



University Discourse

CS 30700 (Spring 2020)

Team 1

Sprint 1 Planning Document

Drishti Agarwala

Mohana Chandiran Dhukkaram

Tae Yoon Kim

Rewati Shitole

Sprint Overview: During this sprint, we aim to build a strong foundation for our project and start with the necessary stepping stone functionalities such as frontend and backend for authentication/registration, setting up the database and displaying several dummy posts on the main page. While we work on providing the above-mentioned functionality, we would also develop effective test cases to test our features for correctness.

Scrum Master: Tae Yoon Kim

Meeting Schedule : Mondays @ 12:30 - 1:30 / Thursdays @ 12:00 - 1:30

Risks and Challenges: The biggest challenge for us is learning about the new frameworks and tools such as React, Node.js, and MongoDB and implementing them in our work. In addition, it will initially be challenging to work as a team because coordination and effective communication among team members will take time to sync in. Since this is also our first time working on developing test

cases, it will be challenging to decide on the most effective framework to serve the purpose.

User Story #1: As a user, I want to be able to create an account

#	Description	E.T.	Owner
1	Create React components for the UI for registering account	2 hrs	Rewati
2	Storing user credentials on the database after hashing it	2 hrs	Rewati
3	Create method to handle request for creating a new account (Backend)	2 hrs	Rewati
4	Create method to send request to server and pass the user credentials	2 hrs	Rewati
5	Create method to display server response to UI	1 hr	Rewati
6	Debug and test methods using unit tests	3 hrs	Rewati

Acceptance criteria:

- Given that the methods which query the database for storing user credentials are implemented correctly, when the user creates an account, then the hashed user credentials should be stored in the database.
- Given that the user has correctly entered the required credentials, and the database has successfully saved the user credentials, when the user attempts to log in, then the UI should change appropriately to display the feed page.
- Given that the method to validate the email format is working correctly, when the user does not enter the email in the correct format, then the UI should change to display an appropriate error message.
- Given that the method to query the database for checking if a similar email id already exists in it is working correctly, when the user enters a new email id which matches the email id associated with another account, then an appropriate error message should be displayed.
- Given that the method to validate the password format is working correctly, when the user does not enter a valid password, then the UI should change to display an appropriate error message.

User Story #2: As a user, I want to be able to log in to my account

#	Description	E.T.	Owner
---	-------------	------	-------

1	Create React components for the UI for logging in	2 hrs	Rewati
2	Create method to verify credentials with the database	1 hr	Rewati
3	Create method to handle login request (Backend)	2 hrs	Rewati
4	Create method that sends user credential to the server	2 hrs	Rewati
5	Create method to store user credentials in cache	2 hrs	Rewati
6	Create method to display server response to UI	1 hr	Rewati
7	Debug and test methods using unit tests	3 hrs	Rewati

Acceptance criteria:

- Given that the methods to save hashed user credentials to cache are working correctly, when the user successfully logs in, then the credential should be stored in cache for auto login.
- Given that the methods to retrieve from cache are working properly, when the user comes back to the application, then the user should be automatically logged in.
- Given that the authentication methods are working correctly, when the user has entered correct credentials, then the UI should change appropriately to display the feed page.
- Given that the authentication methods are working correctly, when the user clicks the login button, then the credentials should be hashed and sent to the server as a request to check for validity by comparing with the database.
- Given that the authentication methods are working correctly, when the user did not enter correct credentials, then the UI should change appropriately to display the error message and ask the user to enter the credentials again.

User Story #3: As a user, I want to be able to change my email

#	Description	E.T.	Owner
1	Create React components for the UI for changing email	2 hrs	Rewati
2	Store mock data on the database to verify that email was changed	1 hr	Rewati

3	Create method to handle email change request	2 hrs	Rewati
4	Create method that sends request to server with user credential	2 hrs	Rewati
5	Create method to display server response to UI	1 hr	Rewati
6	Debug and test methods using unit tests	3 hrs	Rewati

Acceptance criteria:

- Given that the methods are working correctly, when the user selects the option to change their email, then the correct UI should be displayed.
- Given that the method to query the database for checking if a similar email id already exists in it is working correctly, when the user enters a new email id which matches the email id associated with another account, then an appropriate error message should be displayed.
- Given that the method to compare the old and new email id is working correctly, when the user enters a new email id which matches the old email id associated with the account, then an appropriate error message should be displayed.
- Given that the method to validate the email format is working correctly, when the user does not enter the email in the correct format, then the UI should change to display an appropriate error message.
- Given that the method to query the database for storing the update data is working correctly, when the user changes his email id, then the updated information should be saved in the database.

User Story #4: As a user, I want to be able to change my password

#	Description	E.T.	Owner
1	Create React components for the UI for changing password	2 hrs	Chandiran
2	Store mock data on the database to verify that hashed password was changed	2 hrs	Chandiran
3	Create method that verifies old credentials with the database	2 hrs	Chandiran
4	Create method to handle request to change password (Backend)	2 hrs	Chandiran
5	Create method to send request to server along with user credentials	2 hrs	Chandiran

6	Create method to display server response to UI	1 hr	Chandiran
7	Debug and test methods using unit tests	3 hrs	Chandiran

Acceptance criteria:

- Given that the UI methods are working correctly, when the user selects the option to change their password, then the correct UI should be displayed.
- Given that the method to compare the old and new password is working correctly, when the user enters a new password which matches the old password associated with the account, then an appropriate error message should be displayed.
- Given that the method to validate the password format is working correctly, when the user does not enter a valid password, then the UI should change to display an appropriate error message.
- Given that the method to hash and send the password to the server is working correctly, when the user submits the new password, then the password should be hashed and sent to the server as a request.
- Given that the method to query the database for storing the update data is working correctly, when the user changes his password, then the updated information should be saved in the database.

User Story #5: As a user, I want to be able to recover my password

#	Description	E.T.	Owner
1	Create React components for the UI for recovering password	2 hrs	Chandiran
2	Store mock data on the database that will be changed to reflect password reset	2 hrs	Chandiran
3	Create method to handle request to recover password (Backend)	2 hrs	Chandiran
4	Create method to send request to server based on user credential (Frontend, cache)	2 hrs	Chandiran
5	Create a method that sends an email to the user with a link to reset the password	2 hrs	Chandiran
6	Create method to display server response to UI	1 hr	Chandiran
7	Debug and test methods using unit tests	3 hrs	Chandiran

Acceptance criteria:

- Given that the UI methods are working correctly, when the user selects the option to recover their password, then the correct UI should be displayed.
- Given that the method to check the validity of the email entered by the user is working correctly, when the user enters the email id, then the client should send a request to the server to send an email to the given email id with a password reset link.
- Given that the method to hash and send the password to the server is working correctly, when the user submits the new password, then the password should be hashed and sent to the server as a request.
- Given that the method to query the database for storing the update data is working correctly, when the user changes his password, then the updated information should be saved in the database.

User Story #6: As a user, I want to be able to log out of my account

#	Description	E.T.	Owner
1	Create React components for the UI for the button to logout	1 hr	Chandiran
2	Create method to handle logout request (Backend)	2 hrs	Chandiran
3	Create method that sends a request to the server with the user credentials (Frontend, cache)	2 hrs	Chandiran
4	Create method to manage cache data after logging out of account	1 hr	Chandiran
5	Create method to display server response to UI	1 hr	Chandiran
6	Debug and test methods using unit tests	3 hrs	Chandiran

Acceptance criteria:

- Given that the UI methods are working correctly, when the user selects the option to logout, then the correct UI should be displayed.
- Given that the logout handling methods are working correctly, when the user logs out, then the event should be logged in the server.
- Given that the request handling methods are working correctly, when the user logs out, then the server should be able to log it.
- Given that the request handling methods are working correctly, when the user logs out, then the server should be able to give a valid response to the client on when to change the UI.

User Story #7: As a user, I want to be able to delete my account

#	Description	E.T.	Owner
1	Create React components for the UI for deleting the account	2 hrs	Drishti
2	Store mock data on the database that should be removed when the user deletes the account	1 hr	Drishti
3	Create method to handle request to delete account (Backend)	2 hrs	Drishti
4	Create method that send request to server based on user credential (Frontend, cache)	2 hrs	Drishti
5	Create method to manage cache data after removal of account	2 hrs	Drishti
6	Create method to display server response to UI	1 hr	Drishti
7	Debug and test methods using unit tests	3 hrs	Drishti

Acceptance criteria:

- Given that the UI methods are working correctly, when the user selects the option to delete their account, then the correct UI for deleting one's account should be displayed.
- Given that the method to query the database for deleting data is working correctly, when the user deletes his account, then all the information associated with the user's account stored in the database should be deleted.
- Given that the method to manage the cache on the front end is working properly, when the user finishes deleting his/her account, then the cache should clear itself.
- Given that the method to query the database to get every post that the user has made is working properly, when the user deletes his/her account, then every post made by the user should be removed.
- Given that the method to query the database to get every post that the user has made is working properly, when the user deletes his/her account, then every comment made by the user should be removed along with the whole thread.

User Story #8: As a user, I want to be able to view posts on the feed based on the topics I follow

#	Description	E.T.	Owner
1	Create React components for the UI that retrieves the posts from the database	2 hrs	Drishti
2	Store mock data on the database that will used to verify the posts that the user follows	1 hr	Drishti
3	Create method to handle request for fetching posts related to the tags that the user follows	2 hrs	Drishti
4	Create method to send request for fetching posts from the database to server with user credential	2 hrs	Drishti
5	Create method to display server response to UI	2 hrs	Drishti
6	Debug and test methods using unit tests	4 hrs	Drishti

Acceptance criteria:

- Given that the methods that lead us to the homepage after logging in are implemented correctly when the user logs in and reaches the main page, then the user should see a list of posts.
- Given that the methods that let us view the feed are implemented correctly, when there are multiple posts, then the user should be able to scroll down.
- Given that the internet connection is moderate, when the user logs in and reaches the main page, then the user should see a list of posts under 1 second.
- Given that the methods that retrieve data from the database are implemented correctly, when the user reaches the main page, then the user should see posts that are from the tags that the user follows(currently mock data)
- Given that the methods that handle requests from the clients are implemented correctly, when the request is sent from the client, then the server should log the request and make a request to the database for the user's following tags and posts.
- Given that the methods that retrieve data from the database based on the tags are implemented correctly, when the user views their feed, then the server should log what has been fetched from the database. It should also verify that the posts being sent as a response to the client should match the tags that the user is following.

User Story #9: As a user, I want to be able to select between the various options of the post when creating a new post

#	Description	E.T.	Owner
1	Create React components for the UI for selecting the type of post	2 hrs	Drishti
2	Create a method to create request for the server on creating which type of post	2 hrs	Drishti
3	Create a method to send a post creation request to the server with user credential	2 hrs	Drishti
4	Create a method to display server response to UI	2 hrs	Drishti
5	Debug and test methods using unit tests	4 hrs	Drishti

Acceptance criteria:

- Given that the correct homepage was displayed after login, when the user clicks on the create post button on the homepage, then it should lead the user to the page where he/she can choose what kind of post to make.
- Given that the methods that lead us to specific pages on button clicks are implemented correctly, when the user selects an option, then the UI should be updated accordingly.
- Given that the correct UI is displayed, when the user fills out the posting form and submits, then it should display an error message if the contents are incorrect.
- Given that the methods that make the request to the server are implemented correctly, when the user submits, then the correct form of request should be made to the server.
- Given that the methods that handle query commands are implemented correctly, when the user submits, then the correct form of query command should be made from the server to the database.

User Story #10: As a user, I want to be able to post images

#	Description	E.T.	Owner
1	Create React components for the UI for posting images	2 hrs	Tae Yoon
2	Create method to handle request for posting image posts to the database	2 hrs	Tae Yoon
3	Create method to send post creation request to the server with user credential and image URL	2 hrs	Tae Yoon
4	Create UI/method to display server response to UI	2 hrs	Tae Yoon

5	Debug and test methods using unit tests	4 hrs	Tae Yoon
---	---	-------	----------

Acceptance criteria:

- Given that the appropriate UI is displayed, when the user selects one of the create post options, then the user should be able to choose the image option and select an image for the post.
- Given that the methods that check for the correctness of the fields are implemented correctly, when the user does not fill in the appropriate fields for the post, then the UI should display an error.
- Given that the request handler methods are implemented correctly, when the user presses submit, then the correct form of request should be made to the server.
- Given that the methods that send the request to the server in the case of a correct form are implemented correctly, when the user clicks submit, then an appropriate request to the server should be made.
- Given that the methods that add posts to the database under the correct user entry are implemented correctly, when the user clicks submit, then the server should generate the appropriate query command to push the data to the database.
- Given that the request handler methods on the server are implemented correctly, when the post has actually been created in the database, then the server should respond to the client with success and the client should notify the user.
- Given that the correct data was pushed to the database, when the user loads their feed, then the post should be there on their feed if they follow those tags.

User Story #11: As a user, I want to be able to post text

#	Description	E.T.	Owner
1	Create React components for the UI for posting text	2 hrs	Tae Yoon
2	Create method to send request to server with user credential and text input	2 hrs	Tae Yoon
3	Create method to handle post creation request from the client	2 hrs	Tae Yoon
4	Create method for the server to request the database for storing new text post	2 hrs	Tae Yoon

5	Create UI/method to display server response to UI	2 hrs	Tae Yoon
6	Debug and test methods using unit tests	4 hrs	Tae Yoon

Acceptance criteria:

- Given that the appropriate UI is displayed, when the user selects one of the create post options, then the user should be able to choose the text option and input text for the post.
- Given that the methods that check for the correctness of the fields are implemented correctly, when the user does not fill in the appropriate fields for the post, then the UI should display an error.
- Given that the methods that send the request to the server in the case of a correct form are implemented correctly, when the user clicks submit, then an appropriate request to the server should be made.
- Given that the methods that add posts to the database under the correct user entry are implemented correctly, when the user submits the post, then the server should generate the appropriate query command to push the data to the database.
- Given that the methods on the server are implemented correctly, when the post has actually been created in the database, then the server should respond to the client with success and the client should notify the user
- Given that the correct data was pushed to the database, when the user loads their feed, then the post should be there on their feed if they follow those tags.

User Story #12: As a user, I want to be able to edit a post

#	Description	E.T.	Owner
1	Create React components for the UI for editing a post	2 hrs	Tae Yoon
2	Create a method on the server for updating the content of the post in the database	1 hr	Tae Yoon
3	Create method to handle post update request received from the client	2 hrs	Tae Yoon
4	Create client-side method that sends request to server with user credential and updated text	2 hrs	Tae Yoon
5	Create UI/method to display server response to UI	2 hrs	Tae Yoon
6	Debug and test methods using unit tests	4 hrs	Tae Yoon

Acceptance criteria:

- Given that the appropriate UI is displayed, when the user clicks on edit-post button, then the user should be able to see a screen similar to create post.
- Given that the methods that check for the correctness of the fields are implemented correctly, when the user does not fill in the appropriate fields for the post, then the UI should display an error.
- Given that the methods that send the request to the server in the case of a correct form are implemented correctly, when the user submits the form, then the correct request should be made to the server.
- Given that the methods that send the query command to the database are implemented correctly, when the user submits the form, then the correct type of query command should be made from the server to the database.
- Given that the methods that reload data from the database to the feed are implemented correctly, when the post has been successfully updated in the database, then the user should be able to see the updated post and not the old post

Product Backlog

Functional Requirements:

Serial No.	Backlog Item
1	As a user, I want to be able to create an account
2	As a registered user, I want to be able to login to my account
3	As a registered user, I want to be able to change my email
4	As a registered user, I want to be able to change my password
5	As a registered user, I want to be able to recover my password
6	As a registered user, I want to be able to logout of my account
7	As a registered user, I want to be able to delete my account
8	As a moderator, I want to be able to edit the topic associated with the post
9	As a user, I want to be able to report inappropriate content
10	As an unregistered/registered user, I want to be able to search for topics
11	As an unregistered/registered user, I want to be able to filter searches based on university, subject and courses
12	As an unregistered/registered user, I want to be able to view all information which includes posts, comments, course ratings, curve estimations
13	As an unregistered/registered user, I want to be able to view a graphical representation of the estimated curve
14	As an unregistered/registered user, I want to be able to view the plan of studies on the calendar interactively / semester schedules
15	As a registered user, I want to be able to follow topics
16	As a registered user, I want to be able to view posts on the feed based on the topics I follow
17	As a user, I want to be able to select between the various options of the post when creating a new post
18	As a registered user, I want to be able to post images
19	As a registered user, I want to be able to post text

20	As a registered user, I want to be able to post schedule in a calendar asking for feedback
21	As a registered user, I want to be able to comment on posts
22	As a registered user, I want to be able to like the posts
23	As a registered user, I want to be able to share the posts as embeddable links
24	As a registered user, I want to be able to upvote/downvote the posts
25	As a registered user, I want to be able to upvote/downvote the comments
26	As a registered user, I want to be able to upload documents relating to a course
27	As a registered user, I want to be able to input my scores and past semester curves for a course
28	As a registered user, I want to be able to rate the course based on difficulty and professor
29	As a registered user, I want to be able to select a tag for the course (Awesome, good, bad)
30	As a user, I want to be able to have two-factor authentication (*)
31	As a registered user, I want to be able to comment on a comment
32	As a registered user, I want to be able to edit a post
33	As a registered user, I want to be able to edit a comment
34	If time allows, as a moderator, I want to be able to delete inappropriate content
35	If time allows, as a first time user, I want to be able to view top posts (trending posts) in my initial feed.
36	If time allows, as an unregistered/registered user I want to be able to view recommended posts in my feed/explore feature based on my previous activity

Non Functional Requirements :

- (1) Must be able to use this application on desktop browsers
- (2) If time allows, users must be able to use this application on mobile web browsers
- (3) The UI must be easy to understand and navigate for the users
- (4) The database must be easy to maintain and efficient to extract, update or delete data from
- (5) Our app must be able to host enough simultaneous users