Final Sprint 1 Planning Document

Team 18 - Roost

- ➢ Lisa Campbell
- ➢ Zhuoya Chen (Sophia)
- ➢ Joseph Khawly
- ➢ Daniel Piotti
- ➢ Andrew Ring
- ➢ David Worley

Sprint Overview

Basic Overview:
The goal of this sprint is to get a deployable application to iOS, Android, and web. The deployable app will have the basic functionality of user registration and sign in using Google's Sign In API, and we will store basic user settings in DynamoDB. The basic user settings will include a display name that can only be set when registering, whether to allow push notifications, GPS location, and activity radius.

Scrum Master: Lisa Campbell

Meeting Schedule: 6pm on Mondays and Wednesdays

Risks/Challenges:
Some of our team members have never encountered some of the technologies, languages or frameworks that we are using, which are React Native, C#, ASP.NET, DynamoDB, Heroku, Google Sign In API, Gifted Chat API, and Javascript. We also have never worked together before, so it will be a challenge to learn how to adapt to each member on the team in order to work cohesively. Most of us also have more than one exam during the sprint.

Current Sprint Detail

User Story #1
As a user, I would like to be able to choose whether I'm a new or existing user.

#	Description	Time (hours)	Team	Owner
1.)	Set up back-end server	8	Back	Lisa
2.)	Set up DynamoDB	5	Back	Joseph
3.)	Set up production server	5	Back	Lisa
4.)	Set up staging server	5	Back	Andrew
5.)	Set up web client	2	Front	Danny
6.)	Set up web routing calls	6	Front	Danny
7.)	Set up endpoints	10	Back	Joseph and Andrew

8.)	Set up web client templating	6	Front	David
9.)	Set up gitignore and git hook	3	Both	Lisa
10.)	Enable CORS on the front-end	3	Front	Danny
11.)	Create authentication page	4	Front	Sophia
12.)	Create endpoint for authentication page	3	Back	Lisa

Acceptance Criteria

- Given that the homepage is successfully implemented, when a new user views the homepage, then they should see a registration button.
- Given that the homepage is successfully implemented, when a returning user views the homepage that is not signed in, then they should be given an option to sign in.
- Given that a user is already signed in, when they view the homepage, then they will not have the option to sign in.

User Story #2

As a user, I want to be able to create a new user and automatically be logged in after registering

#	Description	Time	Team	Owner
1.)	Implement Google Sign-In API	5	Front	Danny and David
2.)	Create settings page	4	Front	David and Sophia
3.)	Create endpoint for saving settings	5	Back	Joseph
4.)	Save authorization status in DB	4	Back	Lisa
5.)	Save settings in DB	5	Back	Andrew
6.)	Create state container for user information	5	Front	Danny
7.)	Create error banner for rejected credentials settings page	2	Front	David

Acceptance Criteria

- Given that the sign-up is successfully implemented, when a user signs up, they should automatically be taken to the main screen.

- Given that the database portion of sign-up is successfully implemented, when a user signs up or registers, their information will be saved to the database.
- Given that a the sign-up has error handling, when a new user enters invalid registration information, an error banner will pop up explaining the situation.

User Story #3
As a user, I want to be able to login into my already existing account.

#	Description	Time	Team	Owner
1.)	Validate OAuth with Google Sign-In API	5	Front	David
2.)	Create endpoint for sign in	5	Back	Andrew
3.)	Query Dynamo for user info	5	Back	Joseph
4.)	Set user status in database to signed in	2	Back	Lisa
5.)	Reroute to explore page upon successful sign in	1	Front	Danny
6.)	Create error banner for rejected credentials settings page	3	Front	Sophia

Acceptance Criteria

- Given that sign in is successfully implemented, when a user signs in, they should be automatically taken to the main screen.
- Given that sign in has error handling, when a user tries to sign in with invalid information, there should be an error banner displayed to explain the situation.
- Given that the database portion of sign in is successfully implemented, when a user signs in, the status of the user is updated in the database.

User Story #4
As a user, I want to be able to logout of my account

#	Description	Time	Team	Owner
1.)	Set user status in database to signed off	1	Back	Andrew
2.)	Sign out using Google Sign In	4	Front	David
3.)	Create sign out button	2	Front	Sophia

4.)	Create sign out endpoint	5	Back	Joseph
5.)	Reroute to login page after sign out	1	Front	Danny

Acceptance Criteria

- Given that the sign-out is successfully implemented, when a user signs out, they should be taken to the login screen.
- Given that the database portion of this user story is implemented, when a user signs out, the user's status is updated in the database.
- Given that the user is successfully able to sign out through Google services, everything involving communication with Google's API and Google OAuth should be finished by this point.

User Story #5

As a user, I want to be able to See my legal rights as far as information and data usage permits

#	Description	Time	Team	Owner
1.)	Create legal rights popup	3	Front	Sophia
2.)	Route decline button	1	Front	Danny

Acceptance Criteria

- Given that there exists a legal rights pop-up, when a new user signs up, there will be a pop-up displaying their legal rights regarding information and data usage.
- Given that the legal rights pop-up actually pops up on the screen, when a new user clicks accept, then they will be able to use the app.
- Given that the legal rights pop-up actually pops up on the screen, when a new user clicks decline, then they will not be able to use the app and the app exits.

User Story #6

As a user, I want to be able to add profile information

#	Description	Time	Team	Owner
1.)	Add additional fields to profile settings	4	Front	Sophia
2.)	Add additional information to database	3	Back	Andrew

3.)	Query database for existing user data	4	Back	Lisa
4.)	Populate settings form with pre-existing data	2	Front	Sophia
5.)	Validate the user setting fields	1	Back	Andrew
6.)	Error banner if failure	2	Front	David

Acceptance Criteria

- Given that the profile information updating is implemented, when a user adds additional information it should be persistent and appear anytime the user's profile is accessed.
- Given that the user filled out the settings page, the app will automatically populate settings according to their previous information.
- Given that the user entered invalid input, a banner will appear notifying they did.

User Story #7

As a user I want to be able to be able to turn on and off push notifications app-wide for the group chats

#	Description	Time	Team	Owner
1.)	Create toggle element for notifications in user settings page	2	Front	David
2.)	Save push notification status in database	2	Back	Lisa
3.)	Enable push notifications (iOS)	8	Front	Sophia
4.)	Enable push notifications (Android)	8	Front	Danny

Acceptance Criteria

- Given that a user allows push notifications on iOS, the user can see notifications whenever a message pops up or a scheduled activity begins.
- Likewise, given that a user allows push notifications on Android, the user can see notifications about messages or activities.
- Given that the database portion is implemented, when a user toggles notifications, the push notification status should be updated appropriately.

Backlog

- As a user, I would like to
  - Join a group chat once I find an activity
  - Save favorite contacts
  - Search through popular random activities
  - Add activity to activity board with specified max group size
  - Filter activities by category
  - See amount of activities participated in for past month (if time allows)
  - See users interacted with in which activity for the past month (if time allows)
- As a group member, I would like to
  - Send the last 200 messages in a group chat
  - See all the users in the chat
  - Invite a particular person to the group chat
  - Leave a group chat when I'm no longer interested
  - Create polls to make group decisions with a selected duration
  - Participate in polls created in a group chat
  - View results of poll in a group chat
  - See how many users are in a chat
  - See who invited someone to the group if they decided to join
  - See that previously kicked users cannot rejoin the group chat they were kicked from
- As a group leader, I would like to
  - Be able to kick people out of the group
  - Close the group or not allow people to join/see on activity board
  - Delete the group
  - Be able to reopen a closed group
  - Have random teams selected if I need a team for my activity (if time allows)