CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Executive Summary

Our group first voted and chose to create a new calendar organizer application, a survey was sent out to assess the direction we were to approach. The results lead us to design a calendar similar to other existing calendars but with event planning, after TA feedback we concluded that the calendar would have too large a scope, too much work devoted to insignificant highlights of app, and the event planning portion would be neglected. We agreed to abandon this idea, and pursued a unique design focused on event planning. The idea was for everyone to pool their availability into a single shared calendar which showed the optimal days and times to plan any particular event.

Access to the calendar would be granted via a copy and paste code shared by users that grant access to an ad-hoc shared calendar. Initially we tried to avoid the generic calendar interface but other designs seemed of putting and most people are familiar with this design. The calendar design is purely for marking a user's availability. After some users have entered their availability a view shows up listing the potential days for an event ranked by number of people who selected them. This is used to narrow down the days first and then input the time available to ensure users don't waste time filling out their available time for every possible day but rather fill it for a few optimal days.

Some other ideas considered include allowing a repetitive schedule to be applied to multiple days, functionality to notify the final planned day for event. Other feedback from the evaluation was mostly aesthetic, involved changes that would cause trade-offs, or ideas that were not priorities. Some future iterations include allowing users to be more flexible with day and time windows, add a way to finalize planning and sync to existing calendar services using API, toggle for user to change the way they select days with availability or unavailability, allow users to upload pictures to use as flairs, and allow users to specify the time the event must take place in without a specific date.

Introduction

The Human Computer interaction discipline is about making not just useful software, but also usable software. The software must enable the user to perform the tasks that they need to do, in a way that the user can intuitively learn and understand. We initially put a bunch of ideas on the floor before deciding on a Calendar through a group majority vote. There are a lot of

CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Calendars out there: Google Calendar, Apple Calendar, Microsoft Outlook for some major ones, and alternatives like S Planner that adapt these Calendars with some specific twists for use on specific devices. As a result, it is very hard to be innovative in this area. For our research, we took a survey about current usage of Calendar applications and looked into our competitors to decide which direction we wanted to go. At first we wanted our application to be similar to standard calendars only adding the additional functionality needed to facilitate social event planning. As a result, our lo-fi prototype looked very similar to existing renditions. For a variety of reasons to be discussed in further detail, we decided to abandon our existing approach and pivot to tackle specifically social event planning.

Design Problem

After our pivot, we focused in on a specific problem that we wanted to solve. When planning a social event, there are a couple of different possible approaches. For larger events, one might simply pick a day and time before inviting a large number of people. In this case, the event coordinator or host is aware that it is not likely that all of the invitees will be able to attend. In some cases where too few people are able to attend, the host may decide to reschedule. In this case, how do they do so? They could simply select a different day and hope that day is better, or they could try to figure out the group's availability. Or say for our group, if we wanted to keep track of group availability to collaborate on the project, we'd want to consider everyone's availability and make sure that everyone was able to come and work together.

From our research, to be discussed in more detail further on, there is no easy way to keep track of everyone's availability all in one place. While existing Calendar applications allow their users to easily keep track of their personal schedules and even share the Calendar, the functionality is not flexible enough. Existing applications allow the user to share their entire Calendar, this will of course include a lot of information that is irrelevant to the plans.

As a result, we decided to focus on this problem: providing a simple ad-hoc social platform that provides a space for users to quickly mark their availability or unavailability to plan group events with ease. Rather than sharing entire Calendars with irrelevant detailed information, let the user focus on indicating their availability.

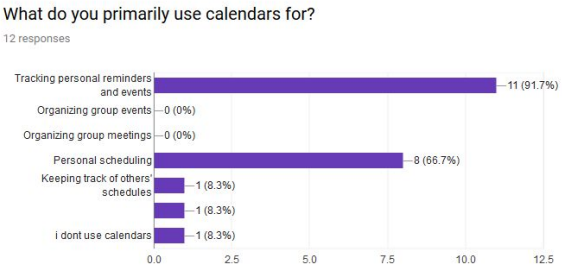
CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

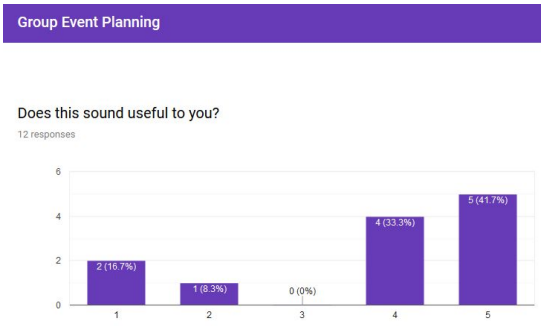
User Research and Findings

For our research, we began with a user survey. As part of the survey, we also asked our users what existing Calendar applications they currently used to perform a competitive analysis.

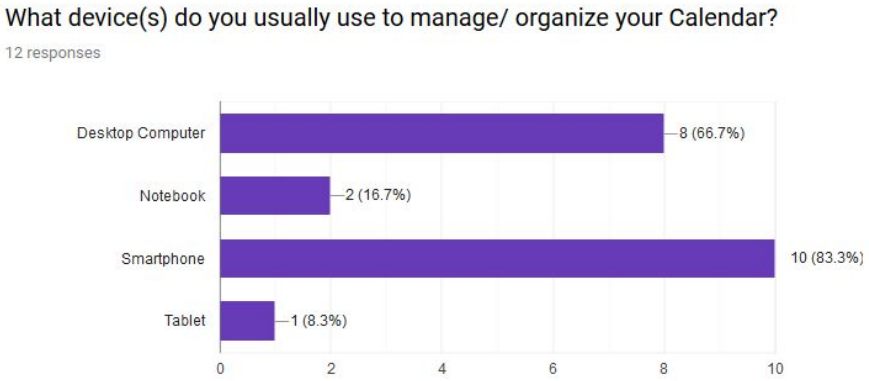
Survey

We chose a survey because our target user base is quite wide. Putting a survey out into the open internet would hopefully allow us to get responses from a more varied population. Unfortunately our time and social connections to plug the survey were limited, so our sample size is admittedly small. Regardless, we believe that some of our results were rather telling.





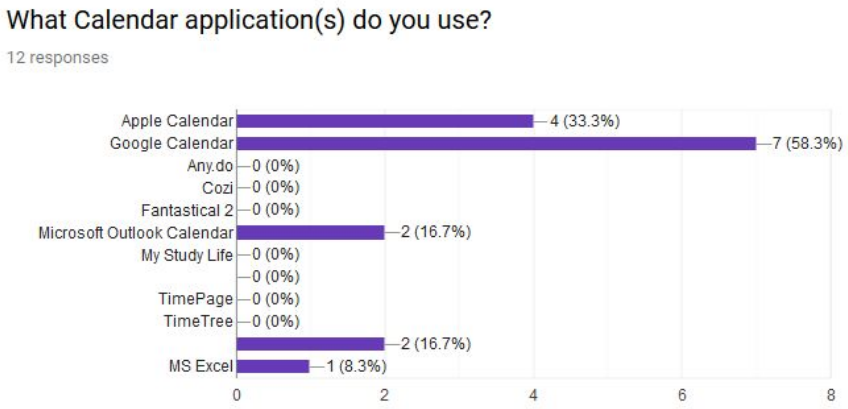
These results show that none of the users that we surveyed currently used Calendar applications to plan group events. In spite of this, the majority of them felt that being able to do so would be useful.



It is from this data that we decided to focus on a design for smartphones. As stated, we initially intended for a more fully-featured Calendar, so the thought of desktop use was still appealing.

However, since narrowing our focus and streamlining the flow of our application, we currently do not see a need for desktop development.

Many of the other questions on the survey were made before we narrowed our focus. While they did give us more general insight into our potential target users, we could not say that these answers were hugely influential in the specifics of our design choices.



These answers helped ups narrow down which competitors to use in our analysis.

Competitive Analysis

Once again, we performed this competitive analysis before we truly narrowed down the specific design problem that we intended to work on. The analysis did however, help push us toward our eventual decisions.

While a lot of the analysis ended up being a feature-to-feature comparison between the Calendars on how they implemented certain things, the key take-away for us was that none of them allowed the user to share events within a specific time frame to other users. Sharing was restricted to entire Calendars, and while this is fine for corporate situations where people need to know extended details of employee schedules over an extended period of time, it is not ideal for the ad-hoc nature of general social events.

Based on the above, we decided that there was a potential niche that could be filled here.

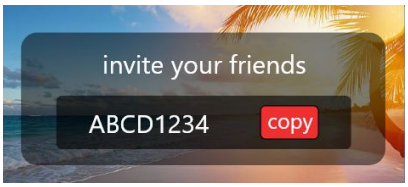
CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Design and Justification

Focus on event planning

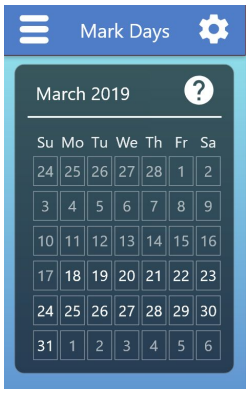
Initially we were really planning to just make a new Calendar application very similar to existing ones, but with additional utility specifically for event planning. We decided to turn away from this primarily for three reasons. First, the scope of such work would be too large. Calendars have a lot of functionality, and mirroring all of that would already a significant project. Second, relative to the large scope of work, the innovative parts that would actually involve design, which was what this project was meant to be focused on, were rather small. Finally, after producing our lo-fi prototypes, which were effectively generic Calendars with slight modifications, some feedback helped us realize that sticking too closely to the idea of a Calendar was not necessarily the best to address the problem defined above.

Social aspect



When the user begins the process to plan an event, they are given a code that they can send to others. The code will give other users access to an ad-hoc shared Calendar and other tools to help coordinate a time. We use a code to keep things simple and flexible. The coordinator can simply copy and paste this to whatever communication platform works best for their use.

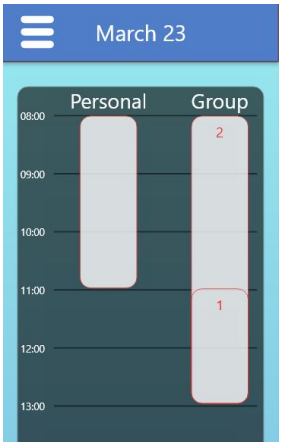
A Calendar, to narrow down days



We tried to branch out with our ideas and get away from a Calendar interface, but in the end we decided that the Calendar interface was best suited for initially narrowing down potential dates. However, we streamline and focus on what's useful for event planning. The Calendar is purely for the purpose of allowing the user to mark days as available or unavailable. As the primary platform that we were designing for is the smartphone, this avoids complex operations on a small screen. Rather than creating detailed on a Calendar blocking time, it's just whether the day is good or not.

Currently it is vague whether or not the user is marking availability or unavailability. Different options will have different trade offs, and will ultimately depend on how busy the user is. If the user has a lot of days where they are available, it would be easier to mark unavailability. If they are really busy and would have to mark a lot of days as unavailable, it is easier for them to mark availability. As this is intended to be an application used to plan events between relatively small groups of people, the current idea would be for the primary planner to dictate whether people should mark availability or unavailability. This still is not ideal, as different users within the group may have different preferences, but we will discuss this more in a later section.

Finding a time



In the next stage of the process we decided to list the days in order of how many people had marked the day or not. The user is able to swap between ascending or descending according to whether they were marking availability or unavailability. Rather than entering times for all of the days, the idea is to narrow it down to a couple of potential dates and then enter more detailed information specifically for those.

We decided to simply list them in this fashion because it seemed simple and it got the job done quickly. Some suggestions were made to make this part a bit more innovative. The innovation could make the design much more aesthetic, as well as emphasize the days that we intend for the user to focus on as well. To be blunt, during the design this screen was meant as a simple stop-gap between the Calendar view and a view to enter times, so there was not as much intentionality behind the screen as there could have been.

Finally the user is able to enter more detailed availability information for potential dates here. The view begins at 8AM and cycles 24 hours. There was a consideration of limiting the time range, but depending on what occasion they're organising for, the event could really be at any time regardless of day or night. There are some considerations that we have for this in future iterations that will be mentioned later.

CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Heuristic Evaluation and Findings

The heuristic evaluation gave us the insight that a lot of the symbols that we were using were unclear. We shaded days that were marked and numbered them with how many people had marked the day in total, but that would not necessarily be clear to a first time user. To help clarify, we added a simple help overlay that the user can refer to if they're feeling lost. We believe that once the user knows, they won't really need to refer to it again after that, so we kept it simple. Additionally there were some bugs in the flow of the prototype that the heuristic evaluation that were discovered and needed to be addressed.

One of the suggestions made that we did not take action on was to alleviate potential repetitive action in needing to enter the same times on multiple days if one had a repetitive schedule. While this could be nice to have, we do not consider it vital. The goal is not for the user to fill out times for every single day that they've listed as available, but only for a couple of days that are most relevant for the group. Another suggestion was that there is no clear end to the planning process. We have not directly addressed this yet, but we will touch on this on the section regarding the next iteration.

Much of the other feedback was either primarily aesthetic where either the changes would involve trade-offs, or not currently a priority. Certain points made in the heuristic evaluation were also left unaddressed as we have not yet found a way to properly address the problem and will be in consideration for future iterations.

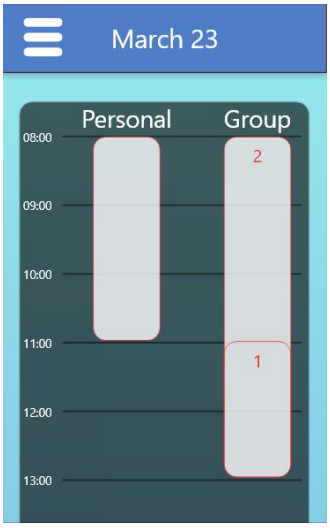
Recommendations for Next Iteration of Design

Examining our hi-fi prototype as well as the results from the heuristic evaluation and testing, here are what we consider as major action points for our next iteration.

CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Users are currently only able to narrow down a day within a month. We did not initially consider the possibility, but once some of our users brought it up, it is certainly true that there are times when one would want to plan an event either at the end of a month, or the beginning of the next month.

For the next iteration of design, we would want to refine the process to allow the event coordinator to be more flexible control with the window of days that they could plan in.

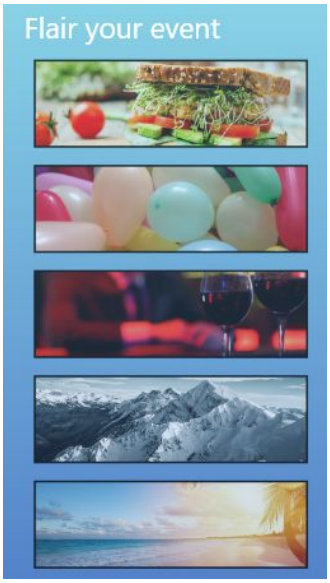


Related to the above, when users are entering their time availability, we allow them to mark off hours within a 24 hour range. While certain events that could potentially happen at any time in the day might benefit from such a large range, it also introduces a large window that the user must scroll through to operate. Some of the users told us that even if they don't have a specific time in mind, they may have a general window that they want the event to be in.

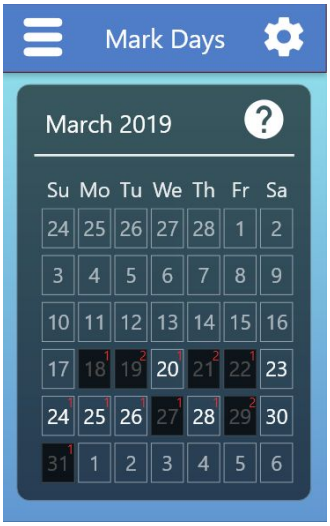
As a result, in our next iteration we intend to refine this so that the event coordinator has more control over the range of hours that is displayed here, cutting down on useless UI elements.

As part of the feedback from the heuristic evaluation, it was mentioned that there was no way to concretely finish planning an event. For our next iteration, we intend to include a way for the event coordinator to settle on a time, either deleting or temporarily archiving the plans before deleting them. Next time the participating users opened the application, the application would also prompt them with the time, asking if the user would like it to automatically create an event on a Calendar service using the service's API.

CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov



During the event creation process, the user is asked to pick an image to represent their event. The choices are limited and there might not be a flair that really fits the coordinator's purposes. To allow users more freedom of expression, we intend to allow users to upload their own pictures to use as flairs.



A common confusion that surfaced during both the user evaluation and testing was that people were unsure whether they were meant to mark their availability or unavailability. Initially we as designers wanted to keep it vague as which one was better seemed to depend on the user. However, as the application is meant to be used to allow multiple users to coordinate a date and time, the vagueness leads to potential errors with one user entering their availability and the other entering unavailability.

To resolve this, for our the next iteration we intend to allow users to indicate with a personal setting on the event whether they are marking availability or unavailability and have that sync appropriately to other users clients based on their own setting.

Alternatively as a solution that is less technically involved, we could simply choose to specify that the user mark availability for example and offer a button to quickly mark all days. This would allow a user who has a lot of available days to only need to unmark the few days that they are unavailable, rather than mark a large number of days.

CPSC 481: HCI Final Report
Dennis Nguyen
Devon Gillis
Gurneck Johal
Mikhail Starikov

Conclusions

In our initial assessment we sought out to redesign the calendar interface to include the new component event planning. After some surveys, reviews, and assessments of our scope, we reached the conclusion that the calendar portion of the app would take too much time and take away from our idea of the event planning portion of the app. Thus we abandoned the idea of creating a true calendar app and sought to focus on the event planning, using a calendar interface as just as a availability marking tool for users to pool their availabilities. This can then be used to plan any event with optimal efficiency and minimal scheduling conflicts.

In the future should we pursue this app further we like to add additional features which include extending the number of days an event can be planned ahead. Allow for users to copy-paste their time schedules to fit multiple days, add functionality to let the app interact with other calendar apps through API, and provide ease in users selecting their availability. Though there are many improvements that could be made over cycles of iteration, we feel that we did achieve a baseline design that allows users to come together to determine availability on an ad-hoc platform.