

Sprint Report #4

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

Crowd Control - Group Management Mobile Application

Company

Bowtaps

Backlog

The following items/features were assigned at the beginning of the sprint, and worked on throughout its duration. It is broken down by week as such:

Week 1

- Android
 - Begin implementing Sinch
 - Create location and messaging views and managers
 - Design models and manager classes for messaging and location
 -
- Cloud Code
 - Group data parsing started

Week 2

- Android
 - Broadcast/receive messages to/from all members in a group
 - Create a layout for messaging
 - Create a MapFragment to display a map
 - Created buttons overtop the MapFragment to correspond to syncing and homing locations
- Cloud code
 - Leaving and joining groups handled
 - Checking existing email upon login (validation)

Week 3

- Android
 - Retrieve locations of group members, place their locations on the map via pins
 - Update group settings and data when changed
 - Update Group members if someone leaves or joins a group
 - Group messaging unit tests
 - GPS Location unit tests
- Cloud Code
 - Returning group information upon changes
 - Functional Group update indicator complete
 - Basic group functionality implemented fully (login/logout, join/leave groups, update on change)

Documentation and Business Plan work was carried out through all weeks of the sprint, and is ongoing.

Deliverables

During this sprint, these are the items/features from the backlog that were successfully achieved:

- Android
 1. Group Messaging
 - (a) Created a Layout
 - (b) Used Sinch code to create a service
 - (c) Implemented group messaging
 - (d) Group messaging is working with no known bugs
 2. Location
 - (a) Page layout created and linked from GroupJoin page
 - (b) MapFragment has buttons for homing and syncing group locations
 - (c) Retrieving the user's location on instantiation of the MapFragment
 - (d) User and group locations implemented
 3. Group update service
 - (a) Checks for updates in near real-time
 - (b) Updates group settings when changed
 - (c) Updates group members if someone leaves or joins
- Server (cloud code)
 1. Functional Group update indicator
 2. Returning group update information
 3. Join group function
 4. Leave group function

5. Check for Existing Email

- Misc/Transitional

1. Business Plan filled out, also a version tailored towards the Governor's Giant Vision contest
2. Documentation done inside and outside of the source code files

Issues and Changes

Some issues that we encountered include:

- Android

- Issues

- * Permissions to obtain contacts and locations from the device posed a challenge - still not handling the request gracefully
 - * Had difficulty implementing a custom AlertDialogFragment that extends DialogFragment, inside of other fragments such as MapFragment, GroupInfoFragment, etc.

- Changes

- * Added group update service - was not part of original backlog
 - * Added user location homing on MapFragment - was not part of original backlog

- Server (cloud code)

- Issues

- * Cloud functions improperly writing data.

- Changes

- * Added join and leave cloud functions - was not part of original backlog

Remaining Backlog

The following items/features remain either incomplete or need improvement for this sprint, and will carry onto the next sprint:

- Android

- Group messaging unit tests
 - GPS Location unit tests

- Cloud Code

- Testing on Group Functions.

Team Details

Here are some auxiliary details about our workflow and division of responsibilities, during the sprint:

Dan and Johnny started off Sprint 4 by working on messaging-related features, while Joe and Evan worked on GPS and map features. Nick focused on the business plan, updating the existing documentation to use the updated layout, and was tasked with installing the Fabric SDK.

For week two of the sprint, Johnny focused on group messaging. Dan and Nick also worked together on cloud code, targeting the leaving/joining groups, and a group update service in Android. Evan wrote a `LocationManager` class and stubbed out methods, that Joe wrote an interface for and made a UI for in the `MapFragment`.

Week three continued with Joe and Evan working on various location features, while Dan and Johnny worked on the update service and messaging features respectively. Nick focused on cloud code and business plan/documentation writing.

Additionally, this was the first sprint in which we had Dan serve as acting Scrum Master, to aid in organization and appointment of responsibilities. Though he did officially take this role on during our Winter Sprint (Sprint 3.5), most of us were either working remotely or unavailable, thus we were unable to fully utilize this new organizational change until now.