Pocket Planner Teamwork Plan

Amanda Liu (ajliu), Everardo Rosales (erosales),
Samantha Briasco-Stewart (erosolar), Sin Kim (kimsin98)

November 13, 2015

Stakeholders

Event Organizers

These are our primary users. We envision our app to be useful to everyone from parents organizing birthday parties for their children to fans organizing large-scale conventions.

Event Attendees

These might only use this app to respond to invitations, but they are still significant because the app’s core features include tracking attendance and contacting attendees.

Tasks

The set of tasks will be divided into 3 categories: User Interface, Server, and Database.
Each of these categories will have one or two people in charge as well as a helper in the event that there is only one person in charge. The person in charge will be responsible for the bulk of the tasks in that section but will also assign the work to other team members as needed. Below is an outline of the tasks per category as well as a list of those responsible for the category.

These tasks have been divided in such a way where we are confident that the individual tasks should take no more than 1 or 2 hours. The majority of them should take no more than 30 minutes.

User Interface (MVP)

Captain: Amanda, Aux: Everardo

User Accounts (3hr total)

- Create login page/dialogue
- Create a dashboard for a logged in user displaying all of the events they have created
- Create event button and functionality in dashboard
- Create event deletion button and functionality in dashboard
- Create event edit button and functionality in dashboard

Event (2hr total)

- Create event dialogue box for inputting the basic required information for the event - name and time.
- Create special category button to add a category to an event and functionality
- Create the side panel with the event summary information
- Create an editing interface for the side panel summary information

Categories (.5hr total)

- Create a way of inputting the name of the category while creating it
- Create button at bottom of category to add To-Do to it

To-Do (1hr total)

- Create a way of inputting the text of the To-Do while creating it
- Create button for marking To-Dos as complete
- Create button for deleting To-Dos

Attendance (1hr total)

- Create attendance marking page (with buttons that just console.log for now)
- Create way to give link to planner
- Create attendance statistic on event page

User Interface (Final)

Captain: Amanda (tentative)

Event (1hr total)

- Create interface for adding email address / inviting people to the event

- Create interface for crafting notification/update emails (subject + body)

Categories (3hr total)

- Create way to move To-Dos from one category to another (preferably by dragging and dropping)
- Create edit button for category names
- Figure what defaults should be in place (from user testing -> poke db and server people to write the defaults into the db)
- Create a way customize defaults depending on type of event [stretch goal]

To-Do (.5hr total)

- clickable todos that make it into an input field

Attendance (1hr total)

- Create an invitation acceptance interface

UI Styling (min(float(“inf”), time_left)hr total)

- Show the app to people and have them try using it and get and record feedback
    - Change the interface as a result of the feedback
- Achieve CSS master level by improving the overall layout and look of the application

Server (MVP)

Captain: Sin, Aux: Everardo

User Accounts (4hr total)

- Create easy methods to verify user account and login information using third party systems (potentially **hard** if we decide to use Node.js and don’t find an easy to use package to integrate these APIs)
- Add access restrictions to the event edit page
- Create a create event method that forwards to database write methods
- Create a delete event method that forwards to database write methods
- Create an edit event method that forwards to database write methods

Event (1hr total)

- Create an easy way of committing summary information changes to the database (poke db person)

Categories (1hr total)

- Store created categories in the proper Event item (package and forward to db)

To-Do (2hr total)

- Store created To-Dos in the proper Category item (package and forward to db)
- Create mark-complete to-do method that forwards to database
- Create delete to-do method that forwards to database

Attendance (1hr total)

- Add functionality to store attendance markings (package and forward to db)
- Add functionality to pull attendance from database (package and forward to db)

Server (Final)

Captain: Sin (tentatively), Aux: Everardo (tentatively)

Event (7hr total)

- Add access restrictions to private event details page (call db check)
- Add verification method of invitees viewing event details page (call db check)
- Impose restrictions on how often update emails can be sent out
- Add functionality to add an attendee to an event (package and forward to db)

- Create email forwarding service to push notifications / updates by email to invitees

Categories (2hr total)

- Add functionality to move To-Dos from one category to another (within an event) (package and forward to db)
- Add functionality to rename categories (package and forward to db)

To-Do (1hr total)

- add functionality to edit To-Dos (package and forward to db)

Attendance (3hr total)

- Create email bot for invitations
- Create method to generate invitation links that map to person + event
- Create a method to write this change to the database
  - Update users vs Update attendees (possible a list of users and then just call .count() on the query object to get the number of attendees)
- Create way of checking that the person marking attendance hasn't done so already (look up in list of attendees). Otherwise retrieve attendance record.

Database (MVP)

Captain: Samantha, Aux: Everardo

User Accounts (4hr total)

- Create User database (with applicable schema)
- Possibly create a User table to integrate third party login system
- Add verification of access restrictions convenience methods (for event edit pages)
- Create a event write to db method
- Delete event from db method
- Create a event edit in db method

Event (3hr total)

- Create Event database (with applicable schema)
    - Contains Categories (create “schema” for categories)
        - Contains To-Dos (create “schema for to-dos)
    - Contains Attendance (create “schema” for attendance)
- Create an easy way of writing summary information changes to the database

Categories (1hr total)

- Create method to store created categories in the proper Event item in db

To-Do (2hr total)

- Create method to store created To-Dos in the proper Category item in db
- Create method to write label To-Dos as complete in db
- Create method to delete To-Dos from db

Attendance (1hr total)

- Create method to write attendance markings in db
- Create method to pull attendance from database

Database (Final)

Captain: Samantha (tentative)

Event (2hr total)

- Create access restrictions check for private event details page
- Add attendees database (per Event)

- Add attendee write to db
- Create method to query for email recipients (attendees related)

Categories (2hr total)

- Add functionality to move To-Dos from one category to another (within an event) and commit this change to the db
- Add functionality to rename categories write to db

To-Do (1hr total)

- add functionality to edit To-Dos

Attendance (3hr total)

- Create method to generate invitation links that map to person + event
- Create a method to write this change to the database
  - Update users vs Update attendees (possible a list of users and then just call .count() on the query object to get the number of attendees)
- Create way of checking that the person marking attendance hasn't done so already (look up in list of attendees). Otherwise retrieve attendance record.

Calendar

Minimum Viable Product (due 11/23)

User Interface

Category	(Tentative) Done by
User Accounts	11/18
Event	11/20
Categories	11/22
To-Dos	11/22
Attendance	11/22

Server

Category	(Tentative) Done by
User Accounts	11/17
Event	11/19
Categories	11/19
To-Dos	11/20
Attendance	11/20

Database

Category	(Tentative) Done by
User Accounts	11/19
Event	11/17
Categories	11/20
To-Dos	11/20
Attendance	11/20

Final Version (due 12/7)

User Interface

Category	(Tentative) Done by
Event	11/26
Categories	11/28
To-Dos	11/27
Attendance	11/27
UI Styling	12/7

Server

Category	(Tentative) Done by
Event	12/5
Categories	11/27
To-Dos	11/26
Attendance	11/29

Database

Category	(Tentative) Done by
Event	11/28
Categories	11/28
To-Dos	11/28
Attendance	11/28

Risks

User Risks

The most important risk we must address is keeping event plans private to the event organizers. We plan to mitigate this by using a strong third-party authentication system and ensuring that all pages related to event planning are accessible only to authenticated users' requests.

The other main risk to our app is misuse of attendance links. Every link must be single-use and applicable only to the invitee who received it. Our solution is to generate unique attendance links by assigning unique ID's for every event and invitee.

Implementation Risks

Difficulty integrating third party account login would be one of our more pressing issues should we have a hard time finding an effective API to use. To mitigate this issue we have allotted plenty of extra time so that the person in charge of implementing this has extra time to implement, if needed.

We might also have a difficult time making the email bot to forward updates and other notifications to the attendees of an event. To mitigate this issue we have allotted plenty of extra time so that the person in charge of implementing it has extra time to implement, if needed.

Since we have segmented the work at different levels as opposed to concepts (ie UI, Server, Database vs. User Accounts, Events, etc) we have created a situation where it is possible that the entirety of one person's work depends on the entirety of someone else's - and if that someone else doesn't get their work done, this could be a big problem. We plan to mitigate this by having frequent check-ins and redistributing tasks and their priorities accordingly.

Minimum Viable Product

Core concepts included in MVP

User Accounts

- Users should be able to create accounts (optionally with third-party services, such as Facebook or Google)
- Users should be able to log in to accounts and access events they have created

Event

- Users should be able to create events
- Users should be able to add Categories to their Events
- Users should be able to edit the summary of their Events

Categories

- Users should be able to create a category
- Users should be able to create and assign To-Dos to a category

To-Do

- Users should be able to create To-Dos for events that they are planning.
- Users should be able to mark a To-Do completed.
- Users should be able to delete To-Dos (completed and uncompleted).

Attendance

- Event planners should be able to have an invitation link to send out (by themselves) for their event.
- Invitees should be able to click the provided link and mark their attendance as yes or no.
- Event planners should be able to see how many people have responded yes (out of how many invited).

Features postponed to final version

Event

- User should be able to create private events whose details are only visible to invitees.
- Event planners should not be able to spam invitees as a form of advertising.
- Users should be able to invite people to their Events

Categories

- Users should be able to move To-Dos from one category to another.
- Users should be able to rename a category.
- Default Categories (Venue, Food) should already exist.

To-Do

- Users should be able to edit a To-Do.

Attendance

- Event planners should be able to send unique attendance links to all invitees by email.
- Invitees should be able to click the link in the sent email and mark their attendance as yes or no.
- There should be a system in place to prevent attendees being double counted.

UI Styling

- Ensure that the user interface is learnable through some user testing.
- Make the application visually appealing.