Babysteps: Reflection on Launching a Mobile App

Emily Best
University of Utah

UUCS-24-006

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

26 April 2024

Abstract

BabySteps is a mobile application designed for newborn caregivers. It helps parents, grandparents, nannies, and other members of a newborn's village stay organized and updated on the child's development. It was developed by a team of four computer scientists as their Senior Capstone Project. We ideated the app in August 2023 and continued shaping its design through October 2023. Development took place from November 2023 through April 2024, and BabySteps is now deployed to Apple's TestFlight and Google Play's Internal Testing services.

We followed an Agile development which progressed through prototype, alpha, beta, and final build phases, with weekly sprints in each phase. Team members brought experience from industry and previous courses and kept learning good software engineering practices throughout the months of development. This project thesis shares the development of BabySteps and reflects on the things I learned through its development.

BABYSTEPS: REFLECTION ON LAUNCHING A MOBILE APP

by

Emily Best

A Senior Honors Thesis Submitted to the Faculty of

The University of Utah

In Partial Fulfillment of the Requirements for the

Honors Degree in Bachelor of Science

In

Computer Science

Approved:

Dr. Aaron Wood Thesis Faculty Supervisor	Dr. Mary Hall Director, Kahlert School of Computing
Dr. Thomas Henderson Honors Faculty Advisor	Monisha Pasupathi, PhD Dean, Honors College

May 2024

ABSTRACT

BabySteps is a mobile application designed for newborn caregivers. It helps parents, grandparents, nannies, and other members of a newborn's "village" stay organized and updated on the child's development.

It was developed by a team of four computer scientists as their Senior Capstone Project. We ideated the app in August 2023 and continued shaping its design through October 2023. Development took place from November 2023 through April 2024, and BabySteps is now deployed to Apple's TestFlight and Google Play's Internal Testing services.

We followed an Agile development which progressed through prototype, alpha, beta, and final build phases, with weekly sprints in each phase. Team members brought experience from industry and previous courses and kept learning good software engineering practices throughout the months of development. This project thesis shares the development of BabySteps and reflects on the things I learned through its development.

TABLE OF CONTENTS

INTRODUCTION

Mobile apps that help with newborn care are becoming increasingly popular (Biviji, 2021). Currently, The Google Play store lists 200 top free apps in their Parenting category, and some newborn tracking apps have over one million downloads (Huckleberry, 2024).

These apps offer various functionalities such as recording daily care (Nara Baby), development tracking (BabySparks), sleep monitoring (Huckleberry), and sharing memories (FamilyAlbum). As parents and other caregivers juggle the responsibilities of caring for a newborn, intentional apps can lighten the cognitive load of childcare and help caregivers stay organized in tracking the baby's growth.

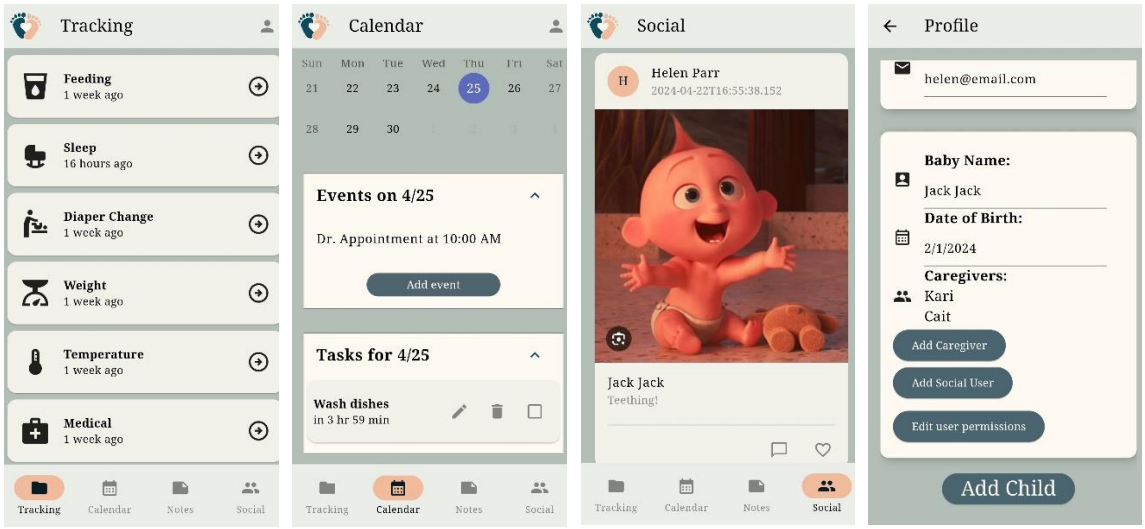
Although newborn care apps can be helpful and are growing more popular, they still have limitations. Several of these apps cost money to download, which deters potential users who are only looking to test different options. Additionally, each app offers different features, so users download multiple apps to have a variety of information and functionality. Some apps do not offer linked accounts, so only one caregiver account can record and see a baby's data. Finally, many of the apps are not customizable, leaving users with cluttered screens and irrelevant information they do not want.

Recognizing these limitations, BabySteps is a comprehensive newborn caregiving app that helps caregivers stay updated and organized on a newborn's development. It was developed as a Senior Capstone Project by a team of four Computer Science students. This project thesis reflects on the development of BabySteps.

Overview of BabySteps

BabySteps has four main pages for newborn care: Tracking, Calendar, Notes, and Social. It also has a Profile page with user account information.

- The Tracking landing page points users to subpages where they can record the following baby metrics: feeding, sleep, diaper, weight, temperature, and medical.
- On the Calendar page, users can save events and create tasks with reminders. It also displays the CDC's relevant development milestones for the baby's age.
- The Notes page allows users to store and edit their own individual notes.
- On the Social page, users can post pictures and text about the babies they care for. They can also see the posts of other users connected to those babies.
- The Profile page stores the user's account information. This is where primary users can connect additional users to their account.



1a 1b 1c 1d

Figure 1: Screenshots of the Tracking, Calendar, Social, and Profile pages.

APP IDEA, DESIGN, & PLANNING

The BabySteps team spent the first 10 weeks of the Fall semester building the app idea, designing its features, and planning its development. We wanted to build an app we were excited about and felt was meaningful. After learning one of our team members was pregnant, I had the idea to build an app that helps with newborn care.

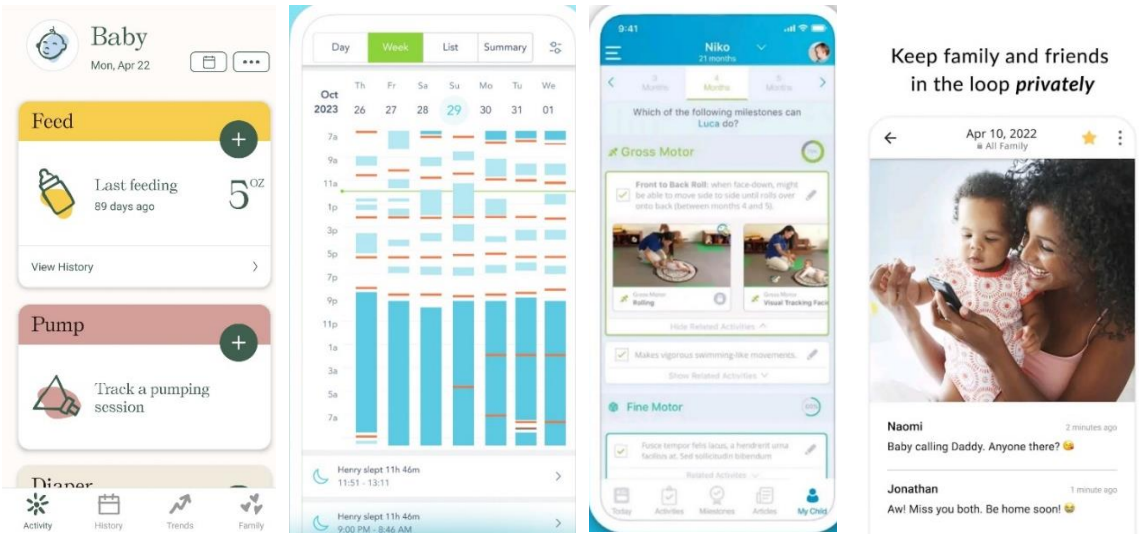
As a team of four women, we have all wondered what life will look like as a mother and a software engineer. Both roles require lots of attention and from our personal experience, we have rarely seen women hold both roles at the same time. Wanting to support our teammate and other working mothers in this exciting transition, we chose to develop BabySteps.

Research and Interviews

After selecting an idea, we spent three weeks exploring newborn care apps on the market and interviewing potential users. This research showed us the usefulness of our idea and helped us focus on features that would be most useful to caregivers.

Market Research

We found hundreds of apps that focus on various aspects of newborn care, such as recording growth metrics, tracking development milestones, and sharing updates with loved ones. Some of these apps have over one million downloads and very positive reviews (Huckleberry, 2024). A few of our favorite apps are pictured in Figure 2.



2a 2b 2c 2d

Figure 2: Screenshots of existing newborn care apps.

2a: Nara Baby, 50,000 downloads

2b: Huckleberry, 1 million downloads

2c: BabySparks, 1 million downloads

2d: FamilyAlbum, 10 million downloads

Most of these apps offer basic functionality on their free downloads with more features and content available on paid subscription. Paid subscriptions often deter potential users who want to test a few options to find the best app for their needs. We liked the variety of features, especially the variety of trackable metrics. However, we wished users could customize their apps to only display the metrics they actively tracked.

The apps generally had clean and easy-to-follow UIs; however, some apps were cluttered with information. Apps that offer activity tracking were often inundated with videos, articles, and written instructions to guide caregivers through activities with their newborn. Although these guides can be helpful, they can also be overwhelming.

Interviews

While we reviewed current apps on the market, we asked potential users about their experiences with newborn care apps and their experiences with newborn care in general. Over these weeks of pre-development research, we interviewed 12 potential users. They included expecting parents and parents of newborns, first-time and experienced parents, and secondary caregivers such as grandparents and nannies. These are some of the questions we asked them:

1. What metrics do you track for your newborn? (sleeping, breastfeeding, etc.)
2. How do you track these metrics? If you use apps, which ones? What do you like/dislike about them?
3. Is it difficult to stay organized and informed with doctor appointments?
4. Is it difficult to update friends and family on your newborn's growth?
5. What resources have helped you learn how to care for your newborn?
6. Who helps you care for your newborn?
7. What would help, or has helped, you stay organized in your newborn care?
8. Would you use a newborn caregiving app? If yes, what would you like to see in the app?

From these interviews, we learned that caregivers track many of the metrics we saw in existing newborn care apps: sleeping, breastfeeding, diaper changes, etc. They often tracked these metrics in notebooks, spreadsheets, or a notes app on their phone. In addition to these metrics, most users record doctor appointments, shopping lists, baby likes/dislikes, and vaccinations. Most users had not considered using mobile apps and said a mobile app that stored all this information in one place would be helpful.

One user had tried using mobile apps but was unimpressed by their lack of customization. The apps suggested articles and activities that did not fit her parenting style, and this information distracted from the useful tracking features of the app. She also did not like its ads.

Overall, users expressed interest in a newborn care app that was easy-to-use, organized, customizable, and free. They wanted an app to store all their tracking, notes, and appointments in one place, and that does not overwhelm them with information. They were also interested in storing baby photos in this app so it could be a one-stop-shop for all their baby information.

Designing the App

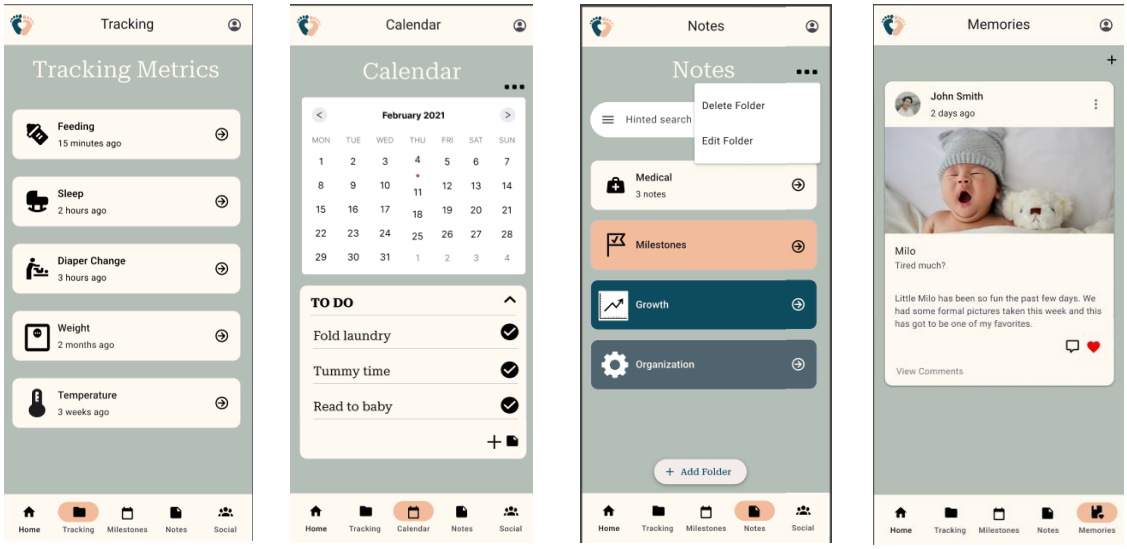
From our market research and interviews, we decided to focus our app's design on tracking metrics and implement other features after this core functionality was strong. We chose to track feeding, sleeping, diapers, weight, and temperature as the main metrics. Users would be able to record these metrics and see the baby's history over time to notice patterns and ensure trends matched those recommended by doctors.

After this functionality was implemented, we planned to add calendar and notes pages. On the calendar page, users would be able to store events such as doctor appointments, and to-do list tasks. On the notes page, users would have default notes for tracking vaccinations, doctor questions, the baby's likes/dislikes, and introduction to solid foods. Users would also be able to add their own notes and folders.

We wanted all this functionality for tracking, calendar, and notes to be accessible for multiple caregivers. All the users we interviewed shared care with their spouse and

occasionally additional family members. We envisioned caregivers having their own accounts connected to the baby's data, where they could record metrics, assign tasks, and note questions as they each cared for the baby.

Finally, we wanted to include a few features that highlight the fun of caring for a newborn. The first months of newborn care often include sleepless nights, isolation from social outings, and a complete change of schedule. Although seeing a newborn grow has many exciting moments, it can also be lonely and exhausting. As stretch goals, our team wanted to implement features that celebrate the baby's development and allow caregivers of the newborn's "village" to share these celebrations. After implementing the core tracking functionality and calendar and notes pages, we planned to implement a feature that tracks development milestones and a memories page where these milestones can be shared and celebrated with all caregivers.



3a 3b 3c 3d

Figure 3: Figma mock-ups of Tracking, Calendar, Notes, and Memories pages.

Planning Development

We broke up our feature ideas into manageable tasks and assigned these tasks to one of our build phases – prototype, alpha, beta, or final. We prioritized the Tracking page metrics, followed by the Calendar and Notes pages, and finally added elements of the Memories page. The tasks built on each other, so we maintained basic functionality throughout development.

With our features outlined and planned throughout our build phases. Our last pre-development preparation was designing our system architecture. We wrote BabySteps with the Flutter framework for its cross-platform functionality. Our team had minimal experience with Flutter and looked forward to learning the framework together.

For our backend, we decided to use Firebase's Cloud Firestore database. Cloud Firestore integrates nicely with Flutter and has an easy-to-use online dashboard. Although we had not used non-relational databases, Firestore has good documentation and we presumed non-relational databases would be easier to adjust in our Agile workflow. We also liked Firestore's user authentication setup and database security rules.

We used Material Design components for a cohesive UI. We also explored a few APIs that could enhance our app's functionality. We ultimately used Table Calendar and Syncfusion Charts in our calendar and graph implementations.

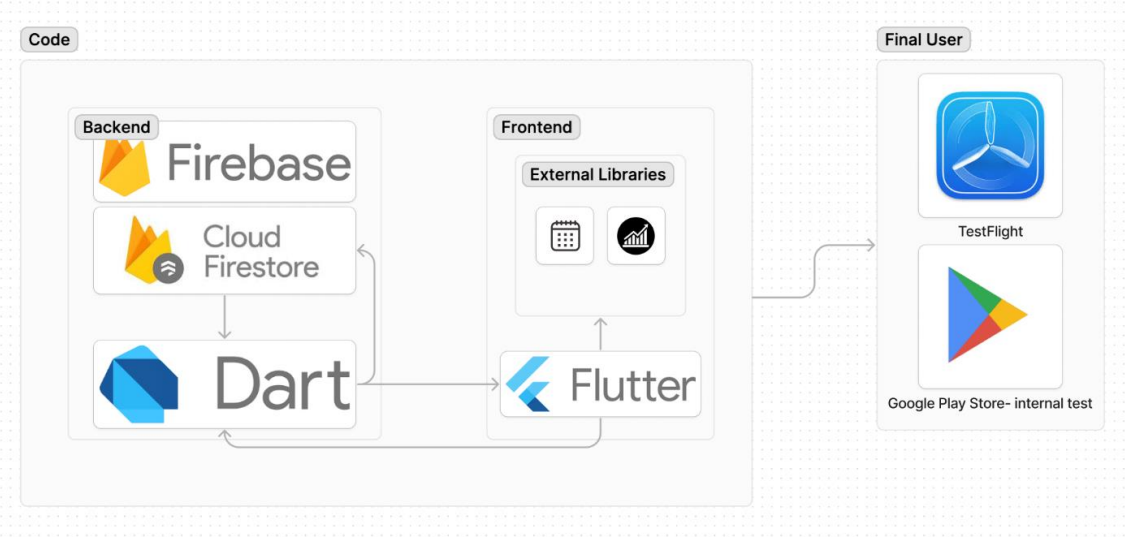


Figure 4: BabySteps system architecture diagram

Reflection on Pre-Development Preparations

Our interviews and market research helped us better understand the space BabySteps lives in. Although we have seen friends and family members care for newborns, our team had limited experience in newborn care. Our pre-development research was crucial to build for user needs.

With so many similar apps on the market all offering different features, our potential users' input narrowed our focus to prioritize features that would be most helpful for newborn caregivers. In the future, we will also include research papers in our pre-development research. After we started development, we found numerous research papers exploring the effectiveness of medical tracking apps, and a few studies highlighted apps relating to newborn care. These papers would have given us more insight into what makes a medical tracking app effective.

Designing Figma mock-ups of our pages was incredibly useful. Figma connected with Material Design components, so our mock-ups resembled the final look of our app.

Figma also supports connecting mock-up images in user flows so we could plan how our app components would connect. These mock-ups allowed us to coordinate the look of our app and ensure we agreed on how the app functions.

Using Cloud Firestore was also a good decision. Although we were skeptical about an unfamiliar database structure, we found non-relational databases are much simpler than the relational databases we implemented in the Databases course. As our app grew more complex, we frequently updated database schemas. Cloud Firestore's non-relational database made these updates very easy.

Our experiences with Figma and Cloud Firestore showed me the importance of having a team with different software experience. Two of our teammates took the Mobile Apps course and used Figma to model their projects. I took Databases with another teammate, and we were equipped to learn how to use a new database structure. Across our team, we had experience with mobile app development, databases, security, UX/UI, and web software development. This variety of experiences strengthened the different areas of our app.

In the future, we will spend more time planning our codebase structure. We had basic separation among the different pages, but identifying common widgets and establishing a system for organizing shared widgets would have improved our codebase. Additionally, establishing a style guide and standard for documentation would have improved our code readability.

DEVELOPMENT

Prototype Build

For our prototype, we focused on implementing the Tracking pages. We prioritized these pages because they are the core functionality of our app. They also were a good starting point to learn Flutter and see the feasibility of our other features.

Each team member implemented one of these pages. I implemented the Weight page, and my code was refactored for the Temperature page as well. I also set up our database and connected basic writes and reads. We quickly learned that real-time database reads were needed to sync user accounts, so I started researching how to implement real-time reads and we added that task to our alpha build.

We set up the Tracking pages faster than anticipated, so we also wrote front-end connections for the Calendar, Notes, and Memories pages based on our Figma mockups.

Alpha Build

My first task in the alpha build was finishing our implementation of real-time database reads. My teammate also implemented user accounts, so we could test that real-time reads synced across linked accounts. I implemented real-time database reads for most of our Tracking pages, in addition to the Calendar and Notes pages.

I also learned about Flutter's form validation widget. Our team had previously decided to implement good security practices to ensure the newborn data was safe. Many parents worry about their child's information online, so we wanted to prioritize data

security. In the alpha build phase, we started getting user feedback, and decided to adjust our planned tasks to prioritize feature development.

Since we were no longer prioritizing data security, I made sure to update any user input fields to use Flutter's form validation. This improved our front-end design for large user input and protected our database from the buffer overflow attacks I learned about in my Computer Security class.

Beta Build

Our beta build was reshaped by user feedback. We simplified the Notes page to make it more customizable and intuitive. We made milestone tracking a stretch goal and updated the Memories page to be an interactive Social page for connected users.

I started manually testing our app for bugs and found several errors in creating user accounts, adding care for another baby, and connecting another caregiver's account. All this code was written by one of my teammates during the alpha build. As I started debugging, I found the code hard to understand and saw that it was built without much architecture planning.

To understand how the components connected and identify the sources of the bugs, I started refactoring our user account code. I mentioned this to my teammate who wrote the original code, however the refactor grew more complicated and time-consuming than I expected.

I eventually ran into an unexpected dead end. I separated the code into different components and needed a child component to send data back to its parent. After

researching online for a couple hours, I learned that Flutter was not designed to support the data transfer I was trying to implement.

At this point, I had been working on the refactor for two weeks without communicating much to my teammate. I brought up this issue and she reviewed my refactor to see how we could restructure the architecture. She agreed that my refactor was more readable and easier to debug, however she was shocked to see so many changes. Especially since her original code also took her two weeks to write.

In the end, we decided to use her original code since she did not run into this dead end and only had bugs in specific use cases. Now that we both understood the code after writing and refactoring it, we were better able to identify and resolve the original bugs.

This was one of my biggest learning experiences in our project. Setting up connected user accounts with multiple babies and data that persists across pages was a very big challenge. My teammate and I had each taken on the challenge alone and tried to brute force a solution. In our post-mortem discussion, we agreed that we should have discussed this problem and drafted an architecture together before starting to implement. Although we did not recognize the difficulty of this challenge before implementing, any new backend data connections are good to discuss before coding. This allows multiple engineers to see potential problems and design a sounder architecture.

We also agreed that we should be more communicative as we resolve bugs. In our programming classes, we do not have two weeks to struggle with an assignment, we have to seek help after two hours of unsuccessful debugging. Instead of putting our heads

down and brute forcing a way through, this problem would have been resolved faster if we both were more communicative during our individual two-week attempts.

Finally, we both saw the value of writing code so that future developers can easily understand it. Class and function comments are essential. Unused code should be deleted. Everyone has bugs in their code, but the author will not always be available to fix them. We need to write code that future developers can build from.

Final Build

During the final build, we polished features, addressed user feedback, and implemented some stretch goals. After adding the ability to manually input stopwatch entries and delete any previous Tracking entry, I started working on voice commands.

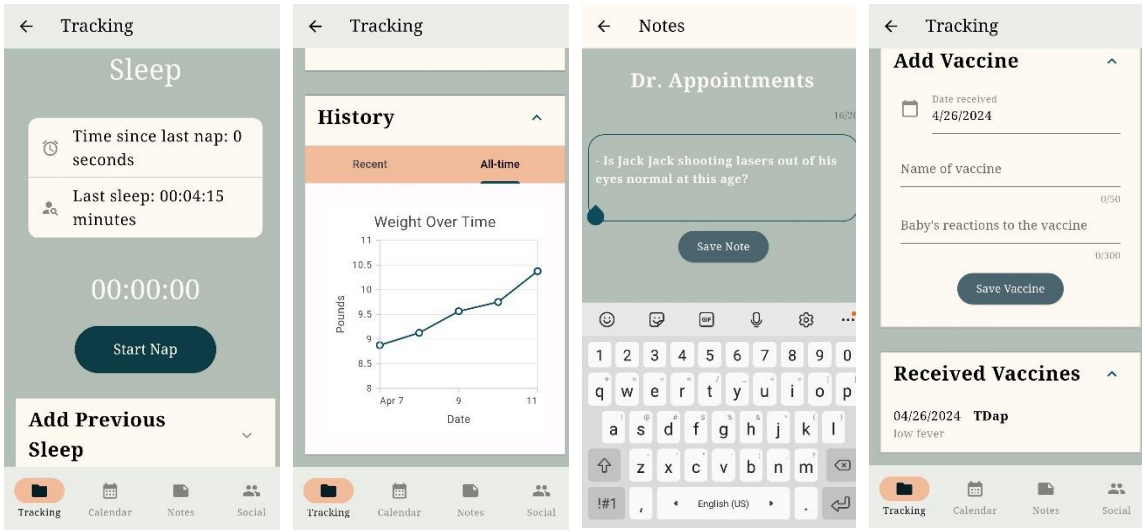
We envisioned our app supporting voice commands for hands-free caregiver metric input. For example, when a caregiver is breastfeeding or changing a diaper, they could say, “Hey Siri, start my BabySteps breastfeeding timer” or, “Hey Google, record a poopy diaper in BabySteps.”

I spent one week researching how to implement these voice commands. I had seen similar APIs on Flutter and hoped it would be a simple API connection. However, these APIs did not support the voice commands we wanted. I found four that implemented voice commands for Google Assistant in Flutter apps. Each tutorial was implemented differently and all required writing native Android code which I am unfamiliar with.

I deployed our app to Google Play Store’s internal testing service to test my code as I worked through tutorials. However, after a few days of attempts I did not have much success. After 1.5 weeks, I decided to drop this feature and implement a different stretch

goal. I did not want to spend another two weeks on code that might not be released. Still, this experience let me practice prioritizing features, triaging their implementations, and learning new technologies. With more time, I know I can figure out how to add voice commands as a future feature.

I pivoted my last two weeks of work to instead implement the Medical Tracking page. Since we decided to simplify the Notes page, the app did not support vaccination tracking by default. I added the Vaccinations and Medications pages, so caregivers can keep track of the baby's received vaccinations and monitor their reactions. The Medications page has similar functionality and was designed for caregivers with newborns that require special care. This special care can be a heavy load on these caregivers, and the Medications page was designed to help alleviate some of the cognitive load of that care.



5a 5b 5c 5d

Figure 5: Screenshots of Tracking, Calendar, and Notes pages.

Versioning System

We stored versions of our codebase in GitLab. Before new code was merged with the main branch, its merge request was approved by a separate team member. The reviewer read the code changes and ran manual tests from their machine's phone emulator. This helped us be familiar with different parts of the codebase and check for edge case bugs. Reviewing my team members code also helped me practice understanding others' code quickly.

User Testing

At the end of the alpha and beta build phases, we shared our app with six user testers and two software developers. They gave us feedback on app features and bugs. Their advice shaped the direction of our app through its beta and final builds.

Many of these testers continued testing our app throughout development and saw how we implemented their feedback. They tested the app on local emulators, or on the Apple TestFlight release which deployed during the beta build.

Overall, users were impressed by our app. They appreciated its clean and intuitive UI. They liked the variety of features and that its simplicity allowed them to customize to their own schedules and needs.

CONCLUSION

BabySteps was a successful and fulfilling Capstone project. The app is deployed to testing services on the two largest mobile application stores and should support hundreds of concurrent users. It fulfills its mission to be a comprehensive newborn care app that helps caregivers stay updated and organized on a newborn's development.

As a team of developers, we exercised good software engineering practices including researching and designing a project, Agile workflows, code reviews, full-stack development, building around user feedback, designing scalable systems, and prioritizing features based on user needs, developer ability, and time constraints. Our experience with these practices will continue to influence our upcoming industry positions.

Overall, BabySteps was a meaningful mobile application to develop and a neat opportunity to apply learnings from various classes we took during our undergraduate degree. We all grew as software engineers and are all proud of what we accomplished.

REFERENCES

BabySparks (2024). *BabySparks – Development Activ* (Version 4.8.4) [Mobile app]. Google Play Store. https://play.google.com/store/apps/details?id=com.babysparks.babysparks

Biviji R, Williams K, Vest J, Dixon B, Cullen T, Harle C (2021). Consumer Perspectives on Maternal and Infant Health Apps: Qualitative Content Analysis. *Journal of Medical Internet Research,* 23(9):e27403. https://doi.org/10.2196/27403

Huckleberry Labs (2024). *Huckleberry: Baby & Child* (Version 0.9.230) [Mobile app]. Google Play Store. https://play.google.com/store/apps/details?id=com.huckleberry_labs.app

MIXI, Inc. (2024) *FamilyAlbum – Photo Sharing* (Version 21.2.2) [Mobile app]. Google Play Store. https://play.google.com/store/apps/details?id=us.mitene

Nara Organics (2024). *Baby Tracker by Nara* (Version 1.43.1) [Mobile app]. Google Play Store. https://play.google.com/store/apps/details?id=com.naraorganics.nara

APPENDIX

Download Links

- iOS: https://testflight.apple.com/join/MUS1mNk8
- Android: https://play.google.com/apps/internaltest/4701690059345031588

Design Document: https://docs.google.com/document/d/1hBFbhBrzJOVDIN-9NAsRlgbzRJVZG-ChfTldR9OEUcA/edit#heading=h.24kwzwtqvuab

Name of Candidate: Emily Best

Date of Submission: April 23, 2024