



विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE- INDIAN INSTITUTE OF
INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR

SOFTWARE DESIGN DOCUMENT

Project Title: - MediCare

Group Members:

- 1) Mohd Wasiuddin Junaid 2018IMT-053
- 2) Narendra Chaudhary 2018IMT-056
- 3) Nitya Chandra 2018IMT-060
- 4) Saurav Kumar 2018IMT-091

Submitted to: Dr. Sanstosh Rathore.

INDEX

1. Design Considerations

- 1.1 Assumptions
- 1.2 Constraints
- 1.3 System environment
- 1.4 Design Methodology

2. Architecture

- 2.1 System Design
- 2.2 System Decomposition

3. Data Design

- 3.1 Data description
- 3.2 DFD
- 3.3 Data Dictionary

4. Component design

- 4.1 Login
- 4.2 Register
- 4.3 Forgot password

5. Software Interface Design

- 5.1. User interface design UI is designed according to UI design principles.
 - 5.1.1 Web pages in a tree
 - 5.1.1.1 Description

6. Class Diagram

- 6.1 Class Diagram Description

1. Design considerations

1.1. Assumptions

For System:

- The web based MediCare application will be on a server with high speed Internet capability.
- The product created here accepts the utilization of an instrument for association between the Site pages and the database.

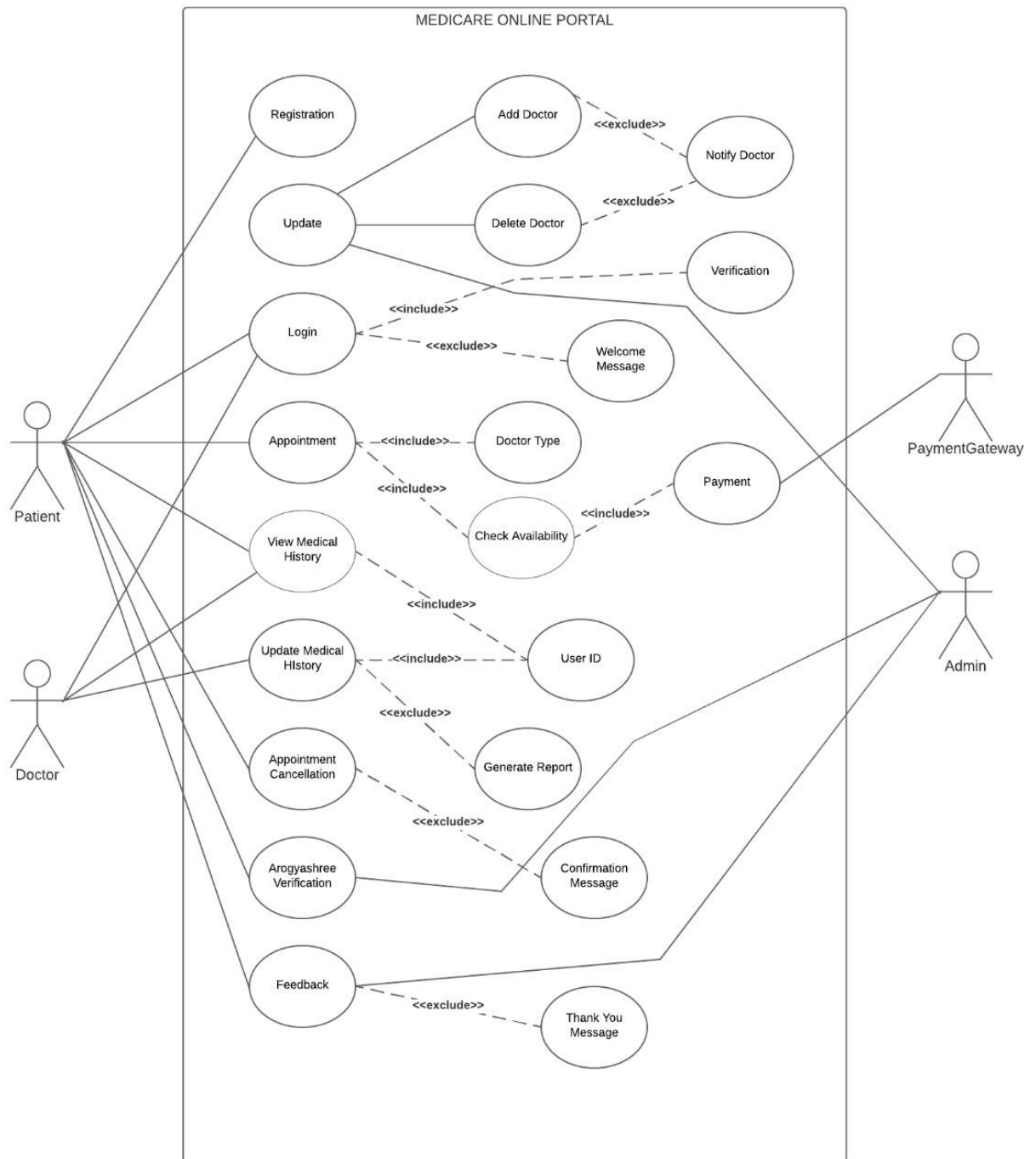
For User:

- The User is expected to be Internet literate and be able to use a search engine.
- The User is expected to be Windows literate and to be able to use buttons, drop-down menus, and similar tools.
- The User is expected to remember his user id and password .

1.2. Constraints

- Before running the client application make sure the server application is running. If the server is not in running condition then you can not book Appointment or search for Doctors.
- As we use react, Javascript should be enabled in the user's browser for smooth functioning of our application.

1.3. System environment



The application has two primary actors (Patient and Doctors) and one secondary actor, the administrator. Administrator is solely responsible for registering new Doctors on our Platform. Doctors are responsible for providing prescriptions to patients by seeing their medical history.

Patients can signup or sign in depending on whether they are new users or old users. They can book appointments with the doctors either by name or by speciality.

1.4. Design methodology

In the system development, there are a number of methodologies, known as SDLC (System Development Life Cycle), which can be applied such as Waterfall, Evolutionary Development and Component-Based Software Engineering. The selected-approach reflects the success of the system. In the waterfall model, the system development process cascades from one phase to another. It comprises six phases, namely:

Phase one : System Planning

Phase two: Problem Analysis

Phase three : System Design

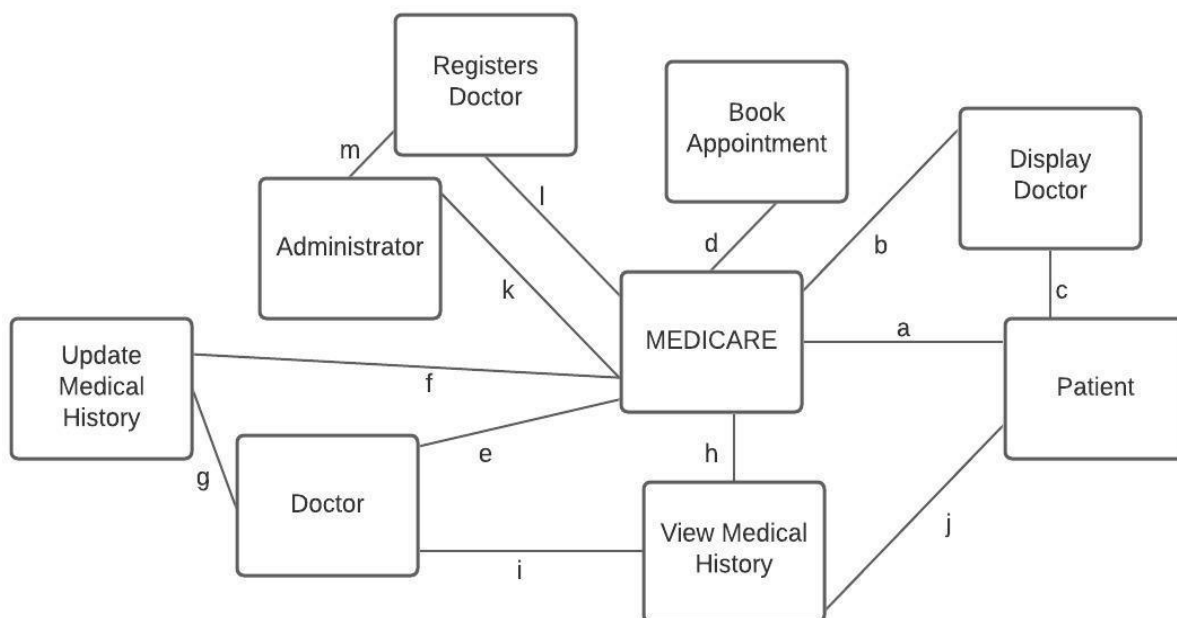
Phase four : System Implementation

Phase five : System Testing

Phase six : Operation and Maintenance

2. Architecture

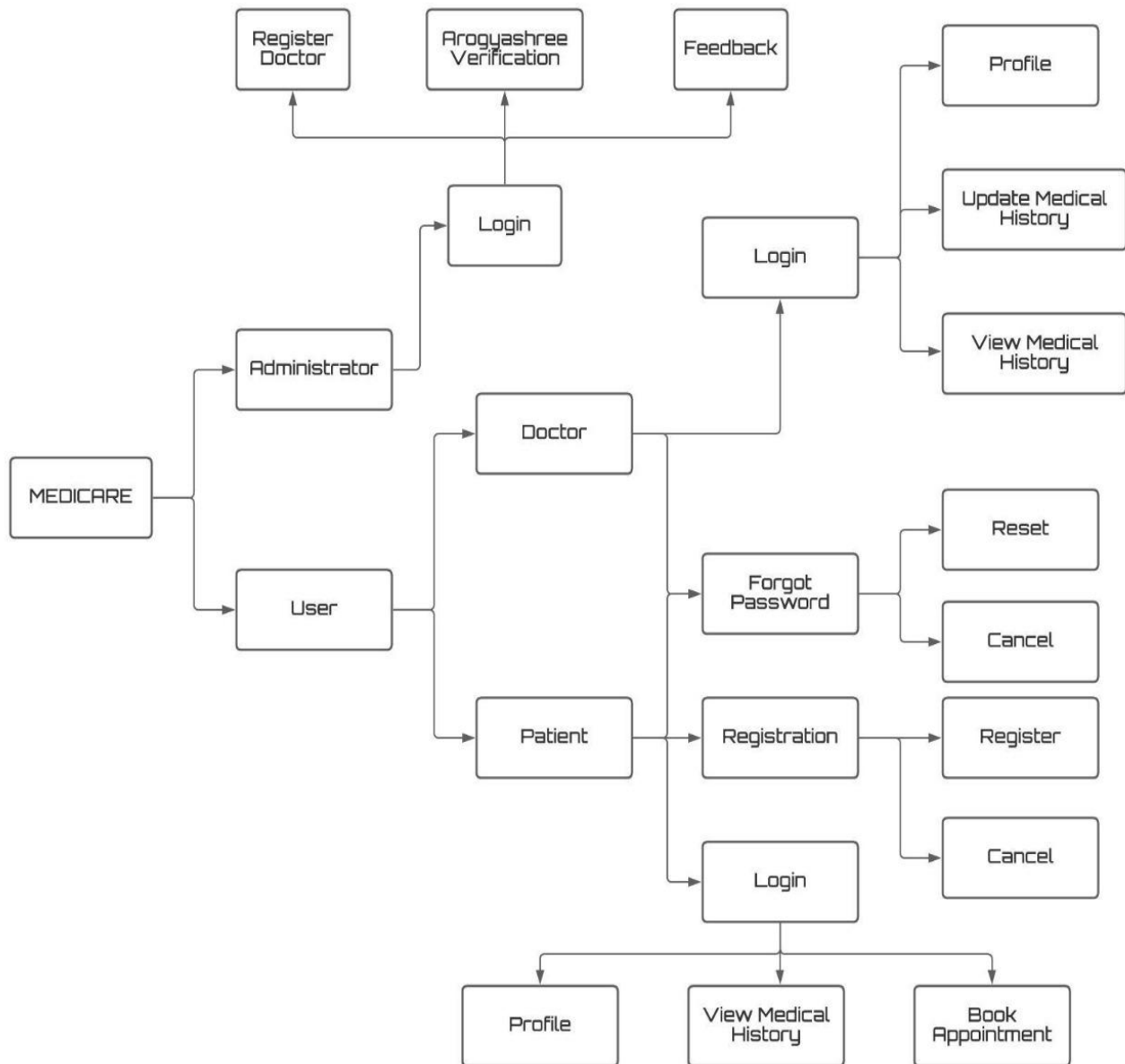
2.1. System design



The context diagram shows the main actors interacting with the system.

2.2. System decomposition

2.2.1. Functional decomposition tree



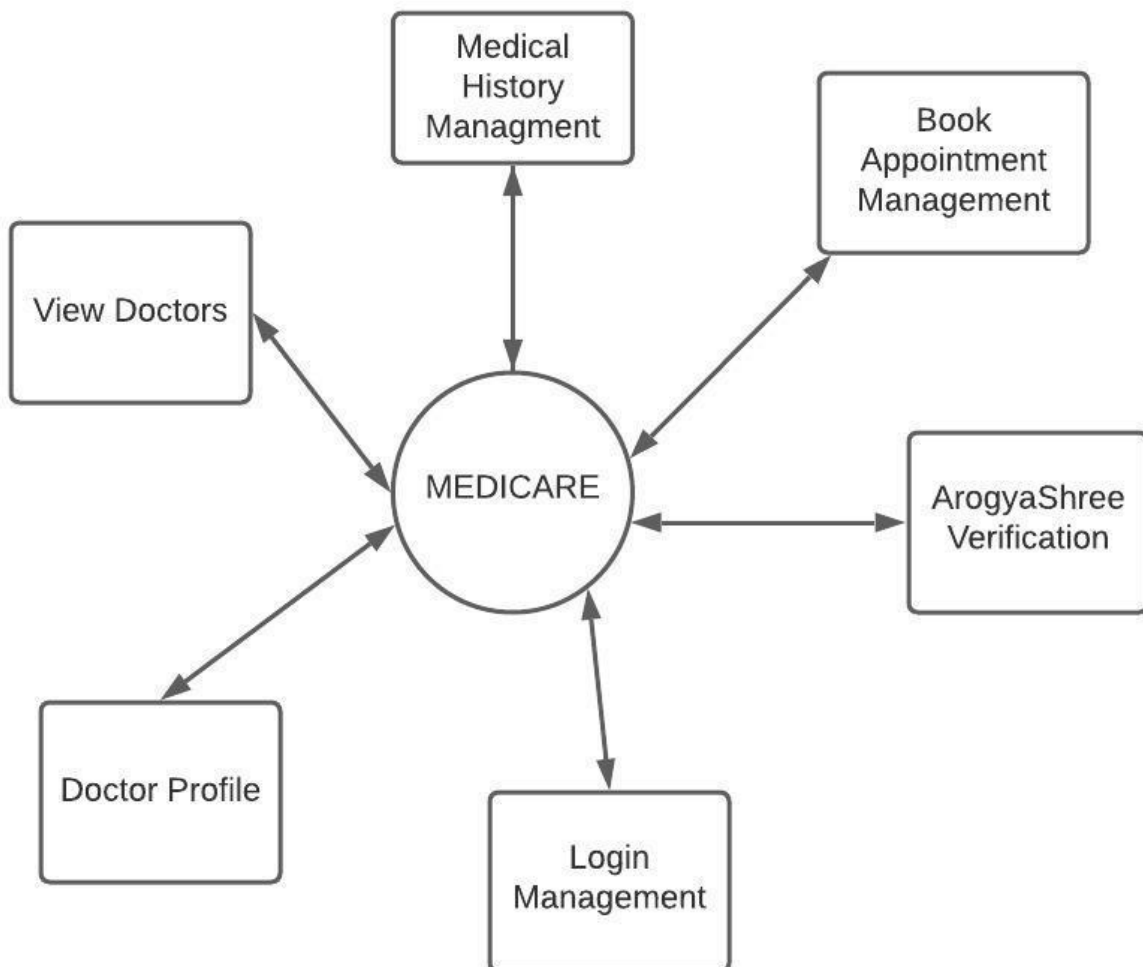
The above figure provides an overview regarding the functionalities of the application.

3. Data design

3.1. Data description

MongoDB-Atlas(Cloud Based) using compass UI software for all types of operating systems .

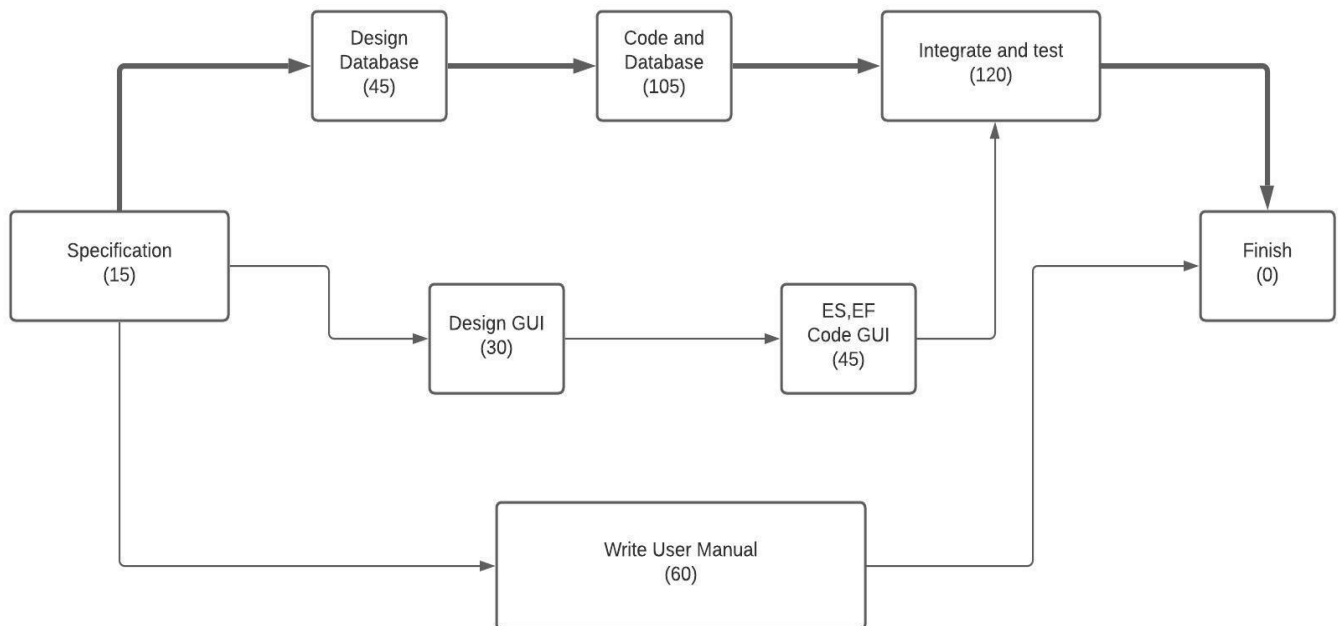
3.2. DFD diagrams



3.3. Data dictionary

	Field	Type	Null	Default
User Details	User_ID	int ()	No	AUTO_INCREMENT
	User_Name	varchar(50)	No	NOT_NULL
	Email_ID	varchar(50)	No	NOT_NULL
	Password	varchar(50)	No	NOT_NULL
	Connections	varchar(1000)	Yes	NULL
	Last_Logged	timestamp	No	CURRENT_TIMESTAMP

4. Component design



This is the activity diagram for the development of this application. . The number below each activity indicates the number of hours that are supposed for the completion of that activity.

The Main Components are:

4.1. Login

This is the Login section . Here Patient or Doctor or Admin can login and can navigate to their specific routes for booking appointments, uploading prescriptions or registering new doctors respectively.

4.2. Register

New patients can only register on our Platform.

4.3. Forgot password

Old users can reset their passwords here.

5. Software interface design

5.1. User interface design UI is designed according to UI design principles.

The structure principle:

UI is organized in such a way that related things are combined together and unrelated things are separate.

The simplicity principle:

It is easy to follow the provided interface. In the case of a mistake, the system displays a 404 error page.

The visibility principle:

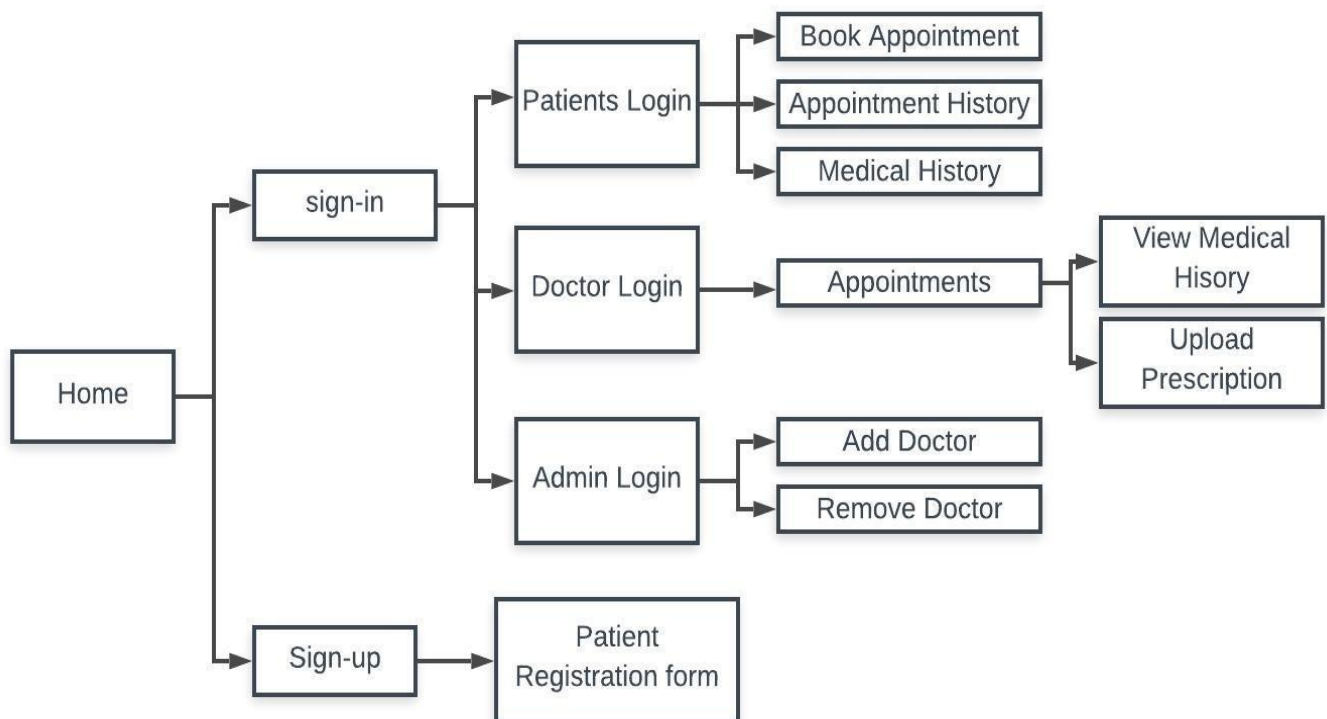
All the system's functions are available through the UI. It does not overwhelm users with too many alternatives.

The feedback principle: Through the system of messages, the design keeps users informed of actions, errors, or exceptions.

The reuse principle:

In design, same names were used to perform the same operations with different objects in order to reduce ambiguity.

5.1.1. Web pages in a tree



5.1.1.1. Description

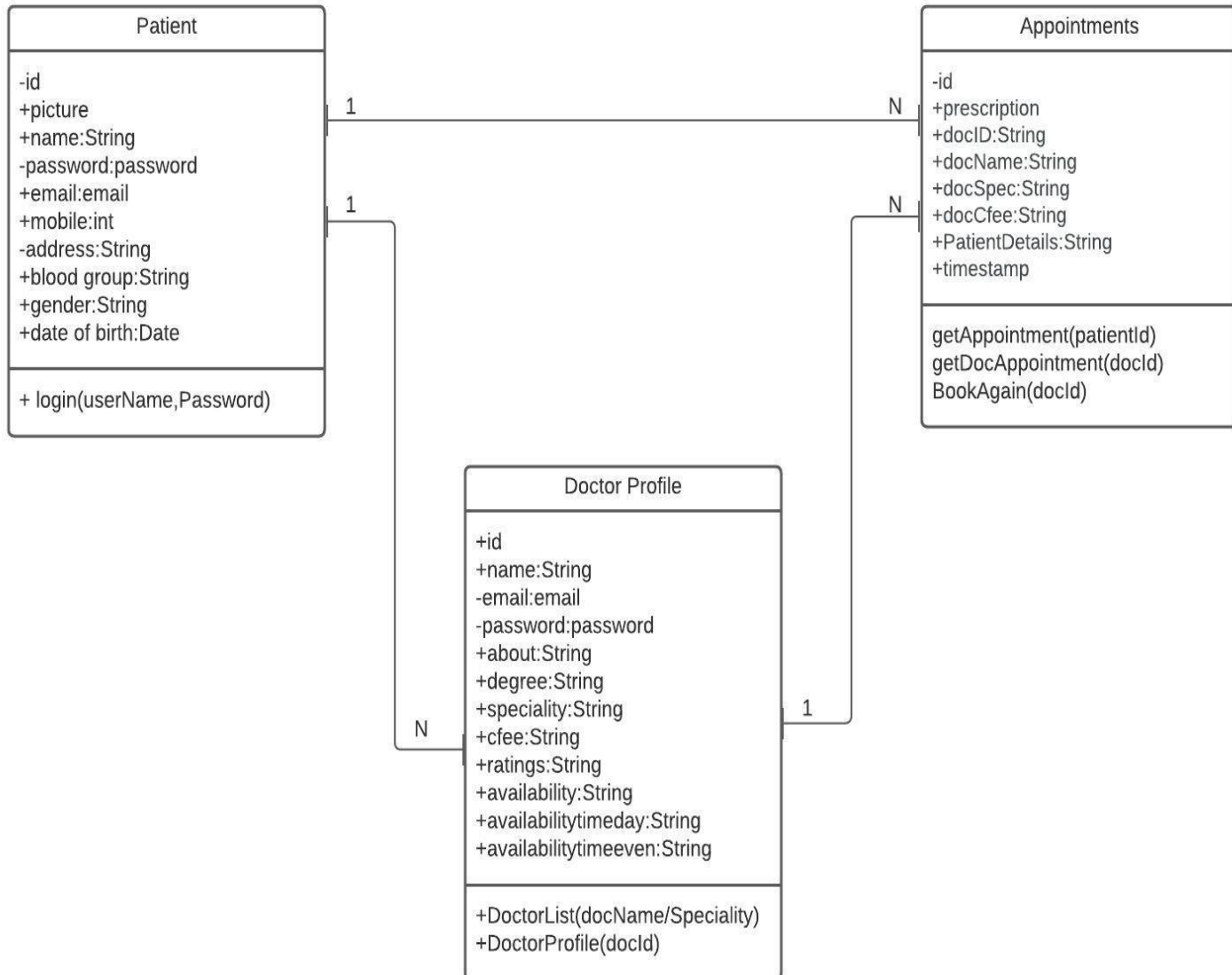
“HOME” page has the initial layout of this web-application. Users can search for doctors either by their name or speciality. After searching, they will land on the doctors list page, and on selecting any doctor, they will land on the doctor details page. If users have done login then they will see the ‘Book Now’ button else they will be redirected to the login page.

“Appointment” page is where users will be able to check their whole booked appointment. There they will get a ‘Book Again’ button which will land them on the doctor profile page and from there, they will be able to book appointments again. They will also be able to see the ‘Download Prescription’ button. By clicking on that, they will be able to download a prescription.

Page “docappointment” will show appointments to the doctor which is logged in. On clicking on Patient Name, they will be able to see their previous medical history.

Finally there is a 404 error page, if a user will try to navigate to some unknown routes, our app will take them to a 404 error page.

6. Class Diagram



6.1 Class Diagram Description

The above Class Diagram Has Four Classes:

- 1) Patient
- 2) Appointments
- 3) Doctor Profile

PATIENT:

The series of variables that user class contains are as follows.

- 1) id: This variable will be created by mongodb itself.
- 2) Picture: Takes input of Picture of user.
- 3) Name: Takes input of user's Name.
- 4) Password: Takes input of user's Password.
- 5) Email: Takes input of user's Email Id.
- 6) Mobile: Takes input of user's Mobile Number.
- 7) Address: Takes input of user's Address.
- 8) Blood Group: Takes input of the user's Blood Group.
- 9) Gender: Takes input of user's Gender.
- 10) Date of Birth: Takes input of user's Date of Birth.

The Function that patient class contain is:

- 1) login: This function authenticates username and password.

APPOINTMENTS:

The series of variables that Appointment contains are as follows:

- 1) id: This variable will be created by mongodb itself.
- 2) Prescription: This field contains the url of the uploaded prescription from the doctor side.
- 3) docID: This variable contains the id of the doctor to whom the appointment belongs.

- 4) docName: This variable contains the name of the doctor to whom the appointment belongs.
- 5) docSpec: This variable contains the specification of that particular doctor.
- 6) docCfee: This variable contains the consultant fee of that particular doctor.
- 7) PatientDetails: This variable contains the object ID of the patient to whom the appointment belongs.
- 8) TimeStamp: This variable will be created by mongodb itself.

The Function that Appointment class contain is:

- 1) getAppointment: This function will help the patient to retrieve all the appointments that he booked from the database.
- 2) getDocAppointment: This function will help the doctor to retrieve all the appointments that patient booked for his/her consultation.
- 3) BookAgain: This function will help the user to book appointment again to same doctor he booked in past. This will help the patient to save time in searching of doctor.

DOCTOR PROFILE:

The series of variables that Doctor Profile contains are as follows:

- 1) id : This variable will be created by mongodb itself.
- 2) name : Takes input of the doctor's name.
- 3) email : Takes input of the email id of the doctor.
- 4) password : Takes input of the password of the doctor.
- 5) about : Contains general information about the doctor.
- 6) degree : Contains educational qualifications of the doctor.
- 7) speciality : Mentions about the speciality of the doctor.
- 8) cfee : The consultation fee of the doctor(must be according to the norms of the organization)
- 9) ratings : This is a result of the patient feedback about the doctor. The patients can check the rating of the doctor before booking an appointment.
- 10) availability : Shows if the doctor is available that day.
- 11) availabilitytimeday : Shows the time when doctor is available during the day.
- 12) availabilitytimeeven : Shows the Doctor's availability in a week.

The Function that Doctor Profile class contain is:

- 1) DoctorList: This function allows you to search for available doctor for consultation
- 2) DoctorProfile: This functions allows to view the doctor profile such as speciality,ratings,experience so that you can book your appointment with a suitable doctor