Introduction for Flock

Flock is an application designed for planning events with friends you know in person. Its main feature is being able to create an event and invite friends or groups to that event. Once a user has accepted an event, they will be able to keep track of it.

First Iteration Requirements

Here is the list of requirements with their descriptions. More details are included on GitHub under the Iteration One Requirements milestone.

Task: Create the database

Responsible: Alex Cheng

- Create tables for Users, Friends, Groups, Group List, Events, Friend Requests, Event Requests, and the Guest List for the events.
- Add proper primary keys, foreign keys, and unique keys.

Task: Sign In

Responsible for Database: Chris Hauser

- The database will receive a username and password, from that, if the combination is correct the user will be logged in by storing a session. If the combination is not correct, a message with the error will return.

Responsible for GUI: Maryssa Crews

- A user will only be able to log in by entering their username and password. When they click sign in, a message will pop up saying an error occurred or a user will be redirected to the main page of the site.

Task: Sign Out

Responsible for Database: Chris Hauser

- The database will properly delete the stored sessions.

Responsible for GUI: Maryssa Crews

- A user will be able to click to sign out. From there, they will be redirected to the log in page.

Task: Sign Up

Responsible for Database: Chris Hauser

- The database will receive the username, first name, last name, email, and password.
- After the hashing the password, all the info will be stored into the database.

Responsible for GUI: Jordan Kayse

- A user will only be able to sign up only by entering their username, first and last name, their email, and the password and password confirmation.
- After making sure the two passwords match, we will send that information to the database side.
- The user will then be redirected to the main page.

Task: View Friends

Responsible for Database: Alex Cheng

- The database will receive a user id for the user logged in from the session. From that a list of usernames and ids will be returned for all the friends of the user that is logged in.

Responsible for GUI: Maryssa Crews

- On the main screen, a logged in user will be able to see the list of users they are friends with.

Task: Add Friends

Responsible for Database: Alex Cheng

- The database will receive the username of the person the logged in user wants to be friends with. With the userId from a session, these values will then be added to the Friend Request table. An error will be returned if the username doesn't exist.

Responsible for GUI: Jordan Kayse

- On the main screen, a logged in user will be able to click a button to add a friend. A popup will then appear asking for the username of who they want to be friends with.

Task: View Groups

Responsible for Database: Chris Hauser

- The database will receive a user id for the user logged in from a session. From that a list of the groups and the usernames and ids for those groups will be returned.

Responsible for GUI: Maryssa Crews

- On the main screen, a logged in user will be able to see the list of groups they have. When they click on the group, the list of friends in that group will appear.

Task: Create Groups

Responsible for Database: Chris Hauser

- The database will receive a user id for the user logged in from the session and the name of the group, and a list of user ids of people in that group from a post. The userId and the name of the group will be added to the Group Table, while the list of friends will be added to the GroupList table with the correct id.

Responsible for GUI: Maryssa Crews

- A user will be able to enter a name for the group and select friends that they want to add to that group. For this, we would need to be able to view the list of friends and select from that.

Task: View Friend Requests

Responsible for Database: Alex Cheng

- Getting the userid from the session, the database will get the list of friend requests from the friend request table. This list will then be returned.

Responsible for GUI: Jordan Kayse

- On the main page, a user will be able to view the friend requests for that user.

Task: Accept or Deny Friend Requests

Responsible for Database: Alex Cheng

- The database will receive the user id of the friend from a certain request and get the userid of the user logged in, from the session. If the request was denied, the request will be deleted from the table. If it was accepted, the pair of friends will be added to the database.

Responsible for GUI: Jordan Kayse

- When a user accepts or denies a certain friend request, it should be removed from the request table. If the request was accepted, the friend will also be added to the friends table.

Task: Create an Event

Responsible for Database: Chris Hauser

- The database will receive the userid from a session. From a post, the user will receive the name of the event, the description, and the times when this is occurring. This will then be added to the event table. From the list of friends, they will be added to the event request table.

Responsible for GUI: Jordan Kayse

- When a user wants to create an event, there will be a pop up that appears with all the information needed to create an event. There will be a section for the name, description, date and times. For the date there will be a calendar option. There will also be two options to add either friends or groups. They will need the list of friends and groups for that. Once each is selected, the names will be added to the invited list.

Task: Sign In for the Android

Responsible: Taylor Ellington

- The user will be able to sign in by entering their username and password they will then be redirected to the main page.

Task: Sign out for the Android

Responsible: Taylor Ellington

- The user will be able to sign out by clicking on a button and be redirected to the sign in page.

Task: Sign up for the Android

Responsible: Taylor Ellington

- The user will be able to create an account by entering the username, first and last name, the email and password. They will then be redirected to the main page.

Task: View Friends for the Android

Responsible: Taylor Ellington

- The user will be able to view all the users they are friends with.

Task: View Groups for the Android

Responsible: Taylor Ellington

- The user will be able to view all the groups they have. And when they click on a group they will be able to view the friends in that group.