Assignment-1 COMP3095

Group Members: Nash Gil - 101024415, Haider Farooqui - 101292102

**Creating SpringSocial microservices**

|  |  |  |
| --- | --- | --- |
| Tasks/Objectives | Completed | Not Completed |
| Create user-service | X |  |
| Create post-service | X |  |
| Create comment-service | X |  |
| Create friend-service | X |  |
| BootStrap Information for user data in relational database | X |  |
| Postman Export Collection | X |  |
| Private Git Repository | X |  |
| Containerization of SpringSocial service and its microservices | X |  |
| Docker-Compose file | X |  |

**Approach:**

The approach taken to begin this project was very similar to that of our in-class exercises. Going with this method, helped us achieve the basic architecture framework of how our project would be. After having the basic architecture framework, we began to implement the JPA and Postgres relational database. It was a little different than the MongoDB approach we had done in class, but with the resources available we were able to implement the relational database. We also used Rest Templates which conveniently helped us make the relationship between users, posts, and comments to posts and comments. Using the Rest Templates, we were able to retrieve the userId from the user-service along with the name of the user so we could tie it to either a post made by that user, or a comment made by that user. We were also able to link posts and comments together so that users can comment on posts using the postId and the userId. The friend service lets users make a friend request and using the requestId they can either accept or reject the request. Users can also check the friends list of a user based on the userId and remove friends using the friendshipId.

**Challenges:**

Initially we coded the user-service, post-service, and comment-service the same way we had done the product-service in class. This let us have a solid foundation with code that was functional. The main challenge was implementing the relational database while working with JPA and PostgreSQL. We are not familiar with JPA or PostgreSQL, so it was a bit of a challenge initially to get our micro-services converted to JPA and PostgreSQL, but we eventually got a hang of it. The next challenge was getting the inter-service communication completed and it took a bit longer than we thought it would. We discovered that we could use Rest Templates to make this communication and were able to bring data over from other micro-services and make a relationship between them. This assignment gave us a challenge but ultimately, we were able to apply what we learned in class. We also ran into a problem with our endpoints for our Rest Template not working after containerizing the micro-services. It took some time to figure out that we had to remove the localhost and replace it with the container name in the implementation code. Coding these micro-services became clearer to us as we were coding them on our own and figuring it out. We feel more confident in our ability to make micro-services after completing this assignment.