



Sprint 3 Retrospective

Team: 4

Project: UniLyfe

Date: April 26, 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 3 planning document. We were also able to successfully complete all the user stories that we had remaining from our previous sprints. Thus, our project for this class is now complete.

Our team dynamic was exceptional. We did well with holding each other accountable each week of the sprint. We regularly met numerous times 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 developer, I would like to store photos in the database.

#	Description	Estimated Time	Owner
1	Add a collection for photo posts in the database.	2 hrs	Carolyn
2	Have users' photo posts show up on the home page.	5 hrs	Carolyn
3	Store the photo of the photo post in the database.	3 hrs	Carolyn
4	Test and debug functionalities	1 hr	Carolyn

Completed: Users can now select photos from their camera or from their device's camera roll. They can submit this photo as a post on the app, and it will appear on the home page as a post. The photo's path, which corresponds to the photo chosen by the user, is stored successfully as a field in the database.

User story #2: As a user, I would like to earn profile awards for reaching a certain number of points.

#	Description	Estimated Time	Owner
1	Make a user attribute for awards in the database.	2 hrs	Carolyn
2	Have awards icons show up on the user profile page if the user is eligible for awards.	4 hrs	Carolyn
3	Find and upload icons for awards.	3 hrs	Carolyn
4	Insert a page in the menu that details the rules for earning awards.	2.5 hrs	Carolyn
5	Test and debug functionalities for the user profile awards.	1 hr	Carolyn

Completed: When users reach a certain number of points, they will earn an award on their user profile. The rules for the awards system can be found in a tab under the menu page. In the database, there is a field for the number of awards that a user currently has. Additionally, the number of awards will correspond with the number of awards listed in the database.

User story #3: As a user, I would like to add more places which not many students know about.

#	Description	Estimated Time	Owner
1	Implement ability for users to add location on the UI for creating a post.	3 hrs	Carolyn
2	Add a field for the location of a post in the database.	3 hrs	Carolyn
3	Connect the location selected on the post to the location field on the database.	3 hrs	Carolyn
4	Test and debug functionalities of the location function on posts.	1 hr	Carolyn

Completed: On text posts and on photo posts, there is a field for users to type in a location. The text from this field will be labeled on the post in the home page, which allows users to know about more places on or near campus. The text from the location text field is stored in the database under the posts collection.

User story #4: As a user, I would like to be able to report users 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.	2 hour	Ramitha
2	Create a series of survey questions to know more about the user's feedback.	4 hours	Ramitha
3	Add the users' feedback into Cloud Firestore.	4 hours	Ramitha
4	To test: Test and debug the survey functionality to ensure that the users are able to access the survey.	1 hour	Ramitha

Completed: We added a report user button to the same section as the report post button for every post. This button is not only present on the main page but also present on the rest of the subpages. Next, we created a series of survey questions to know more about the user's feedback. Once the user completes the survey, the user's information gets stored into Firestore.

User story #5: As a developer, I would like to remove reported posts.

#	Description	Estimated Time	Owner
1	Implement the functionality to count the number of people who have reported the post.	2 hour	Ramitha
2	Implement the functionality to remove the post once the count has been exceeded.	4 hours	Ramitha
3	Store the deleted posts into firebase to still have access to the deleted posts as a developer.	3 hours	Ramitha
4	To test: Test and debug the survey functionality to ensure that the posts get removed properly.	1 hour	Ramitha

Completed: A previous user story was to store the reported posts to firebase. I used the information stored in firebase and added additional information to the reported posts such as the number of times the post got reported. Once the number of times the posts got reported has reached the limit, then we remove the post from the home and subpages. However, the information is still accessible to the developers. The developers can later determine if the post has actually violated UniLife's guidelines and decide whether to remove or keep the post.

User story #6: As a user, I would like to add information about events happening on campus.

#	Description	Estimated Time	Owner
1	create a tag to add information about events happening on campus.	2 hours	Ramitha
2	Allow users to create a post related to events and enable the users to add information about the event.	5 hours	Ramitha
3	Display the events on the subpage.	2 hours	Ramitha
4	Display a notification of the upcoming events which are 5 days away	2 hours	Ramitha
5	To test: Test and debug the events post functionality.	1 hour	Ramitha

Completed: I created another option for users to create an event post. The user can add information about the event on campus such as the location, the time, the date, the description of the event, etc. Once the user has submitted the information, the user gets a notification about the upcoming events which are 5 days away. The event which was created is visible on the main page and as well as on the subpage.

User story #7: As a user, I would like to update my profile picture.

#	Description	Estimated Time	Owner
1	Research more about the image picker that flutter provides	3 hours	Unnati
2	Since the image picker function is not really working on my laptop, collaborate with Gayathri and try to figure out how to update the profile picture on the users screen.	3 hours	Unnati
3	Figure out how to store the profile picture in the database.	3 hours	Unnati
4	To test: Test and debug the updating profile picture functionality.	1 hour	Unnati

Completed: We have successfully implemented a feature where you can update your profile picture. If there is no profile picture selected, the user should have their initial on a random color. If they do select a profile picture, the picture should be visible on their profile, their friend cards, and on the comments they make.

User story #8: As a user, I would like to make friends with those who have similar interests, taking the same/similar classes, and those who are in the same/similar clubs/extracurricular activities as me.

#	Description	Estimated Time	Owner
1	Create a “friends” channel on the profile page .	2 hours	Unnati
2	Create a “friend” card similar to the post card that gets generated when a user creates their account and modified when a user updates their information.	5 hours	Unnati
3	Create a following friend functionality so you can follow friends if you want. These would ideally show up in the “friends” channel.	3 hours	Unnati
4	To test: Test and debug the following friends functionality.	1 hour	Unnati

Completed: We have successfully implemented a feature where you can add friends, and the friends with the same classes or hobbies are indicated as well. You can just press the “add friend” button, and the friend will go to the “friends” channel on the user profile page.

User story #9: As a developer, I would like to access and analyze information that users upload about themselves for their profile.


#	Description	Estimated Time	Owner
1	On the body of the search page, create a “suggested” section.	2 hours	Unnati
2	Add friends who have either the same classes or the same hobbies or both in this suggested section.	7 hours	Unnati
3	To test: Test and debug the suggested friends functionality.	1 hour	Unnati

Completed: We have successfully implemented this. We are able to go through the user data and find the intersection of both the current user’s classes and hobbies and the potential friend’s classes and hobbies, making it easy for users of the app to find others who are similar to them.

User story #10: As a user, I would like to be able to delete any comments or posts I have made.

#	Description	Estimated Time	Owner
1	Research how to remove a document/collection from Cloud Firestore through Dart code.	1.5 hrs	Isha
2	Add a button to all posts with a garbage icon that will delete the post when pressed.	2 hrs	Isha
3	Implement a swipe to delete feature for comments after researching the Dismissable widget.	3 hrs	Isha
4	Add a warning message to make sure the user actually wants to delete the comment/post.	2.5 hrs	Isha
5	Test the functionality to make sure that comments/posts are properly deleted from Cloud Firestore, and that they are no longer displayed after being deleted.	1 hr	Isha

Completed: We successfully implemented this feature. Now, every post made by the current user has a garbage button. Similarly, every comment made by the current user has an option to delete the comment. To make this option appear, users have to swipe left on the comment. When users try to delete a post/comment, they will receive a warning message that asks the user to confirm whether they want to delete the post/comment. If the user proceeds to delete the post/comment, it will get removed from the database. If a post has been deleted and has likes, it will get removed from every




user's 'Liked Posts' section. If a comment gets deleted, it gets removed from the user's comment history. Lastly, if a post gets deleted, all comments are deleted and are removed from the user's comment history.

User story #11: As a user, I would like to avoid COVID hotspots on/near campus which are displayed by red dots on the map by seeing a list of these locations.

#	Description	Estimated Time	Owner
1	Research the Google Maps API, and figure out how to add custom markers to the map. Figure out how this API can be used in conjunction with Cloud Firestore.	2.5 hrs	Isha
2	Create a collection in Cloud Firestore that will store a list of locations users with COVID-19 have been.	3 hrs	Isha
3	Retrieve data from this collection in Cloud Firestore. Display a red dot at each location specified in the data.	5 hrs	Isha
4	Create an info page that displays a list of locations and their number of COVID-19 cases.	2 hrs	Isha
5	Test to ensure that red dots are properly being displayed on the map, and that the list on the info page displays the locations with the most COVID-19 cases.	1 hr	Isha

Completed: We have successfully implemented this feature with the help of a couple of APIs. Now when the user clicks on the map, there is a draggable sheet at the bottom. When expanded, the user can see two tabs: one with an 'INFO' page and one with a




'GOT COVID?' page. On the 'INFO' page are the number of logged COVID-19 cases and a list of locations that COVID-19 positive users have logged, along with the number of cases right next to them. On the 'GOT COVID?' page is a form that asks the user if they have COVID-19. If the user clicks 'Yes,' a search bar will appear, allowing for users to search for locations they've visited while positive. As the user searches for a location, suggestions pop up courtesy of the Google Places API. Once the user selects a place and hits the 'Submit' button, the map will update with a red dot marker representing that location. In addition, the list of locations on the 'INFO' page will update.

User story #12: As a developer, I would like to provide a default avatar to all users when they first sign up.

#	Description	Estimated Time	Owner
1	Add a 'color' field in the userData collection on Cloud Firestore for each user. Assign users a random color when they first sign up.	2 hrs	Isha
2	Research the CircleAvatar class in Dart or similar classes to figure out the best way to construct a default avatar.	1.5 hrs	Isha
3	Using the user's color field and the first letter of their display name, construct a default avatar.	3 hrs	Isha
4	Add an option on the profile page allowing users to remove their current profile picture, replacing it with their default avatar.	2.5 hrs	Isha
5	Test to ensure that the default avatar is being constructed correctly, and that users are able to remove their current profile pic in favor of the default avatar.	1 hr	Isha

Completed: We have successfully implemented this feature. Now when a user signs up for a new account, they are assigned a randomized color code. This code, along with the user's display name, is used to construct a default avatar. The default avatar is used if the user does not already have a profile picture set. It appears on users' comments, on




their profile, and users' friend cards. This avatar updates if the display name is changed (assuming that the first letter of the display name changes). In addition, when users click on the camera button on their profile, there is now an option to remove their current profile picture. When removed, in place of the profile picture is their default avatar.

User story #13: As a user, I would like to write reviews about housing and restaurants that are located on or near campus.

#	Description	Estimated Time	Owner
1	Add a pressable button called "Create a review" located in the (+) or create page	2 hours	Gayathri
2	Once the button is clicked, the user is taken to a page that has a dropdown list of predetermined locations on/near campus	3 hours	Gayathri
3	Once the user chooses a location, they are presented with a 5 star rating system where they can rate that location.	2 hours	Gayathri
4	Implement functionality so that each star is able to be clicked up to the desired value	2 hours	Gayathri
5	Implement functionality so that users can rate reviews with extra precision. For example: 3.5 stars	2 hours	Gayathri
6	Test and debug functionality so that users can create residence hall/apartment reviews	1 hour	Gayathri

Completed: We have successfully completed this feature. When the user clicks the (+) button to create a post, they are presented with an option to create a review post. Once this button is clicked, the user is presented with a list of locations on and near campus. They are also presented with a 5 star rating system to rate that specific location. When



the user wants to rate a location, they can click on the number of stars they want to rate it, and the stars stay highlighted at that value. Each star can be partially clicked as well. Lastly, the user can click the “submit review” button after they finish creating their review.

User story #14: As a developer, I would like to have access to information pertaining to the locations and location ratings.

#	Description	Estimated Time	Owner
1	Ensure that the 5 star rating that a user chooses is saved in Firebase	3 hours	Gayathri
2	Ensure that the location that a user chooses is saved in Firebase	3 hours	Gayathri
3	Display the user's rating for that location on the UI near the 5 star rating scale	4 hours	Gayathri
4	Conduct manual tests that ensures that the Firebase database is updated with the location rating data	1 hour	Gayathri

Completed: We have successfully completed this user story. There is now a “reviews” collection on Firebase. When the user creates a rating on the 5 star rating scale, that rating is saved as a field in the “reviews” collection in Firebase. When the user chooses a location from the dropdown list of location, that location is also saved as a field in the “reviews” collection in Firebase. Lastly, in the UI, the 5 star rating with the user's shaded stars is displayed in the form of a review post.

User story #15: As a user, I would like to *read* reviews about housing and restaurants that are located on or near campus.

#	Description	Estimated Time	Owner
1	Add a section called “Reviews” as a fourth subpage of the UI	2 hours	Gayathri
2	After the user finishes creating their review, their 5 star review should be displayed on the “Reviews” subpage.	3 hours	Gayathri
3	After the user finishes creating their review, the location that they choose from the dropdown list should be displayed on the “Reviews” subpage	3 hours	Gayathri
4	Test and debug functionality that allows users to rate reviews out of 5 stars	1 hour	Gayathri

Completed: We have successfully completed this user story. Now there is another subpage on the UI titled “Reviews”. This page now lists all posts from users regarding their reviews that they created. Essentially, the “reviews” subpage now contains reviews of locations on/ near campus. As mentioned in my description for the completion of user story 13, right after a user finishes creating their review post, they click “submit review”. Now, when they go to the reviews subpage, they can see their post, with the location that they chose along with the rating they chose. There is also additional information about the post, such as the post title, textual description of the review, and time of the post that is displayed.

2. What did not go well?

We had a bit of struggle with time management this sprint. Though in the end we were able to successfully complete all 15 user stories with very high quality, there were a few days we worked for over 15 hours non stop because we didn't spread out our work across the week in an efficient manner. There were also a few issues beyond our control. For example, one member of the team who was working on something related to the profile picture was unable to work on her computer because of her computer's M1 chip. Because of this she had to use the remote control feature of Zoom to work through another team member's computer. In hindsight, we could've planned this out so that someone else could be assigned to this user story.

3. How should you improve?

One way we can improve the issue of not having the appropriate resources to complete a user story, is spend more time during drafting our Sprint 3 planning document. If we had assigned someone else the profile picture user story, then we could've avoided the extra hassle of using the remote control feature of zoom to take control of another member's screen to code. Although, I will mention that this definitely was a good and successful attempt at problem solving.

Another way we can improve is to set realistic daily task goals and aim to accomplish it. Though we do this in our individual report every week, sometimes our plans are a bit too difficult to accomplish because we say we can do a lot more in one day than we can actually get done and this leads us to get overwhelmed. If we set a realistic schedule that we are more likely to accomplish, we will be more motivated and thus more likely to get it done on that day itself rather than last minute.