ACalendar Process Description

- Group Members:

  Qinrong Lin (snailin@cs.washington.edu)
  Qun Zhang (zq0605@cs.washington.edu)
  Ruoyu Mo (rettymoo@cs.washington.edu)
  Tong Shen (shent3@cs.washington.edu)
  Zhengyang Gao (gaoz6@cs.washington.edu)
  Ziyao Huang (ziyaoh@cs.washington.edu)

- Software Toolsets
  - Programming languages: Java and XML, since we are developing an Android app.
  - Data source: Amazon AWS -- the reason why we chose to use Amazon AWS is that several of our group members have worked with AWS before. Plus, Amazon AWS is known for its reliability and performance.
  - Version control: GitHub (https://github.com/CSE403DEATHEATERS/cse403-16au-ACal)
  - Library & API:
    - For the calendar, we plan to use the Google Calendar API.
    - For database interactions, we plan to use the Java Hibernate library.
    - For marking the locations of events, we plan to use the Google Maps API.

- Group Dynamics
  - Product Manager: Zhengyang Gao (gaoz6@cs.washington.edu)
    We chose Zhengyang Gao to be our product manager because we think he communicates with the team well and he can make a good balance between engineering and adding/deleting features. Furthermore, we trust that he will keep the team motivated and on schedule.
  - In terms of member roles, we are not going to designate specific members of the team as designers or testers, as all members will both develop and test their code. Furthermore, to make the software more robust, we will also designate members to test other members' code.
  - We try to create an open environment in the team -- if a disagreement arises between the developers, they can first bring it to the product manager. If the product manager cannot come up with a convincing argument to support/reject the proposal, we will try to bring in a third-party (could be other cs students or the mentors some team member had while interning) to evaluate the problem from an outsider's perspective and try to come up with a solution that satisfies everyone.

- Schedule & Timeline

- By Wednesday, Oct 19, each group member should have decided on:
  - Database representation
  - API's of each module listed below.
- Having decided on the database representation and the API's we are going to implement, we can start implementing the project. Specifically, each group member is assigned the following module(s) and the expected finish time is listed below (for release version 1.0):
  - User accounts & login (Ruoyu Mo, finish by Oct.24)
    - Create account
    - Forgot password (by verification code via email)
  - Event-related modules (event creation/edit/sharing, etc)
    - Calendar view of all events for a user (Qun Zhang, finish by Oct.24)
    - Event dashboard for each event (Tong Shen, finish by Oct.31)
      - Post proposals or updates(input box and display)
    - Event Pool (Ziyao Huang & Qinrong Lin, finish by Oct.27)
      - Searching an event
      - Marking an event on Google Maps
  - User Dashboard
    - Notification-sending module (Zhengyang Gao, finish by Oct.27)
      - In-app
      - Push notifications
    - Email-sending module (Tong Shen, finish before Oct. 24)
  - Friends-related modules (Ruoyu Mo, finish by Nov.5)
    - We support logging in using Facebook accounts.
    - Add friends who is also using our app by searching Facebook friends.
- While implementing the backend of our app, we will also be implementing a rough UI so that we can test our code as we go. After finishing implementing the backend, we will focus on beautifying/improving the frontend UI. We expect this to take about 2 weeks (finish by Nov.15th).
- After finishing the majority of our app, we will set up an extensive bug bash. We expect this to take less than a week (finish by Nov.20th).
- If everything goes smoothly, we will have about two weeks left in the end, which we can use to implement some of our stretch goals.

- Risk Summary
  - Since we are using multiple external libraries/API's, and since none of the group members have used the Google Maps or the Google Calendar API's before, we are most worried about getting these external components to work in harmony

with our app. Therefore, we will do extensive research on these external components to minimize the obstacles that we might run into in the future.

- Another major challenge we face is how to handle different permission levels. Specifically, in our app, there will be two kinds of events: private and public events. For private events, every participant can edit that event, whereas for public events, only the owner can edit that event. Therefore, correctly implementing the permission hierarchy is a challenge for us. We will try to get our original plan to work, but if we can't, we will try to implement an easier logic regarding permission checking.
- Kind of related to the last bullet point, on the front end, we would like to display different UI for different users (event owners vs. participants). None of us have implemented this kind of features before. We will do research on this both beforehands and as we go, to make sure that we will be able to deliver this feature.