GETTICK

FINAL REPORT

PROJECT COVER

TOPIC

Gettick, new event management system is a comprehensive platform designed to streamline the planning, organization, and ticketing of events. This system will provide event organizers and attendees with a seamless experience, offering a wide range of features to enhance event management.

ARCHITECTURE

We developed a comprehensive event planning and ticketing system using the Model-View-Controller (MVC) architecture. The MVC approach was chosen for its efficient separation of concerns, facilitating streamlined development and maintenance. Our system features include user profile management, event searching, ticket purchasing with an interactive seating chart, and more.

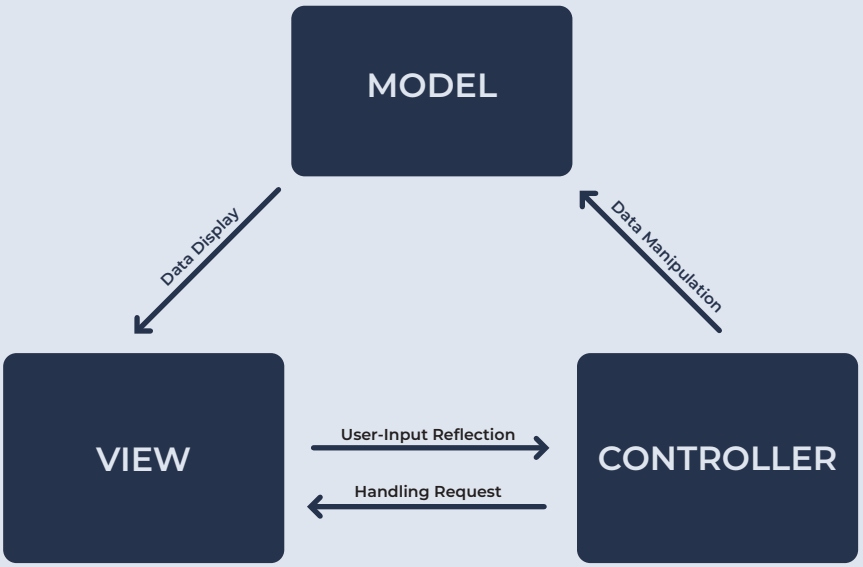
IMPLEMENTATION

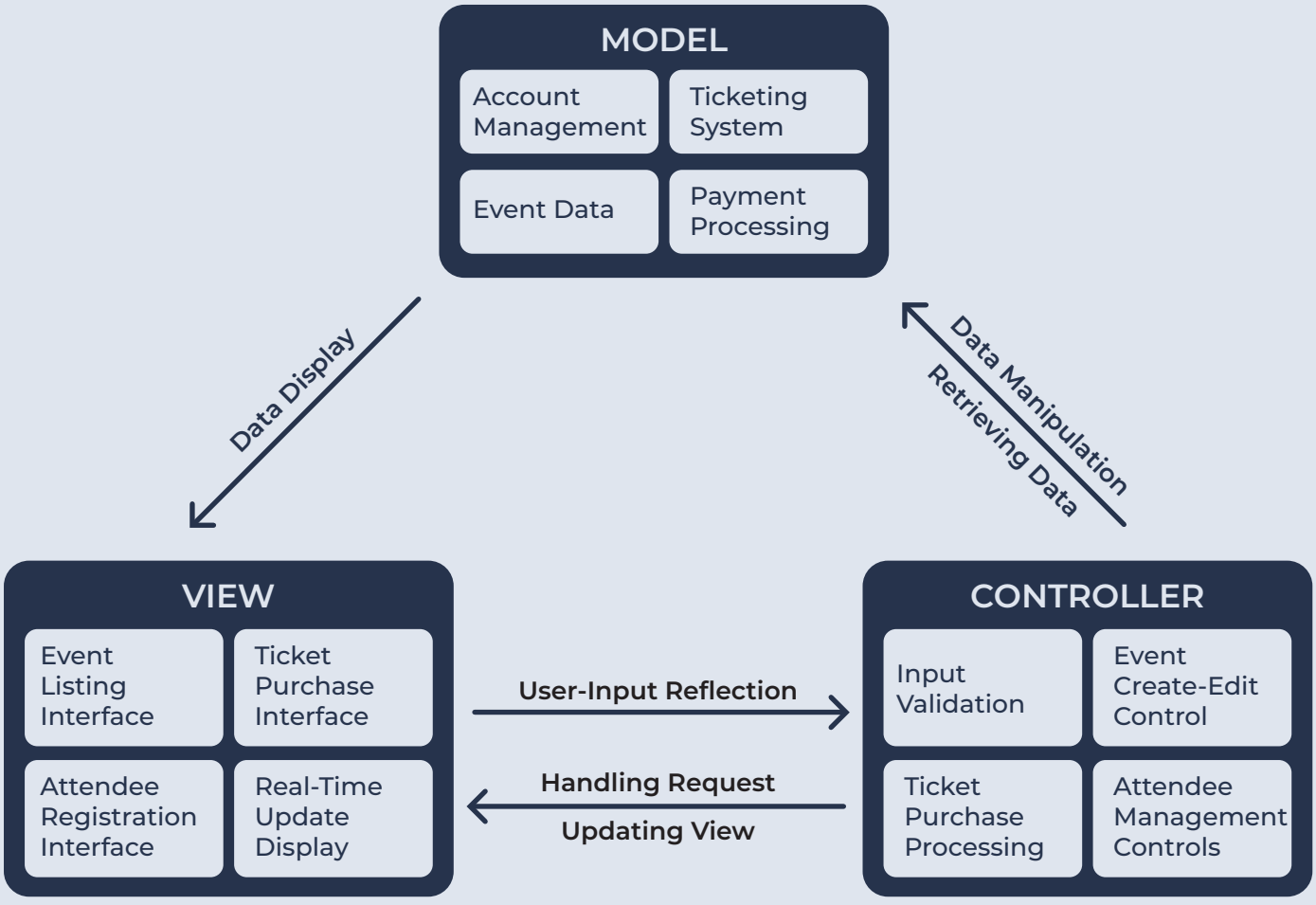
We implemented the project using Python and Flask, capitalizing on Python's simplicity and Flask's lightweight, flexible nature. This combination allowed us to create a user friendly and scalable application, perfectly suited to the dynamic needs of event management and ticketing.

GROUP 4

05190000081 - Safiye Defne Kargı
05200000034 - Muhammet Şancı
05190000104 - Hazal Karataş
05200000761 - Berat Duran

PROJECT ARCHITECTURE





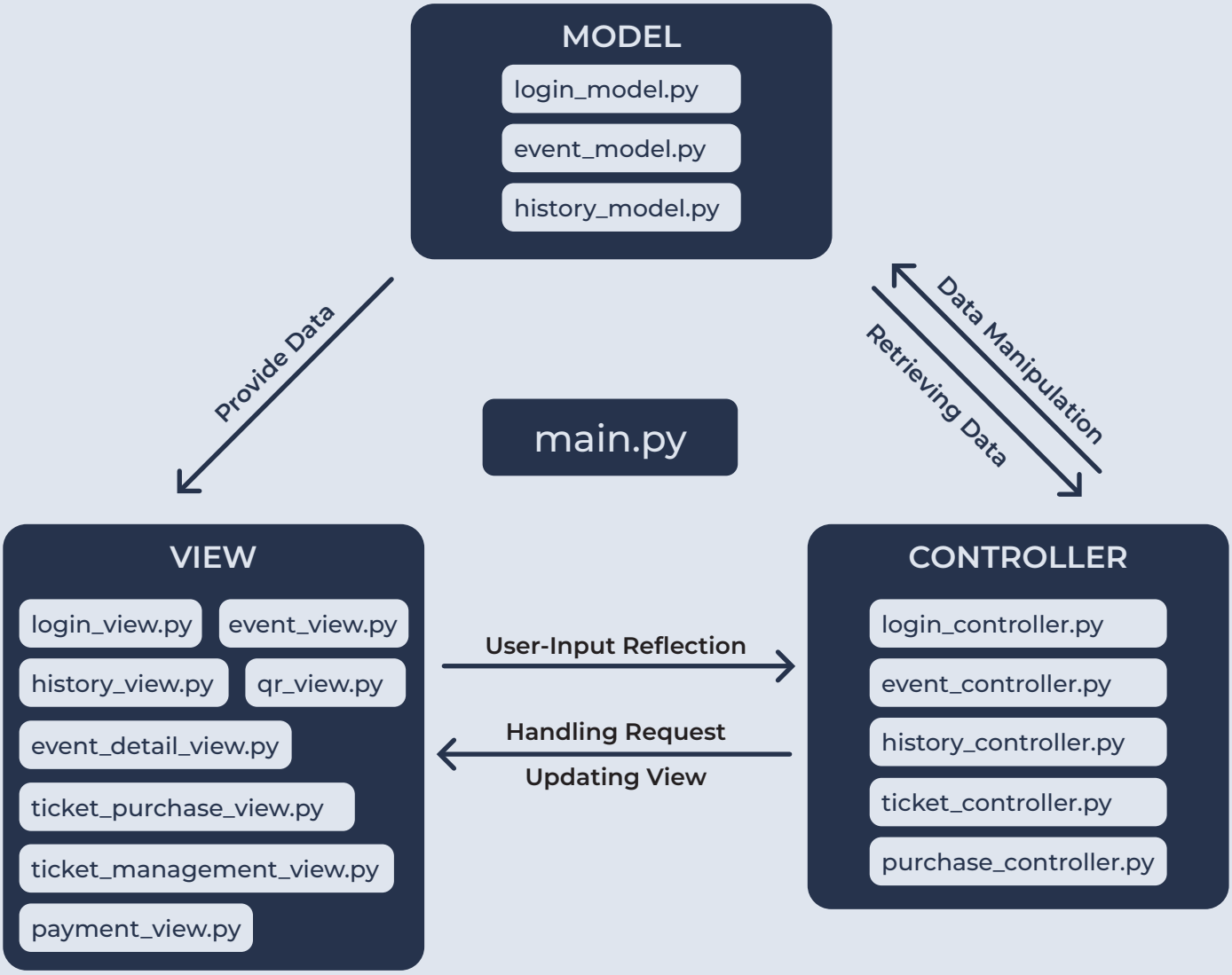
PROJECT ARCHITECTURE

In our project, the file organization directly reflects the **Model-View-Controller (MVC)** architecture:

- Models:
These files contain the data-related logic, handling interactions with the database and processing of business rules.

- Views:
These files are responsible for the presentation layer, defining how data is displayed to the user.

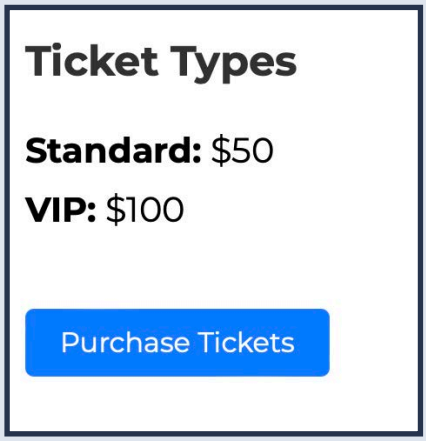
- Controllers:
These act as intermediaries, processing input from the views, using models to interact with the data, and updating the views accordingly.



UI CONSIDERATIONS

UI DESIGN PRINCIPLES

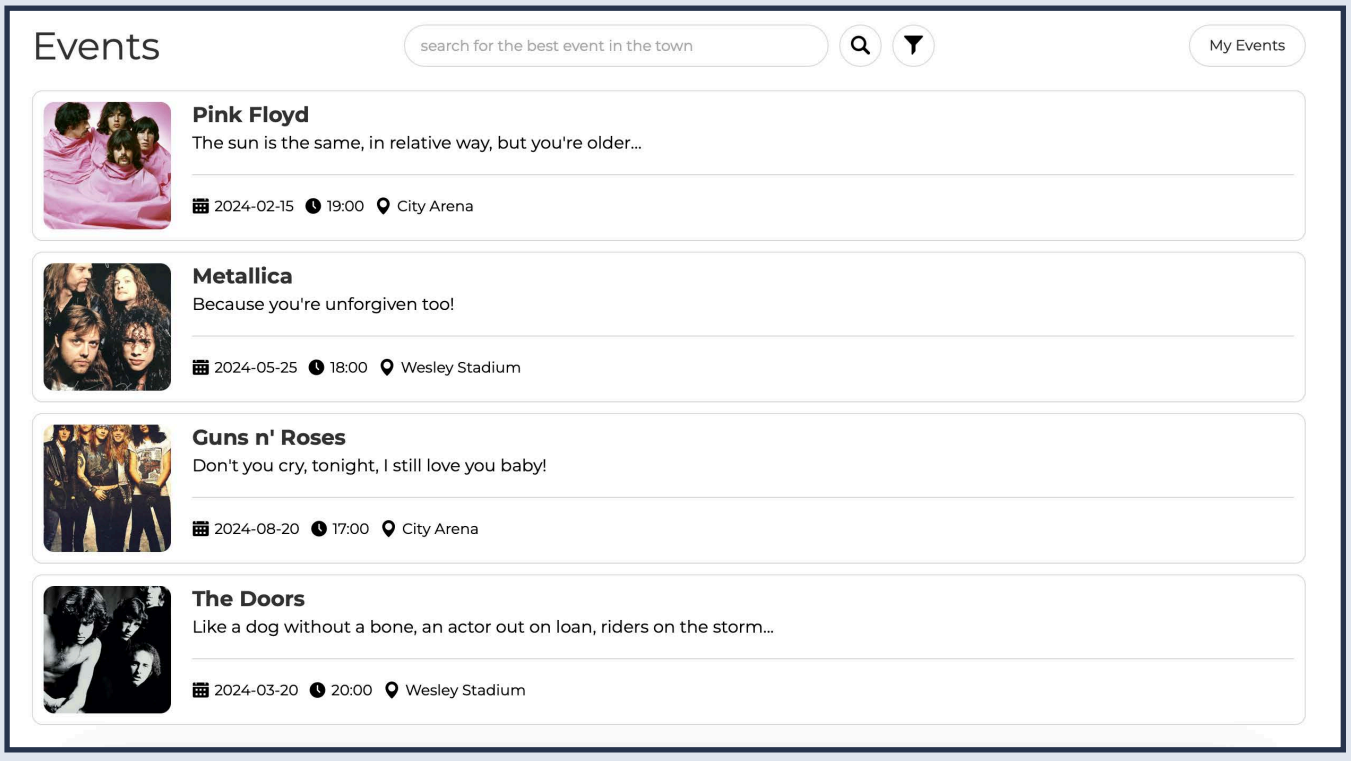
POSITIVE, NEGATIVE, NEUTRAL BUTTONS



Not only one, but one of the UI design principles we use is button color matches. Positive buttons represent normal actions (such as purchase, buy, login etc.), negative buttons represent dangerous actions (such as delete, logout, cancel etc.), while neutral buttons represents ordinary actions.

UI DESIGN PRINCIPLES

MINIMALISM



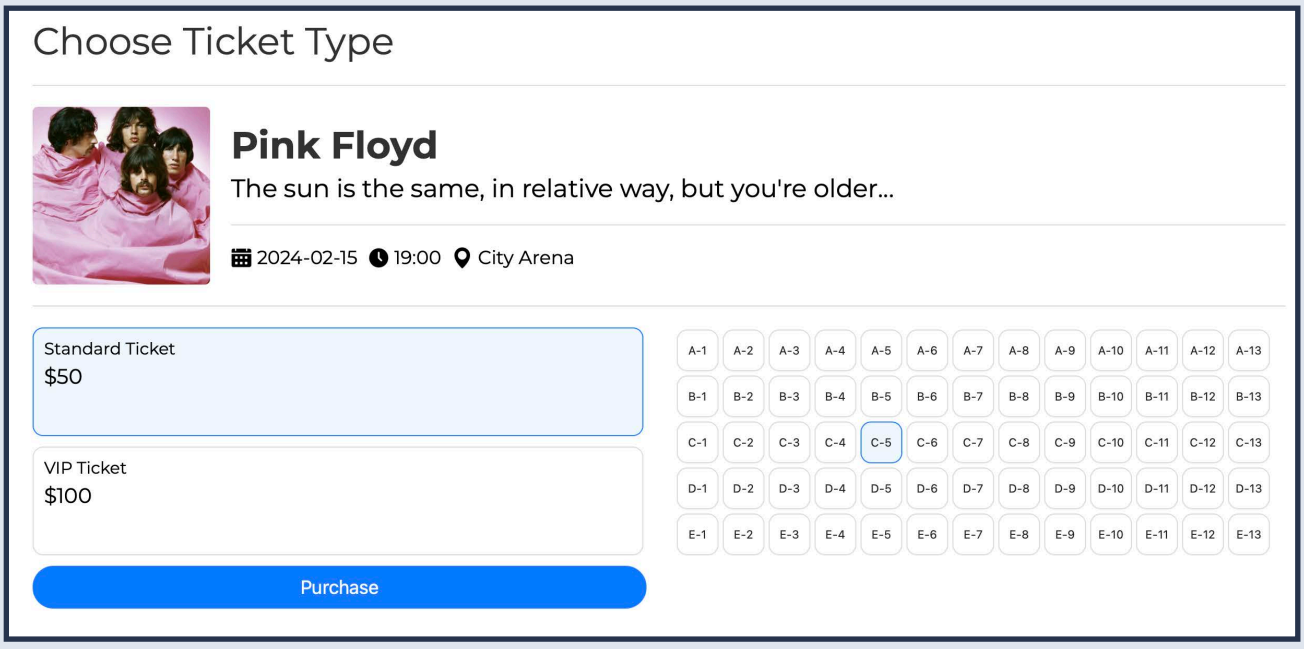
UI CONSIDERATIONS

INFORMATION REPRESENTATION

GROUPING INFORMATION

INTERACTION STYLES

SELECTION HIGHLIGTING



TEST CASES

Test Scenario 1: Ticket Purchase
Objective: Confirm that a user can purchase a ticket.

1. Log in to the system.
2. Search for an event and select it.
3. Choose a seat from the interactive seating chart.
4. Proceed to purchase the ticket.
5. Expected Result: The user should be able to select a seat, complete the purchase, and receive a confirmation with a QR code for the ticket.

Test Scenario 2: Preview QR Code for Ticket
Objective: Ensure that users can preview the QR code for their purchased ticket.

1. Log in to the system and go to the My Events page.
2. Click on the "Preview QR" button associated with the ticket.
3. Expected Result: The system should display the QR code for the selected ticket, which the user can use for check-ins.

Test Scenario 3: Selecting Seat and Ticket Type
Objective: Verify that users can select a seat and ticket type on the "Purchase" page before finalizing their ticket purchase.

1. Log in to the system and go to the event details page for a specific event.
2. Choose a ticket type from the available options.
3. Interact with the interactive seating chart to select a specific seat. Confirm that the seat is highlighted or marked as selected after clicking.
4. Click on the "Purchase" button after selecting the seat and ticket type.
Expected Result: The system should allow the user to select a ticket type and a specific seat from the seating chart. After selection, when the user clicks "Purchase," they should be directed to the payment or confirmation page with the selected seat and ticket type details.