Red --> Not complete!
Blue --> In progress!
Green --> Complete, or very near completion!
US 01.01.01 -- > Just need to reflect in Profile - that part is in progress.
As a participant, I want to add a mood event to my mood history, each event with the current date and time, a required emotional state, optional reason, and optional social situation.
**The amount of work to do: Have a working mood history (functionality), corresponding date and time, entered emotional state, (optional) reason, (optional) social situation. Testing.
Login -- > main, recentEvent --> new_event
Event --> getDate() and getTime() , emotional_state, reason, social_situation
**Complexity : Working mood history, with it's current date and time functions. Working emotional state (String). OPTIONAL reason & social situation. Successfully adding this event works with
or without reason & social situation.
**Risk & Uncertainty: Mood history saved on Cloud per distinct user. Includes date/time, emotional state, and optional reason/social situation.
**Story points for US 01.01.01 = f(amount of work to do, risk & uncertainity, complexity): 5

US 01.02.01
As a participant, I want consistent emoticons and colors to depict and distinguish the emotional states in any view.
**The amount of work to do: Emoticon functionality for corresponding emotional states. Color options. Available in all views. Testing.
emotional_state, colors
**Complexity : All emoticons must be consistent with the users' chosen color, in their view. New event must abide this rule, and any
existing events/history.
**Risk & Uncertainty: Updating on cloud. Updating on old history events?
**Story points for US 01.02.01 = f(amount of work to do, risk & uncertainity, complexity): 2

US 01.03.01 -- > Viewing a mood event is not implemented yet (OnClickListener!)
As a participant, I want to view a given mood event and all its available details.
**The amount of work to do: All mood events are saved to addtodb(event). A UI that can display a list of these events, as well as
more information about the event if clicked on. Testing.
addtoHistory(event)
allEvents
**Complexity : View all information about any mood event.
**Risk & Uncertainty: Fetching information from cloud, update event(s), view details (emotional state, date & time, optional social situation & reason)
**Story points for US 01.03.01: 3
US 01.04.01
As a participant, I want to edit the details of a given mood event of mine.
**The amount of work to do: Check if mood event is login users', allow changes to be made to mood event (button EDIT),
update the cloud after saving changes. Testing.
date, time of update, emotional_state, reason, social situation. submit as new event.
**Complexity : Simple changes/update/push to db server.

**Risk & Uncertainty: Updating information to firebase, make sure we cannot edit other users' moods
**Story points for US 01.04.01: 2

US 01.05.01 -> Near complete, just needs to update with Firebase
As a participant, I want to delete a given mood event of mine.
**The amount of work to do: Delete button/integration into the UI - What view in the UI? Deletion of actual
event from the login users' profile. Testing.
delete Event[?] from AllEvent
**Complexity : Event deletion, update to db server.
**Risk & Uncertainty: More testing required to see if changes are pushed to db server. Relogin with users' username/
password and check after deletion.
**Story points for US 01.05.01: 3
Other Mood Details
US 02.01.01
As a participant, I want to express the reason why for a mood event using a brief textual explanation (no more than 20 characters or 3 words).
**The amount of work to do: Reason_EditText ---> Add a reason field. Check if that field has less than 20 characters,
OR three words. Testing.
reason, getReason()
**Complexity : Integrate reason into profile. Need to check the conditions are met.
**Risk & Uncertainty: Test to make sure conditions are met.
**Story points for US 02.01.01: 2
US 02.02.01 -- > In progress/Near complete
As a participant, I want to express the reason why for a mood event using a photograph.
**The amount of work to do: Integrate UI functions to use data/server image from users' phone/library. Testing.
reason_image
**Complexity : Integrate UI functions to add a image to a mood event. File size? Image size?
**Risk & Uncertainty: Upload image to database. Make sure it saves. Testing required for that users' profile.
**Story points for US 02.02.01: 4
US 02.03.01
As a participant, I want to specify the social situation for a mood event to be one of: alone, with one other person, with two to several people, or with a crowd.
**The amount of work to do: Add selection event for social situation (optional) for the UI (user get's to choose
between the 4 option(s)). Testing.
social_situation -- > alone, with one other person, with two to several people, with a crowd
**Complexity : Make sure the user can un-choose a situation if they have pressed on one, since it's optional.
Update to db server.
**Risk & Uncertainty: Mainly testing the conditions. Remember it's optional.
**Story points for US 02.03.01: 3
Profile
US 03.01.01

As a user, I want a profile with a unique username.

**The amount of work to do: Check username of user to all usernames' entered in the database, before login, after
the user presses login. If not unique, let user know. Testing.

username

**Complexity : Check the conditions. If not unique, let user know.

**Risk & Uncertainty: Mainly testing.

**Story points for US 03.01.01: 2

Mood History

US 04.01.01 --> Need to implement sorting

As a participant, I want to view as a list my mood history, sorted by date and time, in reverse chronological order (most recent coming first).

**The amount of work to do: Sort function for date / time, in both ways. Mood history View. Add "sorting" button
does either reverse/forward sorting depening on the amount of times pressed. (Once/Twice). Testing.

self_profile-->allEvents

**Complexity : Sorter functions pre-built into Java.

**Risk & Uncertainty: Testing with a large amount of history events

**Story points for US 04.01.01: 2

US 04.02.01

As a participant, I want to filter my mood history list to show only mood events with a particular emotional state.

**The amount of work to do: Filter function for emotional state, in the Mood history view - for the unique user. Testing.

emotional_state, allEvents

**Complexity : Make a filter function. UI implementation of a filter button/integration?

**Risk & Uncertainty: How will the filter look? filter for emotional_state entered. New view UI for filtered events.
Lots of testing required!

**Story points for US 04.02.01: 5

Mood Following and Sharing

US 05.01.01

As a participant, I want to ask another participant to follow their most recent mood event.

**The amount of work to do: Send notification to unique user. Accept/Decline UI integration, buttons, code. Testing.

getFollowerReq, getFollowing, follow_requests

**Complexity : Mainly uses the database server to communicate with other users' for follow requests.

**Risk & Uncertainty: Does following the mood event show it's details? How will the UI look for follow requests?

**Story points for US 05.01.01: 4

US 05.02.01

As a participant, I want to grant another participant permission to follow my most recent mood event.

**The amount of work to do: Check if there is a follower request, accept/decline the request. If accepted, send notification
to other user. Add UI integration. Testing.

follow_requests , getFollowerReq

*Complexity : UI integraion of follow granted, display mood event if follow accepted (unlocked)

**Risk & Uncertainty: testing required to make sure mood details are provided when the user accepts

the request only
**Story points for US 05.02.01: 3
US 05.03.01
As a participant, I want to view as a list the most recent mood events of the other participants I am granted to follow, sorted by date and time, in reverse chronological order (most recent coming first).
**The amount of work to do: Fetch database information for particular user followed/accepted. If accepted, add recent following mood events to list. Allow
sorting as per other Story. (already implemented). Testing
*Complexity : merge mood events with allowed follower users. sorting should be already implemented in the UI for sorting
events via date/time reverse chronological order
**Risk & Uncertainty: testing required since we are implementing multiple parts of the UI. slight uncertainty
**Story points for US 05.03.01: 4
Geolocation and Maps
US 06.01.01
As a participant, I want to optionally attach my current location to a mood event.
**The amount of work to do: UI integration of current location. Ask for permission. Upload to db. Testing.
location
*Complexity : Most likely a java function to attach location.
**Risk & Uncertainty: Not much, testing required. UI integration of how it should look, required.
**Story points for US 06.01.01: 2
US 06.02.01
As a participant, I want to see a map of the mood events (showing their emotional states) from my mood history list (that have locations).
**The amount of work to do: Filter out mood events that have locations, display using UI view, for history list. Testing
*Complexity : Filter function for map/location entered for mood Event. display using chronological UI implemented before.
**Risk & Uncertainty: Not much, except for the filtering.
**Story points for US 06.02.01: 2
US 06.03.01
As a participant, I want to see a map of the mood events (showing their emotional states and the username) from my mood following list (that have locations).
**The amount of work to do: Filter out mood events for following users (That have already accepted the follow), in the following list. testing.
*Complexity : Filter function similar to US 06.02.01. display using chronological UI implemented before.
**Risk & Uncertainty: Not much, except for filtering in the correct following list.
**Story points for US 06.03.01: 2