SwipeInvite

Sprint 3 Planning Document

Team 16

Andrew Davis

Kyle Krynski

Paul Ryan

Tejaswi Namuduru

Zening Chen

Overview

The main objective of this sprint is to finish events functionality and polish the user interface and user experience. The major areas that we will be focusing on are the views and controllers for event creation and invitation and polishing the views for viewing groups, events, and invitations. These areas will create an attractive and easy to use experience for the everyday user and allow us to market our application to our target market.

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 creating easy-to-use and intuitive interfaces for our users. We will need to build these with performance and visual appeal in mind.

On the server side, we will be handling new events and invitations for events. The challenge for this sprint will be handling the sheer number of events that will be created by everyday use. Also, we will be finishing up the backend model by writing code to save the model object to the server more frequently, as well as refresh the model from the server.

Details

Sprint 2 Leftovers

User Story 1: 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 2: 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, 3hrs)
  - Create the activity that guides the user in filling out all of the required fields for an event object. (Zening Chen, 5hrs)
  - 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)
    - Retrieving the current location by defining location service callbacks in the control(Zening Chen, 8hrs)

User Story 3: 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, 3hrs)
    - Will not allow the users to add conflict event to the listview unless the conflict is solved. (Paul Ryan, 6hrs)

User Story 4: 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 and 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)

New for Sprint 3

User Story 5: As a Google Calendar user, I would like to be able to add events to my existing calendar.

- Tasks:
  - Design the associated User Interface for syncing (Tejaswi Namuduru, 5hrs)
  - Write the code to interact with the calendar API provided by Google on all Android phones. (Paul Ryan, 5hrs)
  - Show the Toast to notify user if the event is successfully synced or not. (Zening Chen, 2hrs)

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:
  - Design the associated user interface for assigning different roles in the group members (Zening Chen, 3hrs)
  - Implement the control functionality to assign roles and push it to the server (Paul Ryan, 3hrs)
  - Design different view for different roles people in the group (Paul Ryan, 4hrs)
  - Add the associated functions to modify and check permissions (Kyle Krynski, 3hrs)

User Story 7: As a person invited to an event, I would like to simply swipe to accept or reject the invitation.

- Tasks:
    - Design notification element that detects swipe direction and reports the action to the controller. (Andrew Davis, 4hrs)
    - Implement the control functionality to add invite responses to the model and report decisions to the server. (Andrew Davis, 5hrs)
    - Implement background services to handle the server responses to these events in the background. (Kyle Krynski, 4hrs)

User Story 8: As a user who would like to schedule events and invite groups of people, I would like a visually appealing app which is easy to maneuver.

- Tasks:
    - General clean up of the look of the various views. (Andrew Davis, 6hrs) (Tejaswi Namuduru, 3hrs)
    - Design views in a way that adheres to the general theme of the user's phone. (Tejaswi Namuduru, 4hrs)
    - Convert many of the server requests to background services, if time allows. (Kyle Krynski, 5hrs)
    - Add the welcome page in the app when app starts (Paul Ryan, 3hrs)

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 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.
- As a person invited to an event, I would like to simply swipe to accept or reject the invitation.

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 2.3.3 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.