



Course Code : CSE 404

Course Title : Software Engineering and ISD Laboratory

Project name : Bus ticket booking management system

Experiment no: 06

Experiment name: Application of MVC Model of bus ticket management system

**Submitted To**

**Dr. Mohammad Zahidur Rahman**

Professor

**Dr. Md. Humayun Kabir**

Professor

Department of Computer Science & Engineering  
Jahangirnagar University, Savar.

Group No : 02

Group members :

Sl	Class Roll	Name
01	342	Tama Shil
02	370	Prokash Maitra
03	374	Mubasher adnan jihad
04	375	Pritam Saha

## **Mubasher Adnan Jihad**

**Roll:374**

MVC stands for

### **Model:**

The Model represents the application's data and business logic.

It is responsible for managing and processing data, as well as enforcing the rules and logic that govern how the data should be manipulated.

The Model is independent of the user interface and is not concerned with how data is presented to the user or how user input is handled.

In many applications, the Model interacts with a database or other data storage mechanisms to retrieve and store data.

Example, the Model might have classes like Bus, Route, Schedule, and Ticket, along with methods to create, read, update, and delete records for these entities in the database.

### **View:**

The View is responsible for presenting the data to the user. It handles the user interface and visual elements of the application, such as displaying information, collecting user input, and rendering graphical elements.

Views are typically passive components, meaning they don't contain application logic or manipulate data directly. Instead, they observe the data provided by the Model and update themselves accordingly.

In a well-designed MVC application, you can have multiple Views for the same data, allowing for different ways of presenting and interacting with it (e.g., a graphical interface and a command-line interface).

For instance, the View might consist of web pages that display available bus routes, schedules, and allow users to select and purchase tickets. When a user interacts with the UI, the View communicates with the Controller to handle the user's requests.

**Pritam Saha**

**Id:375**

**Controller:**

The Controller acts as an intermediary between the Model and the View. It receives user input from the View, processes it, and communicates with the Model to update data or retrieve information.

Controllers contain the application's logic for handling user interactions and orchestrating the flow of data between the Model and the View.

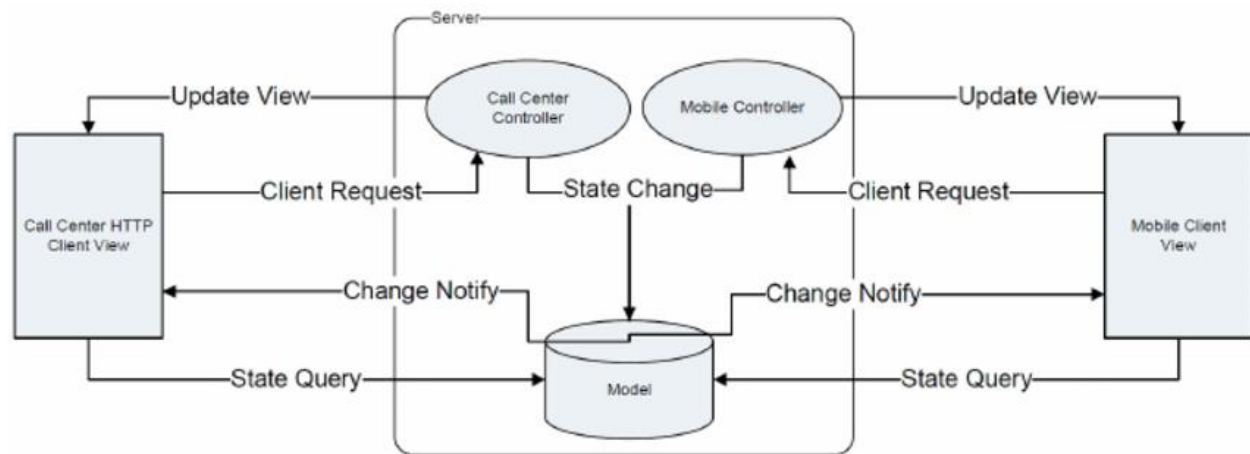
In many MVC implementations, Controllers are responsible for routing and managing requests, ensuring that the right data is fetched from the Model and the appropriate View is updated.

In the bus ticket management system, the Controller would handle tasks like processing user requests to search for available buses, book tickets, or cancel bookings. It would use the Model to retrieve relevant data and then pass that data to the appropriate View for rendering.

Considering Struts or Spring or Hibernate frameworks:

- Model: Java Beans (database)
- View: HTML or JSP page
- Controller: back-end servlet (converted JSP's)

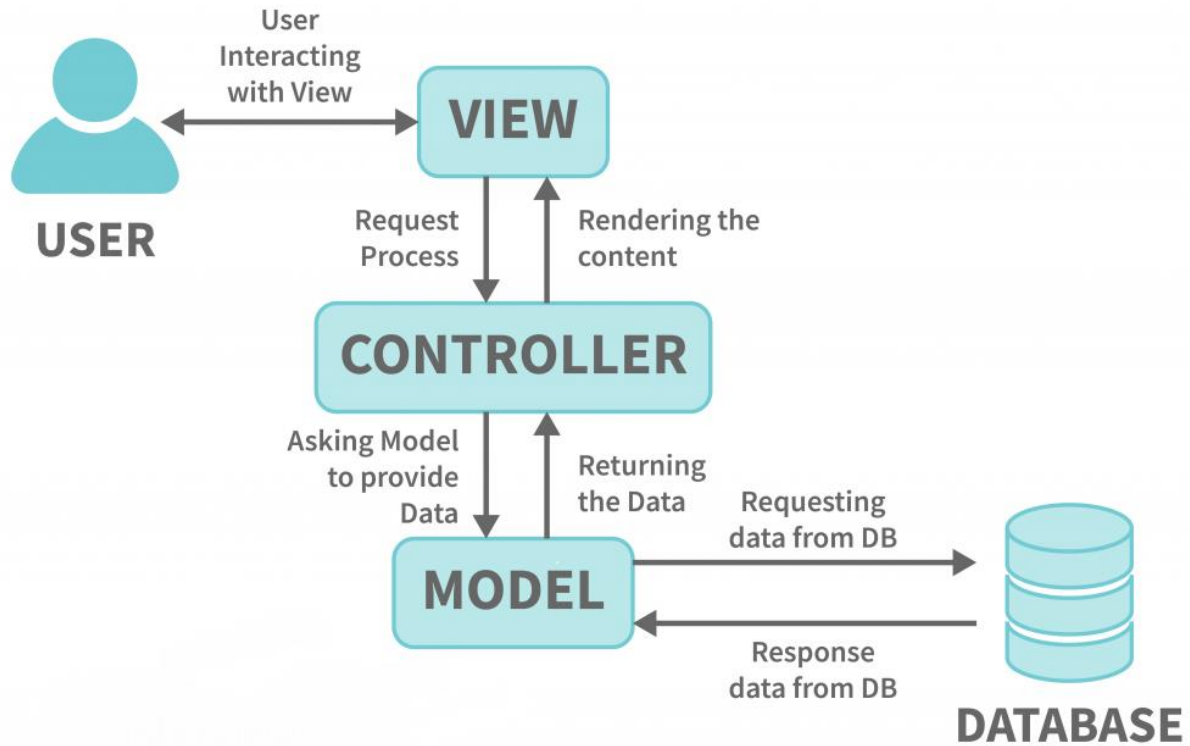
The figure below represents MVC in client-server:



**Prokash Mitra**

**Roll:370**

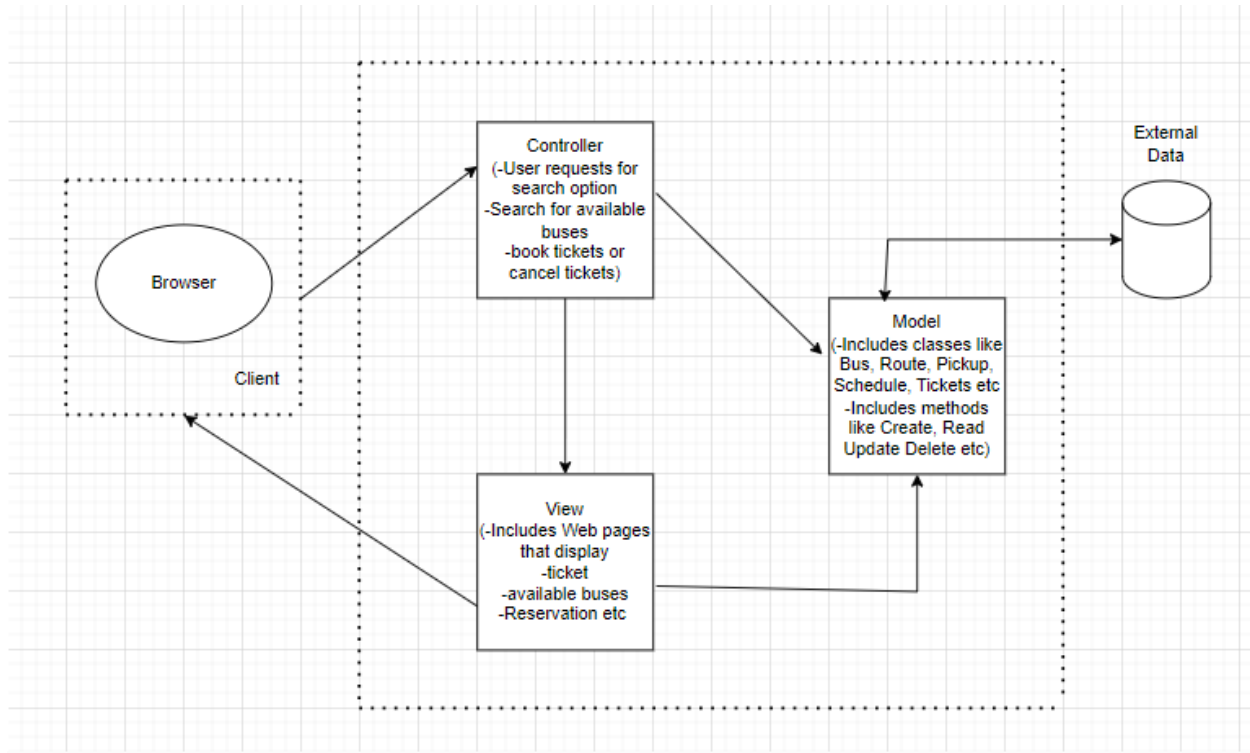
**MVC architecture:**



**Tama Shil**

**ID:342**

The drafted MVC for online bus ticket management system:





Course Code : CSE 404

Course Title : Software Engineering and ISD Laboratory

Project name : Bus ticket booking management system

Experiment no: 06

Experiment name: Application of MVC Model of bus ticket management system

**Submitted To**

**Dr. Mohammad Zahidur Rahman**

Professor

**Dr. Md. Humayun Kabir**

Professor

Department of Computer Science & Engineering  
Jahangirnagar University, Savar.

Group No : 02

Group members :

Sl	Class Roll	Name
01	342	Tama Shil
02	370	Prokash Maitra
03	374	Mubasher adnan jihad
04	375	Pritam Saha