



flourish

promote growth, create memories.



view developmental
building blocks



store memories



find resources



Flourish

Rhea P, Elizabeth Y, Jeff Y

I. Introduction

Problem:

It can be hard for parents to know what developmental milestones their kids should be achieving and when.

Solution:

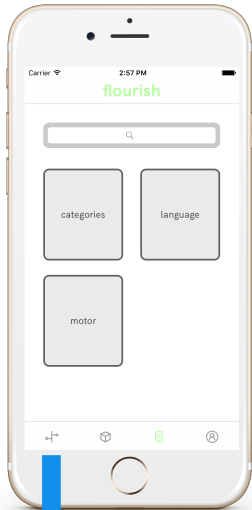
With Flourish, parents will be able to see different tasks that their kids should be doing at their given age and record when their child has accomplished them. Resources linked to these tasks will give parents easy access to information they need. Parents can also store photos and videos associated with individual tasks, giving them a timeline so they can remember exciting moments in their child's life.

II. Task and Final Interface Scenarios

Task One: View Developmental Building Blocks (Simple)

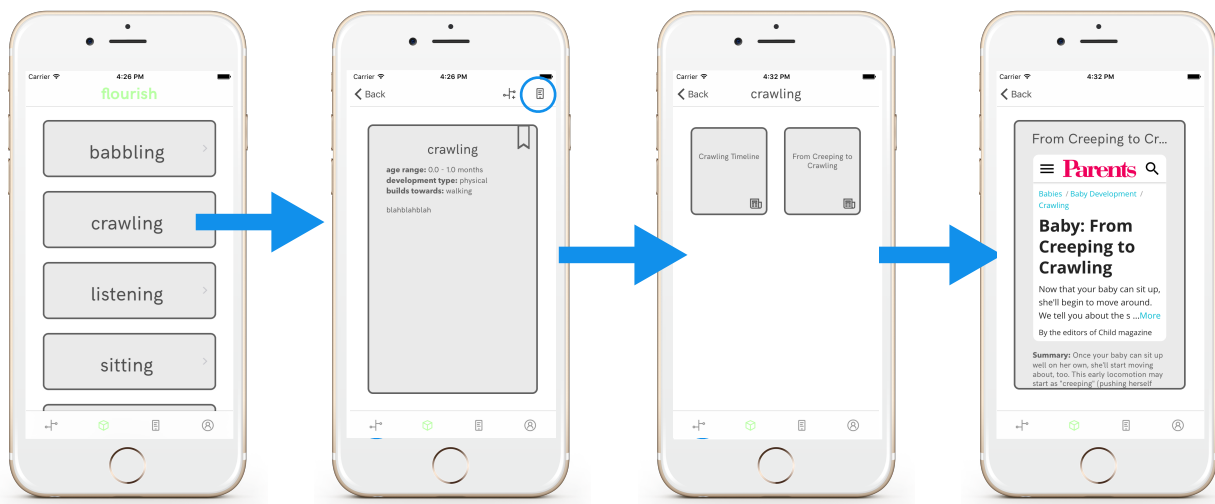
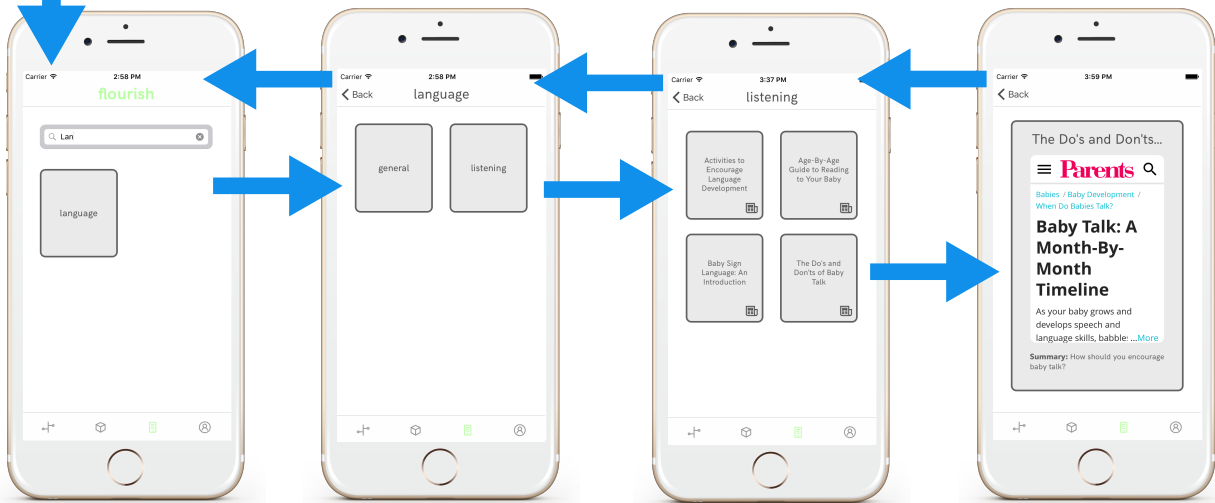
Though simple, our first task is truly crucial to our application. In this task, parents are able to view developmental 'building blocks' for their child, which are generated based on that child's age. These building blocks are meant to give parents a sense of the types of stepping stones that they should focus on to help their kid as much as possible. The building blocks are presented in a list. Each can be clicked on to view more information about that specific block.





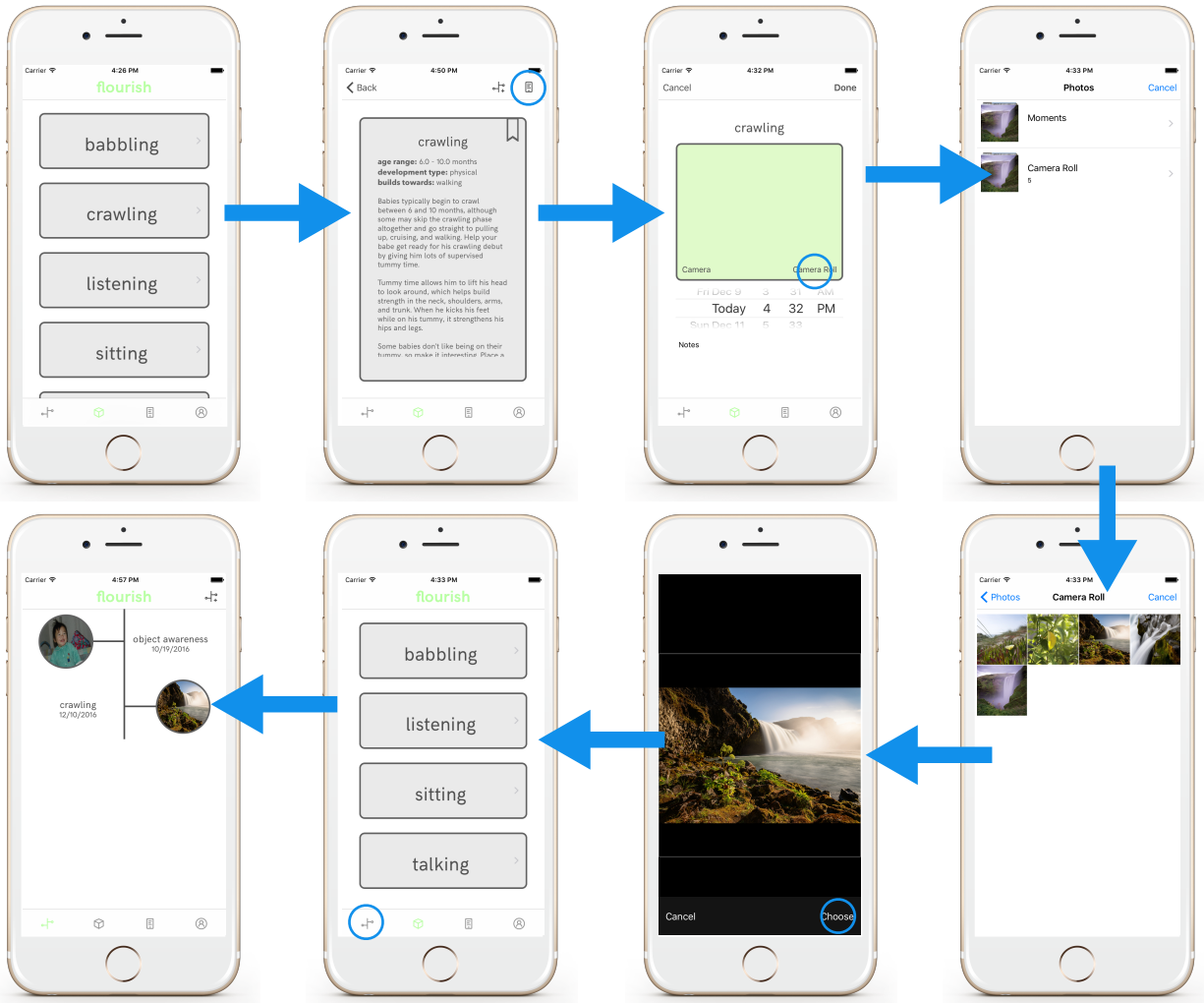
Task Two: Find Resources (Moderate)

The app also links parents to different pediatrician-approved resources so that they can learn more about the different building blocks presented in our first task. We decided to implement this because we want parents to be able to not only understand what types of building blocks to work on with their child, but also give them concrete ways to do that and encourage further exploration into these developmental domains. All resources are available and searchable from the resources tab. Also, if a parent wants to learn more about a particular building block, resources are available straight from the building block's tab.



Task Three: Store Memories (Complex)

Parents love looking at pictures of their children so we wanted them to be able to document their child's life and achievements. When a child completes a building block, the parent can add that building block to the timeline, along with their own notes and pictures.

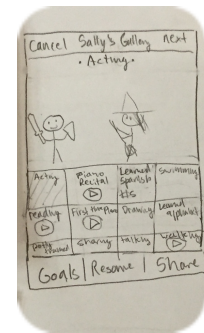


V. Design Evolution

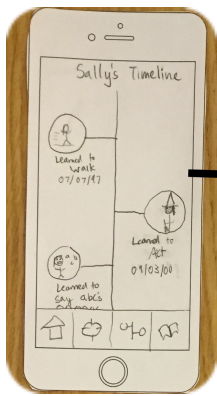
Defining the Core Tabs

We decided early on to do navigation through main tabs that would be located at a bar on the bottom screen. These tabs would lead to the main parts of the app's interface. Of course, that posed the question, what should these tabs be?

Early on, we had defined our three main tasks as “Complete Building Blocks”, “Get Resources”, and “Share Your Memories”. We divided these up into three main sections (see Sketch 1). These idea for tasks grouped the concept of building blocks and a timeline as one idea, putting both on the same page. However, it quickly became apparent that this was way too much information on one page, so we divided it into four tabs (Sketch 2).



Sketch 1

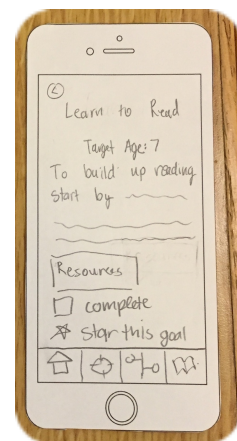


Sketch 2

However, when we did user testing for our low-fi prototype, we got consistently unenthusiastic feedback with the concept of sharing. Though there was nothing wrong with that concept in particular, parents did not feel that it added anything important to our app. With this feedback, we decided to refine our tasks to focus on what we found really important. We realized that seeing memories on the timeline was the aspect of the app that would make people open the app. However, the building blocks were crucial to the purpose of educating parents and helping guide them and we wanted to make sure that the building blocks were there. Ultimately, we decided to define our three main tasks as “View Developmental Building Blocks”, “Get Resources”, and “Store Memories”. We used these tasks as three tabs, along with the profile tab, where user configuration would be done.

Defining Buzzwords

Our group has said the phrase “building blocks” more than any kindergarten teacher this quarter. This is because the buzzwords we use throughout the interface are important parts of our user's interaction with the app. In our initial sketches, we did not think much about the wording. We wanted children to complete goals and when they accomplished them, we added them to the timeline. When testing our Low-Fi prototype, however, parents reacted to our wording in ways that we didn't expect. They thought that words such as “goal”, “milestone”, “target”, “complete”, and “accomplish” were too charged and stressful for the demographic we were trying to reach out to. This made us not only change the words we used in the app from a picture of a target to a picture of a block, it also changed the way we approached promoting childhood development as well. We wanted to make these building blocks as uncompetitive as possible.



Sketch 3

IV. Major Usability Problems Addressed

Landing Page

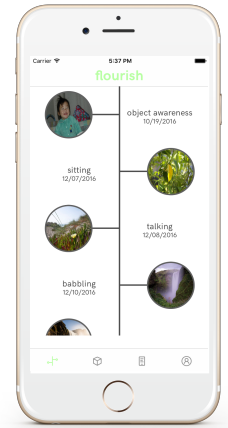
H2-1 Visibility of Status (Sev. 3)



First Landing Page

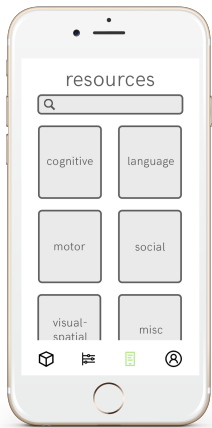
We received feedback that the purpose of the landing page was not quite clear. With no real cues other than some buzzwords, our users did not understand what it was they should be doing. On re-evaluation, we agreed that this might be confusing. We also remembered an interview we had when testing our low-fi prototype with a parent who looked at the timeline and had told us “the amount I would open the app to look at pictures vs. the amount I would open it to complete a milestone is about 80/20, maybe even 90/10.” Other parents we interviewed had also expressed that the timeline is the reason they would keep going back. With this in mind, we decided to change our design so that the first icon on the tab bar and landing page started at the timeline.

Final Landing Page



Resource Categories

H2-2 Match Sys & World (Sev. 3)/H2-7 Flexibility and Efficiency of Use (Sev. 3)



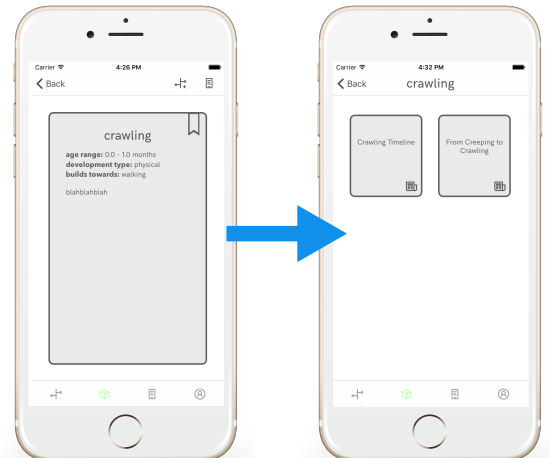
Med-Fi Resources

Some of the users testing our med-fi prototype had concerns about the categories of resources. The users felt that names such as “gross motor skills” were confusing for parents and were worried that parents would have a difficult time locating resources they need. However, we decided to stick with this categorization. We want parents to see, for example, the idea of “crawling” as a building block to success in the “physical” developmental and be able to explore all kinds of resources within that particular domain. However, we did agree with the underlying issue that if you were unsure about finding a particular resource, these domains would be hard to navigate. To combat this issue, we decided to implement a search bar within the resources category and to link building blocks directly to that resource.

Search Function



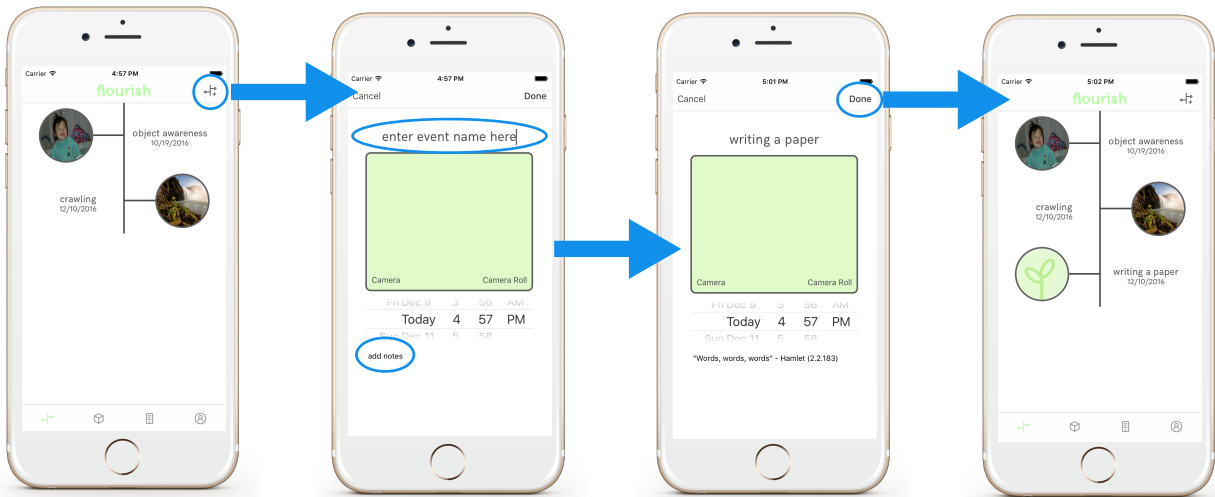
Link Blocks to Resources



Custom Events

H2-3 User Control (Sev. 4)

In our Med-Fi prototype, users expressed that they wanted users to be able to add their own events to the timeline. We agreed! We had not included it on our medium fi prototype since it was not one of our three main tasks, but we did agree it would be beneficial to the app and were able to add it to the Hi-Fi prototype. Note that, though not shown below, the parent can still add pictures from the camera roll in the same way as they can in Task 3 when a building block is added to the timeline.



Block to Timeline Transition

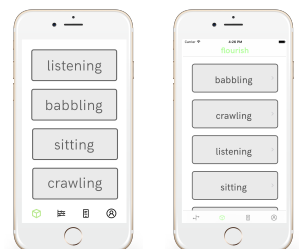
H2-4 Consistency & Standards (Sev. 4)

Our users were confused about the correlation between the building blocks and the timeline. In our Medium-Fi prototype, we displayed our three tabs. However, this prototype did not show the connection between the building blocks and timeline; we had shown the timeline and the ability to see the different memories in detail but had not implemented the transition. Making the prototype, we had seen this as a large gap in our design so far. Refer to our third task to see the final transition.

Screen Titles

H2-4 Consistency & Standards (Sev. 3)

In our prototype, there were titles with 'flourish' on the main page of all three tabs, but not our building blocks tab. This was a mistake, and we fixed it in the main prototype.



Age-Relevant Options

H2-7 Flexibility & Efficiency of Use (Sev. 3)

We got feedback that the options of building blocks should be filtered so that relevant to the age of the child. We absolutely agree; this is something that we would most certainly implement as we continue to develop this project. However, as we will discuss later when talking about Wizard of Oz effects and the fabricated data, these are features we intend to add later.

V. Prototype Implementation

Tools

Swift

We chose to use Swift rather than React Native even though React Native works across platforms because we ultimately decided it wasn't worth using something based on JavaScript. JavaScript is a very unsafe language that has evolved very slowly - I'll take my unambiguous curly braces and doubles, thank you very much. Additionally, our group decided early on in the design process that we were interested in pursuing this project further. We believed Swift is much more likely to exist for the lifetime of our app than React Native.

Firebase

A large part of our app and prototype relied on accessing and storing data. For this, we decided to use Firebase. Our database had three main sections: a section to store each block and its corresponding data types, a section to store information about each resource, and a section to store user information including timeline events and current building blocks. We also used Firebase Storage to keep photos added to the timeline.

A huge benefit to using Firebase is that the platform is easily integrated with Swift using their Firebase SDK which allows the app to dynamically modify the data within. This was particularly important when adding to the timeline, where pictures needed to be uploaded and data added to a specific user's timeline. Furthermore, when looking to extend the app we recognized the benefits of using a trustworthy platform with safe user authentication and a secure database. However, by storing the data remotely it meant that the user must be connected to the internet every time they use the app, or nothing shows up. Furthermore, Firebase event listeners cause a lot of bugs throughout the development of the app.

BitBucket

Bit-Bucket was both a blessing and a curse when it came to collaborating on prototype development. We had many initial issues in this realm. When pushing and pulling storyboard files, git did not work for us because we had multiple people working on the storyboard at the same time. However, we ultimately sorted this out by keeping a few key rules for ourselves as a team. Only one person could touch the storyboard at a time, and we all worked off of the same branch. With this set up, git was useful because everybody had constant access and was able to work on their different sections of the project at the same time.

Wizard of Oz Techniques

The main Wizard of Oz technique that was not yet implemented was coordinating building blocks to the specific age of a child. In theory, when a child is added to the app, the child's age would be entered and building blocks would be generated accordingly. However, in our Hi-Fi prototype we just pretended that the building blocks would be targeted towards the demo-child's specific age.

Hard Coded Data:

There is only one piece of hardcoded data in our application: the user. We did not implement anything related to different users, other using a user id as the key for each user's information in the database. The user id is currently hardcoded to 1, and the user's children are added beforehand in the database.

That said, in order to develop the app we used a lot of placeholder data in the database. In the future, we want the app to have legitimate resources and informative descriptions of each building block. However, since we are not experts and wanted to put our focus into the design instead of the data, we just arbitrarily created these pieces of information. However, given legitimate information, this would only need to be correctly uploaded to the database and it would function as it does now.

Missing Pieces

Within our demo, there are three main missing features that would need to be developed. The first is user configuration. Right now, the demo profile screen shows the ability to toggle between each child; the user can view blocks and timelines for Tessa or Sally individually. However, there is no ability to add children or adjust their ages, which would be very important. Secondly, there is no ability to delete or modify events once they are added to the timeline, which would be an important aspect. Last, we need to get the real data that will populate our database.

There are some improvements related to our three main tasks that we would like to make. First of all, our med-fi design incorporated the ability to add a gallery, not just one picture, to each individual event. We would love to do this, but were only able to incorporate one picture as of now. Furthermore, we want to develop the resources so that there are different types, specifically events near you. Lastly, we want the ability to export and share the timeline event. Of course, this is just a small list of improvements; we all have many ideas of new features and extensions to the application.

Summary

Through interviews, heuristic evaluations, and feedback from prototype testing Flourish has come from an app that only lists a bunch of development goals and related resources into a dynamic app that makes raising a child easier. Flourish not only conveniences parents by clearly suggesting pediatrician approved building blocks for different domains of a child's development but also allows these important moments of a child's life to be stored in an accessible personal timeline so that parents can access these memories forever. Flourish includes custom building blocks to account for possible cultural differences in parenting and timeline events for whatever memories parents find important. With the aim to make life easier for busy young parents, Flourish hopes to provide reliable, easy to follow resources parents can use to get their kids ready for school and, with the timeline, a way to store priceless memories that would otherwise be overlooked in a parent's daily routine.