Architecture Design Description

for UMaine Connect by Maineframe

UMaine Connect will be developed using the Model-View-Controller (MVC) architectural design pattern. This design has been chosen because of its straightforward and intuitive application to web apps like UMaine Connect. Figure 1, below, is an architecture design diagram illustrating the MVC structure of UMaine Connect.

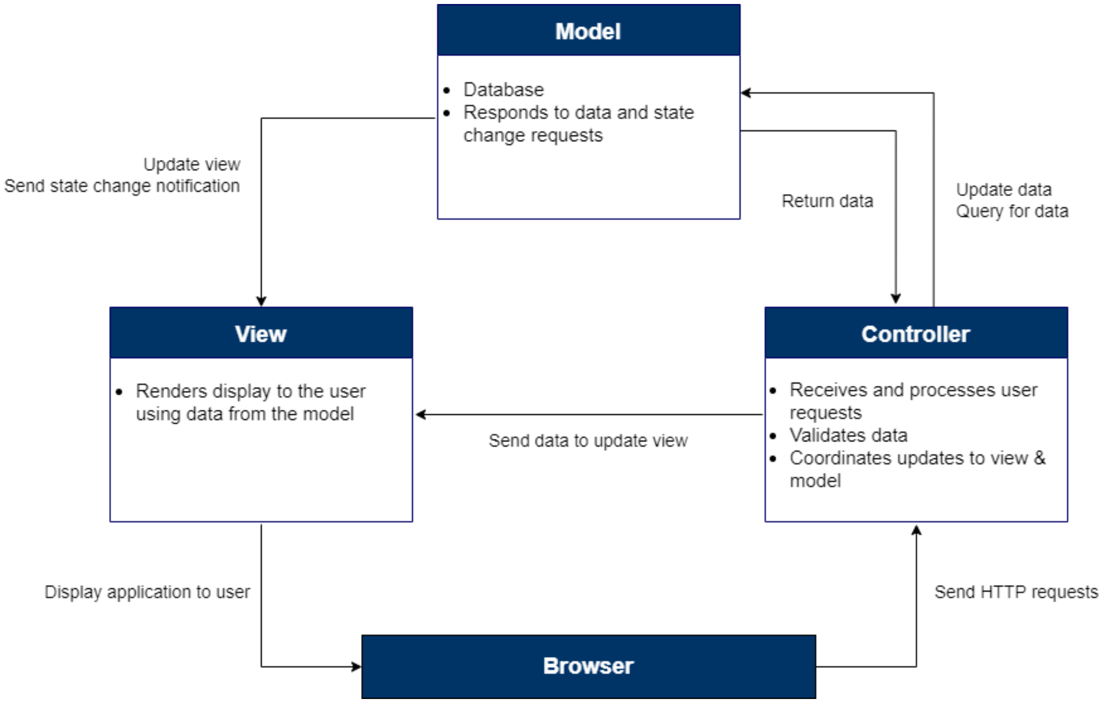
Model
Database
Responds to data and state change requests
Update view
Send state change notification
Return data
Update data
Query for data
View
Renders display to the user using data from the model
Controller
Receives and processes user requests
Validates data
Coordinates updates to view & model
Send data to update view
Display application to user
Send HTTP requests
Browser

Figure 1. *MVC architecture design diagram for UMaine Connect*

Component Descriptions

Model

The model component of MVC manages the behavior and data of the application. It responds to information and state change requests from the controller component. It can also update and send state change notifications to the view component. UMaine Connect's model component is its HTML framework and databases.

View

The view component of MVC manages the way the application displays information. The controller queries the model for information and passes it to the view, which renders it to the user in a way they can easily understand and interact with. UMaine Connect's view component is its web frontend and user interface, implemented with CSS.

Controller

The controller component of MVC manages user interaction with the application and coordinates updates to the view and model components. The controller receives and processes user input, then updates the model and selects a new view as necessary. UMaine Connect's controller component is implemented with JavaScript.

Relationship and Interactions

The Model-View-Controller architecture separates an application into three main logical components: the model, the view and the controller. These three components work together and independently to handle specific development aspects of an application. First the controller receives a request from the user, the controller then interacts with the model to receive the data that was requested from the user. After receiving the data from the model, the controller interacts with the view to render the requested data. The controller repeats this process and updates the view whenever data changes.

Justification

There are many architectural designs but the one that fits UMaine Connect the best is Model-View-Controller. MVC is commonly used for web based applications and allows multiple ways to view and interact with the data. UMaine Connect as a web app needs the three basic principles provided by the MVC architecture. The model will manage application data, the view will provide a display to the user for interaction, and the controller will interpret inputs from the user. These three principles are an absolute necessity for UMaine Connect.