

Participant Login/Signup Screen

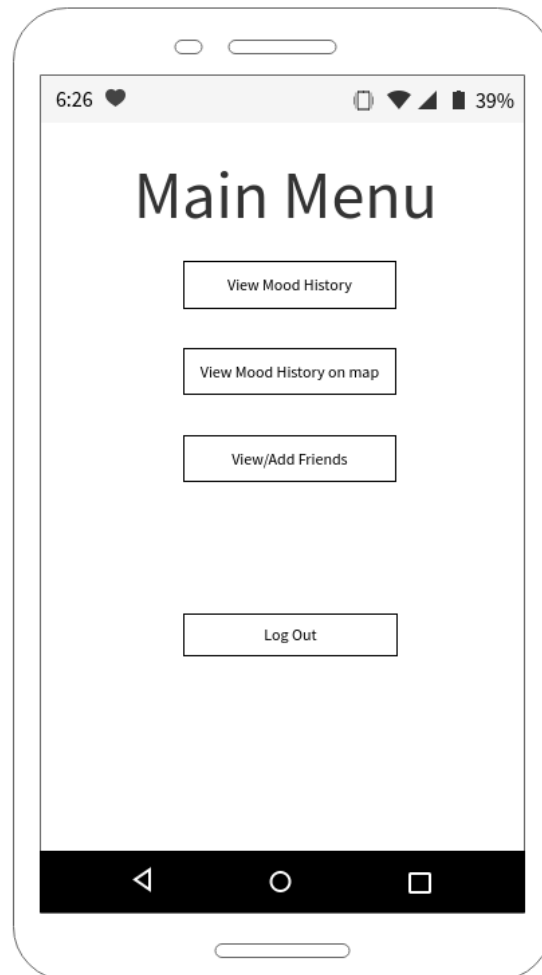
- Login screen is where participants can login to their accounts using a username and password.
- If the participant does not have an existing account then he/she can sign up by providing a unique username and password.



Comments:

The layout should contains the following Widgets

- An Image View displaying Logo (Project Name)
- An Edit Text view for Username
- An Edit Text view for password
- a button with Text "Sign Up" and a button with Text "Log In"
- After pressing Login Button and detected that participant entered wrong password, text should be displayed under password Edit Text in red that saying "Invalid password, please try again".
- After pressing Login Button and detected that participant entered username that doesn't exist, text should be displayed under username Edit Text in red that saying "Invalid username, please try again".
- After pressing Signup Button and detected that the username that participant entered already exist, text should be displayed under username Edit Text in red saying "Username already exists, please try another username".
- If Participant successfully logged in or signed up, should direct participant to Home Screen.

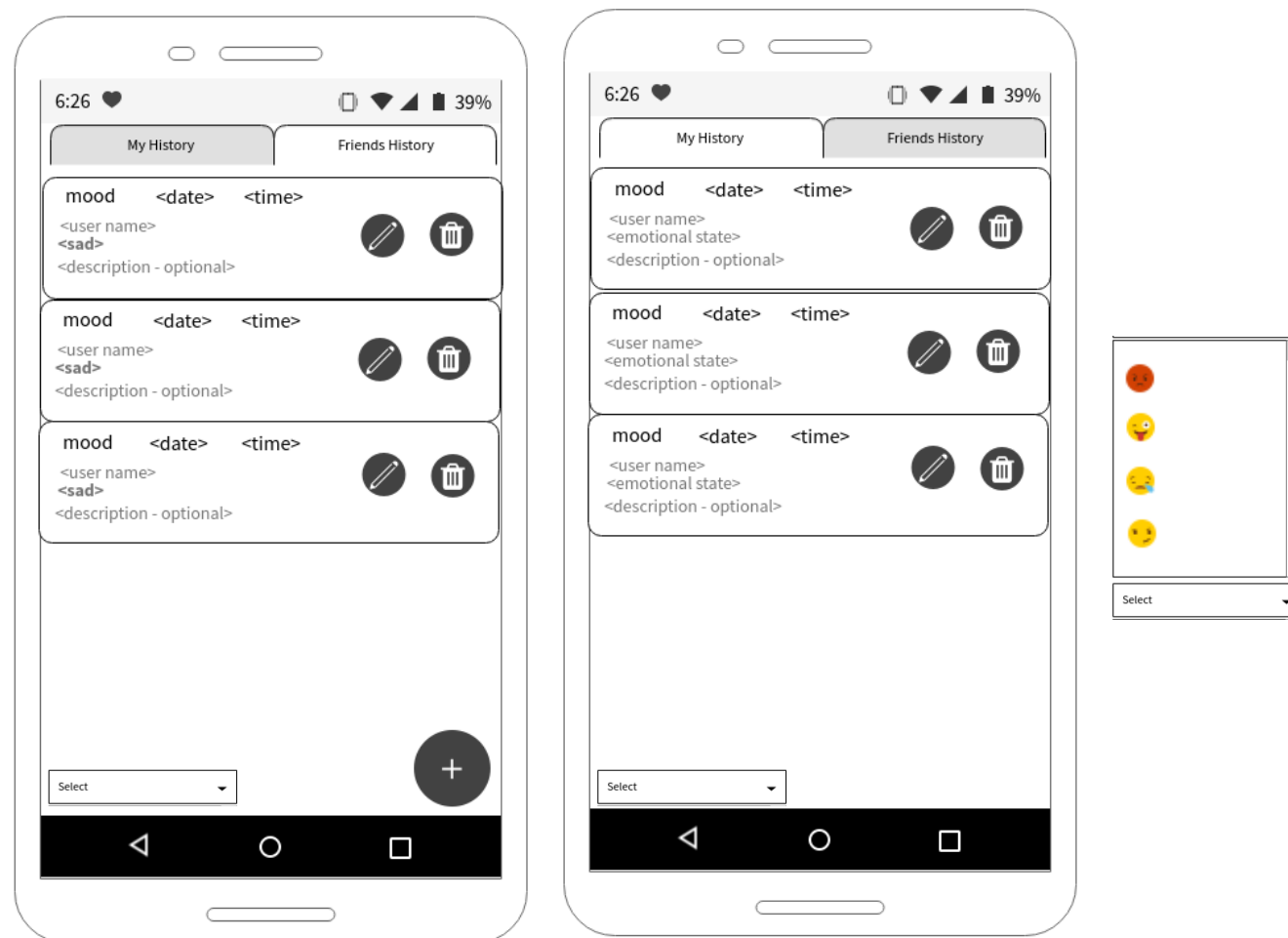


Home Screen:

- Buttons that directs the Participant to the MoodEvent History screen, Map View Of MoodEvent History screen, Add Friends & View Pending Friend Requests Screen. This page also is where the participant will log out from.

Comments:

- the layout should be in constraint layout which contains several buttons as follows:
- should contain a Button that when clicked, go To Add Friends & View Pending Friend Requests Screen
- should contain a Button that when clicked, Go To Map View Of MoodEvent History Screen
- should contain a Button that when clicked, Go To MoodEvent History Screen
- should contain a Button that when clicked, Log out the current Participant and return to the Participant Login/ Signup Screen



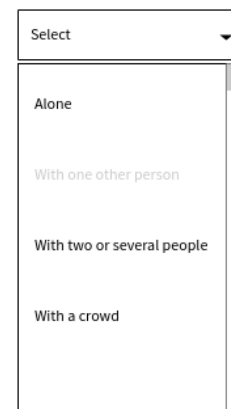
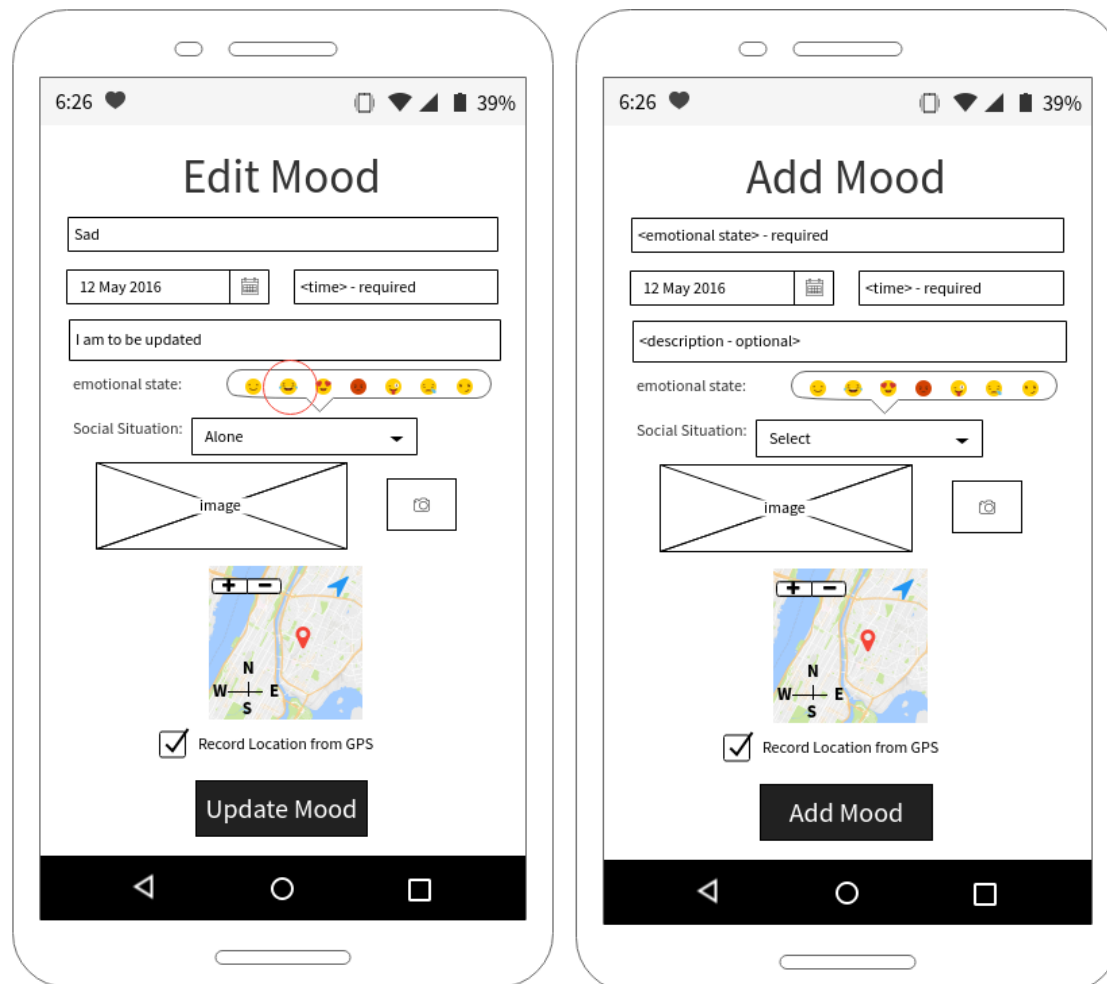
Mood Event History Screen

- This screen displays the participants MoodEvents as well as their Friends' MoodEvents sorted by date and time, in reverse chronological order (most recent coming first). On the participants on MoodEvent history they can also filter by emotional state. They can also edit/view or delete a mood from their history by clicking on the edit or delete button respectively. From this screen they can also add a MoodEvent by clicking the add button in the bottom right corner. For viewing the friends MoodEvent history it is only in a read-only state and cannot be modified by the participant.
 - Other participants on the app that have granted permission to the current participant of their MoodEvents.
- * **Mood Event:** An item that displays a participant's current emotional state at a specific time and date. It is optional to add a reason for the emotional state, an emoticon, a location of where it occurred, a photo related to the MoodEvent, and the social situation of the mood (ie. alone or with others)

Comments:

The layout should be in a tabbed layout with the following two tab:

- **First tab** should contain a RecyclerView which is used to display MoodEvent history of the current participant.
- A search bar above the RecyclerView so that when participant enters, only the MoodEvent with emotionalState value same as the participant's inputs are left on the screen.
- A Floating Action button below the RecyclerView which serves as a "Add MoodEvent" Button, when pressed, directs participant to Add/View/Edit MoodEvent Screen.
- Requires a RecyclerView list item xml which displays minimum information(date, time, emotionalState.) of a MoodEvent, An edit button, when pressed redirects the current participant to Add/View/Edit MoodEvent Screen. Also a delete button, when pressed deletes the MoodEvent from the current participant's MoodEvent history
- **Second tab** should contain a RecyclerView which is used to display MoodEvent history of the Friends of the current participant in read-only mode.
- Requires a RecyclerView list item xml which displays minimum information(date, time, emotionalState.) of a MoodEvent.



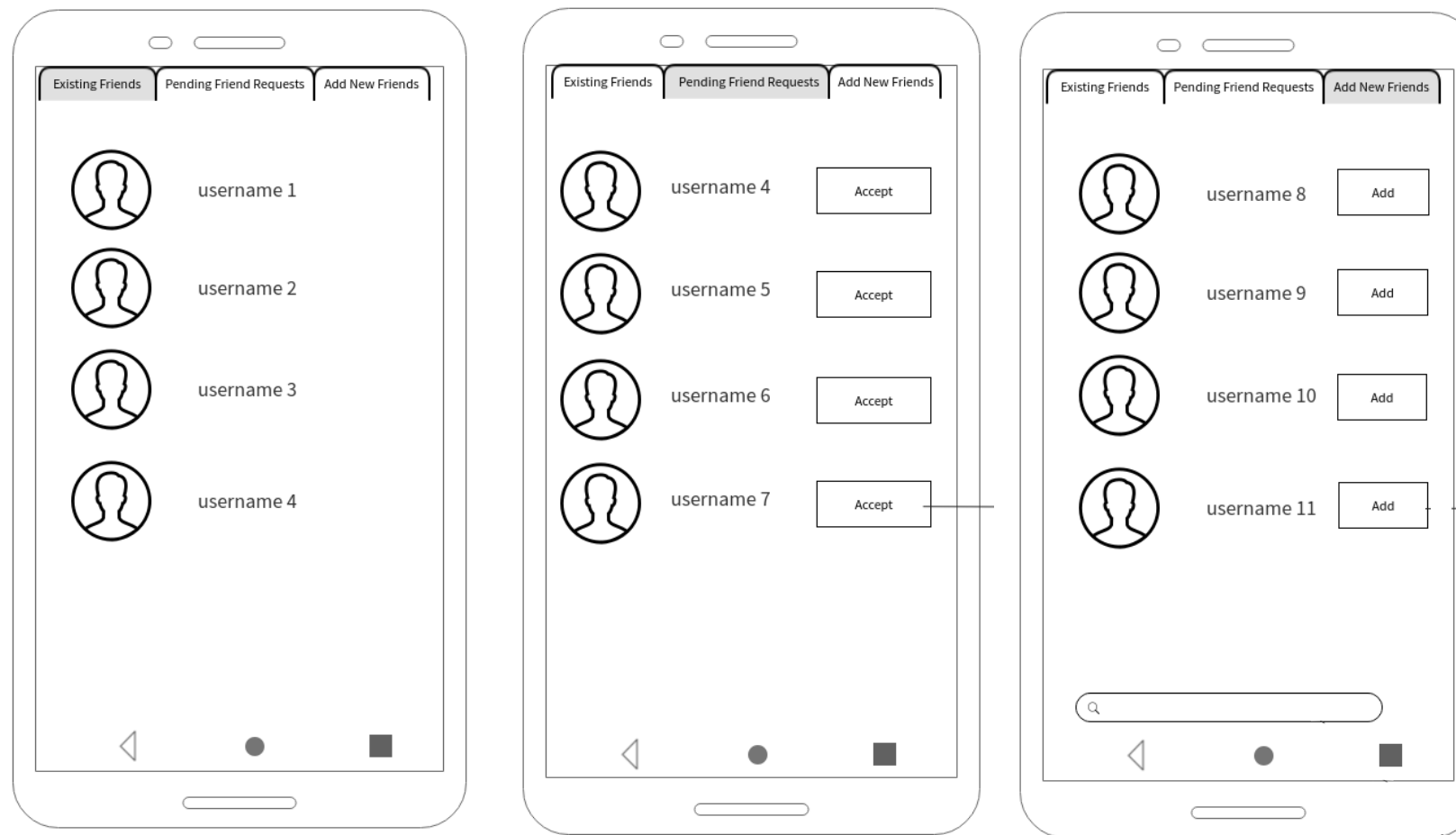
Add/View/Edit MoodEvent Screen

- This screen displays fields for the participant to input such as emotional state, date and time. Optional fields for the participants MoodEvent are adding a reason for the emotional state, emoticon, location and photo. This screen is used when a participant would like to add or view/edit a MoodEvent.

Comments:

The layout file should contain views as follows:

- Edit Text for date and its prompt
- a list of emoticons for emotionalState and its prompt
- Edit Text for reasonInText and its prompt
- an embedded camera view to allow participant takes photo to explain the reason of the mood event.
- an dropdown list for socialSituation; Values of the dropdown are : "alone", "with one other person", "with two to several ppl", "with a crowd".
- A CheckBox, when checked, retrieve geolocation where the MoodEvent takes place from GPS.
- An Embedded Map view that pinpoints the current location on Map.
- A Button, when participant is adding a new MoodEvent, has the text "Add"; When participant is editing an existing mood event, has the text "Update".
- Requires implementations that assists the participant in proper data entry. For example, use appropriate user interface controls to enforce particular data types and avoid illegal values.
- After pressing the "Add/ Update" button, the participant should be directed back to MoodEvent History Screen.

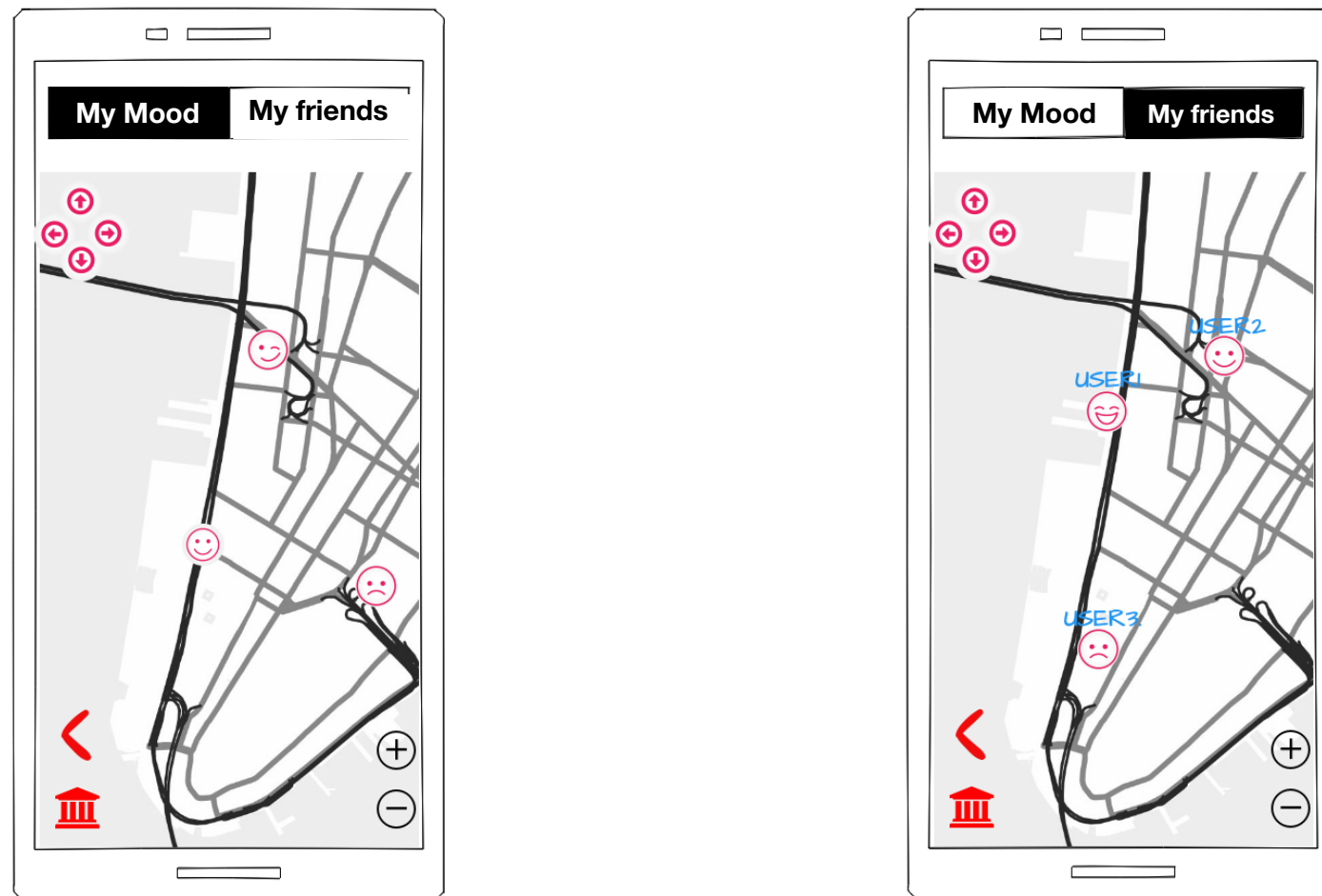


Add Friend&View Pending Friend Requests Screen

- On this screen participants can see who their friends are as well as add other existing participants and see pending friend requests from others. Participants can also add other participants by searching for their username.
- A message sent from the current participant to other participants on the application to request permission on viewing their History of MoodEvent.

Comments:

- the layout xml should contain a tabbed layout which contains three tabs: view existing friends tab, approving friend requests tab and view list of participants to follow tab. Should allow participants to switch among the three.
- RecyclerView should be implemented in each Tab.
- RecyclerView list item layout xml should be implemented for each RecyclerView.
- Specifically, for the RecyclerView that displays list of participants to follow, the list item should contain the participant's username and a button when pressed, sends a friend request to the participant.
- for the RecyclerView that displays existing friends of a participant, the list item should contain the participant's username.
- for the RecyclerView that displays pending friends request, the list item should contain the participant's username and a button, when pressed, grant access of most recent MoodEvent to the participant.



Map View of Mood Event History Screen

- On this screen participants can see either a map of our MoodEvents and where they were located. On a separate tab participants can also see their friends MoodEvent history and where they were located.

Comments:

- the layout xml should be in Tabbed Layout with two tabs. First tab should contain a Map View that displays participant's own MoodEvent on map by showing emotional state of every MoodEvent.
- A second tab should contain a Map View that displays MoodEvents from the following list by showing username and emotional state.
- a button at the bottom left corner which when pressed, directs participant to Home Screen.