

Sprint 3 Planning Document

March 29, 2018

Team: 29

Anjali Malik, Shengqi Wang, Kenan Dominic, Arnav Jain, Sanjit Sama

Sprint Overview

Scrum Master: Anjali Malik

Team Meeting Details: Monday, Wednesday, Friday @ 11:30 pm in HSSE 145

OVERVIEW:

Final sprint will be a lot of work for our team. We have many little things that we would like to perfect from previous sprints. In addition, to optimizing what has already been built we must work on two major aspects of our service. We must first create and finalize what we would like our profile page to include. This includes implementing a review system. The second complex addition would be to create a search algorithm to efficiently sort and organize all listed cards.

RISKS & CHALLENGES:

The obvious challenges include implementing a search algorithm alongside the profile page. However, we face many risks this sprint since this our final sprint. Essentially, a large majority of our time must be dedicated in ensuring that everything we have created has no bugs and everything is to the expectations to that of the vision we have initially planned. The issue with bugs as well as bug fixes are that we never know when they may arise and the time required to fix them.

User Story #1: I would like to be able to search for posts using keywords.

Tasks	Description	Time	Owner(s)
1	Add a UI element in the menu for a search bar so that a user can type in any keyword and click on search to send their search query to the backend	1 hour	Arnav/Sanjit
2	Implement code in JS that accepts the user input and send the request to the server	1 hour	Shengqi
3	Implement an algorithm and an endpoint in the server that uses the keyword to search for the relevant post(s) and send the posts back to the client.	3 hours	Anjali/Kenan
4	Write code in JS that can receive information from the server and output it to the UI.	1 hour	Anjali/Shengqi

Acceptance Criteria:

- Given that the functionality is implemented, if the user wants to search for a post using a keyword, they should be able to input that keyword in the correct search bar.
- Given that the user has sent the request to the backend for searching posts, the algorithm should search for the keyword in the title and/or the description of the post (dependent upon our final design approach).
- Given that the user has successfully sent the request to the backend for searching posts, the algorithm should return with relevant posts and the UI should display the relevant posts.

User Story #2: I would like to be able to view other users' profiles.

Tasks	Description	Time	Owner(s)
1	Implement Server logic to return the link to the user profile corresponding to each clicked user name	2 hours	Kenan/Anjali
2	Implement JS logic to retrieve relevant user profile information upon user request of viewing that profile	2 hours	Shengqi/Anjali
3	Implement UI element to make each user name on a post such that it points to the user's profile	2 hours	Sanjit/Arnav
4	Test using Postman the endpoint as well as manually test the user story by making multiple user profiles and checking whether the profiles of the users are being displayed on directed pages.	1 hour	Anjali

Acceptance Criteria:

- Given that the feature is implemented, user name on posts/reviews should be linked to the corresponding profile pages.
- Given that the user clicks a username on a post/review, they should be redirected to the corresponding user's profile.
- Given that the user selects to visit a user profile and are successfully directed to the relevant page, they should be able to view all correct profile information on the user's page.

User Story #3: I would like to be able to look at a list of reviews on my profile.

Tasks	Description	Time	Owner(s)
1	Implement UI elements for the reviews.	4 Hours	Sanjit
2	Implement JS logic to fetch all reviews from the database	3 Hours	Shengqi/Sanjit
3	Implement JS logic to populate reviews to the UI	3 Hours	Shengqi/Sanjit
4	Create a database table for reviews and implement server code to access it	4 Hours	Anjali
5	Implement logic in server to add a new review to a user profile	2 hours	Anjali
6	Implement server endpoint to retrieve list of reviews for a user	2 hours	Anjali/Kenan
7	Implement UI element to add a review	1 hour	Sanjit/Arnav
8	Implement JS code to accept user input and communicate with server to send the request to database change.	2 hours	Anjali
9	Test the endpoint using different body inputs, and manually test through the client to making multiple reviews and viewing them on multiple profiles.	1 hour	Kenan

Acceptance Criteria:

- Given that the user is on a profile page, the JS code should fetch all necessary information including the reviews for that user from the database
- Given that the JS has pulled all information from the database, it should populate the UI elements with all the reviews from the user
- Given that the feature works, each review should show relevant information about itself in the UI card.
- Given that the database table is made, all user reviews should be saved and linked to the user

User Story #4: I would like to be able to see the average rating given to me.

Tasks	Description	Time	Owner(s)
1	Implement UI element to show average rating.	2 Hours	Sanjit
2	Implement logic to calculate average rating.	2 Hours	Shengqi
4	Implement JS code to send the rating to the webpage	1 Hour	Arnav
5	Implement server logic to retrieve ratings for a user.	2 hours	Anjali
6	Test endpoint using Postman, and manually test through client by giving variable ratings and checking if the average ratings as displayed are correct.	1 hour	Anjali

Acceptance Criteria:

- Given that backend is correctly implemented, the server should use a formula to calculate the new average rating and send it back to the client.
- Given the client receives the response from the server, the client should receive the average rating and display any necessary updates to it.
- Given that the user visits a profile, and the user has no reviews/ratings, an appropriate message should be written in the place of average rating.

User Story #5: I would like to be able to view rated stars by each user who has written a review on another user profile so that I am easily able to navigate between a low and a high rating review.

Tasks	Description	Time	Owner(s)
1	Add JS functionality to include UI element for ratings given by a user along with a review	2 hours	Kenan
2	Add UI in review cards to include the stars	1 hour	Sanjit/Arnav
3	Implement in JS code to retrieve stars for each review from the server to include in the cards	2 hours	Shengqi/Anjali
4	Add server endpoint to serve the request for number of stars for any review	2 hours	Anjali/Kenan

Acceptance Criteria:

- Given the reviews are added by any user, they must be displayed along with stars given by the user.
- Given the feature is correctly implemented, the user should be able to see the names along with rating given by them on each review card.
- Given the feature is correctly implemented, the user should be able to visually see the stars for a corresponding rating, and differentiate between a low and a high rating.

User Story #6: I would like to sort my reviews in ascending or descending order of ratings/stars.

Tasks	Description	Time	Owner(s)
1	Add UI element in frontend to select the order in which the reviews should be listed.	1 hour	Sanjit/Arnav
2	Implement JS code which receives user input and communicates the request with the server for relevant list of reviews back.	2 hours	Shengqi
3	Implement an endpoint in the server to sort the reviews for the user by the specified order of ratings.	3 hours	Anjali/Kenan
4	Test the endpoint by giving both orders, and manually test through the client by making variable ratings on a review list and sorting the list in both orders.	1 hour	Kenan

Acceptance Criteria:

- Given that the frontend is developed for this feature, user should be given options to select from for ordering based on rating of the list of reviews.
- Given that the user selects an order for sorting the reviews, the backend should receive the request and give an ordered list of reviews as result of the user interaction.
- Given that the user selects an order for sorting of reviews, the page should be promptly change to reflect their selection.

User Story #7: I would like to sort my reviews in ascending or descending order of date posted.

Tasks	Description	Time	Owner(s)
1	Add UI element in frontend to select the order in which the reviews should be listed.	1 hour	Sanjit/Arnav
2	Implement JS code which receives user input and communicates the request with the server for relevant list of reviews back.	2 hours	Shengqi
3	Implement an endpoint in the server to sort the reviews for the user by the specified order of date posted.	3 hours	Anjali/Kenan
4	Test the endpoint using Postman, and manually test through the client and sorting the list in both orders.	1 hour	Kenan

Acceptance Criteria:

- Given that the frontend is developed for this feature, user should be given options to select from for ordering based on date posted of the list of reviews.
- Given that the user selects an order for sorting the reviews, the backend should receive the request and give an ordered list of reviews as result of the user interaction.
- Given that the user selects an order for sorting of reviews, the page should be promptly change to reflect their selection.

User Story #8: I would like to be able to write a review on a user's profile

Tasks	Description	Time	Owner(s)
1	Add UI elements in frontend to give rating as well as write review for a user	1 hour	Sanjit/Arnav
2	Implement JS code which accepts the input and sends it to the server	2 hours	Shengqi
3	Implement an endpoint in the server to add the review.	3 hours	Anjali/Kenan
4	Test the endpoint using Postman, and manually test through the client by adding multiple posts on multiple user profiles.	1 hour	Kenan

Acceptance Criteria:

- Given that the frontend is developed for this feature, user should be able to select a rating and write a review for any user.
- Given that the feature is correctly implemented, the user should not be able to write reviews multiple times for the same user.
- Given that feature is implemented, when the user posts a review, the database should store it correctly using the server.
- Given that feature is implemented, when the user posts a review, the webpage should be able to reflect that change.

User Story #9: I would like to be able to search for a user using their name.

Tasks	Description	Time	Owner(s)
1	Add a UI element for search bar in our pages so that a user can type in a user's name and click on search to send their search query to the backend	1 hour	Arnav/Sanjit
2	Implement code in JS that accepts the user input and send the request to the server	1 hour	Shengqi
3	Implement an endpoint in the server that uses user name to search for the relevant user(s) and send the profiles back to the client.	3 hours	Anjali/Kenan
4	Write code in JS that can receive information from the server and output it to the UI while giving the user options to choose from.	1 hour	Anjali/Shengqi
5	Implement code such that the user will be able to click on any of the results/options and be directed to the relevant user profile	1 hour	Kenan/Shengqi
6	Test the endpoints using Postman and manually test the user story by making multiple profiles and making multiple searches for different inputs.	1 hour	Kenan

Acceptance Criteria:

- Given that the functionality is implemented, if the user wants to search for a user using their name, they should be able to input that name in the correct search bar.
- Given that the user has sent the request to the backend for searching profiles, the algorithm should return back with option(s) for the user to choose from.
- Given that the user has sent the request to the backend for searching profiles, and there doesn't exist a matching profile, the UI should display an appropriate response message to the user.
- Given that the user has received options for their search of profiles, if a user clicks on any of the resulted options, they should be appropriately directed to the relevant profile page of the other user.

User Story #10: I would like a more robust notification system incorporating a separate notifications page as well as a real time count of unseen notifications.

Tasks	Description	Time	Owner(s)
1	Implement code in frontend to add an element for pointing to new notifications.	1 hour	Sanjit/Arnav
2	Implement in server an endpoint which will generate unseen notifications by the user	2 hour	Anjali/Kenan
3	Write code in JS to show the user new notifications, and removing the new notifications UI element once the user has seen the new notifications.	2 hour	Kenan/Shengqi
4	Implement code in the application which will either let the user to visit a new page which will contain all old notifications or automatically remove old seen notifications from the table.	2 hour	Shengqi/Anjali
5	Test manually by making multiple comments or likes and logging in from another account to see new notifications. Test by also viewing the notifications to check if they are successfully tagged as 'seen'.	1 hour	Arnav

Acceptance Criteria:

- Given that the functionality is implemented in the code, when the user logs in, they should be able to see their outstanding number of unseen notifications.
- Given that the user has seen the new notifications, the pointer with number of new notifications should disappear and only appear again when a new notification is created.
- Given that the user has seen the new notifications and they click on the option to see notifications again, they should see a message informing them that there are no new notifications.
- Given that the user is clicks to see their notifications, they should have an option to see their past notifications in a new page, or their old notifications should be completely removed from the system (depending on the design approach employed for the user story).

User Story #11: I would like to be automatically logged out after 30 mins (variable) of inactivity on Lance.

Tasks	Description	Time	Owner(s)
1	Re-implement a more powerful authentication system which will keep a check on user activity on the application.	4 Hours	Anjali/Kenan
2	Implement code in JS that will automatically log the user out if their inactivity time period is passed.	3 Hours	Shengqi/Anjali
4	Write code to reset the user authentication token when they are logged out, and send user to a page with displayed reason for sign out.	1 hour	Anjali/Kenan
5	Test the authentication function manually through client by performing variable activities on the client	1 hour	Sanjit

Acceptance Criteria:

- Given that the server and backend code is working, the user should not be able to stay logged in after 30 mins of inactivity.
- Given that the functionality is correctly implemented, and the user's activity session is expired, all of their attempts to send requests or interact with the web page should fail.
- Given that the functionality is correctly implemented, the user should be redirected to a new page with reason of their sign out from the application.
- Given that a user is signed out for inactivity, and they have been directed to the 'session timeout' page, the page should provide the user with link to the login page.

User Story #12: I would like to reset my password by clicking on a 'forgot my password' option.

Tasks	Description	Time	Owner(s)
1	Create a UI element which will allow for a forgot password button in the "Log In" modal.	1 Hour	Sanjit/Arnav
2	Create a modal which will display questions necessary to collect from user for the "Log In" modal.	1 Hour	Sanjit/Arnav
3	Implement an endpoint in the server to reset password of the user	3 Hours	Kenan/Anjali
4	Implement JS code to ask necessary questions to the user or send unique code to the user for verification (depending upon what design approach is decided).	2 hours	Shengqi/Anjali
5	Test by having multiple users clicking forgot password and ensure that they are able to log in with their new password and are unable to do so with their previous password.	1 Hour	Arnav

Acceptance Criteria:

- Given that the user clicks on "Forgot Password" he or she shall be approached by a modal asking further security questions or details about their account on Lance (depending upon what design approach is decided for completing this user story).
- Given that the user changes his or her password they shall not be able to log in with their previous password.
- Given that the user changes his or her password they shall be able to log in with their newly set password.
- Given that the user story is correctly implemented and suitable security measures are taken, the feature must ensure that only the correct user is able to reset their password.

User Story #13: I would like to delete my account from Lance.

Tasks	Description	Time	Owner(s)
1	Create a UI Element under settings for “Removing Lance Account”	1 Hour	Sanjit/Arnav
2	Create a modal which shall require the user password to delete the account	1 Hour	Arnav/Sanjit
3	Implement an endpoint that will remove a user account in the database.	2 Hours	Anjali/Shengqi
4.	Test by having users delete their account and ensure that the user is unable to login to Lance with their previous credentials.	1 Hour	Kenan

Acceptance Criteria:

- Given the frontend is implemented, the application should require the user’s password prior to deleting their account.
- Given the frontend and backend is properly implemented, the user should be asked for confirmation, and informed that all user information will be deleted and may not be recovered.
- Given that the user deletes his or her Lance account, their respective posts/profile information shall be deleted from the user facing application and the database.

(BONUS USER STORY)

User Story #14: I would like to utilize a more intuitive UI for ease of use.

Tasks	Description	Time	Owner(s)
1	Only show edit profile button if it is the current user's profile	2 Hours	Shengqi
2	Change the dropdown lists for choosing category to display the category name when selected	1 Hour	Sanjit/Arnav
3	Change UI elements to accommodate that a user isn't able to log off another user	2 Hours	Shengqi/Sanjit
4	Change the UI of the application to give a more modern and user friendly look.	3 hours	All

Acceptance Criteria:

- Given that the user is logged in, the user will only see the edit button on his/her own profile page
- Given that the user has specified a category in post creation or filters, he/she will also see their option directly in the label
- The user shall not be able to log off someone else by going to their homepage.

User Story #15: I would to like personalize my Lance user experience through provided theme options.

Tasks	Description	Time	Owner(s)
1	Create a UI element for 'Change theme' which will let users choose from options to customize their user experience	1 hour	Arnav/Sanjit
2	Create a modal that has 6-7 different themes as options and functionality for the user to choose them.	2-3 hours	Arnav/Sanjit
3	Choose background images and upload to github.	0.5 hour	Arnav/Sanjit
4	Implement an endpoint to select the theme for each user and save it in the database.	1.5 hours	Anjali/Kenan
5	Write JS code for retrieving client's choice of theme and sending a request to the server to make the change.	1-2 hours	Shengqi/Anjali

Acceptance Criteria:

- Given that the functionality is properly implemented, the user should be able to easily view and select from theme options.
- Given that the backend is implemented, the selection should immediately be reflected on the website.
- Given that the backend and server is properly implemented, the theme should remain updated the next time the user logs in, or refreshes the page.
- Given that the feature is properly implemented, the theme should applied on at least both home as well profile page.

User Story #16: I would like to be able to directly message other users using a chat system on Lance. (IF TIME ALLOWS)

Tasks	Description	Time	Owner(s)
1	Add UI elements to the pages for a direct messaging system.	4 hours	Arnav/Sanjit
2	Implement code in JS to let user to choose who they want to send a message to.	3 hours	Shengqi/Anjali
3	Implement code in JS which can accept user messages and send the request to the server.	3 hours	Shengqi/Kenan
4	Implement endpoints in the server to store the messages for each user that are sent and received along with the user ID who received or sent those messages.	5 hours	Anjali/Kenan
5	Implement in JS code which will let users see all of their messages on a page (sent/received)	4 hours	Anjali/Shengqi

Acceptance Criteria:

- Given that the direct message functionality is implemented, when the user wants to send a message to another user, they should be able to select to do so on the UI.
- Given that the user has selected the other user they want to send a message to, they should be able to write their message on a text area and click send to send the message to them.
- Given that one user has sent a message to another user, the second user should be able to see their new message in their messages list or page (dependent on our design approach).
- Given that the second user successfully is able to receive and view a message, they should also be able to send a reply to the first user.
- Given that the messages are sent through the application by the user, the database should reflect storage of those messages somewhere along with user ID of sender and receiver.

PRODUCT BACKLOG

1. I would like to be able to sign up on Lance using my Purdue email and password
2. I would like to be able to make a personal account on Lance.
3. I would like to be able to create a profile on the platform.
4. I would like to be able to upload and remove my personal documentation to my profile so that viewers will be able to determine my credibility.
5. I would like to be able to add and edit my skills.
6. I would like to be able to add and edit courses that I have taken at Purdue.
7. I would like to be able to add and edit a summary on my profile.
8. I would like to be able to add and edit links of other relevant accounts (LinkedIn profile, portfolio link, etc.) to my profile.
9. I would like to be able to upload and remove a cover photo on my profile.
10. I would like to be able to see the average rating given to me.
11. I would like to be able to look at a list of reviews on my profile.
12. I would like to be able to sign out of my account.
13. I would like to be able to view other users' profiles including information they have made public
14. I would like to be able to view stars by each user who has written a review on another user profile so that I am easily able to navigate between a low and a high rating review.
15. I would like to be able to scroll through a feed that shows me open posts for services offered and requested.
16. I would like to be able to make posts about services I am offering or requesting on Lance.
17. I would like to be able to specify the amount of money I am charging for my service, or willing to spend for a service request, on my post.
18. I would like to be able to edit/close my old post.
19. I would like to be able to browse all open posts on Lance to find my desired posts.
20. I would like to be able to add comments on the discussion board of any open request and offer.
21. I would like to be able to rate other users using stars out of 5.
22. I would like to be able to review other users for their services I've experienced.
23. I would like to be able to sort services on my feed based on their posting dates.
24. I would like to be able to sort services on my feed based on the cost of service.
25. I would like to filter posts using their categories.
26. I would like to receive a notification if someone interacts with my "Offer" or "Request"
27. I would like to be able to report a post.
28. I would like to be able to see the name of the users who have clicked interested on a post. (IF TIME ALLOWS)
29. I would like to be able to share a post of a request/offer. (IF TIME ALLOWS)
30. I would like to be awarded a credibility badge by the Lance platform for my presence on the platform. (IF TIME ALLOWS).
31. I would like to be able to view credibility badge(s) on my profile. (IF TIME ALLOWS).
32. I would like to be able to view credibility badges of other users on their profiles. (IF TIME ALLOWS).
33. I would like to be able to add pictures to my reviews. (IF TIME ALLOWS).
34. I would like to be able to view all pictures that are attached to reviews on my profile. (IF TIME ALLOWS).
35. I would like to be able to view all pictures that are attached to reviews on other users' profiles. (IF TIME ALLOWS).

36. I would like to be able to separately view requests and offers for services in two distinct feeds. (IF TIME ALLOWS).

37. I would like to be able to filter services using tags. (IF TIME ALLOWS).

38. I would like to be able to sort services on my feed using highest or lowest user ratings. (IF TIME ALLOWS).

39. I would like to be able to sort services on my feed based on their posting dates. (IF TIME ALLOWS).

40. I would like to be able to search for a user using their name. (IF TIME ALLOWS).

41. I would like to be able to directly message other users using a chat system on Lance. (IF TIME ALLOWS)

42. I would like to be able to delete my account on Lance. (IF TIME ALLOWS)

43. I would like to be able to click interested on a post. (IF TIME ALLOWS)

44. I would like to reset my password by clicking on a 'forgot my password' option.

45. I would like to be able to search for posts using keywords. (IF TIME ALLOWS)

46. I would like a more robust notification system incorporating a separate notifications page as well as a real time count of unseen notifications.

47. I would like to sort my reviews in ascending or descending order of ratings/stars.

48. I would like to be automatically logged out after 30 mins (variable) of inactivity on Lance.

49. I would like to personalize my Lance user experience through provided theme options.

50. I would like to utilize a more intuitive UI for ease of use.