Requirements Analysis Document: Compliplenty

Tyler Boettcher, JP Lacap, John Moran, Justin Shiflett Trey Williams, Anton Woody

Table of Contents

Functional and Non-Functional Requirements
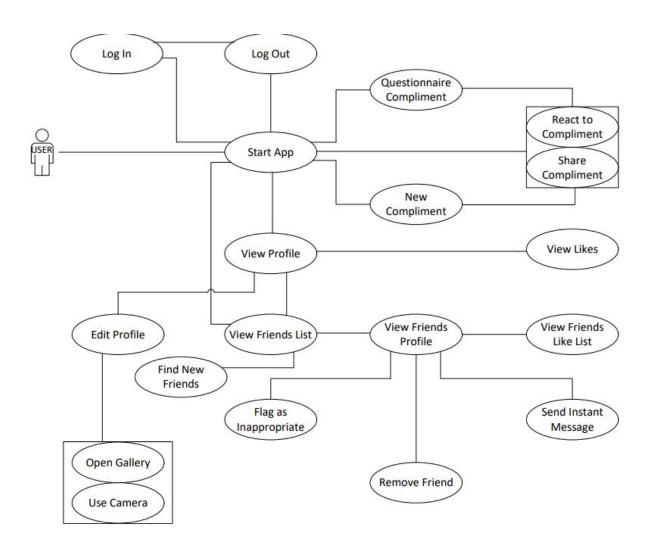
Non-Functional
- Facebook Access
- SQLite Databases containing all local comment and user information
- Fire based database with all users and comment information
- Decision tree for pinpointing which comment to display
- GPS

Functional
- User can change their profile pictures
- User can update their profile's Bio and Favorite Compliment
- User can generate a compliment based off the questionnaire
- User can find friends
- User can log in through Facebook
- User can upvote/downvote and share compliments
- User can flag other user's profile pictures as inappropriate
- User can send friends instant messages
- User can view other user's liked compliments
- User can receive a compliment of the day
- User can receive on demand random compliments
-

Use Case Diagram

Use Case: Flag As Inappropriate

Name of the Project: Compliment App

Primary actor: User

Goal in context: Flag profile pictures that are not appropriate

Preconditions: The user is on another user's profile

Trigger: The user clicks the "Flag as Inappropriate" button next to the profile picture that they are viewing

Scenario:

　　1.　The user is shown a dialog box that prompts them to confirm or decline the flagging.

　　2.　The user confirms the flagging and is returned to the view profile screen (see View Friends Profile use case).

　　3.　Database table is updated reflecting the number of times the flagged profile has been flagged; if they have been flagged three times the user loses profile picture privileges.

　　4.　The server reverts the user's picture to their facebook profile picture.

Exception: When the dialog box pops up, the user does not confirm the flagging: the user is returned to the profile (see View Friend Profile use case)

Priority: Low Priority

When Available: WK 13

Frequency: Low

Open issues:

Use Case: Start App

Name of the Project: Compliment App

Primary actor: User

Goal in context: Allow a user to access the functions of the App

Preconditions: Internet connectivity

Trigger: App is loaded or main activity is loaded

Scenario:

    1. Local database is updated, permissions are requested unless already provided

    2. User is shown the Compliment of the day, and buttons to access most other activities.

    2.   User decides on their intent and uses a function on the Main screen(see New Compliment, Questionnaire compliment, share compliment, react to compliment, view profile, view friends list, log out).

Exception: User is not logged in: User is sent to the login screen(see Log in use case)

No internet Connectivity: The local database is not updated.

Priority: High Priority

When Available: WK 1

Frequency: High

Use Case: Log in

Name of the Project: Compliment App

Primary actor: User

Goal in context: To be able to log into the user's account

Preconditions: The user must have a Facebook account in order to log in

Trigger: The app takes the user to the login screen when not logged in on start activity.

Scenario:

1. A form is displayed with the password and username text field and a "Login" button.

2. The user types in the username and password to log into their facebook account and clicks the "Login" button.

3. Information is sent to Facebook's API to verify the credentials

3. The user is logged in.

4. App navigates to the main screen (see Start App use case)

Exception: If they type in the wrong password: fields are cleared and user is told their information is incorrect, given another attempt.

Priority: High Priority

When Available: WK 7

Frequency: Very low

Open issues:

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 has to be Logged in order to Log out

Trigger: The user clicks the Logout button

Scenario:

1. User is prompted to confirm or decline their choice

2. User presses the "Yes" button, is logged out, and returned to the Login Screen (see Log in use case)

Exception: The user selects 'NO' box when they are prompted: User stays logged in.

Priority: Medium Priority

When Available: WK8

Frequency: Very Low

Open issues:

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 compliments the user has already seen.

Trigger: Pressing the new compliment button on the main screen

Scenario:

1.   The app searches the server for a compliment that the user has not seen in the current year and displays it.

2.   The user can react to the new compliment (see ReactToCompliment use case), like the new compliment to save it to his or her like list (see ViewLikesList use case), or share the new compliment (see ShareCompliment use case).

Exception: If the app would not be able to generate the new compliment the user will keep the original compliment.

Priority: High Priority

When available: WK 5

Frequency: High

Open issues:

Use Case: Questionnaire Compliment

Name of the Project: Compliment App

Primary actor: User

Goal in context: To be able to get a customized compliment that fits the user.

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 sees a question and buttons marked "Yes", "No" and "X".
2. The User fills out a decision tree of yes or no question.
3. The app searches localDB to find the most relevant compliment.
4. The selected compliment, based off their choices, is displayed on the main screen (see Start App use case).
5. The user can react to the new compliment (see ReactToCompliment use case), like the new compliment to save it to his or her like list (see ViewLikesList use case), or share the new compliment (see ShareCompliment use case).

Exception: The user presses the "X" button at the top of the questionnaire: the user is returned to the main screen (see Start App)

Priority: High Priority

When available: WK 2

Frequency: Medium

Open Issues:

Use case: Share Compliment

Name of the Project: Compliment App

Primary actor: User

Goal in context: To send a compliment that you received to another user

Preconditions: Internet Connectivity

Trigger: User hits the share compliment button under the compliment

Scenario:

      1. The user is sent to Facebook to share a compliment.

      2. The user shares the compliment in Facebook.

      3. User is returned to main screen(see Start App).

Exception: No Internet Connectivity: user is left on main screen(see Start App)

The user closes the Facebook app before sharing: The user is sent back to the main screen (see Start App)

Priority: Medium Priority

When available: WK 12

Frequency: High

Open issues:

Use Case: React to Compliment

Name of the Project: Compliment App

Primary Actor: User

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

Preconditions: Received a compliment

Trigger: The user presses the "Upvote" or "Downvote" button

Scenario:

>1. If the user presses "Upvote" the compliment is added to the user's like list (see View Like List use case).

>2. If the user presses the "Downvote" button, the user is prompted by a dialog box with response choice buttons, "Yes" or "No", and a question (see hard coded string 5).

Exception:

Priority: High Priority

When available: WK 11

Frequency: Medium

Open Issues:

Use case: View Profile

Name of the Project: Compliment App

Primary actor: User

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

Preconditions: Their account is stored in the database

Trigger: The user presses the view profile button (see Start App).

Scenario:

1. The user is shown their profile, which consists of their profile picture, a button to view friends list (see View Friends List), View like list (see View Likes), their favorite compliment, and Edit profile (see Edit Profile).

Exceptions: Lost Internet Connection: App notifies user that information cannot be updated

Priority: Very High Priority

When available: WK 7

Frequency of use: Medium

Open issues:

Use Case: Edit Profile

Name of the Project: Compliment App

Primary Actor: User

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

Preconditions: The server can access their profile

Trigger: Pressing the "Edit profile" button on the Profile Screen (see View Profile)

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 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).
3. 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:

1. The user loses internet connectivity: user is alerted that their changes will not be saved.
2. If a user's profile picture is flagged too many times: they will not be able to have a profile picture anymore

Priority: High

When available: WK 7

Frequency: Low

Open Issues:

Use Case: Use Camera

Name of the Project: Compliment App

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.

Preconditions: Must have the permission to use the phone Camera

Trigger: Pressing the "Take Profile Picture" button in the edit profile activity.

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. The user chooses 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: The user has not given the app permissions: the user is prompted to give the app permissions

Priority: Medium Priority

When Available: WK 9

Frequency: Low

Open Issues:

Use Case: Open Gallery

Name of Project: Compliment App

Primary Actor: User

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

Preconditions: 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: The user given permissions to the app to access the gallery: the user is prompted to give the app permissions.

Priority: High Priority

When Available: WK 9

Frequency: Low

Open Issues:

Use case: View Friends List

Name of the Project: Compliment App

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: User presses "View Friends List" button on the main screen or in their profile

Scenario:

1. User is shown a list of friends and users near them (see Find Friends use case).

2. User selects a friend's profile and is shown their profile (see View Friend's Profile use case)

3. User is shown new friend requests at the bottom of the list with an accept or decline button beneath each new friend.

Exceptions: You have no friends: only nearby users and pending friend requests are displayed.

Priority: High priority

When available: WK 8

Frequency of use: Medium

Open issues:

Use Case: View Likes

Name of the Project: Compliment App

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. App loads history of liked compliments
2. User can decide to leave the compliment upvoted, remove the upvote or downvote (see React To Compliment)
3. User can tap the compliment for sharing(see Share Compliment)
4. User returns to previous screen

Exception: No compliments have been liked: activity is empty

Priority: High priority

When available: WK 4

Frequency of use: Medium

Open Issues:

Use case: View Friends Profile

Name of the Project: Compliment App

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 the friend list(see View Friends List)

2. User selects a friend's icon.

3. User can send message request (see Send Instant Message), view user's likes (see View Friends Like List)

4. User returns to previous activity.

Exceptions:

1. Database is down: App loads available information.

2. Friend removed User: User is denied access

Priority: High priority

Frequency: Medium

When available: WK 8

Open issues:

Use Case: Remove Friend

Name of the Project: Compliment App

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 (see View Friends Profile use case)

Scenario:

1. The user is prompted to confirm they wish to remove the friend.
2. The user confirms they wish to remove a friend.
4. The friend is removed from User's friends list.
3. The user is shown a confirmed action dialog box.

Exception:

1. User: selects no when the pop-up appears: the user is returned to their friends profile (see View Friends Profile use case)

Priority: Low priority

When Available: WK 12

Frequency: Medium

Open Issues:

Use Case: Find New Friends

Name of the Project: Compliment App

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 from the list of users near them (see View Friends List use case)

Scenario:

1. The 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 cancels the friend request when asked to confirm: the user is returned to the Friends List (see View Friends List use case)

Priority: High priority

Frequency: Medium

When Available: WK 10

Open Issues:

Use Case: Send Instant Messages

Name of the Project: Compliment App

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. User hits the Send Message button
3. App prompts user with textbox and the user begins typing their message
4. User hits Send.

Exception: The user doesn't want to send messages to others: Use Case will not be used

Priority: Low Priority

Frequency: Medium

When available: WK 13

Open issues:

Use case: View Friends Like List

Name of the Project: Compliment App

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 is redirected to a list that shows all of the other user's likes.
      2.   User will have an option to share (view use case Share Compliment) or to react to any compliment on the list.
      3.   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: High priority

Frequency of use: Medium

When available: WK 6

Open issues:

Gantt Chart

Hard Coded Strings

	Use Case	String
1.	Flag As Inappopriate	"Flag this photo as inappropriate?"
2.	Log In	"Incorrect username or password"
3.	Log Out	"Are you sure you want to log out?"
4.	Share Compliment	"Where would you like to share this compliment?"
5.	React to Compliment	"Would you like a new compliment?"
6.	Use Camera	"Use the photo you have taken?"
7.	Open Gallery	"Use the selected photo?"
8.	Remove Friend	"Do you wish to remove this friend?"
9.	Find New Friends	"Confrim friend request?"

Table Layouts

Table 1: User table

Contains:

- Facebook profile picture
- Custom profile picture
- Facebook login information
- # times profile picture has been flagged

Table 2: Compliment table

- 366 compliments for random compliment of the day
- Holiday compliments(Halloween, Christmas, 4th of July, etc.)
- Compliment keys to be used for questionnaire
- How many times it was liked
- How many upvotes and downvotes each compliment gets

Glossary

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