

ECE 366 Project Prototype 3

Symtagram: A Symmetric Social Media Platform

Yuecen Crystal Wang

Danny Hong

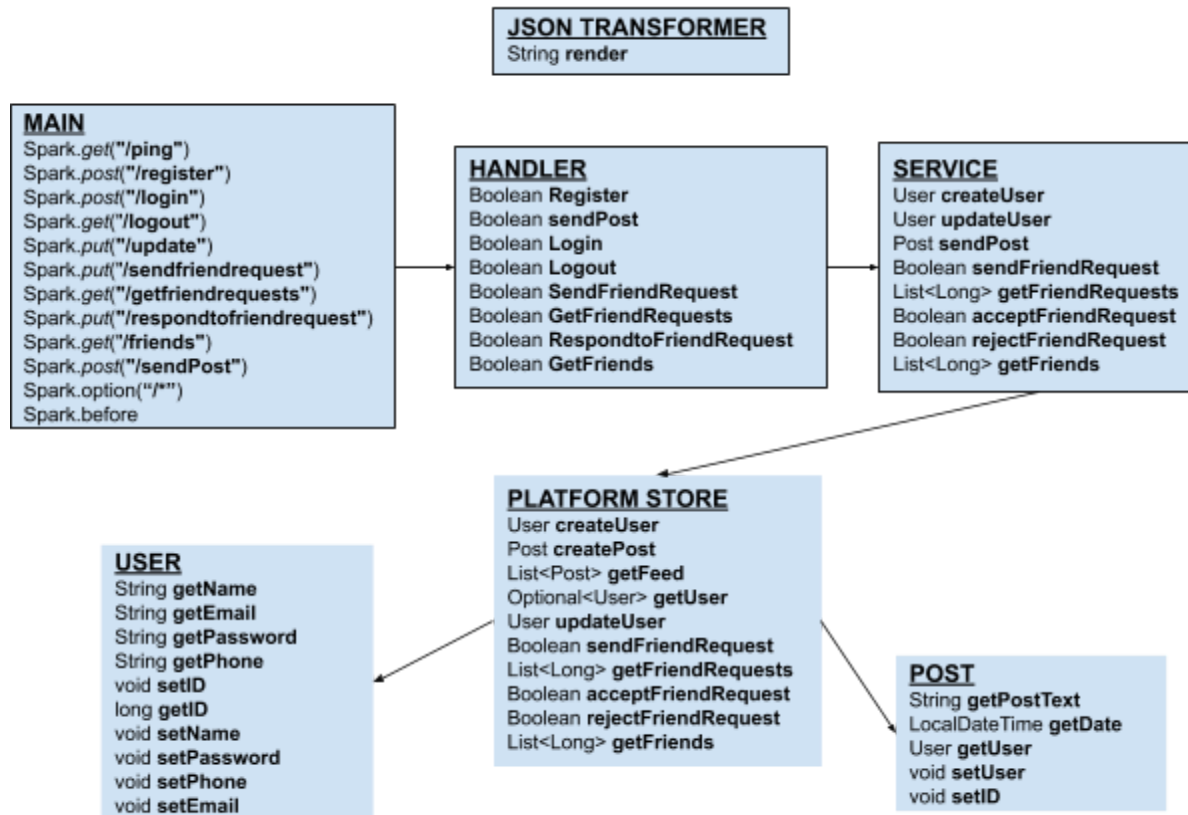
Brian Chung

Backend Development

We first attempted to develop week 1 deliverables with Spring framework. However, none of us had any experience with Spark framework, so getting the project started was very difficult. It also contained verbose configurations that we found complicated to understand. As a result, we presented a very disjointed deliverable. During the middle of the semester, our team decided to switch our backend framework from spring to spark. The main reason was because our professor did have experience in Spark, and we needed a mentor with experience to guide our development. Our switch to Spark greatly sped up development progress, and we were able to present a single system with a working in-memory back end for week 2. Additionally, we also reorganized our code and made changes to the structure of the program.

Current Backend Overview

Our system consists of multiple layers. The separate layers kept our project organized, which then made development, integration, and debugging much easier. The bottom-most layer, PlatformStore class, is responsible for setting up our MYSQL database that contains our implementations for our social media platform. Above layers are the handler and service classes. The JSONTransformer class converts input data in JSON form to strings, the service class is responsible for fetching information from the database by calling on platformStore, and the handler manages user requests that allow for communication between the user and the program. Our core system objects are the User and Post classes.



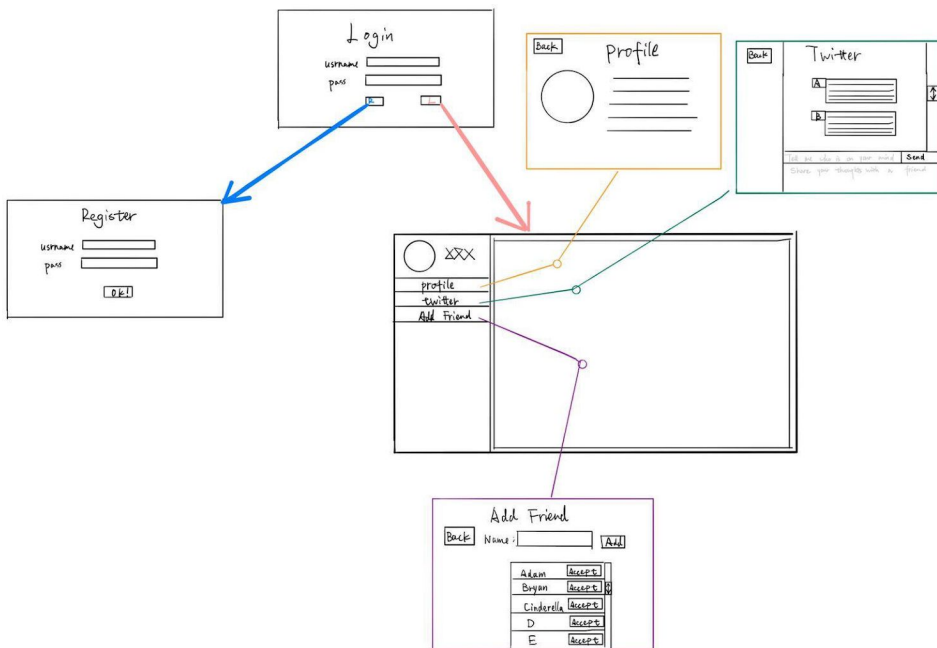
Shown above is a block diagram of our final backend design with each block in our diagram representing a java source code class. The diagram explains the relationships between the classes. At the base layer are the User and Post classes, which are the core objects that our system was designed to manipulate and modify. The next layer is PlatformStore, which deals with directly inserting, deleting, retrieving, and updating User and Post objects into our databases. We used MySQL databases with an Apache server to host persistent data storage. The PlatformStore layer communicates with the databases with JDBI and MySQL commands. Service then directly calls on PlatformStore to retrieve the data from the database and return it to the client. The next layer, Handler, is responsible for handling requests made from the client to the server. The Handler extracts and loads data in the request and response objects. It also makes

the appropriate calls to the service layer, passes in data to the service layer, and returns data from the databases to the front end. This layer also implements error checking to ensure that the request is valid. To report the final result of each request, the Handler layer returns HTTP response codes. Finally, the main function contains the spark requests that sets up the user interface by connecting each specific user request to their respective function in Handler. There are also Spark requests “spark.option” and “spark.before”, which authenticate connections from our backend to our frontend. Lastly, the main function initializes the Service, Handler, and JSON Transformer (Converts the user input data in JSON form to strings) objects and establishes a connection to the MYSQL database.

Frontend Overview

As shown below, this is how our web application looks. It starts off with a login page that requests a returning user to enter username and password to login. For new users, there is a tab that when clicked, will redirect the user to a register page to create an account. Once logged on, the user is directed to a main page in which an icon representing a user's profile picture would be displayed along with three tabs underneath it: profile, twitter, and add friend. Unfortunately, due to time constraints, we were unable to allow a user to upload their own profile picture in place of the icon. If the user clicks on the profile tab, the user will be directed to their own profile page, which displays their profile picture icon along with the information they provided during signup: their name, email, phone number, and password. If the user clicks on the twitter tab, the user will be redirected to a posting page where the user can write a post and send it to a desired friend user of their choice. Finally, if the user clicks the add friend tab, the user will be directed to a page

that requests the user to input the email of the user that they would like to add. This page was also supposed to display the user's friend list, but due to time constraints, we were unable to implement this.



Future Work

Giving more time for the project, we would like to do more edge case checking within each function, and reflect and display exactly what the error is on the screen. We also did not have the opportunity to implement a verifyPassword mechanism in which either a registering user or an existing user who wishes to edit his/her password would have to retype their chosen/corrected password for confirmation. We would love to keep working to have user's profiles with images, implement logout() function in the frontend along with a fixed part for getting friend requests.

What went well/bad

Our group found success in being able to integrate everyone's code together. This was a problem that we all faced in the beginning, but after our switch in backend frameworks from Spring to Spark, it was easier for us to understand how each part of our project connects with one another since the Spark documentation contained useful API. We all also had problems setting up and connecting to our MYSQL database, but we resolved this issue by installing MAMP, which we then used to run a live web server and initialize the MYSQL database.

Our group faced a number of technical and nontechnical challenges throughout the course of the semester. Crystal sometimes has to pick up or take over whatever it's left for the assigned part for other members as she usually puts in extra time to fix some bugs and does more research and would build a sample function/page for other members to learn from. Throughout this semester, some of us found it hard to understand a certain concept or technology, especially during this difficult time. The process of learning isn't that smooth, but we are also glad that we put in the efforts to understand a lot of core software engineering technologies.

Acknowledgement

We would like to appreciate and thank Professor Lusterman for helping and guiding us throughout the whole building journey, Tiffany's, Dave's and Rebecca's group for helping out, and special thanks to Crystal's friend Harvey Li and Karina Zhang from University of Wisconsin who helped us when building the frontend. Also, we acknowledge YouTube and Stack Overflow.