

# SwipeInvite

## Sprint 2 Planning Document

### **Team 16**

Andrew Davis

Kyle Krynski

Paul Ryan

Tejaswi Namuduru

Zening Chen

# Overview

The main objective of this sprint is to build up functionality allowing the user to create events in our application. The three major areas are 1) the user interface associated with event, 2) the client side background operations dealing with groups and events, 3) the server side background operations (Model and Controller) dealing with groups and events. By building these three areas up in tandem, all areas of the project will be able to keep up and react in real time to other systems, essentially evolving together. In order to keep a limited scope, this sprint will primarily focus on allowing users to create and manage events and people associated with specific will be notified with events information.

The SCRUM master for this sprint is **Kyle Krynski**. Meeting will be conducted on **Tuesdays, Thursdays and Saturday at 4:30pm.**

On the client side, the challenges associated with sprints are primarily associated with controller to deal with events associated with different kind of group(public/private). When user searching for groups when adding events, we should implement filter in listview to only show users public group result.

On the server side, we will need to handle the new event, group, and user types and fields that our created during the sprints. This will include adding handlers and interfaces as well as modifying older data to conform to new use cases and models.

# Details

## *Sprint 1 Leftovers*

User Story 1: As a user, I would like to be able to manage my personal account and information within the app.

- Tasks:
  - Implement the interface and associated activity into the application to allow the user to manage their personal profile information. (Zening Chen, 2hrs)
  - Implement the backend data management protocol to save the user object to the server and locally after an edit. (Kyle Krynski, 4hrs)

User Story 2: As a user, I would like to be able to create a group.

- Tasks:
  - Fix the backend as far as the saving of groups on the client side and syncing of groups with the server. (Kyle Krynski, 7hrs)
  - Display the list of groups to the user with a list view, and manage the upkeep of the list with refreshing from the server. (Paul Ryan, 5hrs)
  - Fix the server group objects to hold more of the required information for a group. (Andrew Davis, 4hrs)
  - Design the UI layout as it is shown in the scratch (Tejaswi Namuduru, 6hrs)

User Story 3: As a user, I would like to be able to add specific acquaintances or people to the group I am creating.

- Tasks:
  - Create the methods for polling the server for a list of users. (Kyle Krynski, 4hrs)
  - Create the methods for displaying the list of retrieved users. (Zening Chen, 3hrs)
  - Create the fragment for how the list of users will be presented. (Zening Chen, 3hrs)

User Story 4: As a user, I would like to be able to select and search which groups I would like to become a part of in the future.

- Tasks:
  - Create methods to poll the server and handle the responses. (Andrew Davis, 7hrs)
  - Create methods to apply those responses to the list view.(Zening Chen, 4hrs)
  - Implement Filter into Listview to show only public group information. (Tejaswi Namuduru, 6hrs)

User Story 5: As an organization leader, I would like to be able to find all people interested in my group.

- Will not be implemented.
  - We have decided that this functionality is slightly outside of the scope of necessary requirements for the app. This is on the boundary of a social networking thing, which can possibly be added if time allows.

## *New for Sprint 2*

User Story 6: As a group creator, I would like to be able to control who can modify the events and the members of my group.

- Tasks:
  - Add the currently made activity for editing groups to the app. (Paul Ryan, 5hrs)
  - Create the backend methods for polling the server and managing search results.  
(Andrew Davis, 12hrs)
  - Create notification in the Listview in the Message layout to help group members get updated event information.(Zening Chen, 2hrs)

User Story 7: As a busy person, I would like to be notified / keep track of near-future events simply.

- Tasks:
  - Design the activity for displaying a list of events to a user in the order of date.  
(Tejaswi Namuduru, 7hrs)
  - Design the fragments that go along with those list elements.(Zening Chen, 5hrs)
  - Design the methods for the user to interact with the list. (Paul Ryan, 6hrs)
  - Design the methods for the list to be refreshed. (Zening Chen, 4hrs)
  - Design the server calls to refresh the list from server data. (Kyle Krynski, 5hrs)

User Story 8: As a member of a friend group, I would like to schedule a group study session with my other friends and select the location.

- Tasks:
  - Allow a user to create an event to add to a group. (Tejaswi Namuduru, 6hrs)

- Create the activity that guides the user in filling out all of the required fields for an event object. (Zening Chen)
- Create the methods to give the correct users permission to see the event upon a server poll. (Andrew Davis, 8hrs)
- Create the methods to notify a user of a new event that they have been added to an event. (Kyle Krynski, 10hrs)
- Design the data model for an event. (Kyle Krynski, 2hrs)

User Story 9: As a user, I would like to be able to see and resolve conflicts in events that I choose to accept.

- Tasks:
  - Show the Toast to notify user there is a conflict. (Paul Ryan, 5hrs)
  - Will not allow the users to add conflict event to the listview unless the conflict is solved. (Paul Ryan, 8hrs)

User Story 10: As an introvert, I would like to reject an invitation to a party or outing in a non-confrontational manner.

- Tasks:
  - Allow a user to decide whether to accept or reject an event invitation, without the notification of other users (or at least with anonymity). (Kyle Krynski, 3hrs)
  - Design the UI Message layout with Reject and Accept button as it is shown in the scratch (Tejaswi Namuduru, 6hrs)
  - Design the methods for handling the user response in terms of where to store the event locally for later use. (Andrew Davis, 4hrs)

# Backlog

- Functional Requirements

- As a team leader, I would like to schedule time that every member in the team is available for meeting by checking and comparing their schedules.
- As a professor, I would like to find the best office hours that work for most students and notify all students in a manner such that they will actually address it.
- As a busy person, I would like to be notified / keep track of near-future events simply.
- As a member of a friend group, I would like to schedule a group study session with my other friends and select the location.
- As a person whose 21st birthday is in a week, I would like to invite everybody I know and know how many people are going to show up so I know how much cash to bring.
- As an introvert, I would like to reject an invitation to a party or outing in a non-confrontational manner.
- As a new college student, I want to get updated organization call out information I am interested in.
- As a college student living off campus, I would like to let all my friends in the area know that there is free food somewhere and for a how much time.
- As a business owner, I would like to be able to send promotional event information to possible local customers.
- As a working mother, I would like to send my calender to my paid babysitter to let her know when she has to take care of my child.

- As an organization member, I would like to be able to send organization activity information to students who are interested.
- As a professor, I would like to set exam and homework reminders to students in a colloquial manner.
- As a student, I would like to be able to sync all my courses directly into my calendar by simply connect it to my mypurdue account. (if time allows)
- As a user, I would like to be able to see and resolve conflicts in events that I choose to accept.
- As a non-profit Community manager, I would like to be able to see charity participants availability for simplified coordination.
- As a group leader, I would like to be able to schedule recurring events.
- As an event creator, I would like to be able to choose who sees how many people have accepted the event.
- As a non-smartphone user, I would like to be able to make decisions on events by text and have my electronic calendars get updated. (if time allows)
- As a group leader, I would like to save and load templates for events to send to a group. (if time allows)
- As a user, I would like to be able to change my account settings (password, email settings)
- As a user I would like to be able to invite non-app users via email or text message. (if time allows)
- As a user, I would like to be able to add and remove myself from groups specific to university clubs, organizations, project groups, etc. (if time allows)



- As a Google Calendar user, I would like to be able to add events to my existing calendar.
- As a user, I would like to be able to manage my personal account and information within the app.
- As a user, I would like to be able to create a group.
- As a organization leader, I would like to be able to find all people interested in my group.
- As a group creator, I would like to be able to control who can modify the events and the members of my group.

- **Non-Functional Requirements**

- Storage space: many of our users will probably be considered power-users, who want our app to be as space-efficient as possible. We can achieve this by following the thin-client model for our application. We will also save space on the server by deleting the info of events that have passed.
- Push notifications: We envision our users only using our application in two situations: creating an event and responding to an invitation, so we will have to notify users right away when they receive an invitation.
- Swipe to accept: most users prefer convenience over detail and we hope that this feature will be of high utility. A user will be able to receive invites and swipe to accept or decline an invitation without unlocking their phones. They will be able to view the details of the invitation (host, time, location) in the notification and decide if they want to accept or decline the invitation.

- Security: User person information, passwords, and schedules will need to be hidden from public view.
- Scalability: The main part of the system that will need to be scaled is the server, which will essentially become an algorithmic database challenge.
- Platform: This application should run on most modern Android smartphones, running Android 4.2 and above.
- Reliability: It is important that a user receives every event that they are invited to. We don't want our users to miss out.
- UI: The UI needs to be simple, beautiful, and easy to use. Users won't want to use an app that looks bad or is not intuitive.
- Performance: The client's response time is more critical than server response time. We don't want any freezing or crashing.