

Teamwork Plan

Caroline Chin, Rebecca Corcillo, Kwame Efah, Abraham Quintero

Stakeholders

Our target user is anyone who organizes or attend meetings

- This application is designed to facilitate the planning of small events or meetings (approximately 5-15 people). We realize that the application can be used to organize many different kinds of events--large events (where it is difficult to find times that everyone is available), parties (where people come and go and rarely stay the entire duration of the party) etc. Our application will focus on the use case of organizing small events and meetings that occur within a specific time range.

Tasks and Timeline

Task	Team Member	Expected Effort	Deadline
login / logout	Rebecca	1 day	Nov 16
Availability backend	Rebecca	3 days	Nov 20
Set up server and make workflow for deploying code	Abraham	2 days	Nov 18
Retrieve and display the user's google calendar	Kwame	2 days	Nov 18
Store user availability	Kwame	1 day	Nov 19
Meeting creation, (including user invites, duration and time range preference)	Rebecca	1 day	Nov 21

algorithm to determine "In"	Rebecca	1 day	Nov 20
Not in MVP:			
email notifications	Caroline	2 days	Nov 25
User general preferences.	Abraham	1-2 days	Nov 25
Retrieve user information on pending and scheduled meetings	Rebecca	1-2 days	Nov 25
Edit meeting attendee preferences	Kwame	2 day	Nov 25
UI Elements (in MVP):			
Make login/logout UI page	Kwame	1 day	Nov 20
User overview page	Kwame	1 day	Nov 21
Meeting creator page	Abraham	1 day	Nov 19
User general preferences	Abraham	1 day	Nov 20
Selecting preferences	Caroline	5 days	Nov 20

Risks and Mitigation

- **Implementation risks:**

- Hard to integrate with google api
 - Ensure that all team members must use some part of the API in their portion of the project so that everybody understands the tools used to communicate with Google.
- Google Calendar API does not have all the functionality we expect
 - Write our own methods to compensate for the functionality we expected to have

- **User Experience risks:**

- After specifying his/her time preferences, a user may set up additional events before an "In" is created. In this case, preferences that the application stores for the user are not longer up to date and the created "In" will cannot take the user's updated schedule into account.
 - We will not address this risk in the minimum viable product, but for our ideal product, we will be able to get information about updated events from google calendar and take these into account when determining an "In".
- Rescheduling an event may require all the invitees and the host to input their preferences again or require additional communication between participants
 - We will not address this risk in the minimum viable product, though we can mitigate this by keeping track of the preferences each user has inputted in the past so that in most cases, they will only need to make minor changes
- Users who do not use google calendar to schedule events will need to enter information manually
 - There is little we can do if a user does not use google calendar, but we can streamline the user interface to make the process of selecting availabilities easier

Minimum viable product

Features to be included

- Login/Logout using Google
- Meeting creation
- Platform for users to describe their preferences based on their Google Calendar
- Algorithm for determining the optimal meeting time based on user preferences
- User Information on pending and finalized meetings
- Set up server and workflow for deploying code

Concepts to be included:

- “In”
 - An “In” is at the center of our product, so it must be included in the final product. An “In” will be created when there is a time range of the desired duration for which all users are free. Otherwise, an “In” will be created that maximizes the number of users who are available during the time.
- Preferences
 - A user will be able to set availabilities and occupied times on a user interface that overlays a google calendar.

Justification for Minimum Viable Product

Provides real value to users

- Our application will provide real value to users as they will be able to use the basic functionality of easily entering their meeting time preferences and using an algorithm that determines the optimal meeting time automatically. This is substantially better than having to move information manually from a calendar application to a scheduling application

Provides opportunity for feedback

- As the main idea and value of our application will be implemented in the minimum viable product, we will have the opportunity to use feedback on the most essential components of our application.

On path to full product

- The main components that need to be implemented after this point are email notifications, and retrieving, modifying and storing prior (non-overarching) preferences.

Issues postponed

- Trying to maintain up to date representations of our users calendars.

- If one user takes much longer to fill out their form than everyone else, then the availability data for the rest of the group might be stale- they might have scheduled new events and we wouldn't know about it.
 - This is a small issue that current market leaders like Doodle do not attempt to address, so we think it is safe to postpone this until a later time.
- Priority Level:
 - We plan to allow users to specify priority levels on specified time ranges so that they can indicate whether they would strongly or weakly prefer meetings to be scheduled at a certain time.
- Email notifications
 - We plan to notify all users regarding the “In”, when an optimal meeting time has been determined
- Prior Preferences
 - retrieving, modifying and storing prior preferences to make it easier for users to reschedule meetings
- General preferences
 - A user will be able to set general preferences (ie. no meetings before 8am, after 9pm etc.)