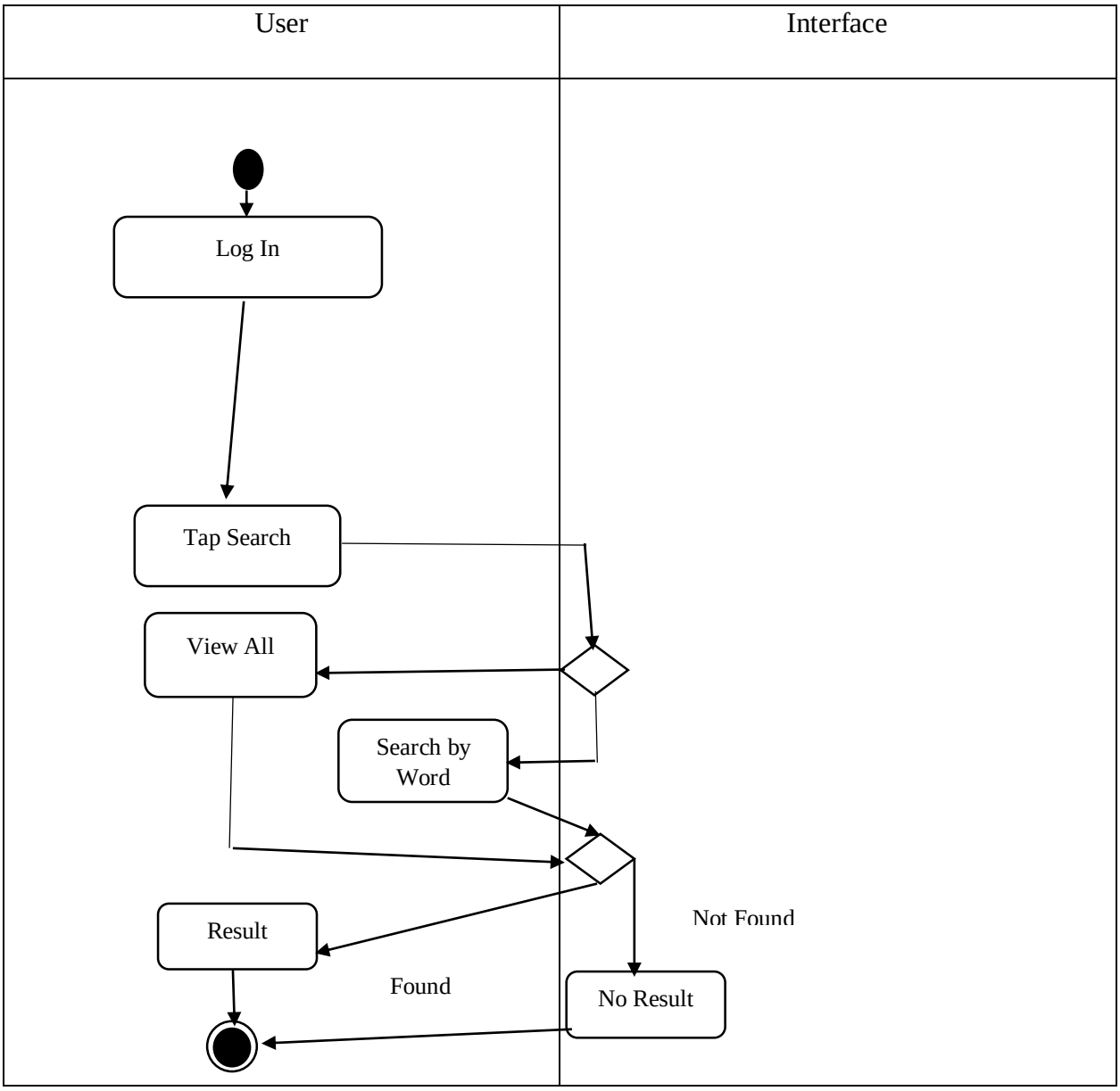
Activity Diagram:

Search:



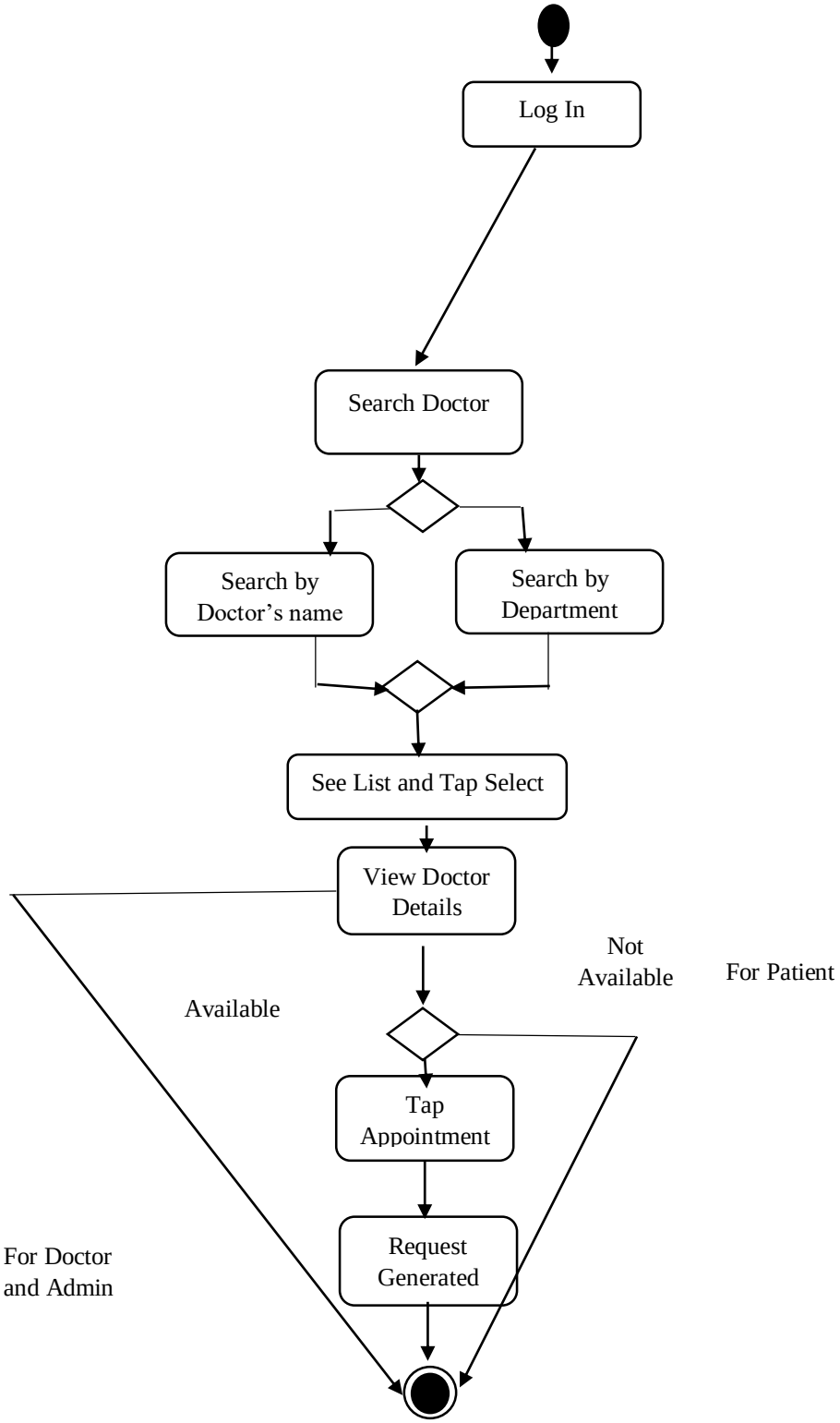
Swimlane Diagram:

Search:



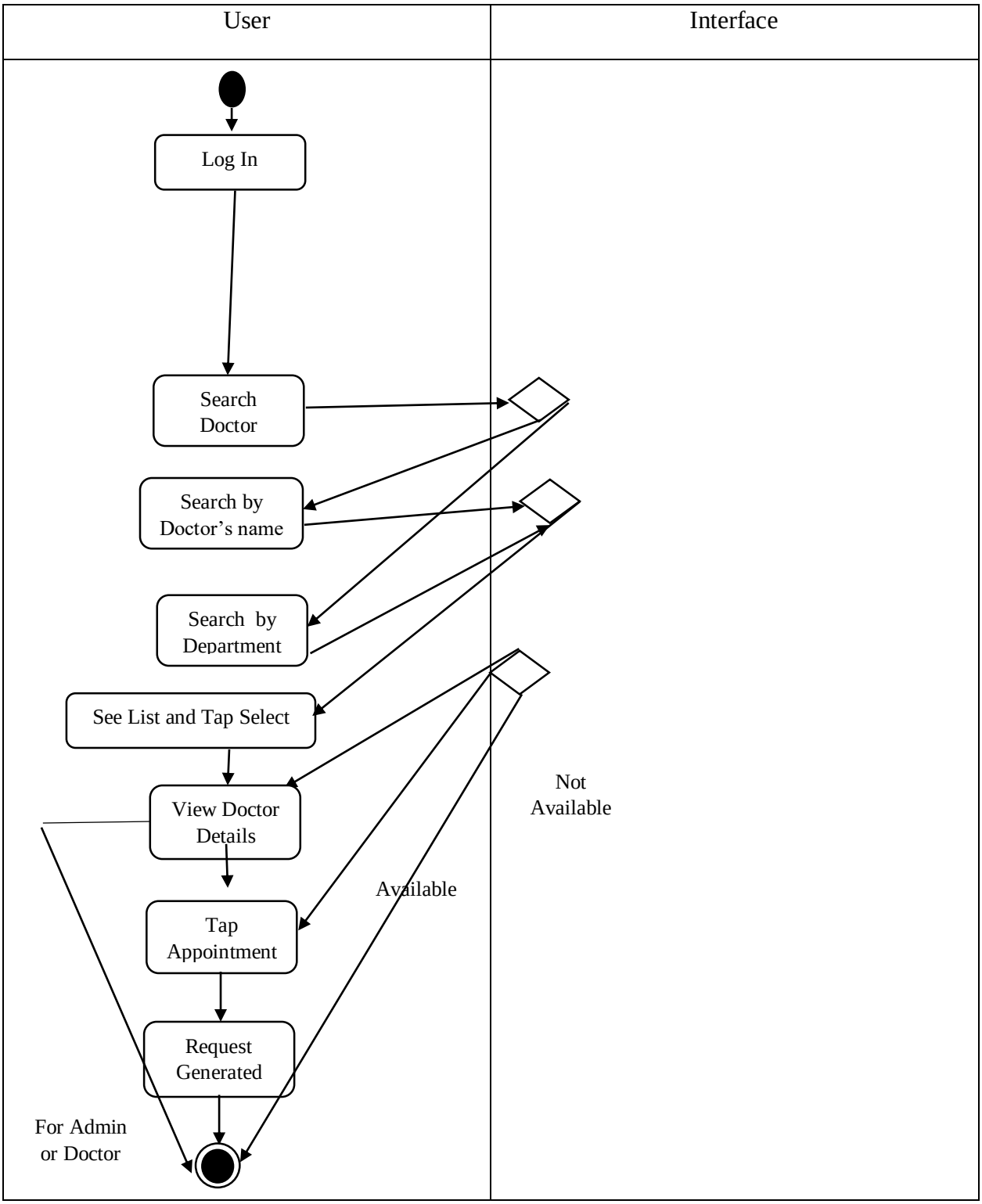
Activity Diagram:

Search Available Doctor, view doctor details and making appointment:



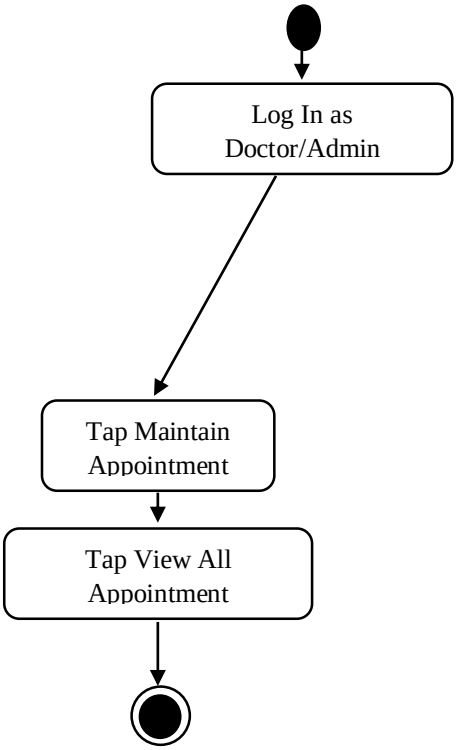
Swimlane Diagram:

Search Available Doctor, View Doctor Details and Making Appointment:



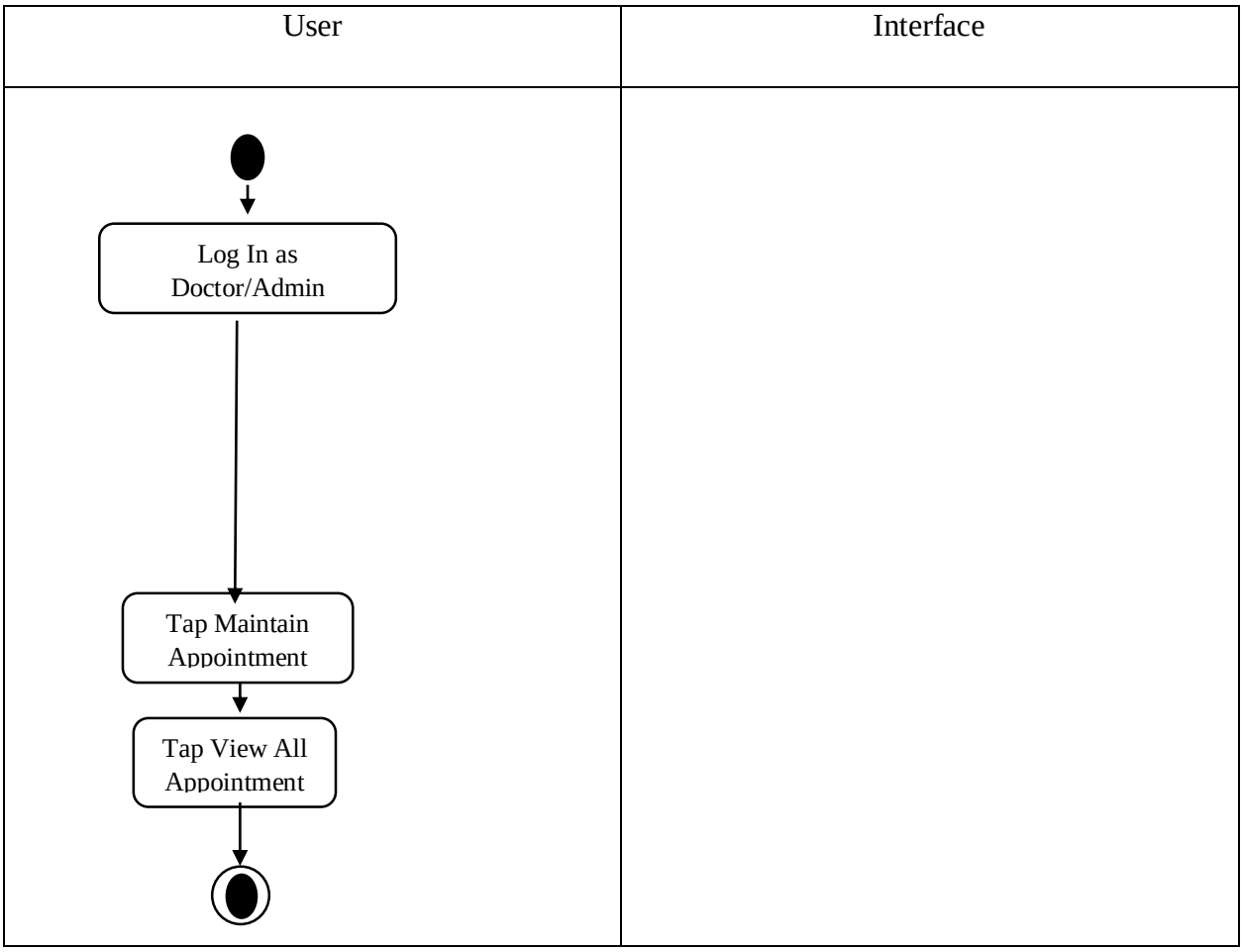
Activity Diagram:

View All Appointment Schedule:



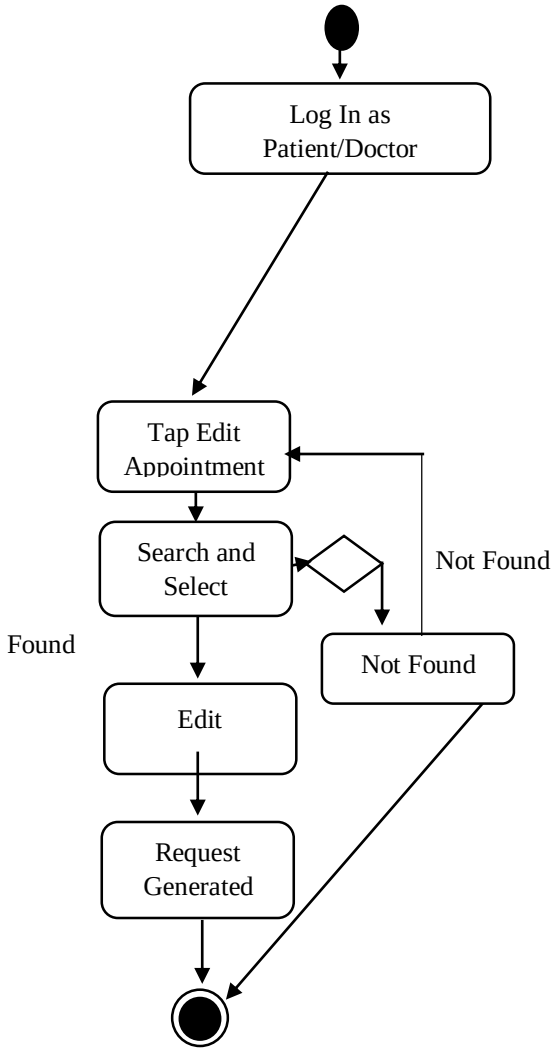
Swimline Diagram:

View All Appointment Schedule:



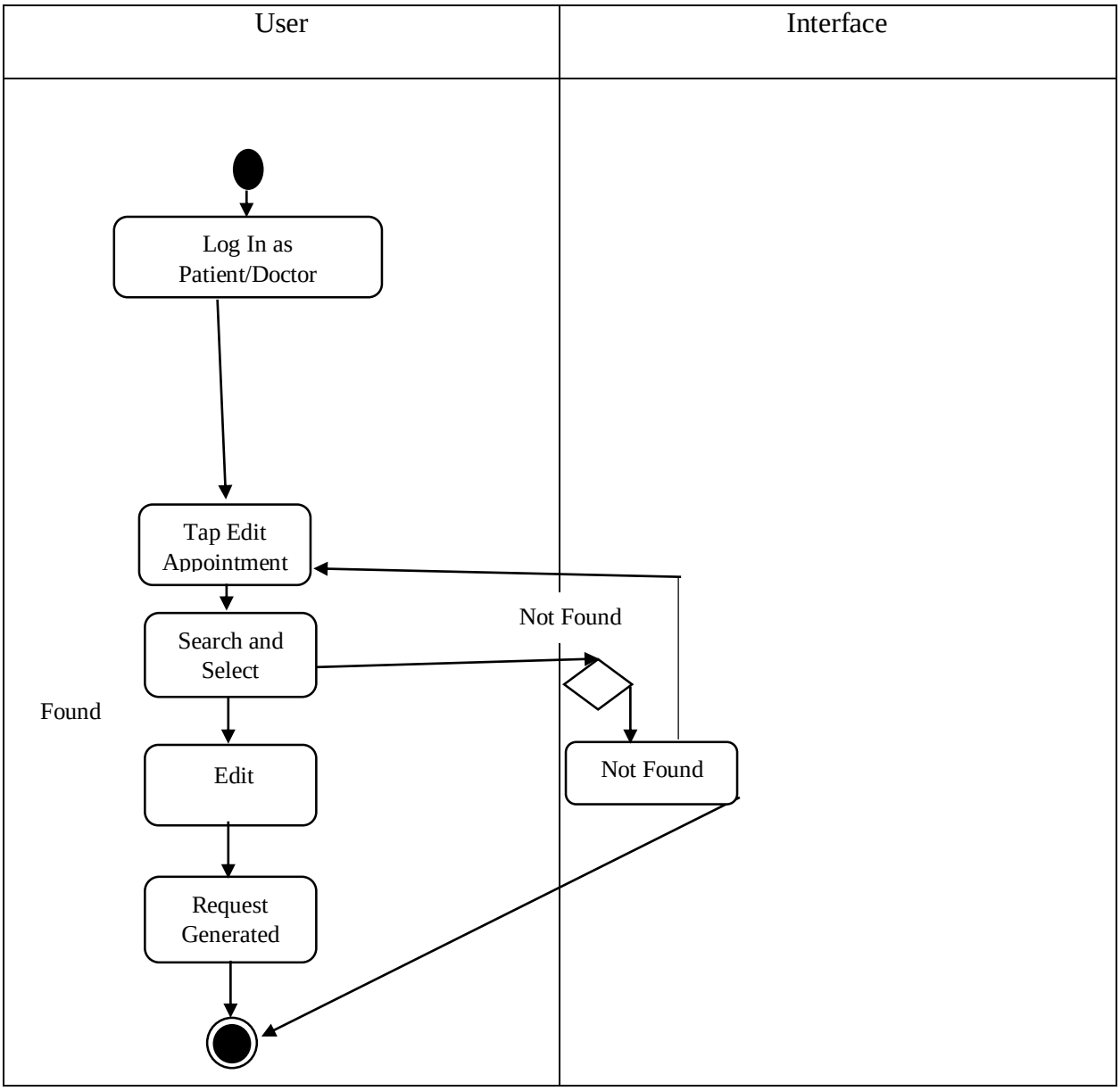
Activity Diagram:

Request for Editing Appointment:



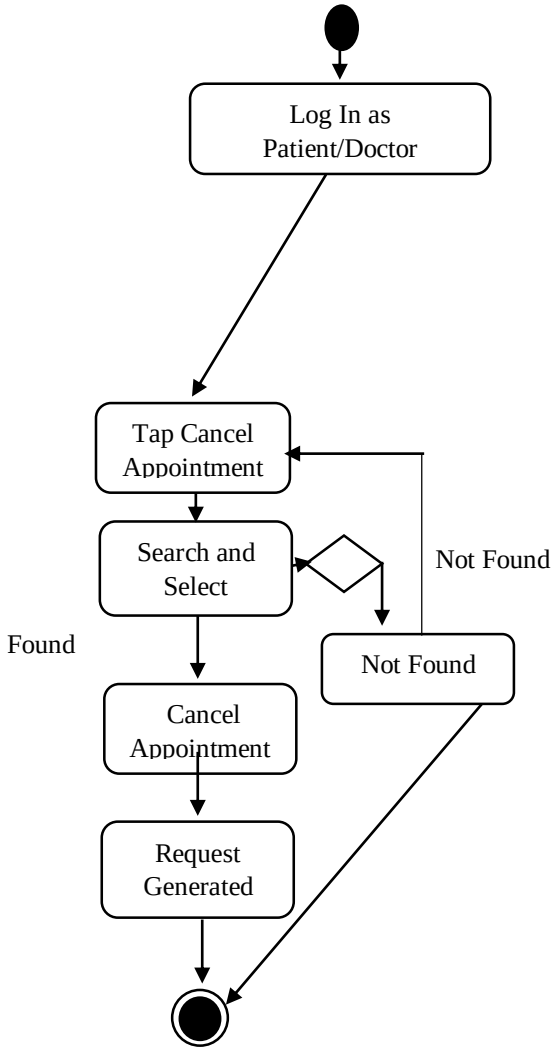
Swimlane Diagram:

Request for Editing Appointment:



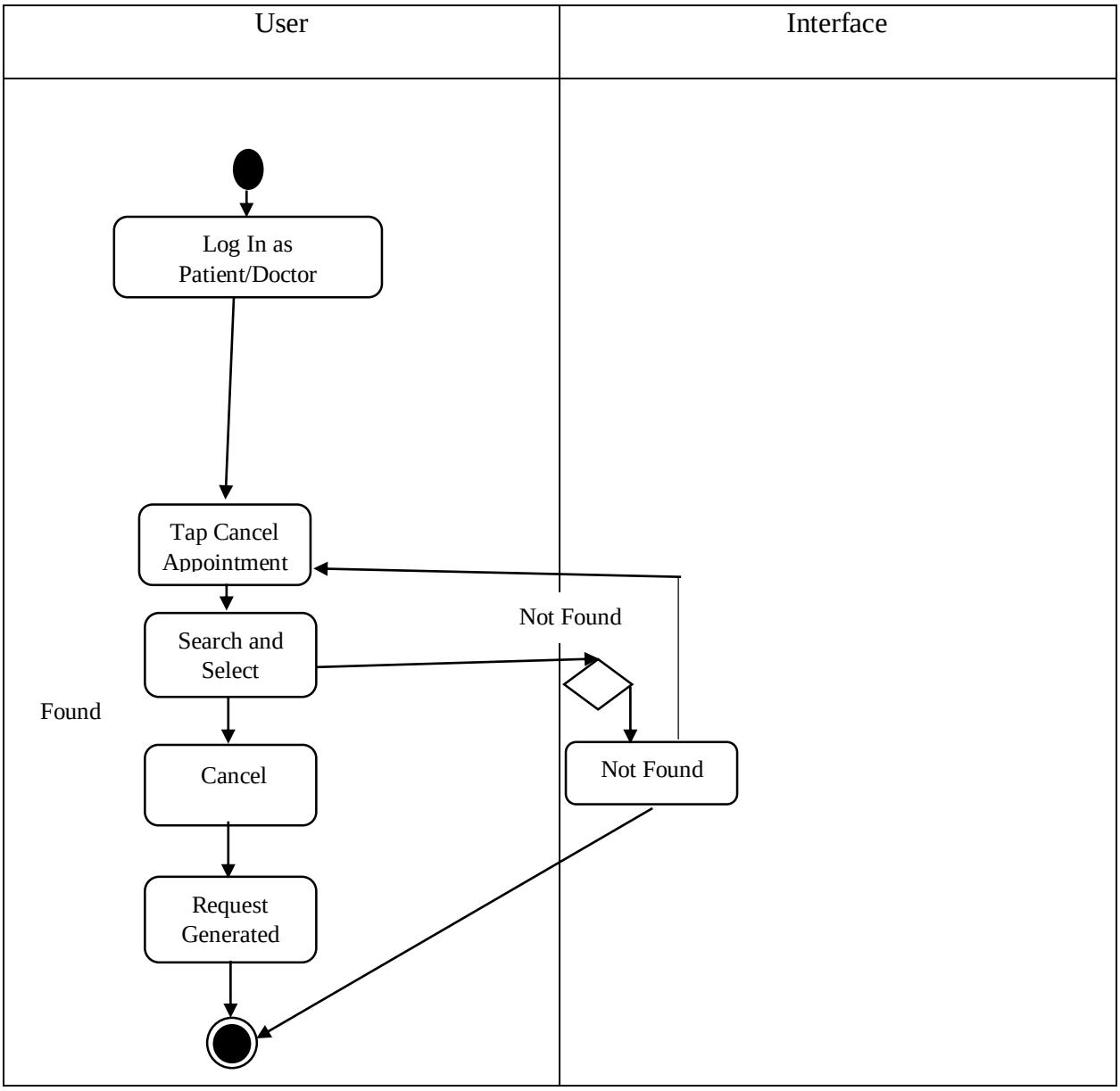
Activity Diagram:

Request for Canceling Appointment:



Swimlane Diagram:

Request for Canceling Appointment:



5. Nonfunctional Requirements

5.1 Performance Requirements

The application should have high performance and low failure rates. It manages facilities required by the casual users quickly and easily. It offers to take appointments faster through online. Application should have their security not compromised by viruses. All database queries and data receiving/transmitting should be done in higher security transmission.

5.2 Safety Requirements

- In Doctor Appointment System admin must be certify the doctor's certificate and qualification. So that, any fake doctor should not be certify as a doctor.
- In Doctor Appointment System I have used firebase database. Data is stored as JSON and synchronized in real time to every connected client. Firebase backups all the data instantly.
- In case the user forgets or loses Password, the repair functionality helps by choosing "forgot password" option in the main login page.

5.3 Security Requirements

All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data. Because, here many information of patient, doctor, admin will be given. So only the legitimate users are allowed to use the application.

5.4 Software Quality Attributes

- **Reliability:** Good validations of user inputs will be done to avoid incorrect storage of records.
- **Availability:** The system shall be available all the time.
- **Maintainability:** During the maintenance stage, SRS document can be referred for any validations.
- **Portability:** This system can be installed in any android phone.
- **Flexibility:** The system keeps on updating the data according to the transactions that takes place.
- **Timeless:** The system carries out all the operations with consumption of very less time.
- **Security**: Security of the system is maintained by giving access to only authenticated user id and password.

6. Analysis Models

6.1 Data Flow Modeling

Data Flow Diagram (DFD):

The DFD takes an input process output view of a system. In the figures, data objects are represents by labeled arrows and transformations are represented by circles.

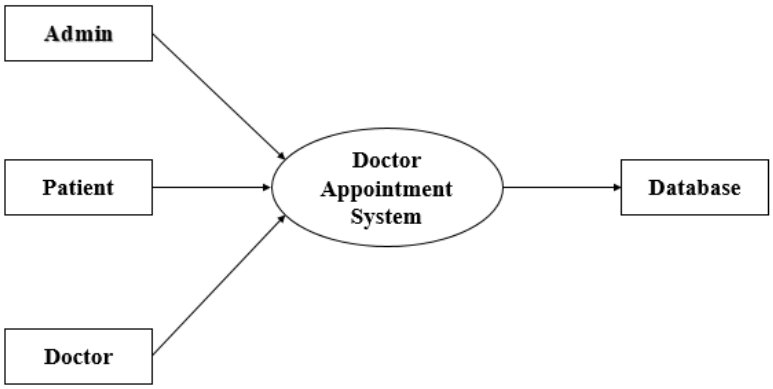


Figure: Level 0 for Doctor Appointment System

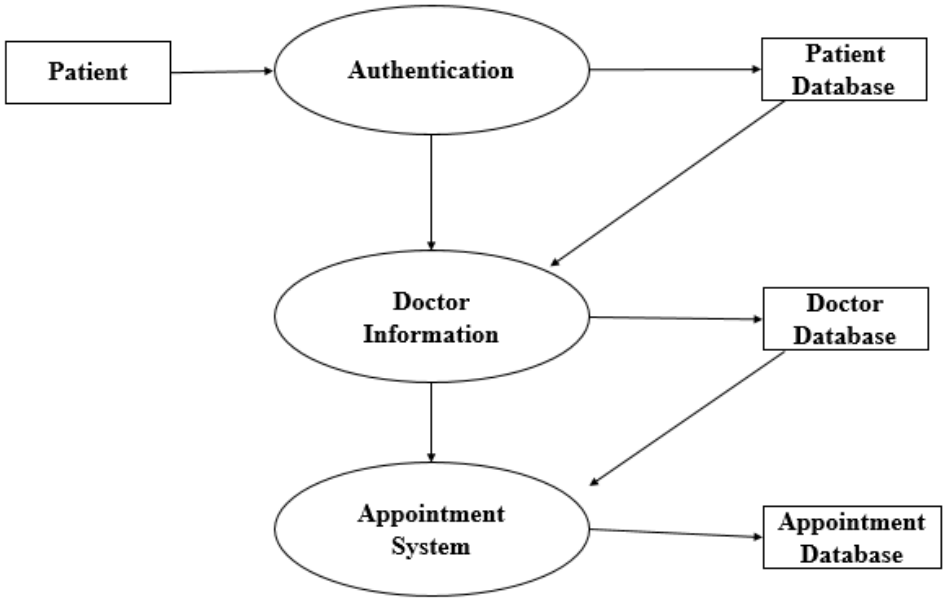


Figure: Level – 1 (Patient) for Doctor Appointment System

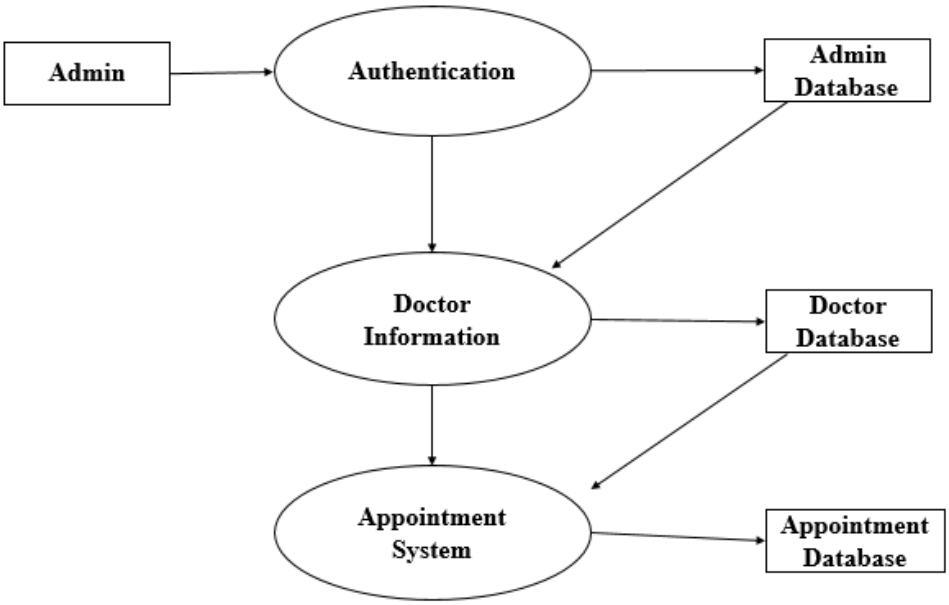


Figure: Level – 1 (Admin) for Doctor Appointment System

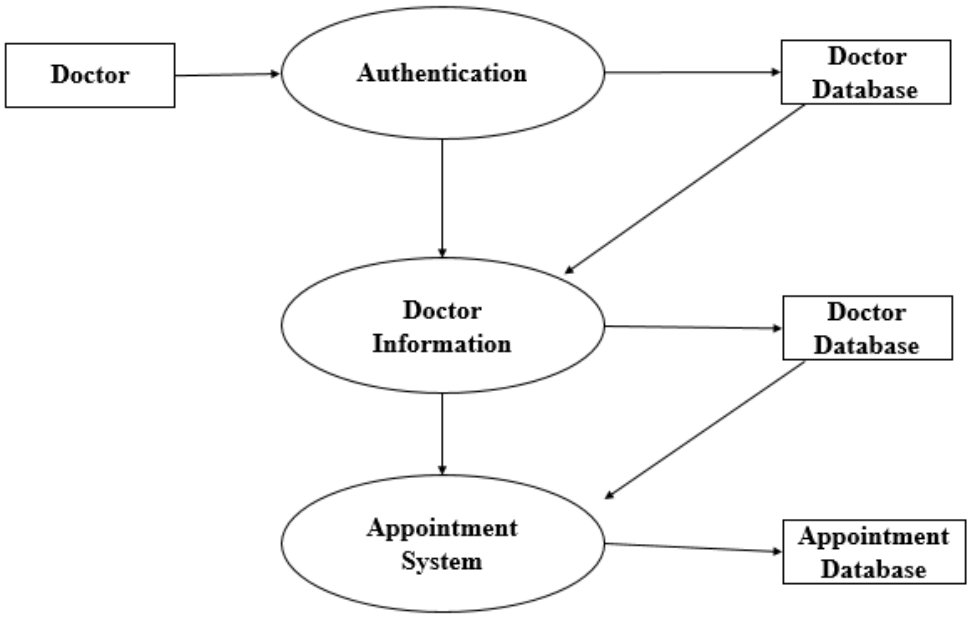


Figure: Level – 1 (Doctor) for Doctor Appointment System

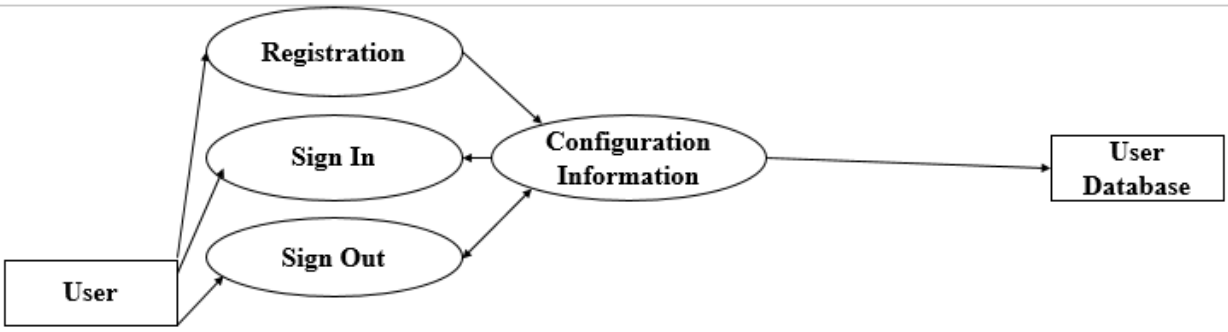


Figure: Level 2.1 Authentication

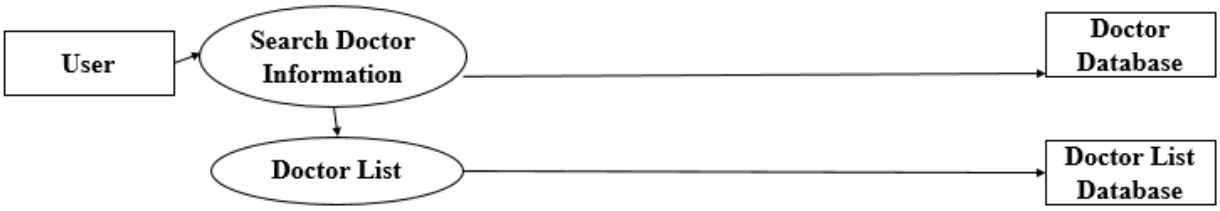


Figure: Level 2.2 Doctor Information

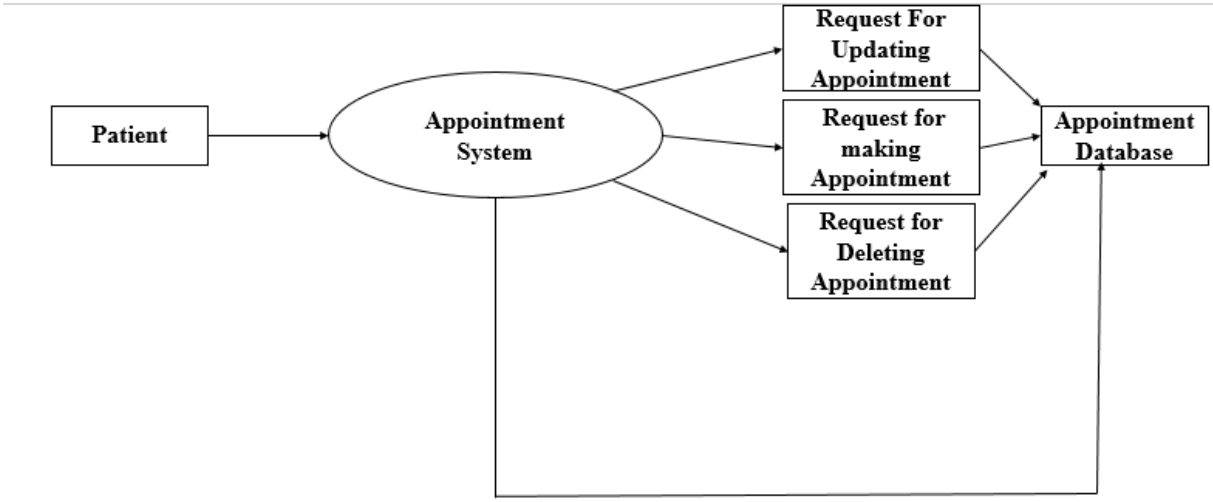


Figure: Level 2.3 (Patient) Appointment System

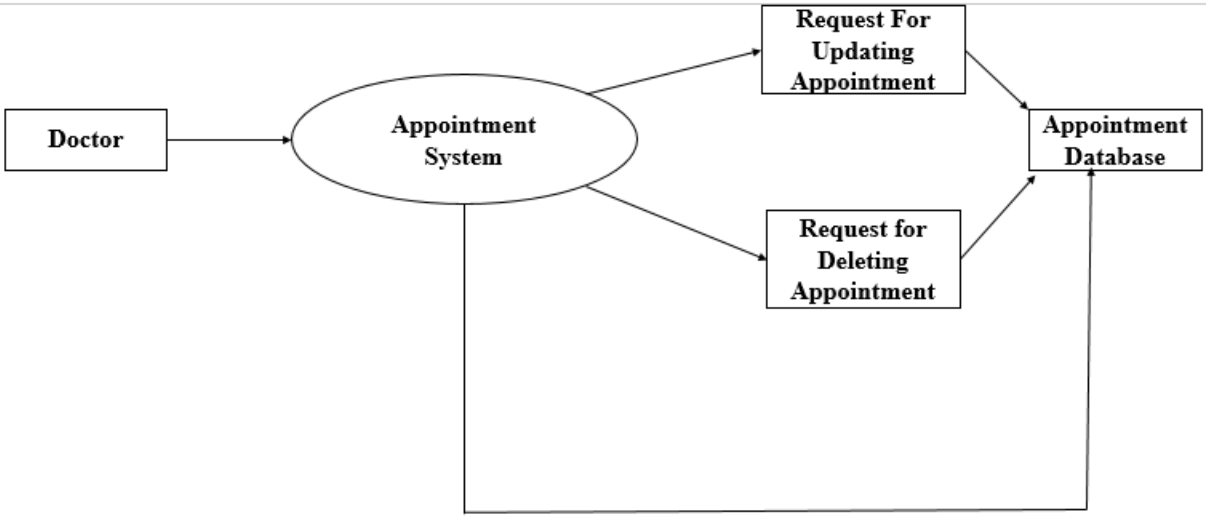


Figure: Level 2.3 (Doctor) Appointment System

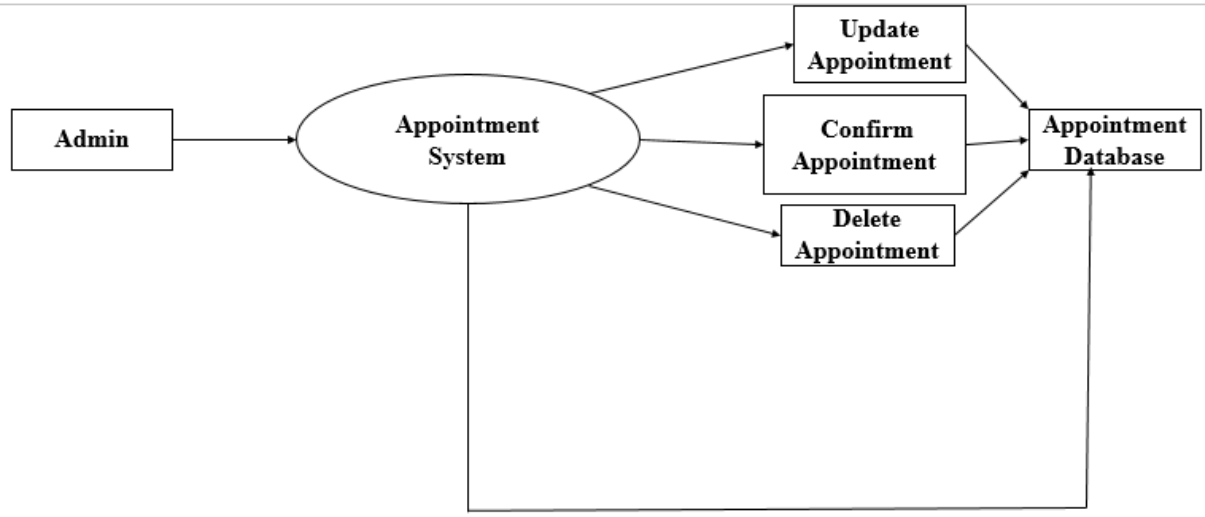


Figure: Level 2.3 (Admin) Appointment System

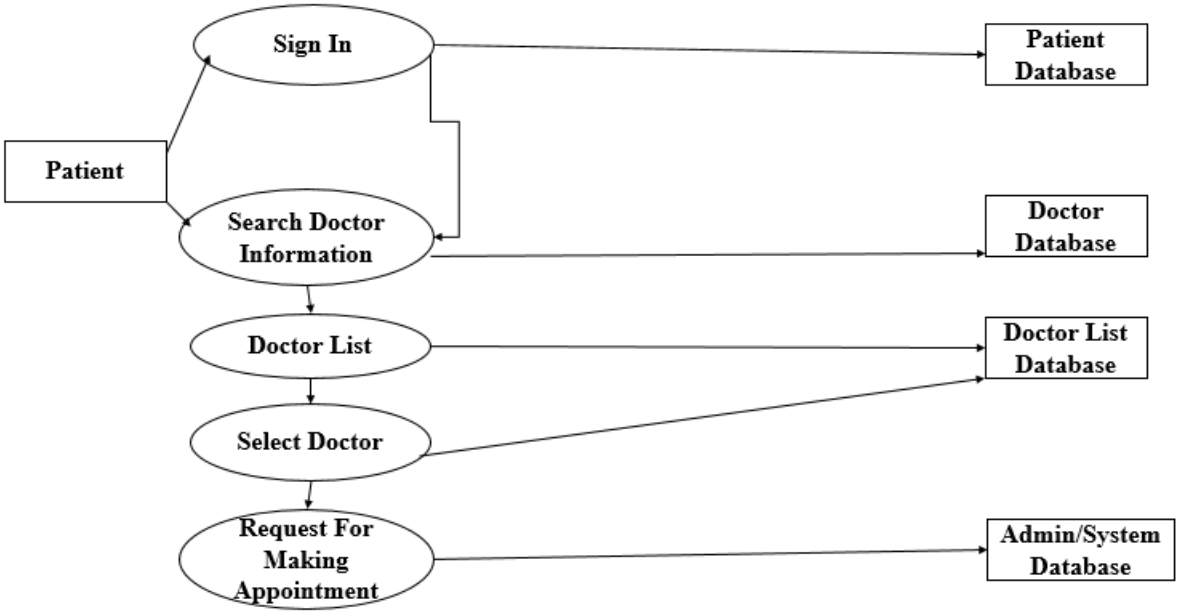


Figure: Level 3.1 (Patient) for Make Appointment

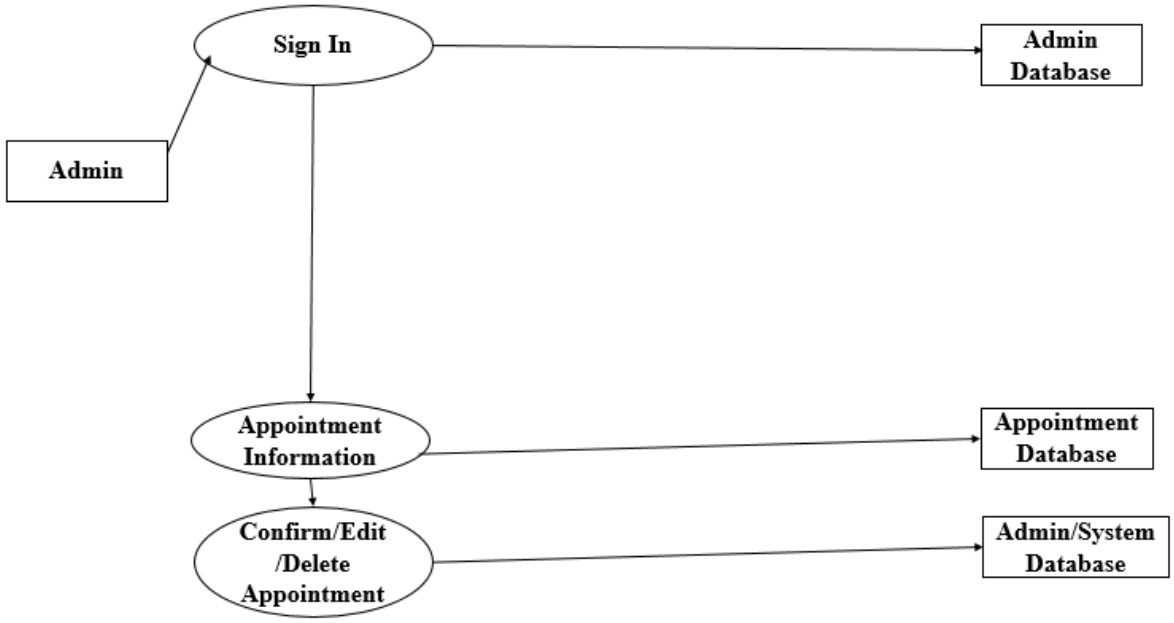


Figure: Level 3.2 (Admin) Confirm Appointment

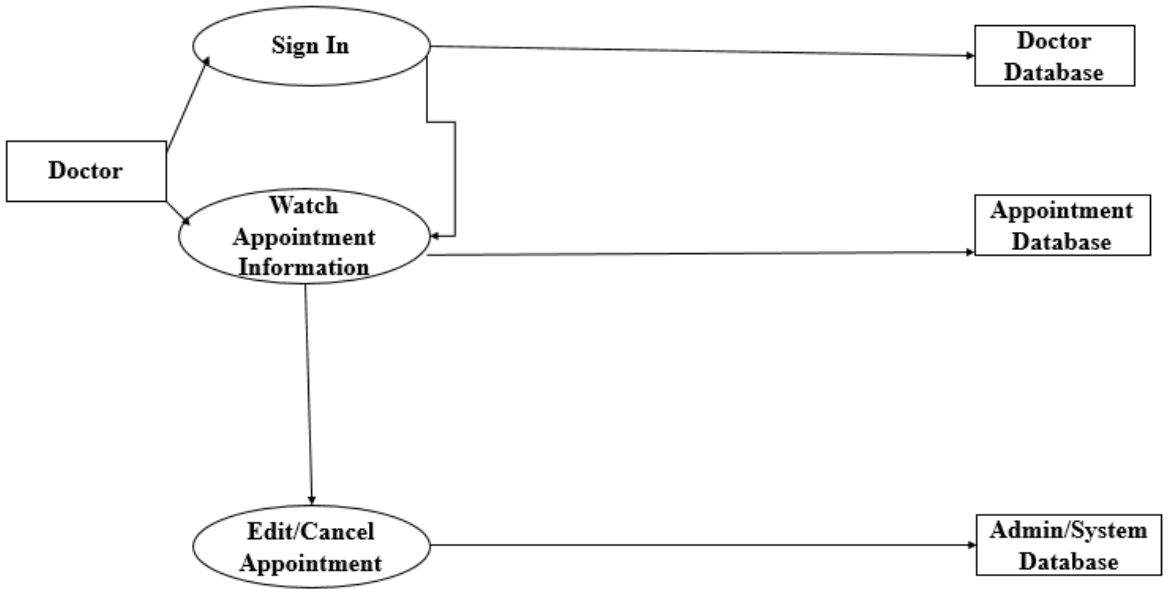


Figure: Level 3.3 (Doctor) Edit Appointment

6.2 Data Modeling

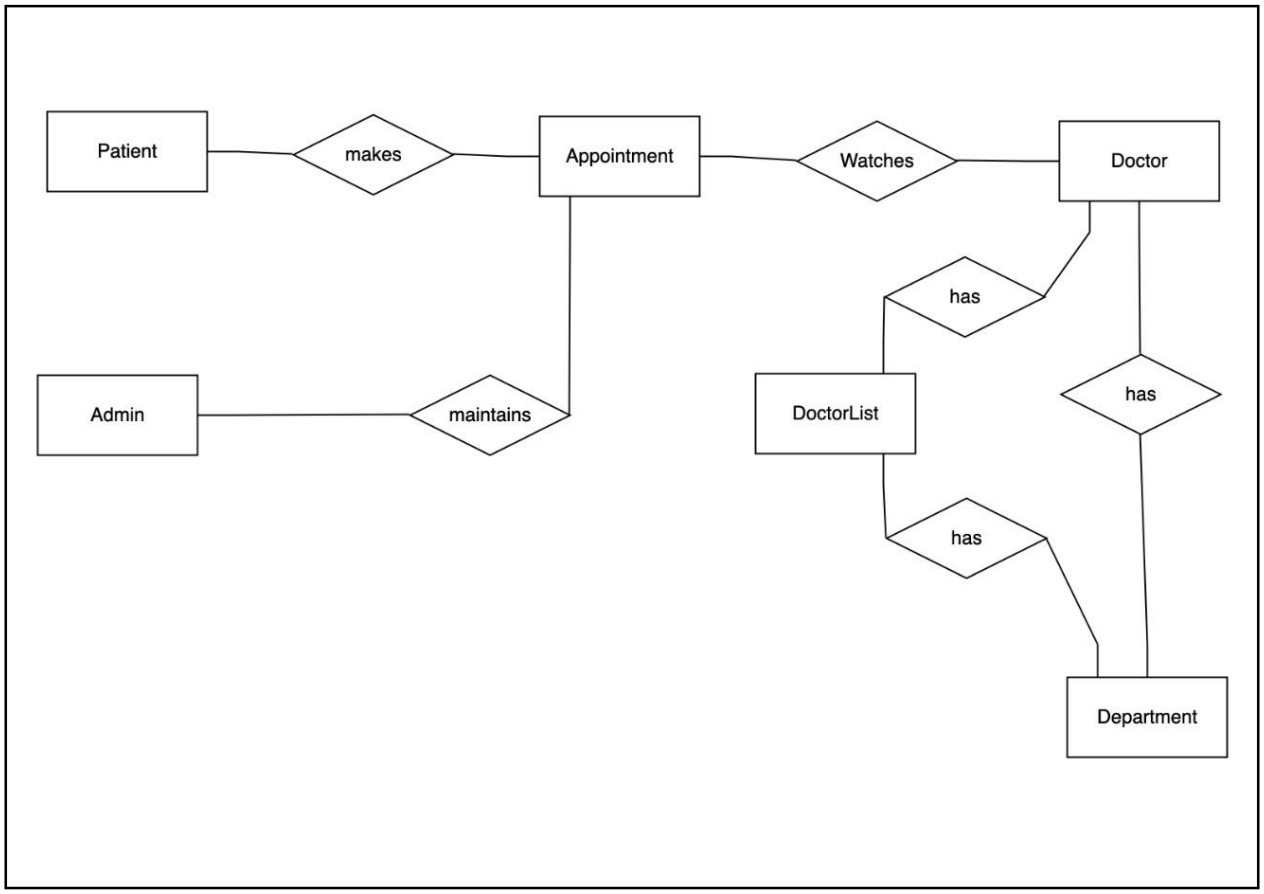
ER Diagram

Entity Identification:

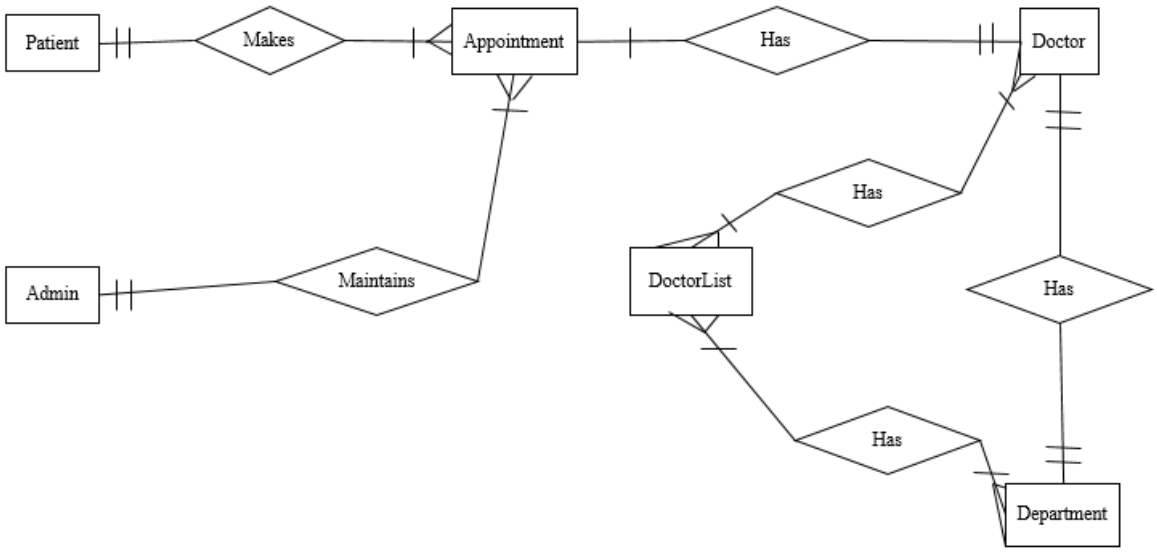
- Patient
- Doctor
- DoctorList
- Appointment
- Department
- Admin

Relationship Identification:

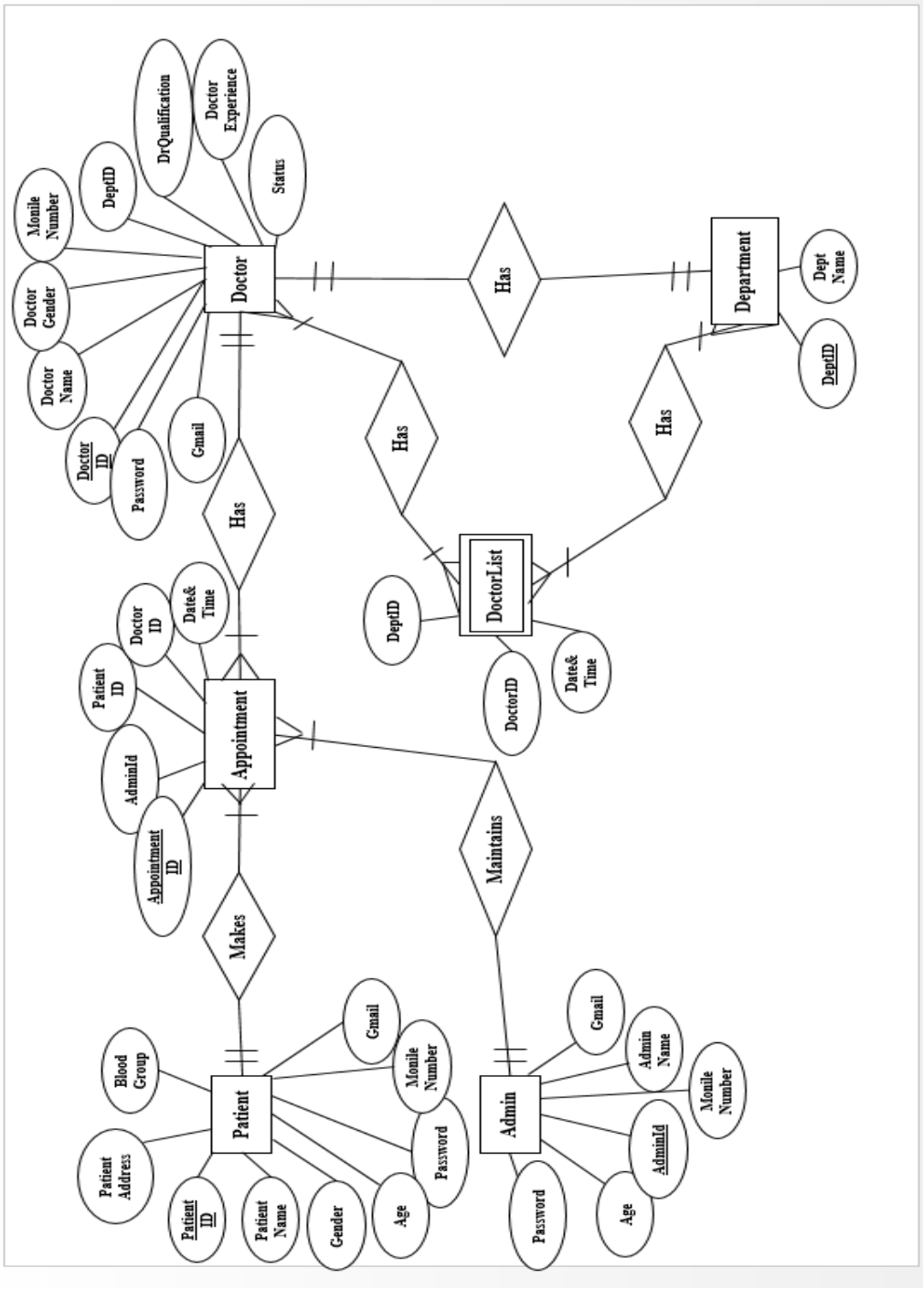
	Patient	Doctor	DoctorList	Appointment	Department	Admin
Patient				makes		
Doctor			has	watches	has	
DoctorList						
Appointment						
Department			has			
Admin				maintains		



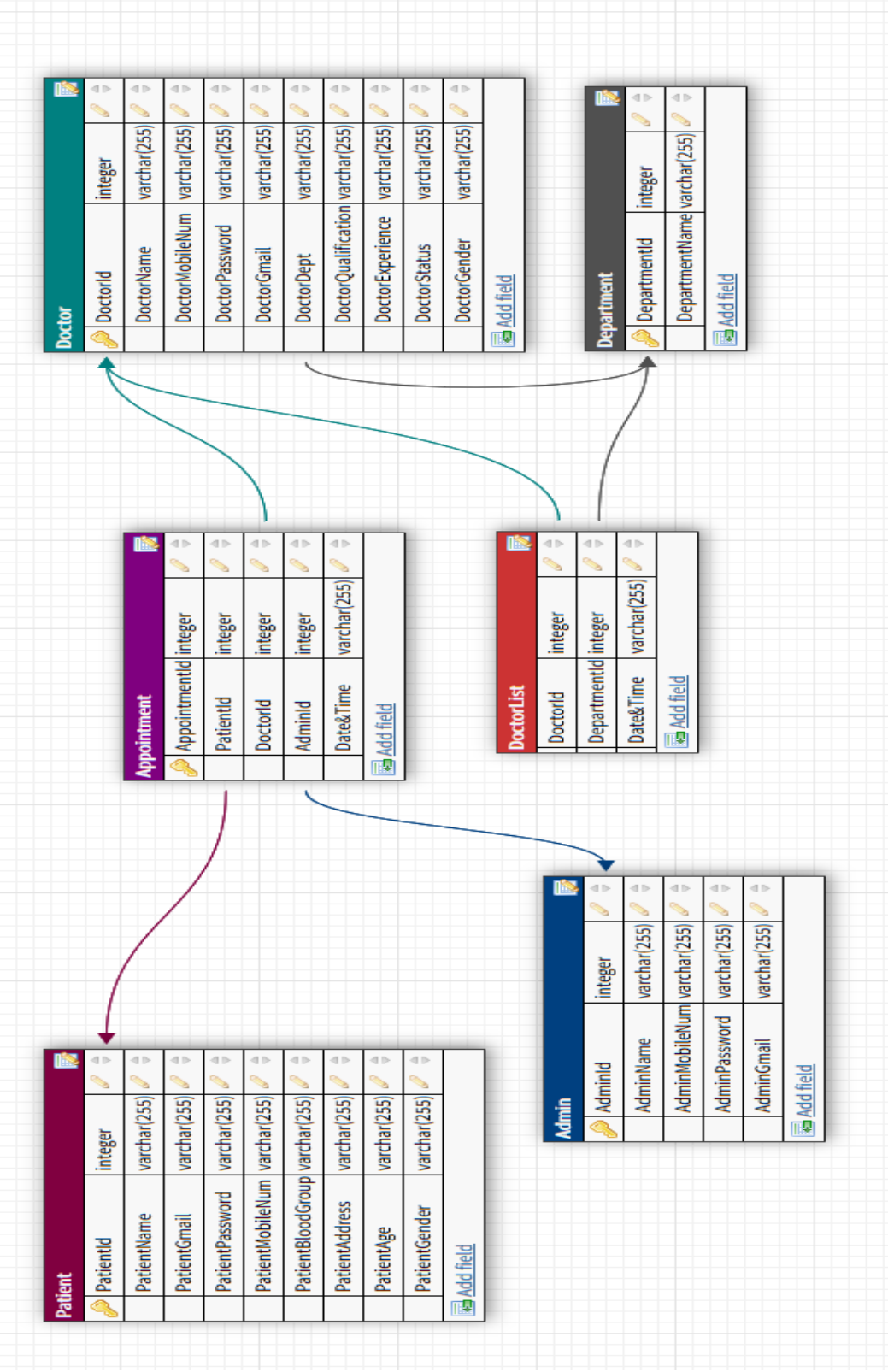
Cardinality Identification:



Create ER Diagram:



Schema Diagram:



6.3 Class Based Modeling

Identifying Analysis Classes:

Step-1: Identifying and categorize all nouns:

External Entities	Patient, Database
Things	Interface, E-mail, Button, Password, Appointment
Occurrences / Events	Search, Make Appointment, Cancel Appointment, Edit Appointment, Delete Appointment, Confirm Appointment, Update, Sign up, Sign in, Sign out
Roles	Doctor, Patient, Admin
Organizational units	Department, Doctor List
Places	
Structures	Internet, Mobile, Computer, Server

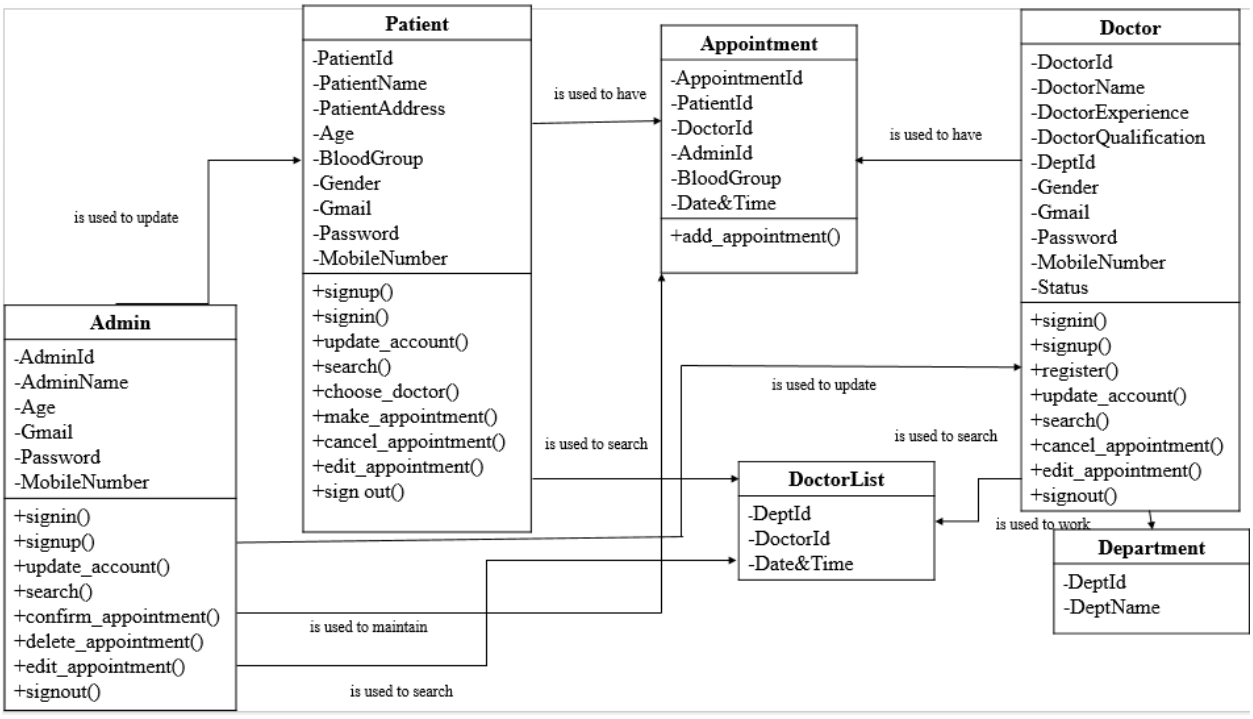
Step-2: Selection of potential class

Selection characteristics:

1. Retained Information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential Requirements

Potential class	**Characteristics Number That Applies**
Patient	Apply: 1, 2, 3, 4, 5, 6
E-mail	Apply: 2, 4, 5
Button	Apply: 2, 4
Password	Apply: 1, 2, 4, 5
Appointment	Apply: 1, 2, 3, 4, 5,6
Search	Apply: 1, 2, 4, 5
Make Appointment	Apply: 1, 2, 6
Edit Appointment	Apply: 1, 2, 6
Delete Appointment	Apply: 1, 2, 6
Cancel Appointment	Apply: 1, 2, 6
Confirm Appointment	Apply: 1, 2, 6
Update	Apply: 1, 2, 4, 5
Sign in	Apply: 1, 2, 6
Sign up	Apply: 1, 2, 6
Sign out	Apply: 1, 2, 6
Doctor	Apply: 1, 2, 3, 4, 5, 6
Admin	Apply: 1, 2, 3, 4, 5, 6
Department	Apply: 1, 2, 3
Doctor list	Apply: 1, 2, 3, 6
Internet	Apply: 1, 2, 6
Mobile	Apply: 1, 2, 6
Computer	Apply: 1, 2, 6
Server	Apply: 1, 2, 6

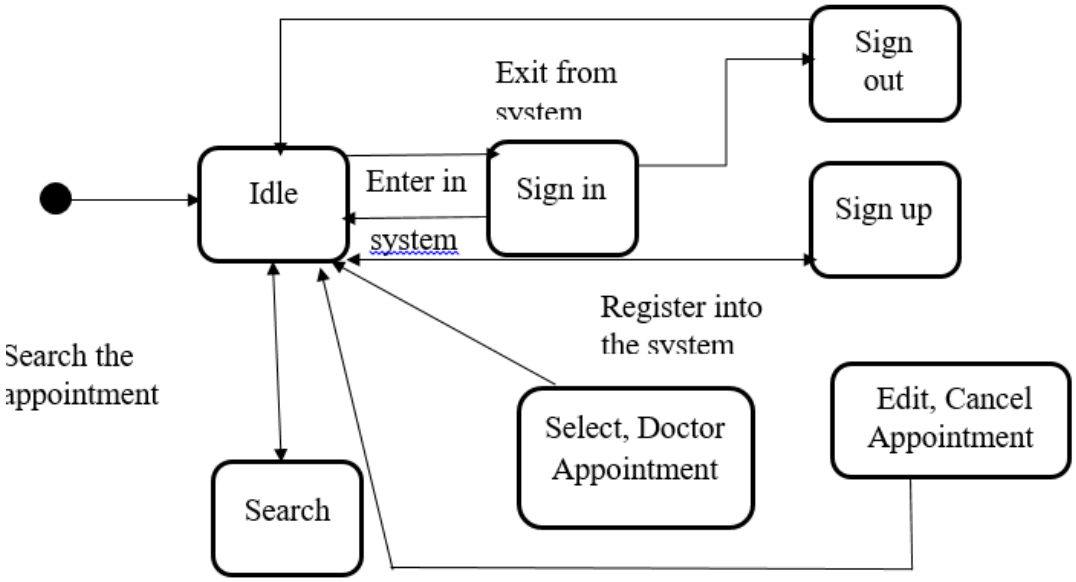
Class Responsibility Collaborator (CRC)



6.3 Behavioral Modeling

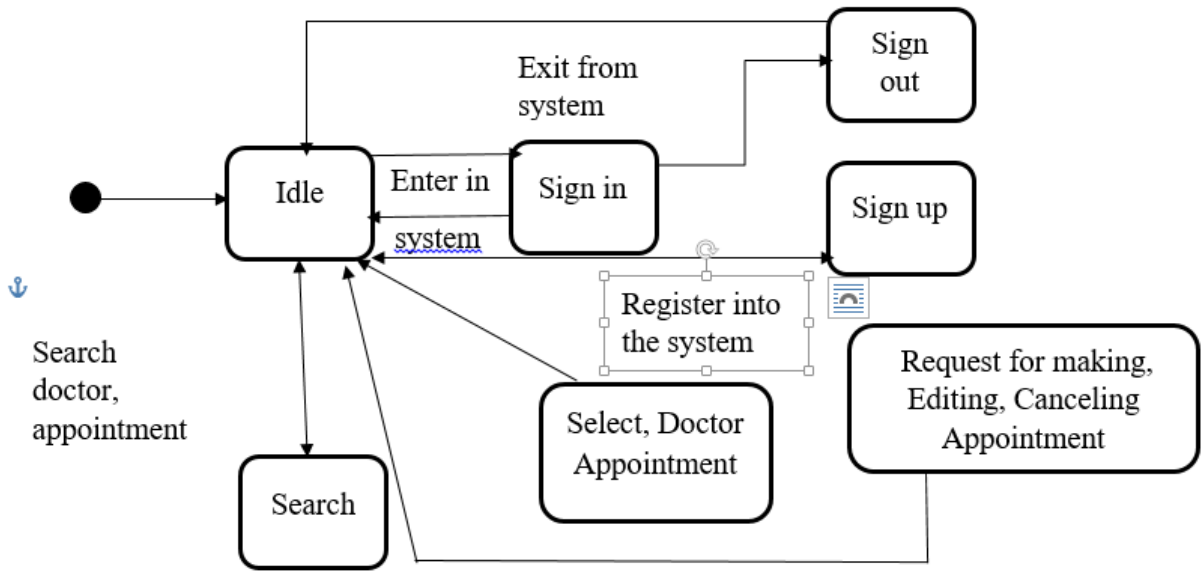
State Diagram

Admin Class:



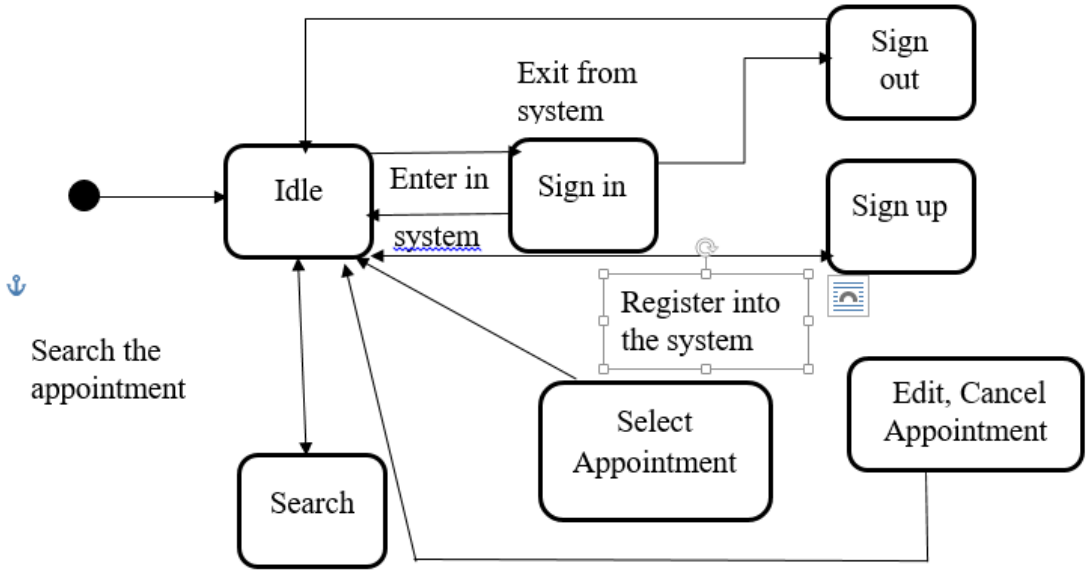
State diagram (Admin class)

Patient Class:



State diagram (Patient class)

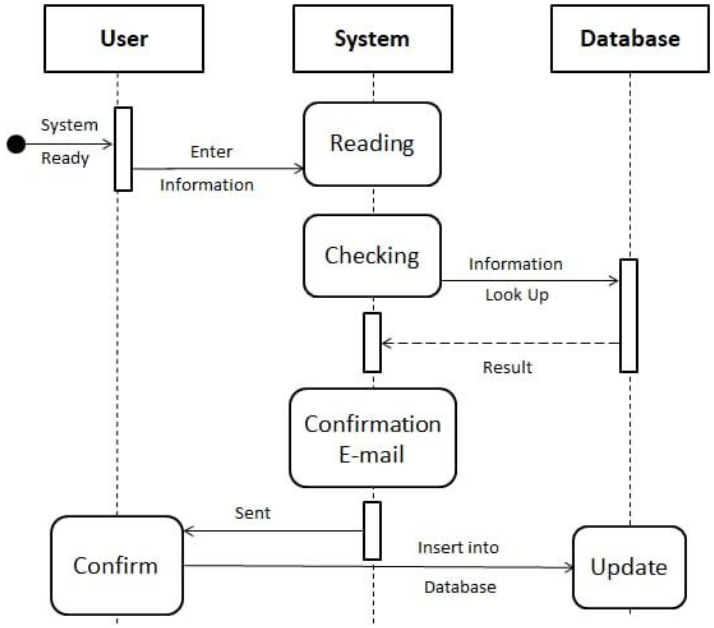
Doctor Class:



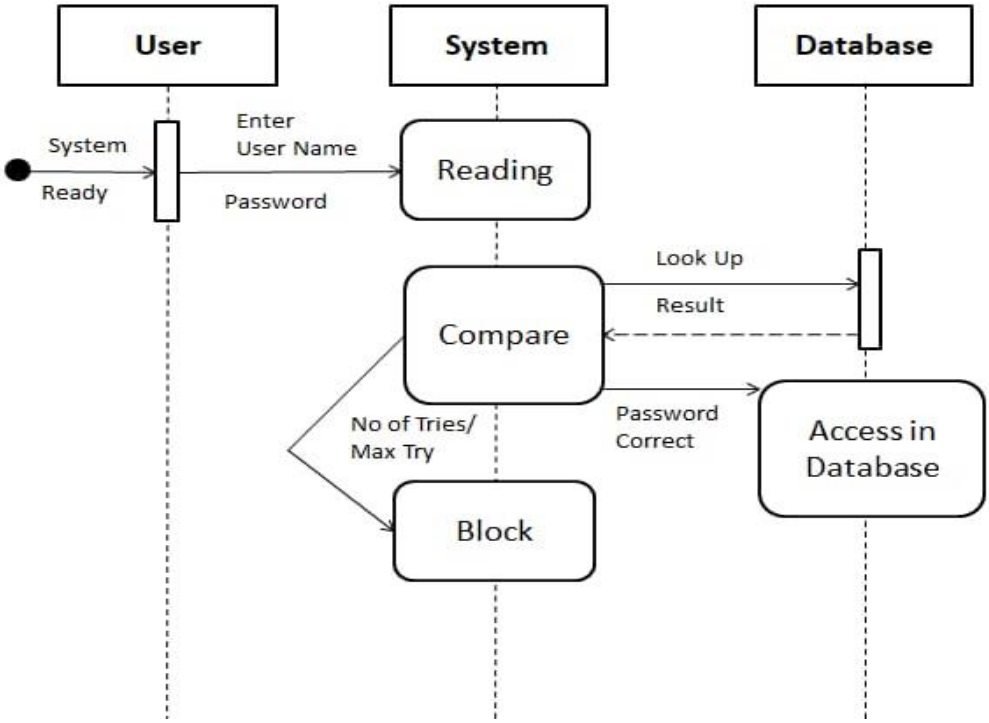
State diagram (Doctor class)

Sequence Diagram

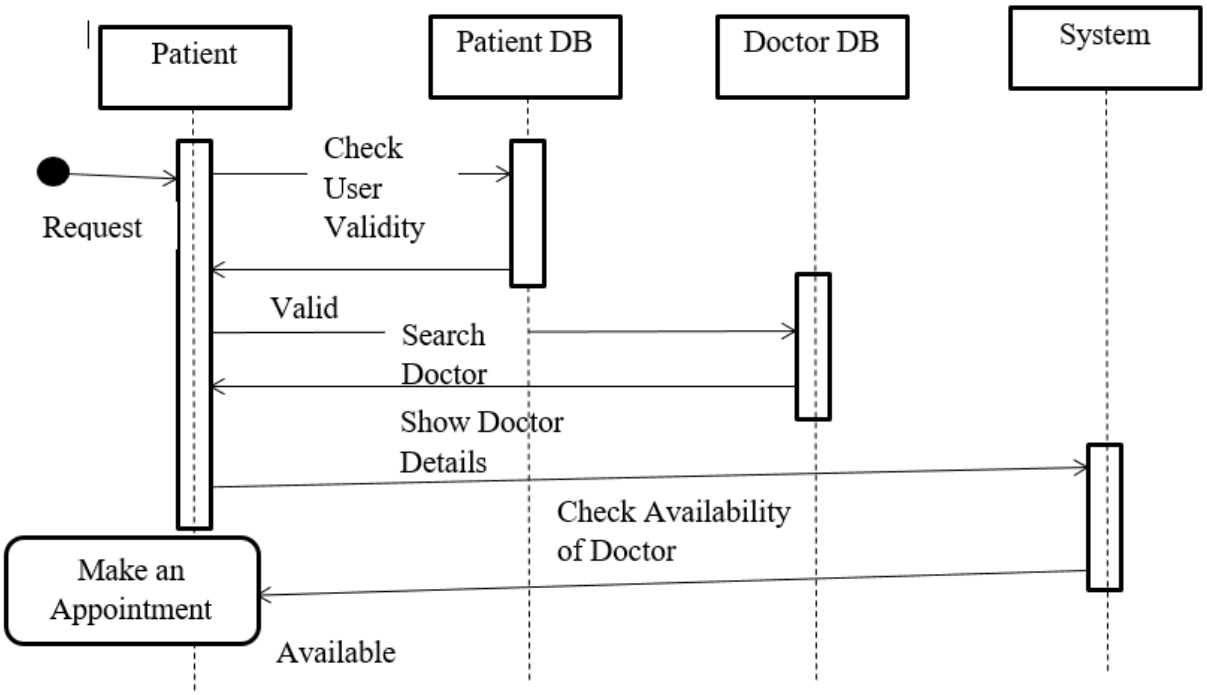
Registration:



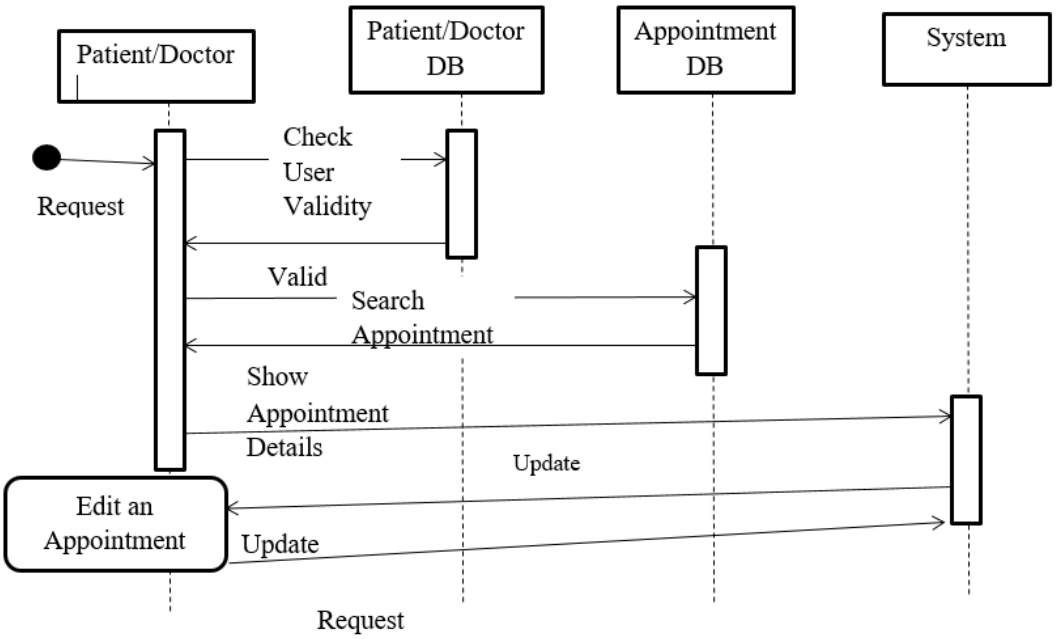
SignIn:



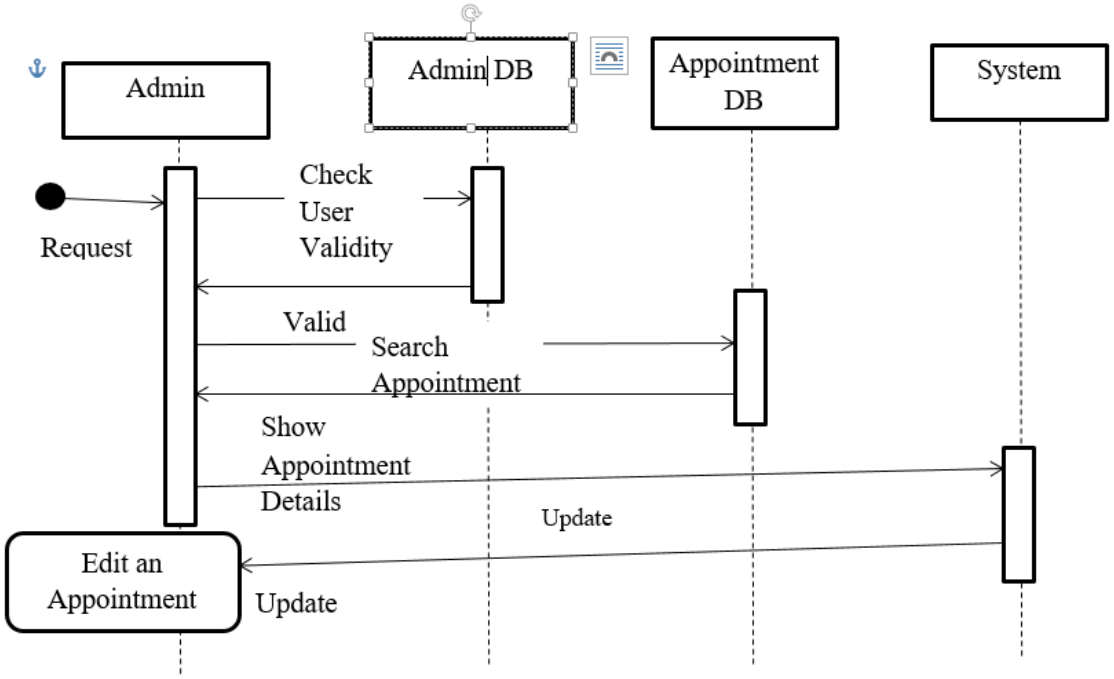
Make an appointment:



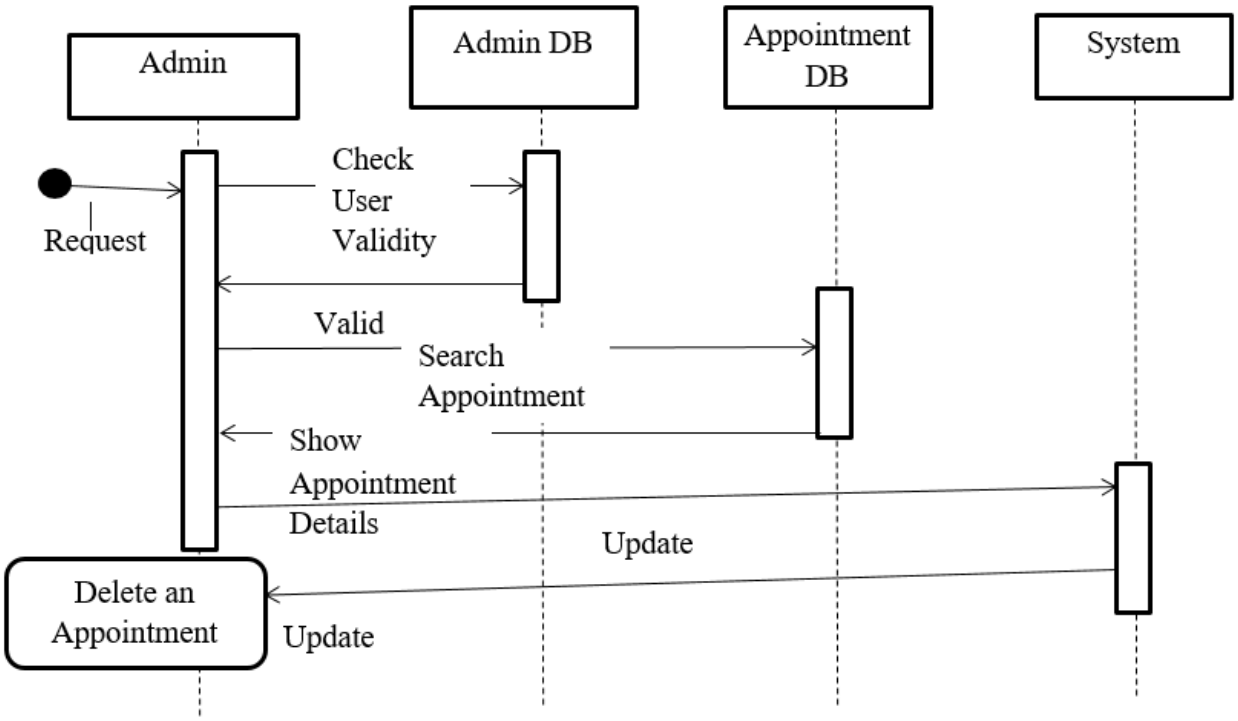
Edit an appointment (Doctor/Patient):



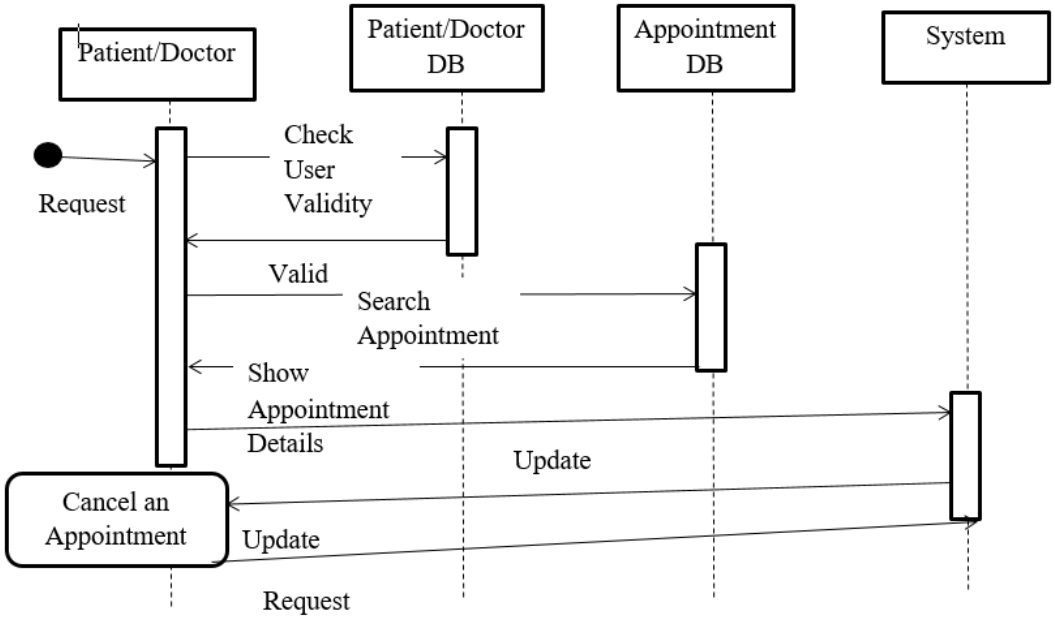
Edit an appointment (Admin):



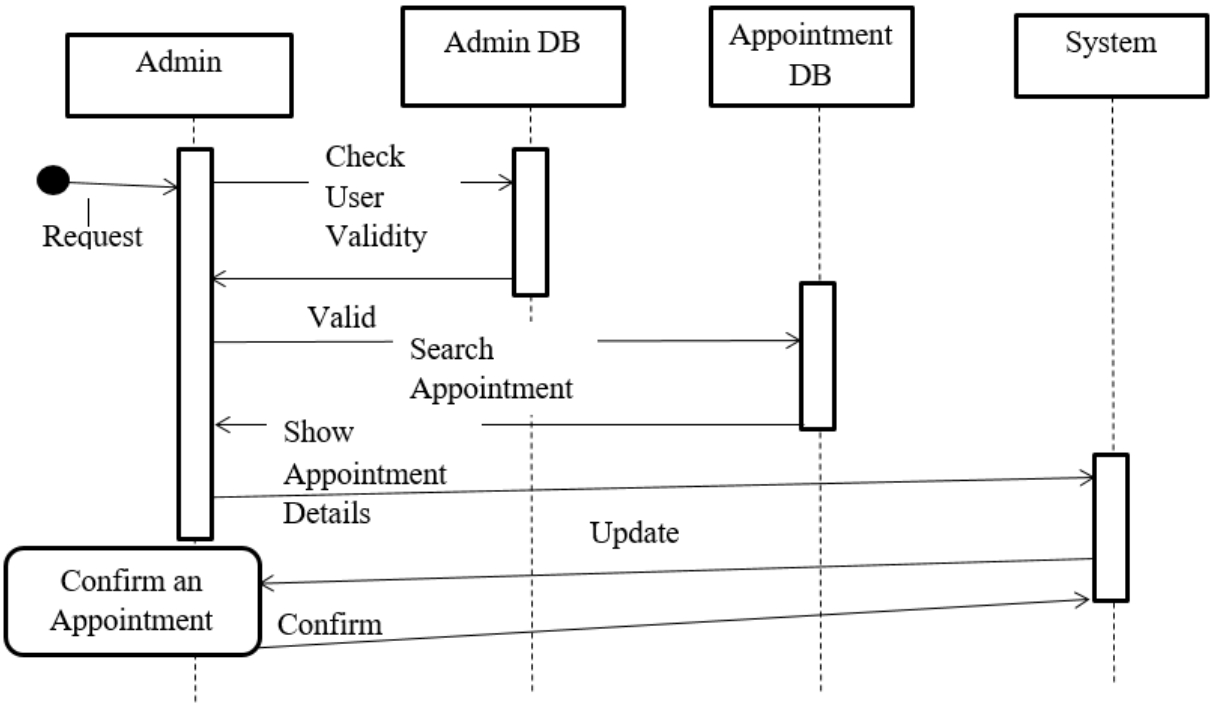
Delete an appointment:



Cancel an appointment:



Confirm an appointment:



Appendix A Glossary:

1. **DB:**
    a. **D**ata**B**ase.
    b. Big memory address block which contains large set of data.
2. **CRC:**
    a. **C**lass **R**elationship **C**ollaboration.
3. **SRS:**
    a. **S**oftware **R**equirements **S**tatement.
    b. Statement clarifying the what a software project is supposed to be engineered to doc. Specifies the limits, constraints, and big-picture, abstract plan of the software engineering.
4. **SQL**:
    a. **S**tructured **Q**uery **L**anguage.
5. **DFD:**
    a. **D**ata **F**low **D**iagram.
6. **ER:**
    a. **E**ntity **R**elationship.
7. **JSON:**
    a. **J**ava **S**cript **O**bject **N**otation