Senior Design

ENG EC 463

Memo

To: Professor Pisano

From: Sharith Godamanna, Camden Kronhaus, Quinn Meurer, Alexander Trinh

Team: 11

Date: 11/22/20

Subject: 1st Prototype Test Report

1.0 Equipment and Setup

The Stod application architecture is split into two parts: the backend and frontend, which communicate through HTTP requests. This allows for modularity and can easily be migrated to a microservices architecture. To start with a backend template, we create a Python virtual environment and install Django within it. To create a new Django application we run `django-admin startproject stod-backend` which creates an empty django application. For the frontend, we create a new React app running `yarn create stod-frontend --typescript` specifying a typescript template. We decided to use typescript to be able to use type safe variables which makes debugging much simpler. To handle global React

state, we are using Redux which manages a global store where we can store and access global variables. To connect our frontend to the backend, we created asynchronous redux actions and used the axios library to make HTTP requests to our server.

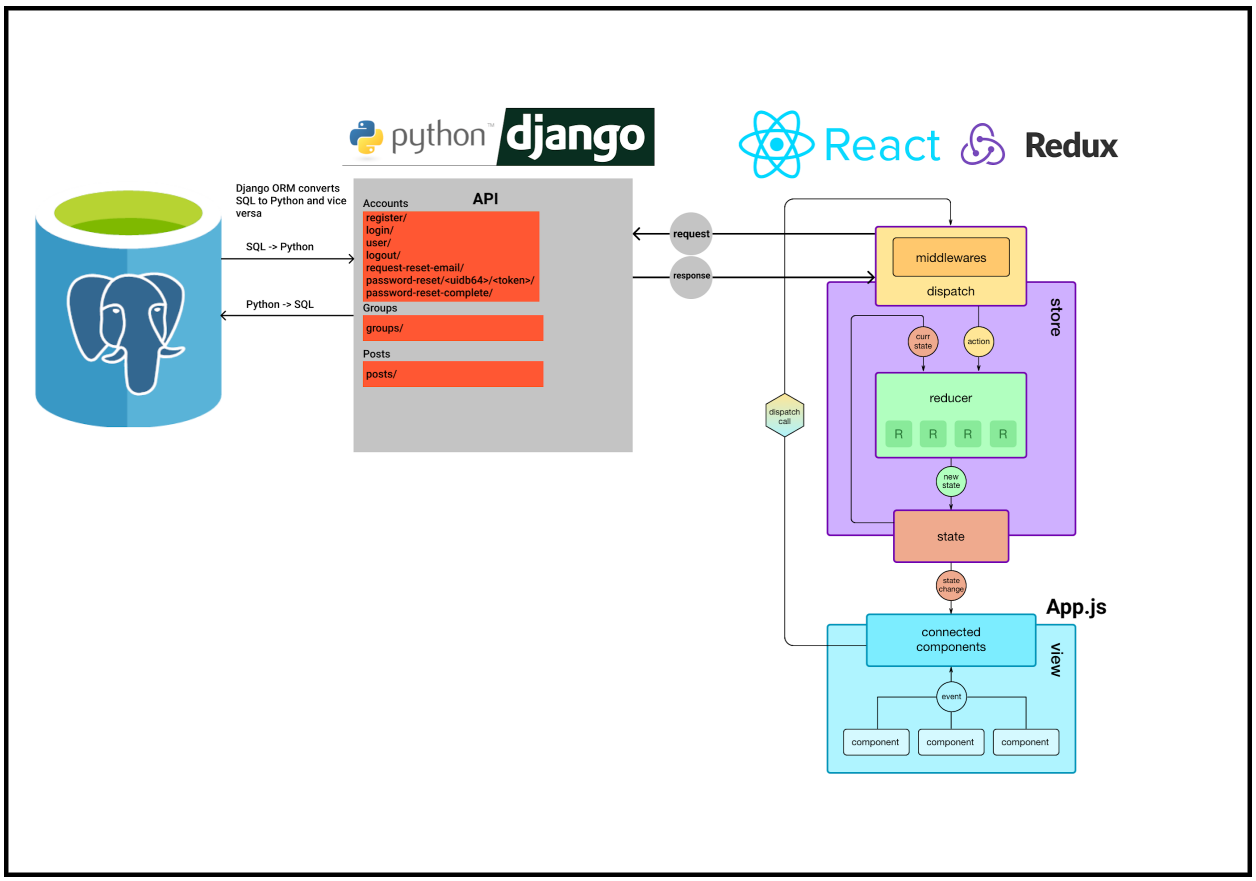

Figure 1: Stod application architecture

2.0 Test Report

2.1 Measurements Taken

Our measurements are scored on an all or nothing basis for each of the tests we did. If the test passed, it scored a one; if the test wasn't completely successful, it received a 0. The final score was tallied out of the 22 total tests. Each of our tests examines the key features that were implemented for the prototype to ensure they all work properly.

Test Parameters	**Score /1**
Register a new user	1
Login with newly created user account	1
Attempt to login with a user that does not exist	1
Register: Field was left empty	1
Register: Username and/or email already exists	1
Register: Email is invalid	1
Register: Password too common	1
Register: Password not long enough	1
Register: After successfully registering, page does not get redirected to /home page.	1

Login: Field was left empty	1
Login: Username and password combination invalid	1
Login: After successfully logging in, page does not get redirected to /home page	1
Posts: Navigate to the /posts route	1
Posts: View all posts currently available on the back end server and confirm all posts in the back end database are visible on the posts page	1
Posts: Create a new post using Django REST framework.	1
Posts: Refresh posts page and view all posts	1
Comments: Create multiple posts available for commenting	1
Comments: Comment on the newly created posts	1
Comments: View all of the comments	1
Groups: View the list of groups on the frontend. The name and description for each group should be visible.	1
Groups: Delete a group on the frontend. The frontend should be updated with the group removed.	1

Groups: Add a new group by specifying a name and a description. The frontend should be updated with the new group visible.	1
Final Score:	**22/22**

2.2 Conclusions

The main focus of our first sprint (which we tested) was getting a lot of the basic functionality working for our project. We succeeded in building the frontend and backend functionalities for user authentication and authorization, groups, and posts, and we still have a bit more work to do on comments due to the more complicated nature of this task. For our next sprint we are going to focus on effectively tying these individual components together to make a fully functional product. This will include keeping track of certain groups that a user is subscribed to, displaying comments under the specific post they pertain to, and other tasks of this nature. Professor Osama gave us feedback to try to get a working MVP completed before the start of the second semester so that we can get user feedback and iterate on it. The tasks in our second sprint work towards the goal of finishing the backend functionality for our MVP. Moving forward with our third sprint, we plan to complete our MVP by building a user-friendly frontend interface. We think that if we stay on track with these sprint goals we will be able to deploy our application so that users in the class will be able to test it by the beginning of next semester, giving us valuable user feedback to further improve.