MODEL VIEW CONTROLLER ARCHITECTURE FOR AN AUTOMATED SYSTEM TO CONNECT PATIENTS AND HOSPITALS BASED ON SERVICE-ORIENTED ARCHITECTURE

Manas Trivedi (181CO231), Omanshu Mahawar (181CO237)
manasdtrivedi@gmail.com, omanshumahawar1234@gmail.com

National Institute of Technology Karnataka, Surathkal, India

Abstract. Model View Controller (MVC) is an architecture which divides an application into three interconnected components - model, view, and controller. This architecture has been widely adopted in developing web applications. In the domain of healthcare, software systems must be fast and lightweight, since such systems may be used in time-critical situations. This document focuses on creating an MVC architecture for a service-oriented system which connects hospitals and patients, with a focus on users residing in remote areas.

Keywords: Model view controller · Healthcare · Design · Architecture

1. Introduction

1.1 Purpose

This document contains the design and architecture of 'An Automated System to Connect Patients and Hospitals' (ASCPH) based on the Model View Controller (MVC) approach. This document follows the functionalities and requirements identified in the SRS document of the project [2]. The primary audiences of this document are the software developers.

1.3 Definitions, acronyms, and abbreviations

Table 1: Definitions

Term	Definition
Appointment	Arrangement to meet a doctor working at a hospital at a particular time
Hospital Executive	A person who handles the updation of hospital details and approval and rejection of appointments.
Patient	A person who uses ASCPH to access services such as finding a hospital and booking appointments for the treatment of diseases.
Service	A self-contained unit of software that performs a specific task.

Table 2. Acronyms and abbreviations

ASCPH	Automated System to Connect Patients and Hospitals
HEx	Hospital Executive
ID	Identification
MVC	Model View Controller

2 Model View Controller

The MVC architecture divides an application into three parts, namely Model, View, and Controller. The model is used to manage the data used by the application. Views show the information present in the model to the user. The controller takes user input and performs operations on the model. The user interface of the application is dynamically created by the views and controllers. MVC follows the concept of layering, which breaks down code into functions into different classes. The main advantage of this approach is the reusability of the code [1].

2.1 Model

The model is used to manage information and notify observers when there is a change of information. The model is also a component used to retrieve data from a database/data source. The functions contained in the Model will be invoked by the Controller. Essentially, the Model is the component that interacts with the database to handle data, logic and rules. It not only captures the state of the process or the system but how the system works. Programmers can define models that contain functions that close the gap between the back-end and the user interface.

2.2 View

The view is the component responsible for the presentation to the user. Layer view is what is commonly called web design or templates. View controls how data is displayed and how users interact with it. View also provides a way of collecting data in the form of input from users. The technologies that are primarily used in view are HTML, CSS, and JavaScript. As a general rule, views should not contain application logic elements. This decouples the views from the back-end. This means logical blocks, which perform business functions, must be sparingly used. View attaches to the model and renders its contents to the surface of the screen. Additionally, when the model changes, the view automatically redraws the affected part of the screen to indicate the change.

2.3 Controller

The controller is a component that serves to call the function in the model and send the results through View. The controller also takes input from the user which will then be processed by Model. Thus, the controller is responsible for mapping the final user action against the application response. The controller sends commands to the model to update the data and also send commands to the view to modify the data received or displayed. When a request arrives on the server, the MVC framework sends it to a method in the controller based on the URL.

3 Architecture

3.1 Models

Patients Model

This model stores the data of patients who register on the Web Application. This model stores Patient ID, Password, Name, Age, and Gender of the patient.

Registered Hospitals Model

This model stores information about the hospitals registered by the HEx. Details included in this model are Hospital ID, Password, Name, Location, Working Hours, Facilities provided by the hospital.

Appointments Model

This model stored all the Appointment requests which are currently active. This model stores Appointment ID, Hospital ID of the hospital for which appointment is requested, PatientID of the patient who has made the request, Disease for which treatment is intended, Status of the appointment which can be either of the following - requested, approved, rejected, or cancelled.

3.2 Views

Hospital Search View

This view allows the patient to find and locate hospitals. The user may select requirement(s) such as treatment of Covid-19, treatment of Asthma, etc. Based on the selection made, a list of nearest hospitals will be displayed to the patient.

Hospital Details View

This view displays the details of the particular hospital selected by the patient. All the details of the hospital including the location and types of treatment available at the hospitals is displayed.

Book Appointments View

This view allows the patient to book an appointment at the chosen hospital. This view provides the patient with a form which is required to be submitted in order to book an appointment.

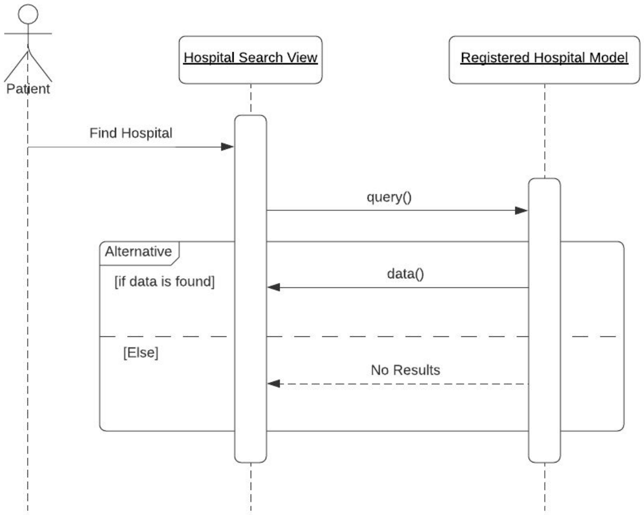
Patient Dashboard View

This view shows the details related to the account of the patient. This view also displays the status of the appointments requested by the patient with the option to cancel the appointment.

HEx Dashboard View

This view will be used by the HEx for managing the hospital details, such as availability of treatment for specific diseases, type and number of available resources, etc., and appointments requested by patients with the hospital. The HEx shall be permitted to approve or reject appointments. If the HEx decides to approve an appointment, the HEx must also specify the time of the appointment.

3.3 Controllers



Sequence Diagram 1. Find Hospital and Hospital Details Controller

Find Hospital Controller

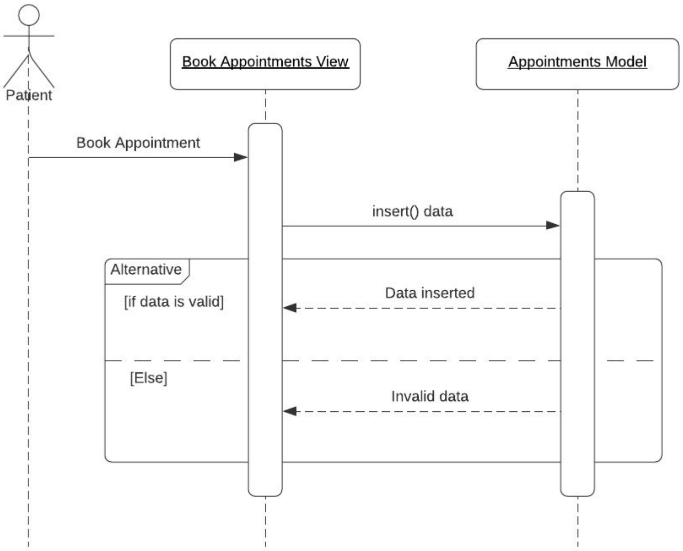
This controller is responsible for retrieving the list of nearby hospitals which satisfy the certain filters applied by the patient. It takes the filters as the input from the Hospital Search View and searches for the hospitals in the Registered Hospitals Model which satisfy the given filters. The result is provided to the Hospital Search View which displays the list of hospitals.

Hospital Details Controller

This controller takes hospital name as the input from the Hospital Search View and retrieves the details of the hospital from the Registered Hospitals Model. The retrieved data is provided to the Hospital Details View which displays the data.

Add Appointment Controller

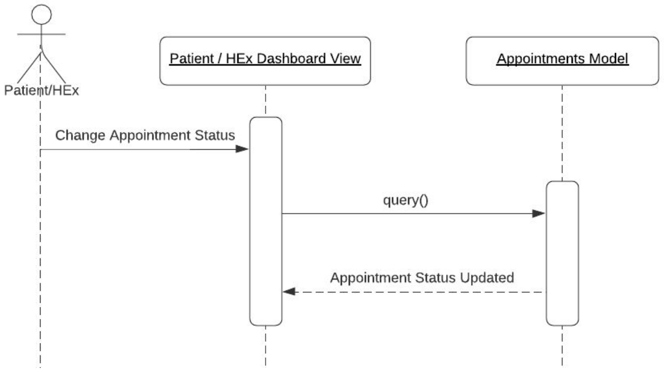
This controller adds a new entry into the Appointments model. This model takes details of the new appointment request as input from the Book Appointments View. The new record of the appointment is inserted into the Appointments Model with the initial status of the appointment request as "Requested".



Sequence Diagram 2. Add Appointment Controller

Change Appointment Status Controller

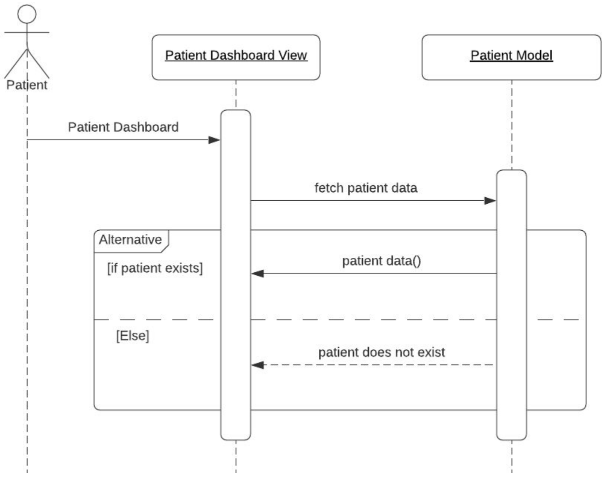
This controller is used to change the status of the appointment by the patient or the HEx. This controller takes AppointmentID as input from the patient dashboard or the HEx dashboard and updates the status of the appointment in the Appointments Model.


Patient/HEx
Patient / HEx Dashboard View
Appointments Model
Change Appointment Status
query()
Appointment Status Updated

Sequence Diagram 3. Change Appointment Status Controller

3.3.5 Patient Details Controller

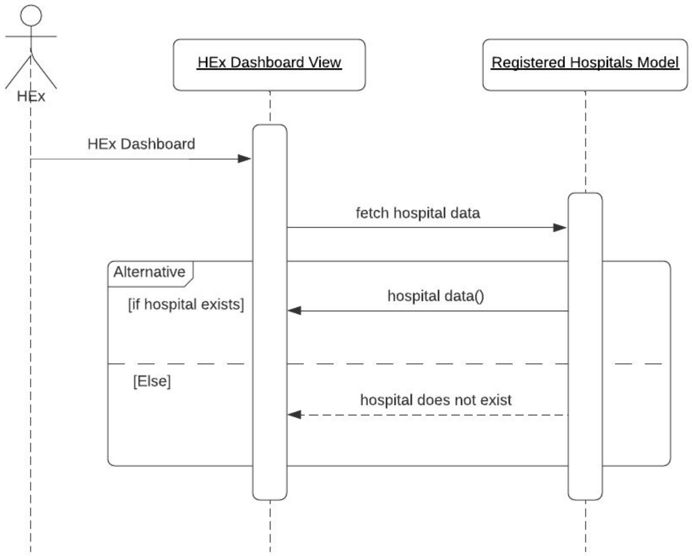
This controller retrieves the details of the patient from the Patients Model and provides it to the Patient Dashboard View.



Sequence Diagram 4. Patient Details Controller

3.3.6 HEx DashBoard Controller

This controller retrieves the details used by the HEx for managing the hospital and provides it to the HEx Dashboard View.



Sequence Diagram 5: HEx DashBoard Controller

Conclusion

An MVC architecture for ASCPH has been described in this document. The models used in the application are the Patient, Hospital and Appointment models. The different views which are presented to the users have been mentioned. The controllers which use the views and models to perform operations of the ASCPH have been discussed.

References

1. Hapsari, R.K., Azinar, A.W., Sugiyanto: Architecture Application Model View Controller (MVC) in Designing Information System of MSME Financial Report. In: Quest Journals Journal of Software Engineering and Simulation, vol. 3, pp 36-41 (2017). doi: 10.17605/osf.io/3z9r7
2. Trivedi, M., Mahawar, O.: Software Requirements Specification for An Automated System to Connect Patients and Hospitals Based on Service-Oriented Architecture (2020)