MVC , ORM , Devise

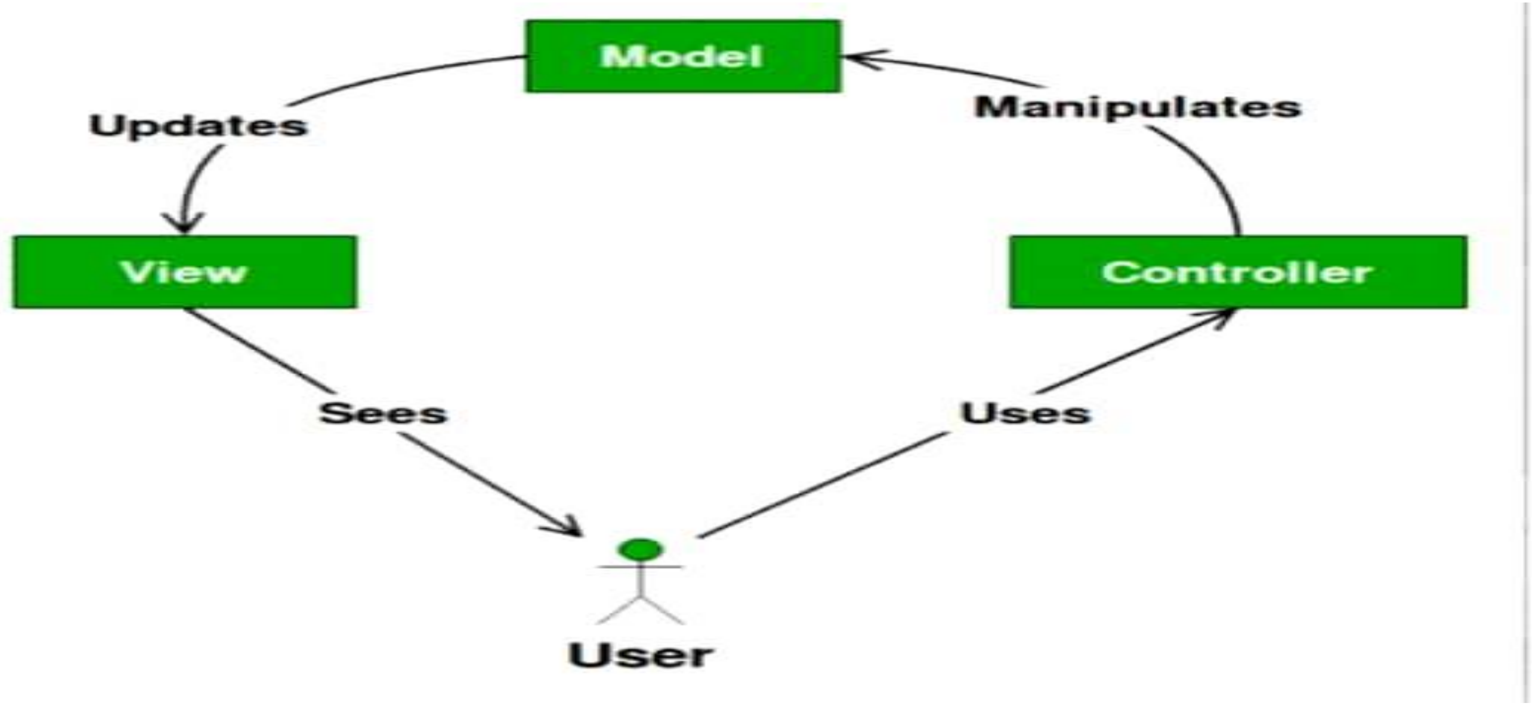
Ruby Topics

MVC (Model View Controller)

The Model View Controller (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

- The Model contains only the pure application data, it contains no logic describing how to present the data to a user
- The View presents the model’s data to the user. The view knows how to access the model’s data, but it does not know what this data means or what the user can do to manipulate it.
- The Controller exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model.

UML Diagram for MVC Design Pattern



ORM (Object Relational Mapping)

Object-Relational Mapping (ORM) is a technique that lets you query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a library that implements the Object-Relational Mapping technique, hence the phrase "an ORM".

An ORM library is a completely ordinary library written in your language of choice that encapsulates the code needed to manipulate the data, so you don't use SQL anymore; you interact directly with an object in the same language you're using.

Using ORM saves a lot of time

- DRY: You write your data model in only one place, and it's easier to update, maintain, and reuse the code.
- A lot of stuff is done automatically, from database handling to I18N.
- It forces you to write MVC code, which, in the end, makes your code a little cleaner.
- You don't have to write poorly-formed SQL (most Web programmers really suck at it, because SQL is treated like a "sub" language, when in reality it's a very powerful and complex one).

DEVISE (Ruby Gem)

The Devise Gem is users for User Authentication, It creates sign-up and sign-in forms, it also can be used to create user accounts for privacy. ... Next, we need to create our Devise model, you can name it anything, (e.g. User, Admin, Member or Staff) depending on your purpose.

So as with all other gems,we need to install in the usual way by adding to your Gemfile, gem 'Devise' Then run bundle install in your terminal. After you've done that, you'll need to run the generator. rails generate devise:install.