SE31520 Assignment 1 Report

Introduction

This application follows the MVC (Model View Controller) design pattern. This pattern is designed to make it easy to abstract different areas of an application away from each other based on how the users interact with them.

The core idea is to take a class and split its functionality into three areas. The model is what manages the data, the controller is what manages the operations and functions of the class and the view is what is presented to the user, that allows them to interact with it.

In a typical use case the user gets presented with a view of a class, say "Books", and is then able to perform CRUD (Create Retrieve Update Delete) style operations from this view. Once the user has given the command, lets say delete, this is then passed on to the controller for the class. The controller will now perform any actions that need to be done with relation to the operation, then call on the model to perform the update to the database, in this case deleting a Book. Once this operation has been completed the controller will then respond to the user by rendering an updated view for the user.

Application

This application has three pages, Users, Posts, and a Home/Index page. The first two pages follow the MVC pattern, as they are both resources that are stored in a database and provide CRUD operations. They are also linked as Posts have a foreign key to the Username of the Users table.

However the Home/Index page doesn't follow the MVC pattern as it has no model. This means that it is following the Model 2 variant which doesn't require any interaction with the model. This is done because the home page is simply displaying static content and has no need for a database object to store and update data with.

The application uses tabbed navigation and pagination as we learned in the workshops editing the CSA application.

Models

The two models for this are the Users and Posts, here there isn't much defined only the validation for the what properties are required for an object to be created, and how the objects link to each other. The rest of the traditional CRUD options are handled by Rails magic.

Controllers

The controllers for the Users and Posts were auto-magically created by rails's scaffolding command. This created the handlers for all of the CRUD operations and the rendering of all the views. I simply had to modify it to allow for the pagination and to define what parameters were allowed to be posted to it.

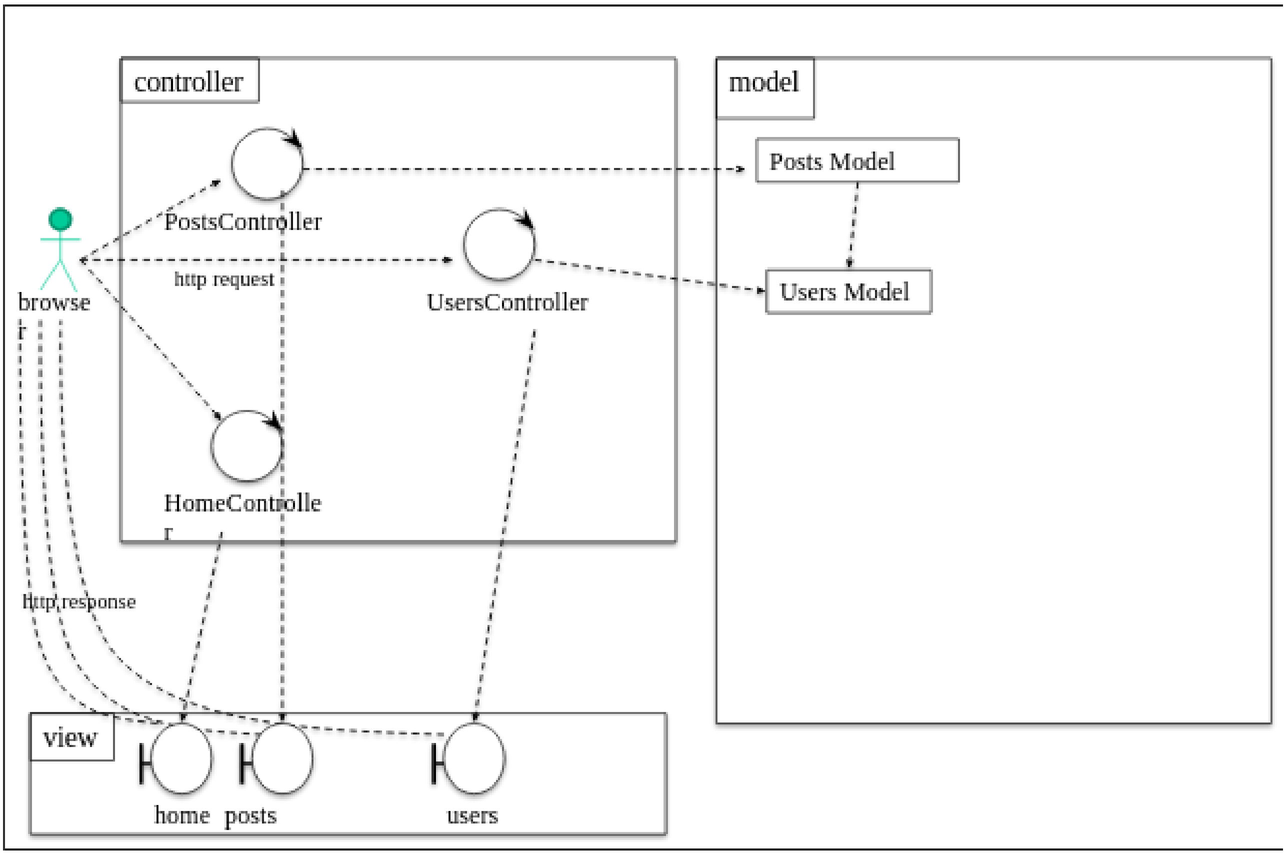
Views

Again the views were created auto-magically by rails when the scaffolding command was ran. I simply had to modify these to reflect the new changes such as pagination and tab navigation, that would be displayed to the user.

Routing and navigation

To make the homepage act as the default I had to edit the routes.rb file to set the homepage to the index, and then for the tab navigation, I just had to update the navigation.rb file to use the new pages.

MVC Diagram


controller
model
Posts Model
PostsController
browser
http request
UsersController
Users Model
HomeController
http response
view
home
posts
users