

Seenit

by
Chaz Chang
Viet Dinh
My Nguyen

TEAM 24 - CS157A

Project Description

Overview:

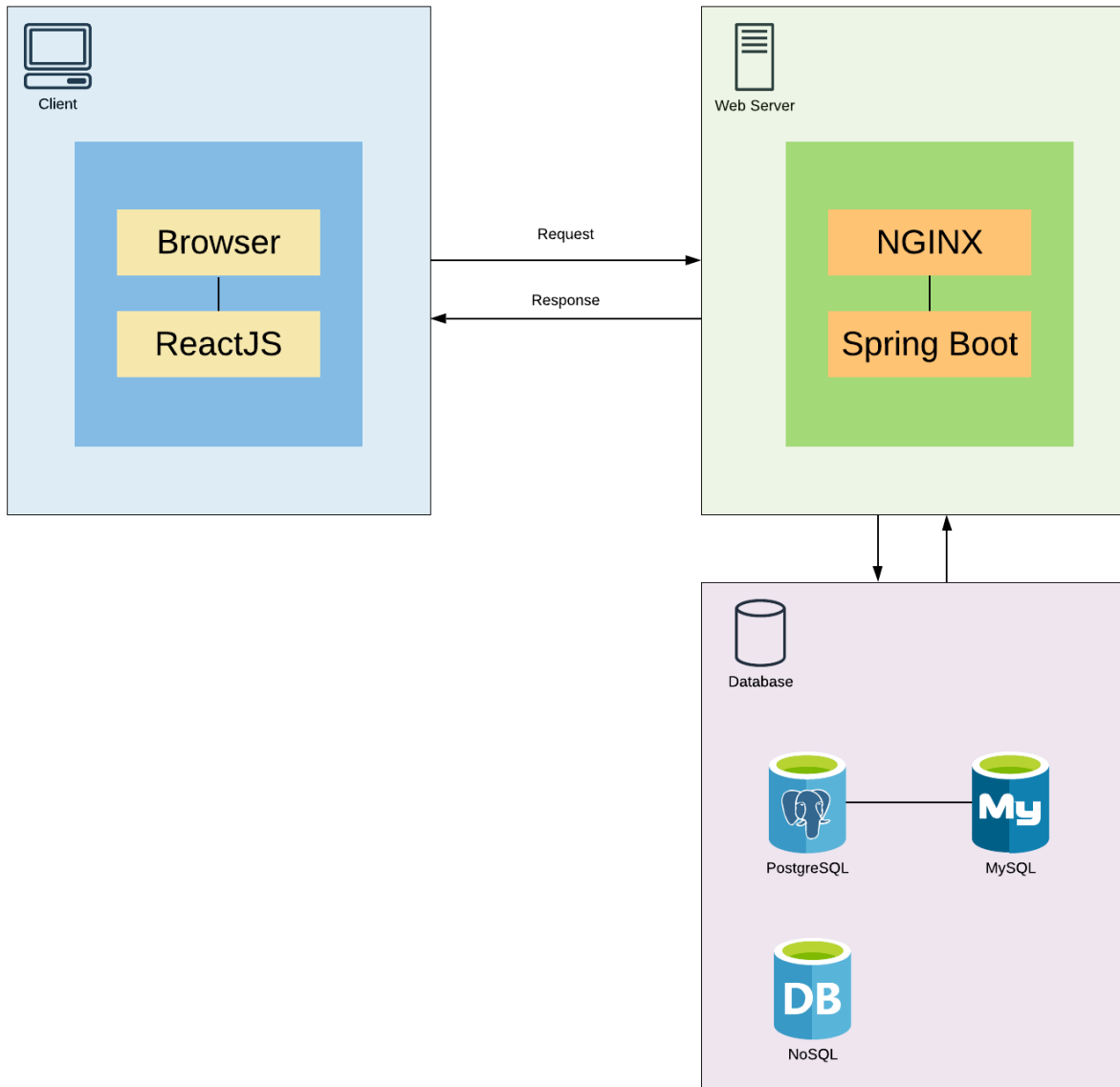
The goal of this application is to create social news, rating site, and discussion website. Users can be able to check daily news, search, filter, and sort through posts. Also, users can create posts, comment on them, like, and share posts. In addition, users can create channels that will function as communities. Users can view posts without logging in, but creating posts, commenting, and liking will require the user to login. This application will allow people to easily voice their opinion or share a story. People can easily ask questions and others can answer them. It will also provide a user-friendly interface and fast response time while maintaining high throughput.

Who would be interested in this application?

Users could be anyone who is interested in forming a community online, and interact with people with similar interests. Users can keep up to date with news, information, and ideas in their community. Companies and organizations can use Seenit to communicate with people and market products.

System Enviroment

Structure of the system (graph based on 3-tiered architecture):



HW/SW used:

- ReactJS
- NGINX
- Spring Boot
- Windows/macOS
- MySQL Workbench
- Visual Studio Code

RDBMS Used:

- MySQL Version 5.6

Application languages:

- Java (with Spring Boot framework)
- SQL (with MySQL)
- JavaScript (with ReactJS)
- HTML/CSS

Functional Requirements

How users will interact with the system:

The application provides the same functionality for all the users. Internet connection is required for any activity on the system. A user shall be able to access the system for read and write. A user can access the system home page freely. They shall be able to read daily news and posts from various websites without logging in. Users can also be able to make a search based on their particular interests. However, users must log-in into the system and have their credentials checked for other activities such as make a new post, comment on a post, share a post, create a channel, and follow another channel. First-time users must register for an account to perform the activities listed under log-in.

Describe each individual function, functional process and I/O:

Functions

Log-in

- Returning users must sign-in to perform the following activities:
 - create or access personal channel
 - add a new post
 - like or comment on someone else's posts
 - share a post
 - follow another channel
 - delete personal posts
- Guests or unregistered users can only view posts
- The system shall check the users' credentials. If the entered information is not matching with the information stored in database, an error message will be displayed; for instance, "ID or password is invalid." The users can either create a new account or reset password/username. Otherwise, the system gives users access when the information is matching.

Register for an account

- There will be only one type of user; no admin account in this application. First time user must sign up for an account to perform those activities listed in log-in function

- There will be a redirect link. User will be asked to fill out some information such as first name, last name, email address, username, and password.
- No limitation. Users can set up multiple accounts and choose which one to log-in with depending on the personal channel which users want to participate in.
- The system shall check if entered username is already existed. The system shall keep asking the users to enter a username until the name is available. Then, the system shall add the users' personal information into database.

Reset

- The users shall be able to reset password when they forget it.
- The system will send a verification code via the user's email.
- The users must check the provided email for verification code, and then enter the code to the verification code box to reset password. Users must request for a new code if they do not see it.
- The system shall verify the code. If the code is correct, the system will update the database after the users finish submitting new information.

Create a channel

- Once users are signed in, users can create their own channels (similar to social media accounts). On their channel, users are able to manage the role of other users, make a new post, delete a post, check out subscribers, and display personal dashboard.
- Users can create multiple channels (up to 4). By creating a channel, a user can get followers or subscribers who are interested in his/her activities.
- The system shall save information of the channel in database.

Search

- The users shall be able to search for daily news and posts by typing in keywords which appear in titles or headlines in search box. The users are also able to search for a channel by entering the name of that channel in a search box.
- If there is no matching results, a message will be shown to the user to indicate that what they look for was not found. Otherwise, display all the posts or news which are related to the searched terms.

Commenting

- The users shall be able to leave multiple comments on someone else's posts by click on the comment button, type what they want to say in the reply box, and hit enter.
- Once a comment is submitted , the data will be saved to the associated channel. They can later check which posts they commented on the channel dashboard.

Share a post

- When the users are interested in a particular post, or they think a post is useful for other people, they can be able to share that post. To share a particular post, the users must click on the share button under a post. The shared post will be automatically added to their channel, and others shall be able to view or share it.

- By sharing a post, users can always go to their channel to continue reading it if they do not have time to finish reading the post at once.

Add a post

- The users shall be able to write their own post on the channel. They can make a post by typing in the new post box and pressing the post button to submit. Users can attach pictures, video, or link to their post. The created posts will be visible on the channel.
- The system shall save all the contents in the database after posting is finished.

Delete

- The users shall be able to delete their comments on a particular post. They can go back to the channel to check the location of the comments. Users can delete a comment by clicking the delete button under each comment, and confirming that they are sure about the deletion.
- The users shall also be able to delete their posts. Similar to delete comments, users can delete a post by clicking the delete button under each post, and confirming that they are sure about the deletion.
- The system will remove the data permanently from database after deletions are completed.

Follow a specific channel

- The users shall be able to follow a particular channel that they are interested in. They will get updates on the new activities of that channel. The users can subscribe for a particular channel by clicking the follow button on the channel page.
- The system shall add the changes to database to keep track of the channels that a user is following.

Display personal dashboard

- User can see how many posts they have opened
- Display how many point/upvote they have
- Display their saved posts
- Display their like/unlike posts
- Get notifications if someone replies to his/her comment

Display dashboard of other users

- By clicking another user's profile, user can see the point/like, posts, etc of that specific user
- When clicking a user's profile, a modal layer will pop up. To exit, simply click outside of the modal.
- User can't view other users' private information

Filter

- Using to filter the posts by date
- Including today, week, month, year

- User can active the function simply by clicking the filter button and selecting from a drop down menu

Sort

- Using to sort the posts by Top, New and Hot
- Sort by Top will sort the posts by the points/upvotes they have.
- Sort by New will sort the posts by date
- Sort by Hot will show the posts that have most attention recently
- User can active the function simply by clicking the sort button and selecting from a drop down menu

Non-functional Issues

Graphical User Interface (GUI):

There are many design principles when it comes to web design. For our website, we will use seven most popular principles, which are Visual Hierachy, Divine Proportions, Hick's Law, Fitt's Law, Rule of Thirds, Gestalt Design Laws, and White Space and Clean Design.

- Visual Hierachy: Certain parts of our website will be more important than others. We want to make those parts easily been seen and noticed by users. For examples the account button, the scrolling posts, the filters, and the search box.
- Divine Proportions: The layout, the size of each components should follow the golden ratio which is 1.618. For example, if the layout width is 1200px, the width of the content area should be 742px.
- Hick's Law: "Hick's Law says that with every additional choice increases the time required to take a decision." So, we plan to minimum the options for dropdown menu buttons. This will encourage new users tring new functions.
- Fitt't Law: Button's size needs to follow a set of rules. The size of the button is proportion to its using-frequency.
- Rule of Thirds: Since our website will allow users to upload pictures. The size of a picture needs to follow the rule of thirds to make it more interesting.
- Gestalt Design Laws: Filter buttons, sorting buttons will be grouped together. Buttons will have consistent sizes.
- White Space and Clean Design: Website without white/blank space is hard to navigate. So, we will use white space to divide the components, boxes that have different functions.

Security

- User accounts need to be highly protected. We don't want their personal information leaked or hacked.
- Passwords won't be store in blank text, it will be hashed.
- We will use https protocol for our web server to make sure the connection between users and server are encrypted.

Access Control

- Anyone with internet can access the website
- A user will be able to view posts/channels and search without logging in
- A user must login to create posts, comments, or channels
- A user cannot edit posts or comments that belong to a different user

Performance

- Fast response time while maintaining high throughput
- The MySQL database will be optimized so queries don't take too long. The right data types and efficient SQL queries will make the database accesses faster and the database size smaller
- ReactJS will be used to build the User Interface. ReactJS will allow the user to navigate through the application quickly by dynamically changing the current page instead of loading a whole new page from the server
- NGINX can be used as a load balancer but it will require money for more resources like computers and databases. We won't use many resources during this semester, but this makes it scalable for later

Scalability

- Able to add new functions and features while developing the app
- NGINX will be used on the web server. NGINX can be used as a load balancer to distribute the work over many computers. Load balancing will help create high throughput and low response time by avoiding overloading a single computer. NGINX will allow more computing resources (CPU and memory) to be easily added to the existing system