Pencil Me In
Caroline Chin | Rebecca Corcillo | Kwame Efah | Abraham Quintero
Design Document
11/13/15

Design Overview

Motivation

Brief description of system to be built

- The system will allow a group of users to find mutually available times based on their google calendar schedules.
- A host user will create a meeting on PencilMeIn and invite others to the event using their gmail accounts.
- Each user will be presented with availabilities determined by their google calendar. Users can choose to modify their availabilities according to their preferences. (See wire frames for a more detailed explanation)
- The webapp then determines a meeting time that matches as many participants' schedules and preferences as possible

Key purposes (what problems does it solve? why should it exist?) , Each purpose summarized in a short sentence and then explained

- To find the optimal mutually available time among a group of users with minimal manual user input
    - why should it exist / problems that it solves:
        - The current methods for determining free time for a group of people requires a significant amount of work from each user.
        - Most scheduling applications also require the creator of the event to manually determine the times for which all users are available.
        - This process creates many opportunities for human error and is extremely inefficient.
    - explanation:
        - Automating this process will reduce both the human error problem and the inefficiency problem to create a better and more integrated user experience for setting up meetings
- To allow users to set priorities on different events if a meeting must conflict with an existing event
    - why should it exist / problems that it solves
        - Some scheduling applications allows for setting preferences for availabilities. In these cases, the creator of the event normally manually determines the optimal time for an event based on these preferences
        - In other cases (for example, in When2Meet), a participant has no way of specifying that they can miss an event in their calendar if necessary
    - explanation
        - The application will allow users to specify whether they are willing to miss or reschedule existing events in their calendar if necessary. This will allow for ease and flexibility in scheduling an event with many people.

- To allow users to accurately represent their schedules with minimal work
    - why should it exist / problems that it solves
        - For some people, their google calendar accurately contains all the events that a user will attend in a given day. For others, the calendar describes only some some events users plan to attend
        - For our scheduling application to work, we should have an accurate representation of the events users plan to attend so that it has the correct information when finding mutually available times.
    - explanation
        - PencilMeIn should allow users to easily provide information of their actual availabilities to more accurately represent their schedules

Deficiencies of existing solutions (if relevant)

- Overview of Existing Solutions
    - The current workflow for scheduling a meeting between several people consists of Three main steps:
        - The meeting host creates a doodle, whenisgood or an event on a different scheduling platform and gives all of the invitees a set of choices for when the meeting will be held
        - Each invitee has to manually input their availabilities for the specified time ranges of the event, referencing their calendars to see when they are free.
        - The invitee responds to the host.
    - We believe that there is a better way to do this since the invitee's calendar is most likely already in a machine readable format. Why not have the scheduling app take their google calendar into account?
- Deficiencies of existing solutions
    - Email
        - Cannot integrate with personal calendar
        - Poor organization of day of week, date and time
    - Doodle, Congregar, and WhenIsGood
        - Cannot integrate with personal calendar
    - Outlook/Company-Wide Scheduling
        - Not used outside of large companies

Design essence

Concepts

List of key concepts with brief definitions

- In:
    - Purpose:

        - To find an optimal mutually available time among a group of users with minimal manual user input
    - Operational Principle:
        - After all users submit their preferences for times, the app then intelligently schedules a single optimized “In”, a time range during which all users are available
- Preference
    - Purpose:
        - To allow users to accurately represent their schedules
    - Operational Principle
        - When a user is invited to a meeting, they can click and drag on an interface overlaid over their google calendar to specify their available and occupied time ranges.
- Squeeze:
    - Purpose:
        - To allow users to prioritize existing events so that a meeting can be scheduled even when there are no mutually available times among all participants
    - Operational Principle:
        - A user indicates that they can miss a commitment in their calendar if it is necessary to allow all users to attend the meeting. While the algorithm is scheduling the event, a squeeze is created if it can increase the number of people who can attend.

Security concerns

- Security Policy:
    - Only allow access with a google account.
    - Account recovery is done through google.
    - We do not store any event titles from users, only ranges that they are unavailable
        - This will limit the information that we can leak.
    - A user should not be able to see another user’s availability; they should only be able to see the “In” that the application has scheduled for the whole group.
- Threat model:
    - Assumptions:
        - Google accounts are secure; attackers cannot compromise google accounts.
            - Google is known to have a very secure infrastructure.
            - There is nothing we can do to protect a user’s information once an attacker has compromised their google account.
        - Attacker will not have physical access to servers:

        - If the attacker has physical access to the machine, there is very little we can do to protect against attacks such as reading off of the ram.
- Mechanism:
    - Use OAuth 2 to authenticate with google using Google's nodejs npm packages.

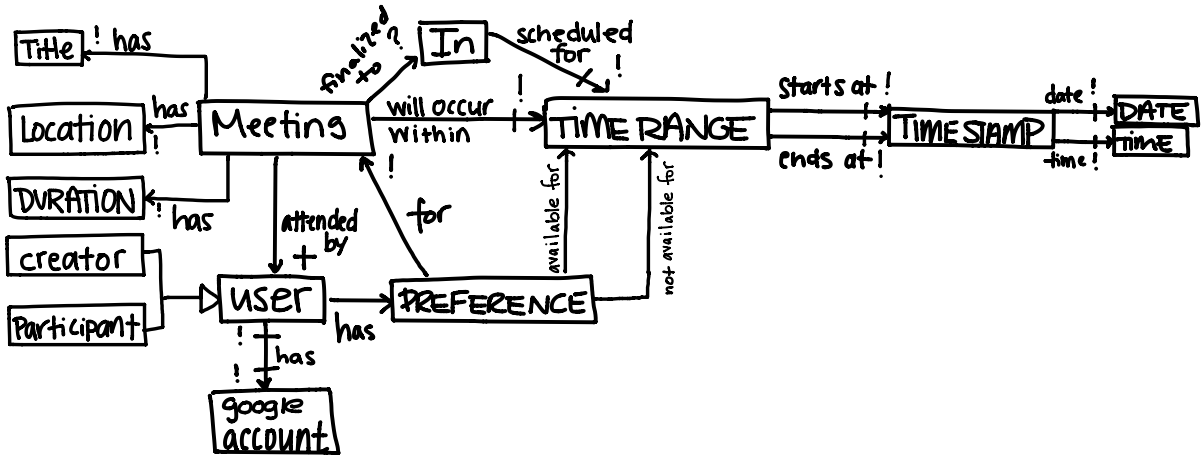
Standard web attacks:

- We're protecting against web attacks by using a standard framework (node.js with express and mongodb)
- Protect from xss by sanitizing inputs
    - The application requires very few text inputs:
        - Location
        - Meeting title
    - Use security reviewed code to sanitize input:
        - santize html npm plugin based on Google caja code that has been code reviewed and is widely used https://www.npmjs.com/package/sanitize-html
- Use CSRF tokens to stop CSRF attacks:
    - Using csurf express module
        - https://github.com/expressjs/csurf
- Use google libraries to interface with google api to protect against OAuth replay attacks:
    - google-auth-library
    - https://developers.google.com/google-apps/calendar/quickstart/nodejs

User Data Model and Interface/WireFraming

Pencil Me In

Data Model



EXPLANATIONS

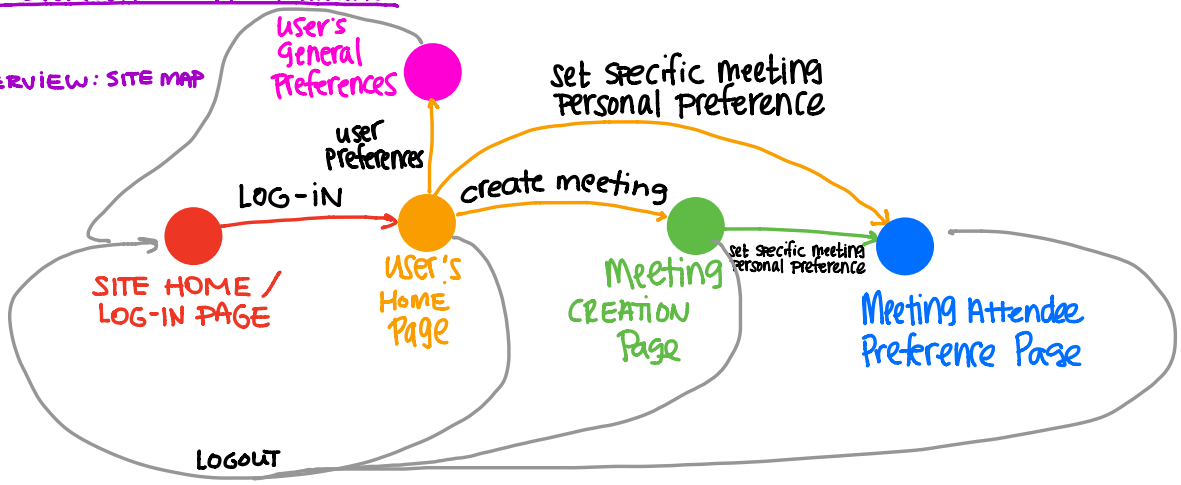
- → A meeting is what our app is trying to schedule
- → The meeting creator sets up the meeting with a location, duration and time window in which the meeting can actually occur.
- → The algorithm looks at information for users involved in this meeting and schedules it for a single "In"
- → An "In" is an optimally scheduled time range for a meeting. We chose to make an In its own object to highlight the fact that it was optimally scheduled.
- → Users can set up preferences for each meeting, which consists of time ranges they want to exclude/re-include for the scheduling algorithm
- → Users log in with their google account which imports their google calendar information which implicitly tells the algorithm which time windows are free/taken before being possibly manually overwritten.

INSIGHTS

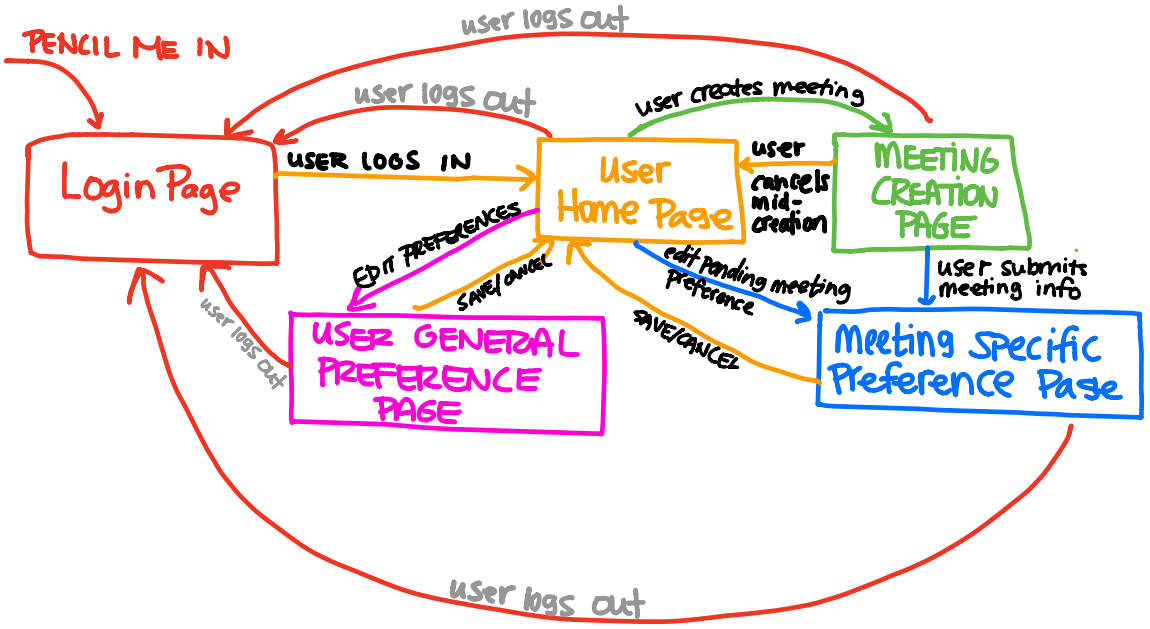
- → If there is no avaible overlapping time that everyone can attend, our algorithm will schedule the In so the most people can attend.
- → If a user changes their google calendar before the In is scheduled, our app will reimport their calendar information again right before it schedules the In via the google calendar API refresh token. Any manually set preferences will override the refreshed calendar.
- → Before the In is finalized, users can go back and view/change their existing preferences.

UI: Overview + Wireframing

OVERVIEW: SITE MAP



OVERVIEW: MEDIUM DETAIL



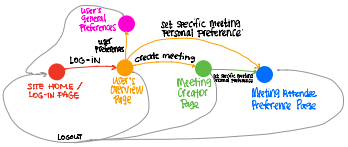
NOTE ON ERROR HANDLING:

SINCE USERS HAVE A VERY LIMITED WAY TO INPUT INFORMATION TO OUR SITE, WE WILL CHANGE THE FORM FIELDS AS USERS INPUT INFORMATION TO PREVENT THEM FROM INPUTTING ILLEGAL THINGS (LIKE ENTERING A LATEST END TIME BEFORE AN EARLIEST START TIME, THOSE TIMES WON'T BE CLICKABLE), OR TRYING TO SELECT OR ENTER INVALID DATA WILL DO NOTHING AND SHOW AN ERROR MESSAGE VIA AJAX (FOR EXAMPLE, YOU CAN'T INVITE SOMEONE WITH A NON VALID GMAIL ACCOUNT.

ALSO, GOOGLE ACCOUNT VALIDATION WILL TAKE CARE OF FAILED LOGINS AND OUR SITE DOESN'T NEED TO. WE WILL ALSO MAKE A CUSTOMIZED 404 PAGE IF A USER TRIES TO GO TO A INVALID OR OFF LIMITS URL THEY AREN'T LOGGED IN TO.

PAGE DETAILED WIREFRAMES

SITE HOME/LOG-IN PAGE



Welcome to Pencil Me In,
We make scheduling meetings easy, as it should be.

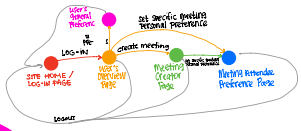
All you need is a google account

[g+ Sign in with Google]

Don't have a google account?

[g+ make Google acct.]

→ USERS MUST LOGIN WITH A GOOGLE ACCOUNT.



User's Overview/Home Page

Hello, Abraham! [YOUR PREFERENCES] [Log-out]

[Schedule a new Meeting]

Your Pending Meetings

- 6.170 Team Design Meeting [Edit Preferences]
- 6.034 Study Group Quiz 3 [Edit Preferences]
- Friend Workout [Edit Preferences]

Recently Scheduled

- 6.806 Study Group Quiz 2 Tue, Nov 17 3pm

→ REDIRECTS USER TO THEIR GENERAL PREFERENCES PAGE

→ Logs user out + redirects to Login Page

→ SENDS USER TO MEETING CREATION PAGE

→ EACH REDIRECTS USER TO MEETING PREFERENCE PAGE SPECIFIC TO MEETING BUTTON IS NEXT TO

→ THIS IS JUST FOR REFERENCE, SO A USER CAN SEE ALL OF THEIR UPCOMING MEETINGS OUR APP SCHEDULED FOR THEM

(EACH SCHEDULED IN WILL SEND USERS AN EMAIL W/ THE INFO + PUT THE EVENT IN THEIR CALENDAR FOR THEM TOO)

Meeting Creation Page

Create a meeting! [HOME] [Log-out]

Meeting Title : [ ]

Location : [ ]

Time Range

Earliest start : [11/8/15] [03:00PM]

Latest End : [11/14/15] [04:00PM]

Invitees

[@gmail.com] [Invite]

caroline@gmail.com ✕

rcorcillo@gmail.com ✕

[SUBMIT] [CANCEL]

→ cancels + redirects to user home

→ ON CLICK, CALENDAR TOOL POPS UP

<NOV> <2015>

→ ON CLICK, TIME SELECTION TOOL POPS UP

12 AM
⋮
12 PM

→ ADDS gmail user to tentative invitee list below

→ REMOVES INVITEE FROM TENTATIVE LIST

→ ADDS FINAL INVITEES AS MEMBERS OF THE MEETING. (ALSO SENDS EMAILS TO ALL)

→ REDIRECTS CREATOR TO MEETING SPECIFIC PREFERENCE PAGE

User General Preferences Page

Your General Preferences

HOME | Log-out

Earliest start Time: 09:00 AM

Latest End Time: 04:00 PM

[Potentially additional settings]

SUBMIT

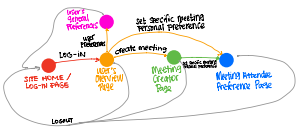
CANCEL

On click, time selection tool pops up

12 AM
⋮
12 PM

Saves user preferences and redirects to user home page

Cancels + redirects to user home



Meeting Preference Page

Scheduling Preferences For 6.170 Team Design Meeting

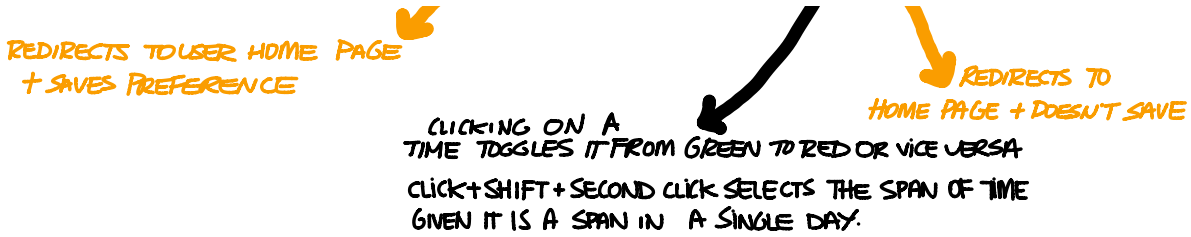
Between Nov. 8 10am — Nov. 14 10pm

HOME | Log-out

10am	10 – 11 6.034lec		10 – 11 6.034lec		10 – 11 6.034lec
11am	11 – 12:30p 6.858lec		11 – 12:30p 6.858lec		
12pm		12p – 1p 6.034rec		12p – 1p 6.170rec	
1pm		1p – 2:30p CMS.309lec		1p – 2:30p CMS.309lec	
2pm					
3pm	2:30p – 4p 6.170lec		2:30p – 4p 6.170lec		
4pm					
5pm					
6pm					
7pm					
8pm					
9pm	9p – 10p 858 lecture quiz	9p – 10p 858 lecture quiz	9p – 10p 6.170 lecture quiz		

SUBMIT

CANCEL


REDIRECTS TO USER HOME PAGE
+ SAVES PREFERENCE
REDIRECTS TO
HOME PAGE + DOESN'T SAVE
CLICKING ON A
TIME TOGGLES IT FROM GREEN TO RED OR VICE VERSA
CLICK+SHIFT+SECOND CLICK SELECTS THE SPAN OF TIME
GIVEN IT IS A SPAN IN A SINGLE DAY.

Challenges

Design challenges

List of problems to resolve in concepts, data model or user interface
For each problem: options available, evaluation, which chosen

1. 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.
    a. Options:
        i. Ignore the new data- this is the way that doodle works currently, it does not update due to new constraints.
            1. Evaluation: Although this lacks a nice feature, it is the standard for our market so users will not expect our app to contain this feature.
        ii. Keep listening to user's calendars and update the meeting based on new constraints.
            1. Evaluation: It might be difficult to keep the connection to the google APIs open for such a long period of time such that we can keep listening to new calendar events that a user has.
    b. Final decision:
        i. We are choosing to ignore the new data for the initial MVP. This is because we think that there are significant technological risks involved in implementing the other option. This is a feature that we might choose to implement later on.
2. If there is no single time that all users can meet:
    a. Options:
        i. We can throw away the meeting and tell users that there are no viable times.
            1. This seems inconvenient because we are not giving our users any information that can help them to make a decision.
        ii. Choose the In so that the host and the maximum number of invitees can attend, then warn everyone that not all invited can attend.
            1. Seems like a better way of dealing with this problem because at least we give our users some information about the largest subset of people that can attend their meeting.
        iii. Let the creator choose which invitees they care the most about and schedule an in just for that subset.
            1. We want to help the user make decisions with a small amount of user interaction, but this particular feature would require a lot of user interaction with little payoff.
    b. Final Decision:

        i. We chose option ii because it allows us to give users the most information while requiring the least user input.
3. How to let users show their preference of times:
    a. Options:
        i. Users can click on a block and set their preference for that block by indicating
            1. Preferences have to be manually set for each block.
                a. Evaluation: We can reduce this issue by having a default preference for blocks.
        ii. Don't let users show their preference of times:
            1. This is an important feature that allows users to select an optimal time.
                a. Evaluation: It seems very limiting to not include this feature.
    b. Final decision:
        i. We chose option i because we believe that it is important to allow users to set preferences and we believe this is an easy feature to implement.

Data design choices and their justifications

- Time range for a meeting is immutable
    - If it weren't immutable, the creator would be able to change the original time range for the meeting such that there is no overlap between the new and original time ranges. If so, all participants will have to fill in their time preferences again.
- A user must have a google account
    - Our application relies on using google calendar, which a user only has if they have a google account