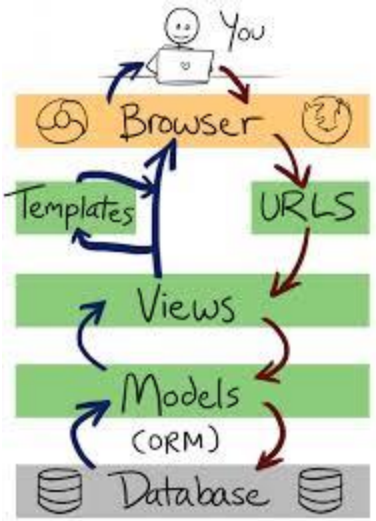
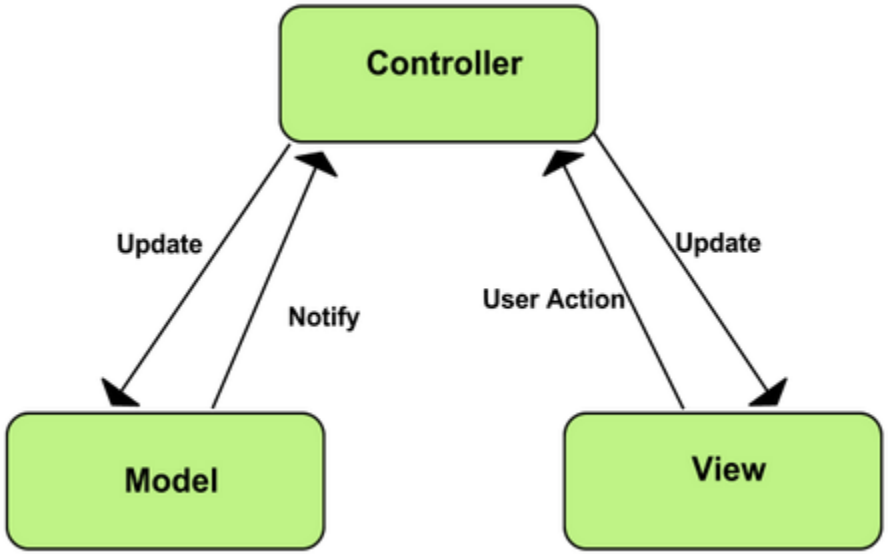
CIS 117 Python Internet Programming



College of San Mateo

Model View Controller Architecture



MVC

- MVC stands for Model-View-Controller
- The Model is the actual internal representation
- The View (or a View) is a way of looking at or displaying the model
- The Controller provides for user input and modification
- These three components are usually implemented as separate classes

The Model

- Most programs are supposed to do work, not just be “another pretty face”
  - but there are some exceptions
  - useful programs existed long before GUIs
- The Model is the part that does the work--it *models* the actual problem being solved
- **The Model should be independent of both the Controller and the View**
  - but it provides services (methods) for them to use
- Independence gives flexibility, robustness

The Controller

- The Controller decides what the model is to do
- Often, the user is put in control by means of a GUI
  - in this case, the GUI and the Controller are often the same
- The Controller and the Model can almost always be separated (what to do versus how to do it)
- The design of the Controller depends on the Model
- The Model should *not* depend on the Controller

The View

- Typically, the user has to be able to see, or *view*, what the program is doing
- The View shows what the Model is doing
  - The View is a passive observer; it should not affect the model
- The Model should be independent of the View, but (but it can provide access methods)
- The View should ***not*** display what the Controller ***thinks*** is happening

Code Independence

- As always, you want code independence
- The Model should not be contaminated with control code or display code
- The View should represent the Model as it really is, not some remembered status
- The Controller should *talk to* the Model and View, not *manipulate* them
  - The Controller can set variables that the Model and View can read

CSM