Architecture Pattern For Fototuner

A desktop based Photo Editing Application

Course No : CSE 3106

Course Title : Software Development Project

Submitted to :	Submitted by :
Dr. Amit Kumar Mondal **Associate Professor** **Computer Science and** **Engineering Discipline,** **Khulna University,Khulna**	**Name : Muzahidul Islam** **Student ID : 200228** **3rd Year 1st term** **Computer Science And** **Engineering Discipline,** **Khulna University, Khulna**

Project Title: Fototuner ,

A desktop-based photo editing Application.

Architecture Pattern I determined for:

Model-View-Controller(MVC) Pattern

The MVC architectural pattern offers a well-defined and structured approach to developing my photo editing application, aligning closely with the software's requirements for efficient image processing, intuitive user interface design, and seamless user interaction. By leveraging the clear separation of concerns provided by MVC, you can effectively address the complexities of building a feature-rich and user-friendly photo editing software while ensuring compatibility with diverse user needs and preferences.

Let’s explore MVC pattern in detail:

Model:

In this photo editing app, the Model component would encapsulate the core functionalities related to image processing, manipulation, and storage. This includes loading images from files, applying various filters and effects, adjusting parameters such as brightness and contrast, and saving edited images.

The Model serves as the backbone of the application, handling all image-related operations and ensuring data integrity and consistency throughout the editing process.

By separating these data-centric functionalities into a standalone component, the Model promotes code reuse, modularity, and ease of maintenance, making it compatible with the photo editing software's requirements for efficient image processing.

View:

The View component in your application would be responsible for presenting the user interface elements to the end-user. This includes displaying the original and edited images, providing intuitive controls for selecting filters and effects, and offering options for adjusting image properties.
The View ensures a seamless and engaging user experience by providing clear visual feedback and interactive elements for users to interact with the editing functionalities offered by the software.
By decoupling the presentation layer from the underlying business logic, the View enables flexibility in designing and customizing the user interface to meet the specific needs and preferences of your target audience, thus enhancing the compatibility of the photo editing software with diverse user environments and usage scenarios.

Controller:

The Controller acts as an intermediary between the Model and the View, orchestrating the flow of data and user interactions within the application. It receives user input from the View, processes it by invoking appropriate methods and operations on the Model, and updates the View to reflect the changes.
The Controller encapsulates the application's logic for handling user interactions, such as selecting

filters, adjusting image settings, and saving edited images. It ensures a cohesive and responsive user experience by synchronizing the actions performed by the user with the corresponding changes in the application state.

By centralizing the control logic in a separate component, the Controller promotes code maintainability, scalability, and testability, facilitating the seamless integration of new features and enhancements into the photo editing software while preserving compatibility with existing functionalities.

Graphical Representation of MVC pattern:

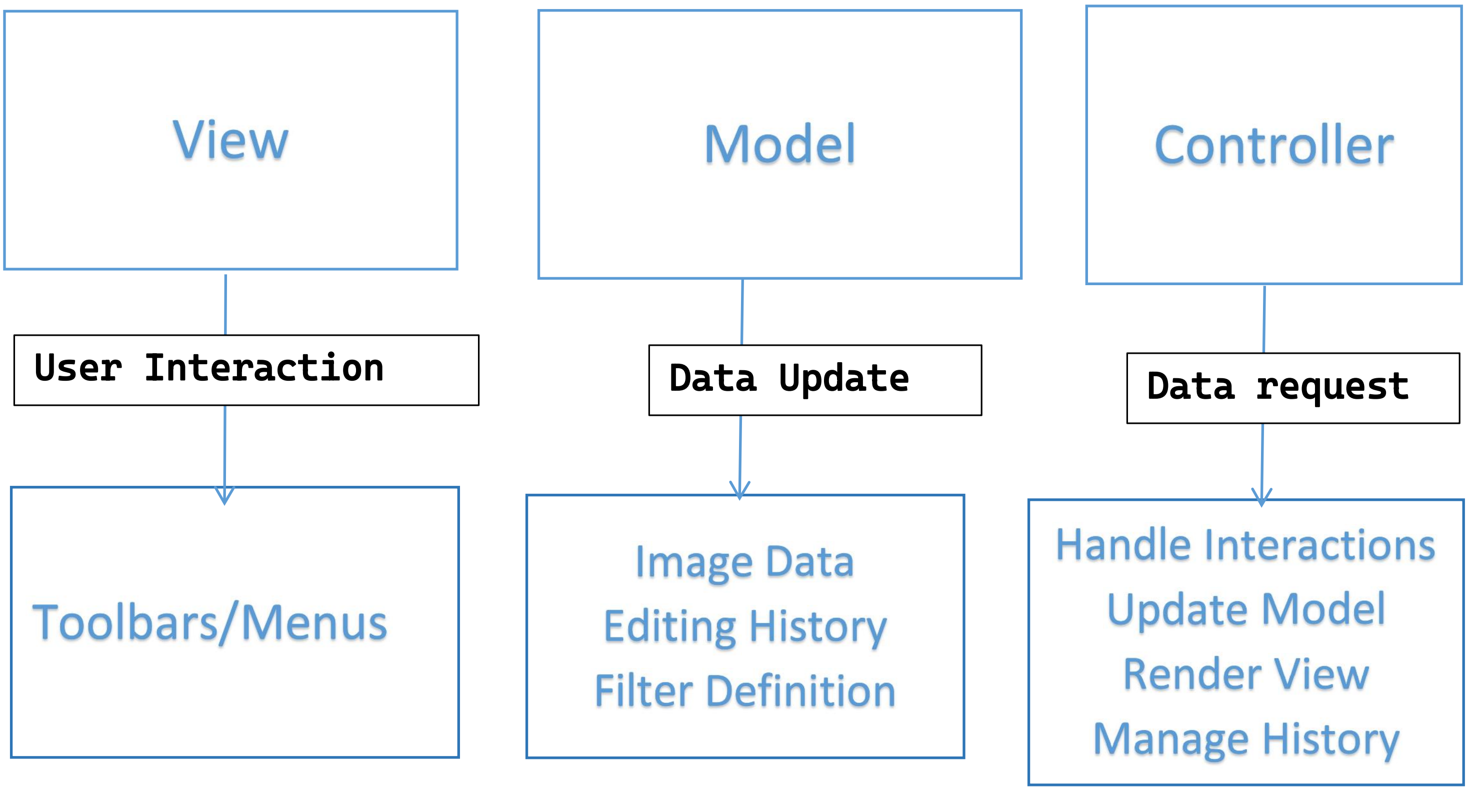

Fig: : Diagram of MVC in Fototuner