UMI's Model View Controller

The website UMI uses the MVC (Model-View-Controller) based architectural pattern where the software is divided into three interconnected parts, each responsible for a different task namely, it's model, view and control. This enables modular coding, making it easier to deploy, maintain and test the code.

Model

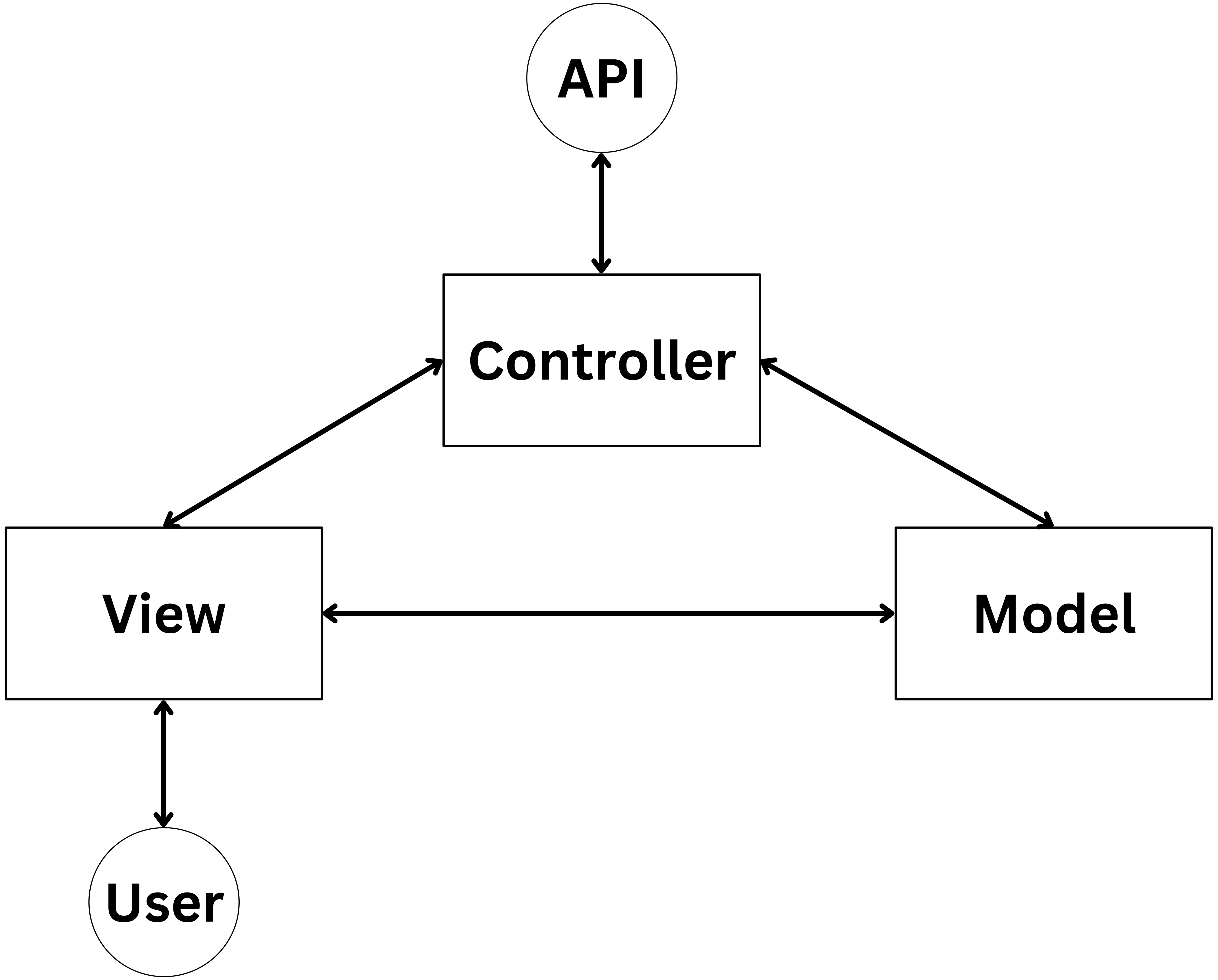
The model in UMI is associated with handling all the data related aspects of the code, which include data storage, retrieval and data updates. The model would store user data ranging from task information, schedule, music preference and such in a structured format in collections or tables. Temporary data such as Pomodoro session information, theme preferences and so on will be stored locally to ensure quick access.

View

The View component in UMI is the interface a user can interact with. It is responsible for rendering data from the model and displaying to the users. The interface includes the task planner, calendar, Pomodoro session information, music information and controls and such all while ensuring ease of navigation, enhancing user experience and catering to theme preferences. HTML, CSS, Javascript and React.JS will be used to achieve our dynamically updated user interface.

Controller

The controller component serves as a middleman between the Model and View components which ensure proper updating of the Model and View components according to the user inputs. The controller interprets the user actions and sends appropriate commands to the Model and subsequently the View to ensure data consistency across our software. It is also responsible for interaction with the external APIs that is, Notion, Spotify, Google Calendar API and Todoist. The controller is also responsible for logic validation to ensure valid data and manage exception and errors from external APIs to ensure seamless integration between UMI and the external APIs.


API
Controller
View
Model
User