

Purdue University

Sprint 2 Retrospective
MonsterChat Application

Team #14:

Rayten Arnold Rex, Cole Baughn, Vishal Gill, Zeyu Pan, Andrew Sytsma, Michael Young

1. What Went Well?

In Sprint 2, we were able to complete many of our user stories mentioned in the Sprint 2 planning document successfully. On Android, implementing most of the GUI and background code went fine, but the networking had issues that were mostly resolved. In the background for Android, reading and writing settings to a file went well; however, we found a better approach to handling the settings, so it more or less got thrown out. In the background for Android, reading and writing the chat and error logs went well after learning the nuances of reading and writing for Android. In the background for Android, creating and joining chat rooms also went well. In the GUI for Android, Settings went well as there is an easy method for saving and accessing settings. In the GUI for Android, creating and linking the pages went well because they were fairly easy to create and change. In the GUI for Android, displaying logs went well because it was similar to displaying messages from the previous sprint. In the GUI for Android, creating rooms went well because there is an alert box that can be repurposed for that use. For the iOS side, we were able to successfully create all the GUIs and be able to go back and forth between them without the use of a Navigation Controller, which was a limiting factor. For the iOS side, we were able to show all connections and chat rooms that were available for us to join. For the iOS side, we were able to implement the in app name and the time for displaying messages. For the iOS side, we were able to display the images sent and received in appropriate size. For the iOS side, we were able to view the messages and images in the order that the user sent with Tableview instead of Textview, which is a limiting factor.

We implemented the following tasks successfully:

- **As a user, I want to be able to change the color setting of the GUI because I want to customize my app experience (Android).**
 - We were able to create a settings page where the color setting could be displayed and selected. The color setting could be chosen and saved for when the app was reopened.
- **As a user, I want to be able to change the font setting of the GUI because I want to customize my app experience (Android).**
 - We were able to create a settings page where the font setting could be displayed and selected. The font setting could be chosen and saved for when the app was reopened.
- **As a user, I want to be able to change my “in app name” across sessions because it will allow me to remain anonymous (Android).**
 - We were able to change the username across sessions. The username can be changed on the homepage.
- **As a user, I want to be able to save my “in app name” across sessions because I do not want to have to type in my name everytime I use the app (Android).**
 - The username setting was saved each time it was changed. Each time the app reopened, the username was the same as the last username entered.

- **As a user, I want to be able to view my past conversation(s) because I would like to be able to access information that I may have forgotten.**
 - We were able to view past conversations by displaying them to the user. We wrote the past conversations to logs to keep track of them and read from them as necessary to display them to the user.
- **As a user, I want to be able to view chat rooms because I want to know what chat rooms are available to me (iOS).**
 - We were able to view all devices that broadcasted a session. Hence, we were able to view all available chat rooms to join.
- **As a user, I want to be able to create and join chat rooms because I want to be able to chat with other users.**
 - We were able to create a chat room and let others users join it or join a chat room. And once the connection was secured, we were able to send messages back and forth between devices.
- **As a developer, I want to be able to log errors because I can use the error logs to figure out why the application failed to work (Android).**
 - We were able to log errors to a file successfully. The errors are logged as necessary in our application and can be accessed if needed, but they are not displayed to the users in the list of logs as intended.
- **As a user, I would like to be able to enable and disable timestamps because I would like to know when a message is received but also have the option to disable it.**
 - We were able to create a settings page where the option to enable and disable timestamps could be toggled. The timestamps could be enabled or disabled and saved for when the app was reopened.
- **As a user, I would like to change timestamps between military and standard time because I want to see the time in different formats to customize my chat experience.**
 - We were able to change the timestamps between military and standard time through the settings page. This leads to the time being displayed in the corresponding format in the chat rooms and in the logs viewed by the user.

2. What Did Not Go Well?

Although we were able to implement many of our user stories, still there were some tasks that were unsuccessful. On Android, implementing the viewing of available chat rooms is buggy and only works sometimes; also, network connectivity problems lead to messages not always being sent and received. In the Android GUI, changing color and theme did not go well because we decided to not implement those setting specifically. In the Android GUI, changing the settings of the Settings Page did not go well as that page is a different type than the rest. For the iOS side, the biggest thing that did not go well is that we were not able to combine the GUI and networking with the background because our code was too independently written to function as a cohesive unit; the background had too much abstraction compared to the code associated with the GUI, and the GUI and the networking portion were too tightly intertwined for us to make a seamless transition from a prototype to a second version because of this a lot of components did not function, specifically the settings which required both the GUI and background to be on the same level of abstraction for it to work properly.

We did not implement the following tasks successfully:

- **As a user, I want to be able to change the color setting of the GUI because I want to customize my app experience (iOS).**
 - We were able to implement a label and corresponding view picker. But we were not able to populate the view picker to show every setting.
- **As a user, I want to be able to change the font setting of the GUI because I want to customize my app experience (iOS).**
 - We were able to implement a label and corresponding view picker for the font settings. But we were not able to populate the view picker to show every font and could not implement the code responsible for actually changing the font.
- **As a user, I want to be able to change my “in app name” across sessions because it will allow me to remain anonymous (iOS).**
 - We were able to have a string variable that is the user’s in app name. But we were not able to implement it to show up in the chat dialog.
- **As a user, I want to be able to save my “in app name” across sessions because I do not want to have to type in my name everytime I use the app (iOS).**
 - We were not able to save the string in a file. And we were not able to reopen it once the app was running again.
- **As a user, I want to be able to view chat rooms because I want to know what chat rooms are available to me (Android).**
 - We were able to view chat rooms, but they would disappear some of the time. If a user was not the server, then the user may not be able to view other chat rooms since the server did not pass along the chat room names.
- **As a user, I want to be able to change the theme of the GUI because I want to customize my app experience.**
 - We chose not to implement this user story. Theme was incorporated into Colors.

- **As a user, I want to be able to change the style setting of the GUI because I want to customize my app experience.**
 - We chose not to implement. It was decided that the style of the GUI should not be changeable.
- **As a developer, I want to be able to log errors because I can use the error logs to figure out why the application failed to work (iOS).**
 - We did not get around to implementing the error logs. We did not have enough time.
- **As a user, I would like to improve the networking because I do not want to have to deal with any issues that could be caused by poor network management.**
 - On the Android side, most of this was out of our control. The model used by WiFi Direct designates one device as the owner and the rest connect through it. This does not fit the model we are trying to achieve with a mesh network. While we could probably achieve the connectivity we wanted somehow, it would be for naught due to the entire network being disconnected by the group owner disconnecting.

3. What Should Be Improved?

During Sprint 3, there are ways we can work to improve ourselves as a group and as individuals. We need to work on branching more often with Github. We need to work on the project earlier in the sprint to avoid doing large amounts of work near the end. Also, we need to utilize the trello board more. iOS development needs to be less segregated to prevent issues with integration.

- To fix the issue of not branching enough with Github, we can become more educated about branching and actively choose to branch instead of just commit.
- To fix the issue of not working earlier on the project, we can start working earlier, so we can do little bits of work throughout the sprint to make it so not as much is done at the end.
- To fix the issue of not utilizing the trello board more, we can make sure to be more active with updating what we are doing on the trello board.
- The iOS developers need to communicate more often and be more aware of each others' code.