Codeversation:

Sprint 1 Planning Document

CS 307 - Software Engineering - Team 23

Alex Aralis, Patricia Chun, Jeremy Lehman, Zach Perry

Table of Contents

Sprint Overview

Summary

For our initial sprint, our goal is to lay a base groundwork for the project and get the project rolling. By the end of the sprint, we should have basic account functionality, displaying posts, and a functional back-end to show at our end-of-sprint demo.

Risks and Challenges

For this sprint, we will be using some technology and frameworks that not all of the teammates are familiar with. It will take time to get these members up to speed before significant progress can be made. Therefore, we will have to take into account that until all members are well versed with our frameworks and languages, some tasks will be slower.

Scrum Master: Zach Perry

Meeting Schedule: Tuesdays and Thursdays at 3:00PM

Sprint Details

User Story #1

As a user, I would like to log in to save my information and post history.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Create database	4 hours	Patricia
2	Create backend server	6 hours	Alex
3	Implement account creation	6 hours	Alex
4	Implement post saving	3 hours	Patricia
5	Implement login screen	6 hours	Jeremy/Zach
6	Implement post history screen	6 hours	Jeremy/Zach
7	Implement backend handling authentication	6 hours	Alex
8	Implement storage of posts on backend	3 hours	Patricia
9	Implement user accounts on backend	4 hours	Alex

Acceptance Criteria:

- Given that the backend to create users is created successfully, when a request to create an account is made then the user's account will be added to the database given that the email has not been taken and the password is strong.
- Given that an account has been successfully created, when the user selects "Create new post" they should be taken to create new post screen
- Given that a user has successfully created an account, when a user attempts to login then they will be successfully authenticated
- Give that user has successfully created an account and are logged in, when they select "post history" then they will be directed to the post history page

User Story #2
As a user, I would like to be able to see the output of my code updated live.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Make post request to compiler microservice.	3 hours	Jeremy/Zach
2	Run compiled JS on client side.	4 hours	Jeremy/Zach
3	Create output component to display results.	2 hours	Jeremy/Zach
4	Detect when output has been invalidated.	4 hours	Jeremy/Zach

Acceptance Criteria:

- Given that microservice has been successfully integrated, when a user compiled run the code, they will be able to see the output if the code runs, or the error messages if the code fails to compile.
- Given that a compiled JS snippet is received from the compiler microservice, when the code does not produce a result within 1s, indicate that the code does not execute and halt the execution thread.

User Story #3

As a user, I would like to post programming-related questions.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Create code snippet form component	6 hours	Jeremy/Zach
2	Create title component	2 hours	Jeremy/Zach
3	Implement snippet storage in database	2 hours	Patricia
4	Implement create new post screen	4 hours	Jeremy/Zach
5	Implement addition of new post to database	2 hours	Patricia
6	Implement create new post on backend	4 hours	Patricia

Acceptance Criteria:

- Given a user has filled in the code snippet form correctly, when they click submit, they will be redirected to the post and it will be submitted properly to the database
- Given a user has incorrectly filled out the snippet form, when they click submit, then an error message will appear with error messages

User Story #4

As a user, I would like to see and make comments about the snippet.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Create comment component	2 hours	Jeremy/Zach
2	Create comment list component	2 hours	Jeremy/Zach
3	Implement comment storage in database	4 hours	Patricia
4	Add comment component to Post component	2 hours	Jeremy/Zach
5	Create comment creation component	2 hours	Jeremy/Zach
6	Implement comment creation in backend	4 hours	Patricia

Acceptance Criteria:

- Given a user fills out the comment form, when they click submit, then comment will appear on the post and be submitted to the database
- Given that the comment functionality has been implemented correctly, when a user is on a post, then user is able to see existing comments about snippet.

User Story #5

As a user, I would like to see nested comments on a snippet.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Implement threaded comments in database	4 hours	Patricia
2	Implement threaded comments in post	2 hours	Jeremy/Zach

Acceptance Criteria:

- Given the that a user comments on a previous comments, a user will see nested comments be able to read the comments organized with code commits.
- Given that user views a post, when there are nested comments, then the comments will be rendered appropriately

User Story #6

As a user, I would like to share my question via a web link.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Make code snippet component.	6 hours	Jeremy/Zach
2	Create Codeversation route.	1 hours	Jeremy/Zach
3	Create Codeversation component.	3 hours	Jeremy/Zach

Acceptance Criteria:

- Given a user has a link to a post, when they go to aforementioned link, they will be directed to the proper post
- Given a user has a link they would like to share, when they share the link, then the receiver of the link will be able to navigate to their post

User Story #7

Create a compiler microservice.

Tasks

Task #	Task Description	Estimated Time	Owner
1	Set up endpoints.	3 hours	Alex
2	Integrate compilers.	6 hours	Alex
3	Deploy server.	2 hours	Alex
4	Create compiler interface.	2 hours	Alex

Acceptance Criteria:

- Given a snippet of code in a language for which compilation is supported, when I send a request to the appropriate endpoint then I will receive a JSON response with executable JS that produces the same output as the original code.
- Given a snippet of code that does not compile, when that code is sent to be compiled send a relevant error message back to the requester.
- Given a snippet of code 3000 characters or less, when I send it to be compiled, I must receive a response within 1s.

Remaining Backlog

Functional Requirements

Profile Features

1. As a user, I would like to have a specific username when I post my question.
2. As a user, I would like to delete my post.
3. As a user, I would like to have onboarding for the app.
4. As a user, I would like to save other user's posts and comments.
5. As a user, I would like to block other users.
6. As a user, I would like to change the theme.

Codeversation Features

7. As a user, I would like to create snippets based on other snippets.
8. As a Codeversation creator, I would like to select any comment or post as a solution.
9. As a user, I would like to have private Codeveration.
10. As a user, I would like there to be specific moderators.
11. As a user, I would like to see things that other users have posted.

Snippet Block Features

12. As a user, I would like to flag comments and posts for inappropriate content.
13. As a user, I would like to tag my post.
14. As a user, I would like to see other posts related to mine.

Front Page Features

15. As a user, I would like a dashboard/front page of popular posts (if time allows).

Server Features

16. As a user, I would like to get email updates. (if time allows)

Anonymous Features

17. As a user, I would like to anonymously browse posts and comments.
18. As a user, I would like to be able to register a Codeversation account.

External Integration Features

1. As a user, I would like to be able to reference threads on Stack Overflow. (if time allows)
2. As a user, I would like to be able to import code from github. (if time allows)
3. As a user, I would like to create an account through github.
4. As a user, I would like to create an account through google.

Non-Functional Requirements

Response Time

1. Must be run on an Isomorphic server (server side rendering).
2. Page must be loaded in 60 seconds, or the page will time-out.
3. As a user, I would like compilation to have low response times.

Scalability

4. Must be deployed on production server.
5. Must be able to service at least 100 simultaneous users.

Usability

6. Must be able to be used on Android. (if time allows)
7. Must be able to be used on iOS. (if time allows)
8. Must be able to be used in a web browser.
9. Must be mobile compatible.

Security

10. User accounts must have a hashed password.
11. Users can view posts and comments, but can not post or comment themselves until they have logged in.
12. New users should have a semi random Captcha to prevent spamming.
13. Users should have a security question and answer to recover their password.