Food Locker: An Online Marketplace for Automated Meals

by

Josiah Buxton

B.S. Computer Engineering, Elizabethtown College, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Masters of Science

Department of Computer Science

2019

This thesis entitled:
Food Locker: An Online Marketplace for Automated Meals
written by Josiah Buxton
has been approved for the Department of Computer Science

Professor Shivakant Mishra

Prof. Rhonda Hoenigman

Prof. Qin Lv

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Josiah Buxton, (M.S., Computer Science)

Food Locker: An Online Marketplace for Automated Meals

Thesis directed by Professor Shivakant Mishra

Food waste is a major problem for the human race, particularly in ”rich” industrial countries. We propose a creative, partial solution to this problem through the development of an online marketplace for left-over food and a physical infrastructure to secure the food throughout each transaction.

The software platform will consist of three separate portions: a mobile food consumer application, a mobile food provider application, and a web food provider application. Restaurants and other licensed food distributors can utilize the food provider applications to broadcast available left-over meals, which can be purchased at a discounted price. The hardware infrastructure will be composed of ”food lockers,” refrigerated units that automatically dispense food to consumers.

As this vision is ambitious, the goal of my thesis was to develop the software platform and a small prototype with two individual lockers to demonstrate the functionality of the entire system. An experiment was constructed to validate the correctness of multiple, synchronous calls to consumer/provider functions. Results indicate that this system is near ready for small scale deployment.

Dedication

To my family.

Acknowledgements

I want to thank the entire Computer Science department, but also a few influential faculty/staff who supported me throughout my time at the University of Colorado Boulder.

First, Rajshree Shrestha who answered any/all questions impeccably pertaining to the degree I was pursuing. She was an invaluable resource, particularly when I was pursuing avenues in professional development. Second, Shivakant Mishra who taught my favorite course in the CU Boulder Computer Science department (Distributed Systems) and who advised me throughout the last year. His advising strategy gave me an ownership mentality in my thesis and he always provided me with gentle direction at pivotal points throughout the process. Finally, Rhonda Hoenigman for whom I was a Teaching Assistant in two courses (Data Structures and Algorithms). During meetings and other interactions with Rhonda, I observed true leadership and management skills. Her ability to make just decisions on controversial issues in a timely manner led to incredibly smooth semesters. Also, her ability to understand a multitude of perspectives and communicate them, made the class feel cohesive and led to satisfaction for everyone who participated in the course.

I also wanted to thank my family and friends who have supported me always and they are the main reason why this thesis was completed.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

Food waste is a recent problem for the human race and already, there are mind-baffling statistics that demonstrate the scope and severity of it. The Environmental Protection Agency released a graphic that demonstrates the rate at which food waste is growing in the United States.

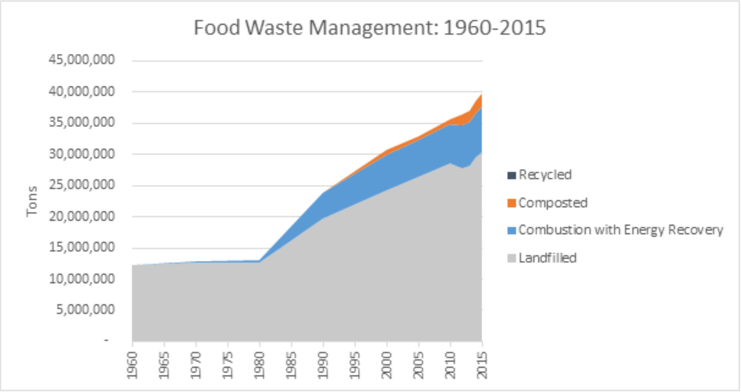


Figure 1.1: This graph is a depiction of food waste management estimates from the EPA (Environmental Protection Agency). Trends show the impact of different food waste management techniques in the United States, as well as the total generation of food waste from the United States as a country (residential and commercial)[4].

Since the 1980's, food waste estimates for the United States more than tripled. Even though this statistic is sensitive to a variety of variables (human population, food generated, etc.), the magnitude is enough to illuminate the growing problem of food waste. Another statistic from the Food and Agricultural Organization of the United Nations estimates that over a third of all produced edible food is either lost or sent to a landfill [11]. Pair this, with a statistic from the World Food Programme that estimates one in nine people go to bed on an empty stomach every night [14]. The paradoxical nature of these statistics indicates that the human race has the opportunity to solve a major logistical problem of diverting food which would normally be thrown away in a landfill, to hungry individuals. As there is not a single optimal solution that will solve this logistical problem entirely, multiple partial solutions can be applied to mitigate both food waste and world hunger.

1.2 Overview

The goal of this thesis was to provide a creative, partial solution to the food waste problem. After performing preliminary research and a multitude of empathy interviews [17] with food providers and consumers, our solution was the development of a platform to facilitate the transactions of residual food. I will begin by presenting related works, each of which heavily influenced our design decisions of the system. Next, I will introduce the design and implementation of the platform along with an experiment to validate the correctness of different functions within the system. Finally, I will present and analyze results of the experiment which will serve as an indication of the system being ready for small-scale deployment. Future work in this area could result in a fully functional, world wide platform for residual food transactions.

Chapter 2

Related Work

2.1 Introduction

There has been an enormous amount of research performed in the area of food waste, so this section will focus more particularly on previously implemented systems (hardware/software) pertaining to food waste/sharing. Restaurants throughout the world have been integrating more and more technology into their daily processes, due to its ability to drive revenue. A recent study proves this, where the authors analyze the effects of implementing different food sharing models (through technological systems) into existing restaurants. After a hierarchical cluster analysis of over 50 companies, the authors came to the conclusion that implementing one of the food sharing models at an organization lead to a larger social impact for the organization [10]. A few of the companies which are very similar to the idea of Food Lockers will be presented in this section, along with a few peer-reviewed publications.

2.2 Organizations

Waste No Food is a registered non-profit organization whose mission is to reduce the amount of food waste at different organizations. They do so, by providing a web-based marketplace application which connects organizations with excess food to charities in need of food [12]. Some of the problems addressed in their application are similar to potential problems in the scope of this thesis. For example, Waste No Food must validate registered charity organizations which is analogous to Food Locker validating registered restaurants with the state. Also, in order to increase the accessibility to the marketplace they created, Waste No Food has developed two mobile applications in addition to the web application. Because Waste No Food is a national organization and has partnerships with large well-known companies, their applied solutions to the stated problems seem successful.

Food For All is another organization that hosts a food-sharing platform for leftover food at different restaurants and organizations. Their business model gives organizations with leftover food the opportunity to broadcast it on a web-based application and have consumers purchase and pickup the food within a designated time interval [6]. The organization seems mostly geared towards restaurants who prepare food in large quantities and tend to have a lot of leftover (hotels, bakeries, etc.). Purchases give consumers a digital coupon to show participating restaurants. Feedback was gathered from a primary source who described the entire transaction process as "clunky and inconvenient." Currently, the startup company has participating restaurants in Boston and New York City and is continually expanding.

Eatsa is a company whose mission is to integrate technology into the restaurant business to lower costs and improve food quality. Rather than focusing on food waste, they focus on streamlining the dining experience. They provide a variety of scalable solutions for restaurants, which include hardware and software to support the business needs. A web and mobile application platform are both options that can be selected to be apart of a solution for a restaurant. One unique feature they offer, is a solution for order pickup called the "cubby pickup system" where fulfilled orders can be picked up by their owners at a secured location in the restaurant. Eatsa has been successful after re-branding from a "robot-run" fast casual eatery to a technology manufacturing company [3].

2.3 Peer Reviewed Publications

Timothy Richards and Stephen Hamilton recently published an article pertaining to food waste in the sharing economy [15]. They performed a study which investigated the "potential for commercial peer-to-peer mutualization systems (CPMSs)" by analyzing a current CPMS firm, Imperfect Produce. After developing an empirical model using data from Imperfect Produce, they were able to make empirical estimates for supply and demand in the market. There were many findings in testing for the viability of a secondary market which included ”ugly foods.” First, they found that increasing the variety of foods food suppliers had to offer led to a rise in consumer demand. Inversely, the food suppliers were more likely to become users on the site if there were more consumers present. The study also was the first to document ”indirect network effects” in CPMS organizations which is an early indicator of success for startup companies.

A group of professors from independent universities in Australia worked together to build a conceptual smart food container for food suppliers to fill when donating food to charity [2]. The container is designed with a multitude of sensors (proximity, temperature, humidity, RFID, vision) in order to detect when food is added to the containers and to maintain the correct environment for different foods. While the container was not physically built, design documentation was provided to conceptualize the entire system.

Professors from independent universities in Austria developed a public cooling system that would facilitate food sharing between individuals and organizations [13]. Focus groups and a field test was performed to gauge the general interest of these cooling stations. The team found that many people were hesitant to participate in using the system because of ”trust” issues with the food they would purchase. The focus groups gave valuable feedback on the design of the system, particularly the registration process and gamification of the system. The registration process was found to be too bulky, the system required too much information from the user. All of the individuals in the focus groups believed that gamification/rewards system would be a great feature to be added to the system. They also found that the most difficult hurdles in developing the system were the legal and hygienic issues.

Chapter 3

Methodology

3.1 Introduction

After initial research was performed, much time was spent in developing the architecture for the proposed system. Within this section, the final architecture, which includes all subsystems, will be presented as well as the experiment, which validates the correctness of the functions implemented in system.

3.2 Design Development

Before the system was even proposed, primary data was gathered to determine what features were desired by the potential users. Over seventy-five different undergraduate students from the University of Colorado Boulder were interviewed. The first few questions pertained to a student's willingness to purchase food, which was considered "left over", at a discounted rate. The responses from students were overwhelmingly positive in the sense that they were very willing to purchase this type of food. This may have been due to the narrow demographic of people we targeted in interviews, but regardless the responses made it clear there is a market for leftover food. Along with asking the students about their willingness to purchase leftover food, questions were asked about the largest complaint they had with their current meal plan on campus. The questions were designed to allow the interviewee to reflect on some sort of experience that invoked much emotion. A recurring theme was heard in the responses: students were disappointed with the time that the dining halls closed. They are unable to access the food they are paying for when they are awake and hungry. The feedback received from the students led to the design of an automated system that would make leftover food from universities' dining halls available to students twenty four hours a day.

Primary data was also gathered from restaurants/organizations to determine whether there was enough wasted food for a solution to be developed. Five to ten minute interviews were conducted with general managers at over twenty restaurant locations. In the interviews, questions about how much food waste was generated at a restaurant and if they thought it would be worth them addressing the problem (if it existed). From the interviews, it was clear that small restaurants who employed long term employees were the least likely to have large amounts of food wasted whereas larger restaurants with a high turnover rate were the most likely. This discovery led to multiple talks with the University of Colorado Boulder who fit the profile of a food provider that generated a lot of food waste. The university confirmed that they generated a lot of food waste but tried to donate as much as possible to different charities. They were willing to help pilot ideas to address food waste but they were adamant about having anything physical certified before adapting it into their current operations.

From all the feedback gathered, a potential solution was proposed to address food waste as well as address the problem of food inaccessibility different students at the University of Colorado Boulder experienced. Food lockers would provide a storage location for excess food that the University's dining halls possessed, and it would allow students to access meals that they've already paid for after the dining halls closed. The idea would be for dining hall employees to package up the excess food at the end of the day into individualized meals, host the meals on FoodLocker's online marketplace, and drop off the meals into the lockers so students could claim them. Design decisions were made to make this process as streamlined as possible, so as to be unobtrusive when introduced into dining hall employees' processes.

3.3 System Design

3.3.1 Firebase

Before development of any of the subsystems could begin, a database needed to selected to hold the information associated with the system. Some major features which were required to be present in the database included: offline functionality of the applications utilizing the database, ability to upload pictures, scalability, analytics, and error reporting. The features are ordered by importance from greatest to least. Firebase is a NoSQL database which exhibits all of the required features to some extent, as well as a multitude of other features. Food Locker only utilizes three of the major modules in Firebase: Authentication, Cloud Storage, and Realtime Database. The authentication module allows the Food Locker system to create, store, and manage unique users associated with the application. The Cloud Storage module gives the system the functionality to store and retrieve different photos related to users/organizations. The Realtime Database module stores almost all of the other information associated with the system formatted in Javascript Object Notation (JSON). All of the modules utilized in Firebase are under the "Pay to Scale" marketing model, which means the service will be free of cost until a certain threshold of traffic is met. Firebase is the core service which every other subsystem interacts with.

3.3.2 Consumer Mobile Application

The first subsystem that was developed was the consumer mobile application. This application allows consumers to browse available meals within different food lockers, purchase the meals, and pickup the meals. The application was developed on the React Native framework mainly due to the astonishing amount of documentation and 3rd party packages on the web, which allowed for rapid troubleshooting of bugs/issues and agile development of components. Also, the framework utilizes a single source of Javascript and compilers can be downloaded, which will convert the Javascript to IOS and Android applications. This means that one code base needs to be maintained in order to develop an application for both mobile platforms.

Table 3.1: Suite of the functions implemented in the consumer mobile application.

List of Functions	
Search Food Locker locations using Google Maps API	Search different available meals using a phrase (Categorically)
Login	Logout
Signup	Search all available meals
Search available meals in a particular Food Locker	Search available meals by meal category (Asian, American, etc.)
Purchase a meal using Stripe API	View order history
Navigate to Food Locker location from current location/specified location using Google Maps API	View/edit personal information including custom headshot
View/edit billing information	View/edit application specific settings

Table 3.1 lists the functions implemented in the consumer mobile application. The application grants the user the ability to search for meals in a variety of different contexts. This was done to ensure that the user can easily view and purchase the available meals. Refer to Appendix A to view screen shots of all the screens in the application.

The entire consumer mobile application consists of 38 different packages (Refer to appendix B for a complete listing). Two of the major development packages that were used were expo and babel. Expo is a toolchain specifically developed for React Native applications. It provides a framework for hosting the application as it was being developed and offers "hot-reloading" functionality which automatically reloads the mobile application on a test device when changes are saved to the code base. This saved an enormous amount of time during the development process. Babel is another tool chain that transcribed code written in newer versions of JavaScript (ES2015+) to older versions of JavaScript in order to provide functionality to older operating systems. It also transcribes JSX and React syntax (which is the base of React Native) into JavaScript.

Four of the major packages that provided a lot of functionality to the overall system were Stripe, React, Yaqrcode, and Firebase. Stripe is a package that provides a channel of communication to an existing stripe database. The database stores sensitive user information, including credit cards, and allows users to make purchases from different organizations. Currently, when a transaction is initiated, money sent from the consumer is deposited into a bank account associated with Food Locker Technologies LLC. The information is recorded and revenue generated will be sent to their respective organizations at the end of each month.

React is a JavaScript library for building different user interfaces. It provides almost all of the visual components and helped expedite the development process. Yaqrcode is a package that generates QR codes after given a string of characters. This package was used to streamline the processes of picking up a meal at a food locker and loading a meal into the food locker. Firebase is the package that provides a channel of communication to each of the Firebase modules: authentication, cloud storage, and realtime database. Each of the 38 packages installed in the project either provides some sort of functionality or enhances the user experience.

3.3.3 Restaurant Mobile Application

The second subsystem that was developed was the restaurant mobile application. This application allows restaurant employees to generate different virtual meals that can be broadcasted to consumers. After a meal is generated, a QR code is associated with the meal and it is used to open an empty locker door on a food locker. After the meal is placed in the locker and the door is closed, the meal is then broadcasted to consumers on a first come, first serve basis. Much of the codebase from the consumer mobile application was used in the development of the restaurant mobile application. To accentuate this point, all of the packages in the restaurant application are present in the consumer application. The only differences in the applications are the functions provided to the user and the screens themselves.

Table 3.2 lists all of the functions implemented in the restaurant mobile application. Refer to Appendix C to view all of the screens associated with the restaurant mobile application.

Table 3.2: Suite of the functions implemented in the restaurant mobile application.

List of Functions	
Generate a unique, virtual meal to be placed in a food locker	Generate a new template to save time when generating meals
View/edit existing templates	View food locker information (claimed meals, active meals, recently picked up meals)
View/edit application specific settings	View/edit personal information including custom headshot
Navigate to Food Locker location from current location/specified location using Google Maps API	View organization associated with current user
Login	Logout

3.3.4 Restaurant Web Application

The third subsystem that was developed was the restaurant web application. This application was developed in order to provide restaurants with an alternative method of accessing FoodLocker's online marketplace platform. Feedback was received from restaurants after building the restaurant mobile application, and the largest concern from the managers of the restaurants was that the mobile application would give an excuse for employees to be on their phones an unreasonable amount of time. Since almost all restaurants own some sort of computerized POS system that already contains a web browser, the web application was a less intrusive solution to accessing the online marketplace as well.

The restaurant web application contains all of the same functionality laid out in Table 3.2 (functions implemented in the restaurant mobile application). The web application also implements a statistics screen, which allows restaurant employees to view data associated with the meals broadcasted and sold through the FoodLocker framework. It also implements a calendar screen, which is still in the development phase, and will eventually be connected to a restaurants inventory. It will show restaurant employees when raw ingredients were purchased and will provide notifications to the users when ingredients are reaching the end of their shelf life.

The web application utilizes 44 total packages (refer to Appendix F for a complete listing).

Most of the packages were similar to the packages implemented in the mobile applications. Two major unique packages that were required for the web application to function properly were Redux and CoreUI. Redux is a state container that helped maintain a user's "state" throughout the application. This stored all of the relevant data pulled from Firebase and allowed the application to access this data as the user browsed between different pages. CoreUI was a free template for an administrative dashboard. It provided the styling of the web application.

Refer to Appendix E to view all of the screens associated with the restaurant web application.

3.3.5 Food Locker

The final subsystem that needed to be designed and constructed was the physical food locker that secures meals during each transaction. Due to time constraints, the food locker was designed to minimally demonstrate the functionality of the entire system. Note that this means the prototyped locker is not currently temperature controlled.

The locker is a 44" x 17" x 13" wooden box that is compartmentalized into two individual lockers and an electrical component container. Each individual locker contains an electromagnetic locking mechanism that is controlled by a Productivity 1000 PLC (programmable logic controller). The locking mechanisms are powered by a 60 watt 12V power supply and can supply up to 600 lbs of locking force.

A Honeywell MK7580 2 dimensional barcode scanner is mounted on top of the locker. The scanner is connected via USB to the computer which runs the C# application and Python script which allows users to interface with the locker. When users scan their QR code associated with an order, the information is retrieved over USB through the python script. The python script then communicates with Firebase to determine which individual locker the meal is stored in. When the locker is determined, the python script communicates over Modbus TCP/IP to the PLC, which then sends a control signal to the electromagnetic locking mechanisms to open the correct door.

The bill of materials for the food locker is specified in Appendix G. User manuals for any component are available by request. An initial 3D model was developed in Google Sketchup and screenshots are available in Appendix H. Pictures of the finished locker are also available in Appendix H.

3.3.6 Experiment

After spending a few months developing the system, an experiment was developed to validate the correctness of the functions implemented within the software sub-systems. The three day experiment consisted of a multitude of users creating accounts, performing a set of daily functions, and logging data associated with the functions into a Google Sheet. The data logged from the users was then cross-referenced with the data within the Firebase database at the end of the experiment which provided an indication of which functions were correct/incorrect.

In order to perform the experiment, first a list of twenty eight users was compiled. The list consisted of the name of the participant, his/her email address, a simulated user category (Consumer or Restaurant Employee), and a user ID. Five out of twenty eight participants were placed in the restaurant employee category and twenty three participants were placed in the consumer category. This was done to simulate an accurate ratio of consumers to restaurant employees when the system is publicly available. Because the participants both used iOS and android devices, the mobile applications were hosted on both Apple's App Store and Google's Play Store. Apple provides a tool called TestFlight, which allows for closed beta testing of an application before making it publicly available. Google offers the same functionality and both these tools were utilized for this experiment.

Next, a list of twenty one unique functions for the participants to perform was developed. Fourteen out of twenty one functions were implemented functions in the consumer application and the same is true for the restaurant application. There were seven functions associated with a user's profile which were implemented in both applications.

Finally, schedules were compiled and shared to each individual user. Users who were in the consumer simulated user category were asked to complete five different tasks each day, that would take them no longer than ten minutes to complete. Conversely, users in the restaurant employee simulated user category were given ten daily tasks to complete which mainly consisted of hosting meals in order to ensure there was enough meals hosted for consumers to virtually purchase (the payment system was disabled so no money was transferred during virtual transactions). The schedules contained columns for logging pertinent data and feedback associated with the performed functions. The data collected in these columns were used to generate the results in the next section. To view templates of these schedules, refer to Appendix I.

Chapter 4

Results

The performed experiment returned data that allowed us to understand the current state of the developed system and how the system performed with a multitude of users using different devices. Twenty five out of the twenty eight participants in the experiment installed the application. 56% of those users installed the iOS application from TestFlight and 44% installed the android application from the Play Store.

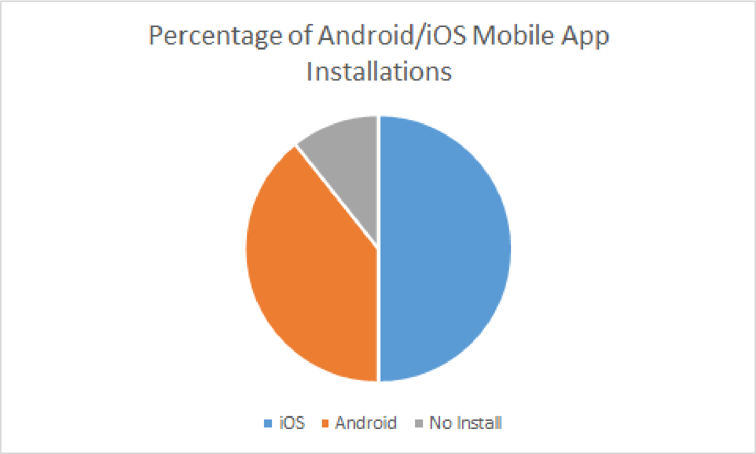


Figure 4.1: This graph is a depiction of the percentages of iOS, Android, and no installations associated with the mobile applications during the course of the 3 day experiment.

After all of the applications were downloaded the first day, the simulated consumers were asked to test updating their personal information associated with their account. For a portion of iOS users, the consumer application exhibited a bug where the navigation drawer would appear as a blank white screen. This behavior rendered six out of the fourteen functions unavailable to the user. The simulated restaurant employees were tasked with uploading 5-10 meals so that they would be available the next day for the consumers to virtually purchase. Over the next two days, consumers mainly were tasked with purchasing meals and testing two other functions associated with navigating to the locker that held their order. Restaurant employees were tasked with uploading more meals/templates and testing out updating information in their user profile. Below are two different graphs that show different statistics associated with the number of meal purchases and listings over the course of the experiment.

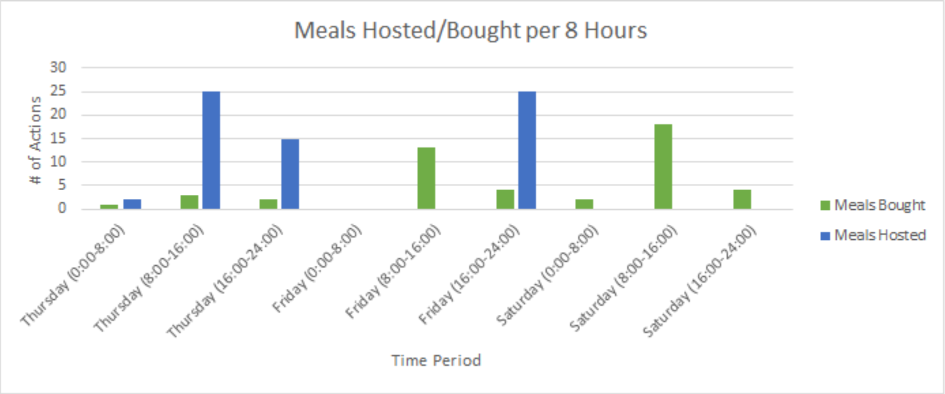


Figure 4.2: This graph is a depiction of the number of meals listed vs the number of meals purchased per 8 hour periods over the course of the experiment.

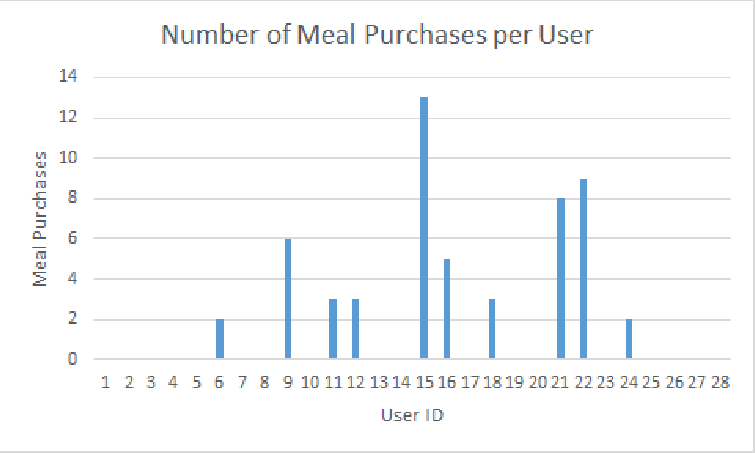


Figure 4.3: This graph is a depiction of the number of meals purchased by all of the users over the course of the experiment.

In figure 4.2, the shape of the data gathered was expected, but the magnitude of the meals purchased was half of the expected value. This was due to a bug in the iOS mobile consumer application. Users could purchase a meal but the payment information field was undefined due to the disabling of the payment system immediately before hosting to TestFlight. The undefined field invalidated the entire transaction but did not crash the application. Figure 4.3 also gave unexpected results due to this bug in the iOS mobile consumer application. Because 56% of all the participants were using iOS, Figure 4.3 showed less than half of the participants made a transaction.

After the experiment completed, all of the Google Sheets containing the participants' schedules and feedback were collected. Each of the "Relevant Information" cells from completed actions was then verified in the database at the appropriate depths. All information from successful transactions were present within the Firebase database and information from the unsuccessful transactions were not present. This was because of unit tests that were implemented within the mobile applications to verify the data before uploading to the database. This ensured that an error would not propagate to the database and therefore potentially crash the applications that interface with it. In figure 4.4 below are two different stacked bar charts that depict the number of correct vs incorrect functions tested in the experiment in both mobile applications. The large amount of incorrect functions in the iOS application were due to the development of the original application on an Android device. When the application was going to be hosted to the App store and Play store, compiling the iOS application led to finding an incompatible package. When I removed the incompatible package, I introduced a few bugs in the payment and profile information update process.

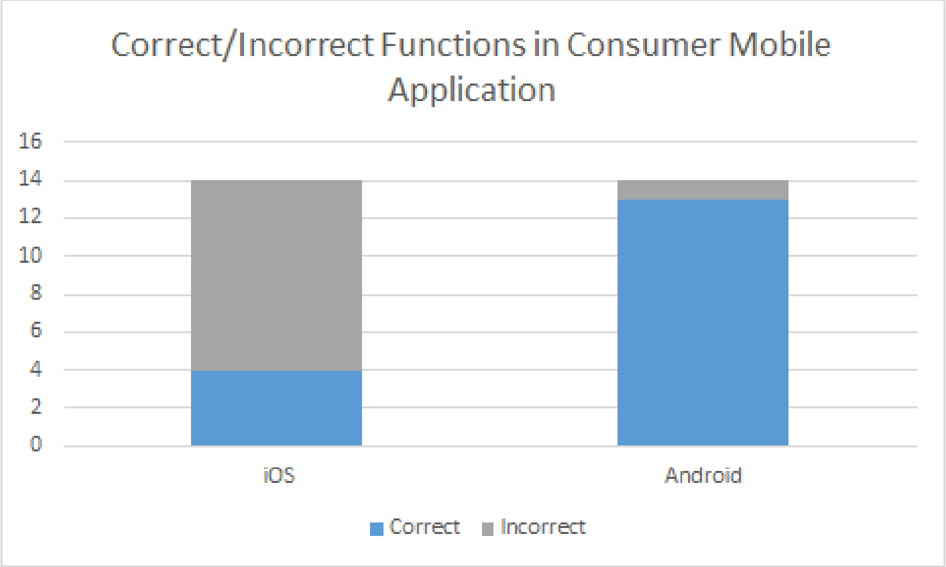


Figure 4.4: This graph is a depiction of the number of correct vs incorrect functions that were tested in the experiment. Data is shown for both iOS and Android applications.

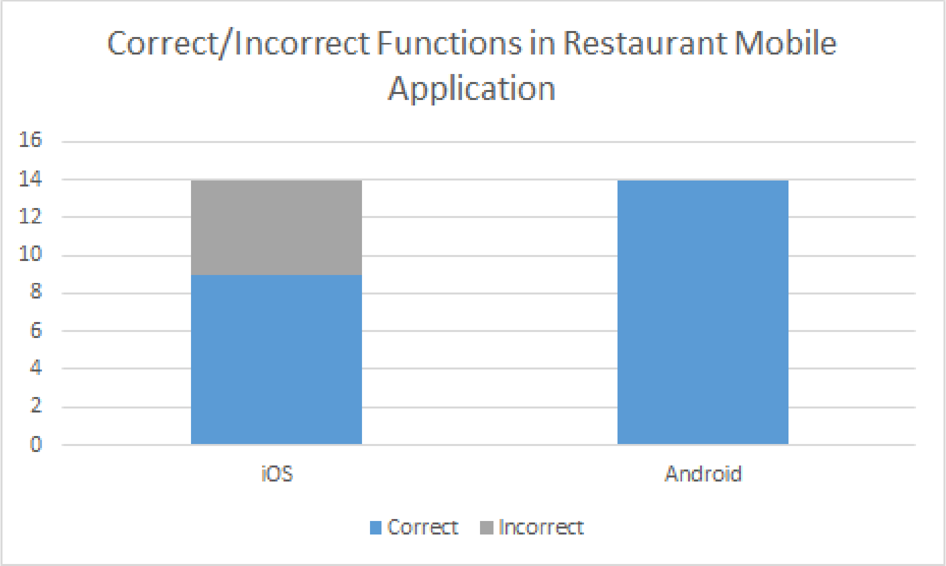


Figure 4.5: This graph is a depiction of the number of correct vs incorrect functions that were tested in the experiment. Data is shown for both iOS and Android applications.

Chapter 5

Discussion

The data presented in the results section makes it clear that there is still work to be done before this system will be able to be tested in a professional environment. The experiment illuminated a list of fourteen bugs to fix and functionality to add into the next version of the consumer application. Since the experiment, thirteen out of the fourteen items have been addressed and new versions have been posted to the Google Play Store and App Store. A table depicting these fourteen issues are available in Appendix J. The results from the experiment also indicate that the database is well suited to support the traffic generated from a single small-sized restaurant (around twenty transactions per day). Users reported a snappy interface, even during transactions when information is being transferred to and from the Firebase database. More experiments will need to be run in the future with a larger participant pool in order to determine whether this system is ready for full-scale deployment.

Future work could include another experiment to test the hardware developed. The hardware has been tested on a small scale with only two to three users, but not formally, and with a larger user pool. Running another small scale experiment in a professional environment will give us the opportunity to test the social utility of the system and gauge how much food a single locker can save as well as how much extra revenue could be generated for the restaurant. We estimate small restaurants could generate a few hundred dollars of extra revenue and save twenty to thirty meals a day (five to twenty pounds of food). Also, the mobile applications could have gamification implemented within them to motivate users to utilize the application more. Design reviews must be done with state food safety departments in order to get the physical hardware to a certifiable state. After becoming certified, the lockers will be able to be sold to restaurants on a large scale.

All in all, Food Locker has been developed to a point where it's almost ready to be dispensed into a small-scale professional environment. Most of the functions implemented in the mobile applications are correct and a physical prototype of a food locker has been built. Work will be continued in the future to gauge the social utility of the system before it's administered to restaurant locations around the country.

Bibliography

[1] C. R. Chen and Rachel J. C. Chen. Using two government food waste recognition programs to understand current reducing food loss and waste activities in the u.s. Sustainability, 10(8):2760, 2018. Name - Environmental Protection Agency–EPA; Copyright - 2018. This work is licensed under https://creativecommons.org/licenses/by/4.0/ (the License). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2019-02-05; SubjectsTermNotLitGenreText - United States–US.

[2] Venura S. A. A. A. Don, Seng W. Loke, and Arkady Zaslavsky. Iot-aided charity: An excess food redistribution framework, 2018. Copyright - Copyright The Institute of Electrical and Electronics Engineers, Inc. (IEEE) 2018; Last updated - 2018-11-07.

[3] Eatsa. Eatsa: About the company.

[4] EPA. Facts and figures about materials, waste and recycling.

[5] Facebook. React native a framework for building native apps using react.

[6] FoodForAll. Food for all.

[7] V. Jain. ebin: An automated food wastage tracking system for dormitory student's mess. In 2016 International Conference on Internet of Things and Applications (IOTA), pages 52–56, Jan 2016.

[8] Lisa Jennings. What restaurants can do to reduce food waste. Food management, Feb 08 2018. Name - Aramark Corp; Twitter Inc; Cornell University; Bon Appetit; Copyright - Copyright Penton Media, Inc., Penton Business Media, Inc. Feb 8, 2018; Last updated - 2018-06-14; SubjectsTermNotLitGenreText - United States–US; New York; Massachusetts; San Francisco California.

[9] Arlene Karidis. Refed's vared discusses what's next in the quest to reduce food waste, Feb 02 2017. Copyright - Copyright Penton Media, Inc., Penton Business Media, Inc. Feb 2, 2017; Last updated - 2017-02-06.

[10] Laura Michelini, Gennaro Iasevoli, and Ludovica Principato. Understanding food sharing models to tackle food waste. Ecological Economics, 2017. SuppNotes - Ecological economics. 2017, Elsevier B.V.; Pre-press version; Last updated - 2018-05-29.

[11] FAO of the UN. Food and agricultural organization: Statistics.

[12] Sakis Platis, Josh Weeks, David Kinch, James Beard, Teri Hope, Manresa, and Los Gatos Coffee Roasting Company. What is wastenofood.org?

[13] Margit Pohl, Elisabeth Weisenbock, Siegfried G. R. Wauschek, Maria Kalleitner-Huber, Gabriele Mraz, and Gabriele Bernhofer. Designing cooling stations for food sharing in public spaces, 2017. Copyright - Copyright The Institute of Electrical and Electronics Engineers, Inc. (IEEE) 2017; Last updated - 2018-06-14.

[14] World Food Programme. Zero hunger.

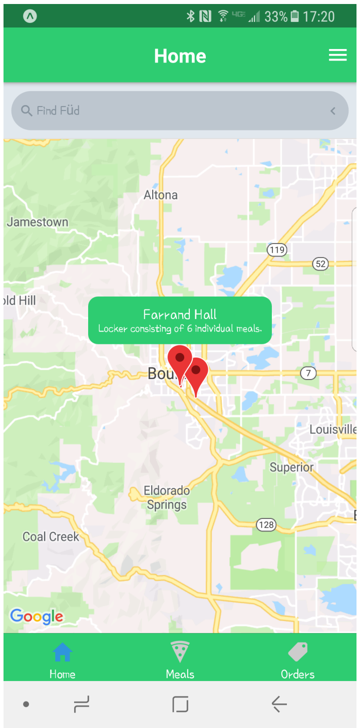
[15] Timothy J. Richards and Stephen F. Hamilton. Food waste in the sharing economy. Food Policy, 75:109–123, 02 2018. SuppNotes - Food policy. 2018 Feb., v. 75 Elsevier Ltd; Last updated - 2018-05-29.

[16] Kirsi Silvennoinen, Juha-Matti Katajajuuri, Hanna Hartikainen, Lotta Heikkilae, and Anu Reinikainen. Food waste volume and composition in finnish households. British Food Journal, 116(6):1058–1068, 2014. Date revised - 2014-08-01; Number of references - 26; Last updated - 2014-08-05.

[17] Techstars. Understand your customers - techstars entrepreneur's toolkit.

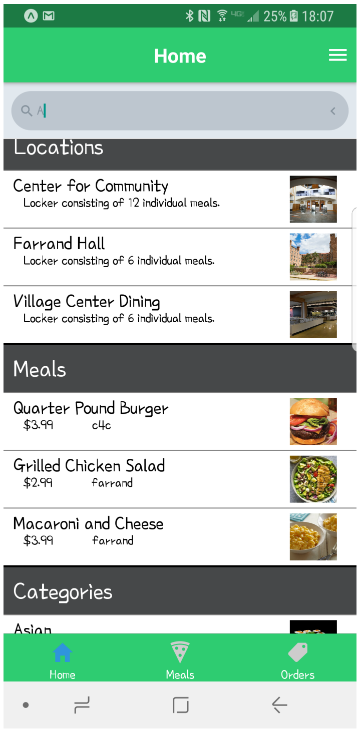
Appendix A

All Consumer Mobile Application Screens



Farrand Hall
Locker consisting of 6 individual meals.

Figure A.1: Home Screen after a user has logged in. Shows different food lockers (using map markers) in a radius around the user's current location. Meal radius is specified in user's settings. A search bar is present at the top to search meals/location/categories using a phrase.



Locations

Center for Community
Locker consisting of 12 individual meals.

Farrand Hall
Locker consisting of 6 individual meals.

Village Center Dining
Locker consisting of 6 individual meals.

Meals

Quarter Pound Burger
$3.99 c4c

Grilled Chicken Salad
$2.99 farrand

Macaroni and Cheese
$3.99 farrand

Categories

Asian

Figure A.2: Screen displaying the overlay that opens when a user begins typing a phrase in the search bar. The inputted phrase is searched over three different categories: locations, meals, and categories. Each line item is touchable and links a user to the associated item's screen.

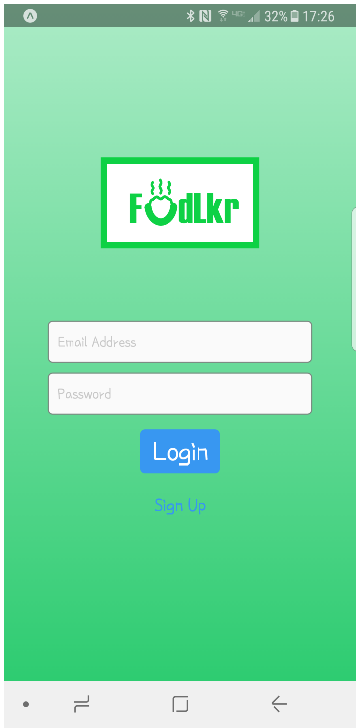


Figure A.3: Login screen, which allows users to log into their personal account. They have the option to sign up which sends them to a different screen almost identical to this one but with an extra confirm password input text box. A successful login sends the user to the home screen.

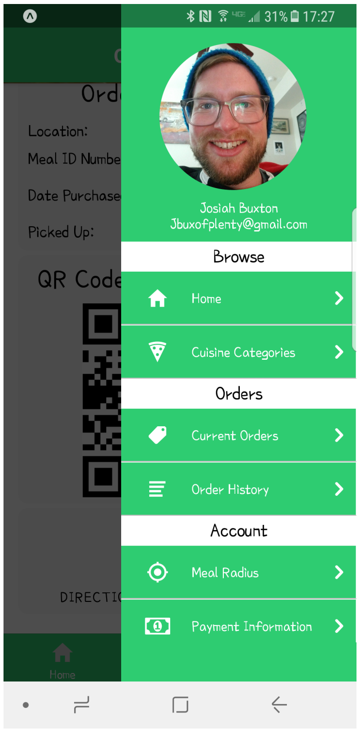
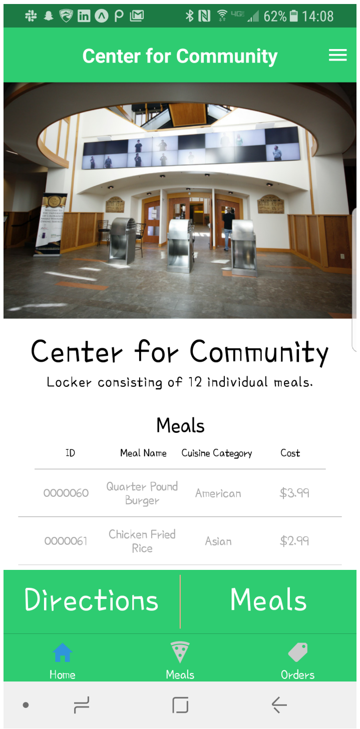


Figure A.4: Drawer overlay screen that enters the user's view when the upper right hand icon is touched (present on almost every screen). Lists different links that enable the user to navigate to most of the screens in the application.



Center for Community

Locker consisting of 12 individual meals.

Meals

ID	Meal Name	Cuisine Category	Cost
0000060	Quarter Pound Burger	American	$3.99
0000061	Chicken Fried Rice	Asian	$2.99

Figure A.5: Location screen that shows the information associated with a particular food locker. You can get directions to the locker using the Google Maps API or view all of the meals currently in the food locker.

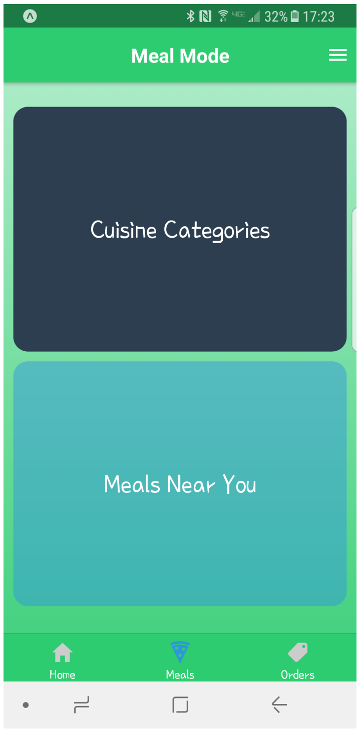


Figure A.6: Meal search screen that allows users to choose to browse meals filtered by cuisine categories or without a filter.

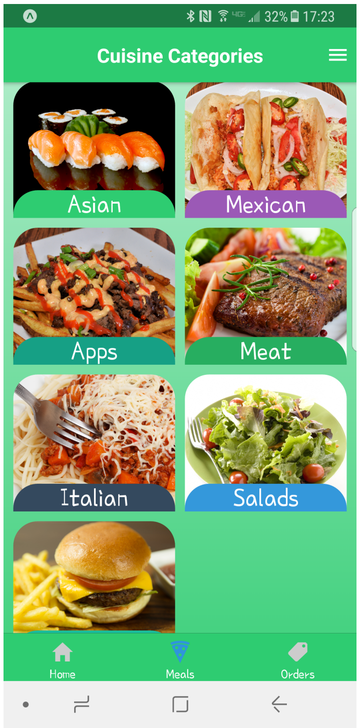


Figure A.7: Cuisine categories screen that shows the 7 different available cuisine categories. Touching one of the categories links the user to the meals within the cuisine category, which are within the user's meal radius.

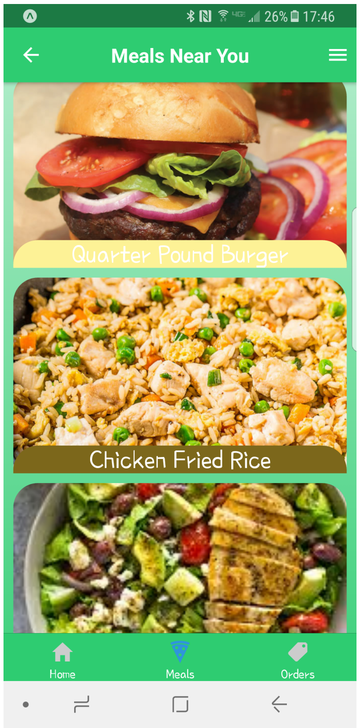
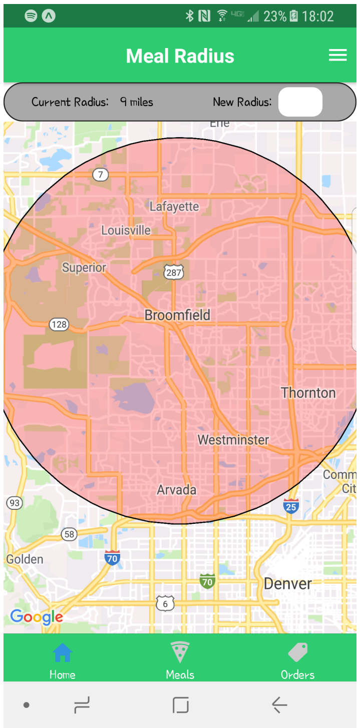
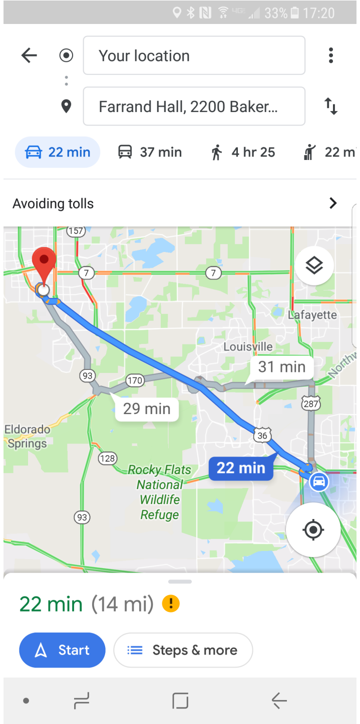


Figure A.8: Meals screen that displays a list of meals. Meals change based on what was the user's previous screen. Touching one of the meals links the user to more information about the meal and gives the option to the user to purchase the meal.


Meal Radius

Current Radius: 9 miles New Radius:

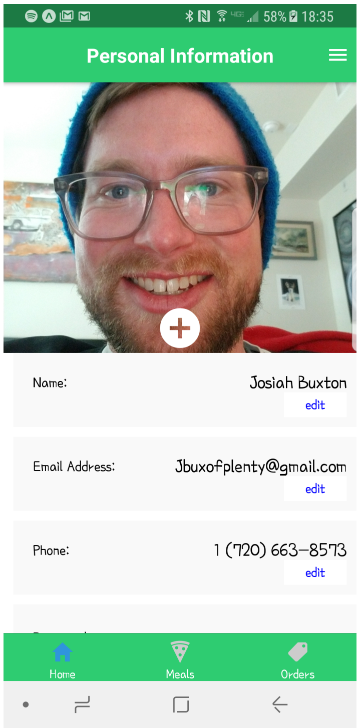
Figure A.9: Meal radius screen that allows user to select the radius (in miles) of the searching filter. All information within the food-locker system that is outside of the radius is not shown in the consumer mobile application's screens.


Your location

Farrand Hall, 2200 Baker...

22 min (14 mi)

Figure A.10: Directions screen which utilizes Google Maps API. Destination location is sent from the previous screen. Source location is populated with the current location of the phone.



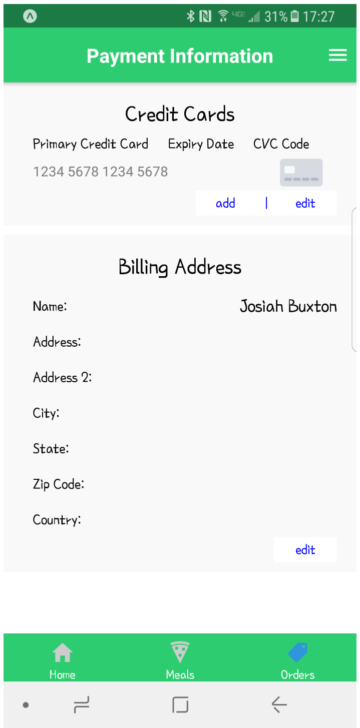
Personal Information

Name: Josiah Buxton
edit

Email Address: Jbuxofplenty@gmail.com
edit

Phone: 1 (720) 663–8573
edit

Figure A.11: Profile information screen which allows users to view/edit 5 different attributes associated with their user profile: head shot, name, email address, phone, and password



Payment Information

Credit Cards

Primary Credit Card	Expiry Date	CVC Code
1234 5678 1234 5678		

add | edit

Billing Address

Name: Josiah Buxton

Address:

Address 2:

City:

State:

Zip Code:

Country:

edit

Figure A.12: Billing information screen which allows users to view/edit billing information data so the user can purchase meals. All data is stored and retrieved through the Stripe API in order to ensure security.

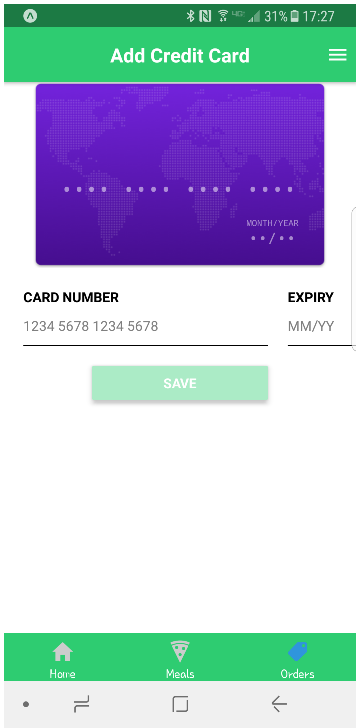
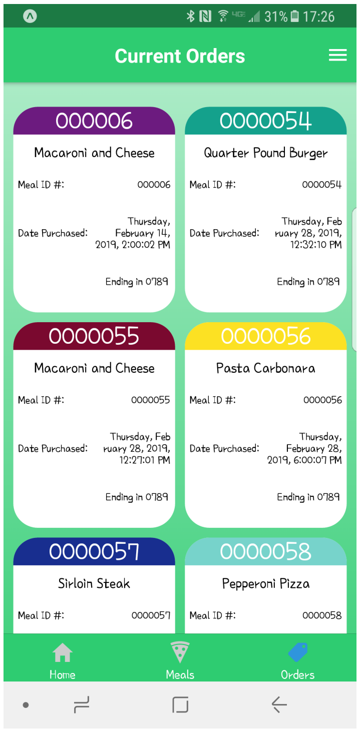


Figure A.13: Screen that allows users to add a new credit card to their profile. Credit card information is dynamically updated on the virtual card as the user enters it. All data is stored and retrieved through the Stripe API in order to ensure security.



000006
Macaroni and Cheese
Meal ID #: 000006
Date Purchased: Thursday, February 14, 2019, 2:00:02 PM
Ending in 0789

0000054
Quarter Pound Burger
Meal ID #: 0000054
Date Purchased: Thursday, February 28, 2019, 12:32:10 PM
Ending in 0789

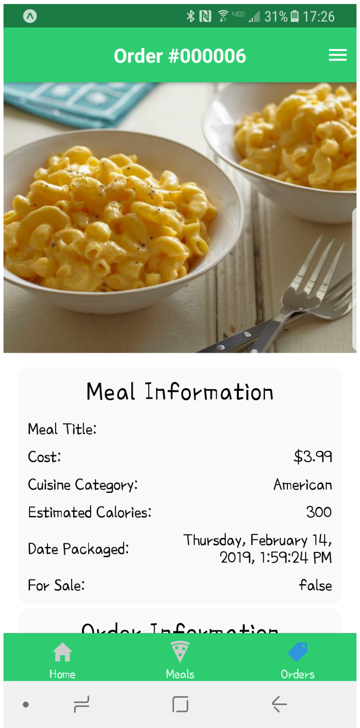
0000055
Macaroni and Cheese
Meal ID #: 0000055
Date Purchased: Thursday, February 28, 2019, 12:27:01 PM
Ending in 0789

0000056
Pasta Carbonara
Meal ID #: 0000056
Date Purchased: Thursday, February 28, 2019, 6:00:07 PM
Ending in 0789

0000057
Sirloin Steak
Meal ID #: 0000057

0000058
Pepperoni Pizza
Meal ID #: 0000058

Figure A.14: Orders screen that allows user to view current orders or a history of orders. A snippet of order information is displayed on each of the cards and touching one of the cards links the user to a screen containing more information about the associated order.



Order #000006

Meal Information

Meal Title:

Cost: $3.99

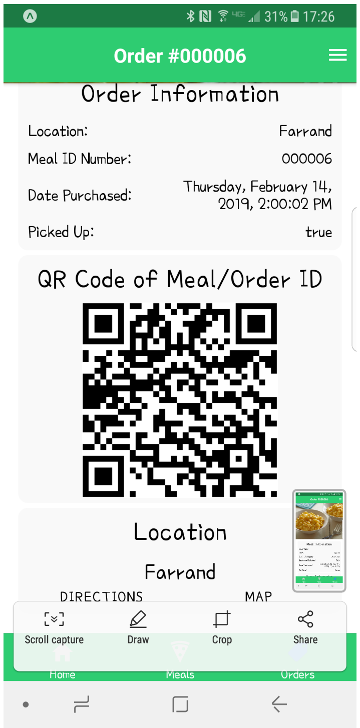
Cuisine Category: American

Estimated Calories: 300

Date Packaged: Thursday, February 14, 2019, 1:59:24 PM

For Sale: false

Figure A.15: Order screen that gives information about a particular order. Also, shows a QR code associated with the order's/meal's unique identification number. First of two screen shots.



Order #000006

Order Information

Location: Farrand

Meal ID Number: 000006

Date Purchased: Thursday, February 14, 2019, 2:00:02 PM

Picked Up: true

QR Code of Meal/Order ID

Location

Farrand

Figure A.16: Order screen that gives information about a particular order. Also, shows a QR code associated with the order's/meal's unique identification number. Second of two screen shots.

Appendix B

Packages Utilized in Consumer Mobile Application (package.json)

```
{
  "name": "fudlkr_consumer",
  "main": "node_modules/expo/AppEntry.js",
  "private": true,
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "eject": "expo eject"
  },
  "dependencies": {
    "dateformat": "^3.0.3",
    "email-validator": "^2.0.4",
    "firebase": "^5.7.0",
    "firebase-admin": "^6.3.0",
    "firebase-functions": "^2.1.0",
    "firebase-tools": "^6.1.2",
    "material-design-icons": "^3.0.1",
    "payment-icons": "^1.1.0",
    "react": "^16.6.1",
    "react-native": "^0.57.7",
    "react-native-awesome-card-io": "^0.8.1",
    "react-native-checkout": "0.0.8",
    "react-native-credit-card-input": "^0.4.1",
    "react-native-elements": "^0.19.1",
    "react-native-fbsdk": "^0.8.0",
    "react-native-firebase": "^5.1.1",
    "react-native-google-maps-directions": "^2.0.0",
    "react-native-keyboard-aware-scroll-view": "^0.7.4",
    "react-native-maps": "^0.22.1",
    "react-native-parallax-scrollview": "^3.0.0",
    "react-native-rate": "^1.1.6",
    "react-native-router-flux": "^4.0.5",
    "react-native-search-filter": "^0.1.4",
    "react-native-super-grid": "^2.4.3",
    "react-native-vector-icons": "^6.0.2",
    "react-navigation": "^3.0.9",
    "react-redux": "^5.1.0",
    "redux": "^4.0.1",
    "stripe": "^6.18.1",
    "toggle-switch-react-native": "^2.0.2",
    "yaqrcode": "^0.2.1"
  },
  "devDependencies": {
    "babel-eslint": "^10.0.1",
    "babel-preset-expo": "^5.0.0",
    "eslint": "^5.10.0",
    "eslint-plugin-react": "^7.11.1",
    "expo": "^31.0.6",
    "expo-gl": "^1.1.0",
    "schedule": "^0.4.0"
  },
  "rnpm": {
    "assets": [
      "./assets/fonts/"
    ]
  }
}
```

Appendix C

All Restaurant Mobile Application Screens

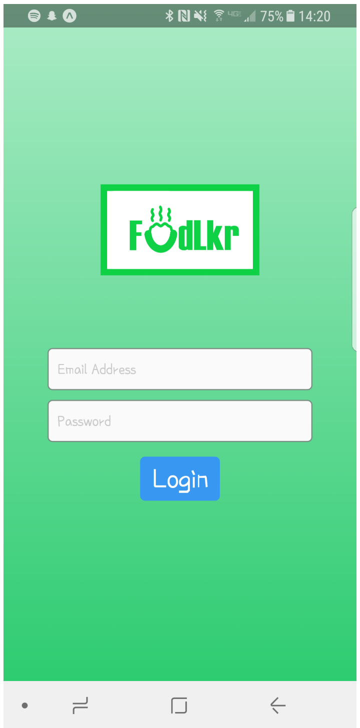


Figure C.1: Login screen, which allows users to log into their work account. Users do not have the option to signup. When a new organization partners with FoodLocker, the organization will provide a technical representative with a file containing all of the information associated with the employees of the organization. A successful login sends the user to the locations screen.

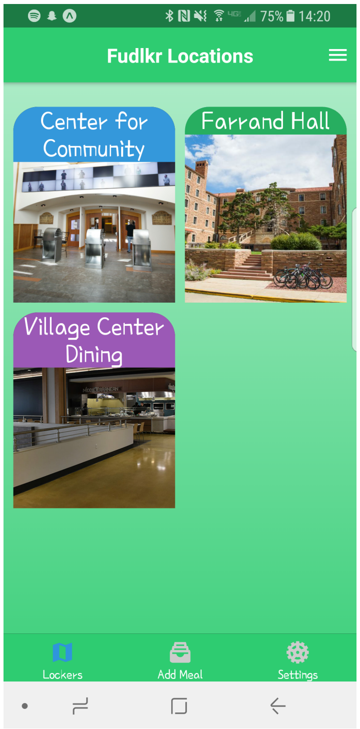
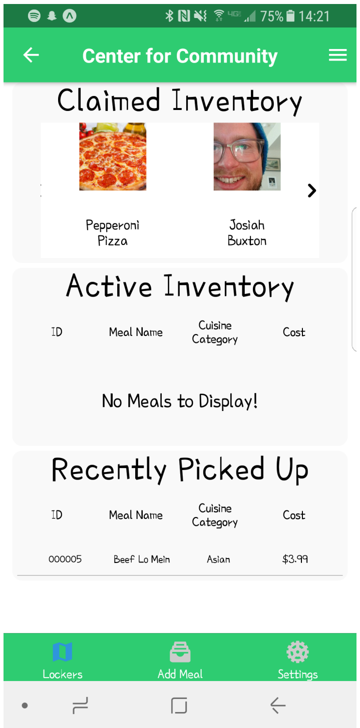


Figure C.2: Screen displaying all of the lockers associated with a particular organization. Allows the user to select a locker to bring them to a dashboard, which provides relevant information associated with the locker.



Claimed Inventory

Pepperoni Pizza

Josiah Buxton

Active Inventory

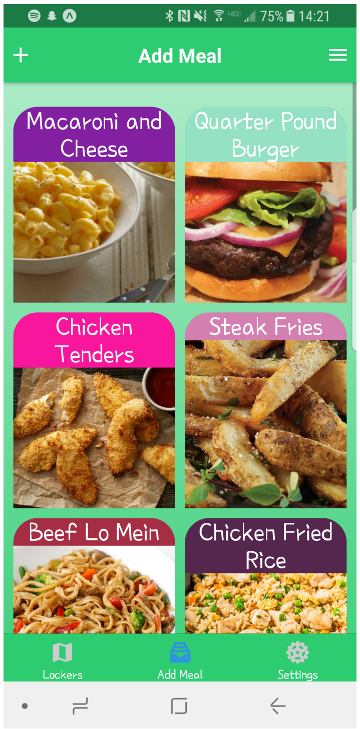
ID	Meal Name	Cuisine Category	Cost

No Meals to Display!

Recently Picked Up

ID	Meal Name	Cuisine Category	Cost
000005	Beef Lo Mein	Asian	$3.99

Figure C.3: Screen displaying three categories of information associated with a particular locker. Claimed Inventory contains meals that have been purchased by a consumer but have not yet been picked up from the locker. Active Inventory contains meals that are currently within the locker but have not been purchased. Recently picked up meals are meals that have been purchased and picked up by the consumer in the past 24 hours.



Macaroni and Cheese

Quarter Pound Burger

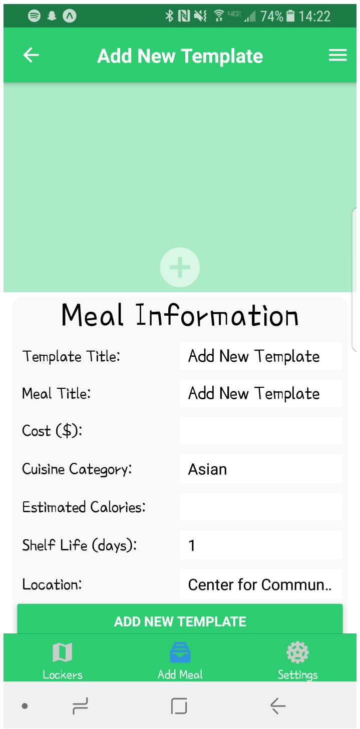
Chicken Tenders

Steak Fries

Beef Lo Mein

Chicken Fried Rice

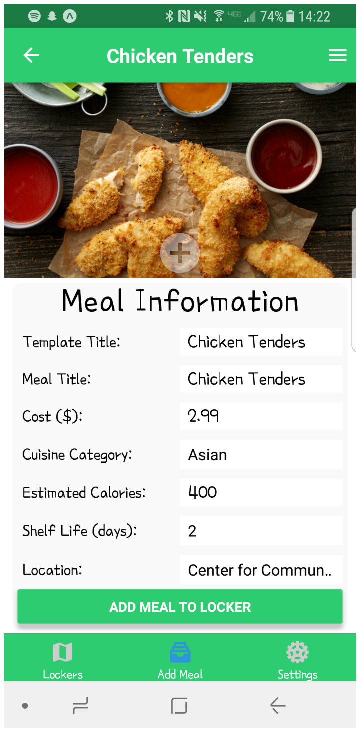
Figure C.4: Screen displaying all of the templates associated with an organization. The templates can be selected to quickly add common meals that are made and hosted by the organization. Selecting a template will send the user to the ”Add Meal” screen. Selecting the plus icon in the top left corner will send the user to the ”Add Meal” screen as well, but with no pre-populated information.



Meal Information

Template Title:	Add New Template
Meal Title:	Add New Template
Cost ($):	
Cuisine Category:	Asian
Estimated Calories:	
Shelf Life (days):	1
Location:	Center for Commun..

Figure C.5: Screen that allows users to add a brand new template and/or a new meal to the locker. After the user populates the information associated with the meal, which is being hosted, the user can press the "Add New Template" button at the bottom of the screen to save the template for future use. An algorithm checks to ensure the template is unique before saving the template to the database.



Meal Information

Template Title:	Chicken Tenders
Meal Title:	Chicken Tenders
Cost ($):	2.99
Cuisine Category:	Asian
Estimated Calories:	400
Shelf Life (days):	2
Location:	Center for Commun..

Figure C.6: Screen which allows a user to host a new meal and add the meal to the locker. After the user populates the information associated with the meal, the user can press the "Add meal to locker" button. An algorithm is run to ensure the locker unit, which the meal is potentially being added to, has space, and selects an empty individual locker where the meal can be placed.

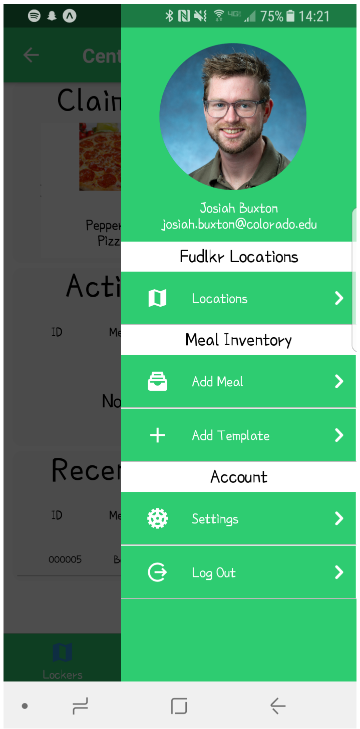
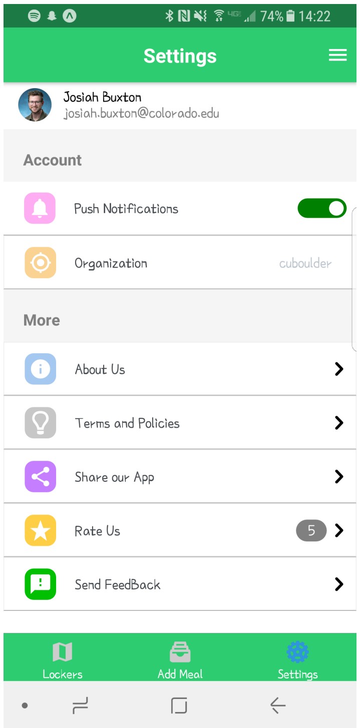


Figure C.7: Drawer overlay screen that enters the user's view when the upper right hand icon is touched (present on almost every screen). Lists different links that enable the user to navigate to most of the screens in the application.



Settings

Josiah Buxton
josiah.buxton@colorado.edu

Account

Push Notifications ●

Organization cuboulder

More

About Us

Terms and Policies

Share our App

Rate Us 5

Send FeedBack

Figure C.8: Screen which allows a user to view the settings associated with his/her personal account. Also allows the user to view information about FoodLocker or FoodLocker's application.

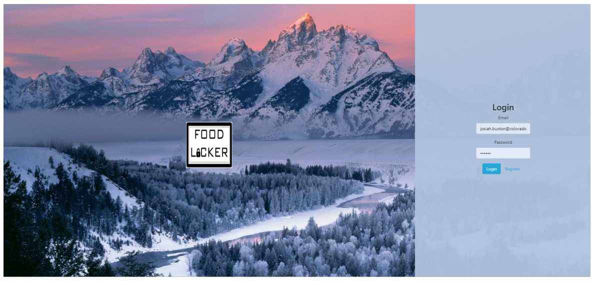
Appendix D

Packages Utilized in Restaurant Mobile Application (package.json)

```
{
  "main": "node_modules/expo/AppEntry.js",
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "eject": "expo eject"
  },
  "dependencies": {
    "dateformat": "^3.0.3",
    "email-validator": "^2.0.4",
    "expo": "^31.0.2",
    "firebase": "^5.7.0",
    "firebase-admin": "^6.3.0",
    "firebase-functions": "^2.1.0",
    "firebase-tools": "^6.1.2",
    "react": "16.5.0",
    "react-native": "https://github.com/expo/react-native/archive/
       sdk-31.0.0.tar.gz",
    "react-native-checkout": "0.0.8",
    "react-native-elements": "^0.19.1",
    "react-native-fbsdk": "^0.8.0",
    "react-native-keyboard-aware-scroll-view": "^0.8.0",
    "react-native-parallax-scrollview": "^3.0.0",
    "react-native-rate": "^1.1.6",
    "react-native-router-flux": "^4.0.6",
    "react-native-snap-carousel": "^3.7.5",
    "react-native-super-grid": "^2.4.3",
    "react-navigation": "^3.0.9",
    "toggle-switch-react-native": "^2.0.2",
    "yaqrcode": "^0.2.1"
  },
  "devDependencies": {
    "babel-preset-expo": "^5.0.0"
  },
  "private": true
}
```

Appendix E

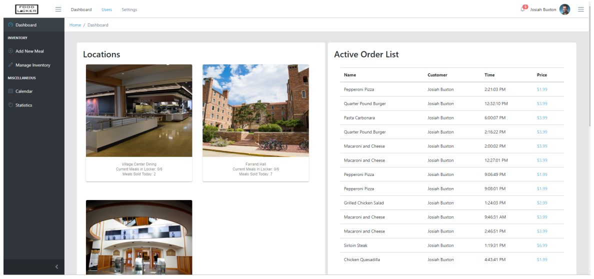
All Restaurant Web Application Screens



Email: josiah.buxton@colorado

Password: ••••••

Figure E.1: Login screen, which allows users to log into their work account. Users do not have the option to signup. When a new organization partners with FoodLocker, the organization will provide a technical representative with a file containing all of the information associated with the employees of the organization. A successful login sends the user to the dashboard screen.



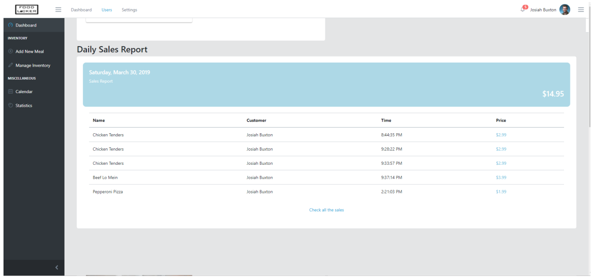
Village Center Dining
Current Meals in Locker: 0/6
Meals Sold Today: 2

Farrand Hall
Current Meals in Locker: 0/6
Meals Sold Today: 7

Active Order List

Name	Customer	Time	Price
Pepperoni Pizza	Josiah Buxton	2:21:03 PM	$1.99
Quarter Pound Burger	Josiah Buxton	12:32:10 PM	$3.99
Pasta Carbonara	Josiah Buxton	6:00:07 PM	$3.99
Quarter Pound Burger	Josiah Buxton	2:16:22 PM	$3.99
Macaroni and Cheese	Josiah Buxton	2:00:02 PM	$3.99
Macaroni and Cheese	Josiah Buxton	12:27:01 PM	$3.99
Pepperoni Pizza	Josiah Buxton	9:06:49 PM	$1.99
Pepperoni Pizza	Josiah Buxton	9:08:01 PM	$1.99
Grilled Chicken Salad	Josiah Buxton	1:24:03 PM	$2.99
Macaroni and Cheese	Josiah Buxton	9:46:31 AM	$3.99
Macaroni and Cheese	Josiah Buxton	2:46:51 PM	$3.99
Sirloin Steak	Josiah Buxton	1:19:31 PM	$6.99
Chicken Quesadilla	Josiah Buxton	4:43:41 PM	$1.99

Figure E.2: Screen displaying pertinent information associated with the lockers owned by the organization. The "Locations" section displays cards associated with each individual locker. The cards show the amount of meals sold, meals currently in the locker, locker meal capacity, and the name of the locker. The "Active Order List" shows all of the recent meals that have been purchased. Scrolling down the screen, displays more information laid out in the next screenshot.

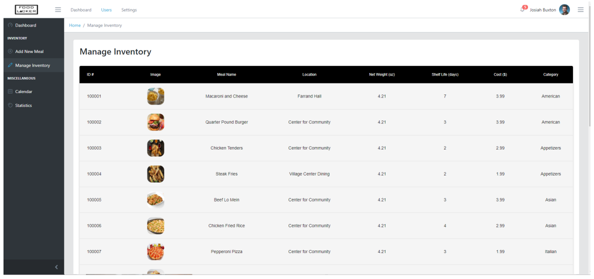


Daily Sales Report

Saturday, March 30, 2019
Sales Report
$14.95

Name	Customer	Time	Price
Chicken Tenders	Josiah Buxton	8:44:35 PM	$2.99
Chicken Tenders	Josiah Buxton	9:28:22 PM	$2.99
Chicken Tenders	Josiah Buxton	9:33:57 PM	$2.99
Beef Lo Mein	Josiah Buxton	9:37:14 PM	$3.99
Pepperoni Pizza	Josiah Buxton	2:21:03 PM	$1.99

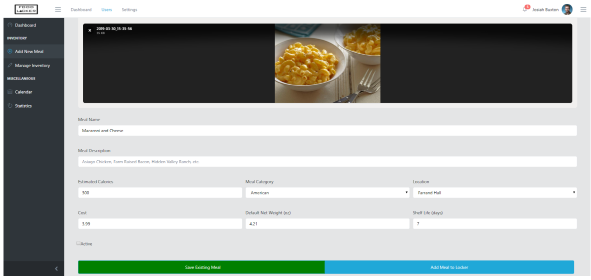
Figure E.3: Continuation of the Dashboard screen. The ”Daily Sales Report” displays all of the completed transactions in the last 24 hours.



Manage Inventory

ID #	Image	Meal Name	Location	Net Weight (oz)	Shelf Life (days)	Cost ($)	Category
100001		Macaroni and Cheese	Farrand Hall	4.21	7	3.99	American
100002		Quarter Pound Burger	Center for Community	4.21	3	3.99	American
100003		Chicken Tenders	Center for Community	4.21	2	2.99	Appetizers
100004		Steak Fries	Village Center Dining	4.21	2	1.99	Appetizers
100005		Beef Lo Mein	Center for Community	4.21	3	3.99	Asian
100006		Chicken Fried Rice	Center for Community	4.21	4	2.99	Asian
100007		Pepperoni Pizza	Center for Community	4.21	3	1.99	Italian

Figure E.4: Screen displaying all of the templates associated with an organization. The templates allow restaurant employees to quickly host different meals to the online marketplace. Selecting a meal sends the user to the ”Add Meal” screen.



Meal Name: Macaroni and Cheese

Meal Description: Asiago Chicken, Farm Raised Bacon, Hidden Valley Ranch, etc.

Estimated Calories: 300

Meal Category: American

Location: Farrand Hall

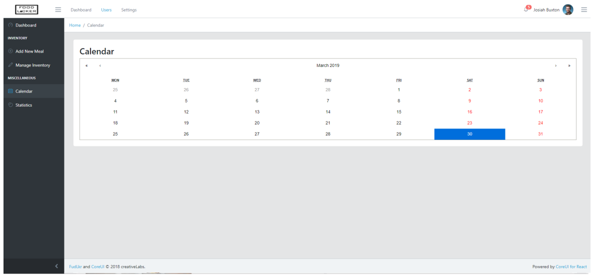
Cost: 3.99

Default Net Weight (oz): 4.21

Shelf Life (days): 7

□ Active

Figure E.5: Screen displaying information associated with a meal that is about to be added to a locker. The user can change information in the fields to correctly describe the meal they are trying to host on the online marketplace. Users can either save the template or add the meal to the locker. Both of the functions are identical to the functions implemented in the restaurant mobile application.

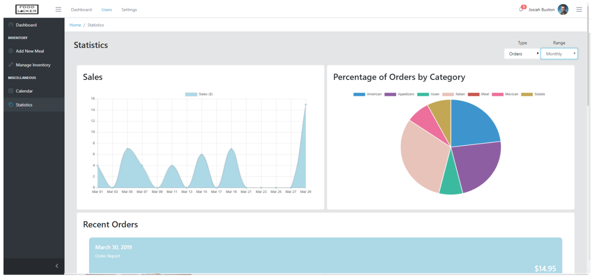


Calendar

March 2019

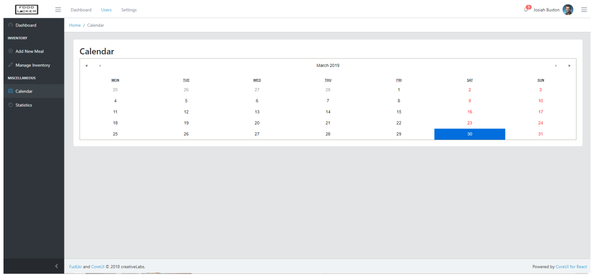
MON	TUE	WED	THU	FRI	SAT	SUN
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure E.6: Screen displaying a calendar. Currently under development...



March 30, 2019
Order Report
$14.95

Figure E.7: Screen displaying statistics associated with meals sold by the organization through the FoodLocker platform. The first graph shows the total sales for a given time period. The bar graph shows the percentage of meals sold that belong to a particular meal category for a given time period. The time period can be changed in the upper right hand corner and the graphs are dynamically updated in real-time.



MON	TUE	WED	THU	FRI	SAT	SUN
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

FudLkr and CoreUI © 2018 creativeLabs.

Powered by CoreUI for React

Figure E.8: Continuation of the statistics screen. Shows a list of the highest frequented users. Shows the total meals purchased by each user in the list. Changing the time period at the top of the page, filters the highest frequented users list.

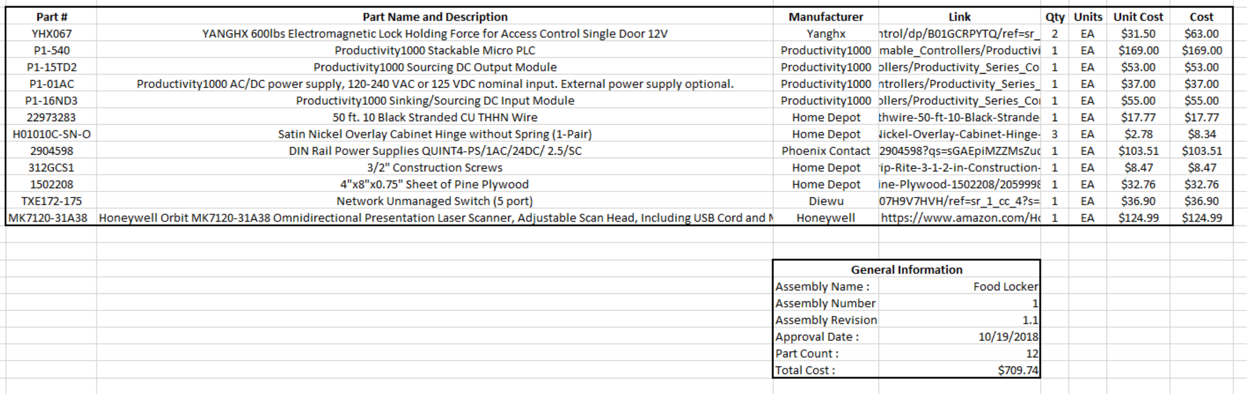
Appendix F

Packages Utilized in Restaurant Web Application (package.json)

```
{
  "name": "fudlkr",
  "version": "2.1.3",
  "description": "Fudlkr Restaurant Web Application",
  "author": "Josiah Buxton and ukasz Holeczek",
  "homepage": "http://fudlkr.com",
  "copyright": "Copyright 2018 creativeLabs ukasz Holeczek",
  "license": "MIT",
  "private": true,
  "repository": {
    "type": "git",
    "url": "git@github.com:coreui/coreui-free-react-admin-template.
       git"
  },
  "dependencies": {
    "@coreui/coreui": "^2.1.5",
    "@coreui/coreui-plugin-chartjs-custom-tooltips": "^1.2.0",
    "@coreui/icons": "0.3.0",
    "@coreui/react": "^2.1.3",
    "@material-ui/core": "^3.9.2",
    "bootstrap": "^4.2.1",
    "chart.js": "^2.7.3",
    "classnames": "^2.2.6",
    "core-js": "^2.6.1",
    "dateformat": "^3.0.3",
    "enzyme": "^3.8.0",
    "enzyme-adapter-react-16": "^1.7.1",
    "filepond": "^4.2.0",
    "filepond-plugin-file-validate-size": "^2.1.1",
    "filepond-plugin-file-validate-type": "^1.2.2",
    "filepond-plugin-image-exif-orientation": "^1.0.4",
    "filepond-plugin-image-preview": "^4.0.3",
    "firebase": "^5.8.2",
    "flag-icon-css": "^3.2.1",
    "font-awesome": "^4.7.0",
    "lodash": "^4.17.11",
    "node-sass": "^4.11.0",
    "prop-types": "^15.6.2",
    "react": "^16.7.0",
    "react-app-polyfill": "^0.2.0",
    "react-calendar": "^2.18.1",
    "react-chartjs-2": "^2.7.4",
    "react-datepicker": "^2.1.0",
    "react-dom": "^16.7.0",
    "react-filepond": "^7.0.1",
    "react-grid-layout": "^0.16.6",
    "react-loadable": "^5.5.0",
    "react-modal": "^3.8.1",
    "react-redux": "^6.0.0",
    "react-router-config": "^4.4.0-beta.6",
    "react-router-dom": "^4.3.1",
    "react-test-renderer": "^16.7.0",
    "reactstrap": "^7.0.2",
    "redux": "^4.0.1",
    "redux-logger": "^3.0.6",
    "redux-persist": "^5.10.0",
    "redux-thunk": "^2.3.0",
    "simple-line-icons": "^2.4.1"
  },
  "devDependencies": {
    "react-scripts": "2.1.3"
  },
  "scripts": {
    "start": "HOST=127.0.0.1 && react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "test:cov": "react-scripts test --coverage",
    "test:debug": "react-scripts --inspect-brk test --runInBand",
    "eject": "react-scripts eject"
  },
  "bugs": {
    "url": "https://github.com/coreui/coreui-free-react-admin-
       template/issues"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": [
    ">0.2%",
    "not dead",
    "not ie <= 9",
    "not op_mini all"
  ],
  "jest": {
    "collectCoverageFrom": [
      "src/**/*.{js,jsx}",
      "!**/*index.js",
      "!src/serviceWorker.js",
      "!src/polyfill.js"
    ]
  }
}
```

Appendix G

Locker Bill of Materials



Part #	Part Name and Description	Manufacturer	Link	Qty	Units	Unit Cost	Cost
YHX067	YANGHX 600lbs Electromagnetic Lock Holding Force for Access Control Single Door 12V	Yanghx	ntrol/dp/B01GCRPYTQ/ref=sr_	2	EA	$31.50	$63.00
P1-540	Productivity1000 Stackable Micro PLC	Productivity1000	mable_Controllers/Productivi	1	EA	$169.00	$169.00
P1-15TD2	Productivity1000 Sourcing DC Output Module	Productivity1000	ollers/Productivity_Series_Co	1	EA	$53.00	$53.00
P1-01AC	Productivity1000 AC/DC power supply, 120-240 VAC or 125 VDC nominal input. External power supply optional.	Productivity1000	ntrollers/Productivity_Series_	1	EA	$37.00	$37.00
P1-16ND3	Productivity1000 Sinking/Sourcing DC Input Module	Productivity1000	ollers/Productivity_Series_Co	1	EA	$55.00	$55.00
22973283	50 ft. 10 Black Stranded CU THHN Wire	Home Depot	thwire-50-ft-10-Black-Strande	1	EA	$17.77	$17.77
H01010C-SN-O	Satin Nickel Overlay Cabinet Hinge without Spring (1-Pair)	Home Depot	Nickel-Overlay-Cabinet-Hinge-	3	EA	$2.78	$8.34
2904598	DIN Rail Power Supplies QUINT4-PS/1AC/24DC/ 2.5/SC	Phoenix Contact	2904598?qs=sGAEpiMZZMsZuc	1	EA	$103.51	$103.51
312GCS1	3/2" Construction Screws	Home Depot	rip-Rite-3-1-2-in-Construction-	1	EA	$8.47	$8.47
1502208	4"x8"x0.75" Sheet of Pine Plywood	Home Depot	ine-Plywood-1502208/205999	1	EA	$32.76	$32.76
TXE172-175	Network Unmanaged Switch (5 port)	Diewu	07H9V7HVH/ref=sr_1_cc_4?s=	1	EA	$36.90	$36.90
MK7120-31A38	Honeywell Orbit MK7120-31A38 Omnidirectional Presentation Laser Scanner, Adjustable Scan Head, Including USB Cord and M	Honeywell	https://www.amazon.com/Ho	1	EA	$124.99	$124.99

General Information	
Assembly Name :	Food Locker
Assembly Number	1
Assembly Revision	1.1
Approval Date :	10/19/2018
Part Count :	12
Total Cost :	$709.74

Figure G.1: Bill of Materials for the built Food Locker prototype. All components were purchased on a budget of $600. $300 was received from ”Get Seed Funding” on CU Boulder’s campus and the other $300 was contributed by family and friends. The last $110 was contributed by myself.

Appendix H

Initial 3D Model of Food Locker and Finished Prototyped Food Locker

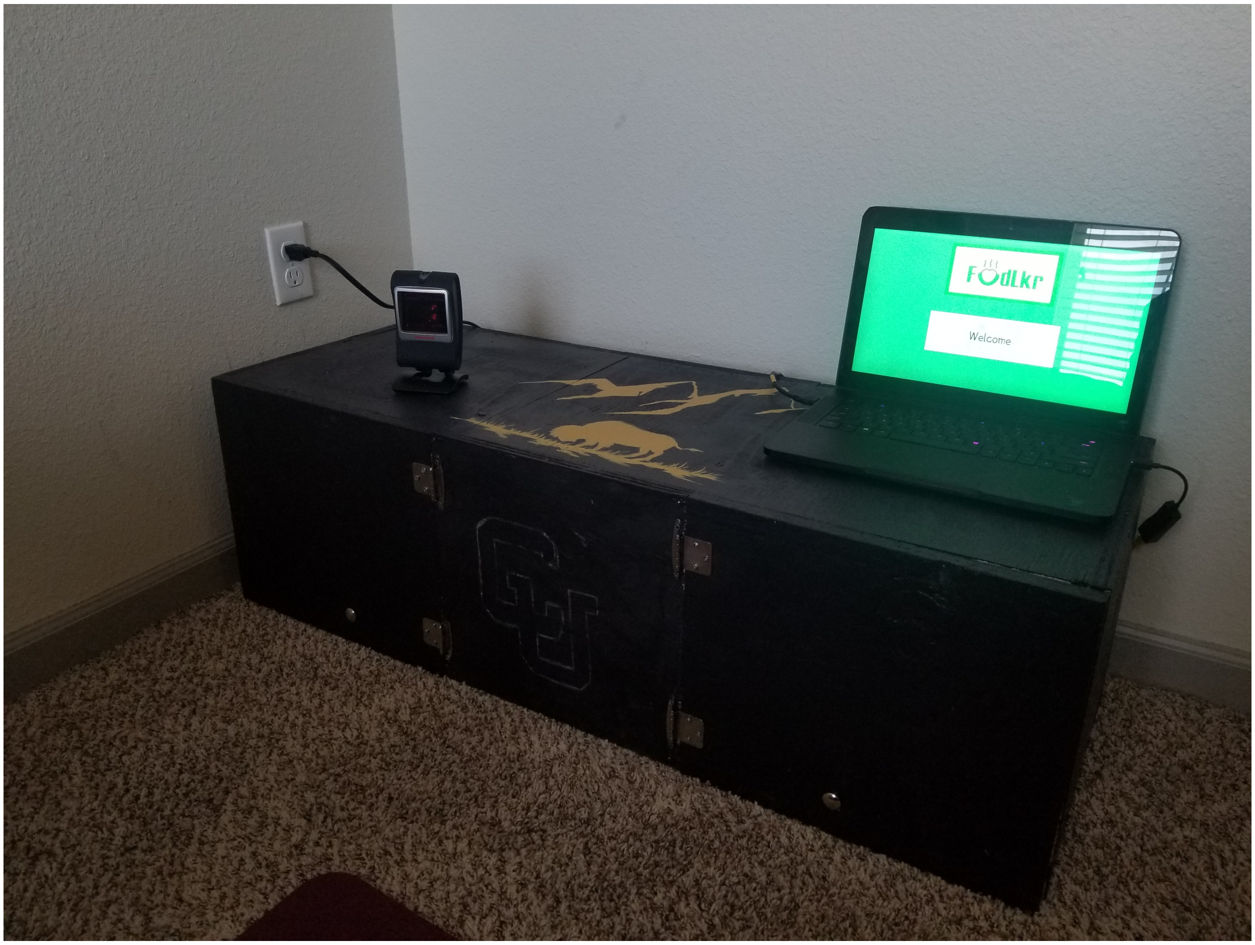


Figure H.1: Finished prototype of the food locker.

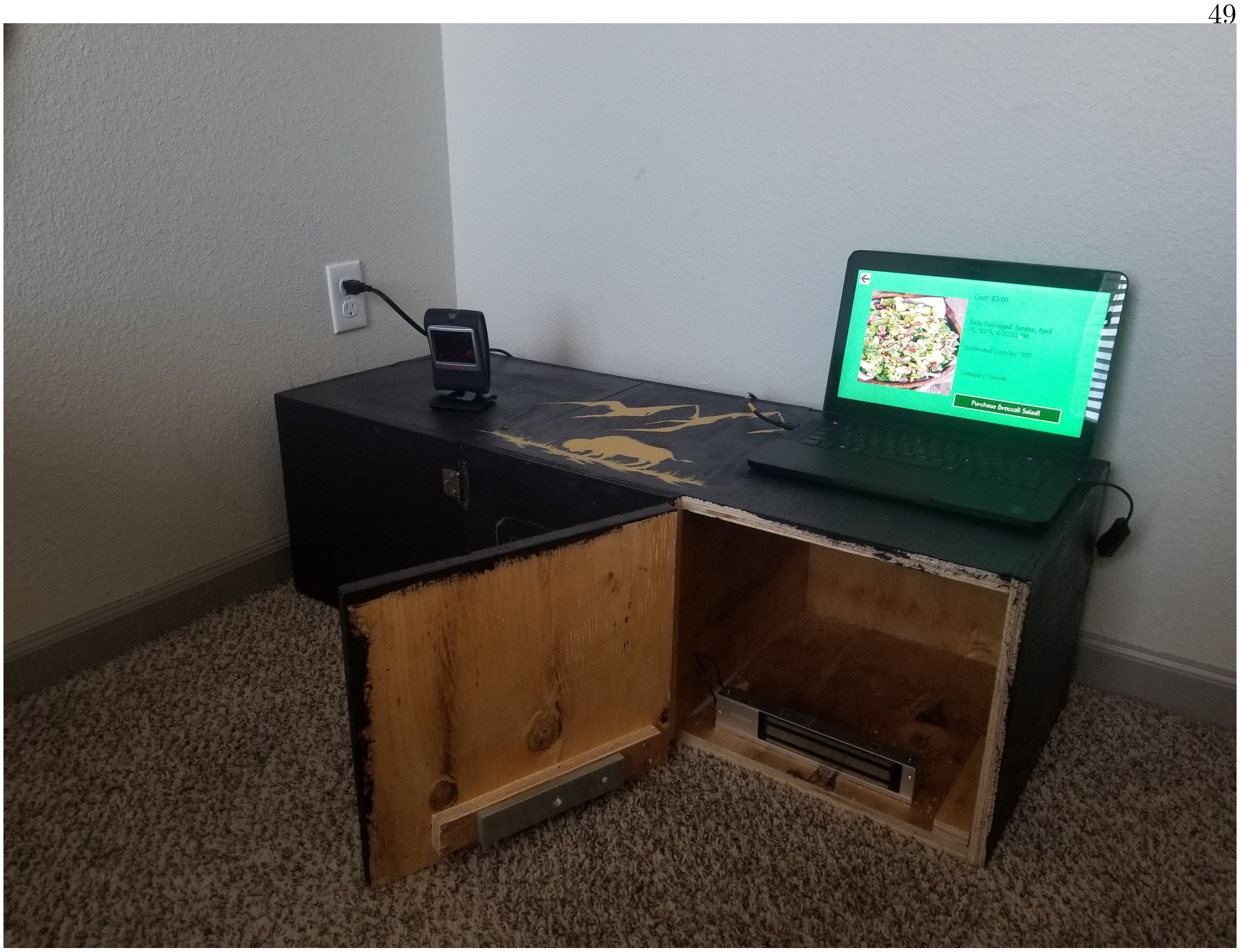


Figure H.2: Finished prototype of the food locker.

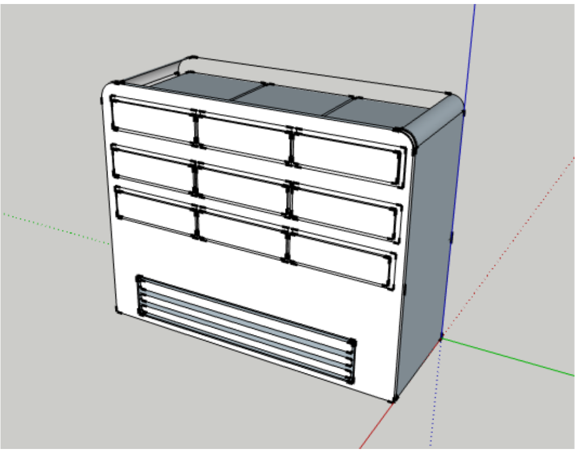


Figure H.3: 3D model of a food locker. The model was built using Google Sketchup. Front angle.

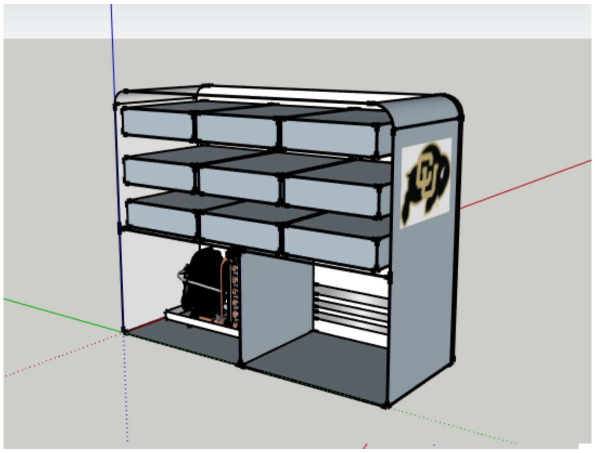


Figure H.4: 3D model of a food locker. The model was built using Google Sketchup. Back angle 1.

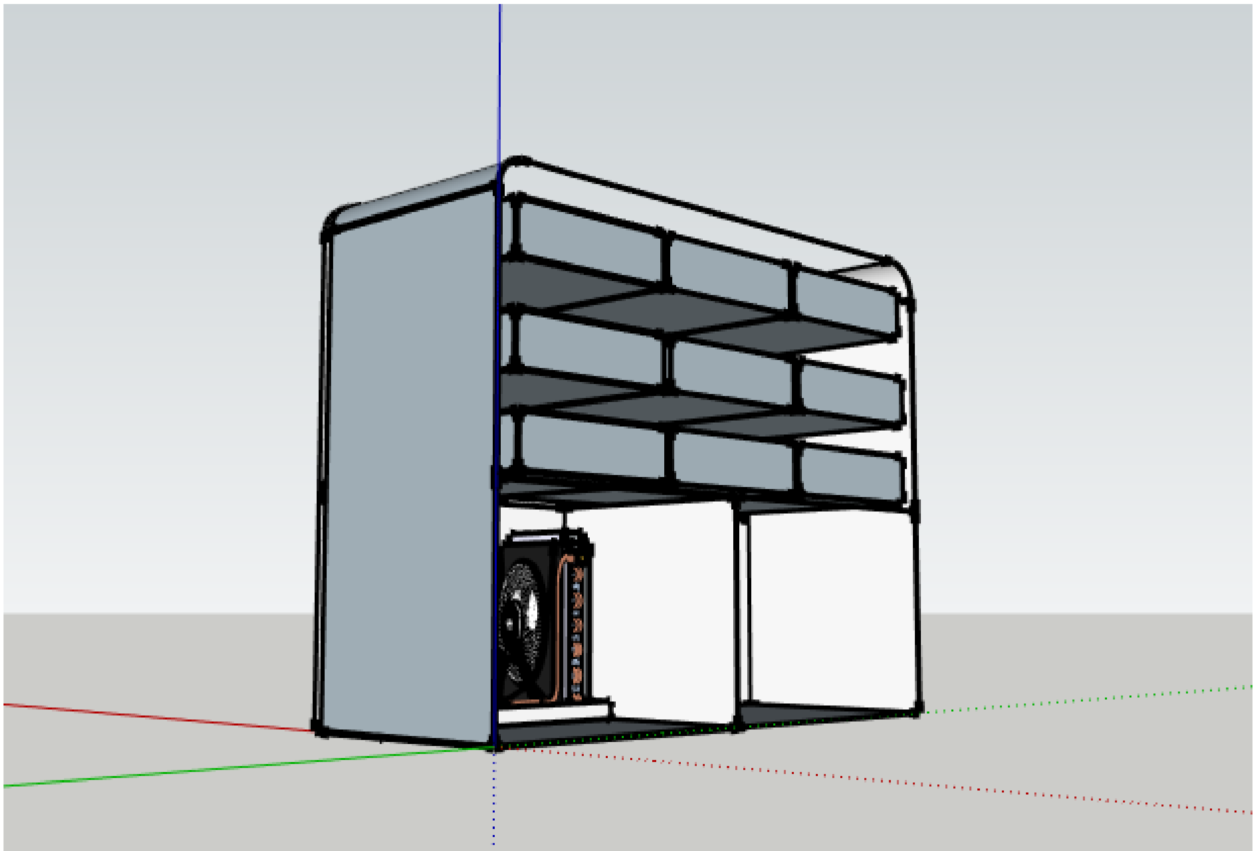
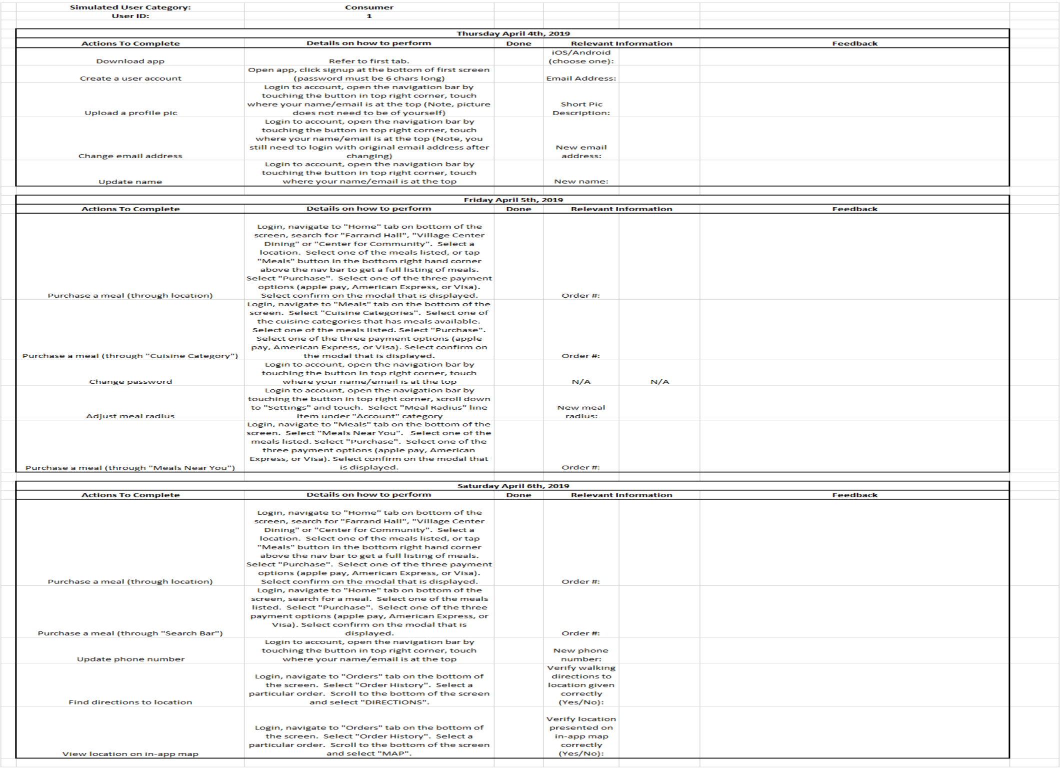


Figure H.5: 3D model of a food locker. The model was built using Google Sketchup. Back angle 2.

Appendix I

Consumer/Restaurant Data Input Templates



Simulated User Category:	Consumer
User ID:	1

Thursday April 4th, 2019

Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Download app	Refer to first tab.		iOS/Android (choose one):		
Create a user account	Open app, click signup at the bottom of first screen (password must be 6 chars long)		Email Address:		
Upload a profile pic	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top (Note, picture does not need to be of yourself)		Short Pic Description:		
Change email address	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top (Note, you still need to login with original email address after changing)		New email address:		
Update name	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top		New name:		

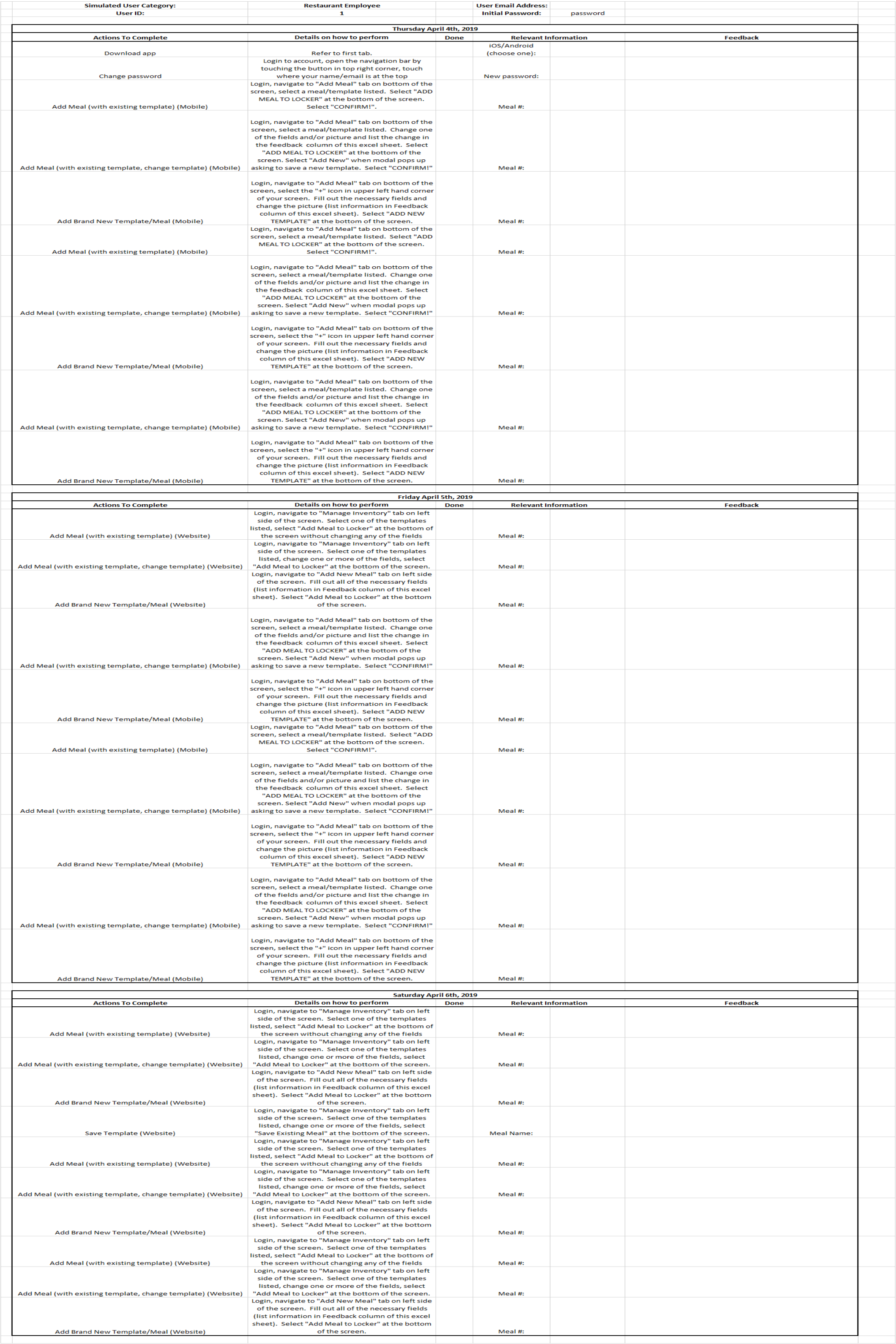
Friday April 5th, 2019

Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Purchase a meal (through location)	Login, navigate to "Home" tab on bottom of the screen, search for "Farrand Hall", "Village Center Dining" or "Center for Community". Select a location. Select one of the meals listed, or tap "Meals" button in the bottom right hand corner above the nav bar to get a full listing of meals. Select "Purchase". Select one of the three payment options (apple pay, American Express, or Visa). Select confirm on the modal that is displayed.		Order #:		
Purchase a meal (through "Cuisine Category")	Login, navigate to "Meals" tab on the bottom of the screen. Select "Cuisine Categories". Select one of the cuisine categories that has meals available. Select one of the meals listed. Select "Purchase". Select one of the three payment options (apple pay, American Express, or Visa). Select confirm on the modal that is displayed.		Order #:		
Change password	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top		N/A	N/A	
Adjust meal radius	Login to account, open the navigation bar by touching the button in top right corner, scroll down to "Settings" and touch. Select "Meal Radius" line item under "Account" category		New meal radius:		
Purchase a meal (through "Meals Near You")	Login, navigate to "Meals" tab on the bottom of the screen. Select "Meals Near You". Select one of the meals listed. Select "Purchase". Select one of the three payment options (apple pay, American Express, or Visa). Select confirm on the modal that is displayed.		Order #:		

Saturday April 6th, 2019

Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Purchase a meal (through location)	Login, navigate to "Home" tab on bottom of the screen, search for "Farrand Hall", "Village Center Dining" or "Center for Community". Select a location. Select one of the meals listed, or tap "Meals" button in the bottom right hand corner above the nav bar to get a full listing of meals. Select "Purchase". Select one of the three payment options (apple pay, American Express, or Visa). Select confirm on the modal that is displayed.		Order #:		
Purchase a meal (through "Search Bar")	Login, navigate to "Home" tab on bottom of the screen, search for a meal. Select one of the meals listed. Select "Purchase". Select one of the three payment options (apple pay, American Express, or Visa). Select confirm on the modal that is displayed.		Order #:		
Update phone number	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top		New phone number:		
Find directions to location	Login, navigate to "Orders" tab on the bottom of the screen. Select "Order History". Select a particular order. Scroll to the bottom of the screen and select "DIRECTIONS".		Verify walking directions to location given correctly (Yes/No):		
View location on in-app map	Login, navigate to "Orders" tab on the bottom of the screen. Select "Order History". Select a particular order. Scroll to the bottom of the screen and select "MAP".		Verify location presented on in-app map correctly (Yes/No):		

Figure I.1: This is the Google Sheet template that was sent out to each of the participants in the experiment who were apart of the "Consumer" category. Participants were asked to perform each of the functions listed in the first column, and provide feedback in the last three columns. The "Relevant Information" column was cross referenced with values in the database to verify the correctness of functions.

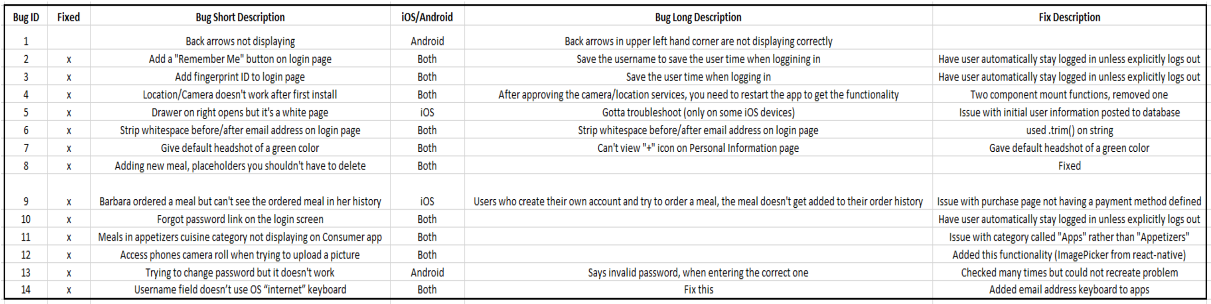


Simulated User Category:	Restaurant Employee		User Email Address:		
User ID:	1		Initial Password:	password	
Thursday April 4th, 2019					
Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Download app	Refer to first tab.		iOS/Android (choose one):		
Change password	Login to account, open the navigation bar by touching the button in top right corner, touch where your name/email is at the top		New password:		
Add Meal (with existing template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "CONFIRM!".		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Add Meal (with existing template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "CONFIRM!".		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Friday April 5th, 2019					
Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Add Meal (with existing template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, select "Add Meal to Locker" at the bottom of the screen without changing any of the fields		Meal #:		
Add Meal (with existing template, change template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, change one or more of the fields, select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Brand New Template/Meal (Website)	Login, navigate to "Add New Meal" tab on left side of the screen. Fill out all of the necessary fields (list information in Feedback column of this excel sheet). Select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Add Meal (with existing template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "CONFIRM!".		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Add Meal (with existing template, change template) (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select a meal/template listed. Change one of the fields and/or picture and list the change in the feedback column of this excel sheet. Select "ADD MEAL TO LOCKER" at the bottom of the screen. Select "Add New" when modal pops up asking to save a new template. Select "CONFIRM!"		Meal #:		
Add Brand New Template/Meal (Mobile)	Login, navigate to "Add Meal" tab on bottom of the screen, select the "+" icon in upper left hand corner of your screen. Fill out the necessary fields and change the picture (list information in Feedback column of this excel sheet). Select "ADD NEW TEMPLATE" at the bottom of the screen.		Meal #:		
Saturday April 6th, 2019					
Actions To Complete	Details on how to perform	Done	Relevant Information		Feedback
Add Meal (with existing template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, select "Add Meal to Locker" at the bottom of the screen without changing any of the fields		Meal #:		
Add Meal (with existing template, change template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, change one or more of the fields, select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Brand New Template/Meal (Website)	Login, navigate to "Add New Meal" tab on left side of the screen. Fill out all of the necessary fields (list information in Feedback column of this excel sheet). Select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Save Template (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, change one or more of the fields, select "Save Existing Meal" at the bottom of the screen.		Meal Name:		
Add Meal (with existing template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, select "Add Meal to Locker" at the bottom of the screen without changing any of the fields		Meal #:		
Add Meal (with existing template, change template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, change one or more of the fields, select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Brand New Template/Meal (Website)	Login, navigate to "Add New Meal" tab on left side of the screen. Fill out all of the necessary fields (list information in Feedback column of this excel sheet). Select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Meal (with existing template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, select "Add Meal to Locker" at the bottom of the screen without changing any of the fields		Meal #:		
Add Meal (with existing template, change template) (Website)	Login, navigate to "Manage Inventory" tab on left side of the screen. Select one of the templates listed, change one or more of the fields, select "Add Meal to Locker" at the bottom of the screen.		Meal #:		
Add Brand New Template/Meal (Website)	Login, navigate to "Add New Meal" tab on left side of the screen. Fill out all of the necessary fields (list information in Feedback column of this excel sheet). Select "Add Meal to Locker" at the bottom of the screen.		Meal #:		

Figure I.2: This is the Google Sheet template that was sent out to each of the participants in the experiment who were apart of the ”Restaurant Employee” category. Participants were asked to perform each of the functions listed in the first column, and provide feedback in the last three columns. The ”Relevant Information” column was cross referenced with values in the database to verify the correctness of functions.

Appendix J

List of Issues Gathered from Experiment



Bug ID	Fixed	Bug Short Description	iOS/Android	Bug Long Description	Fix Description
1		Back arrows not displaying	Android	Back arrows in upper left hand corner are not displaying correctly	
2	x	Add a "Remember Me" button on login page	Both	Save the username to save the user time when loggining in	Have user automatically stay logged in unless explicitly logs out
3	x	Add fingerprint ID to login page	Both	Save the user time when logging in	Have user automatically stay logged in unless explicitly logs out
4	x	Location/Camera doesn't work after first install	Both	After approving the camera/location services, you need to restart the app to get the functionality	Two component mount functions, removed one
5	x	Drawer on right opens but it's a white page	iOS	Gotta troubleshoot (only on some iOS devices)	Issue with initial user information posted to database
6	x	Strip whitespace before/after email address on login page	Both	Strip whitespace before/after email address on login page	used .trim() on string
7	x	Give default headshot of a green color	Both	Can't view "+" icon on Personal Information page	Gave default headshot of a green color
8	x	Adding new meal, placeholders you shouldn't have to delete	Both		Fixed
9	x	Barbara ordered a meal but can't see the ordered meal in her history	iOS	Users who create their own account and try to order a meal, the meal doesn't get added to their order history	Issue with purchase page not having a payment method defined
10	x	Forgot password link on the login screen	Both		Have user automatically stay logged in unless explicitly logs out
11	x	Meals in appetizers cuisine category not displaying on Consumer app	Both		Issue with category called "Apps" rather than "Appetizers"
12	x	Access phones camera roll when trying to upload a picture	Both		Added this functionality (ImagePicker from react-native)
13	x	Trying to change password but it doesn't work	Android	Says invalid password, when entering the correct one	Checked many times but could not recreate problem
14	x	Username field doesn’t use OS “internet” keyboard	Both	Fix this	Added email address keyboard to apps

Figure J.1: This is a table of compiled issues from the restaurant and consumer mobile applications. These issues were illuminated during the experiment and were addressed prior to finishing this thesis.