Architecture

[KUMO]

Version [2.0]

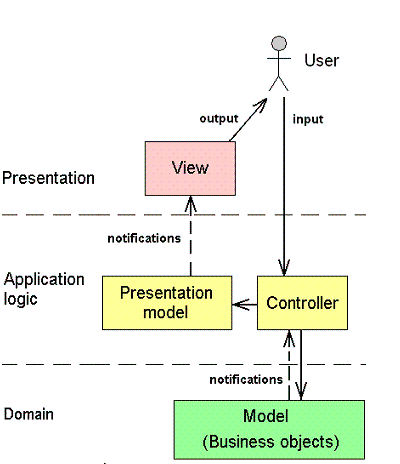
Prepared by

[Alice Huynh]

1 Application (MVC) Architecture Diagram

1.1 Company website

All customers will use the same company website. Company website is the MVC website. It’s like the template. The page that end-user see on their screen is the template + data from database.

****

**1. Model (Business objects)**

Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in a SQL Server database.

**2. Presentation model**

This model pulls presentation behavior from a view.

**3. Controller**

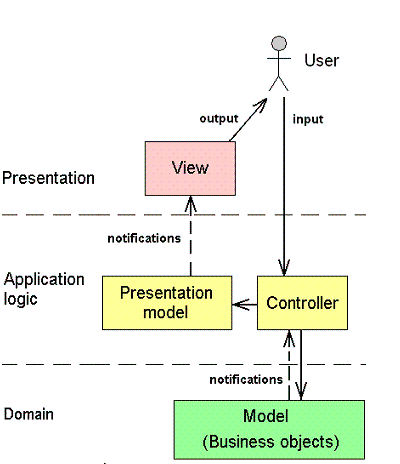
Controller is the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database.

**4. View**

View is the components that display the application's user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Product object.

1.2 Central Admin website

Central Admin website is a MVC website. With this site, Administrator can sign in to add new company, register components for that company.

****

**1. Model (Business objects)**

Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in a SQL Server database.

**2. Presentation model**

This model pulls presentation behavior from a view.

**3. Controller**

Controller is the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database.

**4. View**

View is the components that display the application's user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Product object.