

Software project 3 (Report)

Project Name: Online Face Recognition Attendance System

Developer's Information:

Name: Masrafi Al Amin

Student Id: 221-15-5226

Section: 61_Q

Institution: Daffodil International University

Project Principle: Principle of Optimized Performance

This project addresses to the Principle of Optimized Performance, focusing on:

- Minimizing Rebuilds
- Optimizing Rendering
- Network and Data Optimization
- Memory Management
- UI Responsiveness
- Device Compatibility
- Continuous Monitoring and Optimization

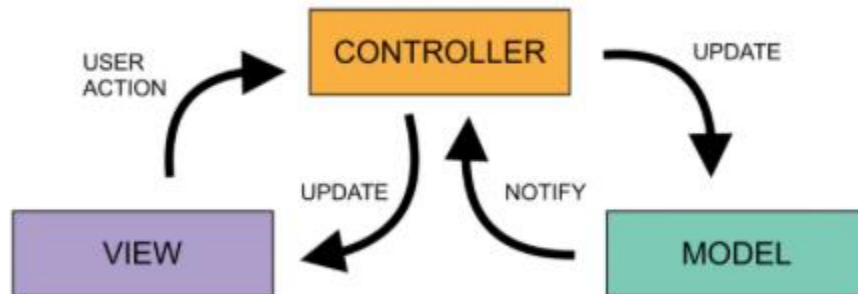
Project Architecture Patterns: To efficiently address common issues in software design, this project employs the MVC and MVP architectural patterns:

MVC Pattern:

The Model-View-Controller (MVC) pattern is a software design pattern that separates an application into three main components:

- **Model:** Handles state management and business logic. It directly manages the data, logic, and rules of the application.
- **View:** Responsible for rendering UI components. It displays the data to the user and sends user commands to the controller.

- **Controller:** Responds to UI actions and updates the view accordingly. It acts as an intermediary between the model and the view.



Example:

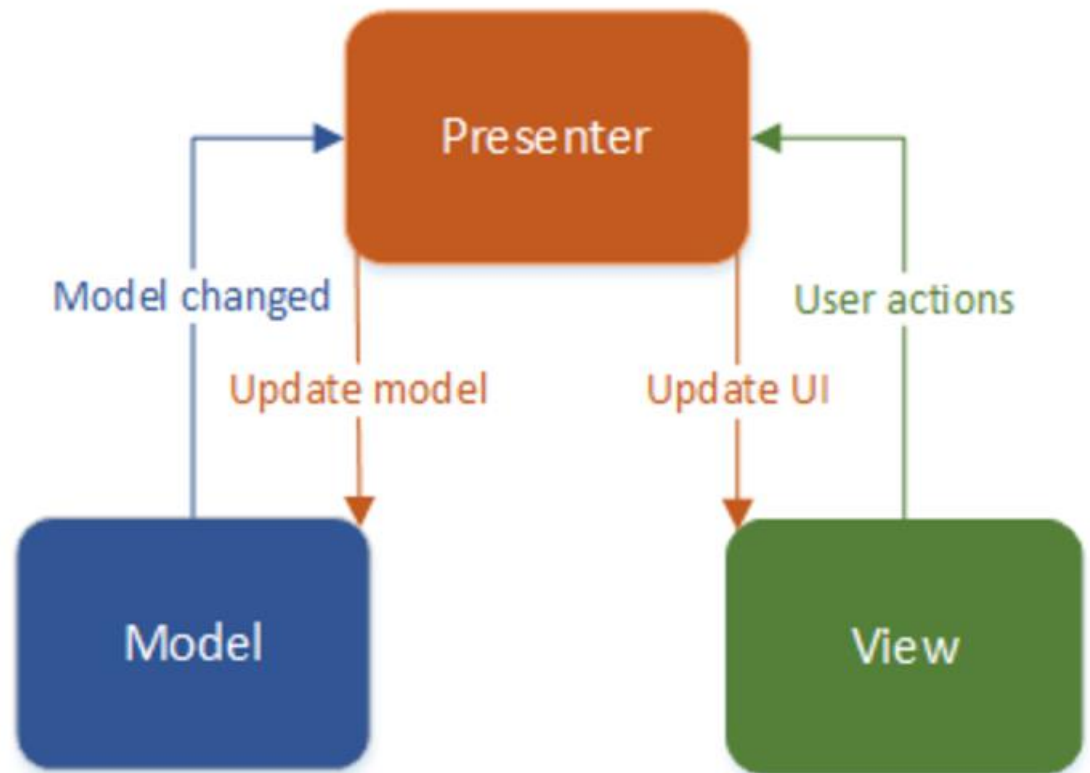
In a face recognition system, the Model would handle storing and processing facial data, the View would display the user's face and attendance status, and the Controller would manage interactions such as registering a new face or logging attendance.

MVP Pattern:

The Model-View-Presenter (MVP) pattern is a derivative of the MVC pattern, primarily used for organizing the user interface. It separates the application logic from the UI, making it easier to test and maintain:

- **Model:** Governs business behaviors and state management. It handles the data part of the application.
- **View:** Renders UI components and communicates with the presenter via an interface. It is responsible for displaying data to the user.
- **Presenter:** Acts as a middleman between the model and the view, handling the business logic. It retrieves data from the model and applies logic to decide what to display in the view.

MVP



Example:

In the face recognition system, the Model would manage the facial recognition data, the View would display the recognition results, and the Presenter would process the data and update the view with the attendance information.

Planning and Requirements:

The project will initially be based on the MVP architecture due to its focus on UI design and presentation layer responsibilities.

Key Features:

- **Face Detection:** The system can detect any trained face.

- **Attendance Logging:** Basic structure for logging attendance (future work).
- **User Interface:** Design a user-friendly UI for ease of use.

Development Tools, Methods & Environment Explanation:

1. Development Tools:

Programming Language: Python

Framework: Django for backend development and application structure.

Face Recognition Library: OpenCV and face recognition libraries for face detection and recognition.

Database: PostgreSQL for storing user data and face recognition data.

Version Control: GitHub for source code management.

IDE: VSCode with extensions like Python, live-server, auto-indentation, etc.

Design Tools: Figma for UI/UX design.

2. Methods:

Development Method: Agile methodology for iterative and incremental development. This involves regular updates, feedback cycles, and continuous improvement of the project.

Project Management: Jira for tracking project progress and collaboration. Tasks will be divided into sprints, allowing for clear milestones and goals.

3. Environment Explanation: The development environment will consist of:

Local Development: Setting up a local server using Django for development and testing. VSCode will be the primary IDE, equipped with necessary extensions for efficient coding.

Version Control: Using GitHub for version control ensures that changes are tracked, and multiple team members can collaborate effectively.

Testing Environment: Separate testing environment to run unit and integration tests, ensuring the application functions correctly before deployment.

Future Features and Development of the Project: Future enhancements will be based on user feedback and evolving requirements.

Upcoming Features:

Attendance Count: Implement functionality to count and log attendance based on face recognition.

Push Notifications: Notify users of their attendance status via push notifications.

Profile Management: Users can create and manage their profiles.

Login System: Secure login with email and password.

Forgot Password: Password recovery option with email verification.

Email Verification: Verify user emails with a 6-digit PIN.

Edit Profile: Users can update their profile information and profile image.

Calendar Integration: Users can view attendance history on a calendar.

Device Compatibility: Ensure the system works across various devices.

Data Analytics: Provide analytics and reports on attendance data.

Overview of Fully Developed Project and Conclusion:

The Online Face Recognition Attendance System will start with the MVP architecture for initial development, focusing on the user interface and face detection. Over time, it will incorporate MVC architecture patterns to handle more complex business logic and state management.

Using Python and Django will provide a robust framework for developing a scalable and maintainable application. The initial launch will focus on the web platform, with future plans to extend functionality and support additional features as described. Despite initial limitations, the project will be fully implemented and maintained for optimal performance and user satisfaction.

This project aims to provide a robust, efficient, and user-friendly face recognition attendance system, streamlining attendance management through advanced technology and design principles. The combination of MVP and MVC patterns ensures a scalable,

maintainable, and testable architecture, enhancing the system's overall reliability and usability.