

Fake Reddit (Phreddit) Specifications

Term Programming Project

CSE 316 – Fundamentals of Software Development

Spring, 2025

Assigned: Monday, 04/14/2025

Due: **Monday, 05/12/2025**

Introduction:

This document describes the specification of the Fake Reddit (or Phreddit) Application. This application is inspired by the actual [reddit.com](https://www.reddit.com). The application will be developed using the [React](https://reactjs.org/) framework for the front end. For the backend, the application will use [MongoDB](https://www.mongodb.com/), a non-relational database. The web server will run on [Node.js](https://nodejs.org/) using the [Express](https://expressjs.com/) framework for server-side routing. The application stores and uses passwords hashes (or digests) for authentication, using the [bcrypt](https://npmjs.com/package/bcrypt) library in Node.

The specification is organized as a collection of anecdotal user stories. Please read each use case carefully and take note of each component (actors, pre-conditions, post-conditions, story, and exceptions).

There are 19 use cases described below. Each use case begins on a new page. Please read the entire document (26 pages).

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 at localhost:3000.**

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

For the backend, you must use MongoDB. The web server should connect to a local instance.

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

Include a script called *server/init.js* in your repository that will be run to create initial data in your schema. This script will be run using Node before starting the server. The initial data must contain the user profile for admin. The email address, display name, and password for an

admin user must be provided as the command-line arguments to `server/init.js`. You must use these credentials to create a user profile for admin in the database.

You must produce and submit 3 UML diagrams:

1. The first diagram should be a UML Class diagram that depicts the data model you created for your MongoDB database. It should document each of the significant entities in your data model (anything with a schema).
2. The second diagram should be a UML Sequence diagram. It should depict the entities and method calls involved in the request for, and return of, Post data when the user clicks on a Post in a post listing and opens the New Post Page view.
3. The third diagram should be a UML State Machine diagram. The diagram should depict the transition over time in a significant component in your React component hierarchy. I am suggesting that you focus on the container component you are using for the main content area, but I will leave this to your discretion.

Save each diagram as a PNG/JPG in the *image* directory of your repository.

User passwords need to be stored securely in the database. Hence, **you must 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. Use the event loop and asynchronous features in JavaScript to enable this most efficiently.

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 must contain instructions on how to reproduce and run your application on another computer. It must also contain the list of features implemented by each member of the team. Without this you will not receive credit.

You are allowed to use any library you deem fit. Make sure they are part of **packages.json** so we can install them during evaluation.

Reputation:

Each user account has an associated reputation value. This value can change based upon upvoting and downvoting of content created by the user (See Use Case 14 about the Post Page

View for further details). Some user actions are restricted based on low reputation. Regular users should start with a reputation of 100. Admin users should start with a reputation of 1000.

Use Cases are presented on subsequent pages:

Use-Case #	1
Use-Case Name	Create Account
Actors	Unregistered user
Preconditions	The user has a working internet connection, is viewing the welcome page, and is not logged in.
Postconditions	A new account is created for the user with their display name, a secret password, and an account name (email address).
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 first and last name (in separate fields), email (use as account name), a display name, and a secret password (the password should be entered twice to verify correctness; the two password entries must match). The user presses the “Sign Up” button. This saves the information in the database. They are then directed back to the initial welcome page.
Exceptions	No two users can create an account with the same email or display name. The email should have a valid form. The typed password should not contain their first or last name, their display name, or their 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 #	2
Use-Case Name	Login
Actors	Registered user
Preconditions	The user has a working internet connection, is viewing the welcome page, and is not 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 #	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 give appropriate feedback to the user if the log out failed.

Use-Case #	4
Use-Case Name	The Banner
Actors	Guest user
Preconditions	The user has a working internet connection, is viewing an application page, and is not logged in.
Postconditions	The banner is presented to the user.
Story	<p>The banner should be present at the top of all application pages.</p> <p>The appearance and behavior of the banner should be the same as in PA03, with the following additions:</p> <ul style="list-style-type: none"> • The phreddit button should take the user to the welcome page. • There should be a user profile button added to the banner. For user's who are not logged in, the button should read "Guest" and it should be non-functional. • The "Create Post" button must be greyed-out and non-functional.
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 #	5
Use-Case Name	The Banner
Actors	Registered user
Preconditions	The user has a working internet connection, is viewing an application page, and is logged in.
Postconditions	The banner is presented to the user.
Story	<p>The banner should be present at the top of all application pages.</p> <p>The appearance and behavior of the banner should be the same as in PA03, with the following exceptions and additions:</p> <ul style="list-style-type: none"> • The phreddit button should take the user to the welcome page. • There should be a user profile button added to the banner. The user profile button should display the user's display name. • There should be a logout button added to the banner. When selected it ends the user's session and takes them to the welcome page.
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 #	6
Use-Case Name	The Navbar
Actors	Guest user
Preconditions	The user has a working internet connection, is viewing an application page, and is not logged in.
Postconditions	The navbar is presented to the user.
Story	<p>The navbar should be present on the left side of all application pages.</p> <p>The appearance and behavior of the navbar should be the same as in PA03, with the following exceptions and additions:</p> <ul style="list-style-type: none"> • The “Create Community” button should be greyed-out and non-functional.
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 #	7
Use-Case Name	The Navbar
Actors	Registered user
Preconditions	The user has a working internet connection, is viewing an application page, and is logged in.
Postconditions	The navbar is presented to the user.
Story	<p>The navbar should be present on the left side of all application pages.</p> <p>The appearance and behavior of the navbar should be the same as in PA03, with the following exceptions and additions:</p> <ul style="list-style-type: none"> • The list of community names should be sorted such that communities which the user has joined appear at the top.
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 #	8
Use-Case Name	Home Page View
Actors	Guest user
Preconditions	The user has a working internet connection, is viewing the home page, and is not logged in (selected “Continue as guest” from the welcome page).
Postconditions	The contents of the Home Page view or, if there is an error, the welcome page.
Story	<p>The user has entered the Home Page View as a guest user.</p> <p>The Home Page View should look and behave as described in PA03 (please refer to the specification document of that assignment).</p> <p>Each post in the post listing should look and behave as described in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Along with the view and comment count, there should be a vote count displaying the current vote value for the post with an appropriate label.
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 #	9
Use-Case Name	Home Page View
Actors	Registered user
Preconditions	The user has a working internet connection, is viewing the home page, and is logged in.
Postconditions	The contents of the Home Page view or, if there is an error, the welcome page.
Story	<p>The user has entered the Home Page View as a guest user.</p> <p>The Home Page View should look and behave as described in PA03 (please refer to the specification document of that assignment), with the following exceptions:</p> <ul style="list-style-type: none"> • The post listing should display posts from the user's communities first, and then posts from communities the user does not belong to after. • Each sublisting should be in the appropriate order (newest, oldest, active). • There should be a clear demarcation between the two sublists, but all posts should still be contained in a single list structure. <p>Each post in the post listing should look and behave as described in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Along with the view and comment count, there should be a vote count displaying the current vote value for the post with an appropriate label.
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 #	10
Use-Case Name	Community Page View
Actors	Registered user, Guest user
Preconditions	The user has a working internet connection and is viewing a community page (clicked on a community title from the nav bar).
Postconditions	The contents of the Community Page view or, if there is an error, the welcome page.
Story	<p>The user has entered the Community Page View for a specific community.</p> <p>The Community Page View should look and behave as described in PA03 (please refer to the specification document of that assignment), with the following exceptions:</p> <ul style="list-style-type: none"> • The display name of the user that created the community should be displayed alongside the created by date. • If the user is logged in, and the user is not a member of the community, then there should be a link or button for joining the community. • If the user is logged in, and is a member of the community, then there should be a link or button for leaving the community. • If the user is a guest, the “join/leave” buttons or links should not appear. <p>Each post in the post listing should look and behave as described in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Along with the view and comment count, there should be a vote count displaying the current vote value for the post with an appropriate label.
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 #	11
Use-Case Name	Search Results Page view
Actors	Guest user
Preconditions	The user has a working internet connection and is viewing any application page.
Postconditions	The user is shown the Search Results Page view containing posts that match the search criteria.
Story	<p>In the search box, the user enters search text and then hits “Enter”.</p> <p>The behavior of the search box and the appearance and behavior of the Search Results page view are the same as for PA03.</p> <p>Each post in the post listing should look and behave as described in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Along with the view and comment count, there should be a vote count displaying the current vote value for the post with an appropriate label.
Exceptions	<p>Show no results found in the Search Results Page view as specified in PA03 in case of search failure.</p> <p>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.</p>

Use-Case #	12
Use-Case Name	Search Results Page view
Actors	Registered user
Preconditions	The user has a working internet connection and is viewing any application page.
Postconditions	The user is shown the Search Results Page view containing posts that match the search criteria.
Story	<p>In the search box, the user enters search text and then hits “Enter”.</p> <p>The behavior of the search box and the appearance and behavior of the Search Results page view are the same as for PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • The post listing should display posts from the user’s communities first, and then posts from communities the user does not belong to after. • Each sublisting should be in the appropriate order (newest, oldest, active). • There should be a clear demarcation between the two sublists, but all posts should still be contained in a single list structure. <p>Each post in the post listing should look and behave as described in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Along with the view and comment count, there should be a vote count displaying the current vote value for the post with an appropriate label.
Exceptions	<p>Show no results found in the Search Results Page view as specified in PA03 in case of search failure.</p> <p>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.</p>

Use-Case #	13
Use-Case Name	Post Page view
Actors	Guest user
Preconditions	The user has a working internet connection and is viewing any application page with links to specific posts.
Postconditions	The user is shown the Post Page view for a specific post.
Story	<p>The user clicks on the link to a specific post and is brought to the Post Page view for that post.</p> <p>The Post Page view should have the same appearance and behavior as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • The comment and reply buttons should be greyed out and non-interactable (or not displayed at all). • If the upvote and downvote elements are present for the post and comments, they should also be greyed-out and non-interactable (but, not displayed at all would be nicer). • The current vote count for the post should appear alongside the view and comment counts, with an appropriate label. <p>The comment listing should appear and behave as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Each comment should display its current vote count with an appropriate label. I will let you choose where it should appear for each comment.
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 #	14
Use-Case Name	Post Page view
Actors	Registered user
Preconditions	The user has a working internet connection and is viewing any application page with links to specific posts.
Postconditions	The user is shown the Post Page view for a specific post.
Story	<p>The user clicks on the link to a specific post and is brought to the Post Page view for that post.</p> <p>The Post Page view should have the same appearance and behavior as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • The post should have upvote and downvote elements. I will let you select their placement. <ul style="list-style-type: none"> ○ Each upvote increases the vote count for the post by 1 and increases the poster's (not the voter's) reputation by 5. ○ Each downvote decreases the vote count for the post by 1 and decreases the poster's reputation by 10. ○ Users with a reputation below 50 cannot vote. ○ Users can only vote on a post once. • The current vote count for the post should appear alongside the view and comment counts, with an appropriate label. <p>The comment listing should appear and behave as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • Each comment should have upvote and downvote elements. I will let you select their placement. <ul style="list-style-type: none"> ○ Each upvote increases the vote count for the comment by 1 and increases the commenter's reputation by 5. ○ Each downvote decreases the vote count by 1 and decreases the commenter's reputation by

	<p>10.</p> <ul style="list-style-type: none"> ○ Users with a reputation below 50 cannot vote. ○ Users can only vote on each comment once. ● Each comment should display its current vote count with an appropriate label. I will let you choose where it should appear for each comment.
Exceptions	<p>The application should show an appropriate error message to the user in case any of the constraints above are violated or 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.</p>

Use-Case #	15
Use-Case Name	New Community Page view
Actors	Registered user
Preconditions	The user has a working internet connection, is logged in, and is viewing any application page.
Postconditions	The new community created by the user is added to the database.
Story	<p>When the user selects the “Create Community” button, they are brought to the New Community Page view.</p> <p>The appearance and behavior of the New Community Page view are the same as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • There is no entry box for the username. The logged in user is implicitly assumed to be the creator of the community and should be recorded in the database as such. • Community names must be unique.
Exceptions	The application should show an appropriate error message to the user in case any of the constraints above are violated or 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 #	16
Use-Case Name	New Post Page view
Actors	Registered user
Preconditions	The user has a working internet connection, is logged in, and is viewing any application page.
Postconditions	The new post created by the user is added to the database.
Story	<p>When the user selects the “Create Post” button, they are brought to the New Post Page view.</p> <p>The appearance and behavior of the New Post Page view are the same as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • The drop-down listing of communities should list the communities the user has joined first. • There is no entry box for the username. The logged in user is implicitly assumed to be the creator of the post and should be recorded in the database as such.
Exceptions	The application should show an appropriate error message to the user in case any of the constraints above are violated or 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 #	17
Use-Case Name	New Comment Page view
Actors	Registered user
Preconditions	The user has a working internet connection, is logged in, and is viewing any application page with comment or reply elements.
Postconditions	The new comment created by the user is added to the database.
Story	<p>When the user selects the “Add a comment” or “Reply” button, they are brought to the New Comment Page view.</p> <p>The appearance and behavior of the New Comment Page view are the same as in PA03, with the following exceptions:</p> <ul style="list-style-type: none"> • There is no entry box for the username. The logged in user is implicitly assumed to be the creator of the community and should be recorded in the database as such.
Exceptions	The application should show an appropriate error message to the user in case any of the constraints above are violated or 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 #	18
Use-Case Name	User Profile Page view
Actors	Registered user
Preconditions	The user has a working internet connection, is logged in, and is viewing an application page.
Postconditions	The user profile of the currently logged in user.
Story	<p>If the user selects the user profile button from the banner, then they are brought to the User Profile Page view.</p> <p>In the main section of the page, the user's display name, email address, a "Member since" timestamp, and the reputation of the user are shown, with appropriate labels.</p> <p>Below this information, is a listing. The listing will contain either the set of communities created by the user, the set of posts created by the user, or the set of comments created by the user. There should be buttons that enable the user to switch between these three listings. The listing should default to listing posts created by the user. Each of the listings is described below.</p> <p>Community Name Listing: A set of community names created by the user are listed. Each community name is a link, which when click shows the <i>New Community Page</i> view, but for editing. The fields should be filled-in with the current state of the community. The user can make changes and then recreate the community saving the changes to the database. This does not count as making a new community. There should also be an option to delete the community. If a community is deleted, all posts that belong to the community and their comments are deleted as well. A dialog should appear asking the user to confirm the choice to delete before executing the</p>

	<p>community deletion.</p> <p>Post Title Listing: A set of post titles created by the user are listed. Each post title is a link which when clicked shows the <i>New Post Page</i> view, but for editing. In this page the user can modify the existing post and post it again or delete it. The form displays the existing information for the post in appropriate fields. Note posting modifications is not considered a new post. Deleting a post will delete all comments associated with it.</p> <p>Comment Listing: Each element should be a post title plus the first 20 characters of the comment left by the user on that post. Each element is a link that opens the New Comment Page view when selected, but for editing. The content field should be filled in. The user can make modifications and resubmit the comment. The user can delete the comment.</p> <p>In all cases “deletion” means that the data is removed from the database. If the user deletes a comment, all replies to the comment are also deleted.</p>
Exceptions	<p>Display an appropriate message if the user has no communities, posts, or comments.</p> <p>Display an error message if a user performs an invalid action as defined in the user story.</p>

Use-Case #	19
Use-Case Name	User Profile Page view
Actors	Admin user
Preconditions	The user has a working internet connection, is viewing any application page, and is logged in as an admin user.
Postconditions	The user profile of an admin user.
Story	<p>If the user selects the user profile button from the banner, then they are brought to the User Profile Page view.</p> <p>In the main section of the page, the user's display name, email address, a "Member since" timestamp, and the reputation of the user are shown, with appropriate labels.</p> <p>Below this information, is a listing. The listing will contain either the set of all users of phreddit, the set of communities created by the user, the set of posts created by the user, or the set of comments created by the user. There should be buttons that enable the user to switch between these four listings. The listing should default to listing all of the users of phreddit. Each of the listings is described below.</p> <p>Phreddit User Account Listing: Each element of the listing should be the display name, email address, and reputation of a user with a phreddit account. Each user element is a link that opens the User Profile Page view for that user. The admin can act as that user by editing or deleting that user's communities, posts, and comments. Some means of navigating back to the admin's own user profile page should be provided.</p> <p>In addition, there should be a delete button next to each user element in the listing. An dialog box should appear</p>

	<p>to allow the admin to confirm their decision to delete a user. Upon confirmation the user, along with all of their communities, posts, and comments will be deleted from the application.</p> <p>Community Name Listing: A set of community names created by the user are listed. Each community name is a link, which when click shows the <i>New Community Page</i> view, but for editing. The fields should be filled-in with the current state of the community. The user can make changes and then recreate the community saving the changes to the database. This does not count as making a new community. There should also be an option to delete the community. If a community is deleted, all posts that belong to the community and their comments are deleted as well. A dialog should appear asking the user to confirm the choice to delete before executing the community deletion.</p> <p>Post Title Listing: A set of post titles created by the user are listed. Each post title is a link which when clicked shows the <i>New Post Page</i> view, but for editing. In this page the user can modify the existing post and post it again or delete it. The form displays the existing information for the post in appropriate fields. Note posting modifications is not considered a new post. Deleting a post will delete all comments associated with it.</p> <p>Comment Listing: Each element should be a post title plus the first 20 characters of the comment left by the user on that post. Each element is a link that opens the New Comment Page view when selected, but for editing. The content field should be filled in. The user can make modifications and resubmit the comment. The user can delete the comment.</p> <p>In all cases deletion means that the data is removed from the database. If the user deletes a comment, all replies to</p>
--	---

	the comment are also deleted.
Exceptions	<p>Display an appropriate message if the system has no users.</p> <p>Display a warning message asking for confirmation to prevent an admin from accidentally deleting a user.</p>

