KHULNA UNIVERSITY

Computer Science and Engineering Discipline

Course No. : CSE-3106
Course Name : Software Development Laboratory

Architecture Pattern

File Sharing Desktop Application

Submitted by:

- **Name:** Gayotry Gope Toma
  Student ID: 210212
- **Name:** Jannatul Ferdous Shova
  Student ID: 210228

Submitted to:
Dr. Amit Kumar Mondal
Associate Professor
Computer Science and Engineering
Khulna University, Khulna

Name of the Project:

File Sharing Desktop Application.

Architecture Pattern:

Model- View- Controller Software Architecture Pattern.

Description:

The Model-View-Controller (MVC) architecture pattern is a design pattern widely used in software engineering for creating user interfaces and organizing the codebase of applications. It divides an application into three interconnected components, each with distinct responsibilities:

Components:

- **Model:** The model represents the application's data and business logic. It encapsulates the data access, manipulation, and validation processes.It does not depend on the user interface or the controller, ensuring that changes to the user interface do not affect the underlying data structures.

- **View:** The view represents the presentation layer of the application and renders the user interface. It displays the data from the model to the user and captures user input events. Views are often implemented using templates, markup languages or user interface components (such as GUI widgets).

- **Controller:** The controller acts as an intermediary between the model and the view. It handles user input, processes requests, and updates the model accordingly. It receives input from the user via the view, invokes appropriate actions on the model, and updates the view with the latest data.Controllers control the flow of data. Controllers are often implemented as classes or components.

Interaction Flow:

1. **Initialization:** The user interacts with the application through the view, triggering an action.

2. **Controller Handling:** The controller receives the user input and decides which actions to perform based on the input and the current application state.

3. **Model Update:** The controller interacts with the model to retrieve or update data. It may perform data validation operations.

4. **View Update:** After updating the model, the controller updates the view to reflect the changes. It provides the view with the necessary data to render the user interface.

5. **User Interaction:** The updated view is displayed to the user, who can continue interacting with the application.

Diagram:

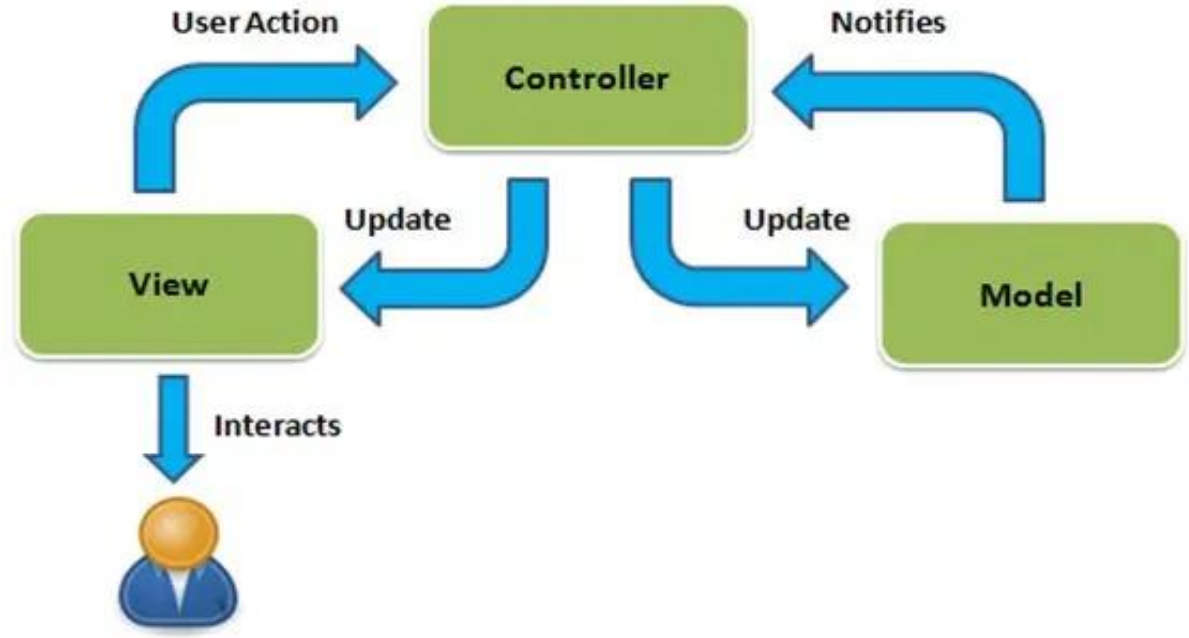

Figure: Model- View- Controller Architecture Pattern Diagram.