

Introduction to Human Computer Interaction

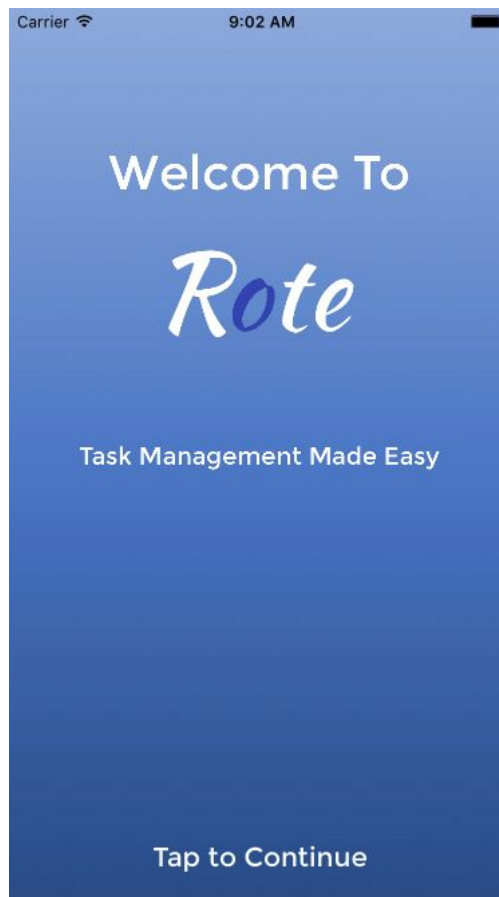
Final Report: Rote

ChangYoung Park, Gavin Mai, Ian Lewis, Haiyin Wang

Problem and Solution Overview

The Need: Current organizational tools are too complicated, distracting and time-consuming. For example, adding events in Google Calendar requires the user to create a task, look at their schedule for an open space, think about if it is an appropriate time, set the time limit, set a reminder etc. Not only are there too many steps, there are too many unnecessary features that take away from the act of organizing our most valuable resource: time.

Basic Approach: Rote is an intuitive and intelligent task manager that uses a combination of minimalist user design and cutting edge technology. Specifically, we strived to create an intuitive user design that's both friendly but powerful. On the technical side, Rote uses natural language processing to understand your schedule and machine learning to understand your habits over extended period of use, to help you organize your calendar in the fewest steps as possible.



Tasks & Final Interface Scenarios

1. Simple Task: Create a Profile

To help Rote have a baseline for your personal schedule, habits and characteristics, we decided to have users create a profile which lets Rote know when you like working, your age and sleep schedule. These are all factors that need to be taken into account for Rote to do a good job helping you organize your time.

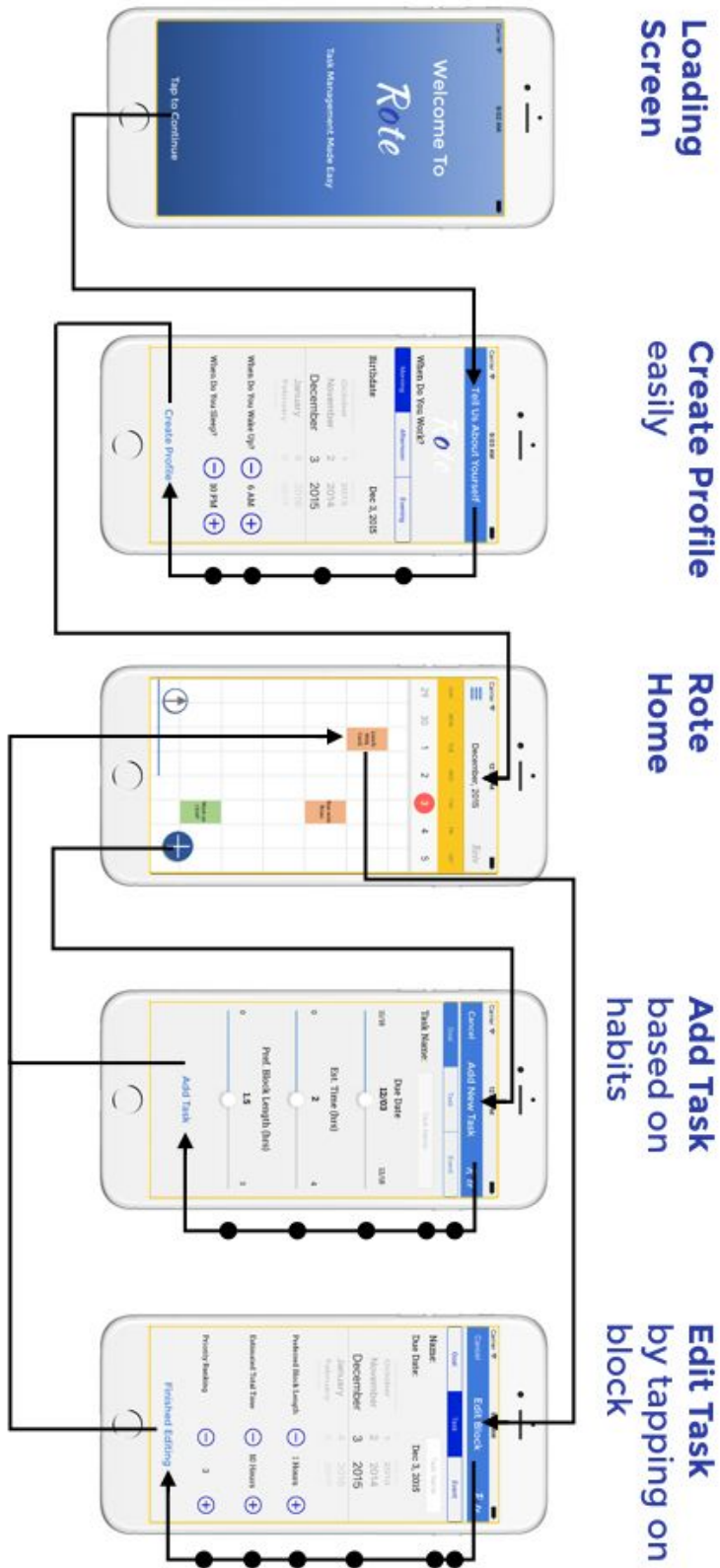
2. Medium Level Task: Add Task

Adding a task is the bread and butter of Rote. Here, users are presented a beautiful interface to add stuff to their schedule. Users give general information about the task such as the name, number of hours it takes and potentially when it's due. Rote uses this information to its advantage and does the heavy lifting by scanning your schedule for openings and filtering only the ones that fit your preferences. All you have to do is tap to confirm and you're all set.

3. Hard Level Task: Edit Task

Sometimes, tasks change and thus, we need to update it manually. This is very easy. Rote allows your to quickly edit your task by tapping on it and changing the duration or date quickly. We see this task as most complicated because there are a few more steps to this but all necessary.

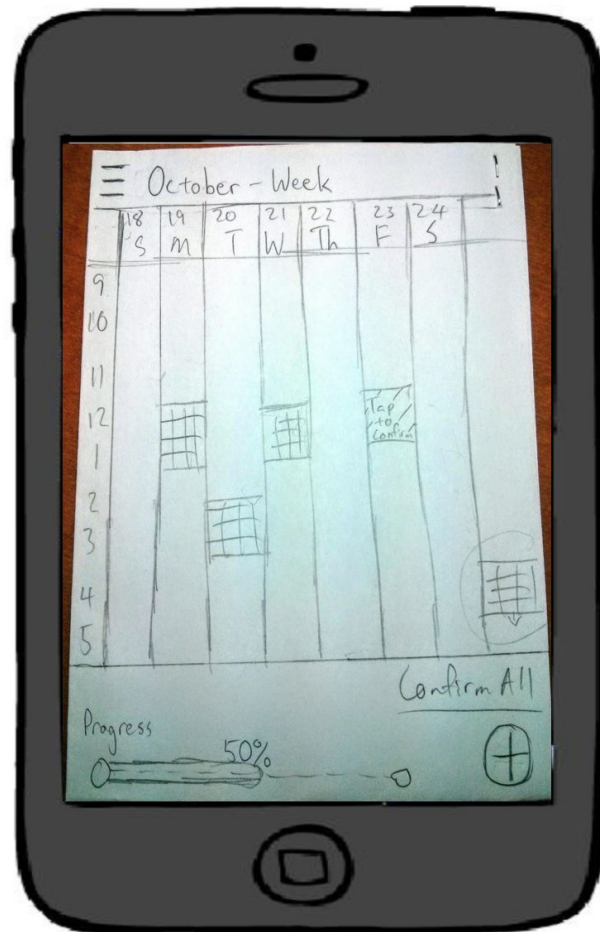
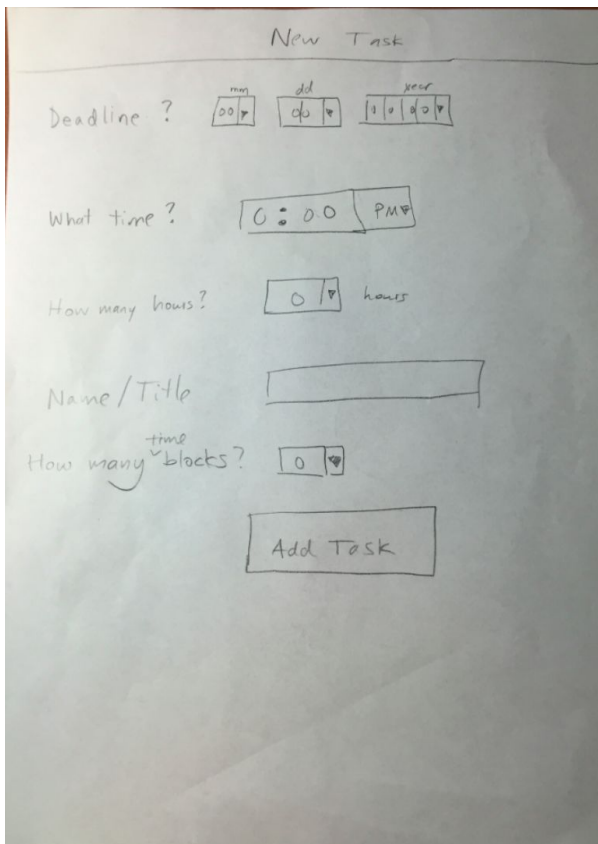
Storyboard walkthrough of each task using final interface images



Design Evolution

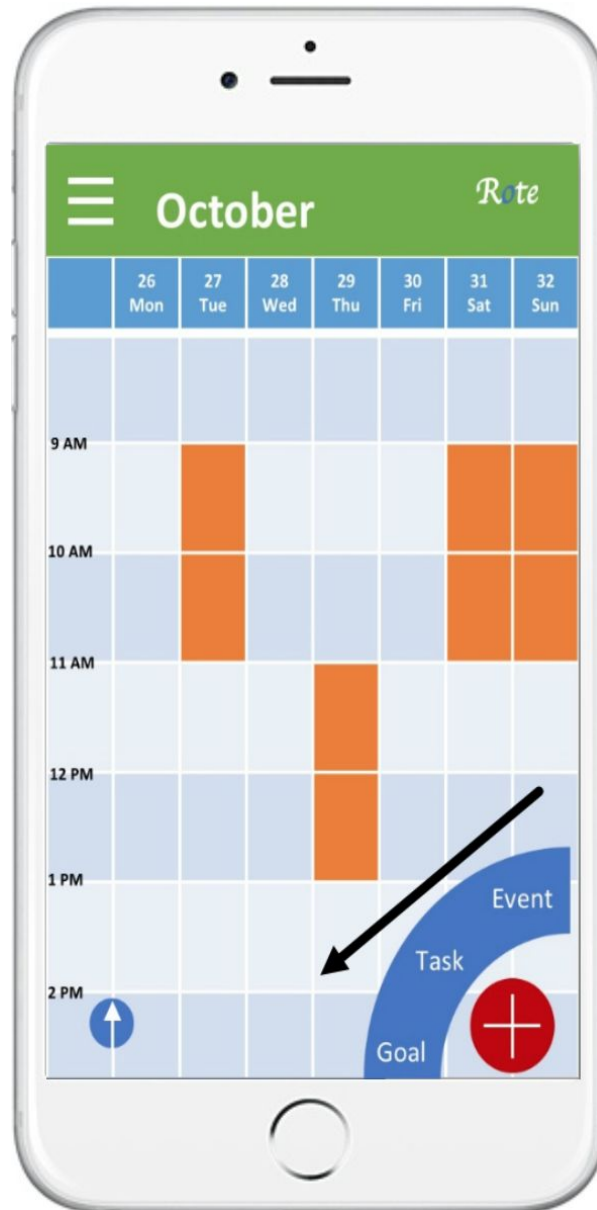


Here, ChangYoung is conducting an interview with a few strangers at the Tressider Student Center. We've drawn up designs on paper in a low-fidelity testing stage to gauge interest and comprehension of our initial idea-testing phase.

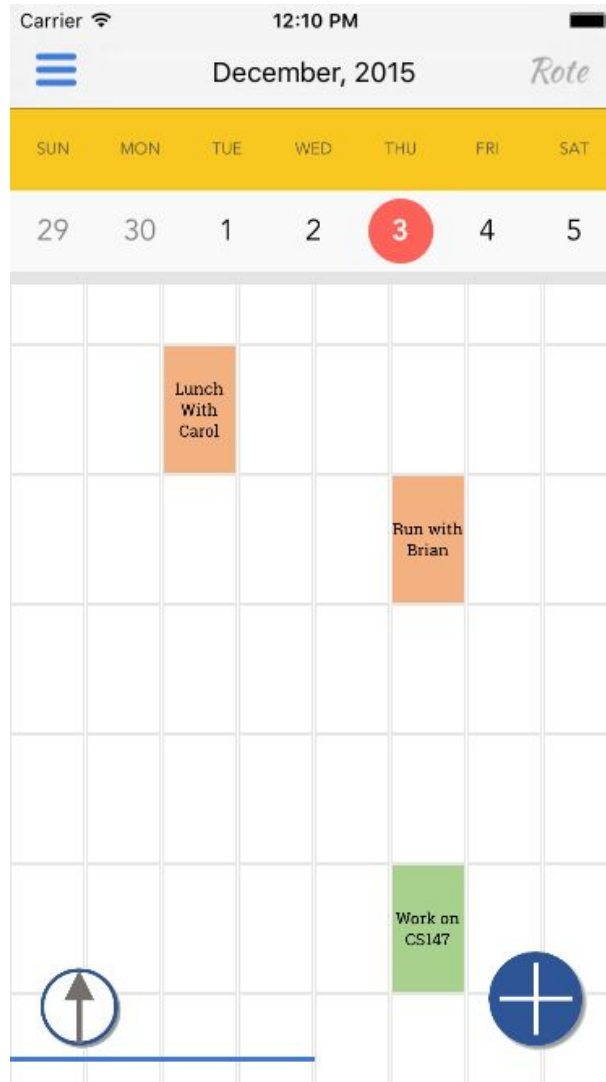


Here we have the main pages from our design process. Our preliminary concept sketch is shown on the left. This was an early attempt to put our untested ideas on paper and get a feel for how it would look. On the right, we have a low-fidelity prototype of our Rote, which incorporates more detail in the form of a progress bar, real time-blocks and an “add task” button on the bottom right corner.

Some of the feedback we received from interviews and in-class presentations was that the progress was too intrusive without really serving a real space purpose. As a result, we tried to get rid of it for the medium-fi prototype. Funnily enough, we reintroduced just a single line for the high-fi prototype to give the indication of how far the user is.



This image above is a medium-fidelity prototype created using the app prototyping tool InVision. The goal was to take some of the suggestions given by the diverse group of interviewees and incorporate them into this new and more developed version of Rote. Some of the feedback included using a certain color scheme that indicated what was important. There was also a strong opinion to make sure the user knew what was important in the app without explicitly stating it.



To incorporate some of the changes and suggestions brought up in the heuristic evaluation, we added more colors to the time-blocks, which allows users to custom identify time-blocks quickly just based on the color scheme. We kept the color scheme throughout the app very consistent with the image of our website and other online presence by using the same color scheme (Rote blue, yellow and white). We even imported the custom Google fonts into the XCode workspace to produce this high-fi prototype.

Although you can't see it, we also implemented some key user interface changes such as the ability to double tap on a time-block to edit it. The arrow at the bottom left allows users to

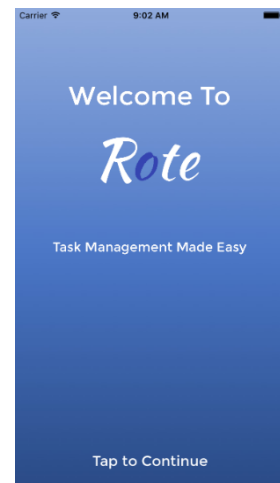
easy return to the current date. The bottom right “plus” sign expands into a fan upon touch and allows you to add an event, goal or task quickly.

Overall, we believe that the high-fi is good representation of how the real app in all its functionality will look. The color scheme is pleasing to the eye and Rote definitely is doing a good job helping the user add a task and and interacting with it on the home screen.

Major Usability Problems:

No “Welcome” page before the profile creation:

- In the medium-fi prototype we didn't have a welcome screen because we were too focused on simply creating a digital version of Rote. After reading this violation we all agreed that a welcome screen was a great way to increase the flow of our app, and it was easy to implement, so our final hi-fi prototype now includes a welcome screen.

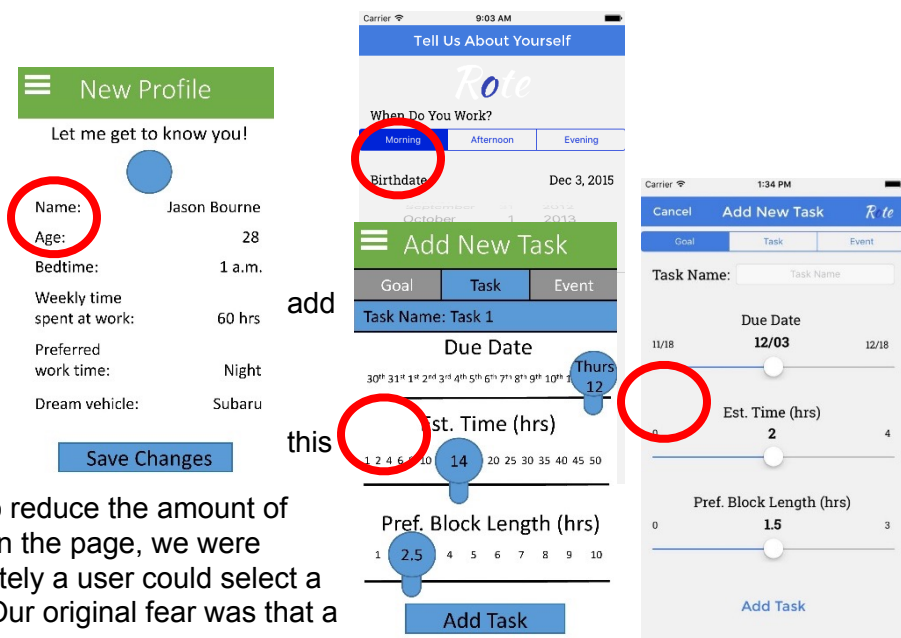


Asking for user's age instead of birthday creates problems for the future:

- We thought this was an excellent thing to point out. Asking a user's age would have brought up issues with what month/day their birthday was, and it had the ability to make users feel self-conscious during an already semi-invasive profile creation process. Asking for a birthday solves all those problems, and although a learning algorithm with age bracket considerations would probably not need the specificity a birthday provides, having that specificity is not a bad thing. If we continue to improve this app we could only ask for the year in which someone was born, but as it stands, we asked for a full birthdate in our final hi-fi prototype.

Timeline in “Add Task” page is too crowded by suffixes:

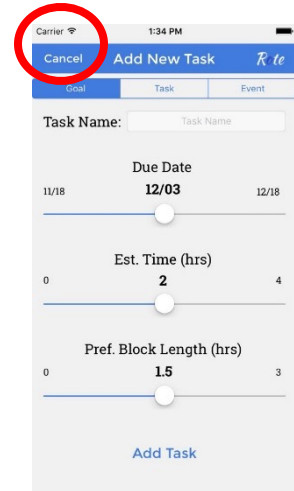
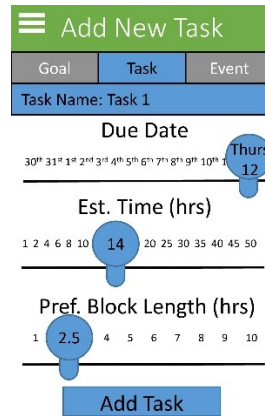
- After looking at our task page in the medium-fi prototype, we originally doubted idea. Although we definitely wanted to reduce the amount of numbers/suffixes on the page, we were unsure how accurately a user could select a date with a slider. Our original fear was that a



user would not easily be able to select the correct date if the bar contained 30-31 possible due dates. In the end we realized it was in fact really easy to select the correct date in a data set that big, and so in our final prototype we only included the limits of the timeline, without suffixes.

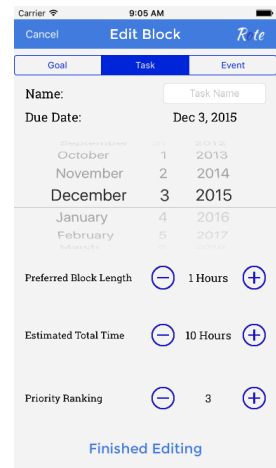
No cancel button during the “Add Task” phase:

- After looking at this violation we all agreed that we needed a cancel button in the Add Task phase. We simply forgot to put it in the medium-fi prototype. As a result, we added it to our hi-fi prototype.



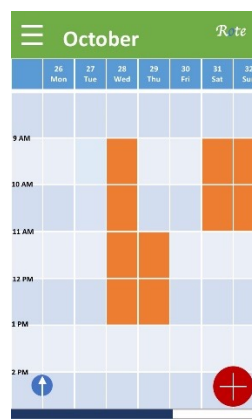
Unclear editing procedure of individual blocks:

- We felt that given the right tools, the editing procedure of individual blocks would be a lot clearer. Although we couldn't add this functionality in our medium-fi prototype, in our hi-fi prototype we wanted to add a “jiggle” animation to the suggested blocks, signifying in iOS style that the blocks could be individually dragged around to edit their location. Unfortunately, due to time constraints and the difficulty of creating a calendar app from scratch, we weren't able to animate the blocks the way we wanted to, or even edit them the way we wanted to. The table we created to represent a calendar is static, and dragging blocks around the page is well beyond the capabilities of that calendar. In the end, we left out the ability to confirm the suggested location of tasks. The “edit task” functionality is still very much alive (see right), and can be reached by double-tapping an individual block of time, however the sleek “quick edit” functionality was too hard to implement in the timeframe we had.
- For the purposes of the demo we did have a colored visual to represent when a task has yet to be locked in, however after editing a task, that becomes the end of the demo, and leaving the app simply wipes the creation of the dummy task. You never actually get to “lock in” a task.



No task names or identifiers on the calendar:

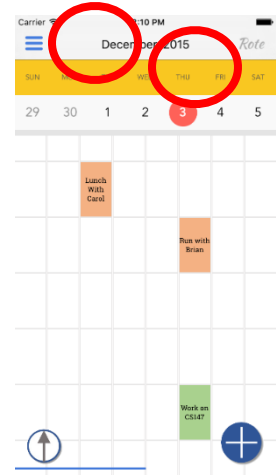
- Our original intention was to have the names events in the calendar, we just forgot to label them in our medium-fi prototype. Our tasks in our hi-fi calendar are



of tasks on the them in our medium-fi labelled.

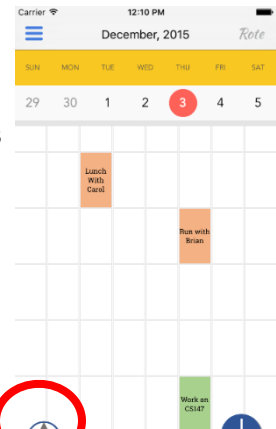
“Pref. Block Length” is an unclear description:

- We agreed that “Pref. Block Length” is an unclear description, but our solution wasn’t to change the wording. “Pref. Time-Block Length” would be too long, and it still doesn’t fix the problem if the user is confused about what a time block is and how it relates to creating a task. After spending some time thinking about it, we couldn’t find a way to intuitively explain the concept of time-blocking to someone who was unfamiliar with the definition of the term. As a result, we agreed that if we are to continue this project past this quarter, we would want to implement a quick 4-5 step tutorial. Not only would this fix the problem here, it would also add clarity throughout the app, giving people the ability to understand how useful it can be before they start using it.



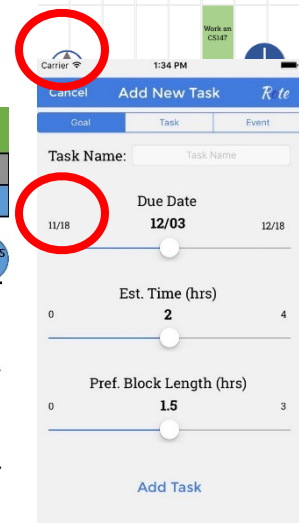
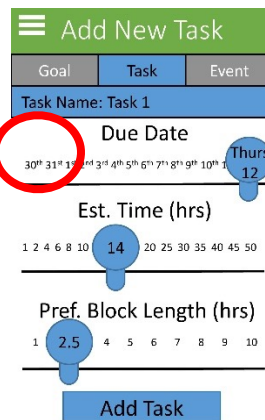
Arrow does nothing when you are on the current day:

- We didn’t agree that this needed to be changed. In the evaluation, the arrow was described as “providing no additional information” when it was on the current day. We agree that all the arrow signifies at that point is that you are on the current day, however we thought that it was valuable information to provide. We also thought that leaving the arrow there provides consistency throughout the calendar experience. We didn’t want two different buttons to resemble the same task of signifying where “today” was, because we thought it would only lead to confusion. As a result, we just left the arrow functionality the way that it was.



The add task page doesn’t label the month:

- Initially we didn’t include the month in the medium-fi prototype because of space issues. After moving to the hi-fi prototype we realized we had a lot more space than we originally thought. The font size on an iPhone screen was actually usually around 6-8, while we were sizing the fonts at around 12 in our medium-fi prototype. After removing the suffixes (as per an earlier suggestion) we found that we had ample space to add the month and “/” symbol. And so we did.

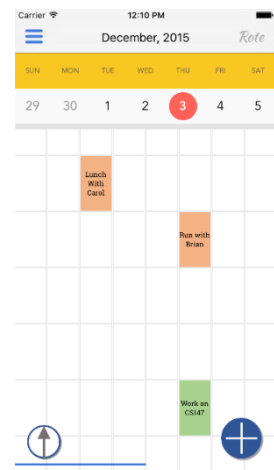


Clicking task would occasionally shift the time back a week:

- This was a limitation of the medium-fi prototype outlined in our read-me file. We couldn't save what page the user was on before, and as a result we would have had to create a separate set of links for each page, past-present-future, for each type of view. In the end, several of the pages ended up linking back to the original week view in order to reset the app before you got stuck on a certain page. This of course wasn't an issue in the hi-fi prototype because we could then generate infinite days and screens without having to manually program each one. As a result, this problem was naturally solved by increasing the fidelity of the prototype.

Color coding was inconsistent:

- Due to limited time (the calendar's ability to retain tasks was finished on Friday morning) we were unable to implement a complex color-coding scheme. Instead, the color green signifies when a task has been suggested but not yet added to the calendar. This allows the user to quickly find which tasks Rote has recently suggested to the calendar, however it was mainly for the purposes of the demo (since we couldn't add the "jiggle" animation). In the future, we agreed that colors should group the type of task (either a task, goal or an event) to allow users to easily find what they're looking for by first looking for the type.



Other changes

Color scheme change:

- We made the color scheme more consistent with our logo and added the color yellow for visual effect. In the end we just thought it looked a lot better, and provided a little more consistency with the "brand" of Rote.

Due Date scroll bar in Add Task screen doesn't move like the other scroll bars:

- We wanted to be able to move the due date past the available points we would put onto the screen, and our original idea was to have "bumpers" on the ends that would move the entire range of dates displayed on the timeline whenever you scrolled over that far (similar to clicking and dragging an object through multiple pages in a word doc). In the end this became too ambitious, so we thought we would try a new idea, where the scroll "button" on the timeline would always return to the center. We felt that this would actually be faster for the user to select a due date within 2 months, even if it was less normal.

Prototype Implementation:

We built our prototype using “cocoa pods” which are nicely packaged tools that allow us to add features like a side menu without having to code it up from scratch. This combined with Apple’s provided tools allowed us to be consistent with most other iOS apps. Apple also provided an easy to use “Storyboard” interface that made organizing screens extremely easy. Parse was used for backend profile handling, which allowed us to create a very real profile creation and storage system, and was also relatively easy to use. The animation functions provided by Apple were also very intuitive and initially required no coding. Overall the look of the UI (so not including the parts containing functionality) was easy to design and implement in a professional looking fashion.

There were a lot of problems creating this particular prototype. We originally assumed Apple would have an easy to use calendar interface we could insert into our app, however that turned out to not be the case. In fact, we couldn’t find any resources online for creating a calendar interface on a recent iOS version. Although the calendar was a core part of the UI and design of the app, the actual part of Rote that makes it unique has nothing to do with building a calendar. As a result, the calendar took the most time to implement even though it was not a part of our 3 tasks. Most of the limited backend functionality we were able to implement had to be done completely from scratch. Overall, there was no mobile programming experience in the group, and there ended up being a very steep learning curve.

In the end we had to hard-code a lot of the calendar data. For example, the Add-Task function doesn’t actually add anything to the calendar, it simply generates a set task on the calendar screen. This also rules out the possibility of actual AI in the app. Similarly, edit task does not actually edit the data, but instead just represents what you could do if the backend functionality actually worked. Essentially, anything that involves actually storing new data in the calendar is hard coded.

In the future we would of course want to change this. We would probably make a much nicer looking calendar interface that would actually store events properly. We would also want to actually implement an AI, giving users custom scheduled tasks without the user having to think about it. We would also want to implement the bumper technique on the Due Date scroll bar (I mentioned this earlier), and add a short tutorial for those who are new to the app. We would want to add functionality to the “status” bar at the bottom of the calendar screen, and in the end we would probably want to add “jiggling and dragging” functionality to the preliminary “edit suggestions” step. Finally, we would really like to add voice functionality to Rote, and hopefully have it integrated into the phone’s AI. In the future, we would want to be able to say “Cortana, block out 50 hours for a Thesis Paper I have to finish by December 12th” and have Rote simply do that in your calendar.

Summary:

This project was way too hard. We liked where we ended up with the design of our app, however even implementing extremely limited functionality created setbacks in the overall app design. We would never recommend doing a calendar app to anyone in future classes until someone releases a good calendar API. That said, we learned a lot of new skills that have a lot of practical applications. Although our idea was not fully implemented in this class, we believe that the problem we found is worth solving. Organized calendars are a hallmark of success in the business world, and the fact that they’re so time consuming right now is a tragedy. If we

don't continue the production of this app we hope that someone else might carry the torch. We really enjoyed being in this class and we really value a lot of the design principles we've learned this quarter. We hope to continue our education out in the real world, and look forward to implementing the ideas we learned in this class in the future.