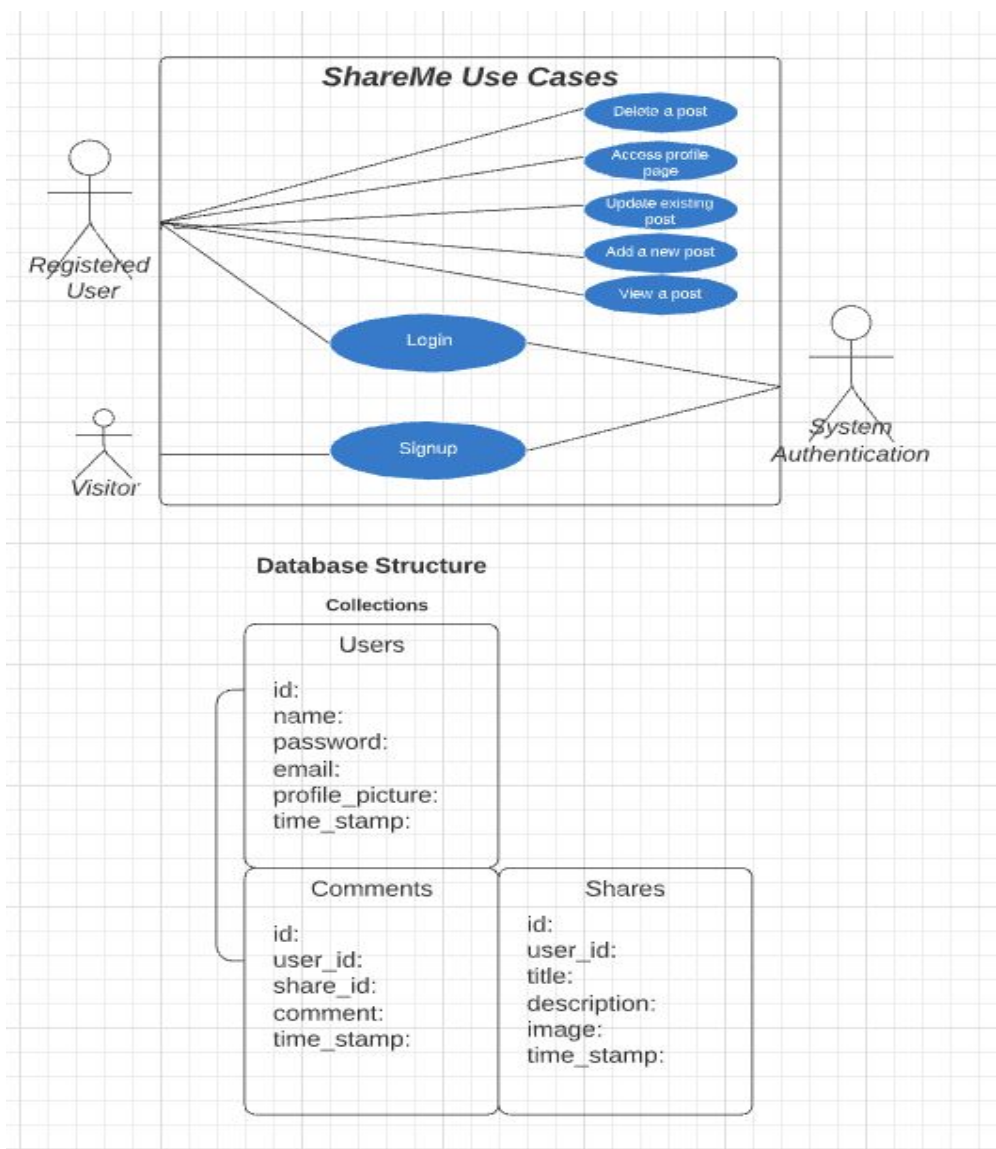


ShareMe Development Documentation

This documentation is a reflection of my collaboration and my personal technical work for this application development.

Introduction

For my application development, my group of 5 members discussed, planned, designed and coded over a 4 week period, a functional online sharing app named "ShareMe." ShareMe allows people to create their own account, then login, so they can view or add posts that get stored to our own database, and then retrieved to display them on the main "feed" page.



User Testing

ShareMe's design was created on Figma based around the concepts of a sharing app and our combined level of expertise with coding etc. During each standup meeting I contributed to the post layout page and where each variable for data e.g the username will be located on the share card so the data can match the display.

After the completion of the first design our group conducted a small survey that asks important questions about the usability and aesthetics of the design. I was responsible for obtaining 1 colleague/candidate to complete the survey form for us, I did not write the survey. The stakeholders were not involved in our group and gave an honest review.

My Participation

In the first week we did the planning and design on Figma. During this time I was planning the database structure and drawing the UML diagram shown above, with a colleague. I was the database administrator so I planned the database structure and created the MongoDB database with all the collections and data inside. I was also the server administrator with a colleague.

I was attending every single daily standup meeting and offered to do the database side of the project. I briefly spoke about my opinion on the design because I needed to follow the posts page so the database could match the displayed content.

After the first week of discussing the scenario and creating the design using Figma, we created the online master repositories for both Server & App via Github. I created the server repository so I could collaborate with a colleague who worked side-by-side with me on the backend server. I created the schemas and imported the necessary modules for the server, and then later on I coded the routers for the users and the shares/posts. These routers were for read, update, delete and create.

During my second sprint I made a few commits to the group React App project and these contributions are published on the Github repository. During this time we are at the React App Development and moving the HTML/CSS template into the XML Javascript Format. I am attending every single standup meeting and letting everyone know how the server and database is doing and asking other group members what they are up to so I am up to date.

In retrospect I made a lot of progress during Sprint 2 as this is when I was putting in the most work and getting the server and database up and running. Helping with the react app and following along with the team by attending 100% of the class time made it easier for me because I knew where everyone was at, and when I made a contribution, I did not overwrite anyone's work and I could ask anyone freely.

In the third week, I branched off from the master to make my own personal version of the ShareMe App to code for myself and create my own CRUD as part of the assessment requirements. I did change many things from the original master repository so I could implement my CRUD with my own database. Overall: The descriptions above of the development is what I contributed over the timeline. My Github commits, 100% attendance & voice cooperation is self-evident.

Group App Repository: <https://github.com/jdb00/ShareMe-React->

Group Server Repository: <https://github.com/jayfrost321/ShareMe-Server>

NPM

Node Package Manager (NPM) was essential to my project, to import the modules that the app will use, in which there for both server and the react app. Firstly I ensured NodeJS was installed on the system to be used. Using the system's terminal I used NPM to import Mongoose, Logger, Express, Fileupload and body-parser for the server. You can view my code to see the project dependencies in use.

Technical Options

There are 2 main options I face to build my front end application. MEAN stack or MERN stack, both are similar except MERN uses ReactJS while MEAN uses AngularJS. React is more of a javascript Library because it is smaller, and has great speed advantages and is more lightweight.

Personally, this benefits me because I only require a knowledge of JSON & JavaScript, and because my application will not be using an extensive database or be very large.

Angularjs is a fully fledged framework while React is a library. Both differ in size primarily which is why I am working with Reactjs. Angularjs is used mainly for larger and complex enterprise applications or social media applications. Angularjs applications can use one language for both the front and back end, client and server side.

Both mean and mern stack use a combination of technologies which are popular and backed by a community, using mongodb, express and nodejs. The mongodb database structure uses json formatting which is very binary and cost efficient. From experience you can store tons of data and still is very responsive. It's easy to retrieve data from this kind of API using javascript, as it is not sql and you can relate it to any application.

Another database manager I considered was firebase, which emphasizes more on mobile applications than web/general purpose applications. It has some great features like monitoring application usage through a console, firebase instances etc.) but mongodb offers more features and is much more general purpose, which at my level of expertise and what my app is trying to achieve, mongodb suits me better.

Coding Conventions

1. All of my JavaScript functions use camel-case name formatting beginning with lower case and next words start with an uppercase letter.
2. I have made sure that the event handler functions in the react app consistently begin with the word "handle."
3. All components begin with a capital letter.
4. Route components start with "route" as I am using front Routes to navigate between pages.
5. Server models must be referred to as singular.
6. Collection names inside the database are pluralized

Group Reflection

Everyone in the group did participate and gave their best. I am proud of the effort from other members and was able to make a commitment to this project because I was in a stable group.

Communication at the beginning was strong, and the daily meetings were focused around planning for the future of development and contributing to the design. Communication in the group slowed around week 3 and 4 when we were all working on our own branch.

The design was mediocre in my opinion, but it did the job and satisfied what I offered to contribute by doing the backend and server setup. I did end up changing the design later on in my own version. I was happy with the collaboration on Github as it was actively being updated and developed by all members.

All of our project management was done on Trello. Trello is a web based tool that keeps track of everything and is extremely flexible for us. We displayed our sprints and checklist dates by which things need to be completed, and what group members were working on what. We used trello to post links as well.

What could be improved for next time

I could perhaps have contributed more code in the group collaboration, as the bulk of my code and focus was in my own branch/version of the app where I changed much of the design.

The deadlines could have been met more swiftly, and just general timing of finishing tasks but this could have been solved easily with questioning more where other members were working.

I would also double down on my expertise level in the group and ensure that the goal we are aiming for is suitable and can actually be met. For example how complex the app is, how functional, number of components, etc. I noticed the original design and app was slightly over my experience level, I did keep up but I modified the design during Sprint 3 in my own crud to suit me.

Overall ShareMe is a success and worked well for me.
