Skill Builder: Assistive Technology for Developing Skill and Habits

Aaron Kay
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/gradreports
Part of the Computer and Systems Architecture Commons, and the Counseling Commons

Recommended Citation

This Creative Project is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact rebecca.nelson@usu.edu.
SKILL BUILDER: ASSISTIVE TECHNOLOGY FOR DEVELOPING SKILLS AND HABITS

by

Aaron R. Kay

A report submitted in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER SCIENCE

Approved:

Amanda Hughes, Ph.D.
Major Professor

Stephen W. Clyde, Ph.D.
Committee Member

Jared C. Schultz, Ph.D.
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2018
ABSTRACT

Skill Builder: Assistive Technology for Developing Skills and Habits

by

Aaron R. Kay, Master of Computer Science
Utah State University, 2018

Major Professor: Dr. Amanda Hughes
Department: Computer Science

The Skill Builder application is assistive technology for helping individuals build skills through reminders and self-reporting feedback. The application has been built to support Android and iOS devices and followed a user-centric design methodology. Skill Builder’s architecture and development processes are set forth using cross-platform development environments and a native software development kit for the three different versions of the application that were built. The strengths and weaknesses of each of the platforms are explored.

Several studies have been proposed for helping individuals with different needs learn the skills to cope with their challenges. Applications of Skill Builder include students with Autism learning study skills, executive skill learning for vocational rehabilitation, and parents learning to care for children with hearing aids. This report includes the design for a user study for Skill Builder’s effectiveness at helping parents of young children with audiology disorders as they learn the skills required to care for their child.

(41 pages)
Skill Builder: Assistive Technology for Developing Skills and Habits

Aaron R. Kay

The Skill Builder application is assistive technology for helping individuals build skills through reminders and self-reporting feedback. The application has been built to support Android and iOS devices and followed a user-centric design methodology. Skill Builder’s architecture and development processes are set forth using cross-platform development environments and a native software development kit for the three different versions of the application that were built. The strengths and weaknesses of each of the platforms are explored.

Several studies have been proposed for helping individuals with different needs learn the skills to cope with their challenges. Applications of Skill Builder include students with Autism learning study skills, executive skill learning for vocational rehabilitation, and parents learning to care for children with hearing aids. This report includes the design for a user study for Skill Builder’s effectiveness at helping parents of young children with audiology disorders as they learn the skills required to care for their child.
I’d like to thank Dr. Amanda Hughes for working each week to help me move forward, organize my thoughts and deliver a product that is better than what I could have built alone. I’m grateful to Dr. Jared Schultz for the inspiration for Skill Builder and the incentive to work in assistive technology. I’m also thankful to Dr. Stephen Clyde for guiding me through the research process and helping understand graduate school. I’d also like to thank my family for their support over the decades of school.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>PUBLIC ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Research Goal</td>
<td>1</td>
</tr>
<tr>
<td>Project Development Approach</td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>2</td>
</tr>
<tr>
<td>Project Overview</td>
<td>3</td>
</tr>
<tr>
<td>CHAPTER 2 SKILL BUILDER DEVELOPMENT</td>
<td>4</td>
</tr>
<tr>
<td>Paper Prototypes</td>
<td>4</td>
</tr>
<tr>
<td>Mockups</td>
<td>6</td>
</tr>
<tr>
<td>Skill Builder Application Walk Through</td>
<td>7</td>
</tr>
<tr>
<td>Application Development</td>
<td>9</td>
</tr>
<tr>
<td>Test Readiness</td>
<td>17</td>
</tr>
<tr>
<td>CHAPTER 3 FUTURE EVALUATION</td>
<td>18</td>
</tr>
<tr>
<td>Audiology Clinic Study</td>
<td>18</td>
</tr>
<tr>
<td>CHAPTER 4 CONCLUSIONS AND FUTURE WORK</td>
<td>23</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>23</td>
</tr>
<tr>
<td>Future Work</td>
<td>24</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>26</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>28</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Paper Prototypes</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Additional Paper Prototypes</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Balsamiq and Lucid Charts Mockups</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Skill Builder Application Screenshots</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Skill Builder in the Google Play Store</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Ionic Framework Skill Builder Running on iOS Simulator</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>HockeyApp Management Screen</td>
<td>17</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

People often use smartphones to store calendars and reminders. Occasionally, individuals use them to help develop a skill or habit such as taking medicine at a certain time every day. One aspect missing from such reminder systems is an integrated journal or self-reporting mechanism. Users working with mentors or clinicians may keep track of the skill or habit on paper; however, clinicians observe that accuracy diminishes between visits [1]. Such journal record systems are typically recorded on paper and sharing with others can be challenging.

The focus of this project is on developing a smartphone application to tie skills or habits with a feedback journal to facilitate better communication between an individual and a mentor, while also allowing the user to self-reflect on his or her own performance. This application is called Skill Builder.

Research Goal

The goal of this project was to develop the Skill Builder application and gather feedback to:

1. Understand how skill or habit improvement occurs or does not occur with consistent application usage.
2. Evaluate the interface of the application for usability.
3. Measure the impact of a feedback mechanism between the user and the individuals helping the user.

Project Development Approach

Skill Builder development followed a user-centric design methodology [2].
Experts in assistive technology suggested different areas of need that a mobile application could assist with. Then, interviews with potential users yielded required features for a habit learning application. With basic requirements defined, engineers prepared paper prototypes, which were in turn presented to users in a cognitive walkthrough. As design changes were proposed by users, the feedback was incorporated into the next revision of the prototype or mockup. High fidelity mockups incorporating all of the user feedback were then used as the template for the software development. The application was then fully developed as a mobile application using the Ionic development platform.

The Skill Builder software is intended to be used for a user study presented in Chapter 3 which will focus on evaluating Skill Builder’s usability and measuring the increased communication between clinicians and parents who are learning to help their young children consistently wear hearing aids.

**Background**

Assistive technologies such as Skill Builder are designed to support individuals with different disabilities or challenges. Research has shown that when assistive technology follows a multimodal approach for user interaction, user benefit increases over a single modality [3][4][5]. In a survey of reminder and alarm applications on the Google Play Store, many applications supported visual, vibration and auditory alerts; however, none of them supported feedback recorded at the time of the alert. Collecting feedback at the time of experience has been shown to reinforce lessons learned [6][7]. Feedback from users of different smartphone reminder systems suggested that new applications must offer a better experience than the built-in alarm applications [8]. Skill Builder builds on these studies to create a user-friendly multimodal method for users to
create skill alerts and record feedback for skill development.

**Project Overview**

This project report consists of four chapters. This first chapter introduces the project and the approach used to complete the project. Chapter 2 details the design and development processes where Skill Builder is created from paper prototypes and high-fidelity mock-ups to applications that work on Android and iOS devices. Chapter 3 outlines a proposed user study for evaluating the application and the project’s research goals. Chapter 4 concludes with findings and implications for the project as well as opportunities for future work.
CHAPTER 2
SKILL BUILDER DEVELOPMENT

Skill Builder’s development process started as an idea for a class project in assistive technology. After building the prototypes and the conceptual model for the application, the plan expanded to build the Skill Builder application and run a user study with Utah State University students with autism spectrum disorder (ASD).

The development of the application proved to be challenging, particularly when choosing a development platform; the application was built three different times. The first version of the application was built on the cross-platform Titanium framework. After some time with development, it became apparent that Titanium would not support the requirements for the application. Next, the application was built for the Android platform using the native Android software development kit (SDK). Skill Builder worked well on Android devices, but to perform a user study, an iOS version was required (the only available users had iOS phones). The third, and final, re-write of Skill Builder used the Ionic framework so that it could support both Android and iOS platforms.

The process of developing the Skill Builder application, from initial paper prototyping to full-functional mobile application, is described in more detail below.

Paper Prototypes

The paper prototyping process began with investigation into different interfaces for inputting calendars and schedules to create alarms. Several different versions of a home screen, skill and situation input screens, peer screens and alarm screens were created (see Figure 1 and Figure 2). Potential users then evaluated the prototypes for easily understood aspects and areas that did not make sense. As feedback was received,
designs were updated to reflect the state of users’ needs and understanding. Several key requirements emerged from this feedback:

1. The user interface needed to be optimized for (1) adding skills, (2) showing when reminders would happen and feedback would be collected, and (3) viewing reports to adjust skill reminders.
2. Unless fully integrated with a calendar application, the user could not just schedule specific times; this application is not meant to be another calendar.
3. Skill reminders come during certain situations and the application needs to determine if the user is in a given situation.
4. Situations may be based on time, location, or manually enabled.

As these requirements emerged, higher fidelity prototypes also emerged.

![Figure 1: Paper Prototypes](image-url)
Figure 2: Additional Paper Prototypes

Mockups

When the paper prototypes became more stable, we began to build higher-fidelity prototypes, first using Balsamiq and then LucidCharts. The Balsamiq prototypes (see Figure 3a and Figure 3b) were based on Apple iPhone devices (versions 4 or 5) and were mostly limited to wire frames with some navigation. Balsamiq can generate a Portable Document Format (PDF) document including links. This allowed users to interact with the application by clicking on different parts of the document and seeing the prototype respond. The responsive prototype served to simplify the interface and only show things that the user wanted to see. We found that the calendar, while easy to read, was too complicated to maintain.
Our second high-fidelity prototype used Lucid Charts to build the mockups (see Figure 3c-e). This allowed the user to see how the final application might look, but it did not have the benefit of user interaction that the Balsamiq mockup provided. These final mockups became the model for the software that was implemented.

The mockups provided the final version of the user interface. The main screen consists of three tabs, one each for situations, skills, and notifications. The situations tab (see Figure 3c) presents scenarios or situations in which you need to be reminded of the skill a user is learning. Using the skills tab (see Figure 3d), a user adds skills. From either screen the user can tie situations and skills together so the app will remind the user only when the user is in that situation. The notifications tab (see Figure 3e) shows a record of skill alerts and other notifications from the app.

**Skill Builder Application Walk Through**

To explain the purpose and features of the Skill Builder smartphone app, we offer a likely scenario for the app. If a student is having difficulty speaking up in class, a
school counselor might suggest the student work on that skill using Skill Builder and set an appointment for follow-up. The student enters the skill of, “making comments during class,” with a reminder question of, “Have you made a comment recently?” (see Figure 4a) The student then sets the skill reminder frequency to trigger four times every hour that he or she is in class. The student also enters the location of the school so that whenever the student enters the school, the class situation becomes active. Throughout the student’s time in class, the smartphone will send a notification sound or vibration to the student and display the question. Each response to the simple question is recorded so that when the student returns to the counselor, they can review the student’s progress in learning the new skill.

The Skill Builder application has screens for adding and editing both skills and situations (see Figure 4a). Situations may be activated manually, by time (see Figure 4b), or by location. From the list of skills or situations, the user can enable skill reminders for
the different situations (see Figure 4c). When the user is in a situation where the skill reminder should alert the user, the question that the user input will be displayed on the screen along with the vibration or sound alert (see Figure 4d). Even if the phone or device is locked, the reminder question screen will still appear on the display, and the user swipes up or down to record the feedback.

**Application Development**

Through development of Skill Builder, several design choices emerged. Foremost was selecting the platform that would minimize development effort and support the necessary features. Some key features from the development framework that Skill Builder requires are (1) vibration alerts, (2) background processing, (3) data storage, and (4) network connectivity. In addition to these basics, we also expected to add features for gyroscopic monitoring. Basic prototyping in several frameworks and environments using proof of concept programs helped determine if the framework would support Skill Builder’s feature requirements. Because the users seemed equally split between iOS and Android, the developers used Android systems regularly, and Android can be developed for free, Android became the target platform. The Titanium framework emerged from testing as the first choice to use for developing Skill Builder. Through roadblocks and new feature requirements, the development platform shifted to the native Android SDK, and later to the Ionic framework. The development effort on each of these three frameworks is described in the sections below.

*Titanium Framework*

The Titanium framework is built to run in a web browser on iOS and on Android operating systems. It provides a native wrapper around JavaScript, HTML, and CSS.
Many of the native functions such as the notification and location libraries were managed by native plugins that were installed directly into the project by Titanium. After about 3 months of development, only 3 of our 4 requirements worked. The application only supported Android because there had been a lot of customization to the user interface (UI) that would not display properly on both devices. To be fast, the decision was made to abandon support for iOS and focus on getting a working application. The missing requirement was background processing. When Android reboots, a trigger named BOOT_COMPLETED is sent to all applications listening for it. The Titanium framework would not support this feature. So, if a user rebooted his or her phone while using the Titanium framework, the application would refuse to launch. Also, the plugin designed for background processing would not respond if the application was terminated by the user manually. So, if a user closed the program, no alert could be displayed. After 4 months of development, the Titanium version of Skill Builder was cancelled in favor of using the native Android Software Development Kit (SDK).

**Native Android Software Development Kit**

Titanium did not work for Skill Builder because it lacked support for the required features. Android’s software development kit uses the Java language and is maintained and supported by Google. Any application that currently runs on Android can be built using the SDK.

Android SDK development started with recreating the user interface for Skill Builder. Because this SDK is the native language for the Android operating system, the libraries are harder to use, but everything is possible. The SDK also enabled testing on a variety of devices from emulators to different phone models. The native SDK version of
Skill Builder fulfilled all four of the main requirements. Skill Builder was architected after the pattern of many Android applications. The application is divided into two main parts: (1) the app that allows the user to enter skills and situations, and (2) the service which launches the alarms and records feedback.

For an application, Android refers to each main page in the app as an activity, while pieces of the activity are called fragments. Skill Builder uses a tabbed navigation pattern for the three main screens: situations, skills, and notifications (See Figure 3a, 3b). There was a separate activity for each screen, and a reusable fragment for the skills and situations lists to create the rich display and functionality from a common component. In addition to the tabbed navigation, a drawer menu could be pulled from the left side of the screen that would allow the user to navigate to different activities in the app. From the main tab activity, tapping on a situation or skill or the plus button would launch a new activity to create or edit the situation or skill. Both the list of situations or skills and the details of the items were saved in the database.

Android SDK supports a SQLite database via a content provider. The single content provider takes a uniform resource locator (URL) and produces the result. For example, the URL content://edu.usu.cs.skillbuilder.app/situation would identify that the request for data be sent to Skill Builder’s content provider. If a new situation is posted to this URL the content provider would add a new record to the database with the supplied information. If a get request was sent to the same URL, the content provider would return all saved situations from the database. With the database data being served through a content provider, Android could use the same content provider for both the main application and the alarm launching service.
The alert service allowed the Skill Builder application to run in the background to check when a situation became active and launch skill alert questions (See Figure 3d); however, the Android operating system seeks to save battery power by closing down apps that run in the background. Rather than leaving Skill Builder running in the background, the Alert Service starts up, checks the database for a situation that should be active (based on the time or location of the device), causes an alert to trigger if required, tells the Android operating system to start it up again in two minutes, and then shuts itself down. The service scheduler allows the service to run without terminating it, freezing it in the background, or draining the battery.

Skill Builder requires network connectivity to send usage data for a user to interact with professional help. The Android SDK libraries provide a relatively simple method to encrypt the data and allow the user to send the data via e-mail to the researcher. Because Android runs Java as the main programming language, an equivalent decryption algorithm runs as a standalone Java application and uses the corresponding private key to decrypt the data. The Android application does not have a way to decrypt the data, so the data remains secure.

The main disadvantage to implementing Skill Builder in Android SDK is that it only works on Android devices running Android version 4.4 - KitKat and above. Android 4.4 covers most Android smartphones from the last 7 years; however, it excluded iOS users.

The Android application is deployed through the Google Play Store and is currently in the beta test phase of deployment. Users are added with their email address, which allows them to see the app on the Play Store and install it (see Figure 5).
After building the Skill Builder application for Android users, many potential users were excluded from the study because they only had access to iOS devices. Rather than use the native iOS development framework, which carries a steep learning curve for a new iOS developer, the Ionic framework offers support for all of the features required for Skill Builder. Ionic is an application framework based on the open source Apache Cordova project. It uses Angular 2 TypeScript and HTML to build interfaces that compile down to native projects. It runs on both iOS and Android operating systems and supports

**Ionic Framework**

Figure 5: Skill Builder in the Google Play Store
the four main application requirements on those operating systems. The Skill Builder app built on the Android SDK would work for any Android device, so Ionic development efforts focused on iOS devices. Apache Cordova has plugins for most of the native features of iOS and Android. To get those plugins working, they must be imported into the Ionic Framework first. Fortunately, the required packages have already been ported to Ionic.

The architecture for Skill Builder’s Ionic version is based around screens (see Figure 6). There are screens for listing situations, skills, and notifications; a screen for the alert with the question and feedback mechanism, and a screen for reporting the data back for study. Each of these pages consists of three files, the HTML for holding the structure, the stylesheet file for changing how the structure is displayed, and the TypeScript file for changing the data and implementing the event handlers. In addition to the pages, Ionic TypeScript components can be included to make services available across all of the different pages. When the compiler runs on the Ionic project, it compiles the TypeScript files to JavaScript files and packages them together as part of the iOS or Android package. Under the covers, it becomes very similar to the Titanium framework version; however, the main difference is in the plugin support from the Apache Cordova community.

With Ionic’s plugins for the database, notification framework, and background data fetching, and test frameworks that support data transfer, the Ionic project moved forward. The pages in Ionic refer to a DataService component which uses the Local Storage plugin to persist data to the local database on the iOS device. When alerts need to be fired, the NotificationService component sends a local notification to the operating
system so that an alert shows up on the device. To wake the device when an alert needs to be run, the Background Fetch plugin is configured to receive a message from the operating system and check to see if any alerts need to happen and then fire those alerts using the NotificationService.

![Figure 6: Ionic Framework Skill Builder Running on iOS Simulator](image)

The iOS background fetch mechanism works by having the operating system wake up a sleeping process and call a single event. That event is given no more than 30 seconds to execute its code and finish. The caveat is that the operating system is the only one who can decide when background fetch events are executed. During development, the Apple’s XCode development application allows developers to simulate background fetch events to verify that they perform as expected. Once deployed to a device, the device manages when background fetch events are sent and documentation suggests that it is at most once every 15 minutes. If Skill Builder received a background fetch event every 15 minutes, it could schedule the alerts to happen during the next 15 minutes without any issue. However, it did not seem to work this way in practice. On one iPhone
5s, the background fetch events occurred only twice a day. Another shortcoming of the Ionic implementation is that if a user terminates the application, iOS will not call a background fetch for that application.

The Ionic framework builds the iOS application by compiling the TypeScript files and HTML files into a native iOS project which can in turn be compiled by XCode. Once the project is compiled through XCode, it can be deployed to the iOS simulator or can be deployed directly to an iOS device. Apple’s App Store and a web application for the browser are the two ways that apps can be installed on iOS. The App Store is the only way an app can be installed on unlimited devices; however, the web deployment works great for managing testers and installing the app on up to 100 devices. Apple prevents developers from deploying directly to all devices by signing all code that goes on the App Store with a well-known digital certificate. Apps installed through a web deployment are only signed by the developer’s certificate and must include each device identifier for devices the app may be installed on. Several apps exist to help manage users and devices when using the web deployment mechanism. Skill Builder leverages the HockeyApp platform (see Figure 7) to collect logs and feedback with attached metadata from users regardless of user location.

The Ionic app is ready to be deployed to a small set of devices in preparation for testing with users. Users will need to be added via e-mail address and then the e-mail will help collect the device identifiers so that they can be added to the app’s list of 100 allowed devices. In case one of our testers cannot be present to collect the usage data from the device, the HockeyApp send feedback mechanism attaches all usage data for analysis.
Test Readiness

The Skill Builder app has been completed for the Android platform using the native Android SDK. It is ready to be installed from the Google Play Store and can be tested on 94.3% of Android devices [9]. Skill Builder requires an email application to be installed so that encrypted usage data can be sent to researchers. The iOS version, built using Ionic, works while the app is in the foreground and is deployed through the HockeyApp deployment system. Usage data is collected using the in-app feedback mechanism.
CHAPTER 3

FUTURE EVALUATION

Skill Builder has been developed for both iOS and Android smartphones. Several studies have been proposed for helping individuals with different needs learn the skills to cope with their challenges. Parents of high school and college level students with Autism have expressed interest in testing the application. Rehabilitation Education counselors feel that people with disabilities beginning new jobs could use Skill Builder to coach the employee with required executive skills. Also, Skill Builder could be used to help individuals with addiction recovery. USU’s Audiology clinic is ready to utilize Skill Builder for parents of children starting to use hearing aids to help them learn the habit of their child wearing hearing aids on a daily basis and much of the work for IRB approval has been completed. The rest of this chapter focuses on a study prepared for USU’s Audiology Clinic.

Audiology Clinic Study

The Audiology Clinic at Utah State University serves many children across Utah, Idaho and Wyoming who have moderate to severe hearing loss. Clinicians work with the parents of young children with different audiological disorders. In particular, clinicians note that parents often struggle with ensuring that their children with hearing impairment are consistently wearing their hearing aids. Establishing a habit of making sure one’s child is wearing his/her hearing aid is a process that we hypothesize could benefit from the use of Skill Builder. Skill Builder can be configured to periodically ask the parent if their child is using their hearing aid. These prompts will remind the parent to help their child put on the hearing aid if they are not currently using it. In addition, the parent
responses collected through Skill Builder could then be shared with the clinicians to evaluate how they are doing and identify ways to improve.

Research Population

Some parents of young children (0-3 years) with hearing impairment often require assistance remembering to put their child’s hearing aids in each morning. In this study, researchers would work with these parents and their clinicians to trial the Skill Builder app as a way to incorporate new routines throughout the day as they train their children to consistently use their hearing aids. The audiology clinic would also like to use the feedback recorded by Skill Builder to tailor their interactions with that parent.

Researchers would work with both clinicians and parents to evaluate Skill Builder’s usefulness. Researchers would work with 5-10 parents who are primary caregivers and the 2-3 clinicians at USU’s Audiology Clinic who work with those parents. The clinicians will identify and recruit parent participants.

Research Methodology

The research will begin with hour-long interviews and Skill Builder training for the staff / clinicians who regularly work with the target population to teach the researchers expected usage patterns for the application and help them know how to best train users on Skill Builder.

Each patient or parent participating will complete a 30-minute interview to train on the app and gather first impressions, followed by a three-week trial. During the trial, the application will record metrics such as where the user tapped, how long they used the app and the types of actions that they performed during each application usage session. Also, times and details will be recorded for the phone alerts and the user responses. After the trial, researchers will conduct a one-hour exit interview to retrieve the metrics from
the participant’s phone and gather impressions and frustrations with the application.

During the three-week trials, researchers will perform a mid-trial interview with the staff / clinicians most familiar with each participant. After all participants have concluded their trial, we will again interview the staff at the clinic to gather feedback and final impressions.

In addition, researchers will work with the audiology clinic staff to collect hearing aid usage data before, during, and after the trial. This data will allow a comparison between self-reported use of hearing aids (collected from Skill Builder) with actual hearing aid usage.

Data Collection

This section summarizes the different types of data to be collected:

Skill Builder App Data

1. Skills entered
2. Situations entered
3. Time reminders were alerted and associated user responses
4. Screen tap locations for all app screens
5. Time spent on each app screen

Additional Data

1. Hear aid usage data
2. Interview data & feedback from clinicians (See Appendix A)
3. Interview data & feedback from parent participants (See Appendix A)

Data Analysis

First, the data will be scrubbed to make sure there is no personally identifiable information when it is gathered from each device. Researchers will transcribe all of the
interviews for data analysis.

Next, data will be analyzed, particularly the interview data, using grounded theory to look for patterns and themes. This analysis will be helpful in evaluating the impact using Skill Builder has on the parents as well as the interactions between Audiology clinic staff and parents. To supplement this more qualitative analysis, researchers will also use the following quantitative metrics:

- Skill growth – did the affirmative answers increase or did consistent affirmative feedback decrease in frequency
- Complexity of activity – length of time spent in a particular part of the app
- Appropriate alarms – missed feedback opportunities divided by total feedback opportunities
- App flexibility – distribution of time spent using app in the different parts of the application
- Quality reports – correlation of report views with skill growth metric over time
- Reporting accuracy – correlation between self-reported usage and hearing aid data

**Summary**

This study has been approved by Utah State University’s IRB and the Audiology Center’s clinicians have already provided valuable guidance for the Android version of Skill Builder. The clinicians showed excitement about helping their patients and could think of several who might benefit from using Skill Builder. At the time of this report, the only people identified by the clinicians as potential users had iOS devices; however, by
the time the iOS version was complete, the window for user testing granted by the IRB had expired. It is recommended this study be conducted when all of the bug fixes identified in *Lessons Learned* (in Chapter 4) have been implemented in the Android and iOS versions of Skill Builder.
CHAPTER 4

CONCLUSIONS AND FUTURE WORK

This Skill Builder project provides a mobile learning feedback application which reminds users of skills and records simple feedback for how they are doing with a particular skill. It was designed using a user-centric methodology and developed to work on both Android and iOS devices. This project also proposes a user study to evaluate Skill Builder’s effectiveness and usability. The remainder of this section discusses the lessons that were learned while developing the Skill Builder application and future work for the application.

Lessons Learned

The Skill Builder development process taught three main lessons about the state of mobile application development. The first lesson is that when developing a novel interface to smartphones, do not rely on cross-platform frameworks to work for those novel features. The frameworks are built to make web developers effective at producing common interfaces with custom styling. Expandable lists cause cross-platform development delays because Apple and Android use a different style for expanding list items, and the frameworks do not support either style very well.

Even when plugins are available, the cross-platform frameworks struggle supporting plugins for doing something the same way on iOS and Android. It reduces to difference in implementing key operating system features like background processing and scheduling. The background-fetch plugin worked for iOS occasionally, but it did not work on Android because Android has a completely different paradigm and plugin for implementing background processing.
The third main lesson for developing using a cross-platform framework relates to user interface testing. Android and iOS share about 70% of the user interface design elements. The 30% difference causes users familiar with one operating system to be completely lost when using the other. The components that are included in the cross-platform frameworks sometimes look like Android components and other times look and behave like iOS components. Having an Android interface on an iOS device causes the user to become confused as the paradigm is even different for the user.

**Future Work**

Skill Builder is designed for users. Testing with those users remains the highest priority for future work to evaluate its effectiveness in helping users develop desired skills. This section covers these future evaluations as well as application enhancement requests from behavior councilors and audiology experts.

**Future User Evaluations**

The study presented in Chapter 3 is ready to be executed. The iOS version of Skill Builder needs to be updated so that alerts come as expected. While we have already interviewed the clinicians, it would be prudent to meet with the audiology clinicians once more to make sure they have patients who could benefit from Skill Builder and that they have access to hearing aid usage data to compare with Skill Builder user-reported usage data.

**Skill Builder Feature Enhancements**

While researching requirements for the Skill Builder application and in demonstrating its uses to the audiology center clinicians and special education councilors, a few features emerged as helpful but not required features. The most requested feature is
a more robust feedback mechanism that can reinforce behavior rather than just record and report. For example, if the user responds they have not practiced the skill, a secondary question would follow-up with a text entry field to record the reason they did not practice the skill. Also, suggested was a decision tree of questions that could be presented to the user. For example, if the user responds “yes,” then the application would ask follow-up question number one, if “no,” then it would ask follow-up question number two.

Skill Builder is designed to be a social application for the user to self-reflect and allow mentors such as parents, councilors, and doctors to help tailor the alarms and feedback questions specifically for the user at any given point in time. For the user to see trends over time, the reports feature should be implemented. Also, reporting data should be shared with mentors to assist in modifying the frequency and/or wording of the feedback question. To build in the social aspect, security and a networking infrastructure needs to be in place so that privacy is retained with potentially sensitive information. The data that Skill Builder maintains may fall under national health record laws, so if social connections and data sharing are allowed, proper legal procedures should be added as well to protect the user and any mentors.

One feature that would be valuable to explore is how Skill Builder receives feedback from the user. The Titanium version of the app supported slapping the device while in a user’s pocket to record the feedback for yes and no answers to the question. It often recorded false positives, but it could be expanded to allow a user to receive a skill reminder and respond without informing people around the user.
REFERENCES


APPENDIX
INTERVIEW GUIDES

Opening Interview guide for Audiology clinic staff

1. Explain the study and obtain informed consent

2. Explain the purpose of the Skill Builder App and how it records feedback from customized reminder prompts

3. How often do you meet with the parents of children in the clinic?

4. What do you ask the parents to do in these sessions?

5. When might you have parents use Skill Builder? (at the beginning / only when there’s problems / etc.)

6. What skills should be used?
   a. What types of reminders do you see parents of children in the clinic using?
   b. How would you phrase yes/no questions for parent feedback?

7. When should reminders and feedback be gathered?
   a. Where do parents typically need reminders?
   b. Are there certain times of the day when they need reminders?
   c. What other criteria could be used to trigger reminders?
   d. What frequency do parents typically need reminders for each skill / task?

8. What feedback would you like to receive?
   a. How often do you like to see the data regarding the parent’s usage?
   b. What types of reports would be helpful?
      i. Response rate
ii. Yes/No percentage per skill

iii. Time chart showing when different responses were received

9. Additional feedback
   a. Outside the Audiology clinic, what other groups of people might benefit from this Skill Builder application?
   b. Any other feedback?

Mid-Study Interview guide for Audiology clinic staff

1. Review informed consent

2. Showing reports and step through with the clinician the type of data gathered
   a. Were these reports helpful/unhelpful? How?
   b. What reports did you wish you could have seen?
   c. What recommendations would you make to the parent?
   d. How would you adjust frequency of reminders & feedback?
   e. How did interactions through the app affect your relationship with the parents?

Closing Interview guide for Audiology clinic staff

1. Review informed consent

2. Parent app usage
   a. What aspects of the Skill Builder application did you talk with the parents about during your appointments?
      i. Skills
ii. Situations

iii. Reports

iv. How to use it

b. What positive feedback did parents offer?

c. What negative feedback or areas for improvement did parents offer?

d. Would you change the questions used to remind and get feedback from the parents? If so, how would you phrase them?

3. **Staff app usage**

a. Did you review reports of how the parents used the app?

b. Were these reports helpful/unhelpful? How?

c. What reports did you wish you could have seen?

d. Did you encounter difficult with using the app?

e. How did interactions through the app affect your relationship with the parents?

4. Additional feedback

a. Outside the Audiology clinic, what other groups of people might benefit from this Skill Builder application?

b. Any other feedback?

**Opening Interview guide for Parents**

1. Explain the study and obtain informed consent

2. Explain about Skill Builder App and its purpose in recording feedback at the time of reminders
3. Gather information
   a. Age / Gender of child with audiological impairment
   b. What things are added to your daily routine since working with the clinic?
   c. What kinds of skills / tasks would you like to gather feedback on?
   d. What type of device do you have?
   e. How long have you had it?

4. Detail the application and how it fulfils the purpose
   a. Describe skills
   b. Describe situations
   c. Describe relationship between skills and situations
   d. Describe feedback recording mechanism
   e. Example feedback test

5. Describe the type of data we’re recording and will gather from the app on their phone
   a. Touch points during app usage
   b. Frequency of app usage / reminders
   c. Results of skills

6. Invite questions about app before install

7. Setup parent’s device
   a. Install app
   b. Setup skills & situations recommended by Audiological Staff
      i. What interval should we start the reminder(s) / feedback at
8. Determine follow up appointment

Closing Interview guide for Parents

1. Review informed consent

2. What types of skills did you work at developing?

3. How did you do at developing ______?
   
   a. Did you accomplish your goals?
   
   b. What influence did Skill Builder have on the development?

4. What did you like about the application?

5. What did you dislike about the application?

6. What would you change about the application?

7. What reminders were helpful?

8. Were there times you expected or wanted reminders for your skill and you didn’t get them? If so, when?
   
   a. Did you try to adjust reminder times?

9. Did you make mistakes in recording feedback?
   
   a. If so, how often?
   
   b. How did you feel about these mistakes?

10. What aspects of the Skill Builder application did you discuss with the audiology center staff?
   
   a. Skills
   
   b. Situations
c. Reports

d. How to use it

11. Did you review the reports at all? If so, what did you learn by reviewing the reports?

12. What reports did you wish you could have seen?

13. Did you feel the application helped you complete the tasks the audiology center staff assigned you and your child?

14. Would you use an application like this beyond this study?