

A CASE STUDY (IEEE Format)

UCS503 Software Engineering Project

Software Requirements Specification Document

**End Semester Evaluation**

**Health Care Portal**

**Submitted by:**

Shreeya Chatterji 102103447

Harsh Jain 102103432

**BE 3rd Year, COE16**

**Submitted to:**

Assistant Professor Dr. Nitigya Sambyal



**TABLE OF CONTENTS**



# Chapter No. Topic Page No.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. |  |  | Introduction | 3 |
|  | 1.1 |  | Purpose of this Document | 3 |
|  | 1.2 |  | Scope of the Development Project | 3 |
|  | 1.3 |  | Definitions, abbreviations and acronyms | 4 |
|  | 1.4 |  | References | 5 |
|  | 1.5 |  | Overview | 5 |
| 2. |  |  | Overall Description | 6 |
|  | 2.1 |  | Product Perspective | 6 |
|  | 2.2 |  | Product functions | 7 |
|  | 2.3 |  | User Characteristics | 7 |
|  | 2.4 |  | General Constraints, Assumptions and Dependencies | 7 |
|  | 2.5 |  | Apportioning of the requirements | 8 |
| 3. |  |  | Specific Requirements | 10 |
|  | 3.1 |  | External Interface Requirements | 10 |
|  | 3.2 |  | Detailed Description of Functional Requirements | 10 |
|  |  | 3.2.1 | Functional Requirements for Patient Chatbot | 10 |
|  |  | 3.2.2 | Functional Requirements for Doctor Booking Screen | 11 |
|  |  | 3.2.3 | Functional Requirements for Medicine Stock Tracker Screen | 11 |
|  | 3.3 |  | Performance requirements | 11 |
|  | 3.4 |  | Logical database requirements | 12 |
|  | 3.5 |  | Quality attributes | 12 |
|  | 3.6 |  | Other requirements | 12 |

1. Diagrams 13

|  |  |  |
| --- | --- | --- |
| 4.1 | Use Case Diagram | 13 |
| 4.2 | Use Case Template | 13 |
| 4.3 | Activity Template | 13 |
| 4.4 | Swimlane Diagram | 16 |
| 4.5  4.6  4.7  4.8  4.9  4.10  4.11  4.12 | Data Flow Diagram  User Story Card  Class Diagram  Sequence Diagram  Collaboration Diagram  State Chart Diagram  Component Diagram  Deployment Diagram | 17 |
| 5.  5.1  5.2  5.3  5.4 | Screenshots of project  Authentication  Patient  Doctor  Worker | 19 |
| 6.  7.  8. | Test Case Templates  Change History  Document Approvers | 19 |

# Introduction

## Purpose of this Document

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

* 1. **Scope of the Development Project**

The objective is to develop a comprehensive Healthcare Management System with a range of utility features, including Access Control and Patient Monitoring, tailored to meet the specific needs of students at colleges.

Within this system, users will log in to their individual accounts and gain access to features that are pertinent to their roles. Doctors will be able to view and manage patient medical records, as well as schedule future appointments. Patients, on the other hand, will have the capability to review their past appointments. All of the data accessed within the system will be securely stored in a centralized repository, namely a database server, and will adhere to DISHA standards to guarantee the confidentiality of Doctor-Patient interactions. The system will also provide a Stock Administrator with the tools necessary to update the medical database and monitor medication inventory. Patients will benefit from a chatbot feature designed to offer personal assistance, addressing common questions related to the platform and their health concerns.

In essence, this system serves as a comprehensive solution to address various healthcare management challenges, offering a unified platform for healthcare facilities and services.

The software must be able to perform the following operations:

1. **User Authentication and Access Control:** Users, including doctors, patients, and stock administrators, must log in to access the system. Access control ensures that each user has the appropriate permissions based on their role.
2. **Patient Monitoring:** The system allows doctors to monitor and manage patient medical data, including records, test results, and treatment plans, ensuring up-to-date and accurate patient information.
3. **Appointment Management:** Doctors can schedule, reschedule, or cancel patient appointments. Patients can view their upcoming and past appointments, enhancing the efficiency of healthcare services.
4. **Centralized Database:** All data, including patient records, appointment information, and medical inventory, is stored in a secure central repository (database server) for easy access and data integrity.
5. **DISHA Compliance:** The system ensures compliance with the Health Insurance Portability and Accountability Act (DISHA) to safeguard the privacy and confidentiality of doctor-patient interactions and medical records.
6. **Medical Database Management:** Stock administrators have the capability to update and manage the medical database, including adding new medicines, updating inventory levels, and ensuring the availability of necessary medical supplies. The stock managers can view the medicine requests of the patients and notify the latter on availability of the requested medicines.
7. **Patient-Doctor Communication:** Patients and doctors can communicate through secure channels within the system, facilitating timely and secure interactions for consultations and inquiries.
8. **Personalized Features:** The system provides personalized features tailored to the needs of individual users, ensuring that doctors, patients, and administrators have access to the functions relevant to their roles.
9. **Chatbot Assistance:** Patients have access to a chatbot feature that offers personal assistance by addressing common queries related to the platform and general health concerns, enhancing the user experience.
10. **Comprehensive Healthcare Solution:** The Healthcare Management System serves as a one-stop solution for managing healthcare facilities, patient data, appointments, and medication inventory, streamlining healthcare operations and services.

## Definitions, abbreviations and acronyms



**Definitions**

Table 1 gives explanation of the most commonly used terms in this SRS document.

**Table 1: Definitions for most commonly used terms**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Term** | **Definition** |
| 1 | DISHA | The Digital Information Security in Healthcare Act (DISHA) of India seeks to establish a National Digital Health Authority and Health Information Exchanges. The purpose behind the potential law is to facilitate electronic health data privacy, confidentiality, security, and standardization. [2]. |
| 2 | Website | A group of World Wide Web pages usually containing hyperlinks to each other and made available online by an individual, company, educational institution, government, or organization [3]. |
| 3 | API | An API, or application programming interface, is a set of defined rules that enable different applications to communicate with each other. It acts as an intermediary layer that processes data transfers between systems, letting companies open their application data and functionality to external third-party developers, business partners, and internal departments within their companies. [4]. |
| 4 | Framework | A framework is a collection of collaborating classes that provides a set of services for a given domain. You customize the framework to a particular application by subclassing and composing instances of the framework classes. [5]. |
| 5 | Chatbot | A chatbot is a computer program that uses artificial intelligence (AI) and natural language processing (NLP) to understand customer questions and automate responses to them, simulating human conversation. [6]. |



**Abbreviations**

Table 2 gives the full form of most commonly used mnemonics in this SRS document.

**Table 2: Full form for most commonly used mnemonics**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Mnemonic** | **Full Form** |
| 1 | DISHA | Digital Information Security in Healthcare Act [1] |
| 2 | API | Application Programming Interface |
| 3 | OTP | One Time Password |

## References



1. DISHA : <https://www.expresscomputer.in/security/data-privacy-in-india-how-does-it-affect-the-healthcare-industry/92656/#:~:text=Even%20before%20the%20recent%20Data,%2C%20security%2C%20confidentiality%20and%20standardisation>.
2. DISHA Meaning: <https://compliancy-group.com/disha-and-hipaa-how-do-they-compare/>
3. Website Definition: <https://www.merriam-webster.com/dictionary/website>
4. API Definition: <https://www.ibm.com/topics/api>
5. Framework Definition: <https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/design-rhapsody/9.0.1?topic=reference-frameworks-operating-systems>
6. Chatbot Definition: <https://www.ibm.com/id-en/topics/chatbots#:~:text=A%20chatbot%20is%20a%20computer,to%20them%2C%20simulating%20human%20conversation>

## Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 2 gives the functional requirements, data requirements and constraints and assumptions made while designing the multi-utility system. It also gives the user viewpoint of product use. Section 3 gives the specific requirements of the product. Section 3.0 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 consists of the Use Case, Use Case Template, Activity Diagram, Swimlane Diagram and Data Flow Diagram.

# Overall Description



## Product Perspective

The Health Care Portal is a comprehensive web-based platform designed to streamline healthcare interactions among three distinct user groups: Patients, Doctors, and Stock Managers. This system prioritizes secure access control, efficient appointment management, prescription handling, and medicine tracking within the healthcare ecosystem.

Users of the Health Portal are required to authenticate themselves using unique login credentials. In cases of forgotten passwords, users can conveniently reset their passwords through OTP (One-Time Password) verification.

Upon logging in, Patients are presented with a personalized dashboard that displays their name, a quote that is being fetched from an online Api, and a record of their total number of visitations and cancellations. There is a section for the patients to schedule appointments by selecting available dates and also have the option to cancel appointments. Access to past medical records is available for reference in the Past Records section, and a helpful chatbot is accessible for medical guidance, information on common medicines, and symptom advice. Patients also have a section to request for specific medicines from the Stock Manager, who receives notifications for prompt action.

Doctors, on the other hand, have access to their dedicated dashboard upon login, which displays their name and a personalized quote along with the total number of bookings they have for the day. Doctors have the authority to confirm or cancel appointments, with email notifications sent to patients. The system enables doctors to upload prescriptions once they confirm an appointment time for the patient. The uploaded prescription is also visible to the patient in the Past Record’s section. The doctor can even access patients' past medical records in case needed for a reference.

Stock Managers, too, have their own dashboard where they can view their name and a personalized quote. Their responsibilities include managing the stock of various medicines, with the ability to add, delete, and update stock levels. The system provides detailed tracking of medicine requests made by patients, and Stock Managers can send email notifications to patients once the requested medicines are available.

In summary, the Health Care Portal offers a secure, user-centric healthcare management solution that optimizes appointment scheduling, prescription handling, and medicine tracking. It enhances the overall healthcare experience for patients, facilitates efficient practice management for doctors, and ensures seamless medicine stock control for Stock Managers.

## Product Functions

The product should be able to perform the following operations:

1. The system must authenticate users, including Patients, Doctors, and Stock Managers, by verifying their login credentials against the stored user data in the database. This includes matching usernames and passwords or utilizing OTP verification for password resets.
2. Patients should be able to seamlessly schedule appointments with doctors by selecting available dates. Furthermore, patients should have the flexibility to cancel appointments when necessary, triggering notifications to both the patient and the Doctor.
3. Doctors, on their end, should be able to confirm or cancel appointments, with automated email notifications sent to patients, ensuring clear communication and scheduling accuracy.
4. Patients should be able to conveniently access their past medical records for reference and review. Meanwhile, Doctors should be able to efficiently manage their patients' medical records, including the addition of prescription information. This integrated approach should streamline patient care and ensure a comprehensive view of medical history.
5. Patients should be able to interact with the built in chatbot in the portal for medical guidance, information on common medicines, and symptom advice.
6. The health Portal should be able to facilitate a streamlined process for patients to request specific medicines from the Stock Manager. Stock Managers, in turn, should be able to effectively manage medicine stock. This includes adding, deleting, and updating medicine information as needed. The system should ensure that Stock Managers are promptly notified when Patients request medicines and that Patients are informed as soon as their requested medicines become available, ensuring timely and efficient medicine procurement.

## User Characteristics



The objective is to develop a comprehensive Healthcare Management System with a range of utility features, including access control and patient monitoring, tailored to meet the specific needs of students at colleges. The user types are listed below as follows:

* + 1. Students
    2. Doctors
    3. Student cum Staff
    4. Stock Manager
    5. Medical Assistant

As one can see from the list, each user will have different educational background and expertise level in using the system. Our goal is to develop a software that should be easy to use for all types of users. Thus, while designing the software one can assume that each user type has the following characteristics:

* The user is computer-literate and has little or no difficulty in using a website over the internet.
* In order to use the system, it is not required that a user be aware of the internal working of the system but he/she is expected to know what happens when they perform each of the operations on the website(which will have clear, easy to understand descriptions).
* The user has an email ID/phone number to receive all required communication regarding their bookings.

## General Constraints, Assumptions and Dependencies



The following list presents the constraints, assumptions, dependencies or guidelines that are imposed upon implementation of the Healthcare System that we plan to develop:

**2.4.1 Constraints:**

1. **Regulatory Compliance:** The software must comply with healthcare regulations and standards, such as DISHA in India or similar regulations in other countries.
2. **Data Security:** Ensuring the security and confidentiality of patient data is paramount. Robust encryption protocols must be implemented.
3. **Scalability:** The system should be scalable to accommodate a growing number of users, patients, and medical data.
4. **Data Backup and Recovery:** Regular data backups and disaster recovery

plans must be in place to prevent data loss.

1. **Resource Constraints:** Availability of server resources, bandwidth, and

hardware infrastructure can impact system performance.

1. **Budget and Time Constraints:** The project should be completed within the

allocated budget and time frame.

**2.4.2 Assumptions:**

1. **User Proficiency:** Users are assumed to have basic computer literacy to use the system effectively.
2. **Internet Access:** Users have reliable internet access to use the web-based system.
3. **Hardware Compatibility:** Users have access to devices (computers, smartphones, tablets) that are compatible with the system's web interface.
4. **Staff Training:** Medical staff and administrators will receive training on using the system effectively.
5. **Data Accuracy:** Data input into the system is assumed to be accurate, and data quality is the responsibility of the users.

**2.4.3 Dependencies:**

1. **Database Management System:** The software depends on a robust and reliable database management system for storing and retrieving patient data.
2. **Web Hosting Services:** The website must be hosted on dependable web hosting servers with adequate bandwidth and uptime.
3. **Third-Party APIs:** Integration with external services, such as laboratory result systems or insurance providers, may be necessary and dependent on the availability of their APIs.
4. **Development Frameworks and Tools:** The development of the software may depend on specific programming languages, frameworks, and development tools.
5. **Security Software:** Dependencies on security software and tools for ensuring data protection and compliance.
6. **User Support and Training:** The availability of user support and training resources is essential for onboarding and assisting users.
7. **Updates and Maintenance:** Regular updates and maintenance are required to keep the software functioning efficiently and securely.
8. **Payment Processing Services:** If the system involves online payments (e.g., for appointments or medication), it depends on payment processing services.

These constraints, assumptions, and dependencies should be thoroughly considered and documented during the planning and development phases of the Healthcare Management System to ensure its successful implementation and operation.

## Apportioning of requirements



The CSC based Multi-Utility System (including Access Control and Attendance Monitoring) is to be implemented in the following three phases:

1. **Pilot phase:** In the initial phase, the Health Portal will be meticulously deployed within the Thapar University campus, extending its benefits to students, professors, and the entire staff. This phase will serve as the foundation, ensuring the portal's efficiency, security, and user-friendliness.
2. **Expanding to colleges across Punjab:** Building upon the success of Phase 1, the Health Portal will expand its reach to multiple college campuses across Punjab. This phase aims to provide a regional healthcare solution, improving the well-being of students and staff in various educational institutions. In Phase 2, the portal will introduce enhanced features, such as collaborative medical record sharing and inter-campus appointment scheduling.
3. **Country wide deployment:** The third phase represents a significant milestone - the nationwide expansion of the Health Portal. Now, hospitals and healthcare institutions from all over the country can register themselves, transforming the platform into a comprehensive healthcare ecosystem.

# Specific Requirements

### External Interface Requirements



The following list presents the external interface requirements:

* The product requires very limited graphics usage with just a simple keypad for taking the user input.
* The product does not require usage of sound or animation. The hardware and operating system require a minimum screen resolution of 320 x 240 pixels (owing to the small form factor) in a Desktop setup and the website can also be accessed through any smart phone using a standard browser with a stable internet connection of at least 5-10 Mbps.
  1. **Detailed Description of Functional Requirements**

Table 3 shows a template that I’ll be using to describe functional requirements for three types of users: student, staff, student cum staff as one can easily deduce the functional requirements for other user types with this template.

**Table 3: Template for describing functional requirements**

|  |  |
| --- | --- |
| **Purpose** | A description of the functional requirements and its reasons |
| **Inputs** | What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input. |
| **Processing** | Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations |
| **Outputs** | The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced as output. |

* + 1. **Functional Requirements for Patient Chatbot**

Table 4 gives the functional requirements for Patient Chatbot.



**Table 4: Functional Requirements for Patient Chatbot**

|  |  |
| --- | --- |
| **Purpose** | To provide patients with medical assistance by judging the symptoms and suggesting them procedures to tackle small scale medical needs and guide them to take appropriate steps in case of a medical emergency. |
| **Inputs** | The patients can type in the symptoms they have and then press the ASK button. |
| **Processing** | The chatbot then checks if the user input is null or not, based on that the prompt entered by user is fed into the ML model via an API and the request is processed. |
| **Outputs** | The patient can get a brief overview about the disease he/she may be suffering from and what precautions should be taken. |

### Functional Requirements for Doctor Booking Screen

Table 5 gives the functional requirements for Doctor Booking Screen.

**Table 5: Functional Requirements for Doctor Booking Screen**

|  |  |
| --- | --- |
| **Purpose** | This screen displays all the appointments made by the patients. The doctor can then either confirm/cancel the appointment. The doctor also has the option to upload prescription for the patient as well. |
| **Inputs** | The doctor has the option to : cancel, accept, upload prescription. If the doctor presses cancel, an email notification is sent to the patient regarding the same. If the doctor presses accept, he/she will set an appropriate time for the appointment. After the doctor presses accept, a new option of “upload” pops up wherein the doctor can enter the required details to create a prescription. |
| **Processing** | Based on the type of button the doctor presses, the state of the UI of the card changes and the input is sent to the database . The SMTP server also boots up to send the required email to the patient. |
| **Outputs** | If the doctor accepts the appointment and sets the time, it is sent to the patient via email and is also visible on the patient’s dashboard. If the doctor presses cancel, an email notification regarding the cancellation of the booking is sent to the patient. If the doctor presses upload, he/she fills the required form and the prescription is added in the medical records section of both the patient and the doctor. |

### Functional Requirements for Medicine Stock Tracker Screen

Table 6 gives the functional requirements for Medicine Stock Tracker Screen.

**Table 6: Functional Requirements for Medicine Stock tracker screen**

|  |  |
| --- | --- |
| **Purpose** | This screen helps the stock manager to track the status of medicines and go through the students request for different kinds of medicines |
| **Inputs** | The stock manager can add, edit, delete the existing supply of the medicines. The stock manager can also view the medicine requests by the patient. |
| **Processing** | The basic CRUD(create, read ,update, deletion) operations take place with the help of an API. |
| **Outputs** | The medicine quantity is updated accordingly. In case a certain medicine has arrived in the clinic which was requested by the patient, the stock manager can send an email notification to the corresponding patient with a single click. |

* 1. **Performance Requirements**

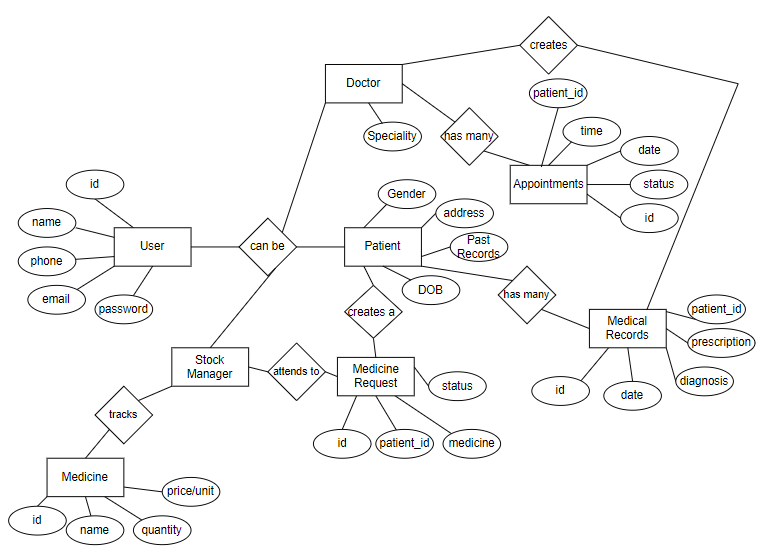
The software is designed for the smart card reader-writer terminal and cannot run from a standalone desktop PC.



* The software will support simultaneous user access only if there are multiple terminals.
* Only textual information will be handled by the software. Amount of information to be handled can vary from user to user.
* For normal conditions, 95% of the transactions should be processed in less than 5 seconds.

## Logical Database Requirements

**Figure 3**: E-R Diagram for the Health Care Portal





## Quality Attributes

The product is target towards a wide variety of users such as Student, staff, student cum staff, etc. The product must load quickly and work well on a variety of terminals. It must also tolerate wide variety of input possibilities from a user, such as incorrect responses or unforeseen keystrokes.



## Other Requirements

None at this time

# Diagrams

# Use Case Diagram

# 

# Use Case Templates

|  |  |
| --- | --- |
| 1. Use Case Title | Booking an Appointment |
| 2. Abbreviated Title | Book Appointment |
| 3. Use Case ID | 1 |
| 4. Actors | Patient |
| 5. Description:With this feature the patients(students, faculty, teacher cum students etcetera) will be able to book their appointments according to their preferred time and date. Cancellation of appointments can also happen by the patient. | |
| 5.1. Pre-Conditions:User must be logged into their patient account. | |
| 5.2. Task Sequence:The option to book an appointment will be visible in the dashboard. Th user needs to click the button.Enter their full name, preferred time and date in the form that is displayed.Click the submit button to confirm their appointment request.The appointment will show up on the bookings section from which the patient can cancel their appointment with a single click of a button. | |
| 5.3. Post Conditions:Patient will successfully send their appointment request to the doctor.A confirmation email will be sent to their registered email id to show the request was sent to the doctor. | |
| 6. Modification History: | |
| 7. Author: Shreeya Chatterji | |

|  |  |
| --- | --- |
| 1. Use Case Title | See Past Records |
| 2. Abbreviated Title | Past Records |
| 3. Use Case ID | 2 |
| 4. Actors | Patient, Doctor |
| 5. Description:This component of our project will allow patients and doctors to see the past records. The patient will have access to all their previous prescriptions and appointment details. The doctors will be able to see the same for all their patients which will be assisted using a search bar to scan through all the data. | |
| 5.1. Pre-Conditions:User needs to be logged in.User needs to have had booked previous appointments to access the full capacity of this feature. | |
| 5.2. Task Sequence:Click on the Past records tab in the navbar to see all past records.To cancel an appointment user will have to click on the cancel button on the appointment they want to delete.For doctors they can confirm an appointment once a request is sent by any patient using the “confirm” button equipped with each appointment.To upload a prescription for a confirmed appointment the doctor will have to click the “upload” button which will open a form for them to type in their prescription.Once the doctor has uploaded a prescription the corresponding patient will have access to the prescription as well as the stock manager who is responsible for handing out the medicines. | |
| 5.3. Post Conditions:View the prescription by stock manager, patient and doctor.Upload prescriptions by doctor. | |
| 6. Modification History: | |
| 7. Author: Shreeya Chatterji | |

|  |  |
| --- | --- |
| 1. Use Case Title | Medicine Stock Update |
| 2. Abbreviated Title | Add/Remove Medicine |
| 3. Use Case ID | 3 |
| 4. Actors | Stock Manager |
| 5. DescriptionThis component of our portal will help keep track of the stock of medicines in the clinic. The stock manager can view, update and delete the medicines information. | |
| 5.1. Pre-Conditions:User needs to be logged in. | |
| 5.2. Task Sequence:1. Click on the Stocks section from the navbar2. The stock manager will be able to see all the medicines alongwith the quantity and price/unit of each medicine.Click on the Update button (present next to each medicine)User can then enter the updated stock quantity by typing in the new count and pressing the save button. | |
| 5.3. Post Conditions:The quantity is updated and the stock manager will see the new count next time they visit the Stocks section. | |
| 6. Modification History: | |
| 7. Author: Harsh Jain | |

|  |  |
| --- | --- |
| 1. Use Case Title | Interacting with the chatbot |
| 2. Abbreviated Title | ChatBot interaction |
| 3. Use Case ID | 4 |
| 4. Actors | Patient |
| 5. DescriptionTo provide patients with medical assistance by judging the symptoms and suggesting them procedures to tackle small scale medical needs and guide them to take appropriate steps in case of a medical emergency. | |
| 5.1. Pre-Conditions:1. User needs to be logged in. | |
| 5.2. Task Sequence:1. The patients can click on the chat icon present on the bottomright corner of the pageThe chat window pops up. The patient can then type in the symptoms they have and then press the ASK button.The chatbot then checks if the user input is null or not, based on that the prompt entered by user is fed into the ML model via an API and the request is processed. | |
| 5.3. Post Conditions:1. The patients can get a brief overview about the disease theymight be suffering from and what precautions should betaken. | |
| 6. Modification History: | |
| 7. Author: Harsh Jain | |

# Activity Diagram

# 

# Swimlane Diagram

# 

# Data Flow Diagram

# LEVEL 0 DFD

# 

# LEVEL 1 DFD

# 

# LEVEL 2 DFD

# 

# User Story Card

# Login Page

# 

# 

# 

# Registration Page

# 

# 

# Booking Popup

# 

# 

# Update Stock

# 

# 

# Add Prescription

# 

# 

# View Past Records

# 

# 

# Class Diagram

# 

# Sequence Diagram

# 

# Collaboration Diagram

# 

# State Chart Diagram

# 

# Component Diagram

# 

# Deployment Diagram

# 

# Screen Shots of project

# 5.1 Authentication

# 

# 

# 5.2 Patient

# 

# 

# 

# 

# 

# 5.3 Doctor

# 

# 

# 

# 

# 

# 

# 5.4 Worker

# 

# 

# Test Case Templates

|  |  |
| --- | --- |
| **Test Case #: 1** | **Test Case Name:** Login Page |
| **System:** Authentication System | **Subsystem:** Login Page |
| **Designed by:** Shreeya Chatterji | **Design Date:** 16/08/2023 |
| **Executed by:** Shreeya Chatterji | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for people to login | |

**Pre-conditions:**

1. The user should have an account by having registered before
2. The user should remember their login details correctly
3. User should have a stable internet connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Enter registered email ID | Field to fill password opens |  |  |
| 2 | Enter registered password | Login is allowed |  |  |
| 3 | Click the login button | The dashboard opens successfully |  |  |
| 4 | **Check post-condition 1** |  | Pass |  |
| 5 | Repeat steps 1,2,3 using a wrong password | The system displays a message informing that the password is **incorrect** |  |  |
| 6 | **Check post-condition 2** |  | Pass |  |
| 7 | Repeat steps 1,2,3 using an unregistered email ID | The system displays a message informing that the email is **unregistered** |  |  |
| 8 | **Check post-condition 2** |  | Pass |  |

**Post-conditions:**

1. The user is able to login successfully
2. Login was unsuccessful and error message is displayed

|  |  |
| --- | --- |
| **Test Case #: 2** | **Test Case Name:** Login Page |
| **System:** Authentication System | **Subsystem:** Registration Page |
| **Designed by:** Shreeya Chatterji | **Design Date:** 16/08/2023 |
| **Executed by:** Shreeya Chatterji | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for people to register | |

**Pre-conditions:**

1. User should have a stable internet connection.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Enter your name | Field to fill email opens |  |  |
| 2 | Enter email ID | Field to fill password opens |  |  |
| 3 | Enter your desired password | Field to confirm password opens |  |  |
| 4 | Enter the same password | Signup is allowed |  |  |
| 5 | Click on Sign Up button | Signup is successful |  |  |
| 6 | **Check post-condition 1** |  | Pass |  |
| 7 | Repeat steps 2,3,4,5 using without filling the name | The system displays a message informing that registration was **unsuccessful** |  |  |
| 8 | **Check post-condition 2** |  | Pass |  |
| 9 | Repeat steps 1,3,4,5 using without filling email id | The system displays a message informing that registration was **unsuccessful** |  |  |
| 10 | **Check post-condition 2** |  | Pass |  |
| 11 | Repeat steps 1,2,4,5 using without filling email id | The system displays a message informing that registration was **unsuccessful** |  |  |
| 12 | **Check post-condition 2** |  | Pass |  |
| 13 | Repeat steps 1,2,3,5 using without filling email id | The system displays a message informing that registration was **unsuccessful** |  |  |
| 14 | **Check post-condition 2** |  | Pass |  |
| 15 | Repeat steps 1,2,3,4 using without filling email id | The system displays a message informing that registration was **unsuccessful** |  |  |
| 16 | **Check post-condition 2** |  | Pass |  |

**Post-conditions:**

1. Message displaying message for successful registration is displayed and the data is entered into the database.
2. Signup was unsuccessful and error message is displayed.

|  |  |
| --- | --- |
| **Test Case #: 3** | **Test Case Name:** Book appointment |
| **System:** Appointment booking | **Subsystem:** Booking popup |
| **Designed by:** Shreeya Chatterji | **Design Date:** 16/08/2023 |
| **Executed by:** Shreeya Chatterji | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for people to book an appointment | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should have a valid login id and password
3. User should be logged in
4. User should have clicked on the book an appointment button

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Enter the date | Field to add remarks opens |  |  |
| 2 | Enter remarks (problems) | User can now click confirm |  |  |
| 3 | Click confirm | Signup is allowed |  |  |
| 4 | **Check post-condition 1 and 2** |  | Pass |  |
| 5 | Repeat steps 1,2 and click cancel. | A message would be displayed |  |  |
| 6 | **Check post-condition 4** |  | Pass |  |
| 7 | Repeat steps 1,2 without filling the date | A message would be displayed |  |  |
| 8 | **Check post-condition 3** |  | Pass |  |
| 9 | Repeat steps 1,2 without filling the date | A message would be displayed |  |  |
| 11 | **Check post-condition 3** |  | Pass |  |

**Post-conditions:**

1. Message displaying message for successful booking is displayed and the data is entered into the database.
2. Successful booking data is displayed in the bookings tab.
3. Error message showing that there are empty fields is displayed.
4. Message saying that booking was cancelled is displayed.

|  |  |
| --- | --- |
| **Test Case #: 4** | **Test Case Name:** Medical Records |
| **System:** Medical records page | **Subsystem:** View prescription |
| **Designed by:** Shreeya Chatterji | **Design Date:** 16/08/2023 |
| **Executed by:** Shreeya Chatterji | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for people to book an appointment | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should have a valid login id and password
3. User should be logged in
4. User should have clicked on the Bookings tab

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Scroll to view bookings | Bookings displayed in order of latest appointments |  |  |
| 2 | Click on cancel button to cancel a specific appointment | Loading button displayed and message for cancelled booking is shown |  |  |
| 3 | **Check post-condition 1 and 2** | Signup is allowed | Pass |  |

**Post-conditions:**

1. The appointment is cancelled and removed from the database.
2. The appointment is no longer visible in the bookings tab.

|  |  |
| --- | --- |
| **Test Case #: 5** | **Test Case Name:** Chatbot |
| **System:** Chatbot | **Subsystem:** Chatbot |
| **Designed by:** Shreeya Chatterji | **Design Date:** 16/08/2023 |
| **Executed by:** Shreeya Chatterji | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface to access chatbot | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should have a valid login id and password
3. User should be logged in
4. User should have clicked on the chatbot icon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Enter your query | A response is displayed by the chatbot |  |  |
| 2 | **Check post-condition 1** |  | Pass |  |

**Post-conditions:**

1. A response is generated by the chatbot from it’s pre-trained model and if an answer match is not found then an error message is displayed by the chatbot itself.

|  |  |
| --- | --- |
| **Test Case #: 6** | **Test Case Name:** Confirm appointment |
| **System:** Confirm appointment | **Subsystem:** Set Time Dialog Box |
| **Designed by:** Harsh Jain | **Design Date:** 16/08/2023 |
| **Executed by:** Harsh Jain | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface to view and confirm appointments | |

**Pre-conditions:**

1. The user should be logged in
2. The user should be a doctor
3. There should be appointments made by the patients.
4. User should fill appropriate time for booking
5. User should have a stable internet connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Click on the Bookings Section | A list of appointments open |  |  |
| 2 | Select a patient’s booking | Accept and Cancel Buttons are displayed |  |  |
| 3 | Click on the Accept Button | A dialog box pops up asking for the time |  |  |
| 4 | Fill in the preferred time | The system stores the typed info in console |  |  |
| 5 | **Check pre-condition 3** |  | Pass |  |
| 6 | Click the Submit button | The time is saved in the database. |  |  |
| 7 | **Check post-condition 2** |  | Pass |  |

**Post-conditions:**

1. The doctor is able to successfully set the time for the appointment
2. An email notification is received by the patient regarding the appointment timings.

|  |  |
| --- | --- |
| **Test Case #: 7** | **Test Case Name:** Update Stock Details |
| **System:** Stock Update | **Subsystem:** Update Stock Dialog Box |
| **Designed by:** Harsh Jain | **Design Date:** 16/08/2023 |
| **Executed by:** Harsh Jain | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for workers to update stock details | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should be logged in
3. User should be a worker

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Open the Stock section | A list of medicines along with their quantity open up |  |  |
| 2 | Click on Update button for any medicine | A pop-up opens asking for the new medicine count |  |  |
| 3 | Fill in the appropriate data and press submit | Updated medicine count stored in database |  |  |
| 6 | **Check post-condition 1** |  | Pass |  |

**Post-conditions:**

1. Updated medicine count should be showed in the Stock Page

|  |  |
| --- | --- |
|  |  |
| **Test Case #: 8** | **Test Case Name:** Add medical record |
| **System:** | **Subsystem:** Prescription popup |
| **Designed by:** Harsh Jain | **Design Date:** 16/08/2023 |
| **Executed by:** Harsh Jain | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface to add medical records | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should be logged in.
3. Doctor should have a confirmed appointment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Open the Booking Section | A list of confirmed and unconfirmed booking appointments is displayed |  |  |
| 2 | Click on a confirmed booking | The Cancel and Upload button are now shown |  |  |
| 3 | Click on the Upload button | A prescription form opens up |  |  |
| 4 | Fill the fields and press submit | The medical record for that patient is added into the database |  |  |
| 5 | **Check post-condition 2** |  | Pass |  |
| 6 | **Check post-condition 1** |  | Pass |  |

**Post-conditions:**

1. The medical record is added for both the patient and the doctor to view in the future.
2. The appointment is removed from the Booking section of both the patient and the doctor.

|  |  |
| --- | --- |
| **Test Case #: 9** | **Test Case Name:** Request for medicine |
| **System:** Patient Request for medicine | **Subsystem:** Request medicine popup |
| **Designed by:** Harsh Jain | **Design Date:** 16/08/2023 |
| **Executed by:** Harsh Jain | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface for people to request for medicines to the worker and doctor | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should be logged in
3. User should be patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Click on the Request Tab | The Requests section for the patient opens up listing all his past and current requests |  |  |
| 2 | Click on the Request for the medicine button | A dialog box pop ups with multiple fields for the user to fill in |  |  |
| 3 | Click on submit | The medicine request is generated and added to the database. |  |  |
| 4 | **Check post-condition 1** |  | Pass |  |

**Post-conditions:**

1. The medicine is request is visible to both the patient and the worker’s screen.

|  |  |
| --- | --- |
| **Test Case #: 10** | **Test Case Name:** Cancellation of appointment |
| **System:** Cancellation of appointment by doctor | **Subsystem:** Cancellation |
| **Designed by:** Harsh Jain | **Design Date:** 16/08/2023 |
| **Executed by:** Harsh Jain | **Execution Date:** 16/09/2023 |
| **Short Description:** Interface to view and cancel appointments | |

**Pre-conditions:**

1. User should have a stable internet connection
2. User should be logged in
3. User should be a doctor
4. User should have the bookings page open

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Click on any one of the unconfirmed appointments | Two options pop up i.e. Accept and Cancel |  |  |
| 2 | Click on Cancel | The booking for that patient disappears |  |  |
| 3 | **Check PostCondition-1** |  | Pass |  |
| 4 | **Check post-condition 2** |  | Pass |  |

**Post-conditions:**

1. Booking removed from the doctor’s view and a “Cancelled” text written in the patient’s booking card in his/her dashboard
2. Email notification sent to the patient regarding his appointment cancellation

# Change History



Version 1.0 – Initial Release

200209

## Document Approvers

SRS for Health Care Portal approved by:



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(name)**

**Designation:**

**Date:**