SOFTWARE REQUIREMENT SPECIFICATIONS

Project Topic: [Waitless]

Submitted by: [Thinley Wangmo]

12190089

1. Introduction

a. Purpose

The purpose of this document is to identify the functionality that is needed by the system in order to fulfill the requirements. It includes the detailed description of the application that will be developed.

AIM:

- To develop Doctor's Appointment android mobile application also known as waitless app.

OBJECTIVES:

- The following are the objectives in developing Doctor's Appointment app to work in android phones:
    1. To design and develop an easy to use and understandable mobile application.
    2. To develop an android application to meet with the referred doctors directly without waiting in lines.
    3. To help people from rural areas where there is only BHU to visit the main hospital or regional referral hospital by making appointment and visiting in the date requested by the patient there by reducing efforts and time eventually.

b. Scope

System Scope

To develop a mobile application with the following features:

1) Online based
2) Registration
3) Login
4) Details of patients
5) Appointment date and time

User Scope

The main scope for this project is to build a user-friendly app that will help patients to get an appointment with the doctor at the requested date without physically going to the hospital and waiting in a line for long time to see a doctor.

2. Requirements

a. Functional Requirements

i. Describe each feature of your application

The Doctor's Appointment app will consist of:

1) Registration – To register with the app the patient which is the user will be required to register with their email.
2) Login – The patient who have been registered can login using their email and valid password.
3) Details of patients – The patient will be required to provide personal information. While providing the details, the patient must include his or her name, phone number, age, gender, and symptoms that includes the current condition of the patient. The patient should also provide the medical history referring to the diseases that the patient has experienced by entering the name of a particular disease.
4) Appointment date and time – After entering the information, the patient must enter a date and time for his/her appointment with the doctor. The receptionist will confirm this date and time with the information send by the patient. If the doctor is available at the requested time, the patient will be contacted by the receptionist, and a message will be sent to the patient indicating which chamber he or she must visit.

b. Non-functional requirements

Some of the non-functional requirements of this application are:

a. Security
   This application will be secure as the user will have to register first and then login using email id and password.
b. Portability and compatibility
   This application is portable as well as compatible as it can be supported in all the platforms of android versions.
c. Usability
   It is very easy to use as it is user friendly.
   The app can be easily portable to different versions of android and is independent of the size of any android phone and tablets.

c. Software Requirements

i. The technology used and version

1) Java version: Java SE jdk 8 and above.
2) Android Studio version 4 and above.
3) Android SDK-25 and above.
4) Operating System: Ubuntu and Windows.

5) Firebase
   - ✓ Firebase is a backend-as-a-service developed by Google for a real-time database which is used to store and synchronize data.

3. Hardware requirements

For Developer

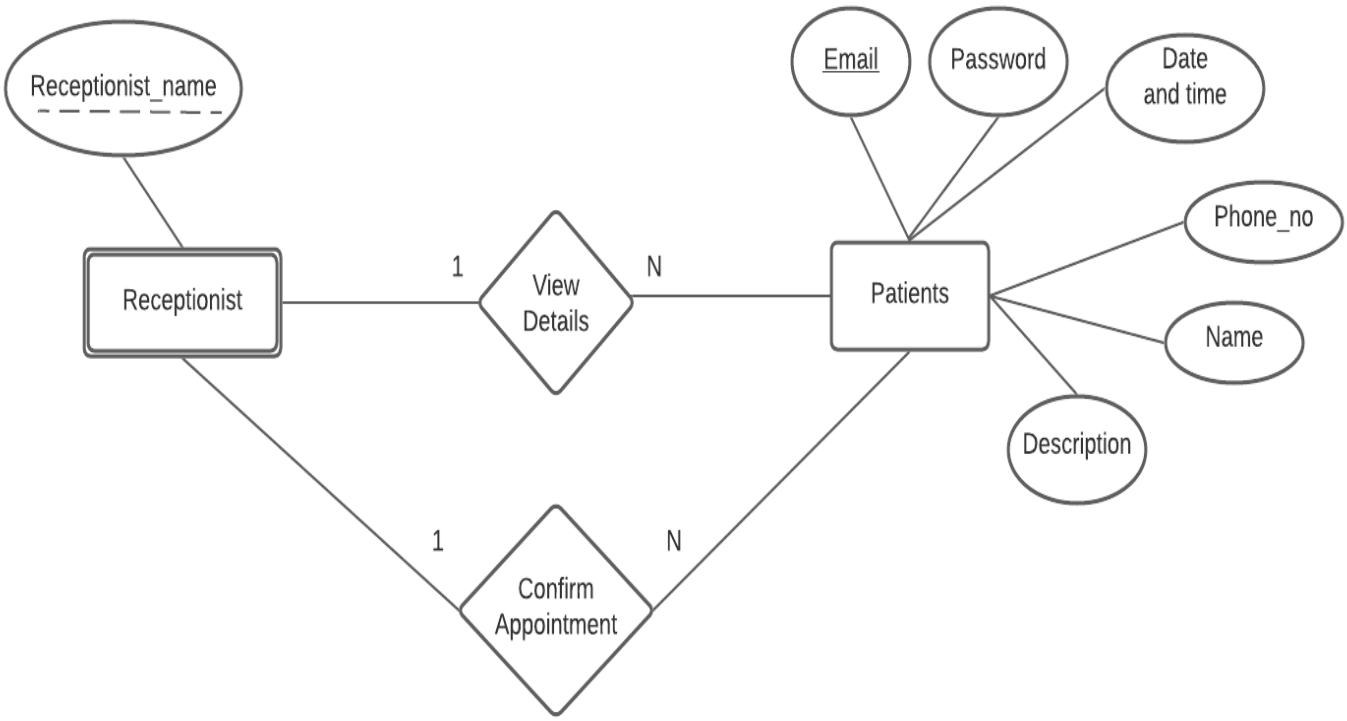
- RAM: 4-8 GB
- 2.00GHz*4 Processors
- Disk Capacity: 1.0 TB and above
- Android phone(Emulator)

For Users

- Android Phone.

4. System designs

a. ERD (Entity Relationship Diagram)



According to this project I have identified only two entities i.e patients and receptionist where receptionist is the weak entity. The entity receptionist consists only one partial attribute i.e

receptionist name. The attributes of the patient entity are – email, password, date and time, phone_no, name and description. Here one receptionist can view details of many patients and one receptionist can confirm appointment of many patients.

b. Relational Schema

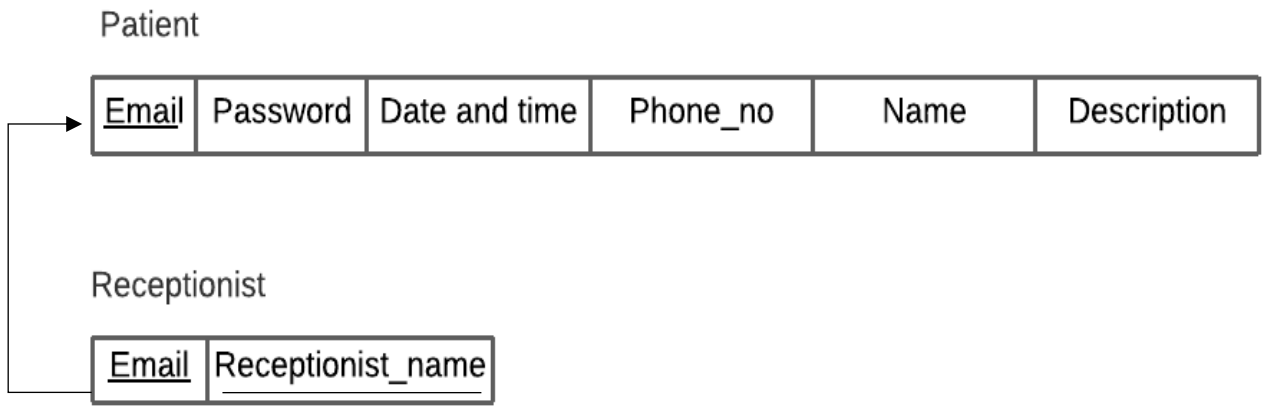


Table name:

- Patient
- Receptionist

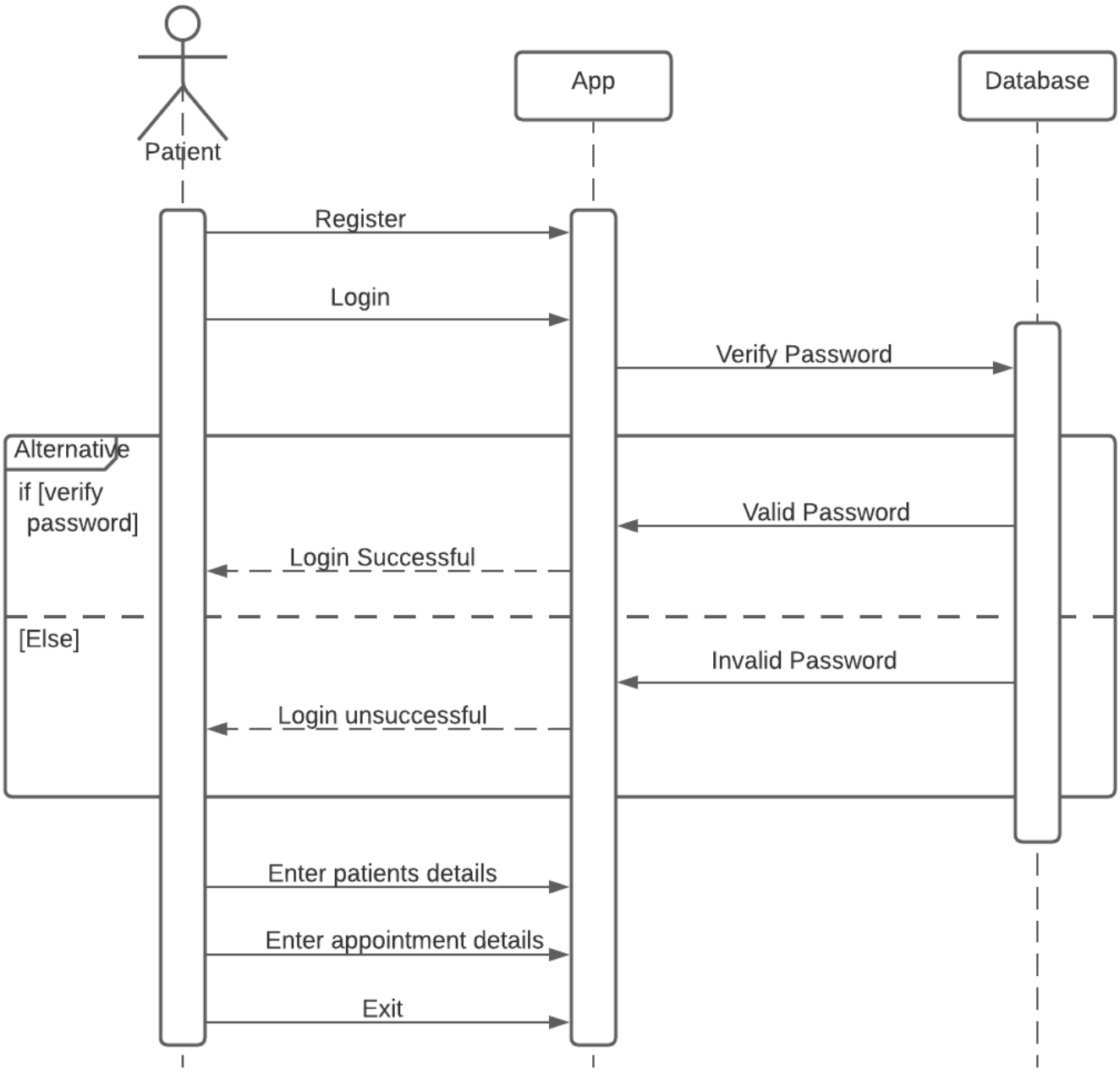
Primary Key

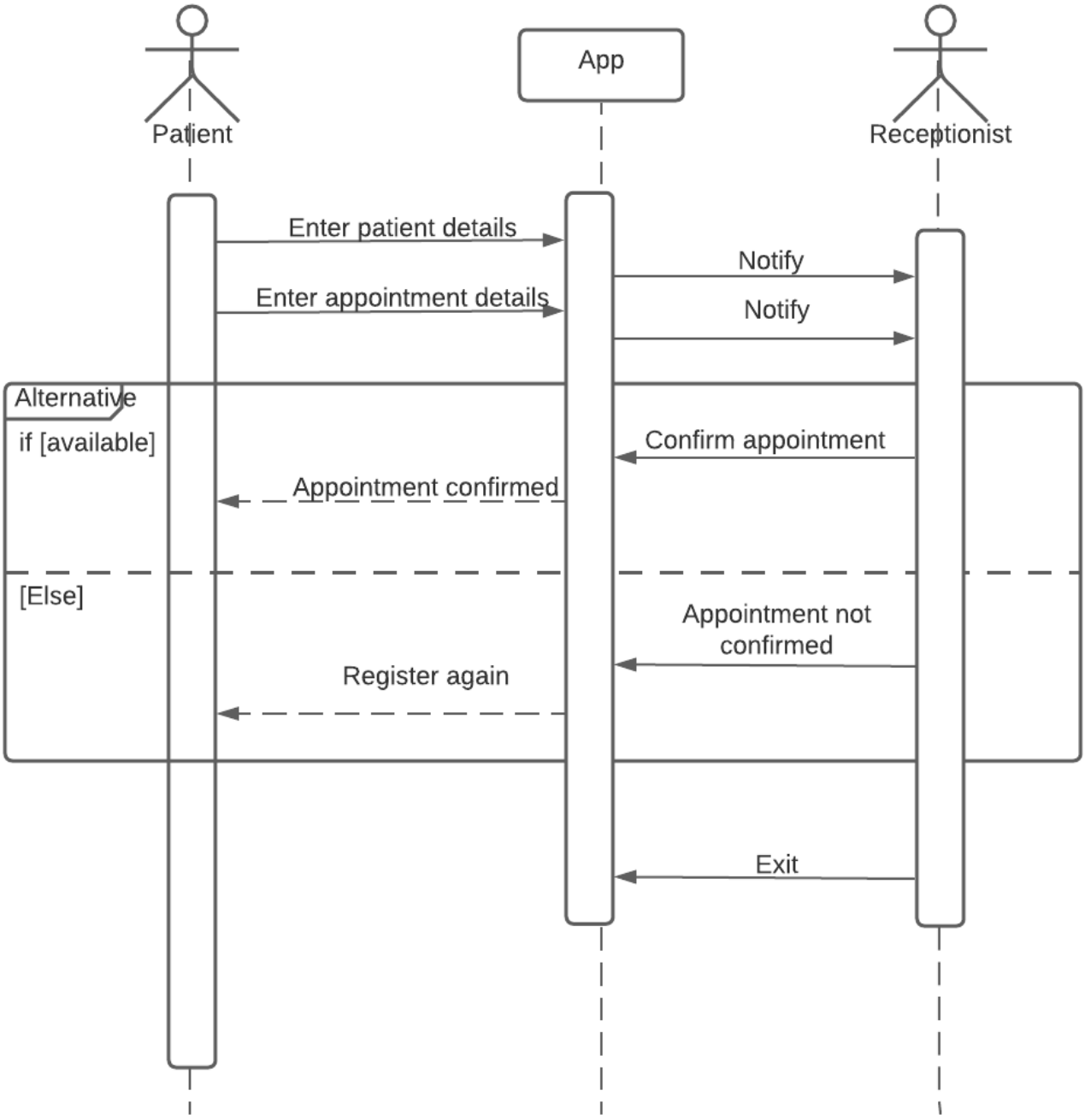
- Email

Foreign Key

- Email

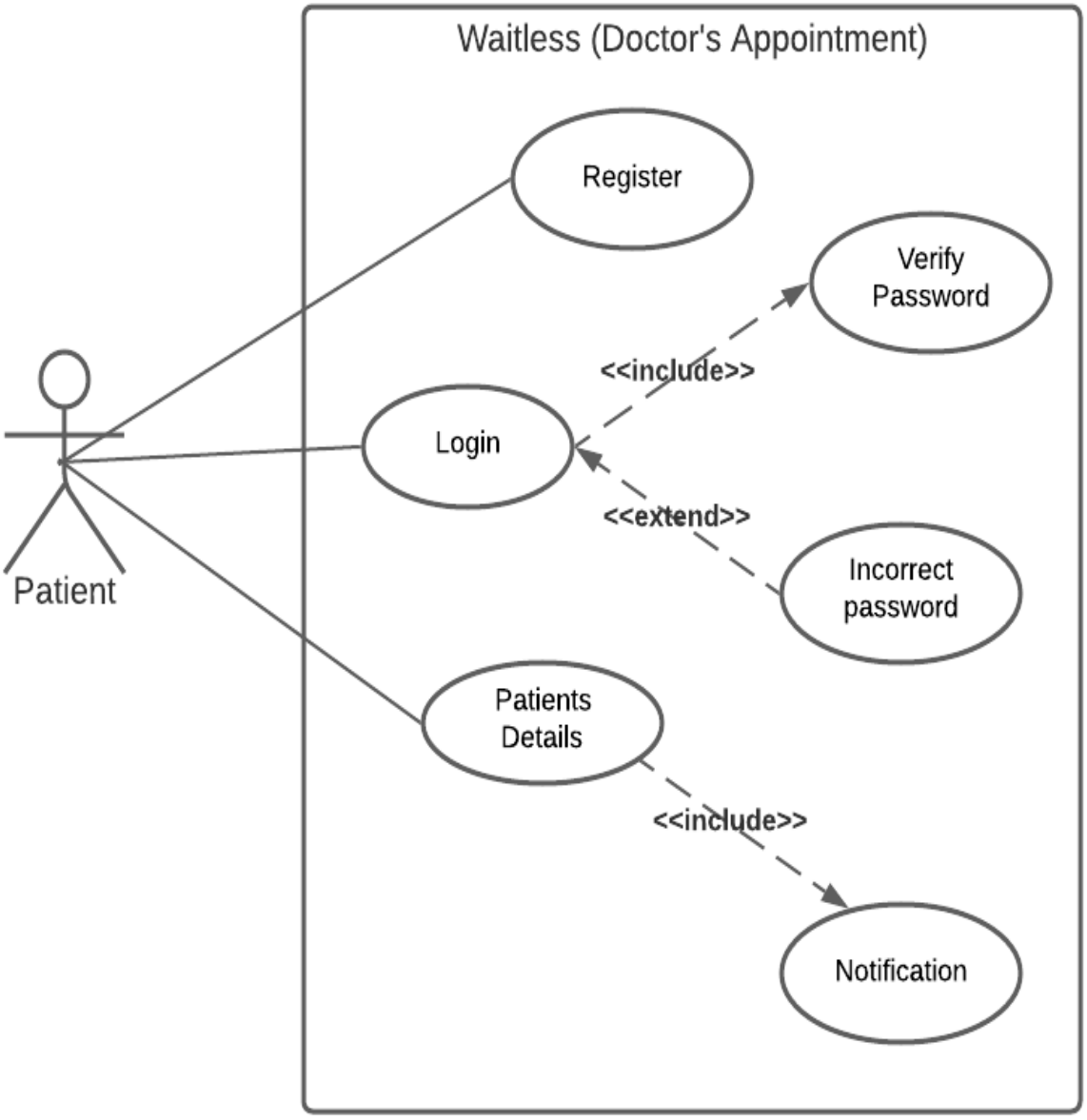
c. Sequence Diagram

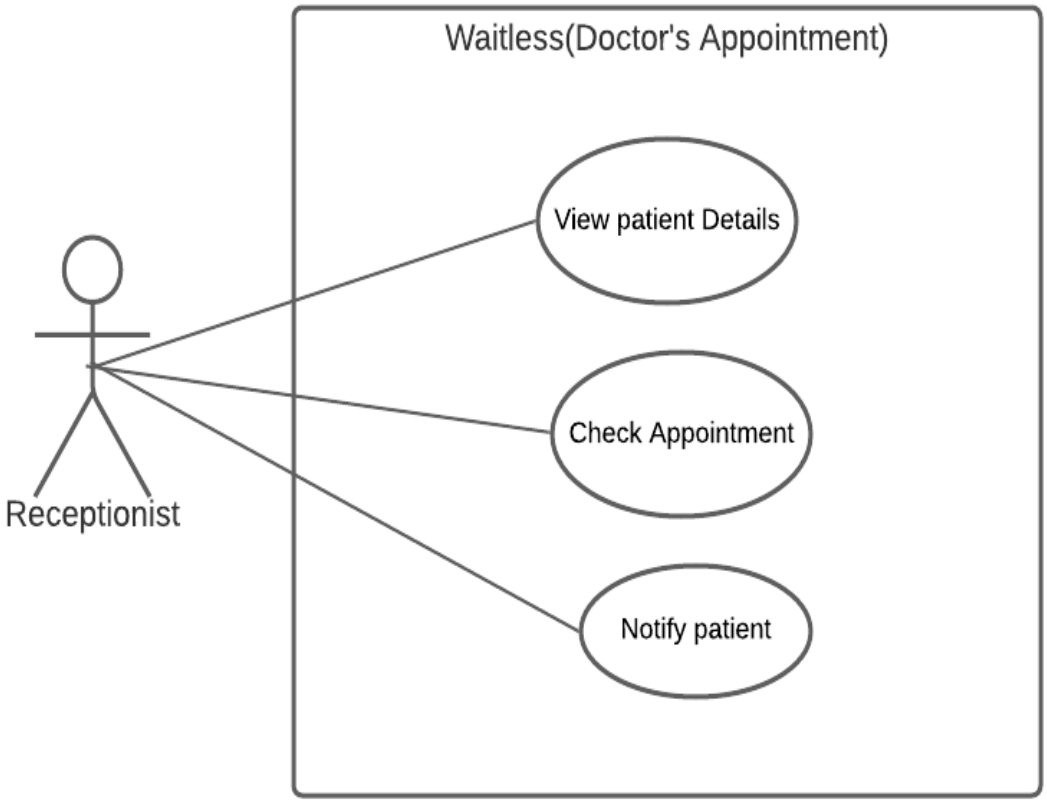



Patient
App
Receptionist
Enter patient details
Notify
Enter appointment details
Notify
Alternative
if [available]
Confirm appointment
Appointment confirmed
[Else]
Appointment not confirmed
Register again
Exit

Sequence Diagrams are interaction diagrams that details how the operations are carried out. They capture the interaction between the objects in context of a collaboration. The diagrams explains the detail logic behind Doctor's Appointment app.

d. Use case Diagram





For Patient

Primary actor – Patient

Functionality – register, login, patient details.

For Receptionist

Primary actor – Receptionist

Functionality – view patient details, check appointment, notify patient.