Software Engineering
UCS503
Project Overview

Submitted To: Dr. Vinod Bhalla

Submitted By:

Arnav Agarwal 101803005
Avinash Jindal 101803002
Neelanchi Mahajan 101853016
Stuti Jain 101853033
Karan Aggarwal 101983065

Table of Contents

1. Introduction..... 3
    1. Purpose..... ..3
    2. Scope..... 3
    3. Definition, acronyms and abbreviations..... 3
    4. References..... 4
    5. Overview..... 4
2. Overall Description..... 4
    1. Product Perspective..... 4
    2. Product Function.....4
    3. Data Flow Diagram..... ..5
        i. DFD Level 0 6
        ii. DFD Level 1.....7
        iii. DFD Level 2 8
3. Object Oriented analysis..... 9
    1. Use case Diagram.....9
        i. Doctor.....9
        ii. Patients.....10
        iii. Admin.....11
    2. Activity Diagram.....12
        i. Login Activity.....12
        ii. Doctor.....13
        iii. Patient.....14
4. User Characteristics..... 15
    1. Overview of Users..... 15
    2. Constraints..... 15
5. Specific Requirements..... 16
    1. External interfaces..... 16
    2. Functional Requirements.....16
    3. Performance Requirements..... 17
    4. Logical database requirements..... 17
    5. Software System Attributes.....17
6. Document approvers..... 18

1. Introduction

1.1 Purpose

It is an online web application which would cater the needs of

- Students
- Doctors

Keeping in view the ongoing difficulties experienced by students to book slots for dispensary during their busy schedule we are designing an online platform to solve this problem. The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online. This is a web based application that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. The already booked space will be marked and will not be available for anyone else for the specified time. This system also allows users to cancel their booking anytime.

1.2 Scope

Extending the scope to all particular people group: This platform can be used directly by all people in a particular region Realtime Availability and Booking of doctors is displayed. Taking an example of our University. The count of patients in medical center varies randomly during the day. In order to reduce the waiting time and centralization of the data, this would be a great success and all these problems will be solved. Even during sick leaves doctors can raise their leaves online and show their absence.

Extending the scope to all different services: Not keeping the services constrained to medical services, this platform can be used for maintenance issues of hostels by their wardens and hostel caretaker. This helps in keeping the check whether the issues are being resolved or not. Moreover, Repairs can be approved and managed on the daily basis instead of waiting for months.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Medical Web Application

An online platform to for a smart appointment booking system that provides patients or any user an easy way of booking doctor's appointment online.

1.3.2 Doctors

Doctor is a user who can check all the appointments fixed.Also he can add or cancel any appointemt at any point of time.

1.3.3 Patients

It is a set of users which can make appointments accordingly.

1.3.4 System

System refers to the existing medical portal or a new medical portal.

1.4 References

1. Udemy – WebDeveloper Bootcamp
2. ColtStelle/Linkedin
3. ColtStelle/Github
4. Tutorials Point SRS document
5. IEEE 830 Template

1.5 Overview

This is a working document and, as such, is subject to change. In its initial form, it is incomplete by definition, and will require continuing refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of the University Academic Portal.

2. Overall Description

2.1 Perspective

Health plays an intrinsic role in the development of nations and we have implemented a number of strategies that addresses and identifies people's health needs. We believe in community engagement and provides its customers more than just specific health treatments. Keeping this in mind, this website brings to you a solution that provides you with a greater accessibility to your healthcare needs. It is an appointment scheduler to facilitate seeking confirming and fulfilling appointments for doctor consultations, health checks and diagnostic services at dispensary through the website or by calling the given numbers. It is a patient-centric service that gives patients the freedom to consult a doctor at his convenience from anywhere, anytime. It's easy and interactive.

2.2 Project Functions

2.2.1 Overview

Patients : Patients are the primary consumers of an Appointment System. They are booking appointment based on the availability of doctors.

Doctors : Doctors are the primary content administrators of an Appointment System.They are managing appointments and may cancel or book an appointment on their availability,

2.2.2 Hardware Interface

a) Server side

The web application will be hosted on a link using Heroku app.

b) Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus.
Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

2.2.3 Software Interface

a) Server side : Node.js will accept all requests from the client and forward it accordingly.A database will be hosted centrally using mongo DB.

b) Client side

An OS which is capable of running a modern web browser which supports JavaScript and HTML5.

2.2.4 Communication Interfaces

The HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

2.3 Data Flow Diagrams

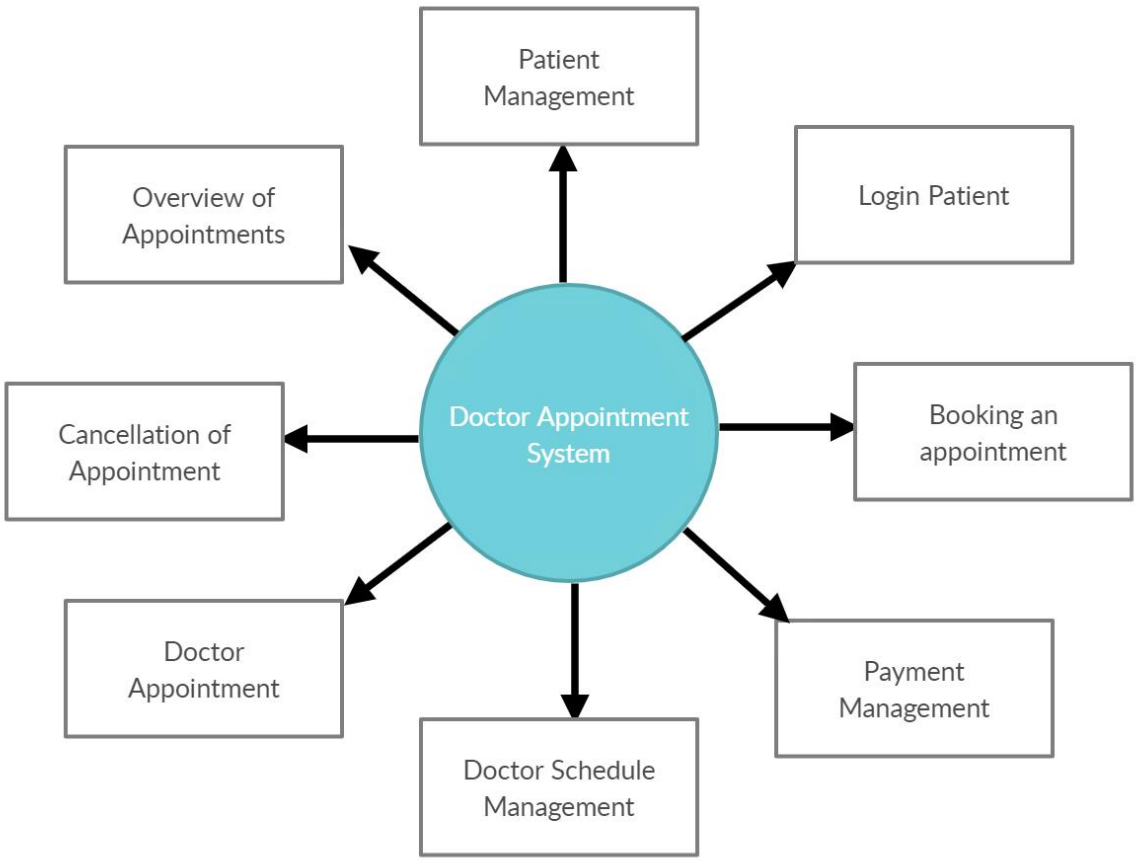
DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer.

The symbols in DFD depict the four components of data flow diagrams.

1. **External entity:** An outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors.
2. **Process:** Any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.
3. **Data store:** Files or repositories that hold information for later use, such as a database table or a membership form.
4. **Data flow:** The route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows.

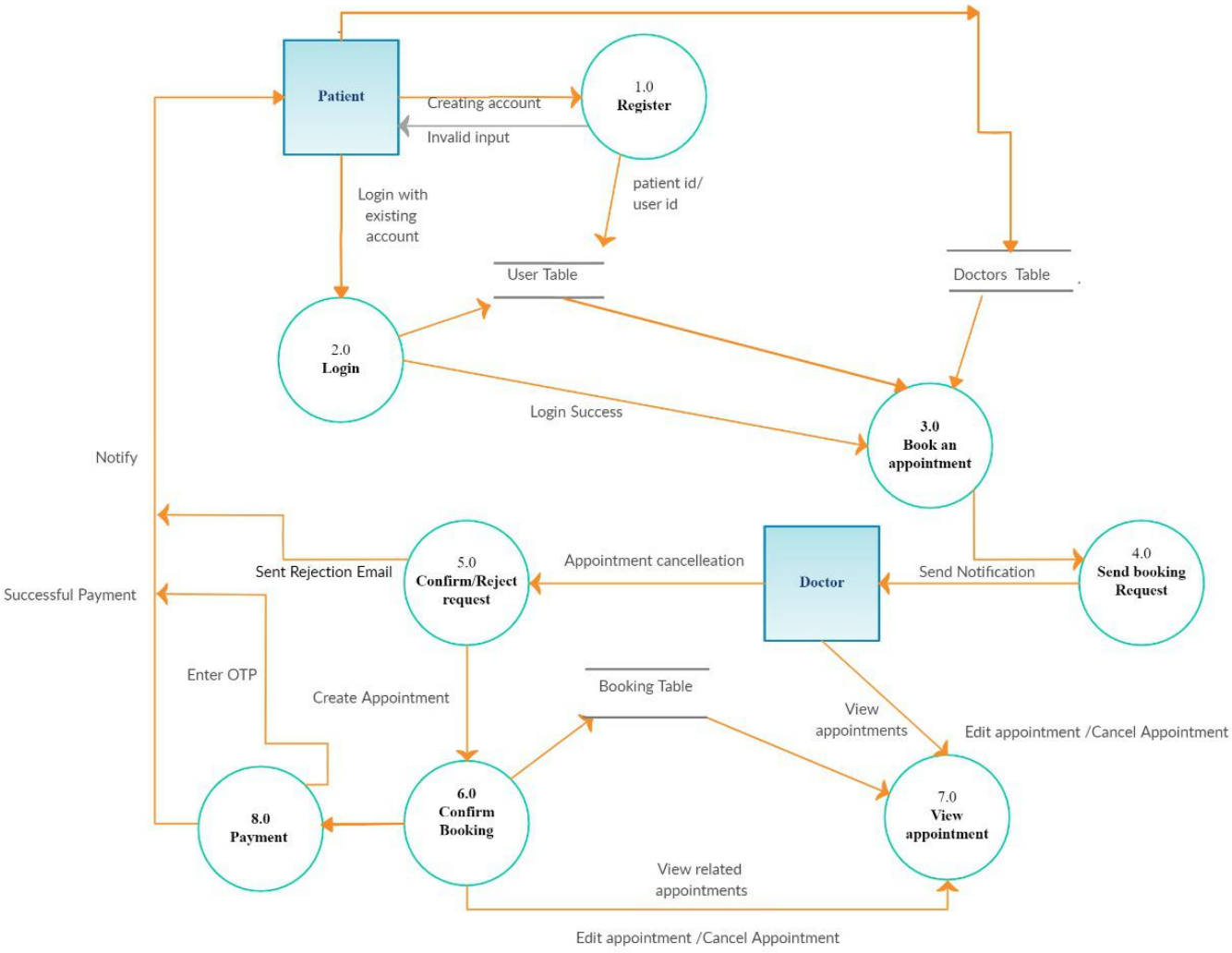
2.3.1 DFD LEVEL 0

It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represent the entire system as single bubble with input and output data indicated by incoming/outgoing arrows



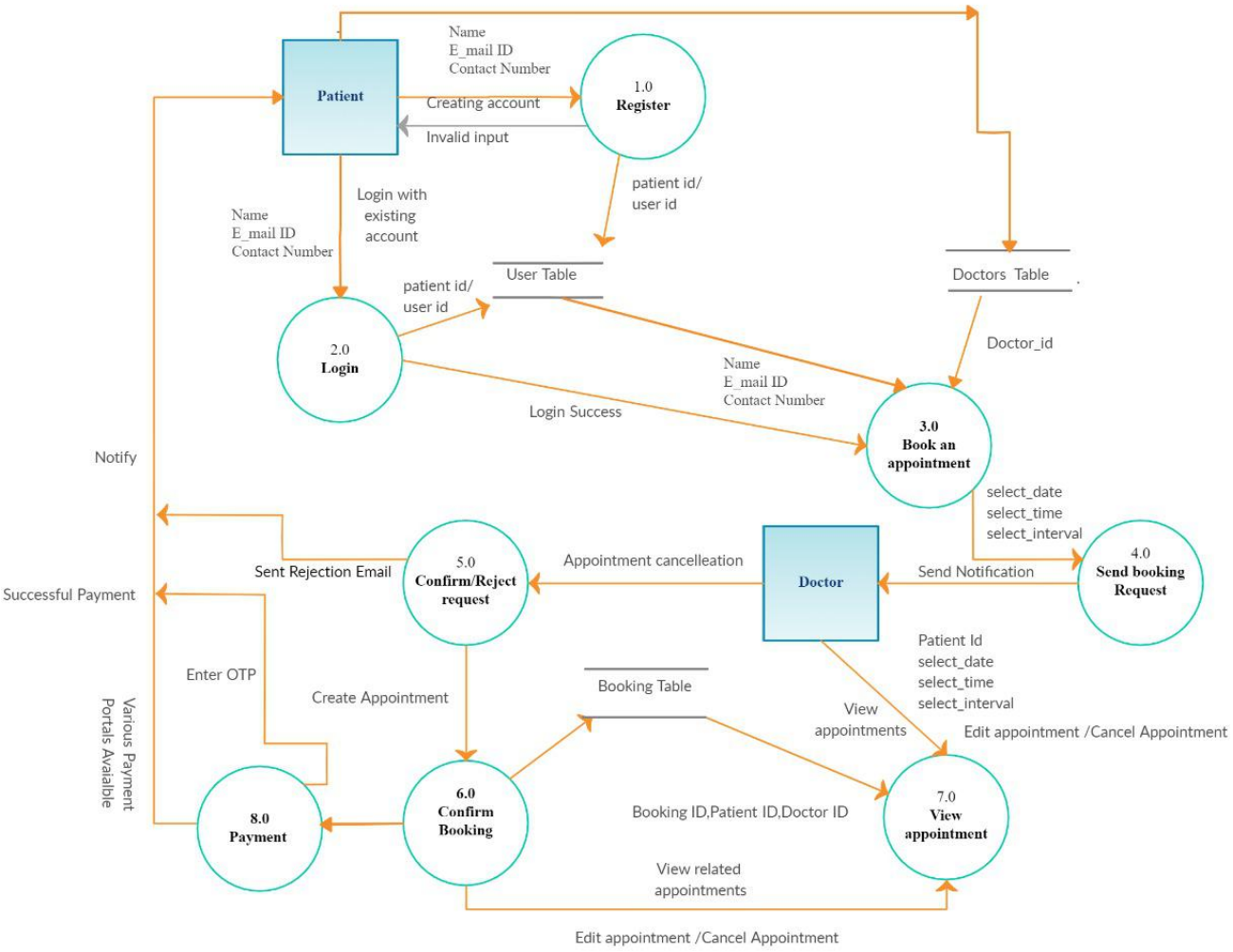
2.3.2 DFD LEVEL 1

In 1-level DFD, context diagram is decomposed into multiple bubbles/processes.in this level we highlight the main functions of the system and breakdown the high level process of 0-level DFD into subprocesses.



2.3.3 DFD LEVEL 2

2-level DFD goes one step deeper into parts of 1-level DFD.It can be used to plan or record the specific/necessary detail about the system's functioning.



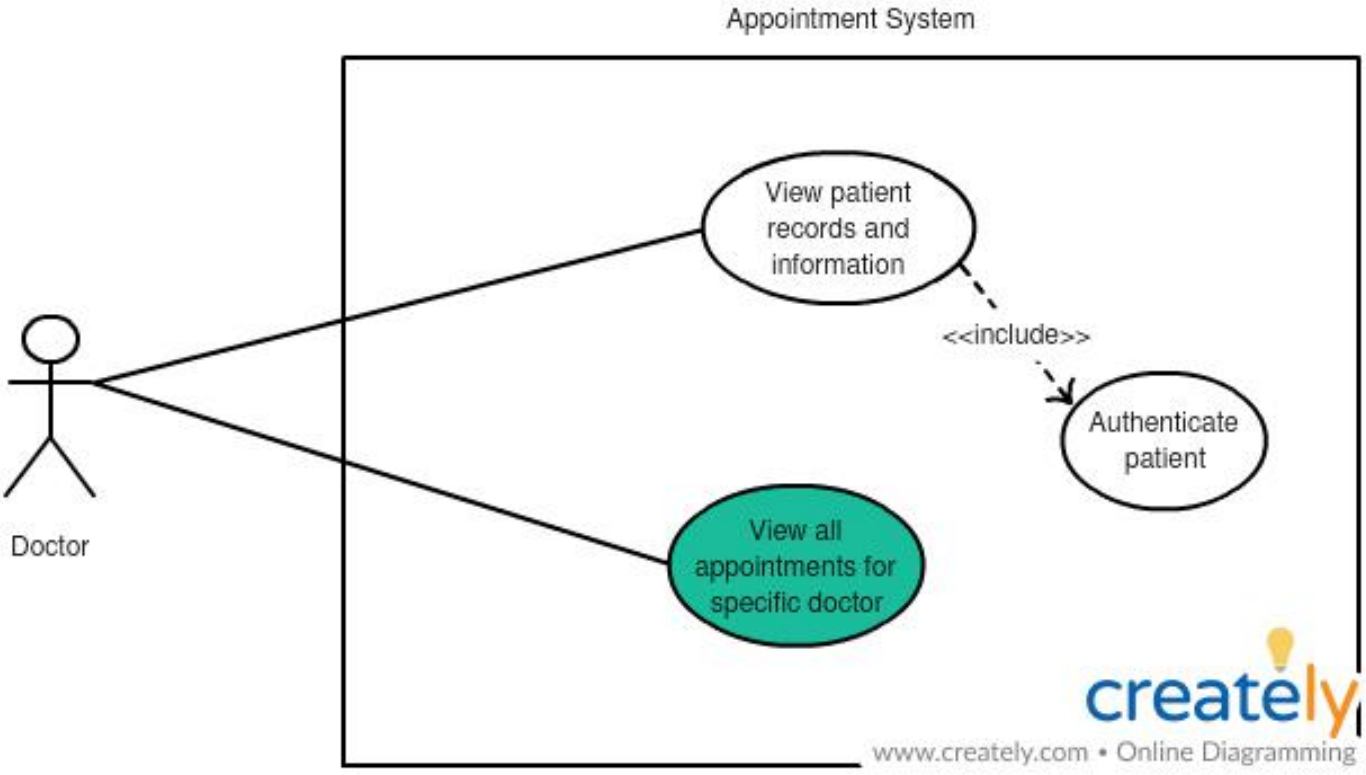
3. Object Oriented Analysis

3.1 Use-Case Diagrams

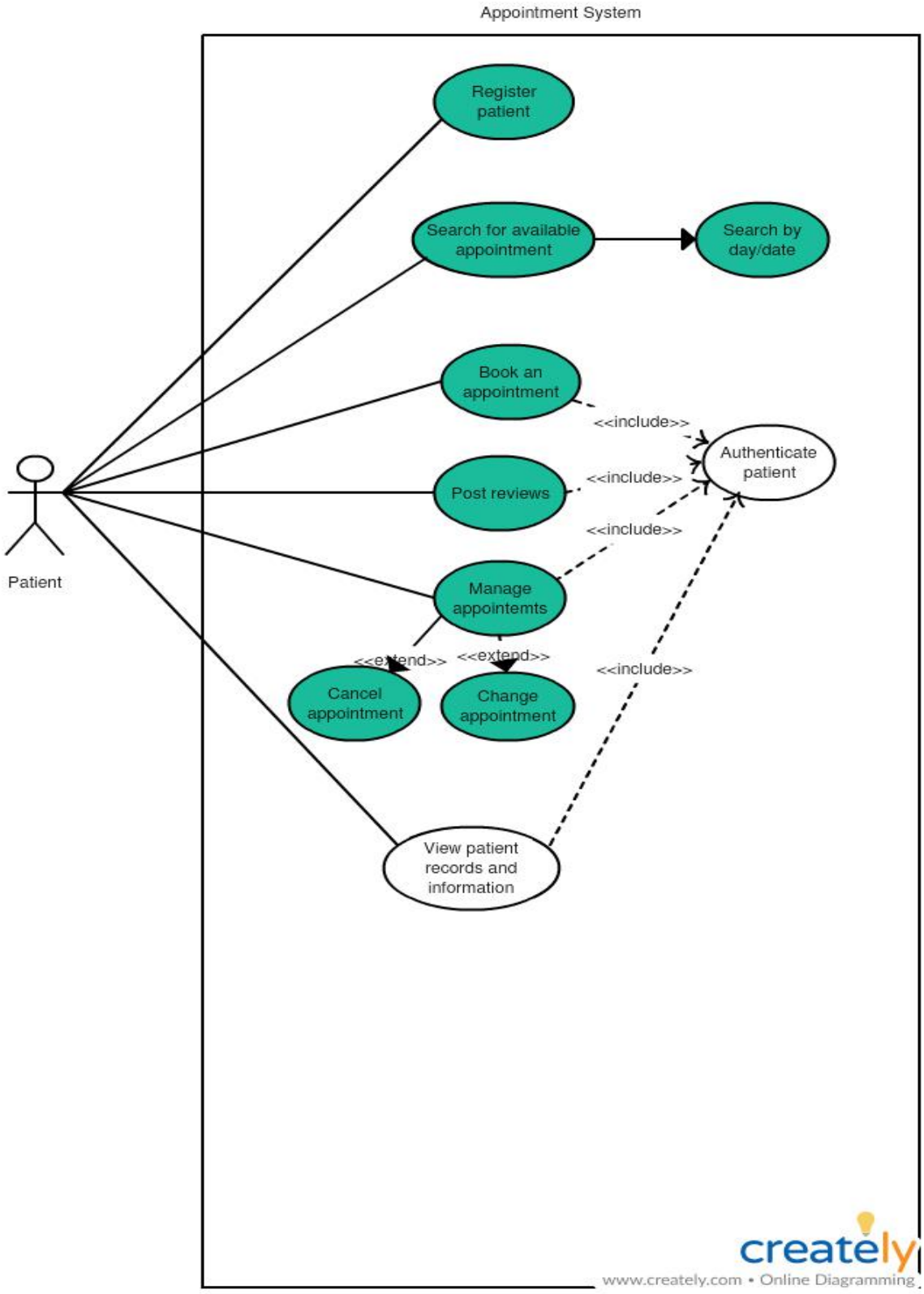
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

A use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

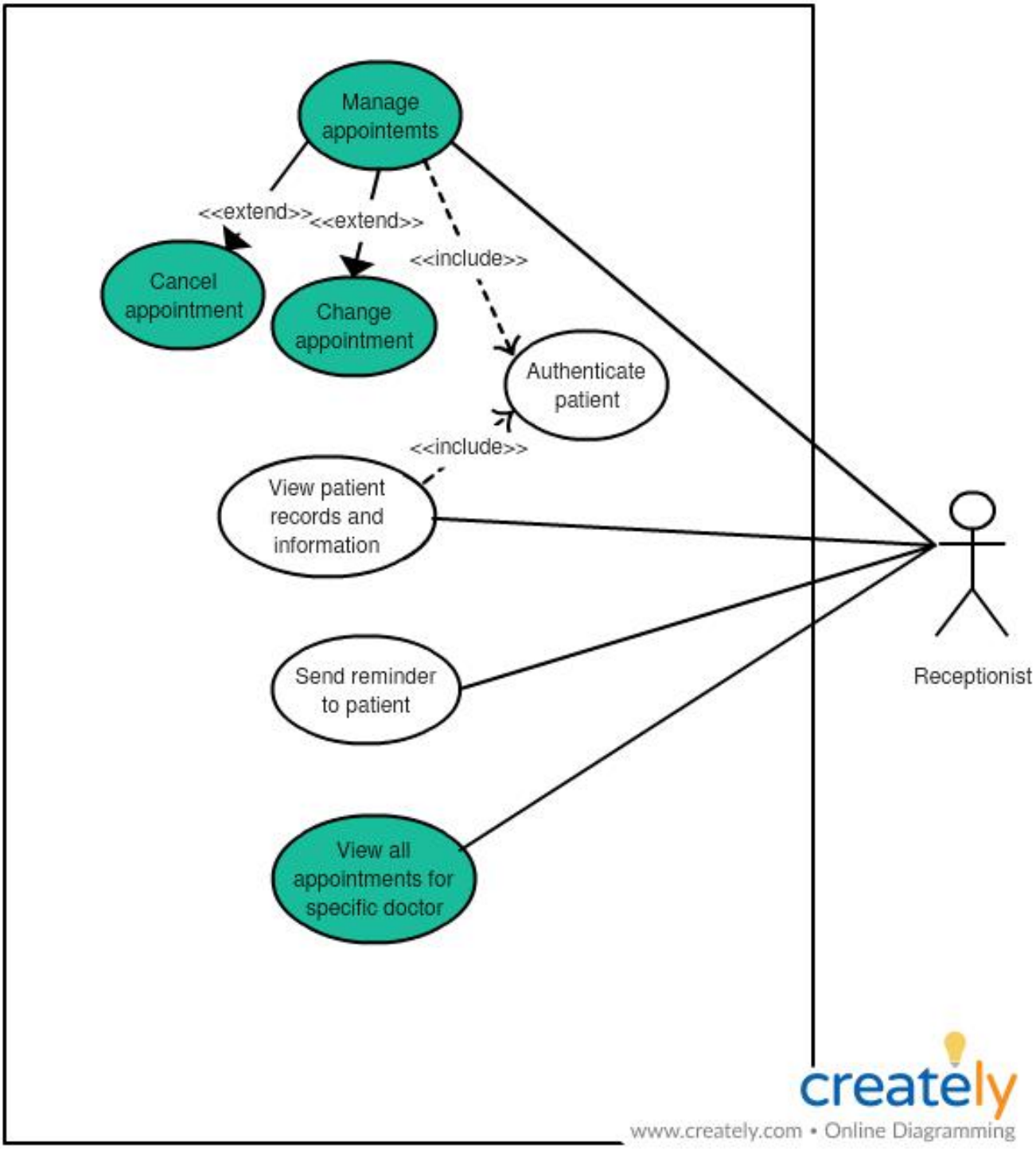
3.1.1 Doctors



3.1.2 Patients



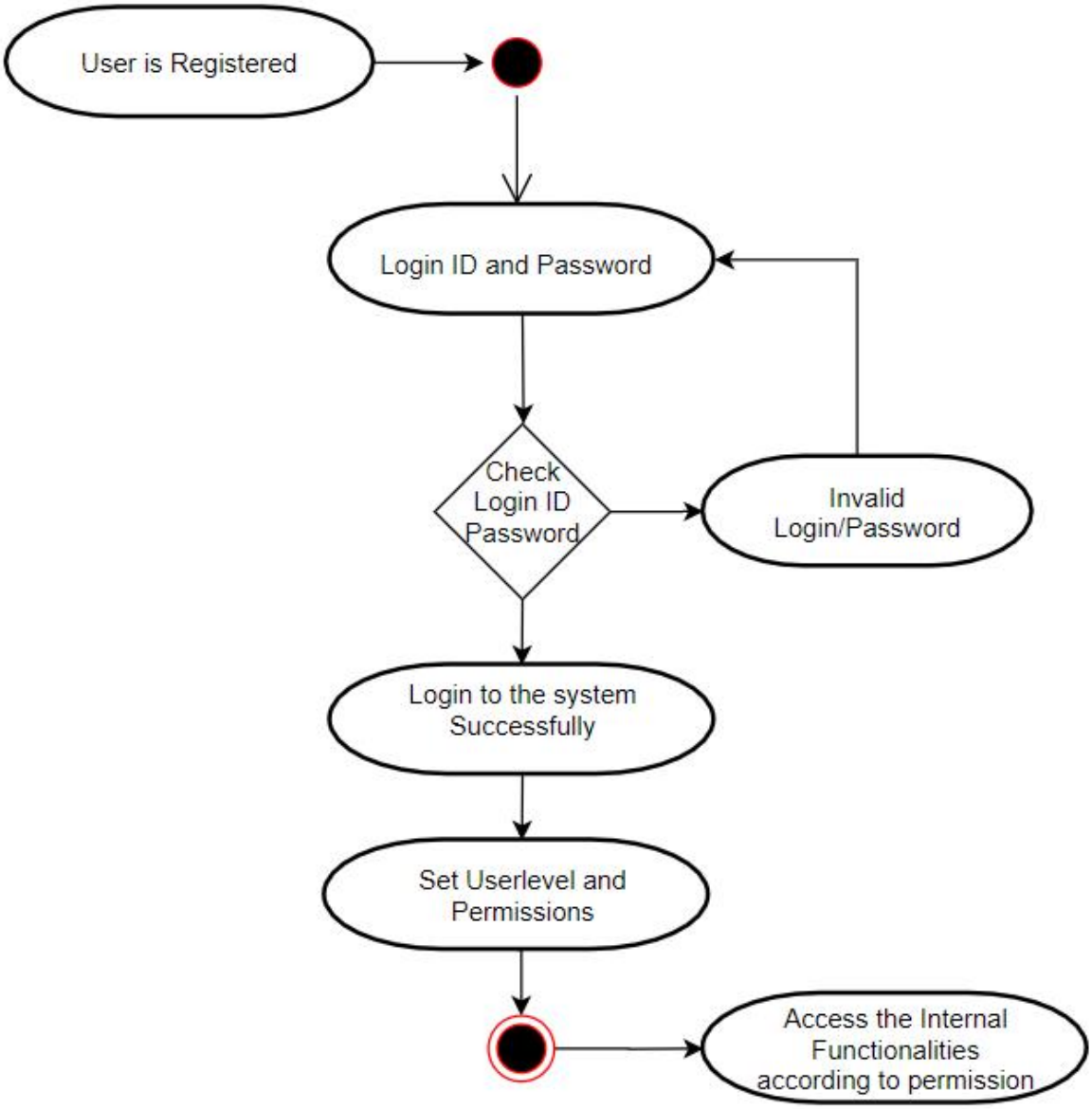
3.1.3 Admin


Manage appointemts
<<extend>>
<<extend>>
<<include>>
Cancel appointment
Change appointment
Authenticate patient
<<include>>
View patient records and information
Send reminder to patient
View all appointments for specific doctor
Receptionist
creately
www.creately.com • Online Diagramming

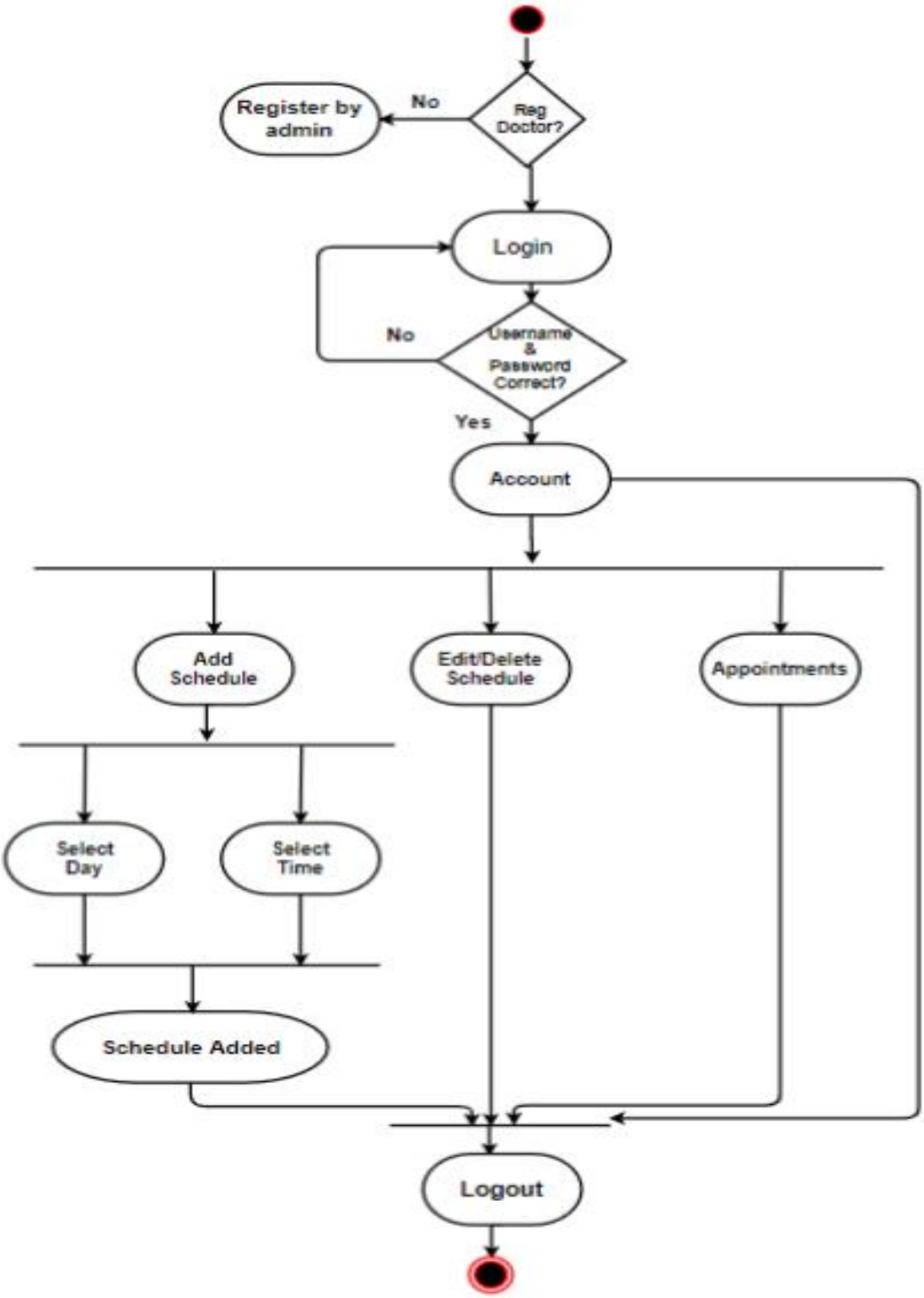
3.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

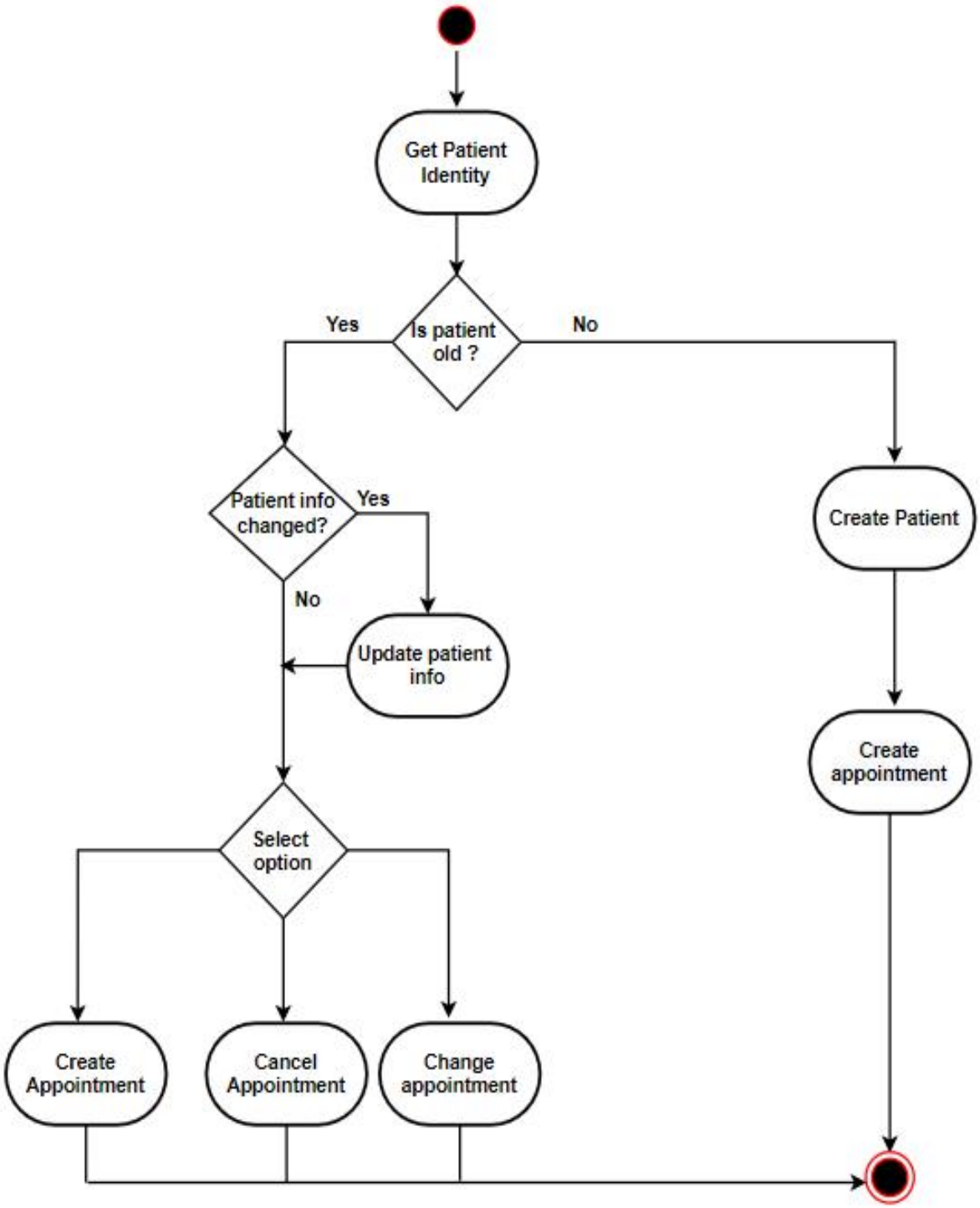
3.2.1 Login Activity Diagram of Doctor Appointment system



3.2.2 Activity Diagram for Doctor



3.2.3 Activity Diagram for Patient



4. User Characteristics

4.1 Overview of users

Patients : Patients are the primary consumers of an Appointment System. They are booking appointment based on the availability of doctors.

Doctors : Doctors are the primary content administrators of an Appointment System.They are managing appointments and may cancel or book an appointment on their availability,

4.2 Constraints

4.2.1 User Interface Constraints

Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

4.2.2 Hardware Constraints

The system should work on most home desktop and laptop computers which support JavaScript and HTML5.

4.2.3 Software Constraints

The system will be intended to run on Firefox 4 and above, Google Chrome 10 and above and Internet Explorer 8 and above.

4.2.4 Data Management Constraints

System shall be able to interface with other components according to their specifications.

4.2.5 Operational Constraints

The system is limited by its operating server in terms of the maximum number of users it can support at a given time.

4.2.6 Site Adaptation Constraints

The component will be adapted to the overarching system at the conclusion of the system creation.

4.2.7 Design Standards Compliance

The system shall be Node js and mongo DB.

4.2.8 Assumptions and dependencies

Most of the medical web application have a lot of redundant features which are rarely used for medical purposes. Our new system focuses on the features which are most important to the users of medical web application.

5. Specific Requirements

5.1 External interface

5.1.1 **Hardware Interface**

a) Server side

The web application will be hosted on a link using Heroku app.

b) Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus.

Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

5.1.2 **Software Interface**

a) Server side : Node.js will accept all requests from the client and forward it accordingly.A database will be hosted centrally using mongo DB.

b) Client side

An OS which is capable of running a modern web browser which supports JavaScript and HTML5.

5.1.3 **Communication Interfaces**

The HTTP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

5.2 Functional Requirements

5.2.1 Service booking and allocation-

PURPOSE	This platform will allow student to request any service along with a description of why they require it.
INPUTS	The details provided will be date of booking, time stamp for which the booking holds valid for, purpose and name of user.
PROCESSING	Checked whether the required service is available or not.
OUTPUTS	The user gets a display of available slots for the service they want

5.2.2 LOGIN

PURPOSE	The platform lets the user enter credentials and login.
INPUTS	The details provided are user ID provided to the user along with password
PROCESSING	The credentials entered are checked with database
OUTPUTS	The user gets logged in

5.2.3 Retrieve and display

PURPOSE	User gets a display of their bookings
INPUTS	The details provided will be user details
PROCESSING	In such a case the database is looked into for the user bookings
OUTPUTS	The user gets a display of their booking with available slots

5.2.4 Cancellation

PURPOSE	The service booked is to be cancelled
INPUTS	The details provided will be booking ID and user ID
PROCESSING	The service booked is deleted from the database and is freed for other users
OUTPUTS	The user gets a confirmation of cancellation, the user looking for the same service gets a notification that slot is vacant.

5.3 Performance Requirements

The system should support at least 1000 concurrent users.This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time.The times when the system will be under the most stress are likely during case of emergencies and situation like global pandemics. Therefore, it must be able to handle at least 1000 concurrent users.

5.4 Logical database requirements

All data will be saved in the database: user accounts and profiles, discussion data, messages etc. (except files which are stored on the disk.) The database allows concurrent access and will be kept consistent at all times requiring a good database design.

5.5 Software System Attributes

The software consists of the following elements:

1. The apache web server
2. The java script,html,css application
3. The Mongodb database
4. The database should remain consistent at all times in case of an error.

5.5.1 Reliability

The reliability of the overall program depends on the reliability of the separate components.

5.5.2 Availability

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system
runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in

case of a hardware failure or database corruption, backups of the database should be retrieved with the Mongodb server and saved by the administrator.

5.5.3 Security

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.
2. The user's IP will be logged.

3. Website has https enforced URL. (Hyper Text Transfer Protocol Secure) appears in the URL when a website is secured by an SSL certificate.

SSL Stands for secure sockets layer. Protocol for web browsers and servers that allows for the authentication, encryption and decryption of data sent over the Internet.

5.5.4 Portability

The application is practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application.

6.Document Approver

SRS for Online Doctor Appointment Booking System approved by:

_____

(Dr. Vinod Bhalla)

05-10-2020