



Meliora

Good Today, Better Tomorrow

CS 407 - Software Engineering Senior Project

Sprint 1 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?	3
• User Story #1	3
• User Story #2	4
• User Story #3	5
• User Story #4	6
• User Story #5	7
• User Story #6	8
• User Story #7	9
• User Story #8	10
• User Story #9	11
• User Story #10	12
• User Story #11	13
• User Story #12	14
• User Story #13	15
• User Story #14	16
• User Story #15	17
• User Story #16	18
• User Story #17	19
• User Story #18	20
• User Story #19	21
• User Story #20	22
What did not go well?	23
How should you improve?	24

What went well?

In general, we developed the basic functionality of the *Meliora* web app. A user can create an account, login, create a post, view posts, edit their profile, and change their settings. We also implemented a strong backend foundation which communicates with the frontend of the web app.

User Story #1 - 2.2, 3.1

As a user, I would like to view the landing page to understand more about the features of *Meliora*.

Completed:

The *Meliora* landing page has a consistent user interface that is reflected across the rest of the application. The navbar consists of links to the about us, features, and community sections of the landing page as well as a link to the sign up and login page. When a user clicks either of these, he or she will be redirected to the respective component of the application.

User Story #2 - 2.1, 2.6

As a user, I would like to see a navbar containing links on each webpage of the application.

Completed:

The *Meliora* app consists of two different types of navbars: one used when the user is not logged in and the other when the user is logged in. When the user is not logged in, the navbar is designed to help the user navigate through either the landing, sign up, or login page. When the user is logged in, the navbar allows the user to navigate between the home, posts, and notifications page. The Meliora logo in each of the navbars described above redirects the user to the landing page or the homepage depending on their login status.

User Story #3 - 1.1, 1.3, 1.4

As a user, I would like to be able to create an account with *Meliora* so that I can access all its features.

Completed:

The *Meliora* sign up page was developed in a robust manner with an exhaustive list of error checks. For instance, if the user fails to fill out all the input text fields, then the application displays an error, preventing the user from logging in. In addition, if the user does not enter the correct credentials or tries to sign up with a set of pre-existing credentials, *Meliora* will not permit the sign up process to complete, flagging it as an error. After the user correctly signs up, their credentials are stored in a MongoDB cluster, to be retrieved at a later time.

User Story #4 - 1.5, 1.6, 1.4

As a user, I would like to login to my account using my username/email and password.

Completed:

Similar to the signup page of the application, the login page conducts the necessary error checks with the credentials provided by the user. If the entered credentials are correct, then Firebase authenticates the respective user and redirects him or her to the homepage.

User Story #5 - 3.2

As a user, I would like to see a motivational quote whenever I access the app.

Completed:

Utilizing the Forismatic REST API, *Meliora* displays a randomized motivational quote every time a user logs in to the application. Within 8 seconds after the quote is displayed, the user is redirected to the homepage. If, for some reason, the API did not return a quote object, the *Meliora* slogan “Good Today, Better Tomorrow” is displayed to welcome the user.

User Story #6 - 2.5

As a user, I would like to have an option of creating a new post from the header.

Completed:

When logged in, the user will see a “Create” option in the nav-bar. When clicked, a form will be displayed allowing the user to create a post with a title, content, and toggle anonymous mode. After creating, the post is added to the user’s author list and loaded on the profile page.

User Story #7 - 2.4

As a user, I would like to view my profile and profile settings from the header.

Completed:

If a user clicks on their username on the right side of the nav-bar, they have the option to view their profile or their settings. When clicking settings, several elements like phone number, email, dark mode, and delete account are displayed. When clicking “My Profile,” the user is taken to their profile page where further editing can be done.

User Story #8 - 6.1

As a user, I would like to easily edit my profile page.

Completed:

The edit-profile page allows the user to easily make all their edits in one place. A user can click the edit icon on the profile page where they are redirected to the edit-profile page, and then edit their handle name and their bio. They also are notified when their changes are saved, and they can easily click the back icon to return to the profile page and see their changes reflected on the profile page.

User Story #9 - 6.6

As a user, I would like to view another user's profile with their public information.

Completed:

When users view another post and click the author's username, they are now taken to the profile page. When viewing your own profile, the edit profile icon is displayed, and anonymous posts are visible. When viewing someone else's profile, there is no pen icon, and you cannot see their anonymous posts. Public information like their profile picture, username, bio, and public posts are displayed.

User Story #10 - 6.2, 6.3

As a user, I would like to describe a short bio about myself in the profile page and see the total number of posts I have made in total.

Completed:

The profile page elegantly displays the total number of posts a user has made as well as their bio underneath it. If a user chooses to edit their bio, they can simply click the edit icon next to their name. Lastly, everytime a user creates a new post, the count on the number of posts will be accurately ended in the backend and the frontend UI.

User Story #11 - 6.8

As a user, I would like to upload a profile picture for others to see.

Completed:

As part of the edit-profile page, a user can choose to upload a profile photo for others to see on their profile and their posts. The file picker in their operating system is called for them to select a file. When they pick a file, it then goes to Firebase. From there, it can be referenced via their authentication session.

User Story #12 - 8.3

As a user, I would like to view the web app in dark mode.

Completed:

Like many other web apps, the *Meliora* app also allows the user to switch their display to dark mode. They can switch their dark mode preference from the settings page, where the changes will be reflected in all *Meliora* web pages. Lastly, when the user logs out of the web app, the user's dark mode preference does not persist.

User Story #13 - 8.1

As a user, I would like to be able to change my password once logged in.

Completed:

If a user wishes to change their password, they can head to the settings page of the app. Here, there is a button, called “Send Password Reset Email,” that works with firebase. Firebase sends the user an email to change their password via a link. The next time the user logs in, they will use the new password they gave to firebase to log in.

User Story #14 - 8.2

As a user, I would like the option to sign out of my account.

Completed:

The user can easily choose to sign out either from the navigation bar or the settings page. When the user chooses to sign out, their authentication token is revoked and their dark mode preference is also removed. Once signed out, the user will be redirected to the login screen.

User Story #15 - 8.4

As a user, I would like to be able to delete my account.

Completed:

Like most websites that require accounts, *Meliora* allows the user to delete their account. They can easily do so by navigating to the settings page and clicking delete account. To make sure this action is intentional, a pop-up alert will ask the user to confirm this course of action since it is not reversible. When a user has deleted their account, they will no longer be able to log back in and their associated posts are also deleted.

User Story #16 - 2.3

As a user, I would like to view a log out button on every page except the landing page.

Completed:

On every page except for the landing page, when clicking your username in the nav-bar, an option to logout is displayed. If selected, the user is logged out of the application and returned to the landing page to login or signup.

User Story #17 - 4.4

As a user, I would like to be able to post anonymously.

Completed:

If a user chooses to make a post where they don't want their name to be shown as the author, they can use the anonymous button on the create post page. When the post gets added, no one, not even the author, can see a username attached to a post. When someone visits the author's page, the post is nowhere to be found.

User Story #18 - 4.5

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

Completed:

Like most social media apps, *Meliora* allows the user to delete any of their posts. They can do so by clicking the red trash icon on their post. To confirm this action is intentional, a pop-up screen will appear. When a user deletes a post, it will no longer show on the profile page.

User Story #19 - 9.1

As a user, I would like to be able to flag harmful/vulgar/offensive posts.

Completed:

If a user finds a post to be inappropriate for Meliora, they can choose to flag a post. On the postcard on the homepage, a little flag button exists so that any user may flag any post. On the backend, the number of flags a post has is checked. When a post reaches more than 5 flags, an attribute called “delinquent” is set to true in the database. In a later sprint, we will implement a way for admins of the app to review delinquent posts and decide if they are actually appropriate for the platform or not.

User Story #20 - 1.2

As a user, I would like to reset the password to my account if I have forgotten it so that I can always have access to Meliora and its features.

Completed:

When logging in, an option is displayed below the login form to reset your password if forgotten. When doing so, a link is sent to the user's email to reset the password. After changing the password, the account is updated so that the user will not be able to login with the old password, only the new one.

What did not go well?

As with all emerging teams, there was (at first) a lack of effective communication and clarification with low-level details surrounding some of the technologies used. Where and when to use certain techniques to help accomplish our user stories were not clear as such. This issue made itself known around the first time we all met after the start of the sprint to review each others' progress. After that first time meeting, much was cleared up and ready to go as far as implementation.

Another major issue we faced is that some of us were new to the tech stack that we had decided upon. Due to this, it took us some time to effectively implement the required features in a timely fashion. Moreover, since we did not have effective communication in the beginning, some of us who were not familiar with, say, Angular, could not get timely help from other members in the team, leaving us to learn the required syntax and structure ourselves.

Merging code was also a hassle at times. Whenever some of us worked on the same file, we got a few merge conflicts, which seemed hard to resolve, which occasionally, even produced a few errors. A related issue is that we merged our code to the main branch at the very end of the sprint. Ideally, this should have been done atleast once in about one or two weeks.

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?

We learned many things from this first sprint as a team. Firstmost, we realized the importance of keeping frequent communication with the team. In the first two weeks of the sprint, we barely updated each other on our progress and met less than once a week. When the third and fourth week rolled around, we had to meet almost every day in order to understand what others have done and finish parts of the project that required more than one person's work. This started to become stressful because it seemed like we had much to accomplish in very little time. Therefore, going forward, we could improve upon this by keeping more constant communication throughout the entire sprint and attempt to meet at least 2-3 times per week to check on each other's progress.

Another important change to make in future sprints is ensuring that our user stories are as detailed as possible. There were several elements not in the planning document that had to be implemented so that there was a link between features. For example, when implementing user story 9, we had to implement a temporary feed in the homepage so that a user could click on another user's username. While not too difficult of a task, it was a feature that we were not planning on completing in this sprint.

Finally, one major takeaway from this sprint is learning how to manage a version control system (VCS) better. Although most of our members were fluent with Git commands, one mistake our team made was not to merge all our code to the main branch until the end. Throughout the sprint, each of us worked on our own branches until the end. Due to this approach, we had to encounter several merge conflicts the day before our sprint demo when merging our code. Hence, as a better practice, we plan to merge our working code to main at least once a week in order to incrementally fix any bugs that may be the result of a merge.