

LoopedIn

Stay in the loop - outside the classroom

Maddy Bayer, Simone Jacobs, and Rachel Grau

Problem and Solution Overview

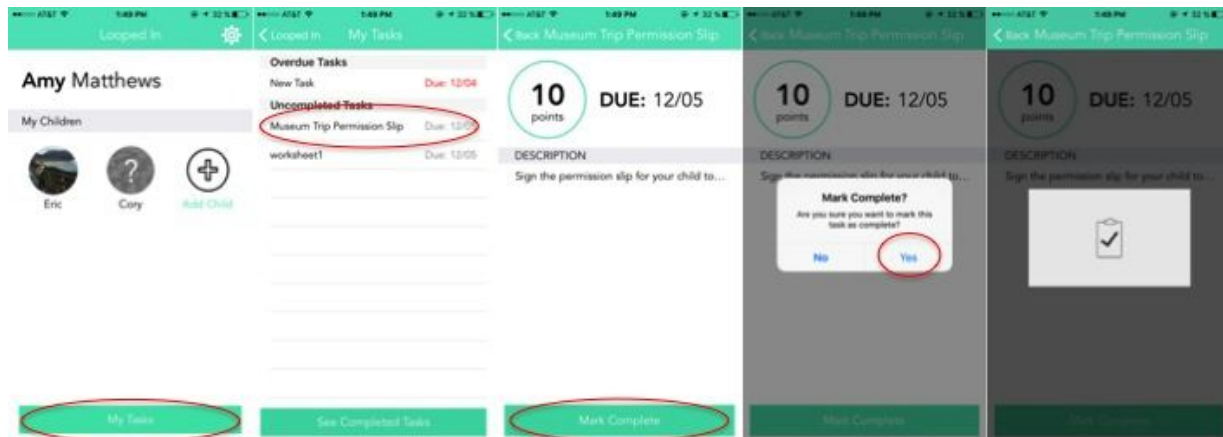
We are LoopedIn and our mission is to encourage positive communication between parents, students, and teachers by streamlining assignment tracking and rewarding accountability. In our needfinding process, we found that the communication that occurs between teachers and parents is almost exclusively adversarial, and that parent-student communication is often lacking in many ways as well. Parents are too busy to get involved in their children's academics, teachers do not have the time to reach out to each student's parents, and students are under too much pressure to relay information back and forth between the two. To address this problem, we made an app that provides a way for teachers to create tasks that students and parents complete to earn points as a team. With enough points, student-parent teams can redeem rewards created by the teacher.

Tasks and Final Interface Scenarios

Because LoopedIn provides a different interface for each user type (student, parent, and teacher), we focused on one task for each group.

Simple Tasks: Parent marks an assignment complete

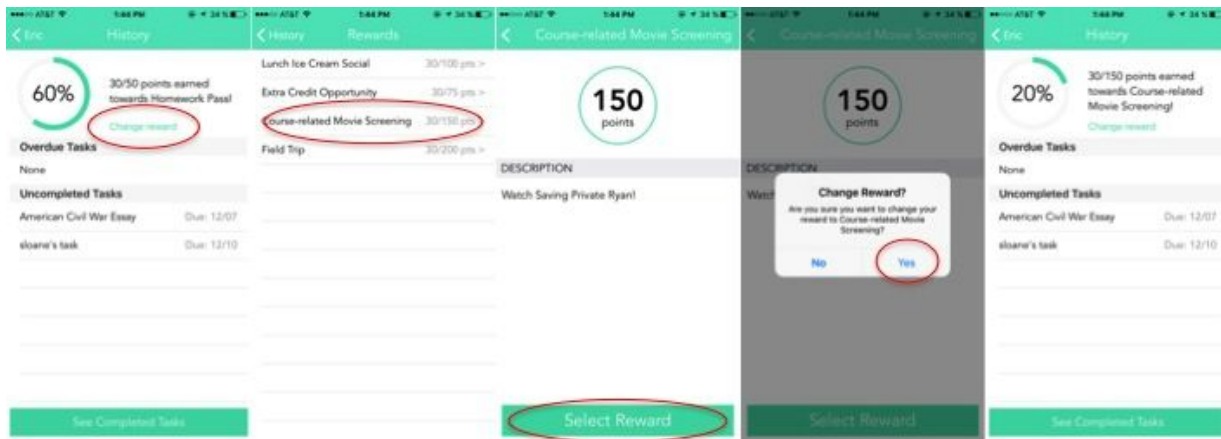
In this task, the parent checks their assignments and marks one of them as complete. We chose this as our simple task because it demonstrates the function of the app at the most basic level. Aside from checking on their children's tasks, this is the main way in which the parent will interact with the app. This kind of task is also very similar on the student interface.



Medium Task: Student chooses a new reward goal

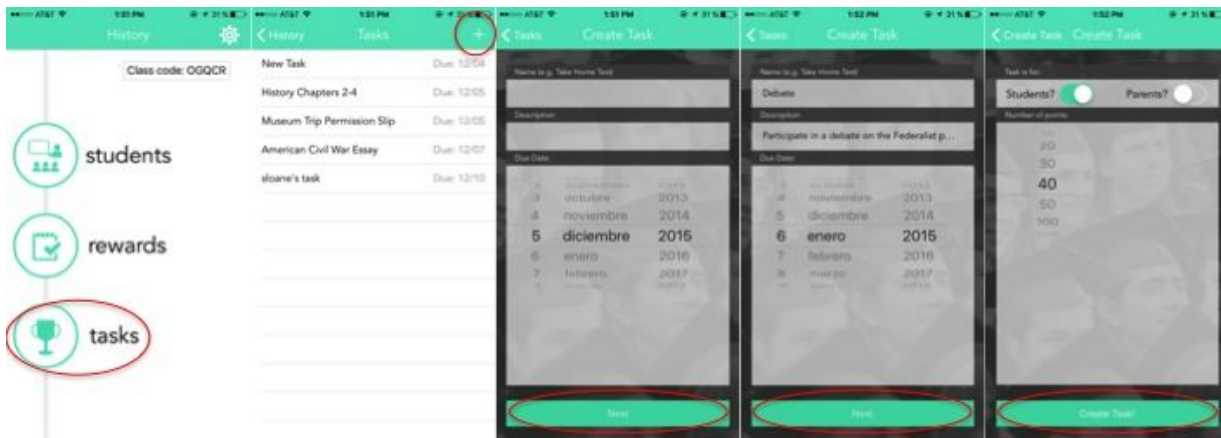
In this task, the student looks at the current reward they are working toward and changes it to a new reward goal. We chose this as our medium task because it requires the user to

consider a lot more information, such as the number of points currently earned, how close they are to achieving their current goal, and what other rewards are available to them. This task also demonstrates the fun element of the app for the student, getting to track progress toward a goal.



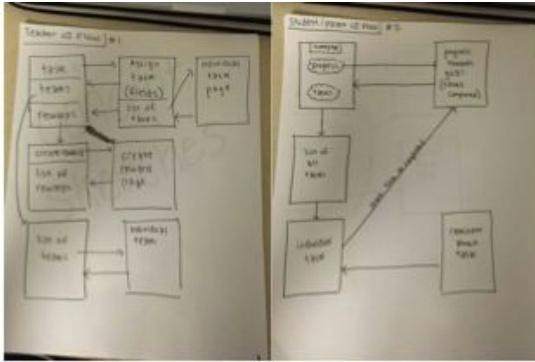
Complex Task: Teacher creates an assignment

In this task, the teacher creates an assignment. We chose this as our complex task because it requires the most input from a user. The teacher has to specify name, description, due date, target assignee (parent or teacher), and number of points before finally “creating” the assignment, when he/she can then track who has or has not completed it. This is an important task because assignment creation is what really populates LoopedIn with content for all users to interact with.

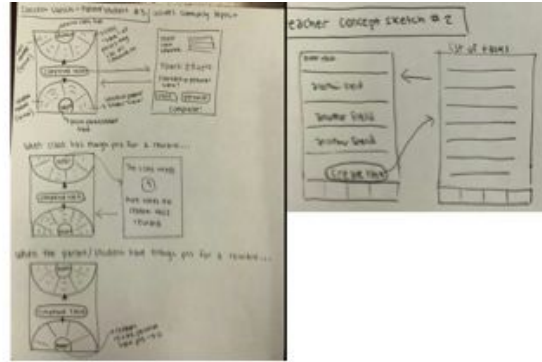


Design Evolution

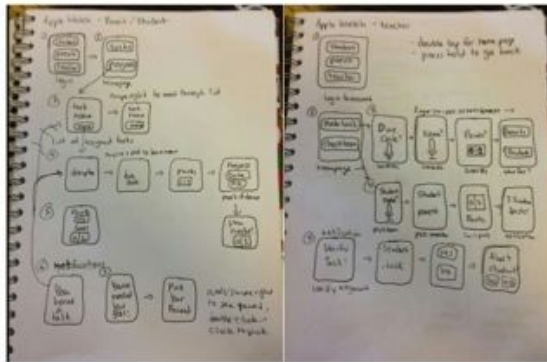
Initial Sketches



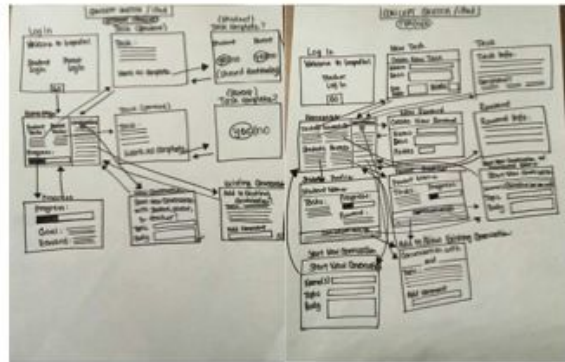
Concept Sketch 1: iPhone



Concept Sketch 2: iPhone



Concept Sketch 3: Apple Watch



Concept Sketch 4: iPad

For our initial sketching we considered three separate interfaces: The iPhone, the iPad, and the Apple watch. Based on feedback from parents, we knew that we wanted to focus on simplicity and ease of use over novelty. For example, one teacher in a low resource community told us that while her students' parents may not have access to a computer or fancy tech like an Apple watch, they were all accessible by their smartphones, leading to our decision to focus on mobile and iOs. iOs is lauded for its usability and familiarity with the general population, and the mobile interface helps address the idea that parents are busy and on the go.

Low-Fi Prototype



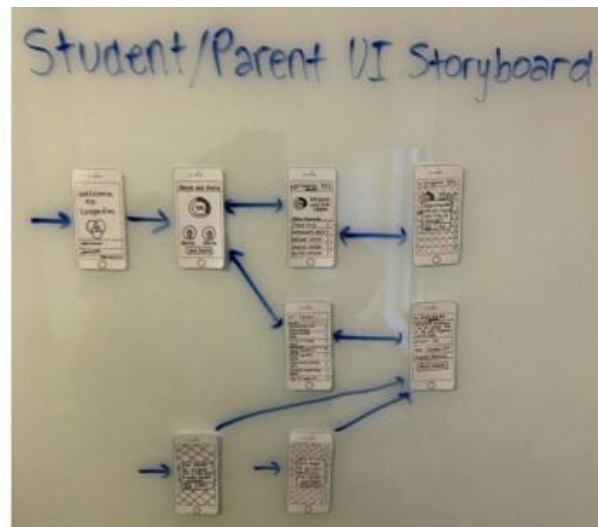
Low-Fi Prototype: Teacher View



Low-Fi Prototype: Teacher Storyboard



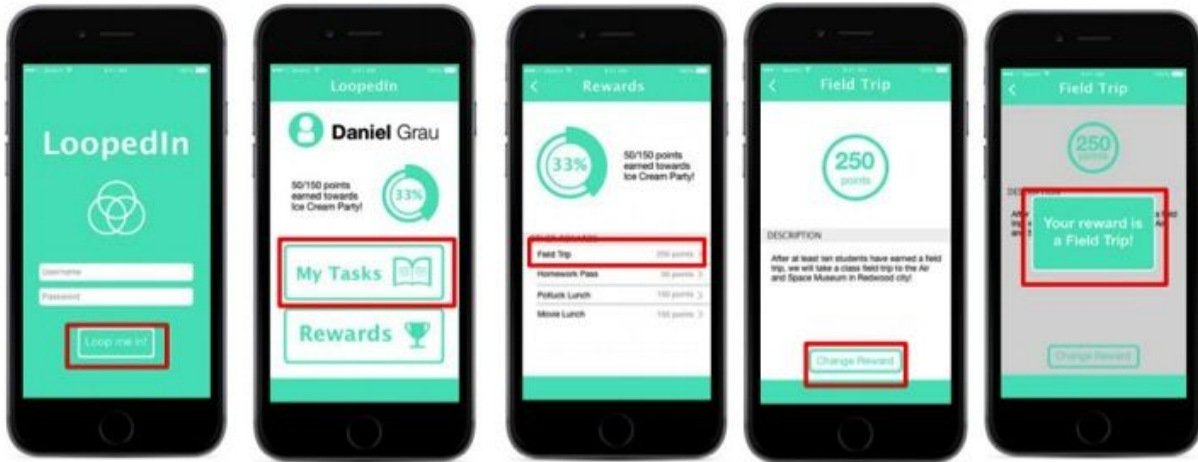
Low-Fi Prototype: Parent/Student View



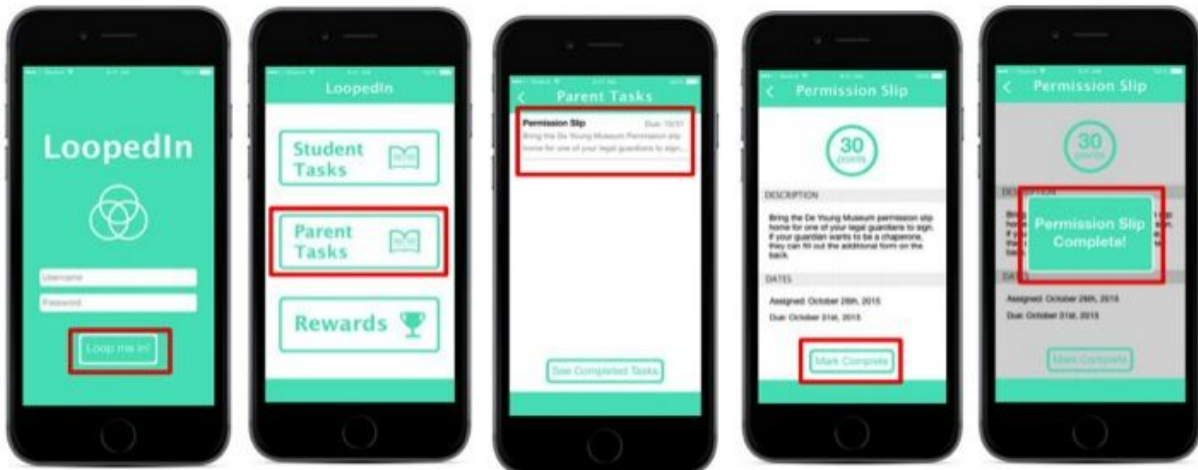
Low-Fi Prototype: Parent/Student Storyboard

For our low-fi prototype we further developed the idea of separate interfaces for different types of users. As is shown in the photos above, we designed a combined student and parent interface in an effort to emphasize teamwork between the two. To keep the interface simple, we tried to see how few screens could be used to make LoopedIn functional, and limit the amount/kind of information on each screen. Because teachers are responsible for populating the interface with content, the teacher storyboard had to be considerably larger than the parent/student.

Medium-Fi Prototype



Medium-fi Prototype: Change Reward/Student View



Medium-fi Prototype: Complete Task/Parent View



Medium-fi Prototype: Create Task/Teacher View

Our medium-fi prototype was heavily influenced by the user-testing we conducted using the low-fi prototype shown above. We ran through our simple, medium, and complex tasks with a student, parent, and teacher respectively, and gained a lot of insight into what our target users liked, disliked, and felt passionately about. We had three main feedback points that we addressed in the medium-fi prototype.

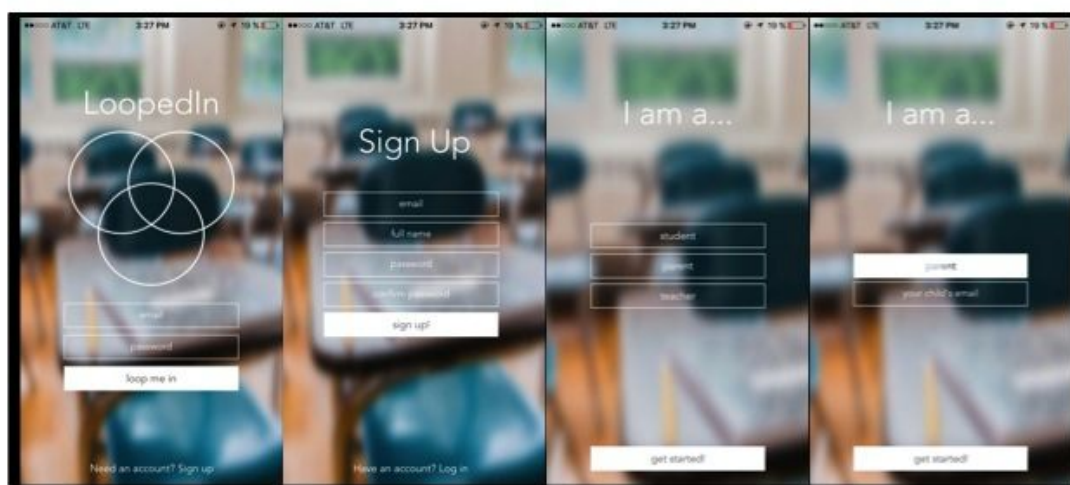
- 1) *Separate role of parent and student.* As you can see in the images above, we separated the student interface from the parent interface. Students can no longer see their parents tasks, and the icons for each user are de-emphasized so that the user no longer tries to interact with them.
- 2) *Real time visualization of progress.* Students and parents told us that they really like the idea of getting to see their progress. In response to this, we emphasized the progress bar and tried to demonstrate how it might change in real time as the student/parent earned points.
- 3) *Focus on the student over the parent.* Teachers told us that they were confused by grouping the parents and students into a team. They did not think of them that way, and were more concerned by checking on their students than keeping tabs on the parents. This realization is reflected in the change in the homepage from “Teams” to “Students”

Major Usability Problems Addressed

Peer-Based Heuristic Evaluation

1) User-Type Specification [H2-3: User Control & Freedom, Severity 3]

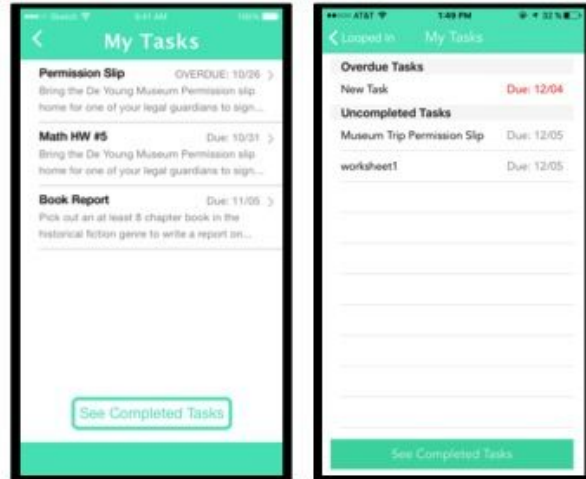
- Problem: Sign-in process does not allow users to specify if they are a student, parent, or teacher.
- Solution: We created a process for signing up and signing in that allows the user to specify what kind of user they are. This process also creates the relationship between students and parents. This way the user establishes their user status at sign-up, and never has to do so again.



High-fi
Sign-In/Sign-Up Changes

2) Overdue Task Distinction [H2-6: Recognition Rather Than Recall, Severity 4]

- Problem: Overdue tasks are not emphasized over in-progress task and need to be made more visible to the user.
- Solution: Overdue tasks were already placed at the top of the queue, but these tasks are now bolded, colored red, and placed under an “Overdue” header



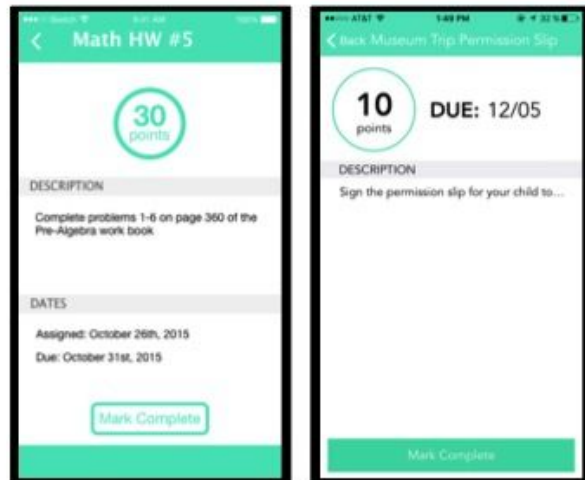
Low-fi

High-fi

Overdue Task Emphasis/Visual Hierarchy Changes

3) Task Page Visual Hierarchy [H2-7: Flexibility & Efficiency of Use, Severity 3]

- Problem: On individual task pages for student users, the task due date is located toward the bottom of the page, making it easier for students to miss it.
- Solution: We have reordered the task page so that the due date is prioritized over the description and points value.



Low-fi

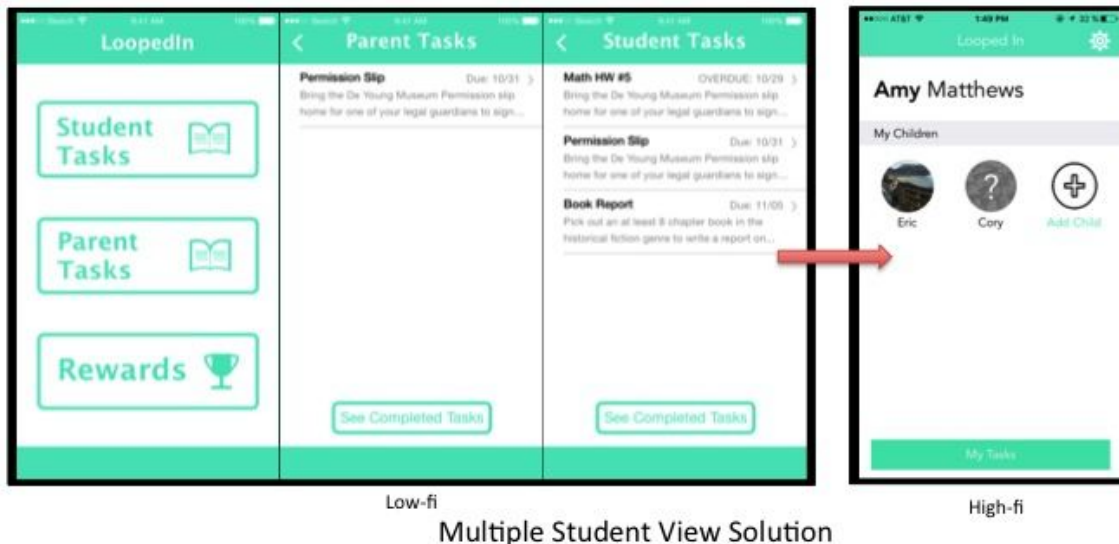
High-fi

Due Date Emphasis/Visual Hierarchy Changes

4) Multiple-Student View [H2-2: Match Between System & Real World, Severity 3]

- Problem: Our hi-fi prototype did not account for parents with multiple children. We need to distinguish which parent tasks originate from which children and a way for parents to view multiple student’s personal tasks.
- Solution: The parent homepage is now just going to feature “My Child’s Tasks” and “My Tasks”. The “My Child’s Tasks” pages lets the parent pick which child they want to view. Instead of having a separate “Rewards” page, which is not very relevant to the parent, the child’s tasks page shows what reward the child is working toward in a single

statement. The “My Tasks” page lists all of their tasks in a single view, in order of due date, but also describes which student they are associated with.

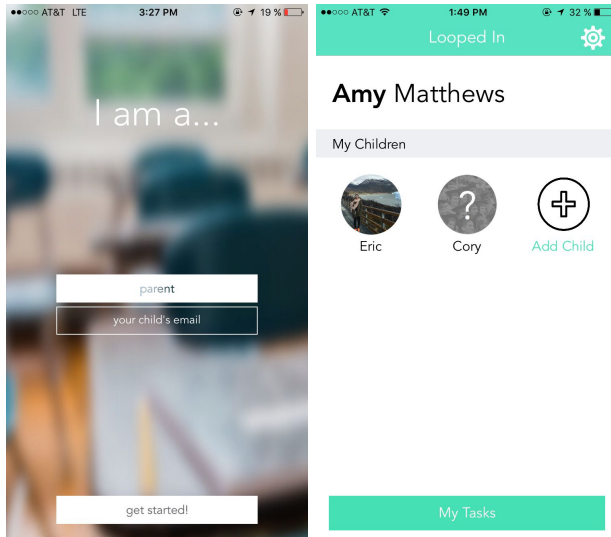


- 5) Archiving/Searching Tasks and Rewards [H2-3 User control and freedom, Severity 3]
- Problem: There is currently no way for the user to search through lists of rewards or tasks, forcing the user to scroll through and search manually.
 - Solution: We have decided not to install a search function. We do not think that so many tasks will be assigned at once that the user will need to search through them. We do not intend for the app to contain so much information that this will be necessary, as it would defeat the point of a super simple interface for parents and students.

Changes Outside of Heuristic Evaluation

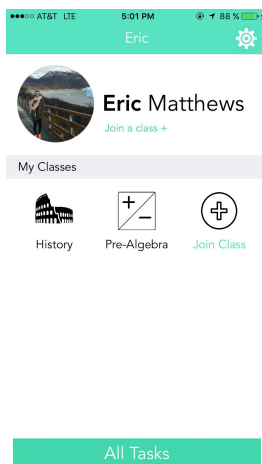
1) Parents with Multiple Children

- Problem: We did not consider how parents would interact with the interface if they had multiple children. This also encompassed the issue that we had not considered how the relationship between students and parents would be created.
- Solution: We completely changed the parent homepage and how parents interact with the interface. Now parents just see “My Tasks” and links to their children’s profiles on their home screen. We also included an entry portion in the sign-up process to create relationships.



2) Students with Multiple Classes

- Problem: Students are often enrolled in multiple classes, and need to receive and track assignments from different teachers.
- Solution: Now teachers can create classes with unique class codes, which students use to sign up for the class. The student homescreen has been simplified in a way that is very similar to the parent homescreen, except that the student view lists classes instead of children.



The two changes above were the most complex, but we also made a lot of small changes to make our app easier to use. These include, but are not limited to:

- customizable icons
- no previews of task/reward descriptions on lists
- phrasing changes to make things clearer
- sleeker log-in page
- specify your user type during sign up
- pop up yes/no to confirm user actions (marking tasks complete, etc)
- less screen real estate given to icons/progress bar
- make tasks editable

Prototype Implementation

We build the prototype using Objective-C/Xcode, with Parse as the backend. Xcode was very helpful, especially the Interface Builder portion, which makes it easier to work with constraints and make the design work. Parse is also a great tool that allows a simple and quick implementation of a backend database. We didn't have any problems with the tools we used.

The only Wizard of Oz techniques were used for the sake of our demo, for which we populated the app with fake teachers, students, tasks, parents, rewards, etc.. We did not have to hard code anything.

There are a few bugs, but those are not involved with our 3 tasks, but more advanced features. For example, there are some issues on the backend with claiming a reward or marking tasks as incomplete (the points total might be incorrect), but the interaction was our priority. Also, if you sign up as a student, you must connect with your parent (e.g. your parent must sign up too) before you can do anything; if you join a class, for example, before your parent signs up, they will not be able to see the tasks for that class. This is also a backend issue; we focused more on the interaction aspects.

There were also a few things we weren't able to implement that we would like to. For example, we would like allow teachers to have to validate assignment completion. Other features we would like to implement: a settings page, a "view all rewards" page for the student (where they can see their progress in every class on a single page), and generally just clean up the backend so everything runs smoother.

Summary

Our team thought that the quarter-long process to complete this project was incredibly worthwhile. Actually putting what we learned from class into practice made the design process feel much more tangible and real. Coming away from this class with an app that could potentially do real good in the world (with a little more iteration), is very empowering. Overall, the most important aspect of our project, and what we think was the key to our success, was our team dynamic. All three of us fulfilled a role on the team that none of the others could, yet we were still able to all contribute in some way to almost every aspect of the project. We consider our studio-mates and our TA, Cathy Zhu, our most valuable assets throughout this process.