A Project Report

On

Doctor Sab

Submitted in partial fulfillment of the requirement of

Project – VI ()

Of

Bachelor's Degree in Computer Application

**Submitted to:**



Purbanchal University

Biratnagar, Nepal

**Submitted By**

Bhuwan Bhatt ()

Manish Mandar ()

Sanjeev Rai ()

KANTIPUR CITY COLLEGE

Putalisadak, Kathmandu

A Project Report

On

Doctor Sab

Submitted in partial fulfillment of the requirement of

Project – VI ()

Bachelor's Degree in Computer Application

**Submitted to:**

Purbanchal University

Biratnagar, Nepal

**Submitted By**

Bhuwan Bhatt ()

Manish Mandar ()

Sanjeev Rai ()

Project Supervisor

Mr. Saroj Pandey

KANTIPUR CITY COLLEGE

Putalisadak, Kathmandu

**Abstract**

**Doctor Sab** is a health-focused web application built using modern Java technologies with **Spring Boot** at its core. The platform serves as a digital bridge between users and medical professionals, facilitating **user-doctor interaction**, **online consultations**, **first-aid guidance**, **symptom analysis**, and **appointment bookings**.

It aims to revolutionize online healthcare access in Nepal by enabling users to register either as patients or doctors. Through its user-friendly interface and robust backend, Doctor Sab offers a trustworthy and secure environment to manage digital healthcare needs.

# ACKNOWLEDGEMENT

We would like to take this opportunity to present our votes of thanks to all those guide post who really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really thankful to Project Supervisor Mr. Saroj Pandey for his active support, valuable time and advice, sincere cooperation during the study and in completing the assignment of preparing the said project within the given time stipulated. We are also very grateful to our Principal Raju Kattel for providing us with an opportunity to undertake this project and providing us with all the facilities.

In the end, we are thankful to all those, particularly the various friends, who have been creating proper, and good environment and also including new and fresh innovative ideas for us during the project, their help, it would have been very difficult for us to prepare the project in a time bound framework.

**Preface**

In today’s fast-paced and increasingly digital world, health and convenience often clash. While many prefer digital tools for health checks and consultations, accessibility and reliability remain issues*.* Doctor Sab is our humble attempt to close that gap by leveraging **Java + Spring Boot** and **REST APIs** to create a healthcare platform that is efficient, scalable, and secure.

This project enabled us to combine backend expertise with frontend user experience to deliver real-world health services online, from symptom checks to direct consultations with verified doctors.

**Declaration**

We affirm that this project report, titled “Doctor Sab”, presented for the fulfillment of the BCA-VI, represents original work conducted under the guidance of Mr. Saroj Pandey. It has not been used to obtain any other degree or diploma from any institution or university. We have adhered to ethical standards by duly acknowledging the contributions of others whenever their findings were referenced.

Bhuwan Bhatt ( )

Manish Mandar ()

Sanjeev Rai ( )

# 

# TOPIC APPROVAL SHEET

It is hereby informed that for the semester project, the topic selected by Bhuwan Bhatt, Manish Mandar and Sanjeev Rai of BCA-VI is found suitable and appropriate as per the credit assigned by Purbanchal University (PU), Biratnagar, Nepal.

The project committee has approved the following topic and assigned a supervisor for the above mentioned students.

Topic Approved: Doctor Sab

-------------------------------------

BCA & BIT Program Coordinator

# 

# CERTIFICATE FROM THE SUPERVISOR

This is to certify that the project entitled “Doctor Sab”, submitted by Bhuwan Bhatt, Manish Mandar and Sanjeev Rai to the Department of Information Technology and Engineering at Kantipur City College, Kathmandu, Nepal towards the requirement for BCA: Project-VI, is an original work carried out by them under my guidance and supervision.

Signature:

-----------------------------------

Mr. Saroj Pandy

Project supervisor

Department of IT and Engineering

Kantipur City College

**Table of Content**

[**Abstract** iii](#_Toc199496549)

[ACKNOWLEDGEMENT iv](#_Toc199496550)

[TOPIC APPROVAL SHEET iii](#_Toc199496551)

[CERTIFICATE FROM THE SUPERVISOR iv](#_Toc199496552)

[Chapter 1: Introduction 1](#_Toc199496553)

[**1.1** **Overview** 1](#_Toc199496554)

[**1.2** **Significance** 1](#_Toc199496555)

[**1.3** **Objectives** 2](#_Toc199496556)

[**1.4** **Features** 2](#_Toc199496557)

[**1.5** **Problem Statement** 3](#_Toc199496558)

[**1.6 Scope and limitation** 3](#_Toc199496559)

[**1.7 Organization of the document** 4](#_Toc199496560)

[Chapter 2: Literature Review 5](#_Toc199496561)

[**2.1 Case Studies: Telemedicine Platforms** 5](#_Toc199496562)

[**2.1.1 Teladoc Health** 5](#_Toc199496563)

[**2.1.2 Practo** 6](#_Toc199496564)

[**2.1.3 Babylon Health** 6](#_Toc199496565)

[**2.2 Implications for Doctor Sab** 7](#_Toc199496566)

[Chapter 3: Methodology and Schedule 8](#_Toc199496567)

[**3.1 Software Development Life Cycle** 8](#_Toc199496568)

[**3.2 Assignment of roles and responsibilities** 10](#_Toc199496569)

[**Chapter 4: System Analysis** 11](#_Toc199496570)

[**4.1 Requirement Analysis** 11](#_Toc199496571)

[**4.1.1 Functional Requirements** 11](#_Toc199496572)

[**4.1.2 Non-Functional Requirements** 12](#_Toc199496573)

[**4.2 Feasibility Study** 12](#_Toc199496574)

[**4.2.1 Technical Feasibility** 12](#_Toc199496575)

[**4.2.3 Schedule Feasibility** 13](#_Toc199496576)

[Chapter 5: System Design 14](#_Toc199496577)

[**5.1 Use Case Diagram** 14](#_Toc199496578)

[**5.2 Class Diagram** 15](#_Toc199496579)

[**5.3 Data Flow Diagram** 16](#_Toc199496580)

[**5.4 ER Diagram** 18](#_Toc199496581)

[Chapter 6: System Development and Implementation 19](#_Toc199496582)

[**6.1 Tools and Technology Used** 19](#_Toc199496583)

[Chapter 7: Testing and Debugging 20](#_Toc199496584)

[**7.1 Test case** 20](#_Toc199496585)

[Chapter 8: Conclusion 21](#_Toc199496586)

[References 22](#_Toc199496587)

[Appendix 23](#_Toc199496588)

**List of figures**

|  |  |  |
| --- | --- | --- |
| Figure no. | Figure | Page no. |
| 1.1 | Waterfall model | 8 |
| 1.2 | Use-case diagram | 14 |
| 1.3 | Class diagram | 15 |
| 1.4 | Sequence diagram | 16 |
| 1.5 | Data flow diagram | 17 |
| 1.6 | ER diagram | 18 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table no. | Table | Page no. |
| 1 | Assignment of roles and responsibilities | 10 |
| 2 | Gantt chart | 12 |
| 3 | Test case | 18 |

# Chapter 1: Introduction

Doctor Sab is an innovative, web-based healthcare platform designed to connect patients with medical professionals and provide essential health services in a user-friendly digital environment. Doctor Sab ensures that users can access essential healthcare services anytime, anywhere, without the burden of costly subscriptions, long wait times, or complex procedures.

## **Overview**

**Doctor Sab** is a free, user-focused healthcare platform redefining how individuals access medical services online. Powered by **Java**, **Spring Boot**, **Llama 2.**  Doctor Sab empowers users with a wide range of features from instant **first aid guidance** and **AI-powered symptom checks** to **real-time consultations** and **easy doctor appointment bookings.**

## **Significance**

* **Accessible** Healthcare Doctor Sab provides free and user-friendly access to essential medical services, breaking barriers to digital healthcare for individuals from all walks of life.
* **Universal Access**: The platform democratizes healthcare by enabling anyone, anywhere to connect with medical professionals, access first aid knowledge, check symptoms, and book consultations—without the constraints of physical location or financial limitation.

## **Objectives**

**Here are some detailed objective of the proposed Doctor Sab:**

* **Provide Free Access to** Healthcare Services: Offer user’s unrestricted access to basic healthcare tools and resources such as first aid guidance, symptom checking, and doctor consultations without any subscription fees.
* **Connect Patients with Doctors Seamlessly**: Facilitate direct interaction between patients and qualified doctors through a secure and efficient consultation and appointment systems.

## **Features**

**Here are some detailed features of the proposed Doctor Sab:**

* **User and Doctor Registration**: Allows individuals to sign up either as a **patient** or a **doctor**, with role-specific features and access levels tailored to their needs.
* **Symptom Checker Tool**: Enables users to input symptoms and receive preliminary health insights, helping them decide when to seek professional care.

## **Problem Statement**

Access to affordable and timely healthcare remains a challenge, especially in remote and underserved areas. Traditional systems often involve long wait times, high costs, and limited accessibility. Many individuals also lack basic health knowledge or immediate first aid support. **Doctor Sab** addresses these issues by offering a free, user-friendly platform for symptom checking, first aid guidance, doctor consultations, and appointment booking making essential healthcare services more accessible, efficient, and affordable for everyone, anytime and anywhere.

## **1.6 Scope and limitation**

The scope of the **Doctor Sab** project is to create an accessible, user-friendly healthcare platform that connects patients with doctors and provides essential health tools globally. Key areas include:

* **Developing a User-Friendly Platform:** Ensuring smooth navigation, easy access to health services, and intuitive interactions for users of all ages and backgrounds.
* **Offering Essential Healthcare Features:** Providing symptom checking, first aid guidance, online consultations, and appointment bookings with qualified doctors.

Here are some limitation of this project:

* Doctor Sab requires a stable internet connection for accessing consultations and online features, which may limit users in regions with poor connectivity.
* The platform does not replace in-person medical emergencies or professional diagnosis.

## **1.7 Organization of the document**

The document is structured into nine chapters, each with its respective sub-chapters. The organization of the document is as follows:

|  |  |
| --- | --- |
| **Chapters** | **Heading** |
| Chapter 1 | Introduction |
| Chapter 2 | Literature Review |
| Chapter 3 | Methodology |
| Chapter 4 | System Analysis |
| Chapter 5 | System Design |
| Chapter 6 | System development and implementation |
| Chapter 7 | Testing and debugging |
| Chapter 8 | Conclusion |
| Chapter 9 | Reference |

# Chapter 2: Literature Review

As we embarked on the development of **Doctor Sab**, a web-based healthcare platform, our goal was to provide free and easy access to essential medical services, ensuring a reliable and user-friendly experience. We conducted extensive research into existing telemedicine and digital health platforms such as Teladoc, Practo, Babylon Health, and WebMD to understand their strengths and limitations.

## **2.1 Case Studies: Telemedicine Platforms**

**Introduction**

The telemedicine industry has rapidly expanded, offering users easier access to healthcare services through digital means. We reviewed several prominent telehealth platforms to identify best practices and challenges, aiming to incorporate valuable insights into **Doctor Sab.**

### **2.1.1 Teladoc Health**

Teladoc is a global telemedicine provider offering virtual consultations with licensed doctors across various specialties.

**User Reviews:**

**Negative Feedback:**

Pricing and Subscription Tiers:

Appointment Wait Times: Some users report delays in scheduling or connecting with doctors during peak hours.

**Positive Feedback:**

Convenience and Accessibility:

Users appreciate the ability to consult doctors remotely, especially during the COVID-19 pandemic.

### **2.1.2 Practo**

Practo is a popular platform combining appointment booking, online consultations, and health record management.

**User Reviews:**

**Negative Feedback:**

App Performance:

Some users mention occasional app crashes and slow loading time.

**Positive Feedback:**

Integrated Features:

Users value the all-in-one approach, managing appointments, consultations, and health records from a single app.

### **2.1.3 Babylon Health**

Babylon Health offers AI-powered symptom checking alongside doctor consultations.

**Weaknesses:**

Symptom Checker Limitations:

Some users report inaccuracies or generic advice from the AI tool.

**Strengths:**

Innovation:

Users appreciate the integration of AI to provide quick preliminary health assessments.

## **2.2 Implications for Doctor Sab**

The insights gathered from analyzing established telemedicine and digital health platforms provide valuable lessons for the development of **Doctor Sab.**

* Deliver Trustworthy Healthcare Services**:** Just as popular platforms focus on high-quality content, Doctor Sab must prioritize accurate symptom checking, reliable first aid guidance, and consultations with verified doctors to build user confidence.
* Provide an Ad-Free, Patient-Centered Experience Avoiding intrusive ads ensures users can focus on their health needs without distractions, enhancing satisfaction and engagement.

# Chapter 3: Methodology and Schedule

For the development of **Doctor Sab**, we selected the **Iterative Waterfall Model** due to the clear project requirements and the straightforward nature of the application’s features.

## **3.1 Software Development Life Cycle**

**Iterative Waterfall Model**

The iterative waterfall model is a software development approach that combines elements of both the waterfall model and iterative development. In this model, the software development process is divided into a series of iterations, or cycles and it provides feedback pathways from each step to the phases before.

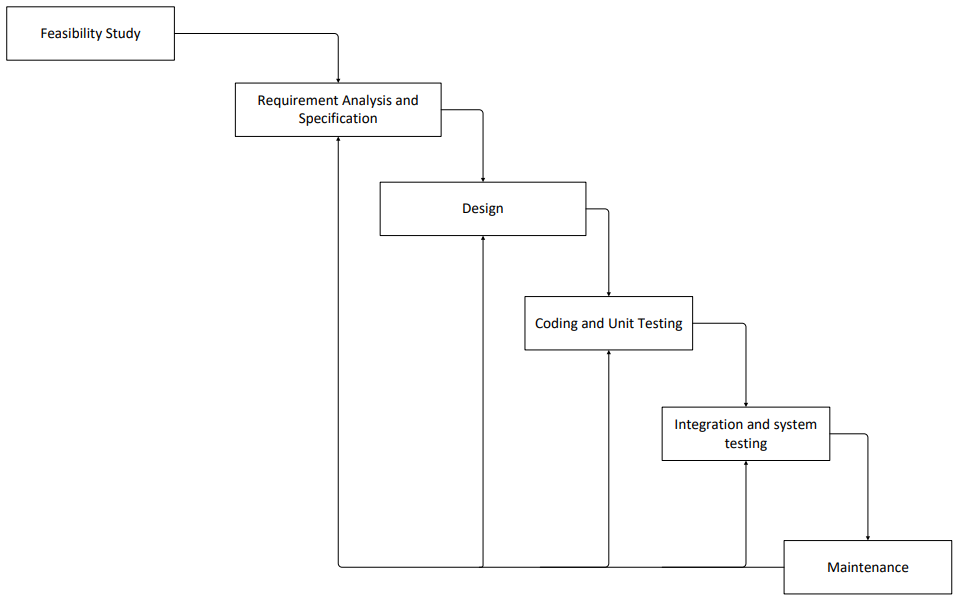


Fig 1.1: Steps of iterative waterfall model

**Requirements Gathering and Analysis:**

* Define User Requirements: Gather requirements related to patient and doctor interactions, ensuring ease of use for health services such as symptom checking, first aid guidance, consultation booking, and secure communication. Focus on accessibility, privacy, and a seamless experience across devices.
* Functional Requirements: Specify key features including user registration with role selection (patient or doctor), symptom checker tool, first aid information, appointment scheduling, real-time consultations, and secure messaging. Ensure responsiveness and quick load times.
* Technical Requirements: Identify technologies such as **Java** and **Spring Boot** for backend development, **Spring Security** for authentication and authorization, and **HTML/CSS/JavaScript** for a responsive front-end interface.

**Design:**

* User Interface (UI) Design: Develop wireframes and mockups with a clean, intuitive layout focusing on simplicity and accessibility. Ensure easy navigation for all user types, with quick access to health tools and consultation features. Prioritize responsiveness across desktop and mobile devices.

**Implementation:**

* Implement backend services using **Spring Boot**, including RESTful APIs to handle user registration, symptom checking, appointment booking, and doctor-patient interactions.

**Testing:**

* Test individual components such as user registration, symptom checker accuracy, appointment scheduling, and messaging modules to ensure they function correctly.

## **3.2 Assignment of roles and responsibilities**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Team Members** | **Roles and Responsibilities** |
| 1 | Bhuwan Bhatt | Documentation, coding and testing |
| 2 | Manish Mandar | Documentation, coding and testing |
| 3 | Sanjeev Rai | Documentation, coding and testing |

## **Chapter 4: System Analysis**

## **4.1 Requirement Analysis**

### **4.1.1 Functional Requirements**

For Doctor Sab, the following functional requirements have been identified:

* **User Registration and Authentication:** The system must support registration and secure login for both **patients** and **doctors**, using role-based access control. Users should be able to manage profiles and preferences.
* Symptom Checker**:** Patients can input symptoms and receive preliminary health information or recommendations.
* First Aid Information**:** Provides users with structured, easy-to-understand first aid guides for common emergencies.
* **Doctor Dashboard:** Doctors can manage appointments, view patient information, and respond to consultation requests.
* **Appointment Booking System:** Patients can view doctor availability, schedule appointments, and receive confirmation and reminders.

### **4.1.2 Non-Functional Requirements**

Non-functional requirements for Doctor Sab include:

Usability: The system should be highly usable, with an emphasis on a seamless user experience.

* Usability: The interface should be intuitive and accessible to users of all ages and technical skill levels. Key features must be easy to access and use across devices.
* Performance: The system should offer fast response times for all operations including symptom checking, appointment booking, and user authentication.

## **4.2 Feasibility Study**

### **4.2.1 Technical Feasibility**

* Frontend: HTML, CSS, JavaScript for a responsive UI.
* Backend: Java with **Spring Boot** for handling business logic, RESTful APIs, authentication, and user management.
* Database: NoSQL for data management.

**4.2.2 Economic Feasibility**

Doctor Sab is an academic project and will be developed using open-source tools and frameworks. There are **no licensing or infrastructure costs,** making the project economically feasible. Development can be carried out on local machines, and deployment (if needed) can use free-tier services for demonstration purposes.

### **4.2.3 Schedule Feasibility**

The progress of our project is shown in the gantt chart below:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PROCESS |  | | | | September | | | | May | | | | June | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week1 | Week2 | Week3 | Week4 | Week1 | Week2 | Week3 | Week4 | Week1 | Week 2 |
| Idea Presentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Required analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing and debugging |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Chapter 5: System Design

## **5.1 Use Case Diagram**

A drawing of a person with a light and lines

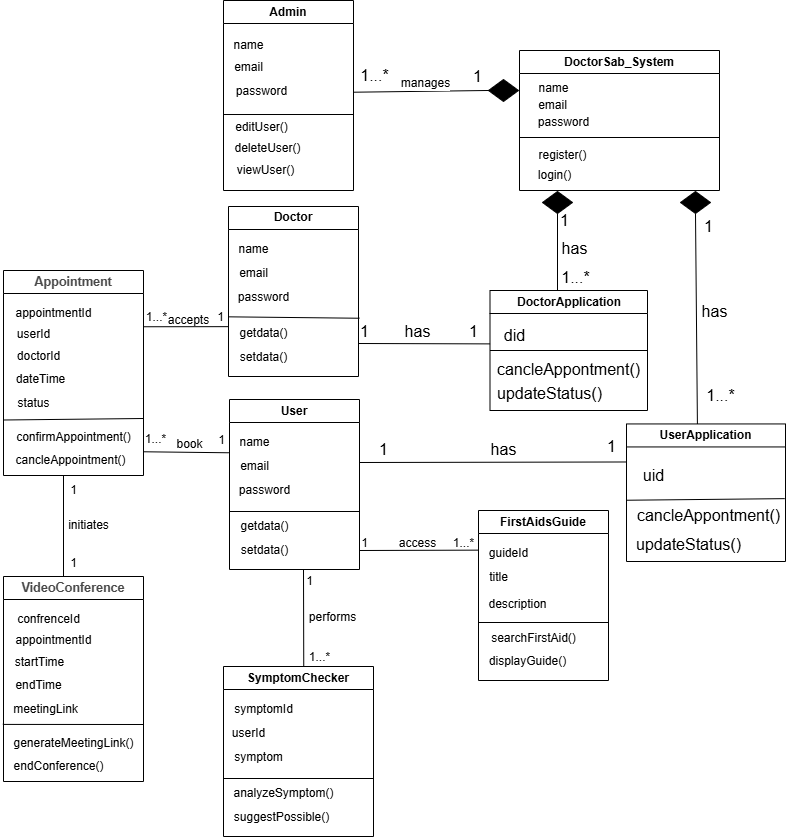
AI-generated content may be incorrect.

Fig 1.2 User Use Case Diagram

A diagram of a person with text

AI-generated content may be incorrect.

Fig 1.3 Admin Use Case Diagram



## **5.2 Class Diagram**

Fig 1.4 Class Diagram

## **5.3 Data Flow Diagram**

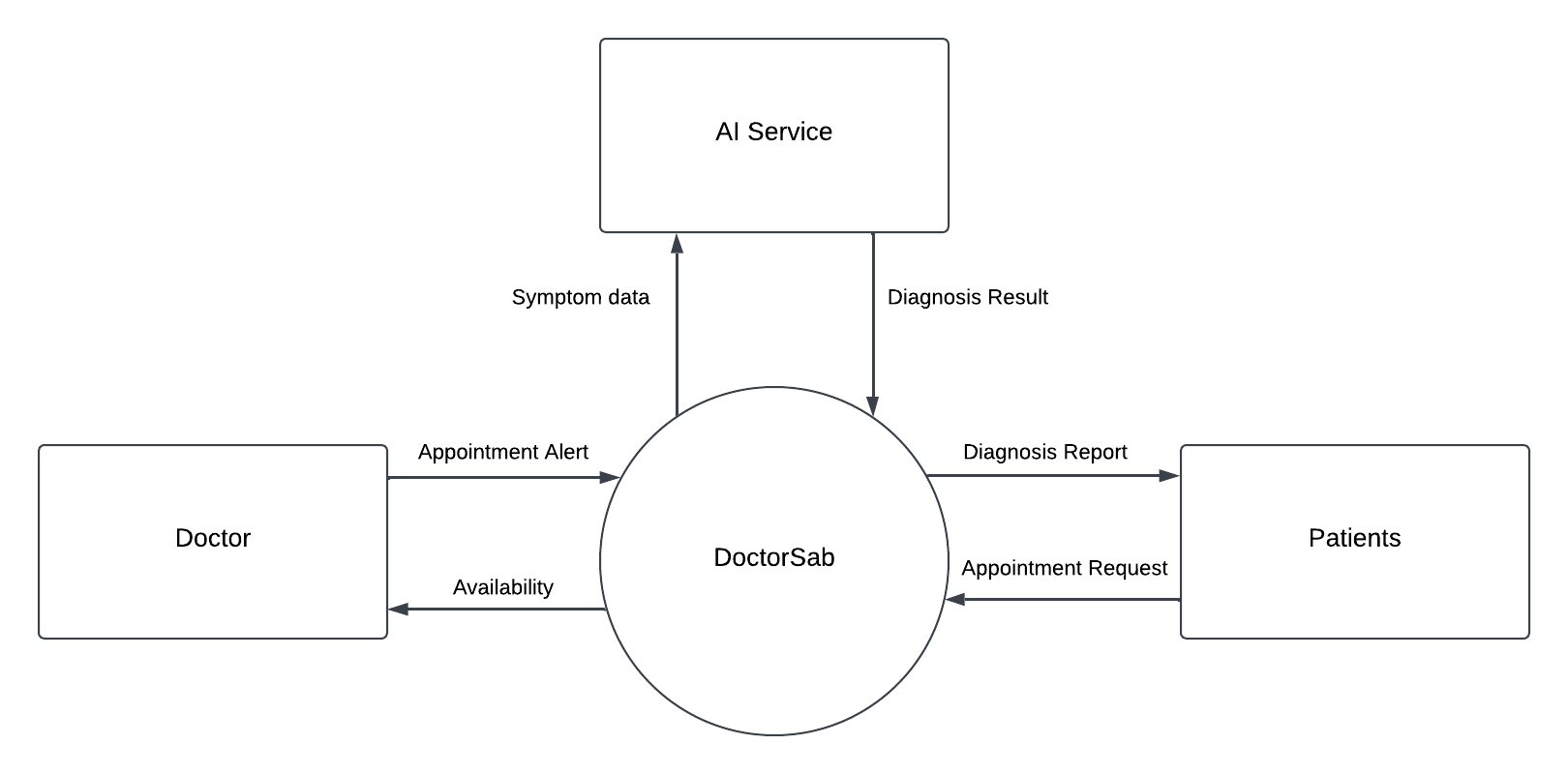


Fig 1.5 0-level Data Flow Diagram

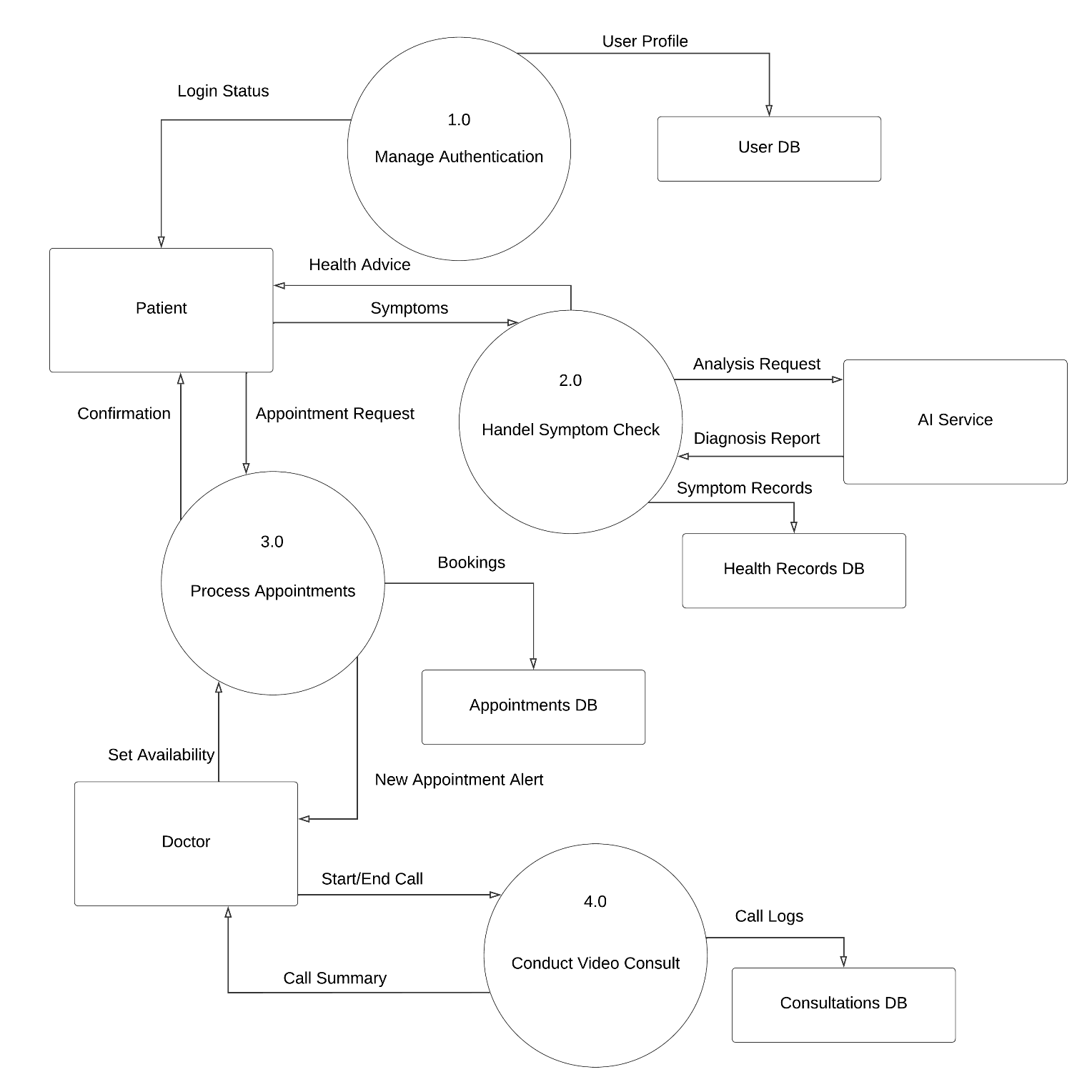


Fig 1.6 1-level Data Flow Diagram

## **5.4 ER Diagram**

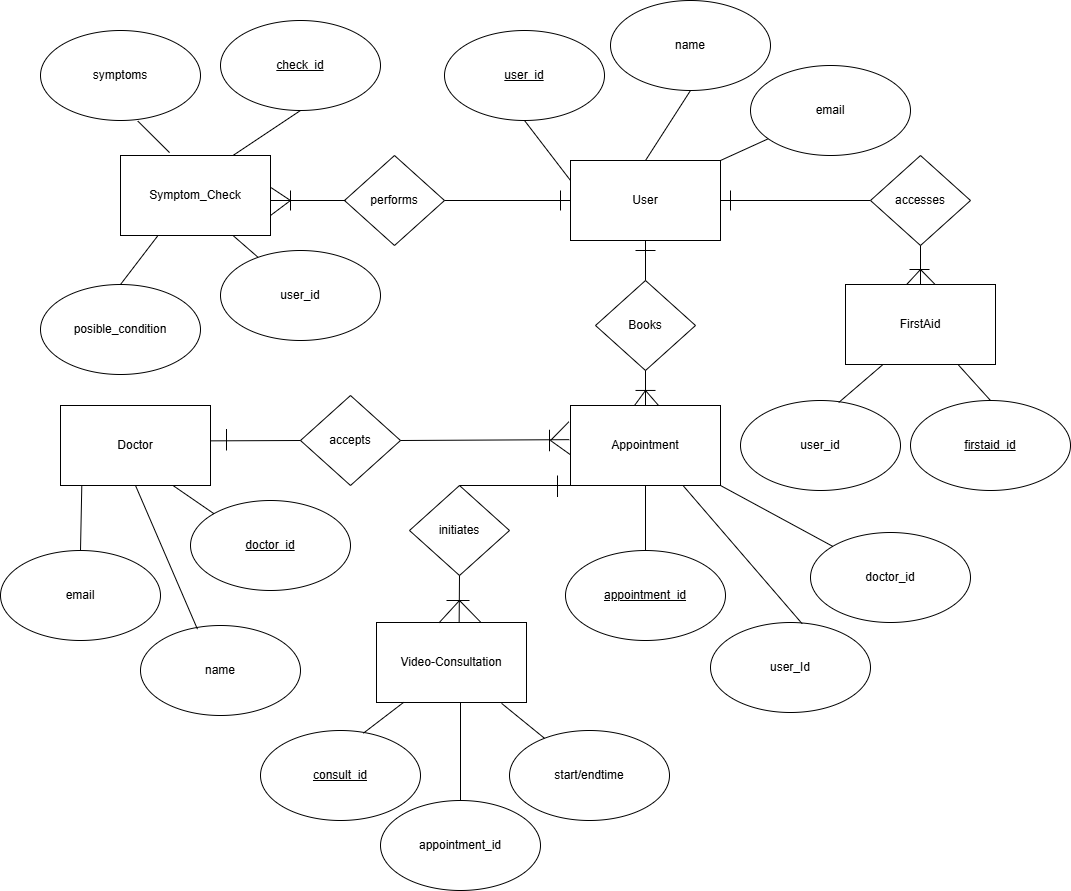


Fig 1.7 ER Diagram

# Chapter 6: System Development and Implementation

## **6.1 Tools and Technology Used**

The **Doctor Sab** healthcare platform was developed using a variety of modern tools and technologies to ensure high performance, security, and a user-friendly experience for both patients and doctors.

**Frontend Development:**

Used for structuring and styling the web pages, ensuring responsive and accessible design across different devices.

**JavaScript**: JavaScript was used to handle interactions and make the site more dynamic, such as managing user inputs, fetching data, and updating the page content in real-time.

**Backend Development:**Java & Spring Boot: Spring Boot was used to build a robust and scalable backend. It handles core functionalities like user authentication, appointment booking, symptom checker services, and REST API development.

**Testing and Quality Assurance:**

Unit Testing Framework: Leveraged a unit testing framework to verify the functionality of individual components and features.

Usability Testing: Conducted usability testing with potential users to ensure the platform’s ease of use and identify areas for improvement.

**Documentation:**

Microsoft Office Suite: Utilized for creating project documentation, including user manuals, technical documentation, and reports.

**Development Environment:**

**Visual Studio Code**: The IDE of choice for development, Visual Studio Code, was used to write and debug HTML, CSS, JavaScript, and integration with Git.

# Chapter 7: Testing and Debugging

## **7.1 Test case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Test | Description | Expected Result | Actual Result |
| 1 | User Login | Valid user credentials | Access to user dashboard | Pass |
| 2 | Doctor Login | Valid doctor credentials | Access to doctor portal | Pass |
| 3 | User Register | New user signup | Account created; verification sent | Pass |
| 4 | Doctor Register | New doctor signup | Account pending verification | Pass |
| 5 | Symptom Check | Submit symptoms | Diagnosis report displayed | Pass |
| 6 | Book Appointment | Available slot selected | Confirmation; doctor notified | Pass |
| 7 | Book Appointment | Occupied slot selected | "Unavailable" error; alternatives shown | Fail |
| 8 | Start Video Call | Doctor initiates consultation | User notified; meeting link created | Fail |
| 9 | Join Video Call | User joins promptly | Stable connection established | Pass |
| 10 | First Aid Search | Search "CPR" | CPR instructions displayed | Fail |
| 11 | User Logout | Logout from dashboard | Redirect to login; session cleared | Pass |
| 12 | Doctor Logout | Logout from portal | Redirect to login; session cleared | Pass |
| 13 | Responsive UI | Mobile/tablet/desktop view | Properly adapts to all screens | Pass |
| 14 | Browser Support | Video call on Chrome/Firefox/Safari | Stable connection on all browsers | Pass |
| 15 | Network Failure | Disconnect during video call | Auto-reconnect prompt | **Fail** |
| 16 | Symptom History | View past symptom checks | Historical records displayed | Pass |
| 17 | Appointment Alert | Schedule future appointment | Reminder sent 30 minutes prior | False |

# Chapter 8: Conclusion

In conclusion, **Doctor Sab** is a modern, user-friendly healthcare web application developed using **Java, Spring Boot.** The platform was designed to bridge the gap between patients and doctors by providing essential medical tools such as a symptom checker, first aid guidance, online consultations, and appointment booking—all within a secure and intuitive interface.

Throughout development, the project leveraged core web technologies like **HTML, CSS**, and **JavaScript** for the frontend, while **Spring Boot** provided a robust and scalable backend framework**. Firestore** ensured efficient management of user data, appointments, and medical interactions, while **Spring Security** helped maintain data confidentiality and role-based access control.

Doctor Sab not only fulfills academic objectives but also demonstrates practical insights into building secure, service-oriented healthcare platforms. The application is designed with future scalability in mind, allowing for potential enhancements such as multilingual support, integration of real-time video consultations, electronic health records, and mobile responsiveness. Overall, **Doctor Sab** represents a step forward in accessible, digital-first healthcare solutions that are both technically sound and user-centric.

# 

# References

# Appendix