UniLyfe

Sprint 2 Retrospective

Team: 4

Project: UniLyfe

Date: April 5, 2021

Carolyn Chen

Isha Jain

Ramitha Kotarkonda

Unnati Singh

Gayathri Sriram

1. What went well?

We were able to complete all fifteen of our user stories from our Sprint 2 planning document, which is a significant improvement from the first sprint. This lightens our workload for Sprint 3 and allows us more time to focus on the user stories dedicated to our final sprint.

One of our goals from the Sprint 1 retrospective was to make our user stories more specific so that the tasks are more manageable. Our Sprint 2 user stories and tasks were more manageable than those for Sprint 1. During this sprint, we completed a lot of features that make commenting and posting more user-friendly. We also implemented a few features allowing users to interact more with our app.

Something else that went well was our team dynamic. We did well with holding each other accountable each week of the sprint. We met regularly 3 days a week during our assigned meeting time. We used this time to share our work updates and gave each other suggestions on how to improve. In summary, we all worked together in an efficient manner to get all of our user stories completed for this sprint.

User Story #1: As a user, I would like to earn points by posting on the app and interacting with people on the app.

#	Description	Estimated Time	Owner
1	Create a user's attribute for points in the database that increments every time the user creates a post.	2 hrs	Carolyn
2	Create a section for points in the user profile UI.	2 hrs	Carolyn
3	Increment the number of points on the user profile UI when a user makes a poll/photo/text post.	3 hrs	Carolyn
4	Create an info button detailing the number of points awarded for each post.	2 hrs	Carolyn
5	Test to ensure that the point system works as intended (user's points increment when a post is made) and debug any errors.	3 hrs	Carolyn

Completed:

Users can now successfully earn points by creating a text post or posting a comment. The user's points are increased in the database whenever they perform these actions. Additionally, the user's points are properly displayed on the user profile page. The number of points on the user profile page always matches the number of points in the user's data in the database.

User story #2: As a user, I would like to input personal information when signing up with a new account on the app.

#	Description	Estimated Time	Owner
1	Users are able to choose a username.	2 hrs	Carolyn
2	Users will be given suggested username if the username they inputted is already existing in the database.	3 hrs	Carolyn
3	Users are able to input a short text biography, display name, their year in school, and profile picture.	2 hrs	Carolyn
4	Test functionality to ensure that this section works as intended, and that user information is properly being updated.	1 hr	Carolyn

Completed:
When users sign up for a new account on our app, they are redirected to a page where they are prompted to enter personal information, such as username, display name, and school year. They are also able to successfully enter their hobbies and classes. All of the information that the user inputted will reflect on the user profile page.

User story #3: As a user, I would like to redeem points for possible incentives (i.e. coupons)

#	Description	Estimated Time	Owner
1	Add a page in the menu about rules of receiving incentives	2 hrs	Carolyn
2	Add a page in the menu detailing the amount of points given for each post made	2 hrs	Carolyn
3	User's points will be subtracted when they click on an item to redeem	4 hrs	Carolyn
4	Test to make sure that the redeeming feature works as intended and debug any errors.	3 hrs	Carolyn

Completed:
Users are able to read about the rules of receiving incentives by clicking on the "Incentives" page in the app menu. When users click the "Redeem" button on the Incentives page, their points in the database will decrease by the number of points that the incentive is. This change will also be reflected in their user profile page.

User story #4: As a user, I would like to see how many people have tested positive for COVID-19.

#	Description	Estimated Time	Owner
1	Add a section where the user can view the number	1 hour	Ramitha
2	Create a counter which increments when the user changes their covid information to yes. Get the information from Cloud Firestore.	5 hours	Ramitha
3	Display the counter in the appropriate section	2 hours	Ramitha
4	Test and debug the counter functionality to ensure that the number of covid cases increase when the user changes their covid information to positive with covid.	5 hours	Ramitha

Completed: The user can view the covid section where the user can answer whether they have covid or not by pressing the yes or no button. When the user presses the yes button their information gets added to Firebase and if the user presses the no button their information gets deleted from Firebase. The counter next to the button also gets updated based on the user's response.

User story #5: As a user, I would like to be able to report posts for inappropriate content (racism/ hate speech)

#	Description	Estimated Time	Owner
1	Add a button to the posts so that the user can click on the button	1 hour	Ramitha
2	Create a series of survey questions to know more about the user's feedback	1.5 hours	Ramitha
3	Add the users feedback into firebase and sort the posts based on feedback	4 hours	Ramitha
4	Test and debug the survey functionality to ensure that the users are able to access the survey	5 hours	Ramitha

Completed: I added a button on each post to report the post. This button is not only present on the main page but also present on the rest of the subpages. Once the user clicks on the button, there are a series of questions that ask why the user is reporting the post. And once the user completes the survey, the post gets added to firebase.

User story #6: I would like to indicate whether or not I thought a review was helpful by liking the post.

#	Description	Estimated Time	Owner
1	Add/improve the button which would increment the count	2 hours	Ramitha
2	Connect the button to firebase	3 hours	Ramitha
3	Ensure all the posts have an increment button	1 hour	Ramitha
4	Test and debug the counter functionality to ensure that number of likes increments	5 hours	Ramitha

Completed: We added a like button to all of the posts and made the button interactable. For example, if the button clicks on the button, the heart shaped button changes to red and if the user clicks on the button again the button changes back to invisible. In addition, the button also remains red if the user navigates to another page within the app. We also displayed the number of likes each post has underneath the like button. And we ensured that if the user likes the post again, the count doesn't go up. And if the user decides to unlike the button, the number of likes returns to the original number.

User story #7: As a user, I would like to be able to search for keywords across all posts on the app.

#	Description	Estimated Time	Owner
1	Read through and understand search tutorial here: https://blog.smartnsoft.com/an-automatic-search-bar-in-flutter-flappy-search-bar-a470bc67fa1f	2 hours	Unnati
2	Implement the search UI on the search page	1 hour	Unnati
3	Make sure the search functionality looks for the keyword in the post (if it's hard to search throughout all the keywords, maybe just search for post name)	4 hours	Unnati
4	Testing functionalities and debugging: see if the search functionality works when I type a keyword and press a button	3 hours	Unnati

Completed: We have successfully implemented a searching feature. If you type a search query that matches the title of a specific post in the search bar, post titles should show up as suggestions down below. Upon clicking on one, you will be able to view and interact with that post. If you continue typing in the search bar, the post you are viewing should fade out of view and other post titles should show up instead. Additionally, if you click on the 'x' at the end of the search bar, the search query should disappear so you can start typing something entirely new.

User story #8: As a developer, I would like to access the data inputted by the user to display it properly

#	Description	Estimated Time	Owner
1	Access data that has already been inputted in FireStore and find a way to retrieve it	2 hours	Unnati
2	Display the data changes on the profile page after it has been retrieved	3 hours	Unnati
4	Testing functionalities and debugging: make sure correct information is displayed on the appropriate pages after a user makes changes	3 hours	Unnati

Completed: We have successfully implemented this. Now, the profile page is quite representative of the user data. Users can input information about their username, display name, biography, classes, and hobbies, and that information will be reflected both in the database and on their user profile. The only thing left to work on is the profile photo, which was hard to get to during this sprint since my laptop had issues with saving images to the database for some reason.

User story #9: As a user, I would like to view all the posts I have ever liked

#	Description	Estimated Time	Owner
1	Understand Isha's code and the way she is storing posts for each of the channels on the home page	2 hours	Unnati
2	Find a way to retrieve the 'like' boolean for each user	1 hour	Unnati
3	Add a list data structure to the user class for liked posts	1 hour	Unnati
4	Try to replicate a similar post channel for each individual's liked posts	4 hours	Unnati
5	Testing functionalities and debugging: make sure all the user's liked posts appear in their liked posts section.	3 hours	Unnati

Completed: We have successfully implemented this. Now, there is an additional channel on the homepage where users can see all the posts they have liked. If you unlike a post from a different channel, that post will no longer show up in the liked posts channel. When you like a post by pressing the heart at the lower left corner of a post, the color changes to red. If you unlike the post, the color will turn white again.

User story #10: As a user, I would like to reset my password if I forget it so that I can continue using my account.

#	Description	Estimated Time	Owner
1	Create a password recovery option on the login page.	1 hr	Isha
2	Create a UI with an input text field for users to insert their email address.	3 hrs	Isha
3	Send the password recovery email.	2 hrs	Isha
4	Update user's password on Firebase.	3 hrs	Isha
5	Test to verify that a user who is resetting their password receives an email with a recovery link and is able to reset their password.	1 hr	Isha

Completed: We successfully implemented this feature. If a user forgets their password, there is now a button on the 'Sign Up' page that directs users to a page allowing them to input their email. If there is no account associated with the email entered, the user will receive a warning message prompting them to enter a valid email. When a user enters a valid email, they will receive a success message as well as an email with a link that will allow them to reset their password. After the user resets their password through that link, they can sign in with their new password.

User story #11: As a developer, I would like to filter out possibly harmful words in the comments/posts.

#	Description	Estimated Time	Owner
1	Create an algorithm to detect harmful words on comments/posts.	3 hrs	Isha
2	Add warning messages that will display if a user tries to submit a post/comment with hurtful words.	3 hrs	Isha
3	Create an algorithm to censor harmful words in comments/posts.	2 hrs	Isha
4	Test to ensure that the user is receiving warning messages if they are about to post/comment something with hurtful words.	1 hr	Isha
5	Test to make sure that harmful words are censored in posts/comments containing them.	1 hr	Isha

Completed: We have successfully implemented this feature with the help of an API that has an extensive list of hurtful words. Whenever a user tries to upload a comment or post with bad words, they will receive a warning message that explicitly states the inappropriate words. If the user proceeds to upload the comment/post after receiving the warning, an algorithm will censor the harmful words with asterisks. There are limits to this feature, which is why we have a separate reporting feature.

User story #12: As a user, I would like to view my comment history.

#	Description	Estimated Time	Owner
1	Create a page where the user can view their comment history.	1 hr	Isha
2	Create a button/tab on the user page that links to the page to view comment history.	1 hr	Isha
3	Add a collection on Cloud Firestore to store the specific user’s comments.	3 hrs	Isha
4	Retrieve the user’s comments from Cloud Firestore and display them on the comment history page.	4 hrs	Isha
5	Test to make sure all the user’s comments appear in their comment history page.	1 hr	Isha

Completed: We have successfully implemented this feature. When the user goes to their profile page, there should now be a button that states ‘View Comment History.’ When clicked, the user should be redirected to a page showing all the comments ever made by the user, ordered from most recently made to least recently made. In Cloud Firestore, each user has their collection that stores all the comments they make. As this collection updates, so will the comment history page.

User story #13: As a user, I would like to view the profile of the post creator or the comment creator.

#	Description	Estimated Time	Owner
1	Add the post creator's username to the post that they create	2 hours	Gayathri
2	Add the post creator's username to the comments that they create	2 hours	Gayathri
3	Link the username that is displayed in the comments/ posts to that specific user's profile	3 hours	Gayathri
5	Test and debug the functionality that ensures the user profile is successfully loaded after clicking the username	3 hours	Gayathri

Completed: We have successfully completed this feature. When a post is published on the feed page, the post creator's username is displayed right above the post title. When a user clicks on the post creator's username, a popup displays with the post creator's biography. In this popup, the post creator's display name, username, biography, hobbies, classes, and their points are displayed. This is also applied to each subpage of the UI not just the feed page. Similarly, when a user makes a comment, the username is now displayed right above their text comment. When the app user clicks on the comment creator's username, a popup with all the aforementioned details are displayed.

User story #14: As a user, I would like to gain a comprehensive understanding about the app and its developers.

#	Description	Estimated Time	Owner
1	Create a page that links to “About the app” and fill it with information about the backstory of the app	2 hours	Gayathri
2	Create a page that links to “About the creators” and fill it with biographies about each developer	2 hours	Gayathri
3	Create a page that links to “Help” and fill it with how the app works	3 hours	Gayathri
4	Create a “FAQ” button and create a page that links to it and fill it with commonly asked questions about the app	2 hours	Gayathri
5	Test and debug functionality that allows users to view information about the app	2 hours	Gayathri

Completed: We have successfully completed this user story. When the user clicks on the side bar that is denoted by three small horizontal lines located in the top left corner of the app, a page that takes up about 70% of the screen shows up. In this side bar, there are now four new buttons visible: “About us”, “About the app”, “Help”, and FAQ”. When the “About the app” button is clicked on, the user is directed to a page that displays the purpose of the app in a paragraph format. When the “About the creators” button is clicked, the user is directed to a page that contains biographies for each developer of the app. (the 5 members of team 4). When the “Help” button is clicked, the user is directed to a page that contains a “User’s guide to navigating through the app”. Lastly, when the user clicks on “FAQ” they are directed to a page that has 4 commonly asked questions about the app and insightful answers are provided as well.

User story #15: As a user, I would like to see a completely randomized post/event/comment/location.

#	Description	Estimated Time	Owner
1	Create a pressable button that says ‘randomize page’ and add it to each subpage of the UI	3 hours	Gayathri
3	Create a pressable button that says ‘revert’ and add it to each subpage of the UI	2 hours	Gayathri
4	Implement functionality so that the order of the page isn’t determined by time posted	3 hours	Gayathri
5	Test and debug functionality so that posts displayed in randomized fashion.	3 hours	Gayathri

Completed: We have successfully completed this user story. Now on each subpage of the UI, there are two buttons that are displayed. “Randomize page” and “Revert Changes”. When the randomize page button is clicked, the order of the posts displayed on that page are now in a random order. When the revert changes button is clicked, the order of the posts revert back to reverse chronological order (most recent posts are displayed first). These buttons and their functionality are applied to each subpage of the UI (food, study, and social).

2. What did not go well?

We had some unfinished user stories from Sprint 1 that we did not have time to complete during Sprint 2. For example, we couldn't find the time to complete the user story allowing users to make image/video posts. Nor were we able to get the profile picture user story working with the database. However, we were able to complete the vast majority of them, and we do not have any user stories leftover from this sprint. Our goal is to finish those stories during Sprint 3 in addition to the ones we plan to complete.

User story #15 (from Sprint 1): As a user, I would like to rate places based on my personal experience.

#	Description	Estimated Time	Owner
1	Stars shaded correspond to the rating out of 5 when a user is selecting a rating for a certain location when creating a post	2 hours	Gayathri
2	Stars are shaded corresponding to the *average* ratings of a location	2 hours	Carolyn
3	On a location's page, their average ratings out of 5 should be displayed	2 hours	Carolyn
4	To test: Test and debug rating functionalities for single ratings and average ratings	4 hours	Carolyn

Underway: This user story is almost complete. However, there are a few functionalities that are missing. We need to display the average ratings of a location once the user has rated the post. We also need to update this information in the database. This shouldn't take too long, and we plan on doing this as a group in Sprint 3.

User story #5 (**from Sprint 1**): As a user, I would like to upload photos/videos so that my peers will have an idea of what the place is like.

#	Description	Estimated Time	Owner
5	Post is taken to the feed page	2 hours	Carolyn

Underway: Our application currently has a form where the user can create an image/video post, open the device's camera or camera roll, and request to access the user's camera roll or camera for the first time. The user is also able to add a text caption to the image post they would like to upload. Due to time constraints, we could not bring the image post to show up on the home feed page.

3. How should you improve?

We can improve by managing our time more efficiently. We tend to complete our user stories close to the deadline. But if we start and complete them earlier, it would allow us to have more time to test and debug the functionalities of our user stories. We've been pretty good at finishing one user story per week, but perhaps we can aim to complete more if we have the time. That way, we can ensure that our app is working as expected before our final demo.

Another way we can improve is to hold more weekly meetings. Right now, we meet about three days a week for updates and to work on group commits. We think it will be beneficial for all of us if we hold maybe one or two more meetings a week. We can use this time to ask each other questions and hold meaningful conversations about our progress. We can also demo what we've accomplished and get feedback from the members of our team. This will make our individual work go faster as it usually takes a while to figure out the work that is done by the rest of our teammates.