Requirements Analysis Document Compliplenty Android Mobile App

Lloyd Williams

Justin Shiflett

John Moran

JP Lacap

Tyler Boettcher

Use Case: Log in

Name of the Project: Compliment app

Primary actor: User

Goal in context: To be able to log into the app

Preconditions: The user must have an account in order to Log in, Internet Connectivity

Trigger: The user is not logged in see the start use case

Scenario:

1. The table is displayed with the password and username
2. The user types in the username and password in
3. After the user have type in the username and password then the user is logged in.

Exception: If they type in the wrong password they would not be able to Log in

Priority: Being able to Log in to make an account

When Available: WK 1

Frequency: They will Login every day and if they want to save their password it will automatically Log you in.

Open issues: log in from different Devices or Log in on two different devices at the same time.

Use Case: Log out

Name of the Project: Compliment app

Primary actor: User

Goal in context: To be able to Logout of the app

Preconditions: The user have to be Login in order to Log out

Trigger: The user clicks the Logout button

Scenario:

1. Than after the user have click the Logout button
2. There is a message displayed on the screen
3. The message say are you sure you want to Log out
4. User is logged out and returned to the Login Screen (see Login use case)

Exception: The user clicks on the no box if they ask them are you sure you want to Log out

Priority: Being able to Log out of the Account

When Available: WK1

Frequency: They will Log out every day

Open issues: Log out from different Devices without being near them.

Use Case: New compliment

Name of the Project: Compliment app

Primary actor: User

Goal in context: If you don't like your compliment you have the ability to change it

Preconditions: The app can access what compliment they have already seen already

Trigger: Pressing the new compliment button so it can show the user a new compliment that the user has not seen before

Scenario:

1. The user than rates the new compliment (see React to Compliment use case)

Exception: The app can't generate the new compliment the users have not already seen before

Priority: High priority

When available: WK 5

Frequency: They will be able to get a new compliment if they click the new compliment button

Open issues:

1. Being able to get a new compliment without waiting a whole day to get the daily one.
2. Get different compliments suggested by friends

Use Case: Questionnaire Compliment

Primary actor: User

Goal in context: To be able to get a customize compliment that fits you

Preconditions: the app has to be able to access the database

Trigger: the user presses the Questionnaire compliment button on the main screen

Scenario:

1. The User fills out the questionnaire
2. User receives a personally generated compliment

Exception: if the user doesn't have the time to fill out the questionnaire

Priority: high priority

When available: Week 4

Frequency: User dependant

Open Issues: if the user doesn't like the compliment given, may feel like they wasted time filling out the questionnaire

Use case: Share Compliment

Name of the Project: Share Compliment

Primary actor: User

Goal in context: To send a compliment that you received to someone else

Preconditions: Received a compliment

Trigger: User hits the share compliment button under the compliment

Scenario:

1. The user chooses where he wants to share the complement to.

Exception: if the user doesn't want to share the complement

Priority: middle priority

Frequency: The user will be able to share any compliment received

Open issues: If the user doesn’t want to share the compliment and or doesn't like the compliment received

When available: Wk 10

Use Case: React to Compliment

Name of the Project: React to Compliment

Primary Actor: User

Goal in Context: To let the app know how you responded to the complement. To give feedback on the complement given

Preconditions: Received a compliment

Trigger: Receives a compliment

Scenario:

1. The user reacts based on a scale from 1-10

Exception: If the user doesn't want to react to the complement

Priority: Imperative for our app

Frequency: The user will be able to react to any compliment given to them

Open Issues: If our reaction options do not show how the user feels

When available: Wk 10

Use case: View Profile

Primary actor: User

Goal in context: Allow a user to view their profile details in an activity

Preconditions: App has their username in memory. Their account is stored in the database

Trigger: The user loads an activity that can navigate to their profile screen.

Scenario:

1. User: presses profile button
2. App: Loads existing information.
3. App: Populates fields of activity.
4. User: observes profile activity
5. User can edit Profile(See editProfile), view Friends List(see viewFriendsList), view liked list(see viewLikedList)
6. User: Return to previous screen

Exceptions: Lost Internet Connection: App doesn't check if Profile has been updated since last use.

Priority: Essential, must be implemented

When available: Week 2.

Frequency of use: Once upon creating account & whenever a user inquires. Semi-frequent

Open issues:

1. Should other ways to navigate to a user's profile be offered?
2. What details are on the profile?

Use Case: Edit Profile

Name of the Project: Edit Profile

Primary Actor: User

Goal in context: Allow the user to modify their bio, profile picture, and likes

Condition Requirements: Internet Connectivity (Through Wi-Fi or phone data)

Trigger: Pressing the “Edit profile” button on the Profile Screen

Scenario:

1. The user is taken to an activity that has a text box they can edit and a “Change Profile Picture” button that allows them to modify the picture associated with their profile.
2. The first textbox will be labeled “Bio” and will be displayed right below the user’s name when it is saved.
3. The user can tap the “Change Profile Picture” Button which will bring up a dialog box that has two buttons labeled “Take new Photo” (see Use Camera use case) and “Choose existing Photo” (see Open Gallery).
4. When user is done editing their profile, they will press the save button and be returned to the View Profile activity (see View Profile use case).

Exceptions: If the user does not have internet connectivity then the changes they made would not be saved and they would have to change their profile again when they had internet, so it would save.

Priority: Essential to have a profile to individualize our product.

Frequency: Dependent on the user

When available: Week 3

Open Issues:

1. Will we monitor profiles for bad language?
2. Will there be a way to report inappropriate profiles?

Use Case: Use Camera

Name of the Project: Take New Photo

Primary Actor: User

Goal in context: Allow the user to use their phone's camera to take a new picture to set as their profile picture.

Condition Requirements: Must have given the app permission to use the phone Camera

Trigger: Pressing the “Take new Photo” button in the pop when you click “Change Profile Picture”

Scenario:

1. The users phone camera app is opened.
2. The user can then take a picture that they wish to use as their profile photo.
3. When they take a picture, they will be asked if they would like to use that picture or take another one.
4. If they choose to use the picture they have taken, they will be taken back to the edit profile activity (see Edit Profile use case) and their profile picture will be updated.

Exceptions: If they haven’t given permissions to the app to access the camera, the app will display and toast informing them they haven’t given the correct permissions to the app.

Priority: Very important so our users can personalize their profile

Frequency: Dependent on the user

When Available: Week 5

Open Issues:

1. Should the picture be reviewed by moderators to make sure it's appropriate?

Use Case: Open Gallery

Name of Project: Choose Existing Photo

Primary Actor: User

Goal in Context: Allow the user to use a photo from their phones gallery as their profile picture.

Condition Requirements: Must have given the app permission to access media files

Trigger: Pressing the “Choose Existing Photo” button in the “Change Profile Picture” pop up

Scenario:

1. After pressing the “Choose Existing Photo” button, the users gallery is opened.
2. The user can then select one of their photos to use as their profile picture.
3. When they choose a photo, they will be returned to the edit profile (see Edit Profile use case) activity and their profile photo will be updated.

Exceptions: If they haven’t given permissions to the app to access the gallery, the app will display and toast informing them they haven’t given the correct permissions to the app.

Priority: Very important so our users can personalize their profile

Frequency: Infrequent

When Available: Week 5

Open Issues:

1. Can you monitor the picture they choose to make sure it is appropriate?

Use case: ViewFriendsList

Primary actor: User

Goal in context: Allow a user to view their Friends list in an activity

Preconditions: App has their username in memory. Their account is stored in the database

Trigger: The user navigates to an activity with access to their friends list

Scenario:

1. User Is on an activity that can navigate to friends list
2. User: presses list button
3. User: observes friends list activity
4. User: observes friend requests and can click to “Accept” or “Decline” the pending request.

Exceptions: Database is down: App loads available information.

Priority: Essential, must be implemented

When available: First increment

Frequency of use: Frequent, whenever a user inquires

Open issues:

1. How many friends are displayed
2. Is there a maximum number of friends
3. How will the list be sorted
4. Should other ways to navigate to a user’s friends list be offered?

Use Case: View Likes

Primary actor: User

Goal in context: Allows a user to be able to view previously liked compliments

Preconditions: App has memory of liked compliments. User has liked compliments in the past

Trigger: User: taps the ‘View Likes’ button

Scenario:

1. User: taps the ‘View Likes’ button
2. App: loads history of liked compliments
3. User can decide to leave the compliment upvoted, remove the upvote or downvote (see ReactToCompliment)
4. User can tap the compliment for sharing(see ShareCompliment)
5. User: return to previous screen

Exception: No compliments have been liked: activity is empty

Priority: Essential, must be implemented

When available: Week 4.

Frequency of use: Varies on user. Occasional

Open Issues:

Use case: ViewFriendsProfile

Primary actor: User

Goal in context: Allow a user to view the profile details of a friend in an activity

Preconditions: App has the username of the friend in memory. The account is stored in the database

Trigger: The user selects a specific friend from the friends list activity

Scenario:

1. User Is on main activity
2. User: Selects a friend's icon.
3. User can send message request (see sendInstantMessage), view user's likes (see viewFriendsLikeList)
4. User Returns to previous activity.

Exceptions:

1. Database is down: App loads available information.
2. Friend removed User: User is denied access

Priority: Essential, must be implemented

When available: Week 8

Frequency of use: Frequent, whenever a user inquires

Open issues:

1. What details are on the profile?
2. Should other ways to navigate to a friend's profile be offered?
3. What details are hidden?

Use Case: Remove Friend

Primary actor: User

Goal in context: To remove a friend

Preconditions: You are currently friends with the user and the screen is on their profile

Trigger: User: taps the ‘Remove Friend’ button

Scenario:

1. User: taps the ‘Remove Friend’ button
2. User: is shown a confirmation pop-up
3. User: taps yes
4. User: is shown a confirmed action screen

Exception:

1. You have no friends
2. The profile being looked at is not linked as a friend

Priority: Non-essential, not necessary for function

Frequency: Occasional

Channel to Actor: Via mobile device

Use Case: Find New Friends

Primary actor: User

Goal in context: To find a friend

Preconditions: You are currently not friends with the user and the screen is on their profile

Trigger: User taps other user’s profile

Scenario:

1. User: taps ‘Send Friend Request’
2. The user is shown an optional message box to say hi
3. The user taps ‘Send’
4. The user is shown a confirmed action screen

Exception:

1. The user and currently viewed profile are already linked as friends
2. The user is blocked
3. The profile being viewed is blocked

Priority: Essential, One of the main features of our app

Frequency: Frequent, users should use this feature as much as they wish

Channel to Actor: Via mobile device

Use Case: Send Instant Messages

Name of the Project: Send Instant Message

Primary Actor: User

Goal in context: To allow users of our app to openly communicate between each other and send text messages

Preconditions: Having people added in their friends list, and wanting to talk to them through our app

Trigger: User hits the Send Message button

Scenario:

1. User sees someone that they want to talk to.
2. They hit the Send Message button, and begin typing their message

Exception: If the user doesn't want to send messages to others

Priority: Being able to send messages to other app users

Frequency: The user will be able to send messages freely with other app users

Open issues: If the people receiving the messages truly want to connect and converse with the user sending the messages

When available: Wk 14

Use case: ViewFriendsLikeList

Primary actor: User

Goal in context: To view a list of his or her friend's likes.

Preconditions: User is friends with the other user and is on his or her profile.

Trigger: The user presses the "View Likes" option on the other user's profile.

Scenario:

1. User: user is redirected to a list that shows all of the other user's likes.
2. User: user will have option to share (view use case Share Compliment) or to react (view use case React to Compliment) to any compliment on the list.
3. User will observe option to remove this user from their friends list (see use case RemoveFriend)
4. User will also have a "Back" option that will return them to the previous screen.

Exceptions:

1. Users are not friends: If the user is not friends with the other user the "View Likes" option will be restricted.
2. Privacy settings: If the user has privacy settings that restricts friends from viewing his or her likes than the option will be unavailable.

Priority: Essential, must be implemented

When available: 6th week

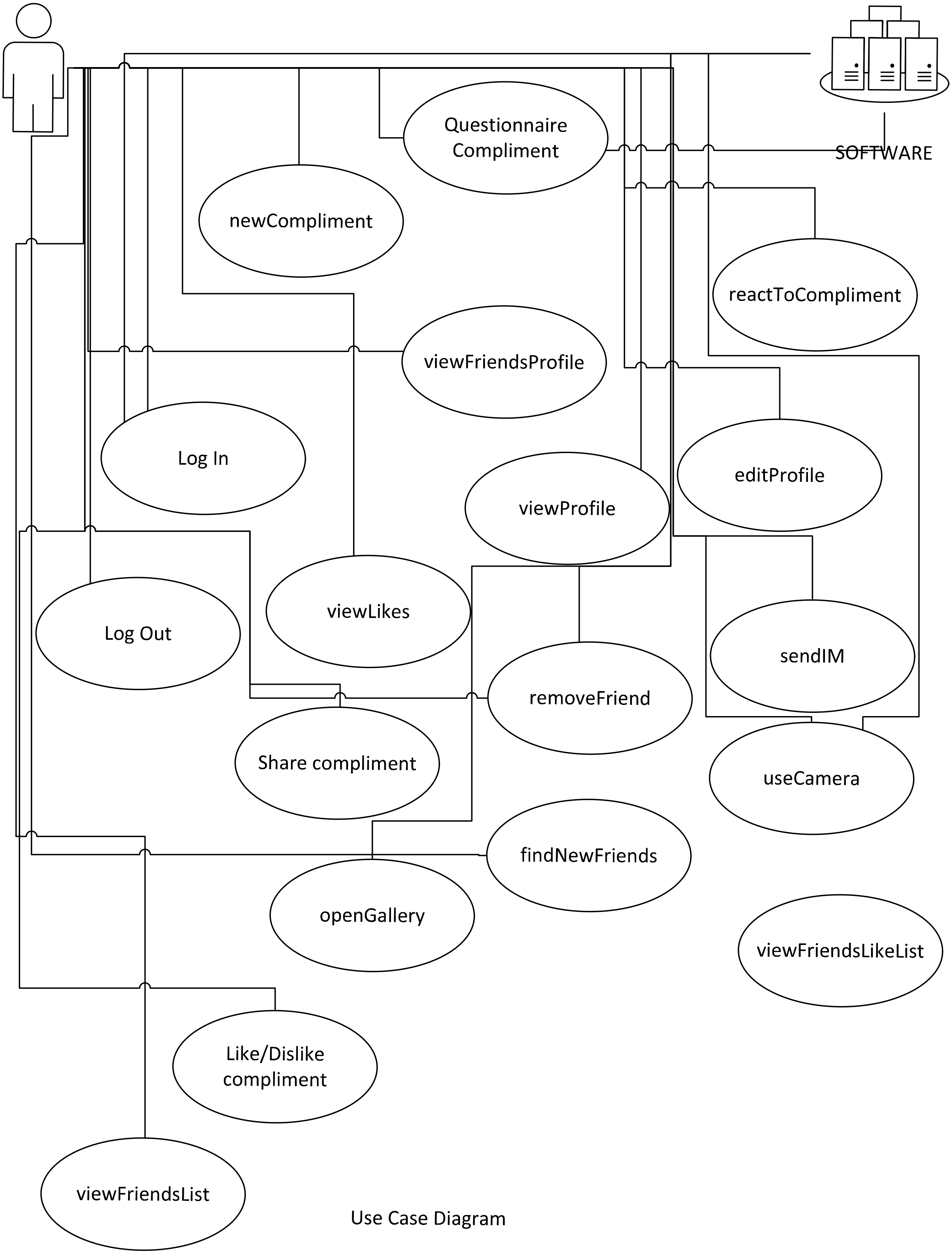
Frequency of use: Many times a day

Channel to actor: Via friend's profile

Secondary actors: Other users

Glossary:

- **Populates-** Fill in with relevant information
- **Like List-** List of all other users related to the primary user through friend relationship.
- **Friend-** another user who has approved access to view your accounts and the primary user to view theirs.
- **Complement-** Strings with positive connotation to be shared between users.
- **Gallery-** Phone’s storage of media to be accessed by the application.
- **React-** Feedback from user. Like or dislike.
- **Share-** Send a complement to another user through messaging services.



Use Case Diagram