CVWO Mid-Assignment Write Up

Basic Use Case:

"I want to see what tasks I have"

VIEW

Shows existing tasks in database

View gets the controller to run the method to obtain the tasks

Controller gets the data from model and passes it to view to show the user

CONTROLLER

Controller requests model to access database and retrieve results

Model processes the data retrieved in a format understood by the controller

MODEL

Model sends the query to the database

Database returns all data related to query as is
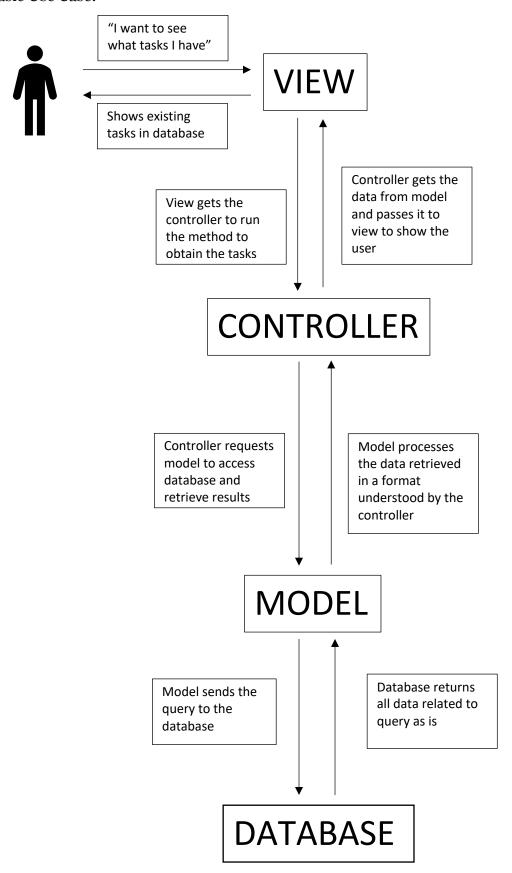
DATABASE

Execution Plan:

Phase 1 - The Basic Implementation
   All todos should have:
- Todo Item: a simple "title"-like statement/string to summarize the task
- Details : text containing more details pertaining to the task
- Done: a boolean to check if completed or not
- Timestamp: Date and time created and modified.

   Main Page - Shows a brief list of all todos in database
      Todo Info Page - Shows all the details/information of the task selected from main page. Option to delete the task and check box when done from here (for now)
      New Todo Page - Create a new todo

Phase 2 - The Better Implementation
      All todos should now also have:
- Deadline: date and time (should also have days remaining on main and info page)
- Tagging
- Priority
- Advanced details? Images, links etc
- Confirmation or option to undo upon delete

      Main Page - Should now have more advanced items like per-task options in-line. Sort by tags, priority or deadline
      Begin to set a CSS theme for the application (layout, typography, colours)

Phase 3 - User Authentication
- Very big goal

Problems Faced:
- Currently, the largest issue I've been facing (although this might sound like an excuse) is the issue of time. I'm currently working a part-time job most days during the break. I've only managed to get research, reading and planning in my free time and only got to actually get to work on the Rails project in 2 days so far. However, I have a very good idea of what I want and it's just a matter of spending my next few free days to get the full functionality of the application working before working on the front-end and making it look as presentable and professional as possible.

- My next issue is deciding between RubyGems and hardcoding certain functionalities. One example is deciding if the Simple-Form Gem is effective in use since I would to learn more functionalities of the Gem. Another example is the deciding on the use of the Acts-As-Taggable-On Gem to fulfil task 2; to create a tagging system for the tasks. I currently have an implementation planned out already and have done the necessary research to not use it, if I can get it to work as expected. I'm thinking if doing it myself and spending time is a viable option or if RubyGems is enough to fulfil the task at hand. Would I be cutting corners? Or would it be a better idea to try both ideas out by making two branches in my Git.

- Bootstrap. I still haven't had enough research done with Bootstrap. Plenty of research left to do. Or explore other frameworks.

- Deciding on the possibility of a Single Page Application (SPA). Came across a very interesting article on using Turbolinks to improve performance (maybe thinking a little too far ahead at this point, since the application is not very demanding, extremely low priority)

- Not really facing any roadblocks that are hindering my progress other than the above decisions to make.