Meliora

Good Today, Better Tomorrow

CS 407 - Software Engineering Senior Project

Sprint 2 Retrospective Document

Team 9

Dithi Saxena (saxena52@purdue.edu)
Abhishek Gunasekar (agunase@purdue.edu)
*Xavier Madera (*xmadera@purdue.edu*)*
*Vidit Shah (*shah444@purdue.edu*)*
*Garrett Lee (*lee2920@purdue.edu)

TABLE OF CONTENTS

What went well?

In general, we developed the advanced features of Meliora. Now, a user can view trending posts, add tags to existing posts, and react to other posts through positive emojis. In addition, we also developed a mental health resources page for struggling users and the ability for users to add their location within their posts.

User Story #1 - 5.1

As a user, I would like to quickly access helpline information (such as suicide, ptsd, etc.) through the mental health resources page.

Completed:

When the user clicks the reach out for help button in the homepage, Meliora displays the mental health resources page. The mental health resources page contains the hotline numbers, important email addresses, and useful websites for users struggling with suicide, domestic violence, child abuse, or alcohol abuse related concerns. This provides a quick way for the user to reach the concerned party for help rather than searching online for help.

User Story #2 - 6.4

As a user, I would like to view the total number of reactions that my posts have received.

Completed:

Since Meliora relies on the social interactions aspect of the user community, tracking reactions to posts is an important functionality of the application. When the user visits their profile page, the total number of thumbs up, likes, smileys, and hugs are displayed, thereby motivating the user to create more posts and improving the user experience.

User Story #3 - 4.7

As a user, I would like to record my location in my post.

Completed:

In the create posts page, when the user clicks the record location button, their respective latitude and longitude are displayed with the help of the Google Maps API. If their location is undetectable, Meliora displays their location as “unknown.” The user can also choose to remove their location once they have added their location to their post.

User Story #4 - 4.13

As a user, I would like to be able to hide my posts from the public eye, to where only I can see them.

Completed:

Privacy is an important feature of Meliora. Hence, if a user chooses to make their post private such that only they can view their posts, they can click the “make private” button in their post. Then, their post becomes hidden for other users in the Meliora community and is only visible to the respective user.

User Story #5 - 4.3

As a user, I would like to be able to restrict my post to where other users cannot post comments on it.

Completed:

The user can successfully disable comments when creating a post on the app. In fact, the default option for creating a post is no comments. When comments are disabled on a post, no user can comment on the post.

User Story #6 - 3.3

As a user, I would like to see a list of dynamic categories that describe the common mental health topics (relationships, grief, stress, etc) tagged by users in their posts as stated in user story 4.9.

Completed:

The user now sees the most trending categories on the homepage. There is also a separate page built where the user can see all the trending categories and posts. The trending categories are calculated by iterating through each of the categories, then checking for the number of reactions on the posts for each category in the last 3 hours. The categories with the most reactions in the past 3 hours would be the trending categories. Trending posts are calculated in a similar manner.

User Story #7 - 4.2

As a user, I would like to be able to leave comments on others' posts.

Completed:

When comments are enabled on a post, the option for any user to add a comment is available. When there are comments on a post, the user will be able to see the content and username of each comment. If the owner of either the comment or the post is viewing, then the option to delete is displayed.

User Story #8 - 4.12

As a user, I would like to be able to see my word count while writing posts.

Completed:

Like many other social media apps, this app also allows users to see their word count when typing a post. This word count dynamically changes when a user starts typing, and is initialized at 0. When the user is finished typing, they will see the total number of words on the bottom left of the text box.

User Story #9 - 4.8, 6.7

As a user, I would like to be able to see my bookmarked posts.

Completed:

When viewing a post, there is now an icon that allows the user to bookmark a post. When clicked, it is added to the database. After bookmarking one or more posts, when viewing your own profile there is an option to toggle between your posts and bookmarked posts.

User Story #10 - 3.5, 3.7

As a user, I would like to follow/unfollow other user profiles in Meliora through their profile or the homepage and see their posts in a “from people you follow” section.

Completed:

When viewing another user’s profile that is not blocked or private, the option to follow is displayed. Once followed, the user will have the option to unfollow and the posts are all displayed on the user’s homepage feed.

User Story #11 - 8.6

As a user, I would like the option to block or unblock another user.

Completed:

When a user decides that another user should no longer be able to see their profile or posts, they can use a button to block them. This way, when the other user visits their profile, they are met with a blank profile.

User Story #12 - 8.7

As a user, I would like to have the option to make my profile private or public.

Completed:

A user may choose to have their profile hidden from others by making it private on their settings page. This way, no one can see the user's information or posts if they do not feel comfortable sharing their posts.

User Story #13 - 4.1

As a user, I would like to react to other people’s posts with positive emojis (hearts, smiley faces, thumbs up, etc.).

Completed:

Like all other social media apps, users can react with positive emojis to a post. A user can click the ‘add reaction’ icon and choose from four different emojis: thumbs up, smiley face, heart, and hug. When a user selects a reaction, that reaction will show up on that post. When a user deselects their reaction, they will be given the option to select a new reaction.

User Story #14 - 3.6

As a user, I would like to follow/unfollow multiple categories through the homepage.

Completed:

Users can now go to a dedicated category page where they can see all of the topic information such as the description, posts, etc. On this topic page, the user has the option of following and unfollowing the given category. The UI is dynamically updated and the data is appropriately sent to the backend.

User Story #15 - 4.9

As a user, I would like to tag each post with an existing category.

Completed:

While creating posts, users can now choose which category they would like their post to fall under. The category of the post is then appropriately stored on the backend and displayed on the posts on the homepage and anywhere else a postcard is displayed.

User Story #16 - 9.2, 9.3, 9.4

As an admin, I would like a backdoor into the app to review delinquent posts and decide if they belong on our platform.

Completed:

We needed a way to ensure that all content on our platform meets our vision of what the platform should be. We implemented a way for us to designate users to help review content that has received negative feedback via flagging and help us decide which of these posts belong on our platform and which don't. A so-called admin can ask our backend for posts that have been marked "delinquent," meaning posts that have been flagged more than five times. They can use the ID of these posts to make a decision on whether or not they belong on our platform. They can then ask the backend to either clear the posts' flags or delinquent status, or to remove the post.

What did not go well?

We had several communication errors that came up towards the end of our second sprint. Some features weren't working properly and had to be modified not far before our sprint demo. This led to complications when overlapping features were involved, meaning more modifications had to be made leaving us with a small time constraint.

Moreover, some of our features were interrelated with different people working on them. The people working on some of these user stories implemented the features that worked for the time being. However, when another person started working on a related user story, the structure according to which the previous feature was implemented hindered the development of the new user story. Thus, we had to restructure the way we implemented the previous user story in order for other user stories to work. This added some unnecessary time to our tasks.

Unsuccessful User Stories

Our group finished all of our user stories planned for this sprint. As a result, we did not have any unsuccessful user stories for this sprint.

How should you improve?

There are many lessons we learned from this second sprint. First, we have realized that we must start on working on our tasks with more urgency during the first week of the sprint. This sprint had more complex features involved, and because of this, many team members did not anticipate the difficulty of some tasks. Some tasks ended up taking more time than we had allotted on the sprint planning document. We could improve on this next sprint by starting our work earlier and not procrastinating on finishing our user stories closer to the sprint deadline. This method would also allow us more time to debug and test our code for optimum functionality with the extra time we would have at the end if we did not spend it at the last minute finishing our tasks.

Another important takeaway from this sprint is to communicate early. For some of our user stories, the respective person assigned did not communicate well with the other teammates regarding the dependency required to finish their tasks. For instance, if the person handling the frontend required a backend route to be set up by another person, they were not proactive in communicating this requirement. This led to adding more code at the last minute when we assumed all the tasks were completed. Hence, communicating early and often is a lesson learned for the final sprint.

Another way we can improve moving forward is to make sure that all design decisions are clarified where one person’s work depends on another’s. This way, we can all do our work and move on, without anyone having to alter or ask someone to alter the person’s work and everyone is once again on the same page.