CSE 360

Team Project Phase 1 Report

Team Project Group 5

Team Member Names:

1. Jad Khayyati
2. Joshua Wright
3. Kyle Ferolito
4. Steven Grisham
5. Jarod Wagner

1. Shared Agreement on the User Experience and Interface

1.1. User Experience Aspects

After deciding the theme, we looked at some different forums for inspiration. We first took a peek at Reddit. Interestingly, Reddit users have two designs to choose from — we chose to look at the more modern one.

- Threads are immediately visible on home page after user login
- Threads are sorted by most recent

1.2. User Interface Aspects

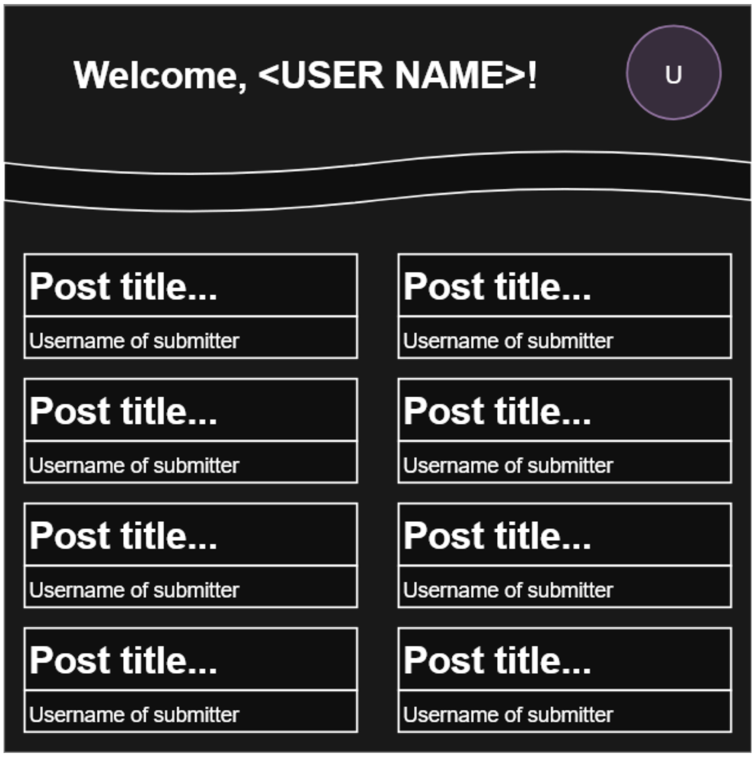
Coming into this project, we had a few things in mind already once the theme was established.

- "Hello, User!" greeting near the top of the page
- List of threads to peruse
- Textbox allowing users to post a thread

Usually, when people come to websites like Reddit, they are looking for information that Google and other search engines struggle to do a good job of finding. That is why we've designed our UX/UI to quickly get users to the list of threads.

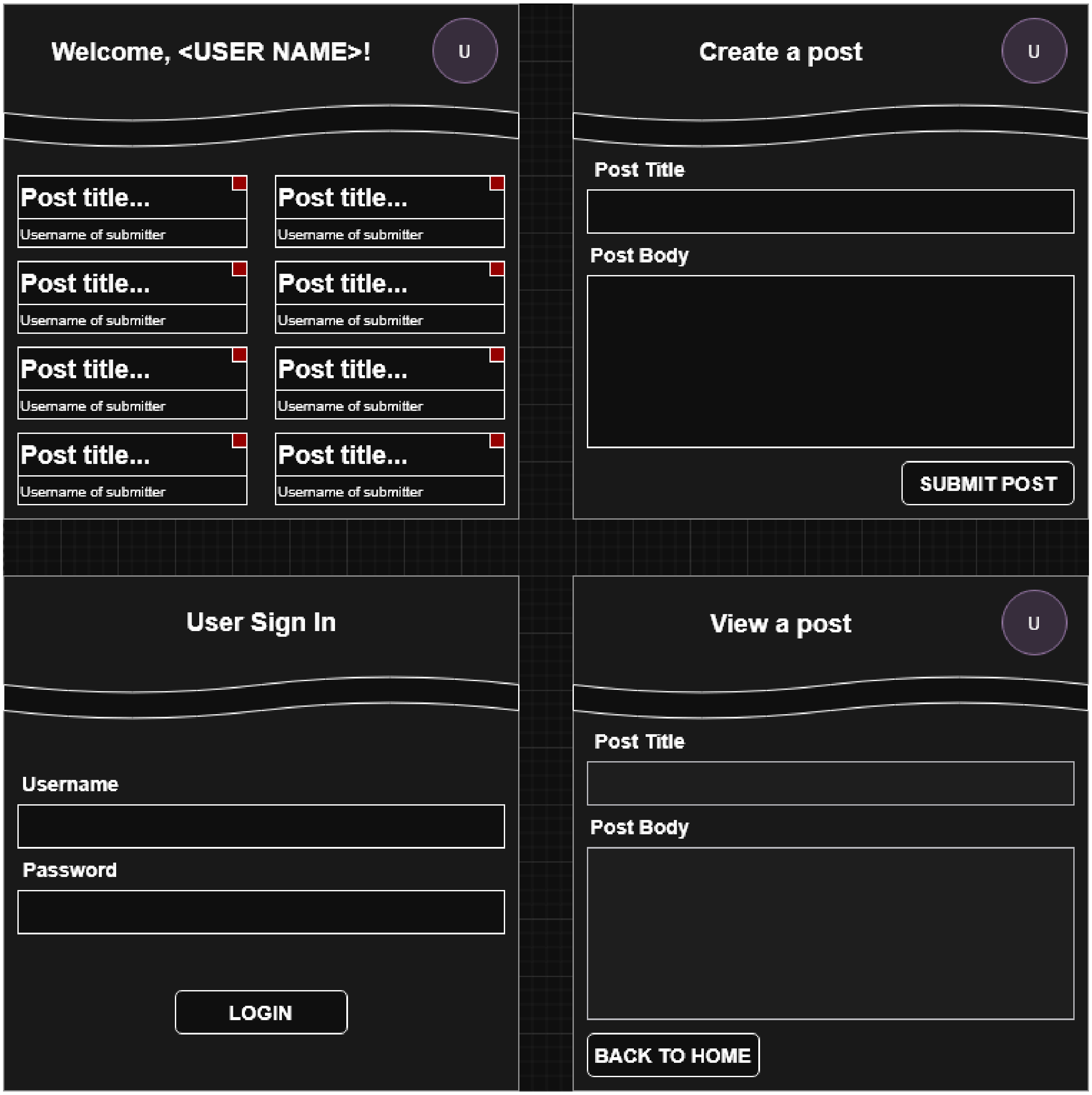
1.3. Concept Art

To the right is a quick mock-up of a home page. This design takes inspiration from Spotify with its simple palette and curved lines, but also borrows from Reddit — mainly in its boxy, modular posts that also behave as buttons.

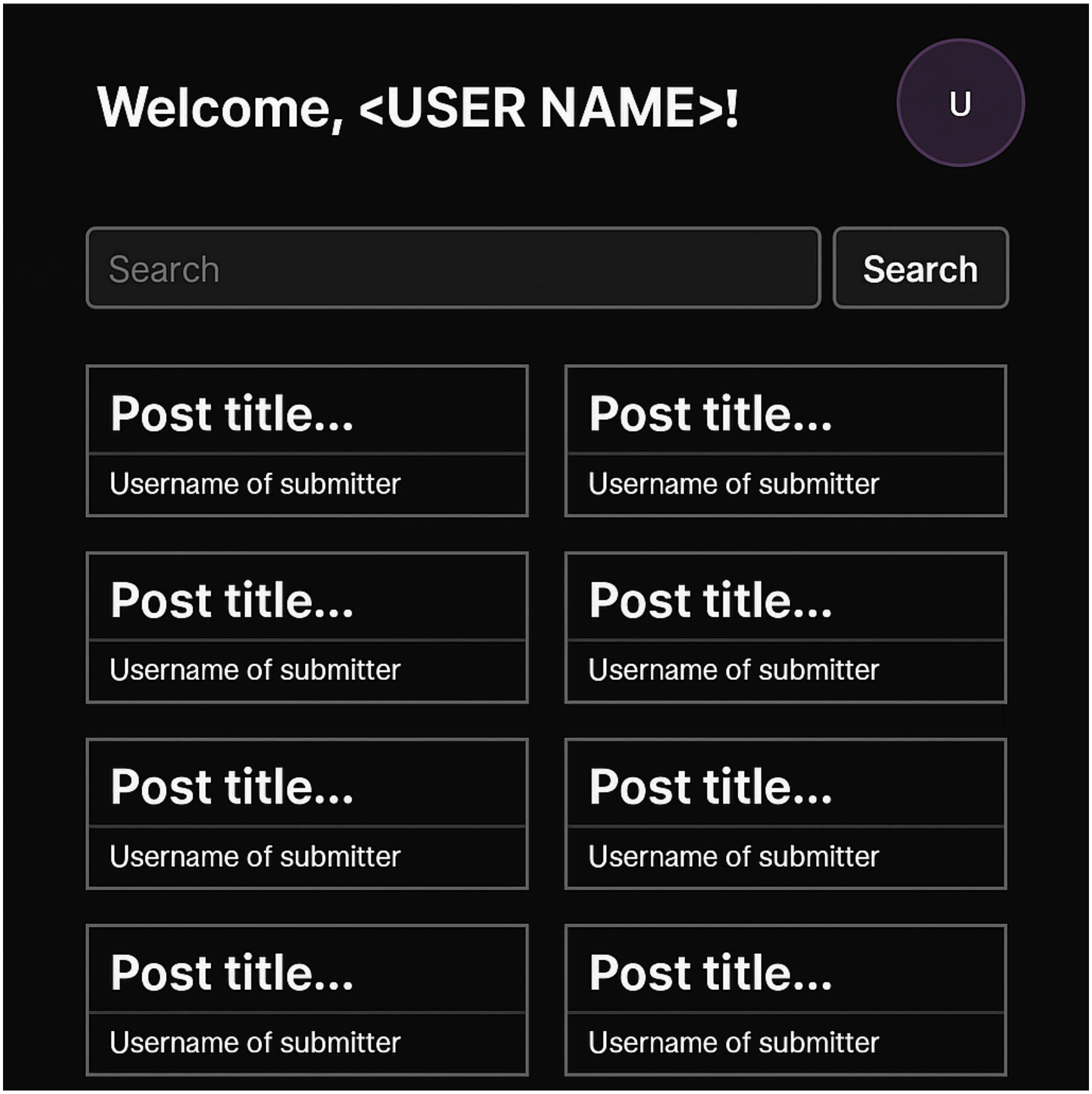


On the following page is a collection of four mockups for application pages. Here, the Reddit influence becomes more apparent.

You may notice red boxes on the home page. These will serve as buttons that allow Admins to delete posts. They will also appear for posts whose submitter's username matches the current user's username, allowing users to delete their own posts.


Welcome, <USER NAME>!
U
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Post title...
Username of submitter
Create a post
U
Post Title
Post Body
SUBMIT POST
User Sign In
Username
Password
LOGIN
View a post
U
Post Title
Post Body
BACK TO HOME

Below is another design featuring a search bar. The search bar sits at the top of the page and is wide and centered for visibility. It has rounded edges for a clean look. The search button appears on the right side and acts as a clickable submit button. A magnifying glass icon would look cool!



2. Input Validation

2.1. Input Fields

The input fields are as follows:

- Username and password for login
- Text for new thread

2.2. Validation Strategy

When prompted to login, users will need to enter information that the program expects. To this end, input validation is necessary.

Input validation will occur at points where users log into the system — mostly within the user registration and user login pages.

The criteria for usernames is as follows:

- No longer than 16 characters

The criteria for passwords is as follows:

- At least 8 characters long
- Contains
  - Uppercase letters
  - Lowercase letters
  - At least one number
  - At least one special character
  - No spaces

2.3. Feedback Messages

Invalid entries will be met with error messages that indicate what the problem is. Such messages will include:

- Username is too long
- Password is too short
- Password does not contain an uppercase letter
- Password does not contain a lowercase letter
- Password does not contain a number
- Password does not contain a special character
- Password does contains a space

3. Implementation Plan and Progress Made

3.1. User Stories:

User Story 1: “As a user, I want to be able to comment on threads so that I can interact with the people on the thread.”

- Jad Khayyati - Will implement the comment input box including the text field and the submit button under each thread.
- Kyle Ferolito- Will implement the layout for displaying comments in a scrollable list.
- Steven Grisham- Will add logic to link comments to their original thread.
- Joshua Wright- Will develop editing or deleting comments.
- Jarod Wagner- Will integrate this entire comment system with the rest of the code.

User Story 2: “As a user, I want to be able to upvote or downvote a comment/post to show my reaction about said post.”

- Jad Khayyati - Will create the upvote and downvote buttons with icons that fit the style of the website/app.
- Kyle Ferolito- Will implement a counter for upvotes and downvotes.
- Steven Grisham- Will add logic to prevent one person from upvoting/downvoting multiple times.
- Jarod Wagner- Will add logic to connect the vote counter to the database.
- Joshua Wright- Will integrate this entire voting system with the rest of the code.

User Story 3: “As a user, I want to have a profile page that shows all my posts.”

- Joshua Wright - Will design the layout for the profile page.
- Kyle Ferolito- Will implement the list view of all posts made by the user.
- Steven Grisham- Will implement the list view of all comments made by the user..
- Jarod Wagner- Will add the ability to customize the user’s profile (username etc.)
- Jad Khayyati- Will integrate this entire profile system with the rest of the code.

3.2. Schedule:

3.2.1. Tuesdays, 3:00 p.m. PST

3.2.2. Thursdays, 3:00 p.m. PST

3.2.3. Saturdays, 3:00 p.m. PST, as needed

3.3. Standup Meetings Notes:

3.3.1. Scrum Meeting 1:

- Chose project theme: a Reddit-style forum app with user and admin roles
- Team norms drafted and nearly complete; each member adds about two norms
- Shared agreement on user experience: scrollable message board, users can post, admins can delete posts or threads
- Building on homework one (foundation code), adding UI and database adjustments if needed

- Input validation will reuse homework one’s work; written section nearly complete
- Screencasts will demonstrate working code and changes made
- GitHub repo set to public for easier sharing
- Deadline confirmed as Sunday -Next meeting scheduled for tomorrow at 3:00 PM PST (edited)

3.3.2. Scrum Meeting 2:

- Confirmed that standup notes are being taken and posted in chat
- Discussed whether team norms should be in GitHub or just the deliverable (decided Appendix B in PDF is sufficient); signatures added as names at bottom
- Appendix A being filled with individual contributions (six tasks per member)
- Missing pieces: automated tests and input validation; JUnit considered but not strictly required, team leaning toward using it
- Screencasts clarified:
  - Technical explanation of code
  - Walkthrough of code execution
  - Planning and division of work (all members must present)
  - Testing demonstration
- Agreed to finish input validation and automated tests by tomorrow or Sunday
- Third screencast (team planning) recorded together during meeting; remaining screencasts assigned to individuals
- GitHub code confirmed as up-to-date; members sharing work via repo and Discord if needed
- Team comfortable finalizing asynchronously in chat after today’s joint recording

4. List of Automated Tests

Test: UserNameRecognizerTest
Screencast: *Input Validation*
Presenters: Kyle Ferolito

1. Empty input field - usernames must not be empty
2. First character validation - usernames must begin with an alphabetical character
3. Special character validation - usernames must not contain consecutive special characters
4. Minimum length validation - usernames must contain at least 4 characters
5. Maximum length validation - usernames must contain at most 16 characters
6. Invalid characters - usernames must contain only alphanumeric characters, hyphens, underscores, and periods

Test: PasswordEvaluatorTest
Screencast: *Input Validation*
Presenters: Kyle Ferolito

1. Empty input field - passwords must not be empty
2. Upper case validation - passwords must contain at least one upper case letter
3. Lower case validation - passwords must contain at least one lower case letter
4. Special character validation - passwords must contain at least one special character
5. Numeric digit validation - passwords must contain at least one numeric digit
6. Length validation - passwords must contain at least eight characters

5. GitHub Repository

5.1. The URL and Password to access the GitHub Repository with a ReadMe file

https://github.com/JoshWright22/CSE360Group5/tree/main

5.2. The Plan for Screencast 1

Once our `UserHomePage.java` file is fully implemented we will:

A. Introduce the file and its overall structure/purpose
B. Point out the main show(Stage primaryStage) method that controls everything on the screen.
C. Breakdown our layout and styling (i.e Explain the VBox layout etc.)
D. Talk about the code for the welcome message the user gets upon entering.
E. Break down the thread container and show it's a scrollable list.
F. Break down the code for the post button and explain how posting creates a new clickable thread and adds it to the top of the list.
G. Walk through the TextArea and the "Post Thread" button
H. Show the code behind threads being clickable and displaying the contents of a thread through an alert.

5.3. The Plan for Screencast 2

Once our `UserHomePage.java` file is fully implemented we will:

A. Run the code and log in as a user (make sure a user account has already been created beforehand)
B. Once in the userhomepage, show the overall layout of the window
C. Start posting threads while showcasing the post button and the text area
D. Add many threads to showcase the scroll functionality (also show that the latest posts will always be at the top of the thread list)

6. Appendix A: Credit Sheet

This appendix lists the members of the team and includes a description of the contribution that the team member has made to this submission. Each team member must provide text for their contribution at a team meeting just prior to the submission of this deliverable and the entire team must agree. If a team member fails to provide this information and/or does not participate in the agreement process to fill out this table, that member will receive no credit for the submission of this deliverable.

Team Member Name	Contributions
Joshua Wright	Scheduled stand–up times Recorded each stand-up Wrote the notes for each stand-up Created GitHub repo Helped create team norms Implemented UserHomePage thread posting and browsing
Kyle Ferolito	Added team norms Described home page and thread page Added username and password evaluation Added test cases for the PasswordEvaluator Added test cases for the UserNameRecognizer Described automated tests
Jad Khayyati	Helped the team setup stand-up times (rescheduling too) Added team norms Worked on screencast 1 (explanation of the code) and wrote the plan Worked on screencast 2 (aimed at potential users) and wrote the plan Helped with example websites for UI Helped with a theme for our UI/UX
Steven Grisham	Created collaborative documents on Google Drive Added team norms Wrote documentation for user interface Wrote documentation for user experience Wrote documentation for input validation Created concept designs for various pages
Jarod Wagner	Added team norms Created design for thread searching Reviewed code for PasswordEvaluator Reviewed code UserNameRecognizer Reviewed test cases for PasswordEvaluator Reviewed test cases for UserNameRecognizer

7. Appendix B: Team Norms

Digital communication at least every other day:

To ensure that team members are communicating, regular communication will be expected.

At least two scrum meetings per week:

To ensure that everyone is on the same page, everyone should have an opportunity to ask questions and review progress. To this end, brief scrum meetings are required at least twice a week, but thrice a week is preferable. Additionally,we should plan meeting times at least 3 days in advance.

Seek peer approval before pushing code to GitHub:

To ensure that the correct implementation of the project is committed, at least two people must approve the code before it is submitted.

Consult with team over client specification disagreements:

If the specifications for the assignment are too vague, they must be approved by the whole team before you can proceed to address them.

Include automated tests in commits:

When submitting code for part of the project, the person submitting the code must make automated tests for that part of the code.

Follow cohesive coding conventions:

Code must be clean and self-documenting. Comments should be used when appropriate to clarify or explain sections of code.

Engage in respectful communication:

Team discussions should remain respectful to ensure a positive work environment.

Regular updates on availability:

Each team member should update the group on their availability for meetings and tasks. If they’re unable to attend a meeting or complete a task on time, they should notify the group in a timely manner.

Sharing knowledge:

Team members should share new techniques or fixes with the group by posting in Discord with a brief explanation and example. Helpful resources should be added to a folder.

Team Signatures: Writing your name confirms you agree with the team norms:

Jad Khayyati
Kyle Ferolito
Steven Grisham
Joshua Wright
Jarod Wagner