**Assignment 2**

**For**

**Second Year Project**

**Bachelor of Science in Information Technology**

Detap

**Submitted by**

**DEMA LHAMO (12190047)**

**Gyalpozhing College of Information Technology**

## 1. Introduction

**a.** Purpose

- The purpose of this document is to show not only the software and hardware requirements of the system but also the functionality and scope of this system will also be described in this SRS document.

- The SRS document assures that the project developer and the team has really understood the requirement of the project properly. It also provides that the team will develop the functionality of the project which has been detailed.

- This document helps in breaking down of activities and deliverable into smaller components so to make the participant understand the concept of the system clearly.

- This document gives an opportunities in planning ahead which can save both time and cost involved.

The name of the system is **"Detap"** which enables a user to get an appointment in a easy way without having to stand in queue. The purpose of this system are:

- To develop a system which can help patients to make appointment with the doctors in an easier way despite having to stand in queue.

- To have a better schedule of a doctors.

- To save time.

**Scope:**

**a) User Scope:** The user scope is  Gyalpozhing BHU, Mongar.

**b) System scope**

**Receptionist:** Receptionist needs to login with username and password and in the his home screen, he can see the basic functionalities of Receptionist. He can view the details of  registered doctors and patients. He can also view the patient's request and doctors requests.

**Doctor:** Doctor need to be registered by giving the necessary details like category, timing, etc. After registering he need to log in and in the home screen he can view the basic functionalities. He can view the patient request forwarded from admin.

**Patient:** The patient needs to be registered and log in. After entering into his/her home page he/she can search the doctor availability by giving its category(Department name) or doctor's name.  Patient can view the doctor's details and they can view their own status.

**Home:** Any user can view the first page without logging into the system. There are three division namely; Doctor, Patient and Receptionist. The Receptionist can view or have access to the overall profile (details) of doctors, and patients.

**Registration:** The patients have to register with the system to check in for the follow up checkups with the respective disease doctor of the department. The doctors and receptionist should also register with the system.

**Login:** The doctors, receptionist, and patient will login after registration with the system in order to make changes. They have to login with the valid details.

**CRUD:** The Receptionist will perform crud operation on the overall details of doctors' schedule and department which are not in use. They can also perform crud operation on patients details.

**Search:** The patients can search for the doctors where they would have to visit based on the health issues suffered, of different department and fix the appointment accordingly.

**Status:** In this, patient can view the status of whether the booking is confirmed or not.

**Appointment:** In this, patient have to fill up the form where it includes their information like name, medical report and they also have to mention the name of a doctor.

## 2. Requirements

### A. Functional Requirement

**Receptionist:** Receptionist needs to login with username and password and in the his home screen, he can see the basic functionalities of Receptionist. He can view the details of registered doctors and patients. He can also view the patient's request and doctors requests.

**Doctor:** Doctor need to be registered by giving the necessary details like category, timing, etc. After registering he need to log in and in the home screen he can view the basic functionalities. He can view the patient request forwarded from admin.

**Patient:** The patient needs to be registered and log in. After entering into his/her home page he/she can search the doctor availability by giving its category(Department name) or doctor's name. Patient can view the doctor's details and they can view their own status.

**Home:** Any user can view the first page without logging into the system. There are three division namely; Doctor, Patient and Receptionist. The Receptionist can view or have access to the overall profile (details) of doctors, and patients.

**Registration:** The patients have to register with the system to check in for the follow up checkups with the respective disease doctor of the department. The doctors and receptionist should also register with the system.

**Login:** The doctors, receptionist, and patient will login after registration with the system in order to make changes. They have to login with the valid details.

**CRUD:** The Receptionist will perform crud operation on the overall details of doctors' schedule and department which are not in use. They can also perform crud operation on patients details.

**Search:** The patients can search for the doctors where they would have to visit based on the health issues suffered, of different department and fix the appointment accordingly.

**Status:** In this, patient can view the status of whether the booking is confirmed or not.

**Appointment:** In this, patient have to fill up the form where it includes their information like name, medical report and they also have to mention the name of a doctor.

**B. Non-functional Requirements**

    1) *Performance Requirements:*

- *User-interface*: The web page generate by this system shall be displayed in 5 second without taking more time.

- *Respond time*: The system should respond faster when the information is to be displayed.

- *Confirmation*: Confirmation needs to be done for the users to get an appointment with the doctors.

    *2) Security Requirements:*

- *Patients identification:* The system requires the patients to be identified by its valid phone number.

- *Login ID*: The Users who need to be login with the system needs to input valid Username and password.

- *Registration*: The users while registering they need to use email in order to get the code.

    *3) Software Quality Attributes*

- *Availability :*the system is always available to any user.

- *Correctness:* The system shall give the right information and procedures for the patients to work on.

- *Stability :* the display/output of the system shall be stable and should not be changing time to time with every input.

- *Re-usability:* the system can be used again and again without any alteration.

- *Flexibility:* the system should be flexible for necessary update with system anytime.

**C. Software Requirements:**

➢ Operating System: Windows or Ubuntu

➢ Android SDK: Android SDK version 16

➢ Android Studio: Version 5.0 and above

➢ Java version: JDK 8

➢ Database: Firebase version 19.2.1

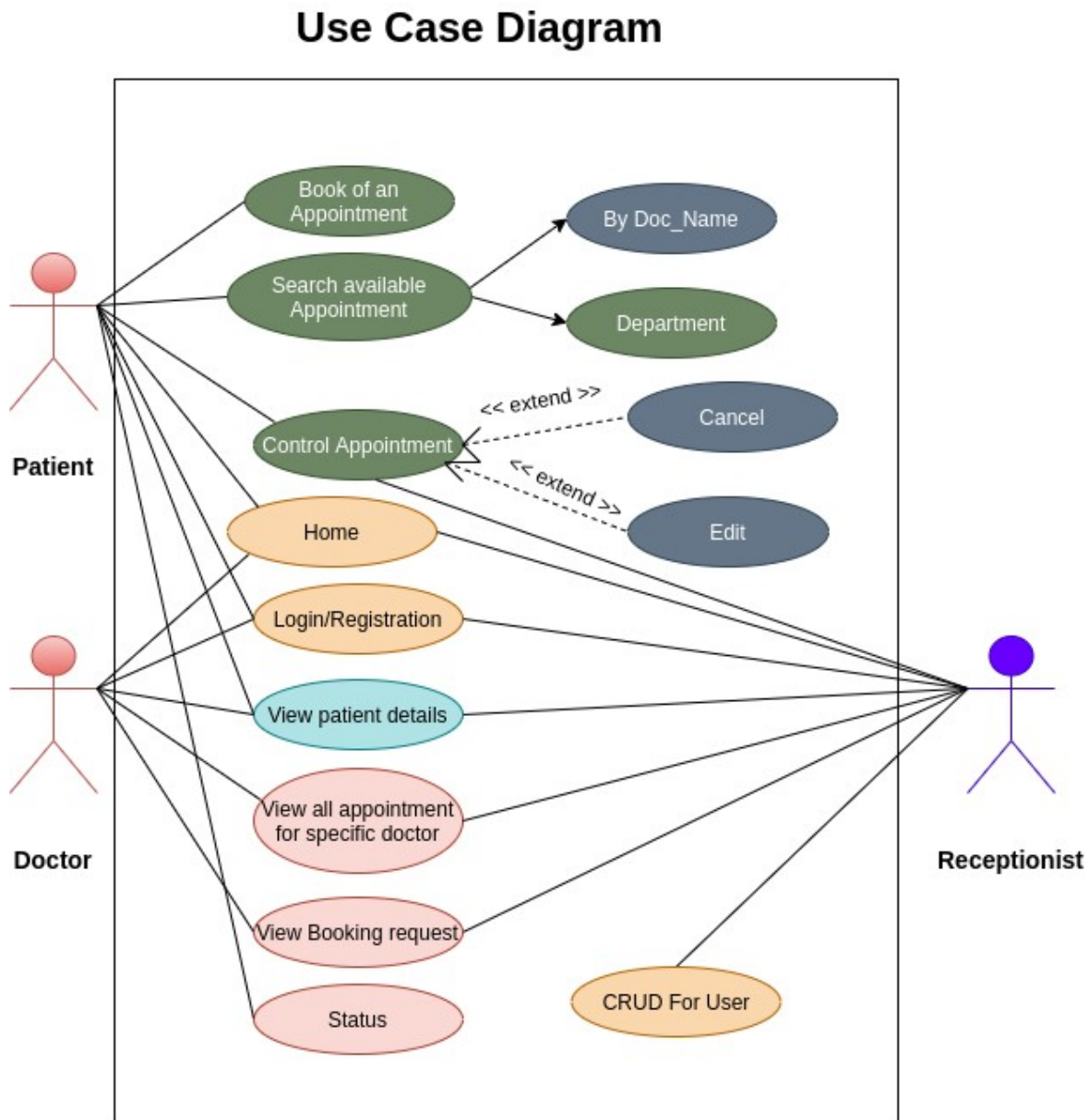**D. Hardware Requirement**

**1. Developers Requirement**

➢ RAM: 4GB and above

➢ Processors: Core i3 and above

➢ Disk space: 4GB minimum(8GB recommended)

**2. User's Requirement**

➢ Smartphone

**3. System Design**
**A. USE CASE DIAGRAM**

## Use Case Diagram



The usecase diagram shows some of the basic function of the system. There are three actor where two are primary actor Patient, Doctor, and other one is secondary actor Receptionist. All three actors needs to register and login before going into the system. While logging in, they have to enter the valid information. The user can book the appointment based on their sickness or diseases. They can view their own status like whether their appointment is approved or rejected. They can also view the doctor's details. In this, doctor can view the patient's details who has been registered for the appointment. The receptionist can view the details of both the patients and doctors and they can manage the registered user. The receptionist can also view the booking details.

**B. ERD(Entity Relationship Diagram)**
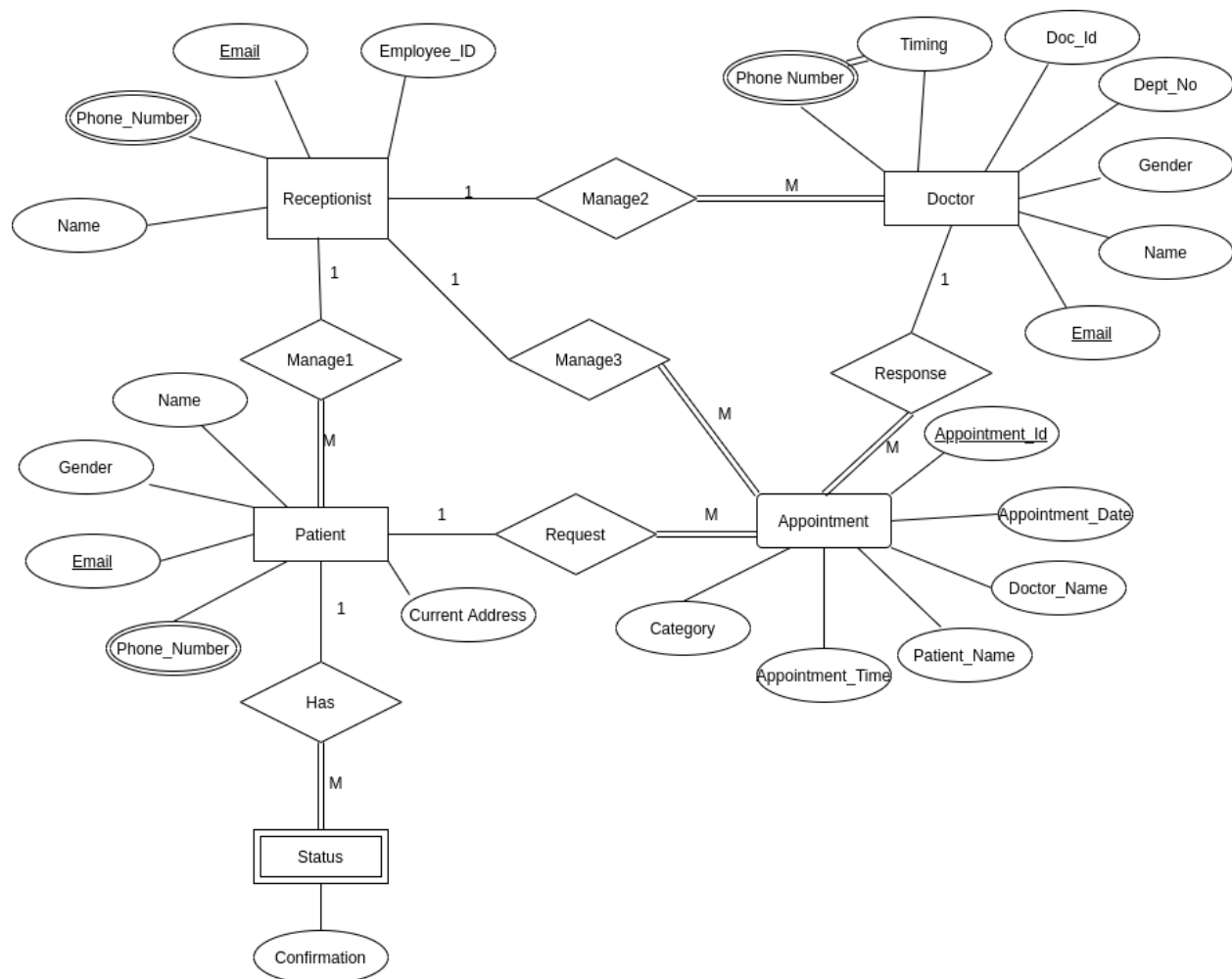


There are four strong entity with some attributes that is Patients with name, email, current address, phone number and gender attributes, Receptionist entity with name, email, employee_id and phone number, Doctor entity with name, gender, email, doctor_id, dept_no, timing and phone number attributes and Appointment entity with appointment_id, patient name, doctor name, appointment date, appointment time and category attributes. In this four entity, the primary key is email attributes. There is a one week entity that is Status with confirmation attribute.

**Relationship**

➢ The relationship between Receptionist and patient is manage1 where the receptionist can perform crud operation on patients details. It's cardinality ratio is 1 to many because one receptionist will have many patient and many patient will have one receptionist.

In this, patients takes total participation as every patient has to be managed by at least one receptionist where as not every receptionist has to managed the patients.

➢ The relationship between Receptionist and Doctor is manage2 where receptionist can also perform crud operation on doctors details and schedule. It's cardinality ratio is 1 to many because one receptionist will have many Doctors and many Doctors will have one receptionist. In this, doctors takes total participation as every doctors has to be managed by at least one receptionist where as not every receptionist has to managed the doctors.

➢ The relationship between Receptionist and Appointment is manage3 where receptionist can manage the Appointment where they will whether the patient's appointment is approved or cancel. It's cardinality ratio is 1 to many because one receptionist will have many appointment and many appointment will have one receptionist. In this, Appointment takes total participation as every appointment has to be managed by at least one receptionist where as not every receptionist has to managed the appointment.

➢ The relationship between Patient and Appointment is requests. Its cardinality ratio is 1 to many because one patient will have many appointment and many appointment will have one patient. In this, appointment takes total participation as every appointment is requested by at least one patients where as not every patient will request for the appointment.

➢ The relationship between Doctor and  Appointment is response. It's cardinality ratio is 1 to many because one doctor will have many appointments and many appointment will have one doctor. In this, appointment takes total participation as every appointment has to accept or reject by at least one doctor whereas not every doctors has to respond for the appointment.

➢ The relationship between patient and status is has. Its cardinality ratio is 1 to many where one patient can have many status and many status can have one patients. The status take total participation as each status must has at least one patient where as not every patient will have to patient.

**C. Relational Schema Diagram**
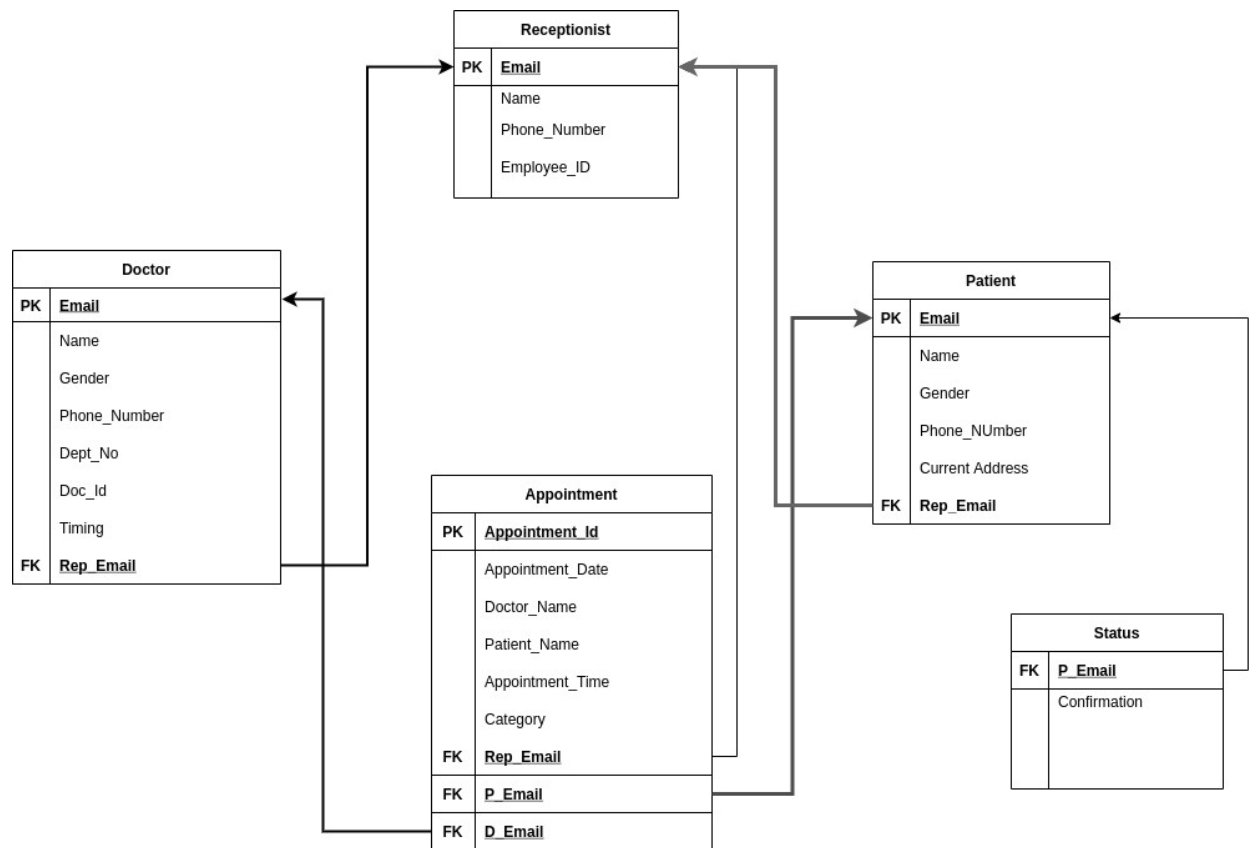


This schema was derived from the ER diagram. The arrow represent from where that particular foreign key is brought, Here, the doctor entity has a Rep_email(foreign key) which is brought from receptionist entity. The appointment entity have three foreign key that is Rep_email from receptionist, P_email from patient and D_email from doctor entity. The Doctor have Rep_email which is brought as a foreign key from receptionist entity. The Status entity has P_email as a foreign key which is brought from patient entity.
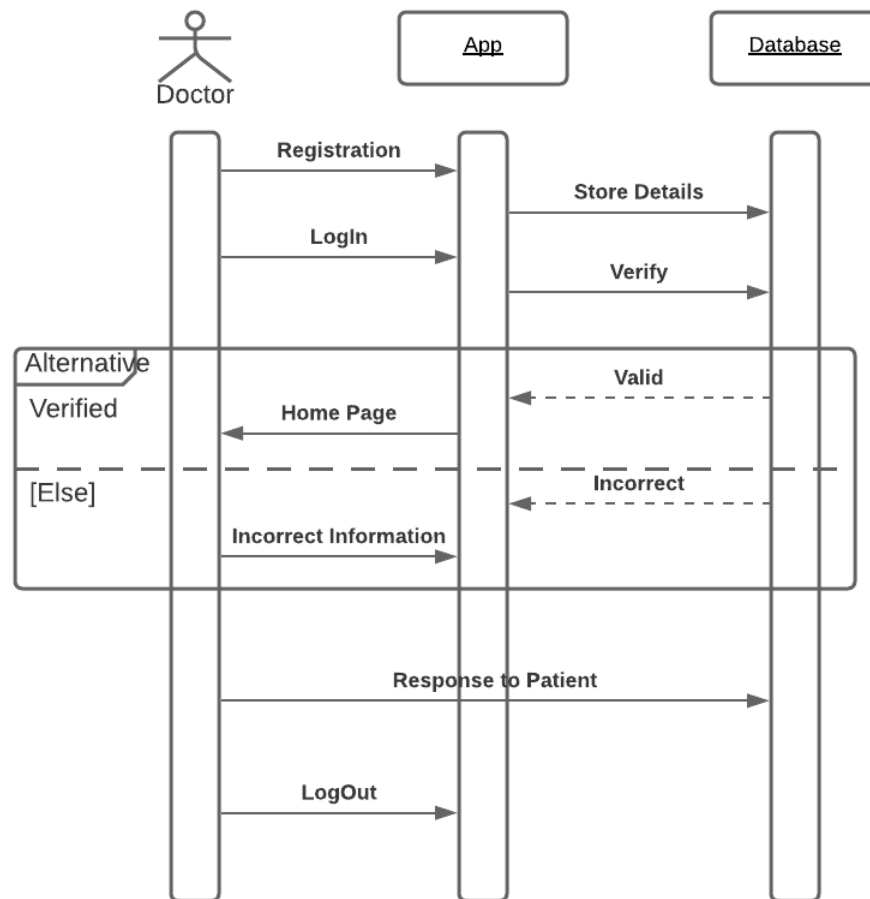
**E. Sequence Diagram**
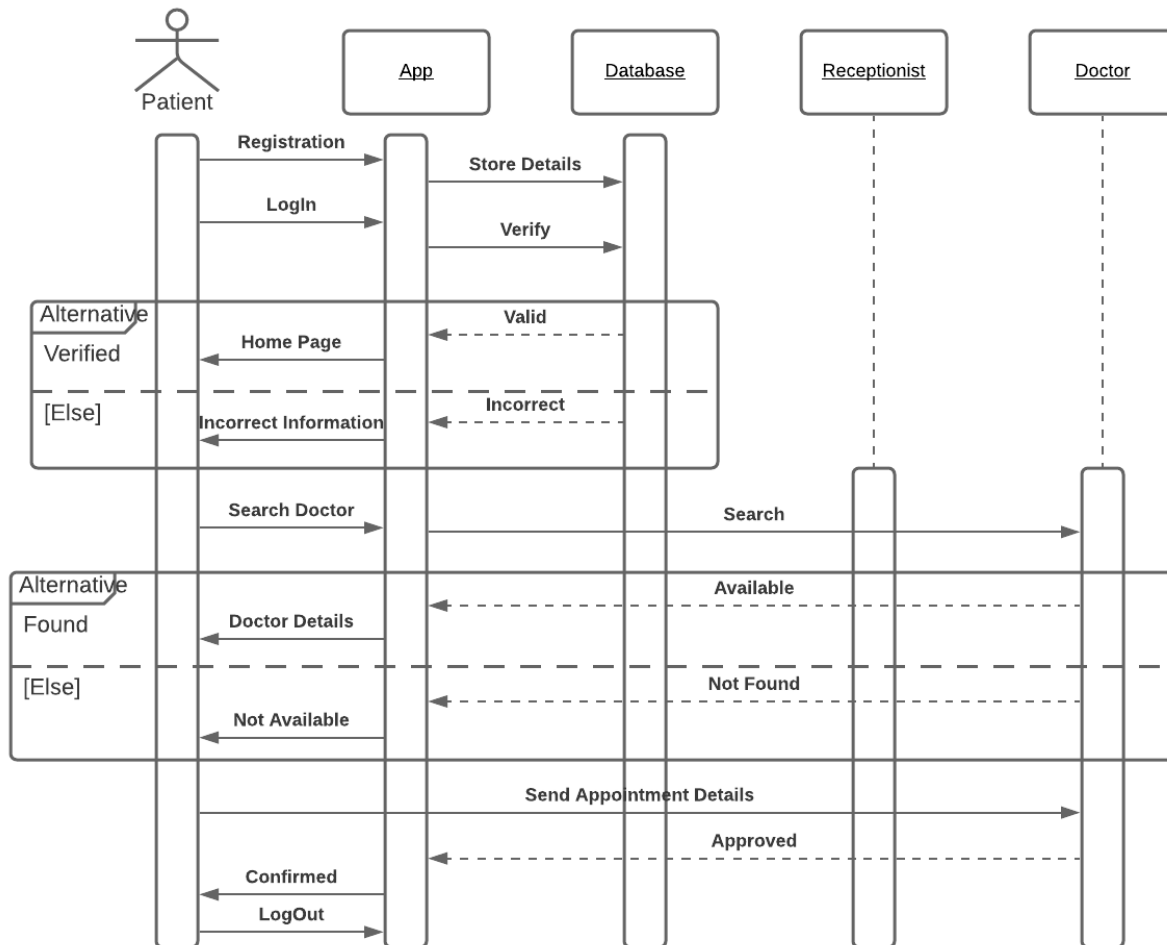


This is the sequence diagrams of the system where it will show the sequence of the process on how this system will work. Firstly, the patient has to register and their details will stored in the database. Then, the patient has to login to the system and the database will verify whether they have registered or not and whether they have entered the correct information or not. If they have entered the valid information, the app will display the home page but if patient entered an incorrect information then the patient has to login again with the valid details. After successful login into the system, the patient can search doctor availability by name or category. After searching that they can send their appointment form filling with required information to the system. The system will send it to receptionist and doctor. The doctor has to respond to that particular appoint through receptionist and receptionist will notify to the patient whether they have approved or not. If the patient want to logout from the system, then they can logout from the system.

**F: Sequence Diagram for Receptionist**
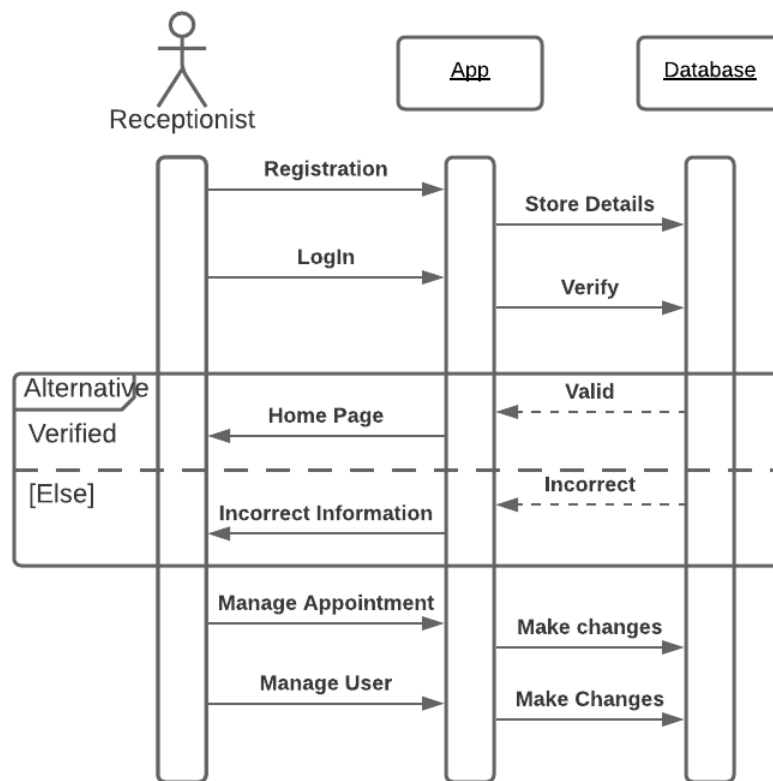


This is the sequence diagrams for receptionist. Firstly, the receptionist has to register and their details will stored in the database. Then, the receptionist has to login to the system and the database will verify whether they have registered or not and whether they have entered the correct information or not. If they have entered the valid information, the app will display the their home page but if receptionist enter the incorrect information then the receptionist has to login again with the valid details. After successful logging into the system, the receptionist can manage the details of both the patient and doctor. They can also manage the appointment.

**F: Sequence Diagram for Doctor**



This is the sequence diagrams for doctor. Firstly, the doctor have to register and their details will stored in the database. Then, the doctor have to login to the system and the database will verify whether they have registered or not and whether they have entered the correct information or not. If they have entered the valid information, the app will display the their home page but if doctor enter the incorrect information then the doctor has to login again with the valid details. After successful logging into the system, the doctor can view the patient details those who have been requested for an appointment and they can send the approval through receptionist. If he/she wants to logout from this system, then they can logout from the system.