

Senior Project - Final Report

Modern Yard Sale Software Application

Lauren Epling & Matthew Piasecki

California Polytechnic State University, San Luis Obispo

CSC Department

March 10, 2019

Advisor: Professor Franz Kurfess

Table of Contents

Senior Project - Final Report

Modern Yard Sale Software Application

Introduction	2
Motivation	2
Background	2
System Requirements	2
Use Case Diagram	2
Initial Class Diagram	3
Project Goals	4
Related Systems	4
Project Overview	5
Implementation	5
Code	5
System Screenshots	7
iOS App Made in XCode	7
iOS/Android/Web App made with Native React	9
Conclusion	10
Findings	10
Future Work	11
Resources	12

Introduction

Motivation

After 4 years of driving around the suburbs of San Luis Obispo, California, we noticed there was a tremendous amount of yard and/or garage sale advertisement posters on display by locals. Being the programmers we are, we instinctively realized: we could make an app for that! There is an astounding need for a safe space where users can comfortably post their yard sale address and items for all locals to easily see (without needing to drive down a specific street to find out). Currently, there does not exist an application for users that accomplishes what we set out to accomplish. As a team, we truly believe YardSail could be a popular application that helps users sail through the experience of hosting or visiting a yard sale.

Background

As explained in our motivation, for our senior project, we wanted to create a software application for users to comfortably post and view local yard/garage sales. During Fall Quarter 2018, we completed CSC 491, which resulted in our preliminary design, storyboards, use-cases, diagrams, and general prototyping. During that time, we were able to create a shell program from which we could develop our ideal application.

System Requirements

We decided to include 3 basic system requirements:

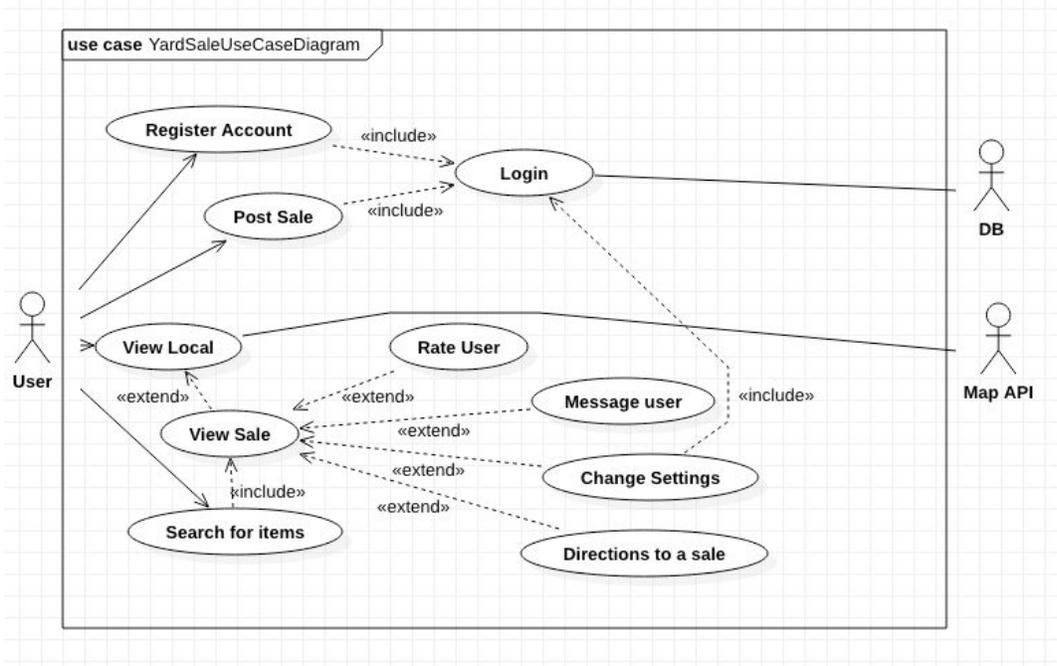
R1: The system shall provide the user the ability to create an account.

R2: The system shall provide a place to write a description for a new yard sale.

R3: The system shall provide a settings page.

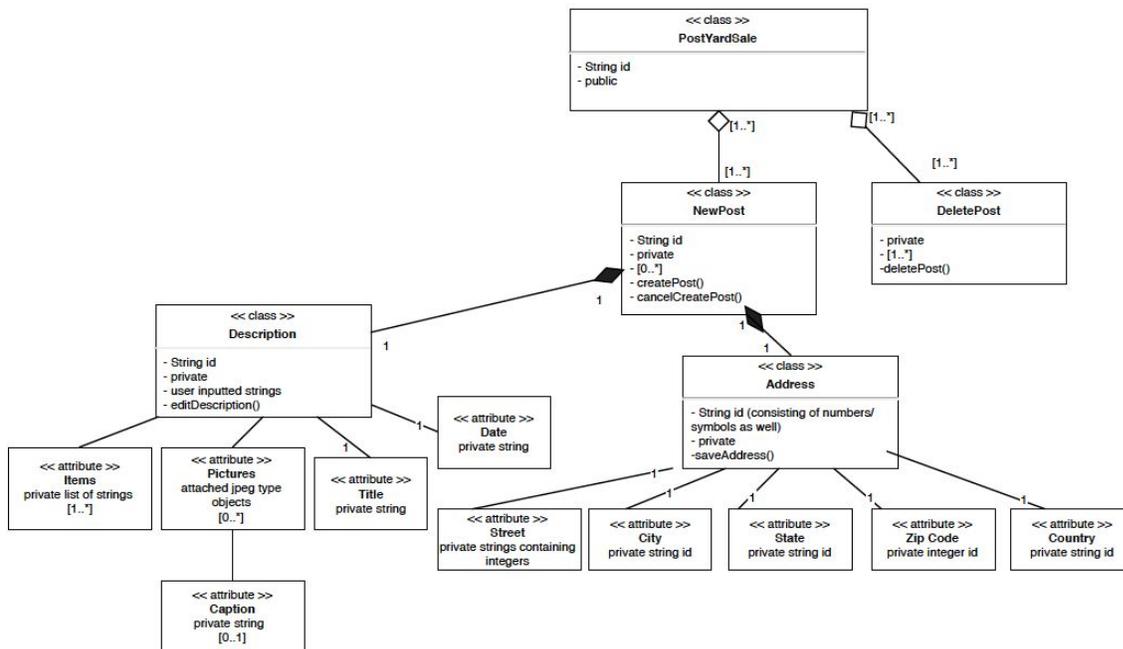
Use Case Diagram

Below is a preliminary use-case diagram:



Initial Class Diagram

Class Diagram



Project Goals

Our first project goal was to research and learn more about building a software application. Although we have both had experience programming and building tools or smaller-scale applications, we had never created a full front and back-end complete project. To accomplish this goal, we completed extensive research during Fall Quarter 2018 and created a [shell application](#)¹ that took a start and end location from a user, as well as an item to search, and output all locations from the start and end location that might contain said item. This shell application helped us learn more about building a more involved back-end to serve users based on Yelp's API.

Our second goal was to create a basic application that accomplished our goals (see motivation section above for more details). We ultimately hoped to implement 5 pages in the application, including a Home page, Search, Messages, Profile, and Settings. We also aimed to connect the application to Firebase, in order to save user login data.

Related Systems

When completing initial research regarding building our application, we wanted to compare and contrast our goals with other related systems that already exist. The three systems we found were: Craigslist, Facebook Marketplace, and a Yard Sale Treasure Map application.

Craigslist allows users to see sales of all kinds, not limited to yard or garage sales. The website itself is grid and list-oriented, making visibility of pages/items difficult for beginning users. With our application, we wanted to stray away from Craigslist's "sketchy" reputation and create a better organized, safer environment for users.

Facebook Marketplace had a better layout (in comparison to Craigslist), but had limited filtering and sorting capabilities and seemed quite cluttered in general. There was no map view or radial filtering option (based on location), something we hoped to implement with YardSail.

Lastly, the Yard Sale Treasure Map application had many broken features, a lack of categorization, and required users to pay a subscription fee. All of these related systems helped us rule out features we did not want, as well as plan things we hoped to accomplish with YardSail.

¹ <https://yardsale-224318.appspot.com/>

Project Overview

Our project can be broken up into two stages: research and development. During the research stage, we researched all tools we wanted to use, related systems, and completed all preliminary design and planning of our application. We were able to learn quite a bit about everything it takes to design an application, including: use-cases and use-case diagrams, creating practical storyboards, selecting tools/databases/libraries to utilize, sticking to a schedule and working as a team.

During the development stage of our project, we were able to learn first-hand how to develop an extensive software application, with full front and back-end features. We spent 8 weeks iteratively implementing pieces of our system, with back-end features being some of the last work we completed. We learned more about Firebase and Native React, two tools we had never used before this course. Additionally, we learned how to successfully work as a team to collaboratively develop our software piece-by-piece.

Implementation

Code

The entirety of our code can be found on our [github](#)². At first, we decided to utilize Google's Web Toolkit (GWT) to create a web application. This decision was easy to make, as both of us are advanced programmers in Java, yet are not as confident in programming in JavaScript. We made a basic shell application with Google's Web Toolkit, available [here](#)³. We additionally planned to use Google's Cloud Platform to use Google's material design components to help build the application easily.

Our plan to use these tools changed entirely as we began Winter Quarter. We realized it would not be feasible to create our extensive application from the shell application we had created, although it was a great starting point. After attempting to create our entire frontend programmatically, we stumbled upon specialized IDEs such as React Studio & Deco, which instantly made our lives easier. React Studio allowed us to use drag-and-drop components and export the code to our project, in order to build the application quicker and save time for the harder stuff (like connecting to Firebase). We were lucky enough to find a React Native [starter kit](#)⁴, which helped us tremendously

² <https://github.com/eplinglauren/YardSale>

³ <https://yardsale-224318.appspot.com/>

⁴ <https://github.com/mcnamee/react-native-starter-kit>

when starting our application from scratch. We were able to use this starter kit to build and customize our application, as we could add and remove whichever features we needed.

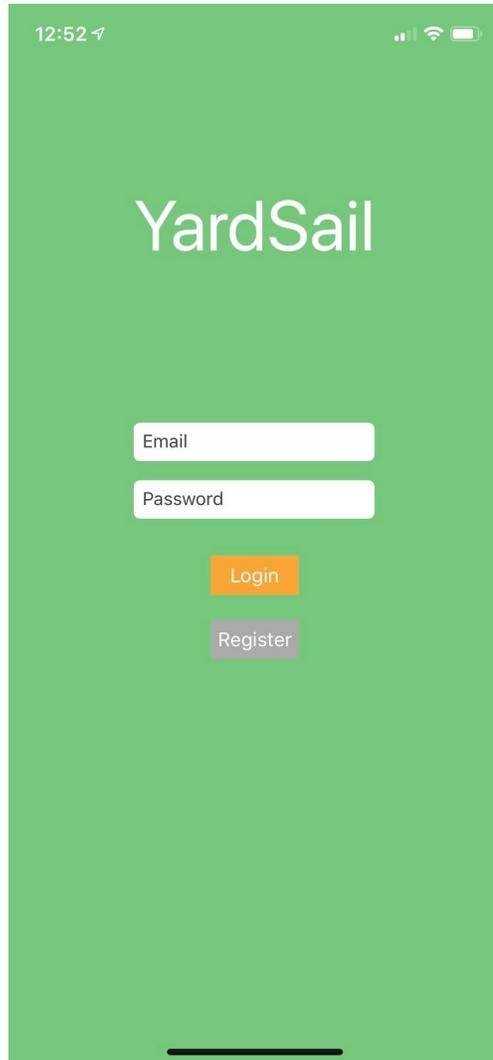
React Studio helped us understand how each UI component corresponds to the backend code. However, the starter kit further improved our understanding as we had a solid boilerplate application somewhat resembling what we needed, thus we were able to make minor adjustments to the source files and immediately see a response on the iOS simulator. In addition to our React Native implementation, Matt was also making the same app with Swift & iOS for his Mobile Application Development class. Compared to the Swift implementation, we found that it was much easier to create our app in React Native as it gave us the ability to standardize the design patterns.

One of the main reasons we chose to use React Native was its ability to provide a unified code base for three forms of the application (iOS, Android, and Web). React Native also propagates your design choices across these three forms which was an important as we wanted our app to eventually be available cross-platform. Of the many React Native starter kits out there, we ultimately chose Matt Mcnamee's boilerplate kit because of his well organized file structure, clean UI design, and boilerplate functionality to connect our React Native application to Firebase.

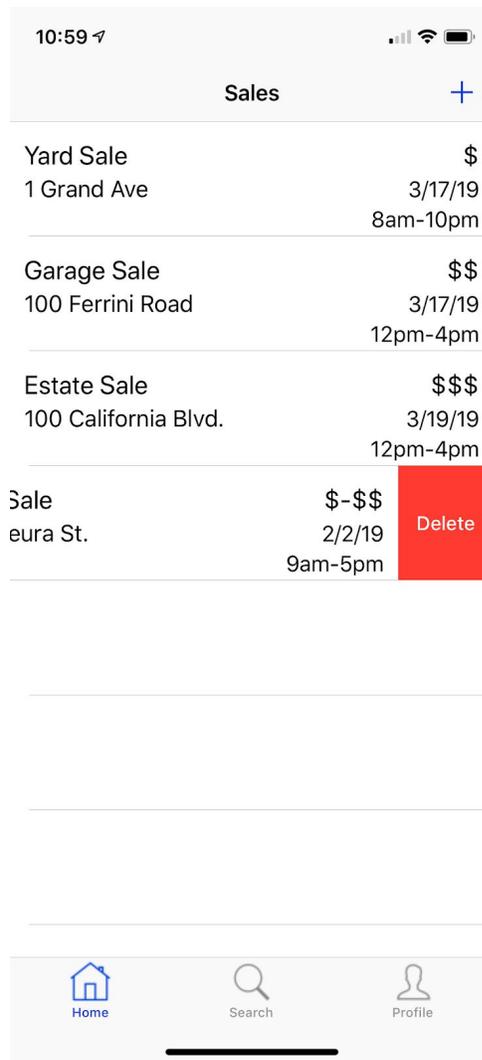
Our decision to use Firebase was easy to make as Prof. Randy Scovil of Matt's Mobile App. Dev. class demonstrated how straightforward it was to connect an iOS application in Swift to a database for cloud storage and user authentication. Both cloud storage and user authentication were absolutely necessary for our app in order to allow users to have a profile, post listings, and message other users.

System Screenshots

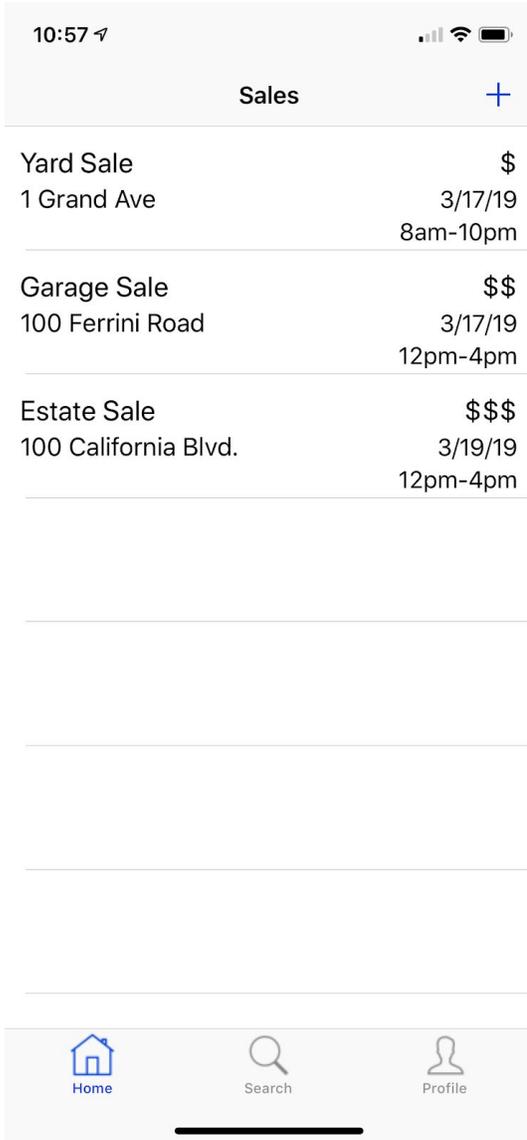
iOS App Made in XCode



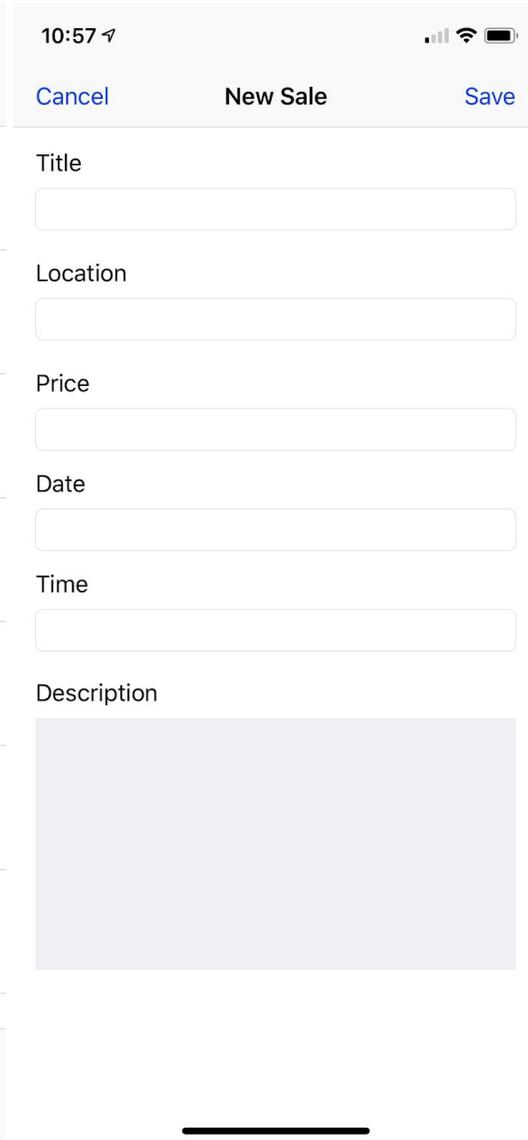
Login Page



Deleting Sale Post

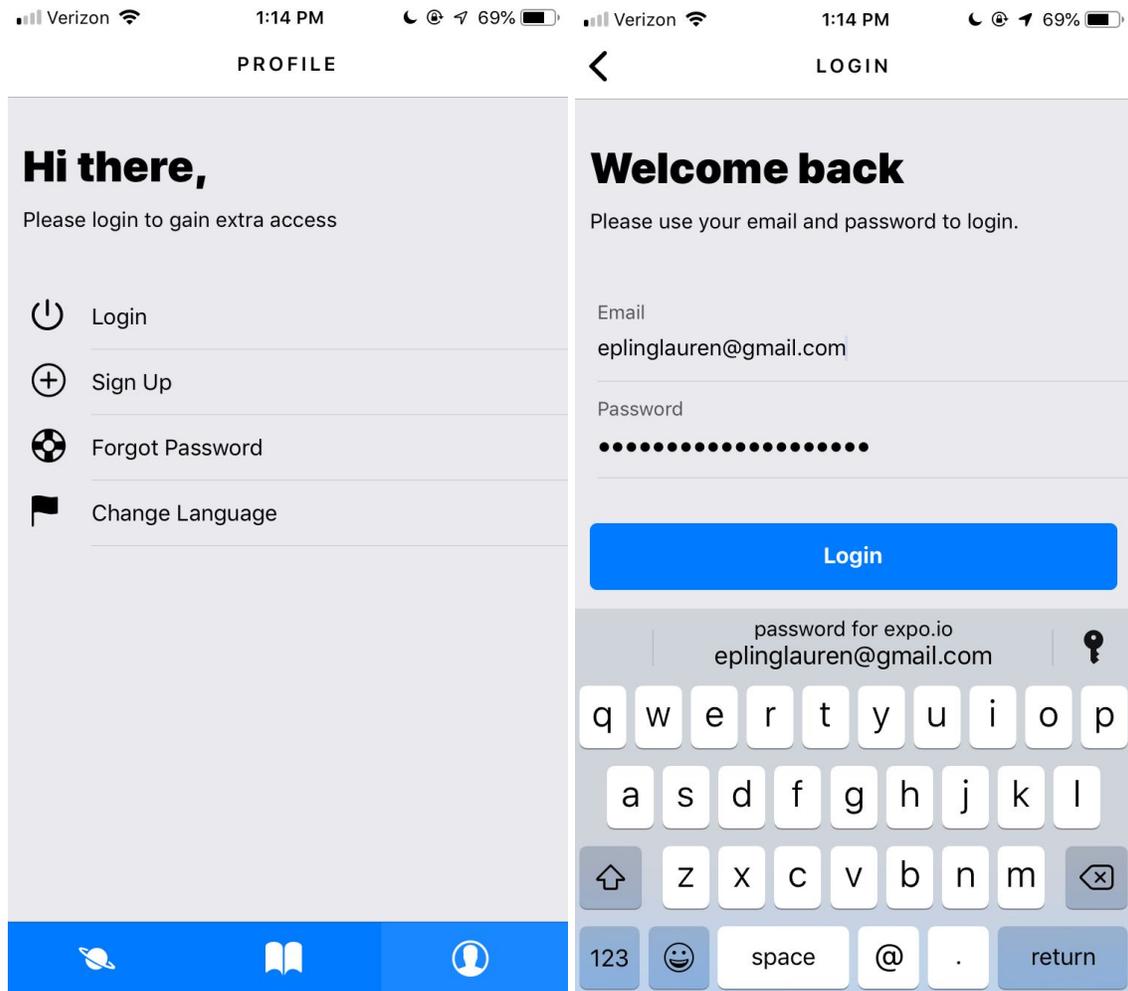


Home/Sales



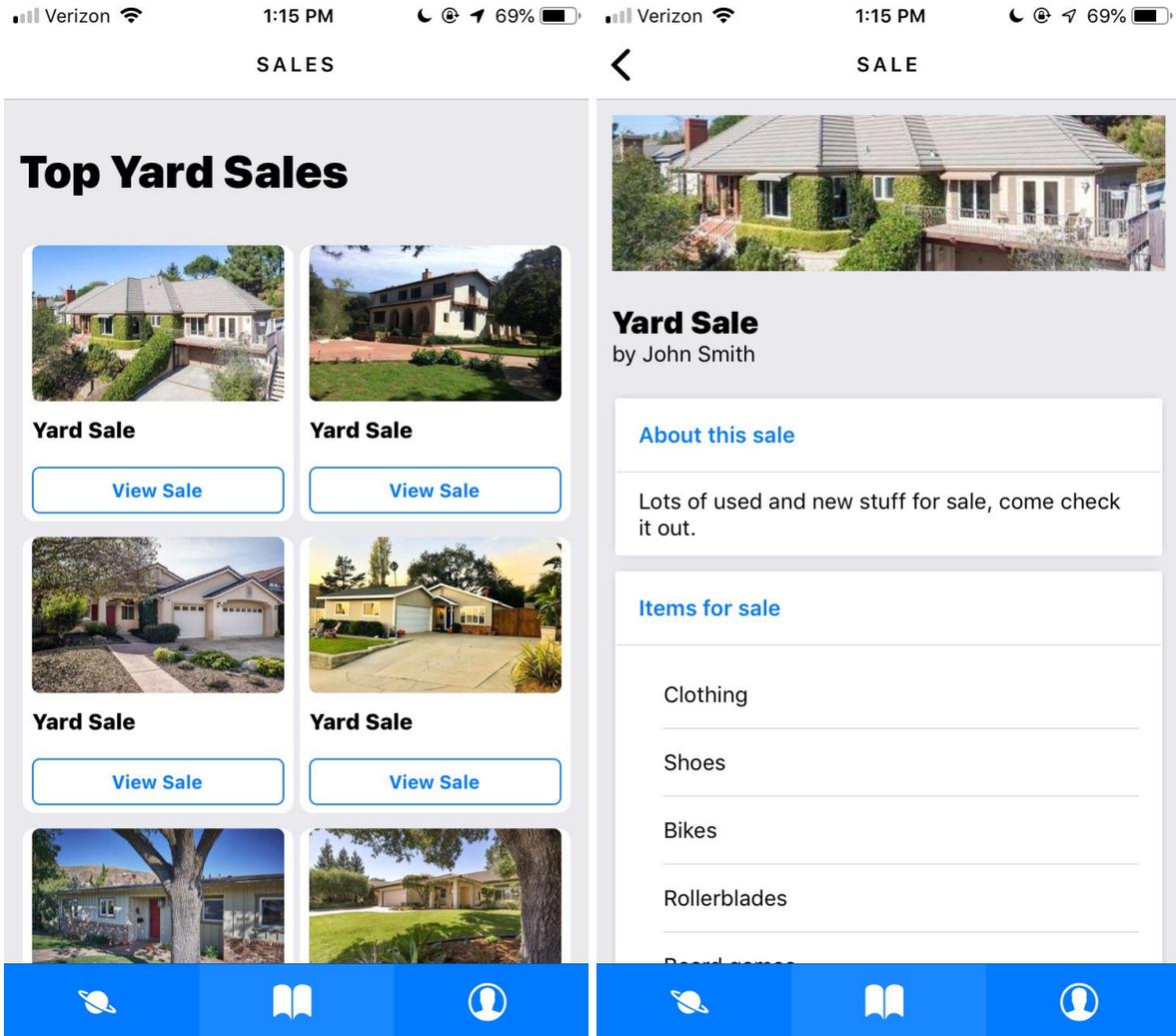
Posting New Sale

iOS/Android/Web App made with Native React



Login Page/Sign Up

Login/Welcome Back



Yard Sale Listings

Example Yard Sale Post

Conclusion

Findings

Our senior project was an amazing learning experience. We were able to explore application and web development, and a better comparison of how to implement applications in Swift and React Native. Considering neither of us had experience with either Swift or React Native at the beginning of Fall Quarter, we are proud of how far our skills have come with both tools in such a short amount of time.

We found that React Native provided more flexibility overall, as we were able to build a website, an iOS application, and an Android application using a single code base. Programming with

Swift in XCode did not provide the same ability, since it is dedicated for making only iOS/macOS/etc applications. However, we found that implementing certain back-end features of our application was easier with Swift, as Swift has modern, lightweight syntax and integrated debugging.

We would like to thank Professor Franz Kurfess, our advisor, for supporting the development of our project and application over the past two quarters at Cal Poly. It was a very rewarding experience to explore web and application development on our own for the last many months, something we had no experience with prior, and we are glad we chose to do so with such a supportive professor. Additionally, we learned how to effectively work as a team to develop an application we would be happy using ourselves, and happily completed much of what we set out to do.

Future Work

In the future we will continue to finetune certain aspects of our application, as well as add new features. This includes the messaging page, to allow users to message one another regarding sales, as well as the calendar page, to allow users to export sales to their device's calendar. Once we have the iOS in React Native fully implemented to our likings, we will work on the Android and Web applications to get them to compile and build successfully. From there we hope to deploy our projects to the respective application store and host the web application on our own domain. We truly hope that our application will be a competitor amongst established marketplaces such as Craigslist and Facebook Marketplace.

Resources

“Learn the Basics · React Native.” *React Native Blog ATOM*,

facebook.github.io/react-native/docs/tutorial.

Mcnamee. “Mcnamee/React-Native-Starter-Kit.” *GitHub*, 16 Feb. 2019,

github.com/mcnamee/react-native-starter-kit.

“Productivity for Developers, Performance for Users.” *GWT Project*, www.gwtproject.org/.

“React Native IDE.” *Deco*, www.decoide.org/.

“React Studio.” *React Studio*, reactstudio.com/.