6-5-2019

Agora Teaching App

Sam Burns

Daisuke Kurita

Casey Xuereb

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 5, 2019

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Sam Burns
Daisuke Kurita
Casey Xuereb

ENTITLED

Agora Teaching App

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREES OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING
BACHELOR OF SCIENCE IN WEB DESIGN AND ENGINEERING

Thesis Advisor

Department Chair

Agora Teaching App

by

Sam Burns
Daisuke Kurita
Casey Xuereb

Submitted in partial fulfillment of the requirements
for the degrees of
Bachelor of Science in Computer Science and Engineering
Bachelor of Science in Web Design and Engineering
School of Engineering
Santa Clara University

Santa Clara, California

Agora Teaching App

Sam Burns
Daisuke Kurita
Casey Xuereb

Department of Computer Engineering
Santa Clara University

ABSTRACT

Professional development of teachers normally ends after formal collegiate education. Agora desires to help educators continually train, to learn the skills of good teaching in classrooms by using various, non-traditional methodologies. By translating Agoras current platform to an app, Agora can widen their scope and serve Latin American countries outside of Peru, where they are currently based. We will simply take their current business and course platform and provide a mobile user interface for it. By expanding the reach of Agora, Latin American teachers will be well equipped to teach their classes with innovation and effectiveness.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Agora is an organization that offers educational training to teachers. The professional development of most educators ends after their collegiate or masters program education. After that, time and money for more teaching education is impractical or unavailable. In addition, a university teaches theoretical methods as opposed to the actual practice of teaching, and these methodologies can be stagnant and ineffective in the classroom. Agora's mission is to alleviate these problems in the education sector. However, their current platform cannot be effectively scaled outside of Lima, Peru because courses are taught in person. Additionally, their current in-person platform is not easily accessible because it lacks technology and mobility.

The users of our app include enterprises (schools, or school systems), topic experts, coaches and educators. Enterprises can use the app to delegate courses to various schools or educators. Experts provide the coursework materials and projects for the system. Coaches help train and teach the educators through the app. Finally, educators use the app to continue their education and collaborate with other teachers on their progress.

Current solutions include collegiate online courses and various other apps that provide courses for educators. Many of these apps, such as Canvas, a modern educational platform, are too expensive for the Latin American market. Other apps, such as Coursera, only provide educators with stagnant methodologies like lectures and quizzes. Skillshare focuses on the interaction between users, but not on the teaching and learning. The bottom line, though, is that these solutions are not designed for the Latin American market, so they are less likely to succeed and fall short of Agoras requirements.

1.2 Solution

Our solution is to create a web-based, mobile application that can deliver courses to users and allow for collaboration with others. In addition, the platform will allow experts and coaches in their fields to pay it forward by teaching and training educators. By creating an app for the model already in place, it automates the process, making it simple and

accessible for everyone. The app will promote various methodologies of teaching such as project-based learning. The app will provide course suggestions to users based on interests/taken courses. Our business model will be sustained by commission. We will attract users by offering free courses, similar to a free trial run.

Our solution is specifically customized for the Latin American market, a market in which this type of app has not been created yet. It is also customized for each individual user, who can filter their results based on interests or other circumstances. It connects experts, coaches, and educators to provide the best training for our users. Certain systems like Coursera only provide the educators with experts, who might not be able to teach in an effective way. Our system also allows for schools to delegate courses as they see fit, rather than what the educator might see fit. All courses are provided through Agora.

Chapter 2

Requirements

Based on the problems outlined above, we have derived functional, non-functional, and security requirements for our project.

2.1 Functional

2.1.1 Critical

- The system will provide courses for educators to gain training in their fields.
- The system will allow users to provide educational interests to filter course results.
- The system will allow administrators to provide courses to users.
- The system will allow users to submit projects for feedback.
- The system will recommend courses to users based on their previously taken courses.
- The system will allow for experts to schedule and facilitate in-person training in addition to the online courses.

2.1.2 Recommended

- The system will allow educators to collaborate with one another and with their coaches throughout the course progression.

2.1.3 Suggested

- The system will allow for free trials for the user, offering free courses at the onset of membership.
- The system will allow schools to delegate courses to teachers.

2.2 Non-Functional Requirements

- The system should have a simple and pleasant user interface, especially designed for the educators.
- The system should be accessible to people with disabilities.
- The system should be available at all times except when a portion is under further development.
- The system should be robust, able to handle errors and bugs.
- The system should be communicated in Spanish.

2.3 Security Requirements

- The system should not store plaintext passwords on the server.
- The system should not share data of one user with another user.
- The system should not allow users to view other users' coursework.
- The system should not keep payment information in plaintext on the server.
- The system should not allow more than 5 login attempts per minute.
- The system will allow the user to reset their password.

2.4 Design Constraints

- The system must be a mobile app.
- The system must be available to the Latin American market.
- The system must be available on the App store.
- The system must be completed by May 2019.

Chapter 3

Use Cases

Figure 3.1 displays the use case diagram for our system. Use cases are examples of how our system is used by the actors. The main actors are educators, who take courses offered by the system, and experts, who provide courses to the system.

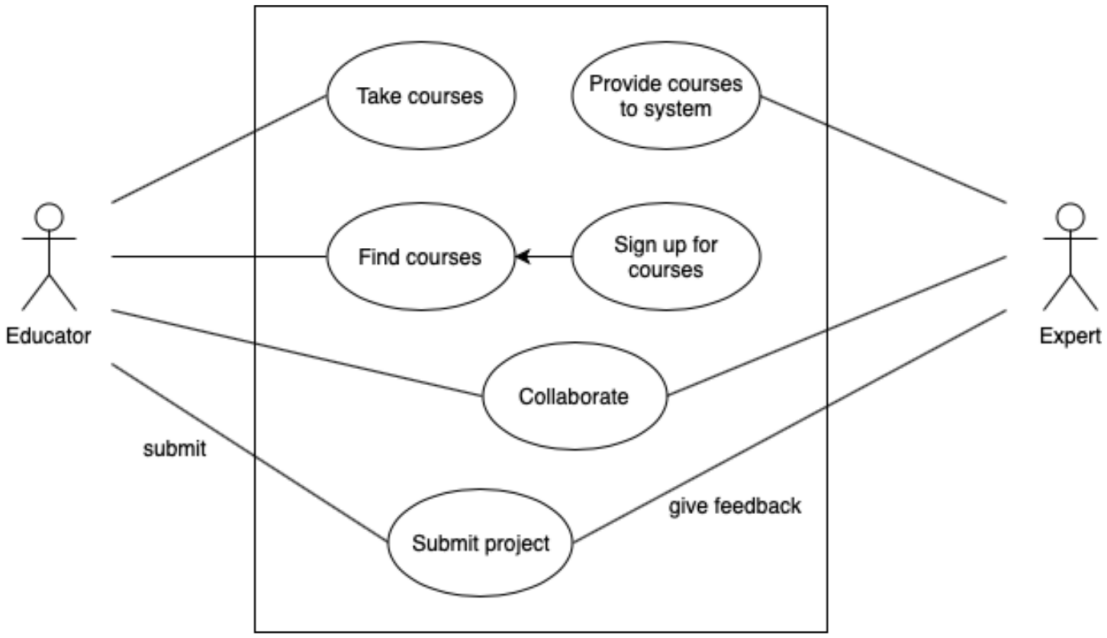

Figure 3.1: Use Case Diagram of the Agora App

3.1 Take Courses

Goal: Educators are able to take lessons from courses that they are registered in

Actors: Educator

Preconditions: Educator has a lesson that they can take

Postconditions: Educator has taken the lesson

Steps:

1. User selects course
2. User takes lesson from course

Exceptions: N/A

3.2 Provide Courses to System

Goal: Experts are able to provide courses to the system

Actors: Expert

Preconditions: Expert has course material available for submission

Postconditions: Expert has submitted the course material for approval

Steps:

1. User goes to course submission page
2. User prepares course material in format specified by system
3. User submits course

Exceptions: N/A

3.3 In-person Training

Goal: Experts are able to train educators in person

Actors: Educator, Expert

Preconditions: Educator is enrolled in class that Expert is responsible for, Training is scheduled

Postconditions: Training has been done

Steps:

1. Users go to course page
2. Users go to training section of course
3. Expert trains educator

Exceptions: N/A

3.4 Find Courses

Goal: Educators are able to find courses

Actors: Educator

Preconditions: User is looking for a course

Postconditions: User has found desired course

Steps:

1. User goes to list of courses
2. User searches list based on their interests or recommendation by the system

Exceptions: N/A

3.5 Sign up for Courses

Goal: Educators are able to sign up for courses

Actors: Educator

Preconditions: Educator has a course they want to sign up for

Postconditions: Educator is signed up for the course

Steps:

1. User goes to course page
2. User signs up for course

Exceptions: N/A

3.6 Collaborate

Goal: Educators and Experts are able to collaborate with one another

Actors: Educators, Experts

Preconditions: Multiple users enrolled in or responsible for the same course are willing to collaborate

Postconditions: Users have collaborated

Steps:

1. All users go to the relevant course page
2. Users go to collaboration section
3. Users collaborate

Exceptions: N/A

3.7 Submit Project

Goal: Educators are able to submit projects and experts are able to give feedback

Actors: Educators, Experts

Preconditions: Educator has a project they can submit

Postconditions: Educator has submitted project, Expert has given feedback

Steps:

1. Educator goes to course page
2. Educator submits project
3. Expert gives feedback to project

Exceptions: N/A

Chapter 4

Activity Diagrams

Our activity diagrams show the various tasks users will complete throughout their experience using the application.

4.1 Educator Activity Diagram

Figure 4.1 shows how educators will use the application. The two main uses for educators are signing up for courses and taking classes. If signing up for a course, they will first go to a list of courses. The user is then able to search the list based on their interests or recommendations by the system. The user can take classes if available. This may involve collaborating with other users and submitting projects. If training is scheduled, they can also receive training.

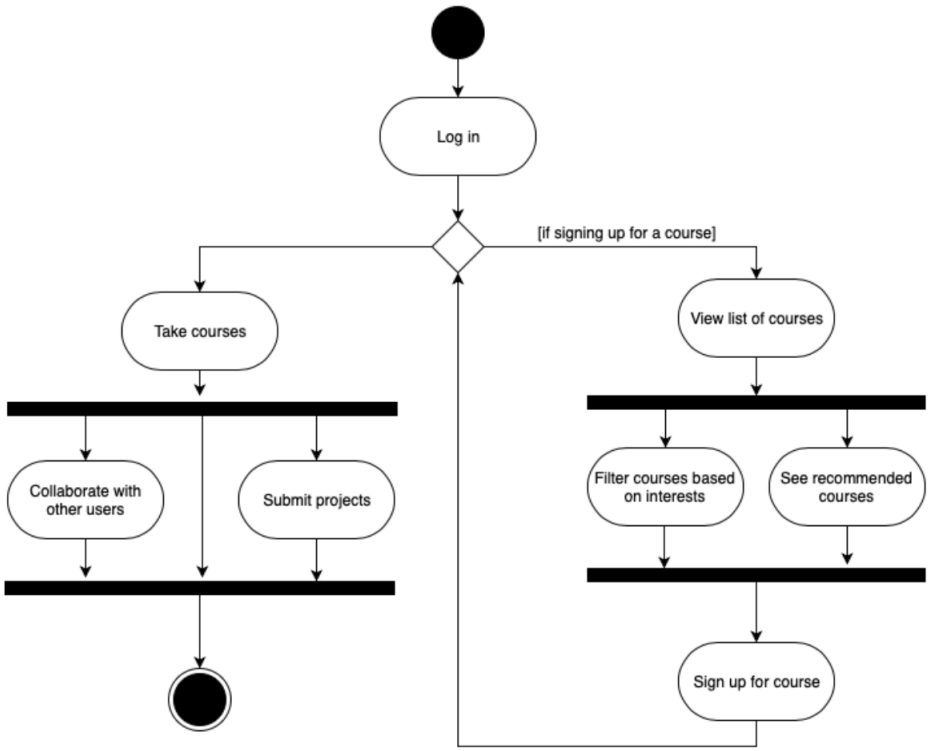

Figure 4.1: Activity Diagram for Educators

4.2 Expert Activity Diagram

Figure 4.2 shows how coaches and experts will use the application. After logging in, they can submit courses, give feedback to projects, collaborate with other users, and schedule training. Then, if they have a training already scheduled, they will give training to educators.

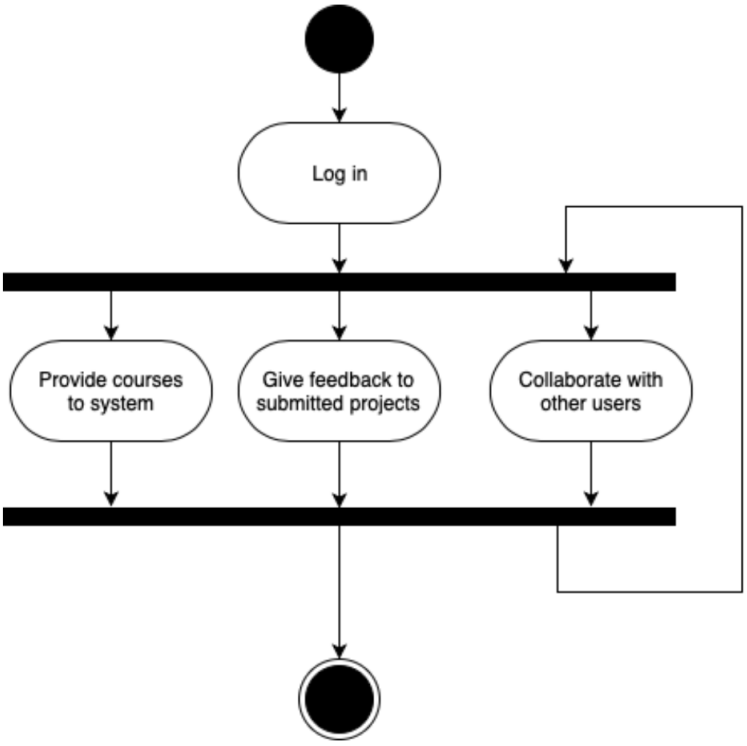

Figure 4.2: Activity Diagram for Coaches/Experts

Chapter 5

Technologies Used

- React Native
  - React Native is our frontend framework for the application. It can be used for Android development and allows for cross-platform mobile-apps.
- Thinkific API
  - The Thinkfic API is the backend of our mobile app. Almost all the information our app needs can be pulled from Thinkific's database using their API. We can create users on the thinkfic site, as well as manage everything to do with courses. The only thing not done through the Thinkfic API is our user's authentication.
- AWS Amplify
  - AWS Amplify is the library we use to build a database on the AWS servers. We use this database to store information about our users (username, password). Using the AWS Amplify Auth feature, we are able to use this database for authentication purposes.

Chapter 6

System Architecture

The system uses a data-centric architecture to handle information from the database and format it for use on both desktops and mobile devices. Through the Thinkific API, and using the Thinkific database, we are able to use and display all information about the school, courses, etc. We use an Axios GET/POST request through the API to get this information into React Native, and display it to the user. We also have a separate AWS Amplify database that stores our users information for authentication.

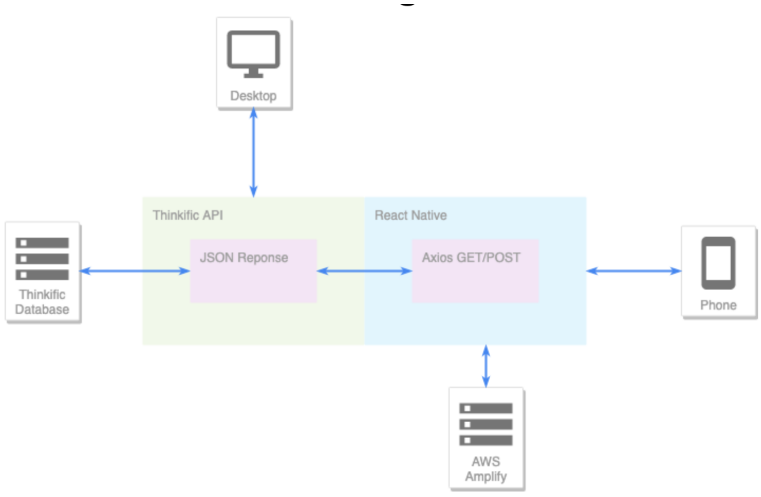

Figure 6.1: Architectural Diagram

Chapter 7

Design Rationale

7.1 User Interface

We chose our design for the user interface because we wanted to focus on the specific functionality of the application without overloading the user and the screen with information. Having a clean UI is important as it allows for all users to easily navigate the application and find what they are looking for. There is never an excess information on the screen at once, and a simple drop down menu allows for easy navigation throughout the site.

7.2 Technology

We chose to use a React Native based frontend. One of the main reasons we chose to use React Native is the ability to develop for multiple platforms. The teachers and coaches that will be using this application will be accessing it from a variety of devices, so the ability to have the application both on the web, on IOS, and on Android is a huge benefit. In addition, React Native allows for code reuse which allowed us to cut down our development time and focus on implementing all of the features needed. Also, React Native is focused mainly on building a mobile UI. We know that there are some issues with connectivity to the internet for the users of this application, so designing an efficient mobile app that will provide all the necessary functionality regardless of internet connectivity is crucial, and React Native enabled us to do that.

We chose to use the Thinkfic site and API as the backend to our application for a few reasons. First, it was a technology that was specifically requested by our client because it was affordable. Second, we chose to use this API because the documentation for it was excellent and it made actually using the API much easier. Finally, the Thinkific API allowed for almost zero backend development on our side. We simply interacted with the Thinkfic site to display all the necessary information to the user.

The last technology we chose to use was AWS Amplify. This goes in conjunction with React Native, as it is part of the React Native framework. AWS Amplify is what we used for user authentication. We chose to use this technology because it goes well with React Native and allows for easy setup for authentication. It is also safe and secure, which

is extremely important when handling sensitive user data.

Chapter 8

Test Plan

- Unit testing throughout development
  - We did modular development which allowed for testing of modules as we developed our app. Some modules could be working while others are not and the app will still function.
  - With modular development, we wrote tests for different parts of our application as we went along. We wrote tests for each of our different use cases and made sure the requirements that we outlined were fulfilled.
- Alpha/Beta testing through Agora's clients
  - Agora is currently putting our application through functional testing in order to make sure our implementation is functioning as intended.
  - Testers will be given no directions to ensure our application is easy to use for our intended user-base. It will also allow us to find any mistakes that were difficult to find from a developer's perspective.

Chapter 9

Description of System Implemented

The system we implemented is a mobile app that allows users to log in to Agora's system and take courses through the Thinkific site.

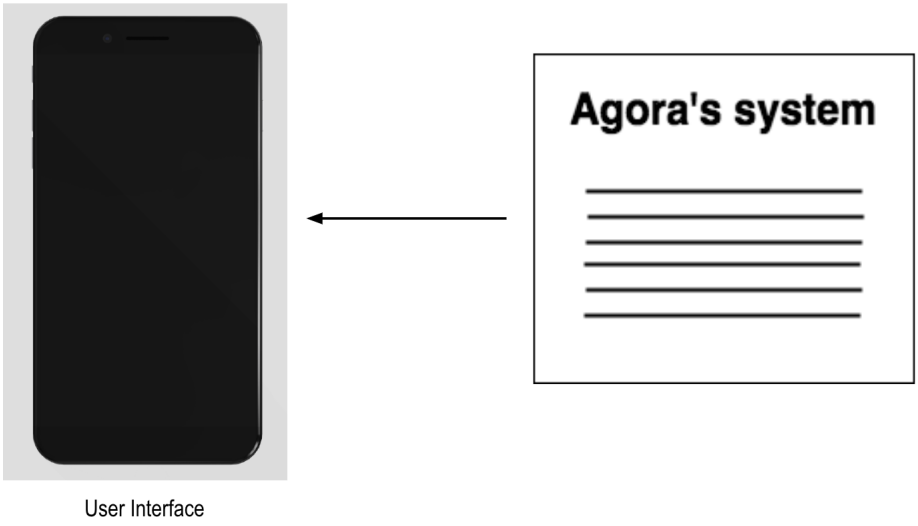

Figure 9.1: Our system logic

On starting the app, the user is met with a login screen. Users that are not registered can also create an account. Once the user logs in, the app displays the home page, with tabs at the bottom that lead to other pages. The pages are a list of courses registered by the user, a list of all courses, and a search function for the courses. On selecting a course, the app opens a web browser so that the user can take the course on Thinkific's site inside the app interface. Then, there will be instructions to login again, with the same credentials, into the Thinkific site. From there, the site will instruct the user to take the course through whatever curriculum an expert or coach has provided. Agora will upload their courses accordingly onto Thinkific in order for this to become a reality.

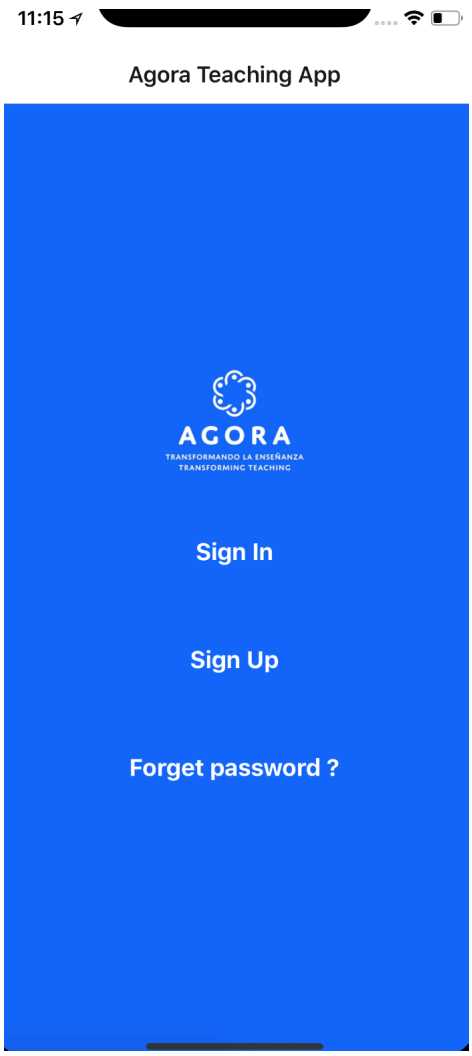

Figure 9.2: Welcome Screen

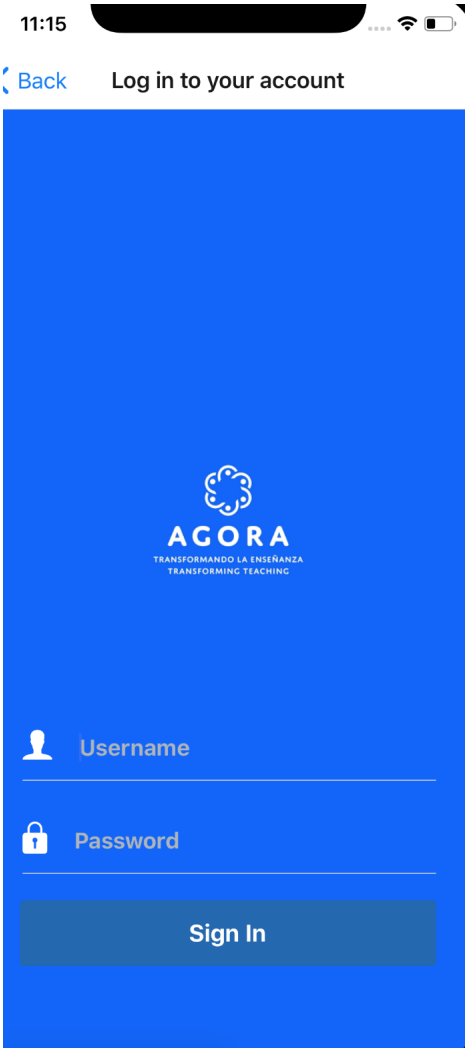

Figure 9.3: Login Screen

Figure 9.4: General Layout of App with Tab Bar

Figure 9.5: Screen to Search for Courses

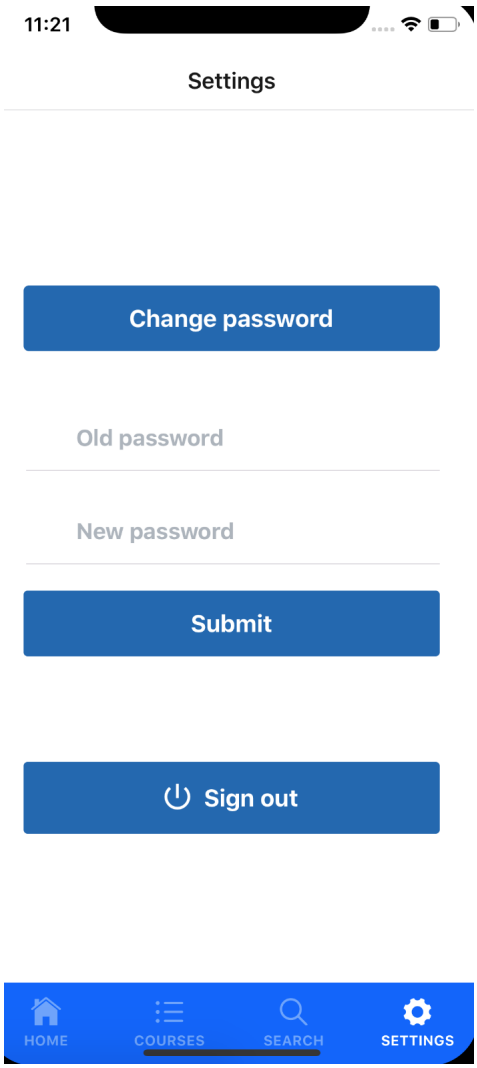

Figure 9.6: Settings Screen

Chapter 10

Social Implications

10.1 Ethical

An ethical question that was raised about our project was the sharing of user information and data privacy. We had to make sure that our security requirements of the app kept user's private information safe from hacking and other users. This included both login information and course information, such as grades or submitted work.

10.2 Social

Our app has a very strong social impact on the lives of educators, coaches, and students in Latin America. When education is easier to access and curriculum enhanced, everyone in society benefits. Education is the foundation of a successful and thriving community, and we see our app as bolstering the education system in Peru and Latin America.

10.3 Political

The biggest impact our app could potentially have on society is helping to improve the education systems in Latin America. Through our app, Agora's content will be able to reach more people and therefore there will be more Latin American students benefiting from their teachers improving. Since our app is very accessible, the potential for this to happen is high.

10.4 Economic

We had to consider that we were developing this app for people in Latin America to use. With this in mind, we knew that internet connectivity may be a problem as well as the type of device the app may be run on. Thus, we had to try to make the app viable within Peru, by making sure that it would be communicated in Spanish as well as making it available for both iOS and Android. Thankfully, the actual cost of development was minimal and will have no impact on the users as it will be a free app.

10.5 Health and Safety

Our app is very safe. There are no health or safety implications of using this app one your phone. Any implications would be of simply using a mobile phone, which we can not force any user to do.

10.6 Manufacturability

As the developers of this app, we are the key to it actually being manufacturable and making it easy to deploy. By choosing to use Expo and React Native, we the actual deployment of our app very easy. All we have to do is make sure there are little to no bugs, and then publish it on the iOS and Android app stores.

10.7 Sustainability

Our app will be viable and useful for a reasonable amount of time. The app will need to be updated from time to time as technology develops, especially the technological advancement of information systems on phones. But the foundation of our app can be used for many years to come.

10.8 Environmental Impact

The environmental impact of our app is indirect, because mobile phones have a large carbon footprint. But, none of the users we are trying to reach would buy a new mobile phone specifically to have this app on their phone. Therefore, there is a slim environmental impact of our project.

10.9 Usability

Our app is very usable due to its simple user interface. The functionality of the app is very intuitive and can be learned by users easily. This was done by creating screens with little text, eye-catching call-to-actions, and no clutter.

10.10 Lifelong learning

In working on this project, we had to learn how to use technologies, such as React Native, that we were unfamiliar with. We learned the importance of studying the technology even before starting to work with it as a group. We considered lifelong learning when choosing the Thinkific API as well. The Thinkific API is very well documented, which made it easy for us to learn how to use it compared to alternatives. In addition, our project was built to help educators with their lifelong learning. In working with Agora, we realized the importance of continuing our education.

10.11 Compassion

Agora came to us with a project they knew they could not complete on their own. We had compassion for their problem and desired to help them enhance their business platform. The app will touch many educators in Latin America, which will in turn touch many students' and children's lives as well.

Chapter 11

Difficulties Encountered

We encountered multiple difficulties throughout the development of our system. At the beginning of the project, we started our work by developing a python backend for our system. This ended up being a mistake because we took a few months developing it, and then ended up deleting all of the work to instead use the Thinkific API as the backend of our system. In addition to this, our first frontend to this application was a web application using React. We knew that the end goal was a mobile application, and figured the transition between our React web app and our React Native mobile app would be simple, and we were wrong about this. In addition, we found that actually attempting to use React Native, without actually knowing anything about the framework, was extremely difficult. So we ended up taking a few weeks that could have been spent on development, and spending that time learning more about ther React Native framework. Finally, we had a bit of difficulty surrounding the Thinkific API. It was a technology that was requested by our client, and even the ability to use the API required paying Thinkific $100 per month. Also, the ability to use their single sign on feature, which could have eliminated the need for our own user database and authentication, was locked behind a $500 per month paywall. Despite this, we were able to successfully use the Thinkfic API to the best of our ability.

Chapter 12

Suggested Changes

With our project in completion due to a designated timeline, we have some suggested changes to the system that would help improve its usability and experience. First off, we would purchase Thinkifics premium plan that included single-sign on authentication so that the user would not have to sign in twice on the app (once with Agora and then once with Thinkific). The login credentials are the same for the user and this would simplify their user experience.

Another suggested change would be to have the capability to take courses on the actual app interface rather than a browser opened inside the app interface. This was outside of our scope due to our strict deadline, but is definitely a possibility if Agora were to hire someone for this specific task. The hire would have to work closely with the Thinkific API and create a larger database to hold all the information.

Chapter 13

Lessons Learned

Some lessons we learned throughout our design process include the importance of learning new languages before starting to work as a group, of communicating with one another in a straightforward manner, and of planning before coding.

First off, we began our project designing for a web application because we all had experience with web development before and we thought the transition from web to mobile would be easy. This was not the case and once we had finished creating a rough idea of our web application, we tried to move it over to mobile and were not having any luck. Therefore, we learned that if you are building a mobile app, start with mobile development and build solely for mobile. A web application is a whole other project.

Second, once we started with mobile development, we chose the language of React Native, a framework that none of us had worked with previously. At first, we started to try and learn as we went along but then we realized that our time was not spent efficiently or effectively. Therefore, we decided to take some time to ourselves over spring break and learn React Native on our time with various online tutorials. Then, when we came back to spring quarter, we were ready to go with developing our app in a timely manner.

Lastly, in order for our meeting times to be efficient, we needed to communicate with one another better. This includes being upfront about what technologies or languages we know, and what task we are working at any moment. In the end, we shared a collective to-do list with one another where we could assign deadlines and group members to various tasks. This helped us with project management greatly.

Appendix A

Installation Guide

A.1 Getting Code from GitHub

Begin by cloning our repository and switching to the AgoraNative branch or get the code directly from source:

```
git clone https://github.com/samburns2/AgoraApp.git
cd AgoraApp/AgoraNative
```

A.2 Installing Dependencies

The next thing we have to do is install the required packages used to make our application function. From the AgoraNative directory, you need to run the following command:

```
npm i
```

Once this finishes, all the required dependencies are now installed

A.3 Install Expo CLI

Expo CLI is a tool for developing apps with Expo. We used this to host/run/and build our app. To install, run the following command:

```
npm install -g expo-cli
```

Follow all the steps of the installation. At this point, everything required is installed.

A.4 Running the App

Finally, now that we have all the dependencies installed, the app can be run. Navigate to the AgoraNative directory:

```
cd AgoraApp/AgoraNative
```

Once you are there, you can run the app using npm:

```
npm start
```

This will start up the Expo CLI both in your command line and your browser. From the site on the browser you can either click on one of the buttons to run the app in a simulator, or you can download the Expo App from the Apple or Android app store, and scan the QR code with your phone, to test the application directly on your phone.

Appendix B

User Manual

B.1 Create Account

1. Create an account by clicking on the Sign up now button. This account will automatically create an account for you both on Agora and Thinkific.

B.2 Log In

1. Log into your account, and you will now be able to enroll, search for, or take courses on the app.

B.3 Dashboard

- Upon logging into your account, your dashboard is the first screen you will see. This shows the courses that you are already enrolled in. If you are not enrolled in any courses, your screen will be blank.

B.4 Course List

- The next screen, Courses, shows all available courses that Agora has provided. If you would like to enroll in a course click on the course card, and then click on the enroll button. If you would like to take a course, you can use either your Dashboard screen of Courses Screen. The course will open in a web browser hosted by the app.

B.5 Search

- If you would like to search for courses based on keywords, use the Search screen. Simply type in a phrase such as math or reading and the courses that match will be displayed.

B.6 Settings

- Finally, if would like to change any login settings or log out of the system, use the Settings screen on the far right of the tab navigator. Follow given instructions and forms.