

## Final Writeup

Designing Human Centered Software  
Spring 2015

*Evan Metsky*

*Elliott Tan*

*Jean Kim*

*Nathaniel Fruchter*

*Neeharika Taneja*

## The final idea

Have you ever been stuck on that silly calculus question asking you to take the partial derivative of some ridiculous equation? Or maybe you just can't quite grasp the concept of why the mitochondria is "the powerhouse of the cell." You're looking for help from friends, but they're all busy doing their own work or hitting the town with their hot dates. With your homework due later this evening, you don't have time to schedule an appointment with your tutor or go to "academic development", which is conveniently scheduled an hour after your homework is due. Tutyr comes to the rescue.

Now with Tutyr, you can quickly and easily find a tutor instantly rather than having to schedule an appointment. All you have to do is select your subject and Tutyr will connect you with an experienced college student in that subject and give you directions to their location on campus. Along with giving you instant access to tutors, our application will also be cheaper since the tutors get all the money, rather than a corporation or middleman taking a cut. Tutyr will not only benefit students in need of tutoring, but will also allow tutors a better opportunity. Tutors can work on-demand, or when they please, rather than having to predetermine schedules while also giving them a chance to do their own work in between sessions.

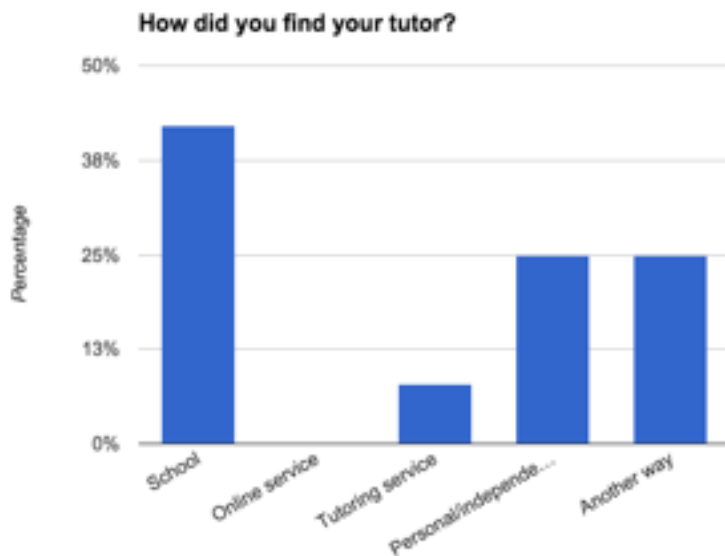
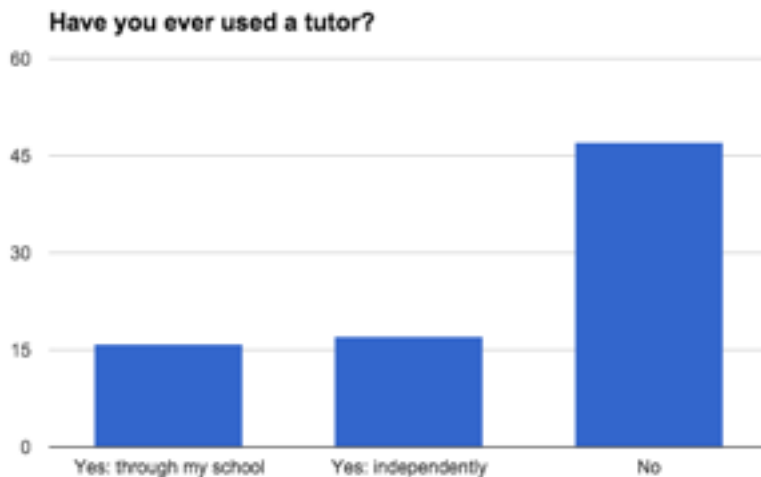
Tutyr will give students the chance to benefit from instant access to tutors while also giving them the opportunity to get paid for tutoring students in a field they have experience in.

## Team member roles

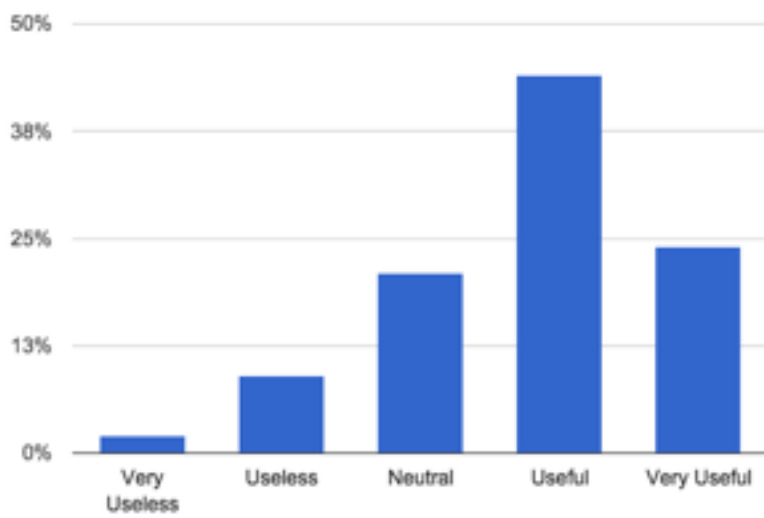
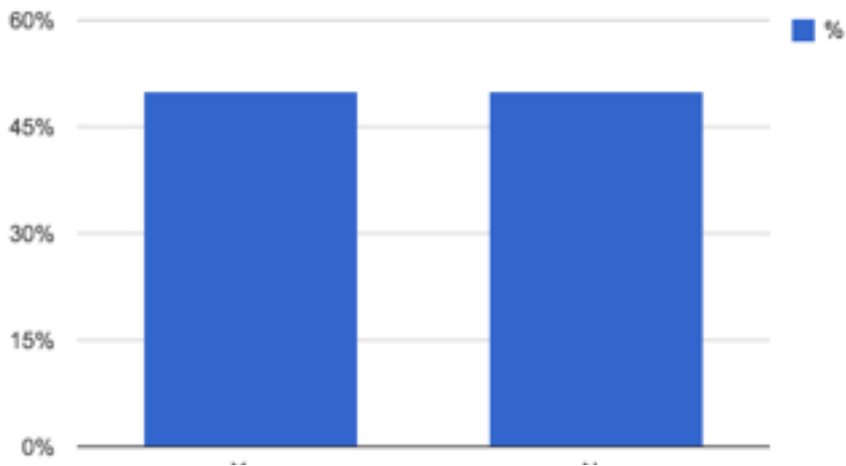
- **Elliott:** developed backend web server and integrated with the front end.
- **Evan:** helped develop app idea, needs finding and research, initial design and app flow, project management, created final videos.
- **Jean:** designed lo-fi to hi-fi interfaces and interactions, and created design language for branding.
- **Nathaniel:** implemented app front end using Javascript and Ionic Framework and helped integrate with back-end.
- **Neeharika:** worked on some initial application flow, some front end screens and the final write up.

# Needs finding and research

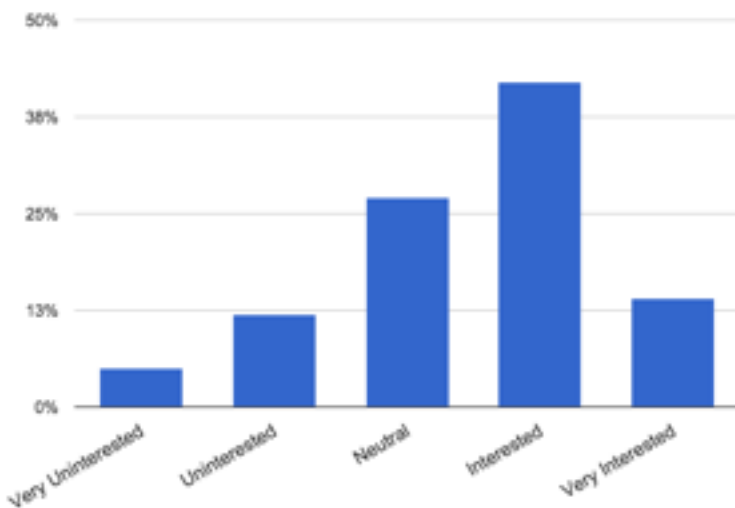
We conducted online surveys through Qualtrics and received responses from 80 participants. This survey was distributed through various CMU Facebook groups and our own social networks. In essence, it served as informal market research—a way of assessing general receptiveness at a larger scale than interviews. Some of the important results are shown below.



### Ever been a tutor for an academic subject?



“How useful would an app be if it could find you a tutor immediately, whenever you need help?”



“Would you be interested if this app could help you meet up with a tutor in person?”

# Early designs, prototyping, and iteration

## Our process

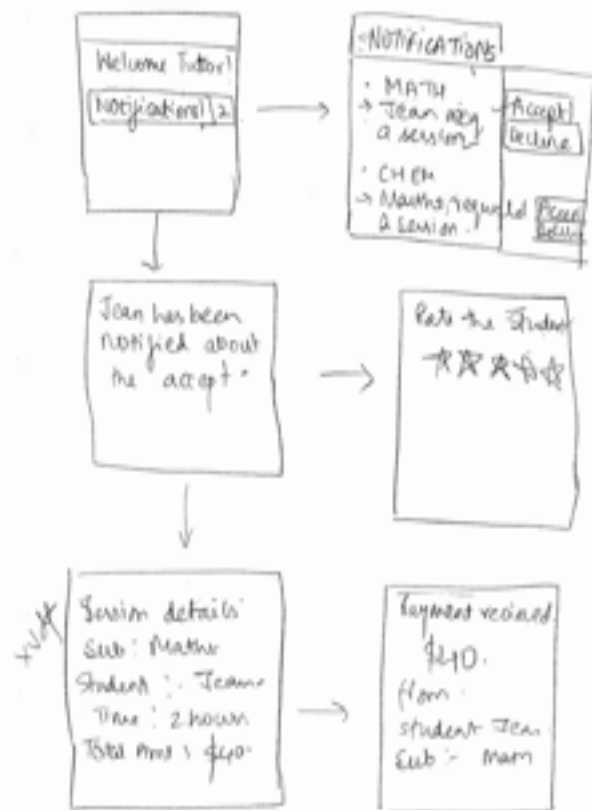
Our initial development process started with brainstorming, ideation, and needs finding. We got together and iterated on many possible user flows, with each member creating their own app storyboard. These were then discussed and used to create an initial idea of how the user flow would work. This process also happened alongside our research and needs finding phase, which mainly consisted of a “market research” style survey that examined the needs and wants of the CMU student population.

This information was then used to collaboratively construct a full wireframe for the app, which was used as the basis for an interactive lo-fi prototype. At this stage, we also started constructing the necessary spec for the API that our back end would implement and started evaluating various server solutions for implementation. Once this process concluded, we had a relatively solid idea of our backend’s capabilities and our user flow, so we were able to conduct several user studies before starting the development process.

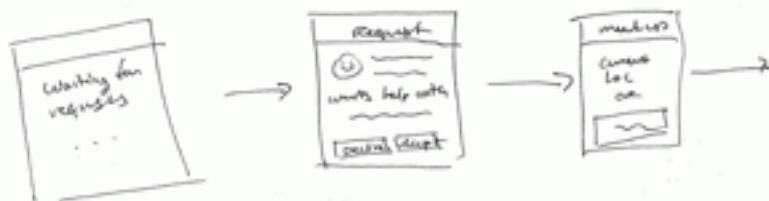
At this stage, we started front and back end development after the creation of a final, hi-fi interface prototype created in Illustrator. Once this implementation process finished and we had a ‘rough draft’ (so to speak) of our app, we iterated on several elements of the design and user flow while debugging the front and back end.

## Ideation and storyboarding

Below are some example sketches from our ideation and storyboarding process.







REQUEST → MEET → FINISH







## Wireframing

After the initial ideation and storyboarding phase, we moved forward by creating an interactive low-fi prototype on the cloud-based Justinmind wireframing service. Unfortunately, our trial expired before we were able to archive this prototype and we do not have any screenshots from the wireframing process.

## User studies

We also ran several user studies concurrently with the wireframing process. Below is an abridged version of our script and findings.

**Background:** Tutyr is an application for your smartphone that connects you, a student, with on-demand tutoring offered by other students like you. As a user, you are able to see nearby, active tutors and their areas of expertise and request a session with one of them. At the end of the session, you are able to rate tutors and give feedback about your session.

**Motivation (Tutee):** You are a new Tutyr user who has just downloaded the app because it sounded useful. Your goal is to sign in with an existing account, request a session, and end it.

Key questions:

- Do you prefer Facebook login, or signing up for an app-specific account?
- How would you like to handle payment? In-person, cash, Venmo, credit card, or something else?

**Motivation (New Tutor):** You think it would be cool to try and offer your expertise in history and English on Tutyr. You decide to enable to tutoring mode for the first time. Your goal is to successfully enable tutoring mode and start waiting for new requests.

[Also, get general feedback on the in-session and ending screens, but no need for a formal walkthrough.]

### Notes

- send message/request shouldn't be separate items
  - wants to be able to request the tutor then make sure the tutor is able to answer the question he has
    - I think the user wanted a more integrated way of having messaging with the tutor and not two different steps?

- SOLUTION: Remove message button, instead have a popup display with a message/comments field and “confirm request” button
- hard time understanding what happens between navigation and finished
  - needs to be a screen that sort of says the session is in progress once the session starts
  - SOLUTION: Use in-progress screen for user with timer(?) [adapt screen-22 for tutee]
- should have tutors that can be seen as “available” or maybe “standby” where a standby mode would indicate this person can be sent a message when available would be for in person. or the tutor can be both
  - SOLUTION: DO NOTHING
- doesn’t understand why the tutors are ranked by distance. thinks that subject is much more important
  - SOLUTION: Move subject to the first thing to filter by, replace plain-text search with dropdown of subjects
- messaging should somehow be integrated in to the request of the tutor
  - so maybe this means sending a message with the request (being forced to send some message)
  - SOLUTION: see #1
- most think they would sign up with facebook if given the option because of the convenience
  - Remove one button, just keep FB button
- about even between people who would like venmo, CC, or cash for payment
  - SOLUTION: Link to Venmo, but the official policy is “work it out”
- when viewing the tutor there are 5 stars, but when rating him there are only 4
  - SOLUTION: Five stars
- wants to be able to make sure the tutor can answer his question before he goes there
  - SOLUTION: See #1
- subject tab is most important tab - OK

## High fidelity prototyping

After this process, we started creating high fidelity prototypes based on our knowledge from user studies using Illustrator. These would be then used as reference material for creating the HTML and CSS necessary for the actual front end.







**Tutyr On Demand Homework Help**

Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit, sed do eiusmod tempor  
 incididunt ut labore et dolore magna aliqua.  
 Ut enim ad minim veniam, quis nostrud  
 exercitation ullamco laboris nisi ut aliquip  
 ex ea commodo consequat. Duis aute irure  
 dolor in reprehenderit in voluptate velit esse  
 cillum dolore eu fugiat nulla pariatur.  
 Excepteur sint occaecat cupidatat non  
 proident, sunt in culpa qui officia deserunt  
 mollit anim id est laborum

THIS IS A BUTTON

**TUTYRONDEMANDHOMEWORKHELP**

Tutyr On Demand Homework Help

**Tutyr On Demand Homework Help**

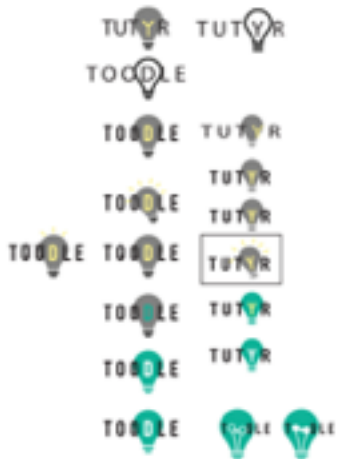
Tutyr On Demand Homework Help

Tutyr On Demand Homework Help

Tutyr On Demand Homework Help

Tutyr On Demand Homework Help

Tutyr On Demand Homework Help



## Tutyr On Demand Homework Help

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

THIS IS A BUTTON



iphone app icon



android app icon



main logo



Header font is Hind Bold

Body font is Open Sans Regular

Button font is Open Sans SemiBold

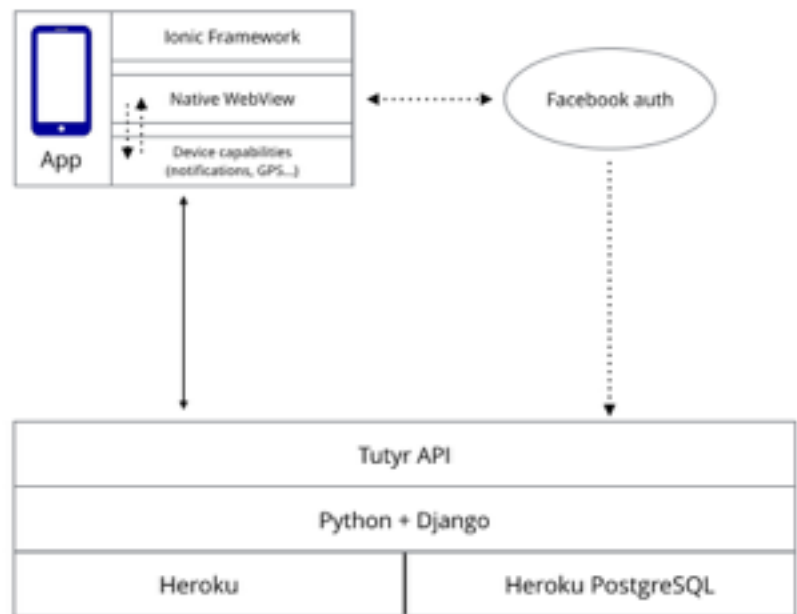
# System architecture

## Front-end

For the front-end, we decided to use a hybrid mobile app framework based on Ionic Framework and Apache Cordova. This combination of frameworks allowed us rapidly develop and iterate on the app and its design using familiar web technologies like HTML, CSS, and Javascript while still giving our app a native feel. As an added benefit, Ionic allowed us to fully utilize the native capabilities of phone (such as location services) while still allowing us to use one code base for all major mobile platforms.

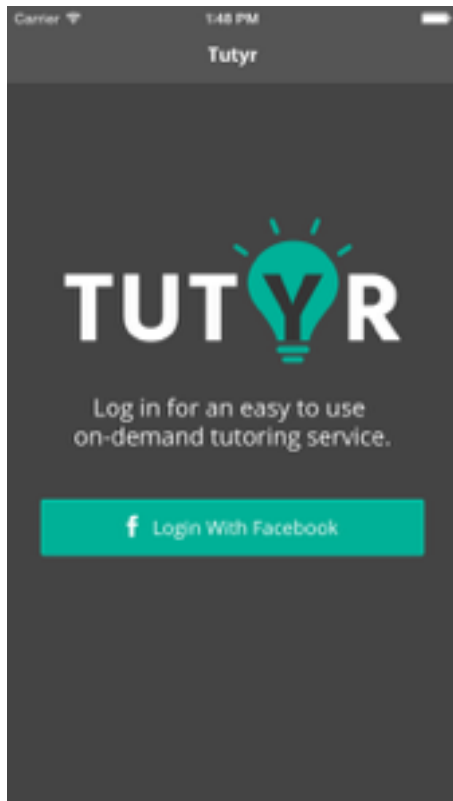
## Back-end

We developed a Django web application in Python to serve as our backend server which utilizes the Django REST framework. JSON requests are sent from the front-end to specific URLs whenever data needs to be retrieved from the back-end database. These requests are serialized and processed by the web application, either reading from or writing to the database as needed. The benefits of using Django is that is easy to add new features, as database queries are abstracted into Python functions, independent of the specific implementations. In order to take our service live, we hosted the back-end server on Heroku, a cloud platform service which provides simple integration of the Django web application using a PostgreSQL database.

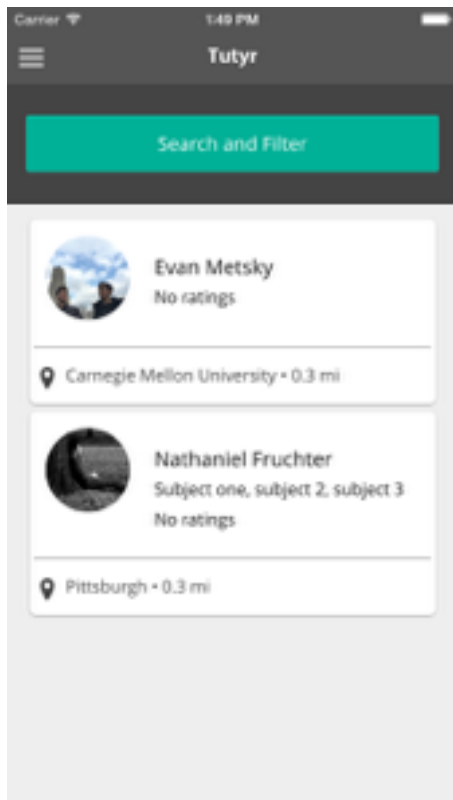




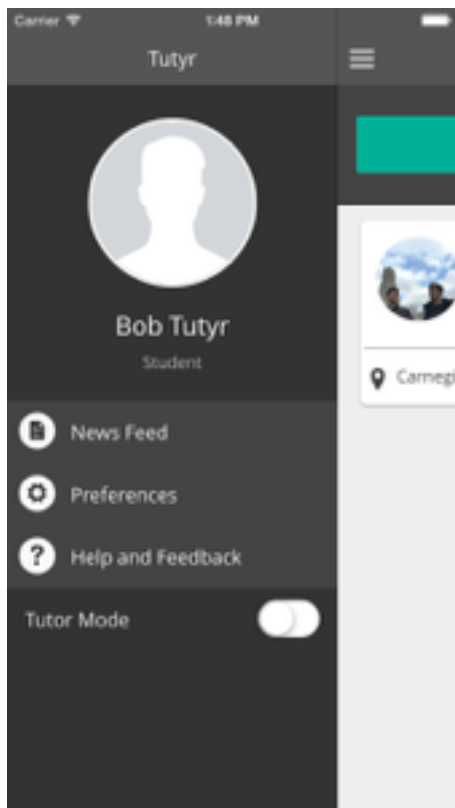
## Features / storyboard - tutor requests



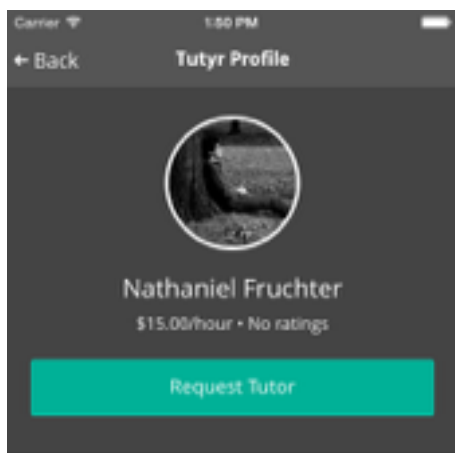
Users are first presented with a login screen where we give them the option to log in with Facebook.



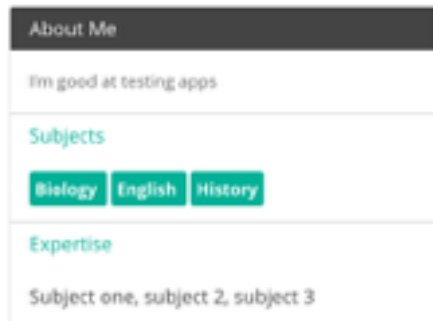
The newsfeed is the first thing that users are presented with. It gives a live feed of tutors currently logged into the app. Users can filter by location, subject, and do a general free-text search to find tutors.

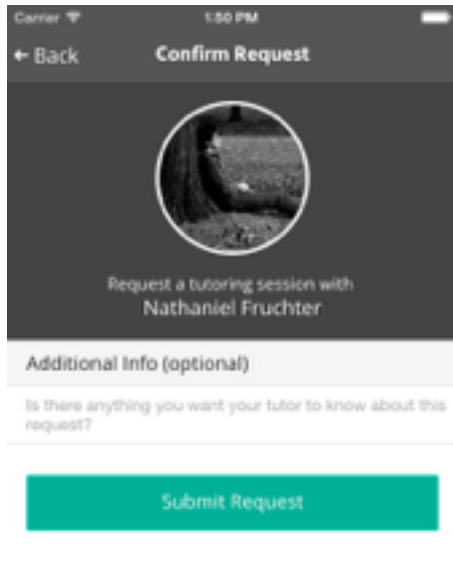


The user mode menu consists of various options that our app provides such as the live newsfeed and switching to tutor mode from student mode.

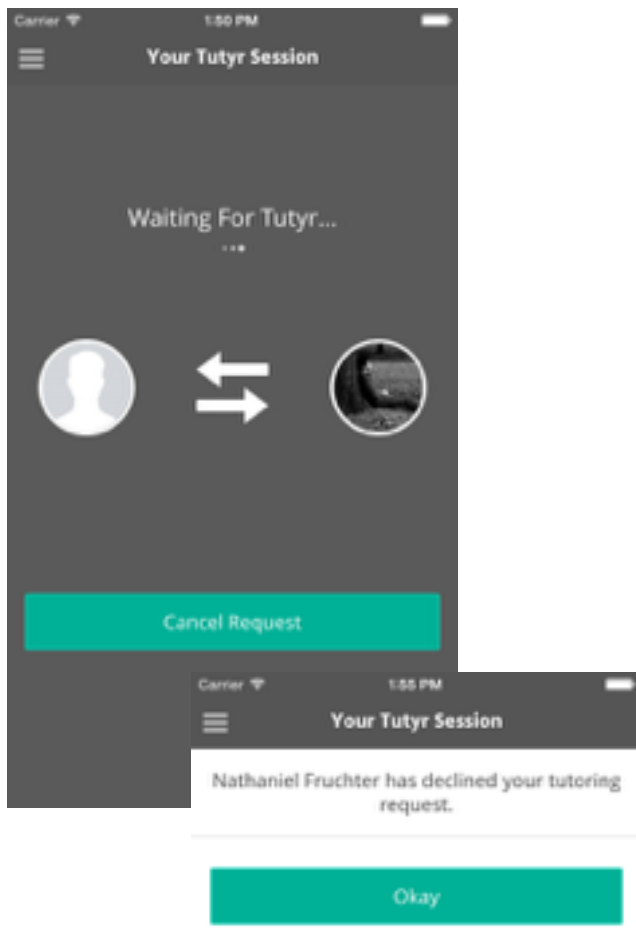


After a user has found a tutor they are interested in requesting, they can view that tutor's profile. The tutor specifies the subjects he/she is good at and thus can help other students with those.



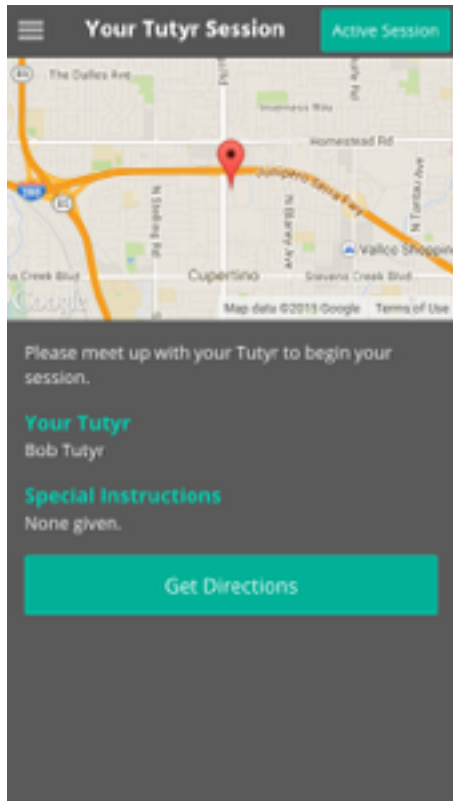


Before requesting a tutor, users can provide additional information to a tutor about their specific request.

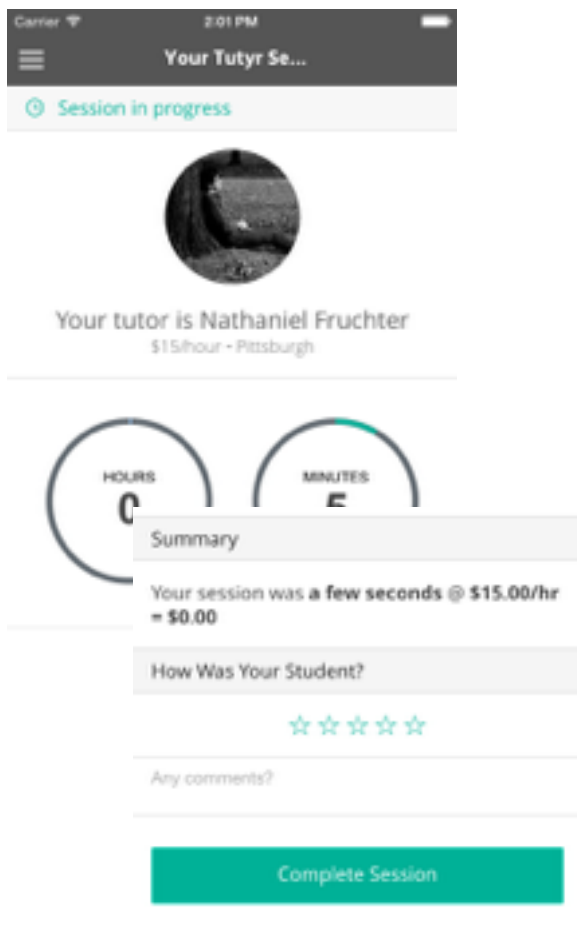


When the user requests a tutor, this waiting screen is shown while the session is pending. The student can cancel the request if they feel the session is no longer necessary.

The user is also notified if their request is denied by the tutor (see below).



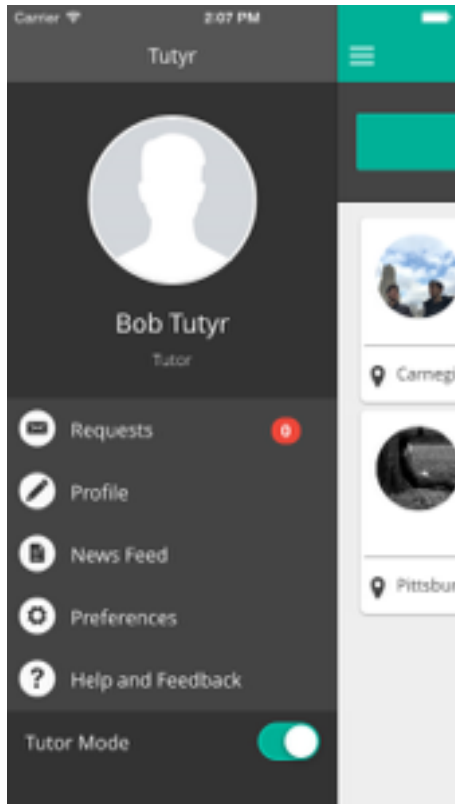
Once a tutor accepts a student's request, the student is shown a map of the tutor's location (if available), and/or custom directions about the tutor's location. We provide the option to navigate to reach the desired location.



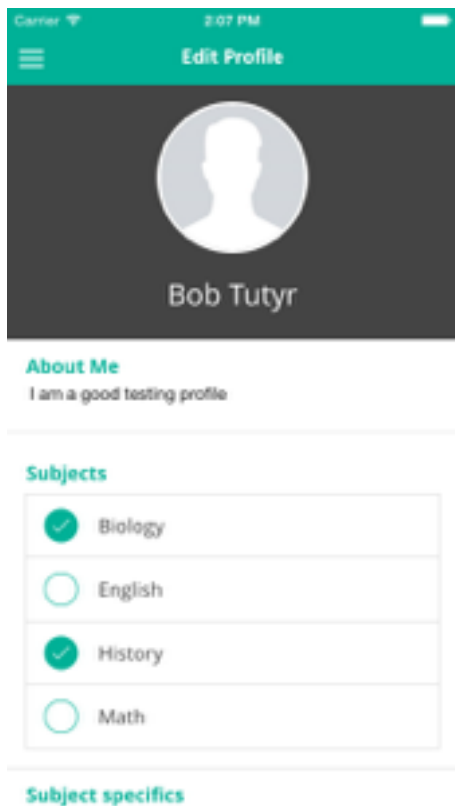
Once a tutor formally starts the session, users are shown a timer with information about their tutor.

Users are then able to rate their tutors and are optionally shown a link to pay with Venmo if the tutor provides an ID.

# Features / storyboard - tutors



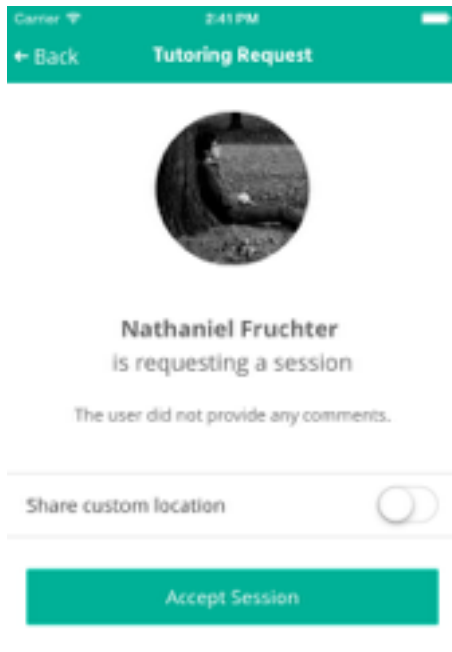
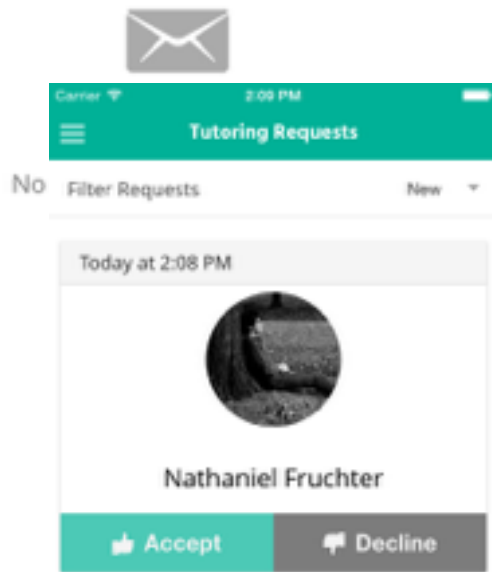
Users can easily become tutors by tapping the tutor mode slider in the side menu. When tutor mode is on, the app's header changes colors to indicate the new state.



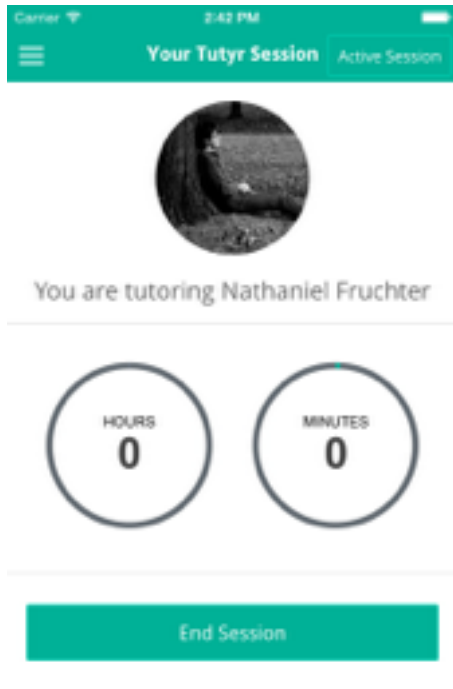
New tutors are redirected to the profile editing screen to fill out their Tutyr profile. They can create a short bio, categorize subjects they're experts in, set an hourly rate, and optionally add a Venmo user name as a payment shortcut.



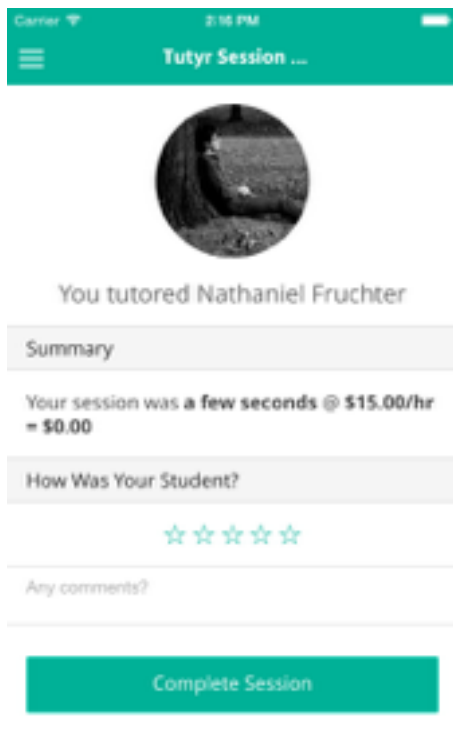
Tutors with a completed profile can view their request inbox. Tutors receive a notification whenever a student requests a session and they are able to accept or decline the session.



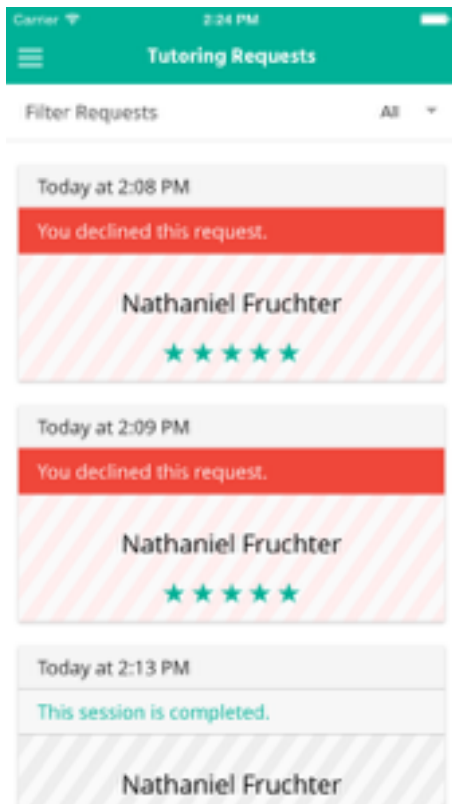
If tutors accept a session, they are able to optionally share a custom location (such as “the third floor of the library”) if they want to be specific beyond their GPS location.



Tutors see a similar session interface to students with a timer. However, they are also given the option to end the session.



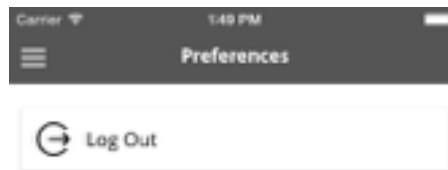
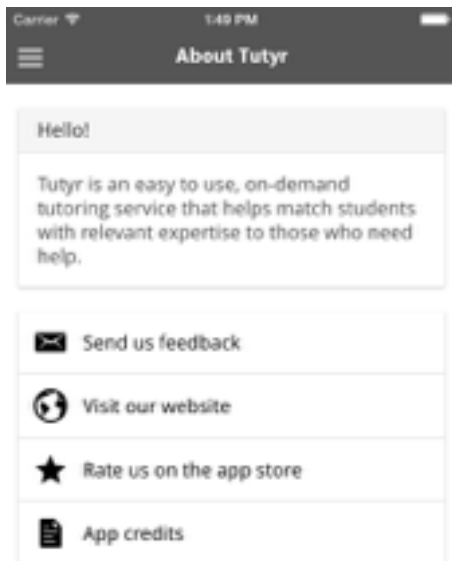
Just like students can rate tutors, tutors are also part of the two-way feedback system and can leave ratings and comments.



Once sessions are completed, tutors can view past sessions through the inbox.

## Other screens

The about and preferences screen are not part of the typical user flow, but are available.





## Further work

Given an extra semester, there are a few things that we would work on, including “micro-tutoring”, better integrating payment options, and further evaluating the overall user experience of the application.

### Micro-tutoring

One valuable suggestion we received during this semester’s feedback process was the idea of “micro-tutoring”—in other words, tutoring related microtransactions. For example, many students may have a one-off question that they need an answer to that wouldn’t warrant an entire tutoring session in person. By integrating some sort of two-way messaging dynamic in the app (such as video chat, text chat, or even some sort of threaded commenting system), we could address users’ needs in this area.

### Payment

Currently, we do not have any sort formal payment integration with our backend and rating system. (Although we support Venmo by providing a payment shortcut, it is not fully integrated with our systems.) Because of this, all payments are done through an informal process that could be enhanced by better integrating payment options. The proliferation of easy payment APIs such as Stripe and Braintree would allow us to streamline the process for the user and tutor, automatically factor payment (or nonpayment) into user ratings, and help users keep track of their transactions in an easier manner.

### User experience

Finally, as we have learned throughout the semester, it is never too late to continue improving and iterating on an idea. Because of this, we could always use extra time to improve on the design, user experience, and overall flow of our app to optimize its usefulness to users. Extra time would allow us to improve user experience details (such as more robust error handling), come up with better interaction methods, and improve the robustness of our backend.

## Try it out!

Our app is live on the Google Play Store. Simply search for “Tutyr” and download it. You can also go to <http://tutyr.me/> for more information. (iOS support is ready, but we did not have time to go through Apple’s longer approval process.)

## Videos

- **Tutyr ad:** [https://www.youtube.com/watch?v=yXh\\_nQ58IkA](https://www.youtube.com/watch?v=yXh_nQ58IkA)
- **About Tutyr / Kickstarter video:** <https://www.youtube.com/watch?v=p5xmPUIHYKk>