An iOS Shopping App for the Southern Community

Thomas Manu

***Abstract*—Online personal classified advertisement has been popularized by the need to buy and sell online. Websites such as Craigslist are very popular platform. Such solutions have failed for the use of Southern Adventist University's community due to the amount of scammers on these sites. This paper explains the project idea to create a mobile application that satisfies the need of having a safe and secure platform to post personal classified advertisement. The application will be tested and made sure that it is successful at its job.**

I. Introduction

Craigslist is a popular personal classified advertisement website with more than 60 million users in the US, an average of 50 billion page views, and about 80 million classified posts every month [1]. However, Craigslist is openly known to attract scammers in certain communities around the world on specific categories [2]. Southern Adventist University's solution to the problem of scammers was to create Campus Talk [3]. Unfortunately, not many students know about Campus Talk and it has been phased out for students. This has created a void for a safe and secure personal classified advertisement platform. We have designed a mobile application that will take care of this void. SouthernSales will be a place for only Southern students and faculty to post and view personal classified advertisements on the mobile platform. This mobile application will be more secure in terms of not letting just anyone post advertisements on the software, have a user friendly user interface that will be easy to use, and available to the Southern Adventist University's community.

SouthernSales will only be available on the iOS platform for iPhones starting from iPhone 6s to the latest device. The application will also only run on the specified iPhones that have iOS 10 and up. This application will also only allow users who have an @southern.edu email to sign up and use the application.

The rest of this paper will be discussing the background material in Section 2, the development and implementation of the project in Section 3, and the testing and evaluation process in Section 4.

II. Background

A. Current Personal Classified Advertisement Solutions

Campus Talk, the now defunct classified advertisement website meant for the Southern Adventist University community, was designed to let students and staff post personal advertisements for items they are selling, services they are offering, and requesting for items and services. However, barely anyone used this website as it was not openly known by everyone on campus. It also had a clunky user interface that was designed as a forum, instead of a personal classified advertisement website, meaning less people are inclined to use it.

There are similar personal classified advertisement websites and mobile applications that are currently in place on the Internet. Facebook Marketplace, the new big website for consumer-to-consumer selling, is being used by people around the world ever since it came out in 2016 [4]. Craigslist is another website that has been in place ever since 1995. It however has a clunky website that is not user friendly and is prone to scammers [5]. When it comes to the mobile platform, applications such as LetGo and OfferUp are similar in such that they offer a place to post personal classified advertisements with a better user interface than Campus Talk and Craigslist.

However, all the aforementioned websites and mobile applications (except Campus Talk) have issues that are glaring for Southern Adventist University's community. Posting appropriate advertisements is one as recently, Yik Yak was shut down from Southern's WiFi due to issues even though that was what students used to post personal advertisements [6] [7]. Another issue is that every software available is for anyone to use, meaning there will be scammers trying to gain money by selling fake items. One last issue is that because it is for anyone to use, there is no single current application meant for Southern's community as a whole for both students and faculty to post personal classified advertisement to sell and buy items and request and post services.

B. Cloud Storage

Cloud storage and cloud computing have been the current way of outsourcing complex processes and mobile storage into the cloud. With the cloud comes many concerns for both developers and users. How secure is the cloud and how fast is it? Lately there have been quite a few providers of cloud storage for developers to support their applications. A few of them are:

- Google Cloud [8]
- Microsoft Azure Cloud [9]
- Amazon Web Services (AWS) [10]

Each of these have their own database model and security used to secure their cloud storage. In each of the three mentioned providers, Google Cloud uses a NoSQL database as their storage model whilst the others use variant types of relational databases. NoSQL data models, according to Li and Gao, "are more popular in the mobile cloud databases" over relational

T. Manu is with the School of Computing, Southern Adventist University, Collegedale, TN, 37315 USA (e-mail: thomasm@southern.edu.)

data models [11]. With all the different types of databases and providers, we are faced with privacy and security issues as lately there are news articles on users data being leaked from the cloud storage. Li and Gao have proposed using security on top of what is being sent and queried by the application to the databases. They also mentioned having a security model for the user's privacy as to make sure they do not get leaked to the public.

Although there have been proposals for a different mobile cloud storage as given by Pedersen in his research [12], where instead of relying on an overlayed network, we store everything on each mobile phone and use a peer-to-peer network, it is a topic for the future. Looking at the different listed cloud providers above, we have decided to go with Google Cloud because of their database model choice and security for privacy. They also have a large amount of data centers that their services are in. In terms of speed in processing jobs, they have a very low overhead and a fast processing speed especially with the increase of parallel jobs according to a benchmark done in 2012 by Prodan [13]. With cost not being expensive and having a none relational database with security on top of the implemented security, Google Cloud storage is the way to go for this project.

III. SouthernSales iOS Development and Implementation

SouthernSales is a personal classified advertisement mobile application that will be available for download sometime in 2019. It is a standalone application that will only run on iOS devices. This application will allow Southern Adventist University's students and staff to post and view advertisements pertaining to their category of choice. As cloud-based application, it will need access to the internet to function properly. The application will be working in conjunction with a database to store all information of every listing made on the application.

A. Application Summary

The SouthernSales application will be running on only the iOS platform. It will only work on iOS version 10.0 and up and will only run on iPhones that start from iPhone 6s to the latest. This application might be available on the Apple App Store on eligible devices as specified above. Below, we describe the app flow that the user will encounter just as viewed in Figure 1.

Upon opening the application for the first time, users will be welcomed with a welcome screen that will highlight the core features that the application offers with an option to skip all the pages. On the last page, the user will press a "Get Started" button that takes them to a login page that uses the Google OAuth 2.0 API. The user will be taken to the southern.edu log in page to use their official email and password and will be asked by Google if they want to send over basic information to this application. After the user signs into the application, they will be greeted with the home page, which is the page that shows a selection of the classified advertisements that has already been posted. When opening the application for the second and many other times later, if the user isn't already

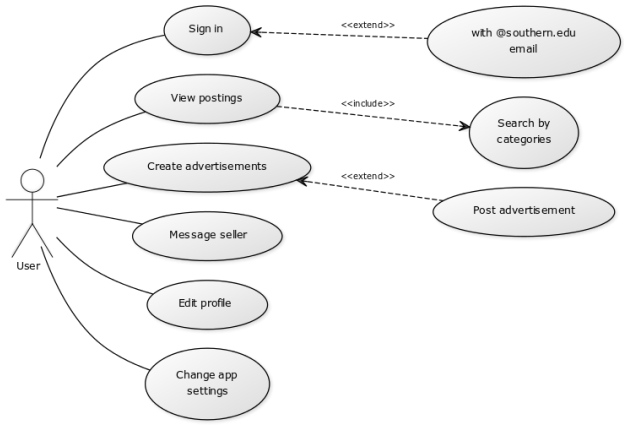


Fig. 1. Use case diagram that follows the application flow.

signed in, they will have to sign in to use the application as it is required to avoid users that are not part of the Southern Adventist University. After signing in, they will also be taken to the home page.

From the main home page, the user will be able to go to different tabs which will consist of home, saved, post an advertisement, chats, and settings. The user will be able to pull down the screen to refresh the home page. The home page will allow a user to filter between different categories such as personal advertisements and services. When the user presses on a listing he/she wants to take a look, they will be presented with a screen with the images (if any) at the top and description of the advertisement below. The user is also given the chance to save the listing with the press of a heart button. At the bottom of the advertisement, the user can press a button that will take them to the chat tab where they can message the seller to inquire about the product being rendered. The user will be able to search for specific advertisements across the entire database with the search bar at the top of their screen. The search results will have the same view as how the main page looks like. After picking a listing, the user will be able to do everything that was listed above.

On the saved tab, the user will be able to view saved listings. When a listing is viewed, everything a user can do with a viewed listing as said above, he/she can do. If the saved listing was listed as sold or deleted, the user will not be able to view the listing but can delete it from their saved tab.

Jumping to the chats tab, this is where the user will be able to communicate with the seller and vice-versa without having to exchange numbers to text but can do so if they so wish to. The only chats that will show are ones initiated by the buyer to the user from the first chat to the latest. The chat will support the use of emojis. Each chat will also show whether the item they messaged the seller for was listed as sold or deleted.

Moving on to the profile tab, the user will be able to view their personal profile that will have their profile picture, if they so wish to add, their name, email, and other personal information that was added. At the top of the screen, the user will be to edit their profile and save it. From this page, the user will also be able to change their password. The user will

also be able to view the posts they have made in the past and current and edit them or delete them.

Also at the top of the profile page, the user will be able to press a cogwheel button to go into the settings page where they can view the license and application version. All other settings such as notification settings, use over cellular, and permissions will be stored in the phone settings which can be found in the Settings application from the device's home screen.

Going back to the post an advertisement tab, the user will be able to take one or more picture of what they are trying to sell. If the user declines to add any picture, they will be taken on to the next step which will be the same step after taking the pictures. The next step will be a page where the user can include information such as the title of the listing, the price, and the description. After inputting all the information needed, the user will be able to preview what the listing will look like to buyers. On this page, the user will be able to post the final advertisement for the entire user base to see. Throughout this entire process, the user can cancel their listing without the option to save the draft. The posted listing can be edited from either the home tab or from the profile tab.

All details of the SouthernSales application's features and requirements are listed in the SouthernSales Requirement Specification located in Appendix A.

B. Development Approach

The SouthernSales application will be implemented with native code using Swift 4 for iOS devices on Xcode IDE. The application will use the modern Model-View-Controller (MVC) design pattern to build the application. There will also be a heavy use of delegation between different pages to communicate information throughout the application. When it comes to user interface, we will follow the Human Interface Guidelines [30] set by Apple. Xcode will also help guide us with coding styles and conventions with the help of SwiftLint.

The application will be talking to a Google Firebase database that will be created in conjunction with the application. This database will store everything that the user sees in the SouthernSales application, from listings to profile settings. Settings that affect the application only will be stored in the user's device using the User Defaults [31].

We will be using over fourteen libraries/frameworks to help with the development process in achieving a better application than to write everything from scratch. Most of the libraries are licensed under the MIT License [32] and one or more are under the Apache 2.0 License [33]. Table 1 lists all the open-source libraries we will be using.

Since this application is an iOS application, a computer running the latest macOS with Xcode 10 installed will be required. This computer will need 8GB of RAM and at least 128GB of Solid State Disk space. A separate monitor will be helpful in a multi-window development process. Multiple iPhones from iPhone 6s and up running iOS 10 and up will be needed to run the tests during the beta and final stages of testing and evaluation.

TABLE I
OPEN-SOURCE LIBRARIES TO BE USED IN SOUTHERNSALES

Names	Purpose
Alamofire [14]	Allows asynchronous downloads of data in JSON
CocoaLumberjack [15]	Helps with logging into a file for debugging use
CocoaPods [16]	A dependency manager to manage all the libraries in this project
Eureka [17]	Helps to build forms with ease of use
Firebase [18]	A SDK to help connect to the Google Firebase realtime database
Google Sign-In [19]	A SDK that allows authentication with the Google server and @southern.edu emails
ImageSlideshow [20]	Allows the use of an image slideshow when there is multiple images for a product
IQKeyboardManager [21]	A keyboard manager that helps with movement between text boxes
MessageKit [22]	A message UI library that helps with the design for a messaging platform
NVActivityIndicatorView [23]	Gives activity indicators if loading data will hang the UI
Onboard [24]	A library to help with the welcome page
SDWebImage [25]	An asynchronous image downloader and cache manager to help with images used for advertisements
SnapKit [26]	An autolayout helper library that will make dealing with NSLayoutContraints easier
SwiftLint [27]	A library to help us write code that follows Swift styles and conventions.
SwiftMonkey [28]	A random input testing library to help with breaking the application
SwiftyJSON [29]	A JSON data helper that helps read JSON data into objects

TABLE II
TASK DELINEATION FOR SOUTHERNSALES APPLICATION

Tasks	Work Hours Estimate
Database and Server Implementation	25
Classified Listings	25
Chat Messaging	18
Profile and Settings	15
Beta and User Testing	25
Final Changes and Fixes	20
Project Completion	128

C. Task Delineation

The SouthernSales application project will have a minimum of a hundred work hours to build the application until it is ready to publish. This project will be split into multiple different tasks to work on different aspects of the application to bring it into fruition. Table 2 has a summary of the tasks listed below.

1) Database and Server Implementation:
The first task for the SouthernSales project is to start building the Google Firebase realtime database and configure the API

that would go together with it. All the data such as the classified advertisements, users, and authorization will be stored in this database except user settings, which will be stored on the iPhones. All this work will be done behind the scenes as it is a background work and needs to be implemented first before the actual application.

2) Personalized Listings:
The second task that we will be working on will be with the personalized listing as a whole. During this time, we will be working on implementing the home page with all the posted listings, the search function, and also work on the part that deals with capturing pictures and inputting information to create a new post. This will be the main focus of the entire project as this is the reason why we are building this application.

3) Chat Messaging:
Chat messaging will be the next task that will be worked on. Without this section, the user will have to resort to talking to the seller or buyer using their phone number instead of through the convenience of the application. Users will only be able to send messages to sellers that have a post available. Notifications from new messages will also be implemented during this stage.

4) Profile and Settings:
Every user will have a profile with the required name and email. Profile pictures and more information will be available for users to add. During this task, we will also work on adding the settings page for the users to be able to change their settings to their preferences. A settings page is required in an application as not everyone has the same taste and style.

5) Beta and User Testing:
During this task, the application will be going through testing and evaluation with every beta version made. Every single part of the application will be thoroughly tested and any bugs found will be fixed. Such tests will include chaos testing, acceptance testing, unit testing, and system testing. As Steve Quirolgico et. al. tells us in "Vetting Mobile Apps" [34], many applications are published without ever being checked for vulnerabilities and so this task will be dedicated to making sure there are no bugs and that application reporting will be implemented.

6) Final Changes and Fixes:
The final task in this project will be for the final changes and bug fixes made to the application. During this task, we will also be making sure we have documented every function that is not one of the external libraries listed above in Table 1. Once this task is done, SouthernSales application will be ready to be eligible to be published on the Apple App Store.

D. Final Deliverable

At the end of the allotted time chosen to be spent working on the SouthernSales project, we will have the following to be presented to the School of Computing:

- Source code
- Access to the project source control
- Final project report

IV. Testing and Evaluation

A. Acceptance Testing

SouthernSales will go through a list of acceptance testing. One of the tests will be a survey of a group of 5 to 20 people with one or two coming from each class standing and one or more staff members. The survey will have a rating of 1 to 5, where 1 is bad or hard, 3 is satisfactory, and 5 is good or easy. It will consist of questions such as:

- How was the overall experience with the app?
- How user friendly is the app?
- How easy is it to find different categories of classified advertisement?
- Is the application fast enough for everyday use?

This survey hopes to earn a result of at least a 3.5 out of 5 points.

B. Unit Testing

The plan for SouthernSales iOS app is to use Apple Xcode XCTest framework that is built-in to run unit tests by checking for memory leaks, battery usage, data models, performance issues, and other input tests. Using Xcode Instruments, we check to make sure that there are no memory leaks in our code, however, Apple's code could have memory leaks but that will be out of our control. Checking for battery usage is key in making sure our app does not eat up a lot of battery whenever a user is using the SouthernSales app. Performance tests with our code will be used to check to measure how long blocks of code take to run, in seconds.

The XCTest framework will run tests on both Xcode Simulator for all iPhones from iPhone 7 to iPhone XS, both the smaller and larger versions. All iPhones will be running iOS 10, 11, and 12 for every test. We will also run tests on real devices, mainly the iPhone 6s, 8, X, and XS Max, all running the latest stable iOS 12. All these tests will be automated.

We will also be testing every single function and classes in the code source to make sure everything runs right. This will be done manually throughout the entire process.

C. Module, System and Integration Testing

UI AutoMonkey framework will be used to test both module and system on the SouthernSales app. This testing tool is meant to stress test the application as a whole. It will be pressing random buttons, moving to random pages, entering random characters, trying to break the application. Such a tool is very useful as many specific combination of presses can cause an application to crash due to some anomaly.

We will also be loading the application onto an iPhone XS Max and given to a group of people, up to 5, and tell them to run free with it and try to crash the application. The interaction

will be logged and once it crashes, will be looked through to fix the crash.

Due to the nature of SouthernSales mobile application, almost every press on the screen will be using an API. Due to this, integration tests will be continuous throughout every acceptance, module, and system testing.

V. Conclusion

SouthernSales is a standalone application that will be made for the Southern Adventist University's community. This application will only run on certain iPhones and certain iOS versions. It will be developed with Swift 4 and will need internet access for the application to work properly and efficiently. Features like posting advertisements onto the listings and messaging sellers will be included. People with an @southern.edu email will be the only ones eligible to sign in and use this application. The SouthernSales application will go through thorough testing to make sure everything works right before being released to the public. After version 1.0, we hope to see further enhancements and changes made for the better of this application.

Acknowledgment

The authors would like to thank Dr. Tyson Hall and Professor Robert Ordoñez of Southern Adventist University for overseeing and editing this paper throughout its life ecycle.

References

[1] Craigslist, "factsheet," 2017. https://www.craigslist.org/about/factsheet, Last accessed on 10/22/2017.
[2] V. Garg and S. Niliadeh, "Craigslist scams and community composition: Investigating online fraud victimization," in *2013 IEEE Security and Privacy Workshops*, pp. 123–126, May 2013.
[3] S. A. University, "Campus talk information," 2018. http://www.southern.edu/connect/campustalkinfo.html, Last accessed on 11/1/2018.
[4] F. Inc, "Introducing marketplace: Buy and sell with your local community," 2016. https://newsroom.fb.com/news/2016/10/introducing-marketplace-buy-and-sell-with-your-local-community/, Last accessed on 10/16/2018.
[5] F. Guides, "Craiglist scams and fraud: The ultimate guide," 2016. https://www.fraudguides.com/internet/craigslist/, Last accessed on 10/16/2018.
[6] S. Magazine, "Racist social media firestorm erupts during Southern Adventist University vespers," 2016. https://spectrummagazine.org/article/2016/02/28/racist-social-media-firestorm-erupts-during-southern-adventist-university-vespers, Last accessed on 10/16/2018.
[7] WTVC, "SAU unplugs Yik Yak after racially-charged posts," 2016. https://newschannel9.com/news/local/sau-unplugs-yik-yak-after-racially-charged-posts, Last accessed on 10/18/2018.
[8] Google, "Google cloud," 2018. https://cloud.google.com/.
[9] Microsoft, "Microsoft azure cloud," 2018. https://azure.microsoft.com/en-us/.
[10] Amazon, "Amazon web services (aws)," 2018. https://aws.amazon.com/.
[11] S. Li and J. Gao, "Moving from mobile databases to mobile cloud data services," in *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 235–236, March 2015.
[12] M. V. Pedersen and F. H. P. Fitzek, "Mobile clouds: The new content distribution platform," *Proceedings of the IEEE*, vol. 100, pp. 1400–1403, May 2012.
[13] R. Prodan, M. Sperk, and S. Ostermann, "Evaluating high-performance computing on google app engine," *IEEE Software*, vol. 29, pp. 52–58, March 2012.
[14] A. S. Foundation, "Alamofire," 2018. https://github.com/Alamofire/Alamofire.
[15] Deusty, "Cocoalumberjack," 2018. https://github.com/CocoaLumberjack/CocoaLumberjack.
[16] E. Durn, "Cocoapods," 2017. https://github.com/CocoaPods/CocoaPods.
[17] Xmartlabs, "Eureka," 2015. https://github.com/xmartlabs/Eureka.
[18] Google, "Firebase," 2018. https://github.com/firebase/firebase-ios-sdk.
[19] Google, "Google sign-in for ios," 2018. https://developers.google.com/identity/sign-in/ios/.
[20] P. Zvonek, "Imageslideshow," 2015. https://github.com/zvonicek/ImageSlideshow.
[21] I. Qurashi, "Iqkeyboardmanager," 2017. https://github.com/hackiftekhar/IQKeyboardManager.
[22] MessageKit, "Messagekit," 2018. https://github.com/MessageKit/MessageKit.
[23] V. Nguyen, "Nvactivityindicatorview," 2016. https://github.com/ninjaprox/NVActivityIndicatorView.
[24] M. Amaral, "Onboard," 2014. https://github.com/mamaral/Onboard.
[25] O. Poitrey, "Sdwebimage," 2017. https://github.com/rs/SDWebImage.
[26] S. Team, "Snapkit," 2018. https://github.com/SnapKit/SnapKit.
[27] R. Inc, "Swiftlint," 2015. https://github.com/realm/SwiftLint.
[28] Zalando, "Swiftmonkey," 2016. https://github.com/zalando/SwiftMonkey.
[29] R. Fu, "Swiftyjson," 2017. https://github.com/SwiftyJSON/SwiftyJSON.
[30] Apple, "Human interface guidelines," 2018. https://developer.apple.com/design/human-interface-guidelines/ios, Last accessed on 10/18/2018.
[31] Apple, "Userdefaults," 2018. https://developer.apple.com/documentation/foundation/userdefaults, Last accessed on 10/18/2018.
[32] "The MIT license," 2018. https://opensource.org/licenses/MIT.
[33] "Apache license, version 2.0," 2018. https://opensource.org/licenses/Apache-2.0.
[34] S. Quirolgico, J. Voas, and R. Kuhn, "Vetting mobile apps," *IT Professional*, vol. 13, no. 4, pp. 9–11, 2011.

APPENDIX

APPENDIX A
SOUTHERNSALES REQUIREMENT SPECIFICATION

A. Task Delineation

Most of the requirements listed in ... falls under one of the tasks listed below. Most corresponding requirements will have square brackets with the task number in it.

1) Database and Server Implementation
2) Classified Listings
3) Chat Messaging
4) Profile and Settings
5) Beta and User Testing
6) Final Changes and Fixes

B. Hardware and Software Requirements

- This application will only support iPhones starting from iPhone 6s and up. This will support all sizes from the specified iPhone. The devices listed below fall under this specification:
  - iPhone 6s
  - iPhone 6s Plus
  - iPhone 7
  - iPhone 7 Plus
  - iPhone 8
  - iPhone 8 Plus
  - iPhone X
  - iPhone XR
  - iPhone XS
  - iPhone XS Max
- This application will only run on iOS 10.0 and up.
- The device will need internet access to connect to the database to get post listings and to be able to message sellers.

C. Application Requirements

1) General:

- When first-time users open the application, it should present the user with an onboarding feature/tutorial page.
- Upon completion of tutorial, the user will be taken to a log in page.
- When opening the application after the first time, the app will present a log in page if the user is not logged in already.
- [1] The application will use Google OAuth 2.0 to let the user login to their @southern.edu email.
- [1] The application will only support the English language.
- The application will only support portrait orientation.
- The application will use a dynamic font throughout the application that follows the font size the user set in their settings app.

2) Classified Advertisement:

- [1] All post listings that will appear on the home page will be pulled from the database.
- [1] The home page will refresh with new listings when the user opens the application from fresh start.
- [1] The user will be able to pull to refresh on the homepage
- [2] After signing in, the user will be presented with a page with all the current listings.
- [2] The user will be able to pick a listing of their choice and the application will show them a page with any images that the seller has added, and below it will have the description of the product/service.
- [2] The user will be able to save the listing by pressing the heart button.
- [3] The user will be able message the seller about the product/service by pressing the ”Message Seller” button.
- [2] The user will be able to search for listings by title or by keywords from the search bar in the home page.
- [2] The user will be able to upload images to create an advertisement.
- [2] The user will have the option to skip uploading images to an advertisement.
- [2] After uploading one or more images, the user will be required to input information about the advertisement they are creating.

- [2] The information required by the application of the user for a new advertisement will be title, price or price range, and the description of the product or service being offered.
- [2] Throughout the creation process of an advertisement, the application will have the option to cancel the new advertisement without the option to save the draft.
- [2] The application will present a preview of the advertisement before the user will be allowed to post the listing.
- [2] The user will be able to edit or delete an already created advertisement that is their own.

3) Chat Messaging:

- [3] The user will be able to message sellers directly from the listing page.
- [3] The user will be notified of new messages from the application.
- [3] The user will be able to block a user that has already contacted them.
- [3] The application will let a user know that an item has been sold or removed when viewing the message thread with the user.

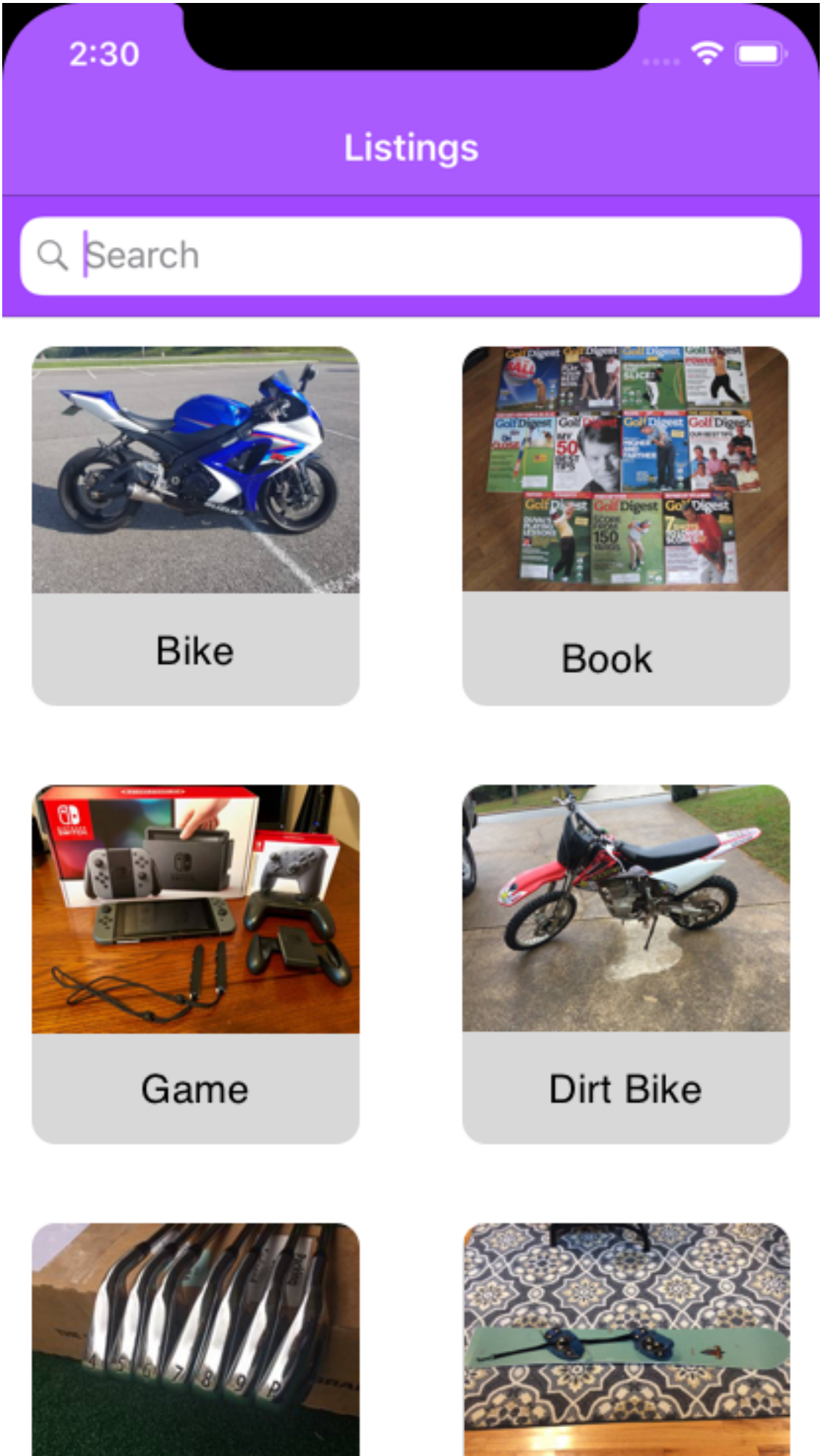
4) Profile:

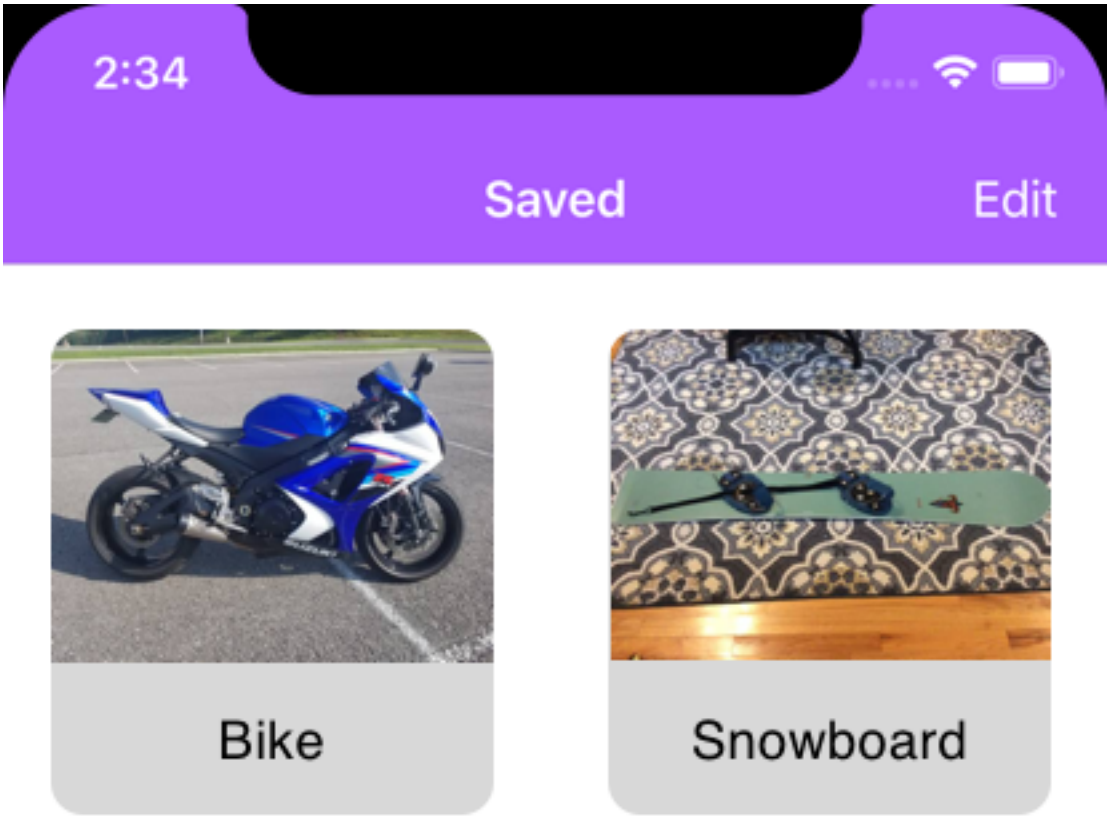
- [4] The application will show the user a summary of their profile on the profile tab.
- [4] The user will see their profile picture, if any, name, and email information.
- [4] The user will be able to edit their profile by adding a new profile picture or replacing the existing one.

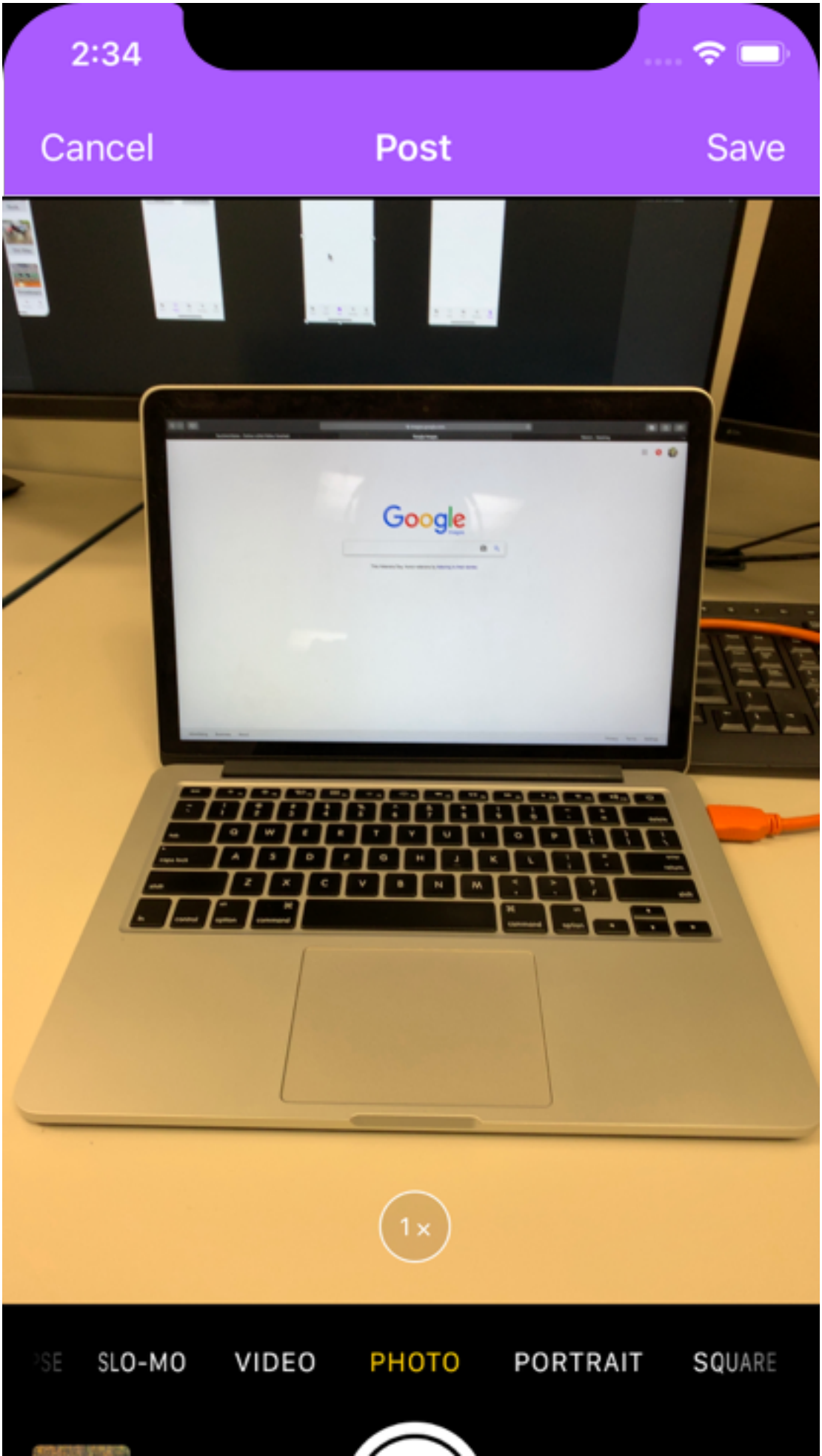
5) Settings:

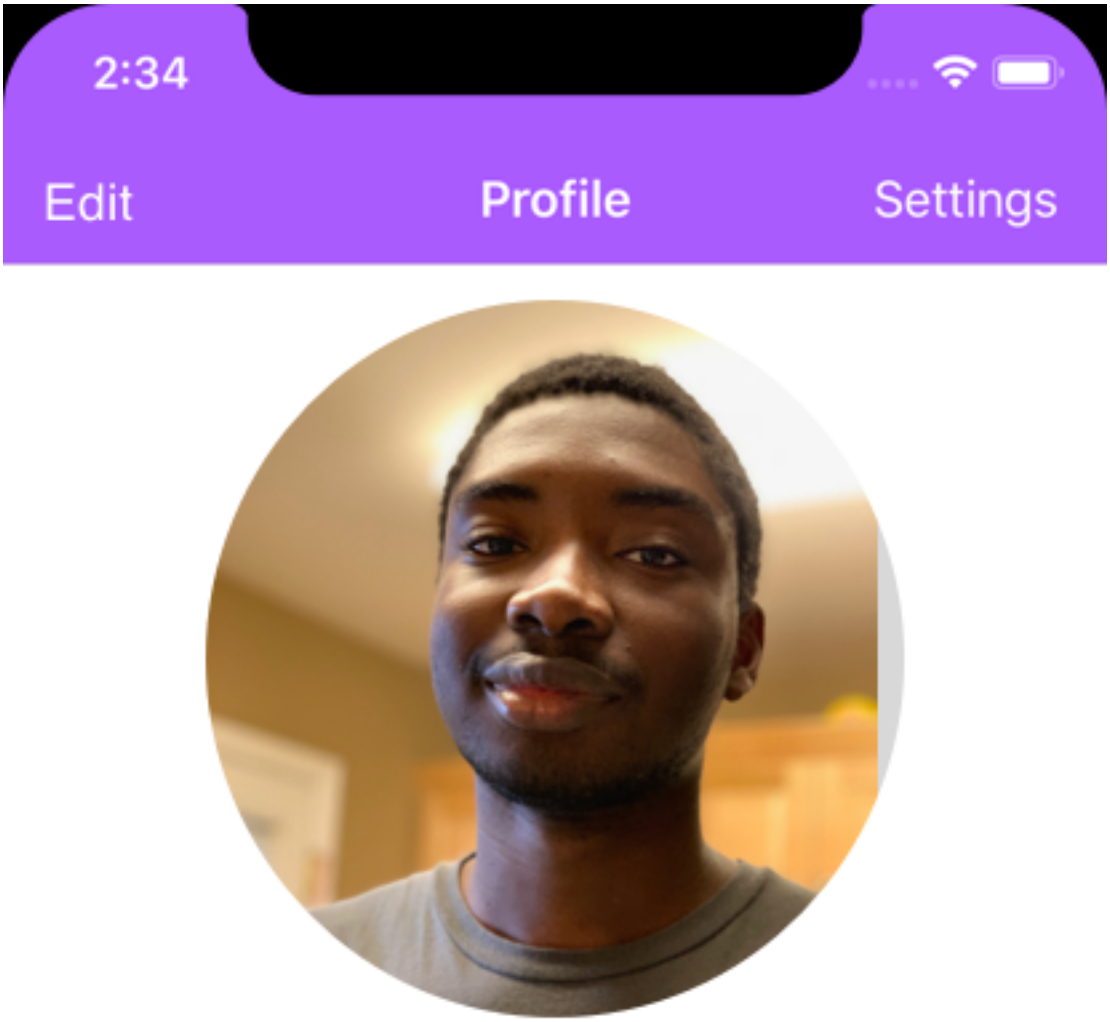
- [4] The user will be able to change different settings when the cogwheel/settings button is pressed from the profile page.
- [4] The application will offer the option to enable Face ID/Touch ID if the user wants to save their email and password combination.
- [4] Notification settings for the chat messages will be available to edit.
- [4] Permission settings will be linked to the Settings app from the settings page.
- [4] The licenses and credits will be found in the settings page.
- [3] The user will be able to view a list of blocked users;
- [3] The user will be able to unblock a user that has already been blocked from the block list.
- [4] The user will be able to log out from the application, and the application will show them the log in page.

APPENDIX B
SOUTHERNSALES PROTOTYPES


2:30
Listings
Search
Bike
Book
Game
Dirt Bike


2:34
Saved
Edit
Bike
Snowboard


2:34
Cancel
Post
Save
Google
1×
SLO-MO
VIDEO
PHOTO
PORTRAIT
SQUARE



Thomas Manu

Email: thomasm@southern.edu
Bio: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vitae ipsum quis nisl mollis finibus a a purus. Aenean id mauris nec magna sodales euismod. Phasellus interdum tincidunt velit, vitae posuere ligula convallis quis. Integer convallis id ipsum dignissim mattis. Etiam sem nulla, dignissim id vehicula non, pulvinar at augue. Etiam bibendum tincidunt tellus ut suscipit. Curabitur volutpat luctus volutpat. Orci varius natoque penatibus et magnis dis