Fake Stack Overflow Specifications

Introduction

This document describes the specification of the Fake Stack Overflow Application. This application is inspired by the actual stackoverflow.com. The application will be developed using the React framework for the front end aspects. For the backend, the application will use MySQL, a relational database, or MongoDB, a non-relational database. The application server will be in Nodejs using the Express framework for server-side routing. The application secures sensitive data such as passwords using the bcrypt library in Node.

Use Cases

Use-Case number	1
Use-Case Name	Create Account
Actors	Unregistered user
Preconditions	The user has a working internet connection, has loaded the application page, and is not currently logged in.
Postconditions	A new account is created for the user with their username, a secret password, and an

	account name (email).
Story	The user arrives at the welcome page of the application. The page shows options to register as a new user, login as an existing user, or continue as a guest user. They select the register as a new user option. On selecting this option, they enter their username, email, a secret password, and a password verification. The user presses the SignUp button. This saves the information in a database. They are then directed to a Login page.
Exceptions	No users can create an account with the same email. The email should have a valid form. The typed password should not contain the username or the email id. Nicely styled feedback must be presented to the user if the account could not be created due to the above reasons or any other reason.

Use-Case number	2
Use-Case Name	Login
Actors	Registered user
Preconditions	The user has a working internet connection, has loaded the application page, and is not currently logged in.

Postconditions	Login successful or failed.
Story	The user arrives at the welcome page of the application. This page has options to register as a new user, login as an existing user, or continue as a guest user. They select the login as an existing user. On selecting this option, they are asked to login with their email and password. Upon pressing the login button, they are taken to the home page if login is successful.
Exceptions	The user enters an unregistered email or an incorrect password then the application should report back appropriate feedback to the user on the same page.

Use-Case number	3
Use-Case Name	Logout of account
Actors	Registered user
Preconditions	The user has a working internet connection, has loaded the application page, and is currently logged in.
Postconditions	The user logs out of the application.
Story	The user is at a page with a logout button. They click on the logout button and the user is taken back to the welcome page if log out

	was successful.
Exceptions	The application should show an error message if the log out failed due to a communication error or a database error.

Use-Case number	4
Use-Case Name	Home Page
Actors	Guest user
Preconditions	The user has a working internet connection, has loaded the home page, and is currently not logged in.
Postconditions	The contents of the home page.
Story	The user has entered the home page as a guest user. The home page displays a horizontal menu at the top of the page with options to view all questions, view all tags, and a search box to search for questions. Initially, the home page shows a list of all questions in the database. The most recent question appears first. For each question in the list it displays, the question title, question summary, the list of associated tags, the no. of times it has been viewed and voted, the no.

	of answers it has, the username of the questioner and the date (e.g., Jan 12, 2020) and time it was asked (e.g., 21:39:32). The question title is a link which when clicked shows the answers page. The page displays 5 questions at a time. The list of questions is scrollable to fit all the 5 questions if they overflow. For more than 5 questions 2 buttons – *next*, and *prev* are displayed at the bottom of the list. The *next* button displays the next 5 questions, *prev* displays the previous 5. Button *next* is disabled when the last 5 questions are shown. Similarly, when displaying the first 5, *prev* is disabled. Finally, these buttons are outside the scrollable list of questions.
Exceptions	The application should show an appropriate error message to the user in case of a system failure or a communication failure. In case of an error, the user should be given a way to go back to the welcome page and restart.

Use-Case number	5
Use-Case Name	Home Page
Actors	Registered user
Preconditions	The user has a working internet connection,

	has loaded the home page, and is currently logged in.
Postconditions	The contents of the home page.
Story	The user has entered the home page as a registered user and is logged in. The home page displays a horizontal menu at the top of the page with options to view all questions, view all tags, a search box to search for questions, and logout of the application. It also displays the username of the user as a link which when clicked opens the user's profile page. The menu also has an option to post new questions. When the home page loads, it displays a list of all questions in the database. The most recent question appears first. For each question in the list, the question title, question summary, the list of associated tags, the no. of times it has been viewed and voted, the no. of answers it has, the username of the questioner and the date (e.g., Jan 12, 2020) and time it was asked (e.g., 21:39:32) is shown. The question title is a link which when clicked shows the answers page. The page displays 5 questions at a time. The list of questions is scrollable to fit all the 5 questions if they overflow. For more than 5 questions 2 buttons – *next*, and *prev* are displayed at the bottom of the list. The *next*

	button displays the next 5 questions, *prev* displays the previous 5. Button *next* is disabled when the last 5 questions are shown. Similarly, when displaying the first 5, *prev* is disabled. These buttons are outside the scrollable list of questions.
Exceptions	The application should show an appropriate error message to the user in case of a system failure or a communication failure. In case of an error, the user should be given a way to go back to the welcome page and restart the application.

Use-Case number	6
Use-Case Name	Searching
Actors	Registered user, Guest user.
Preconditions	The user has a working internet connection, and has loaded the home page.
Postconditions	The user is shown a list of questions that match the search text.
Story	The user has entered the home page which initially displays a list of all questions. The home page displays a horizontal menu at

	the top of the page similar to use case 4 and 5. In the search box, the user enters search text. The search results include all questions that contain at least one word in the search string in their question title or summary or the question text. For search text with individual words enclosed within [] then the search result includes all questions corresponding to each tagname within [], The search results should also be displayed 5 at a time with buttons to view the next and previous as described above.
Exceptions	Show “no results found” in case of search failure.

Use-Case number	7
Use-Case Name	All Tags
Actors	Registered user, Guest user.
Preconditions	The user has a working internet connection, and has loaded the home page.
Postconditions	Shows all tags in the database.
Story	The user has entered the home page that

	initially displays a list of questions. The home page displays a horizontal menu at the top of the page similar to use case 4 and 5. When the user selects the *all tags* option from the menu, a list of all tag names is displayed to the user. Each tag name in the list is a link and shows the no. of questions associated with the tag. Upon clicking the link, the list of questions associated with the tag name is displayed.
Exceptions	Show an appropriate message if there is a system or communication failure below the horizontal menu. The message should be stylized so the message is visible clearly to the user.

Use-Case number	8
Use-Case Name	New Question
Actors	Registered user.
Preconditions	The user has a working internet connection, has loaded the new questions page, and is currently logged in.
Postconditions	The new question posed by the user is added

	to the database.
Story	The user has entered the new question page. They see a horizontal menu as described in the home page use case and a form to enter new question details. The form has the following: - A field to enter the question title (max. 50 characters). - A field to enter the question summary (max. 140 characters). - A field to enter the question text. - A field to enter tag names. Tag names are separated by whitespace. A new tag name can only be created by a user with reputation 100 or more. - A button to create a new question. Pressing the new question button, saves the question to the database and loads the home page, which displays the new question along with other questions in the database.
Exceptions	An appropriate error message should be displayed if the above mentioned constraints are violated.

Use-Case number	9
Use-Case Name	Answers

Actors	Guest user.
Preconditions	The user has a working internet connection and has loaded the answers page for a particular question as a guest.
Postconditions	Show a question and all its answers.
Story	The user has entered the Answers page and sees a horizontal menu at the top of the page as described in the home use case. In a separate section, below the menu, the page displays the question title, the total no. of answers, and the total no. of views (including the current view). Below this information, the page displays the question text, the list of tags, and the username of the user who asked the question along with the date and time the question was asked. It also displays the no. of votes the question has received. The list of comments associated with a question are displayed below the question text and its list of tags. (see later for *comments* use case.) Below the question details section, the page displays the list of all answers for the question. The most recent answer should appear first. Each answer has the answer text, the no. of votes it has received so far, and the username of the answer giver along with the date and time. It displays 5 answers at a time with two buttons – *next* and *prev* to display the next 5 answers and the previous 5

	answers, respectively. The *next* button is disabled when the last 5 answers are shown and the *prev* button is disabled when the first 5 answers are shown. Answers also have comments. They should be displayed below each answer.
Exceptions	An appropriate error message is displayed if the expected data could not be rendered due to system or communication failure.

Use-Case number	10
Use-Case Name	Answers
Actors	Registered user.
Preconditions	The user has a working internet connection, has loaded the answers page for a particular question, and is currently logged in.
Postconditions	A question's details, all its answers, and ability to add new questions, answers and comments.
Story	The user has entered the Answers page and sees a horizontal menu at the top of the page as described in the home use case. In a separate section, below the menu, the

	page displays the question title, the total no. of answers, and the total no. of views (including the current view). Below this information, the page displays the question text, the list of tags, and the username of the user who asked the question along with the date and time the question was asked. It also displays the no. of votes the question has received so far. The list of comments associated with a question are displayed below the question text and its list of tags. (see later for *comments* use case.) Below the question details section, the page displays a scrollable list of all answers for the question. The most recent answer should appear first. Each answer has the answer text, the no. of votes received, and the username of the answer giver along with the date and time. It displays 5 answers at a time with two buttons – *next* and *prev* to display the next 5 answers and the previous 5 answers, respectively. The *next* button is disabled when the last 5 answers are shown and the *prev* button is disabled when the first 5 answers are shown. The buttons should be outside the scrollable list of answers. A button to add new answers is displayed below all the answers, in a separate section. This button is outside the scrollable list of answers. Answers also have comments. They are

	displayed below each answer. The question and each answer has buttons to upvote and downvote the question or answer. Upvoting increases the vote by 1 and downvoting decreases the vote by 1. Upvoting a question/answer increases the reputation of the corresponding user by 5. Downvoting a question/answer decreases the reputation of the corresponding user by 10. The user can vote if their reputation is 100 or more.
Exceptions	An appropriate error message is displayed if any of the constraints above are violated or there is a database or a communication error.

Use-Case number	11
Use-Case Name	Comments
Actors	Guest user.
Preconditions	The user has a working internet connection, and has loaded the answers page for a

	particular question as a guest.
Postconditions	Shows the comments for a question and all its answers.
Story	A question or an answer has comments. The most recent comment is displayed first. For both Q&A, 3 comments are displayed at a time along with the username of the commenters. Two buttons – *next* and *prev* display the next 3 comments and the previous 3 comments, respectively. The *next* button is disabled when the last 3 comments are shown and the *prev* button is disabled when the first 3 comments are shown.
Exceptions	N/A

Use-Case number	12
Use-Case Name	Comments
Actors	Registered user.
Preconditions	The user has a working internet connection, has loaded the answers page, and is

	currently logged in.
Postconditions	Shows comments for a question and answers and adds new comments.
Story	A question or an answer has comments. The most recent comment is displayed first. For both question and answers, 3 comments are displayed at a time along with the username of the commenters. Two buttons – *next* and *prev* display the next 3 comments and the previous 3 comments, respectively. The *next* button is disabled when the last 3 comments are shown and the *prev* button is disabled when the first 3 comments are shown. A text field is displayed below the comments to capture a new comment by the currently logged in user. The new comment is added to the database and displayed on pressing enter.
Exceptions	- If a new comment is more than 140 characters then do not add it to the database and display an appropriate error message to inform the user. - If a new comment is added by a user with less than 100 reputation, display an appropriate message to inform the user and reject the comment.

Use-Case number	13
Use-Case Name	New Answer
Actors	Registered user.
Preconditions	The user has a working internet connection, has loaded the new answers page, and is currently logged in.
Postconditions	The new answer posted by the user is added to the database.
Story	The user has entered the new question page. They see a horizontal menu as described in the home use case. Below the menu a text field is shown for the user to enter their answer and a button which when pressed saves the answer in the database. After posting the answer the user is taken back to the answers page for the question.
Exceptions	An appropriate error message should be displayed on this page if a database error or communication error happens.

Use-Case number	14
Use-Case Name	User Profile
Actors	Registered user.
Preconditions	The user has a working internet connection, has loaded the user profile page, and is currently logged in.
Postconditions	The user profile of the currently logged in user.
Story	The page displays the horizontal menu as described in the home use case at the top. Below the menu the length of time the user has been a member of fake stack overflow and the reputation of the user is shown. Below this information, the list of question titles ever asked by the user is displayed. The list of questions displayed is similar to the list of all questions in the home page. A sidebar shows links to view all answers and tags created by the user. They are displayed in the same way that all tags and and all answers for a question are displayed. Selecting a question, answer, or tag allows the user to edit or delete the respective question, answer, or tag.

Exceptions	Display an appropriate message if the user has no questions, answers, or tags.

System Requirements

You must use the React framework along with CSS to design and implement the front end. You can reuse and extend the CSS and React components you created for the homework assignments. The react app should run on localhost:3000.

The server should run on Node and should connect to a backend database. The server should run on localhost:8000.

For the backend, you can use either MySQL or MongoDB. For either system, your application server should connect to a local instance.

For MongoDB, the server-side script should connect to *mongodb://127.0.0.1:27017/fake_so*. Your server should set up the backend documents based on the schema you have designed.

For MySQL, the script should connect to an instance with the configuration:

host : 'localhost',
user : user,
password : pass,
database : 'fake_so'

User and Password for MySQL instances should be provided as arguments to the server script identical to the format described in HW4.

If using MySQL, you should include a script called *server/setup.js* in your repository that will be run to create the tables in your schema. This script will be run using Node before starting the server.

For both MongoDB and MySQL, you can assume the local installation is up and running when your server is started.

You should provide a design document for the schema of your database. For a relational database provide an E/R diagram. For non-relational, provide a class diagram. Include these images in the *image* directory of your repository.

User passwords need to be stored securely in the database. Hence, we should hash the passwords and store their hashes instead of the original passwords. To this end, install and use the bcrypt library in Node.

Your application should be designed to handle requests from multiple users.

The Repository

The repository has a directory for **server** and **client**. The file **server/server.js** is the main launching point of the application server. The **client** directory hosts the front-end resources of your application.

The *README.md* file should document your application and how to run it. This file should contain instructions on how to reproduce and run your application on another computer. **If you have used design patterns while implementing your application, you should list them in the README and explain where you have used them and why you used them. You will get 10% extra credit for using design patterns in your code.**

You are allowed to use any library you deem fit. Make sure they are part of **packages.json**.