Senior Design

ENG EC 464

Memo

To: Professor Pisano

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

Team: 11

Date: 4/09/21

Subject: Final Test Report

1.0 Equipment and Setup

The Stod application is built using a microservice architecture currently in three parts: a chat server, a general backend server, and a frontend which communicates with both backend components using HTTPS requests. Both the chat server and the general backend server run on a public Ubuntu server tied to https://stodbackend.app and communicate locally with Nginx to allow for secure communication from requests coming from users' browsers interacting with the front end. The chat server runs using MongoDB and Node.js while the general backend server runs using Django and PostgreSQL. Both are started by running their respective docker builds to allow for a consistent startup process. The front

end is statically generated and hosted automatically with pushes to a production branch in Github using Netlify. The front end can be reached by going to https://www.stod.app. In the front end, we are using typescript in order to use type safe variables and for easier debugging, as well as Redux to manage global state. To connect our frontend to the backend, we created asynchronous redux actions and used the axios library to make HTTPS requests to our backend services.

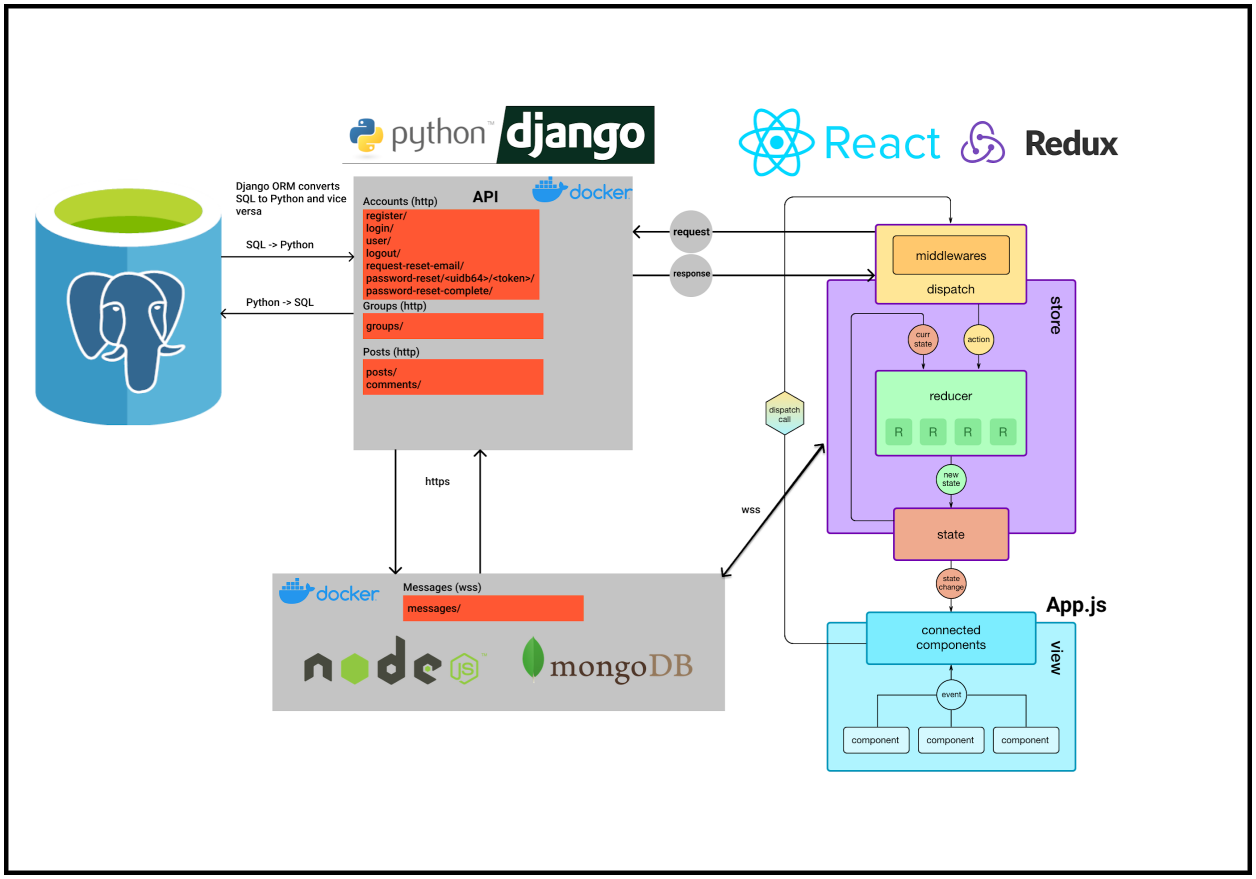

Figure 1: Stod application architecture

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, i.e. the measurable criteria was successful, it scored a one; if the test wasn't completely successful, it received a 0. The final score was tallied out of the 30 total tests. Each of our tests examines the key features that were implemented for the prototype to ensure they all work properly. The test parameters are formatted such that it is: [test]->[measurable criteria].

Test Parameters	**Score /1**
Navigate to https://www.stod.app in a web browser -> Login page loads successfully with HTTPS	1
Register a new user -> A new user can be created and logged in as that user at any point	1
Login with newly created user account -> A new user can be created and logged in as that user at any point	1
Attempt to login with a user that does not exist -> A new user is unable to register if there are errors at login	1

Check login security features -> A new user is unable to register if there are errors at login	1
Navigate to home page and view all posts loaded on the home page -> Navigation to the home page results in all posts shown to the user and the groups they were created in	1
Navigate to a subscribed group to view all the posts within that group -> Navigation to a subscribed groups results in only posts for that group being displayed	1
Create a new post from within a subscribed group, specifying a title, body, and tags -> Creation of a post is reflected accurately in both the front and back end after submission	1
Edit a post created by the current user, and view updates -> Editing and Deleting a post is available to the user who created them, Edits to a post are reflected accurately in both the front and back end after submission	1
Delete post created by a user, and view updates -> Editing and Deleting a post is available to the user who created them	1
Filter the displayed posts by selecting specific tags -> Filtering mechanism	1

accurately filters so only posts with certain tags are shown	
Create a post that contains a bad word in the title -> Creating a post that contains a bad word should alert the user and not show up for current or other users on the app	1
Create a post that contains a bad word in the body -> Creating a post that contains a bad word should alert the user and not show up for current or other users on the app	1
View the two flagged posts as an admin -> A post with a bad word should be sent to an admin page for approval/denial	1
Approve one of the flagged posts and attempt to view it in the main group page -> A post that has been approved should be removed from the admin page and be visible to users within the group	1
Deny the other post and attempt to view it in the main group page -> A post that has been denied should be deleted from the database	1
Comment a single comment under each post within a group -> Able to create a comment under a previously created post	1

Choose a post and add multiple comments -> Comments should be updated and visible automatically without refresh	1
View each comment under their respective posts -> The comments should be connected to the id of the post, Only comments pertaining to a certain post should be loaded under that post, All previously created comments under a post should always load under that post	1
Reply to a comment -> A reply should be loaded under its respective post like a new comment and reference the original comment	**1**
Click on the second users profile, and send a friend request -> Friend requests are shown to a user	**1**
Accept friend request on User 1's profile -> Accepting friend request immediately adds them to the friends list on the sidebar	**1**
Create a third user, send a friend request, and reject the friend request -> Rejecting friend requests must not show on the sidebar.	**1**
User 1 sends user 2 a message -> Starting a conversation initiates one to one connection between two users,	**1**

Sending a message gets broadcasted immediately to recipient, User 1's message shows on the right side and user 2's messages show on the left side.	
User 2 sends user 1 a message -> Sending a message gets broadcasted immediately to recipient, User 1's message shows on the right side and user 2's messages show on the left side.	**1**
Navigate to home page and click "Find Groups" to view all existing groups -> All available groups are displayed to the user after clicking "Find Groups"	**1**
Click "subscribe" on a group -> User is able to subscribe to new groups	**1**
Navigate back to the home page see list of groups in the sidebar -> Subscribed groups are displayed in sidebar	**1**
Click on a subscribed group in the sidebar and view all the posts to that group -> Upon clicking on a subscribed group in the sidebar, user can view posts from that group	**1**
Unsubscribe from a group -> Group is remove from list of subscribed groups upon user's unsubscription	**1**
Final Score	**30/30**

2.2 Conclusions

For our final project testing, we wanted all the main functionality available to test as well as have it all be on the public servers that would be accessible to everyone, including the Professors during the presentation. All the main features we wanted at the very least to have for our MVP were present and able to be tested including a filtering system with a tagging mechanism to let users tag certain posts with tags such as “sensitive topic” and allow other users to filter by those tags. We were also able to test a moderation/flagging system in which words with certain bad words would be automatically detected by our system and sent to a private (admin only) route to be reviewed and either accepted and posted or denied and removed. We also improved the functionality of messaging since our last prototype test, and were able to test all previous mechanisms of all other features on our new public back end server with our public domain www.stod.app.

We had a nearly flawless test with the only hiccup being a bug with the presenter’s web browser, and not with the app itself, and otherwise everything ran smoothly, even with Professor Osama adding things to the app during the presentation. We got good feedback from all the Professors on a few aspects that we overlooked to further improve our app including: A feature to block and/or unfriend other user, using an API or github list of sensitive/”bad” words to use for our flagging system, and a notification for users whose posts were flagged detailing whether the post was approved or denied (and why it was denied). We would like to also improve on the other features we already have (such as allowing for deletion of comments) as well as polish the app as a whole (improving security and UI components). Knowing that all the main functionality is

there and working, we plan to work on making everything run better for the final presentation on ECE day.