

GoCart

by

Zachary Basford, Jacob Mastroianni, and Casey Wroblewski

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2020 GoCart

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Zachary Basford, Jacob Mastroianni, and Casey Wroblewski 4/27/2020
Date

Abdou Fall 4/27/2020
Faculty Advisor Date

University of Cincinnati
College Of
Education, Criminal Justice, and Human Services

April 2020

TABLE OF CONTENTS

TABLE OF CONTENTS	I
LIST OF FIGURES	II
ABSTRACT	1
INTRODUCTION	2
PROBLEM STATEMENT	2
SOLUTION	2
PROJECT GOALS	3
OVERVIEW	3
DISCUSSION	4
PROJECT CONCEPT/SOLUTION	4
DESIGN OBJECTIVES	4
METHODOLOGY/TECHNICAL APPROACH	4
USER PROFILE	5
USE CASE DIAGRAM	7
BUDGET	8
GANTT CHART	8
PROBLEMS ENCOUNTERED AND ANALYSIS OF PROBLEMS SOLVED	10
FUTURE RECOMMENDATION FOR IMPROVEMENTS	10
TESTING	10
OVERVIEW/METHODOLOGY	10
SCOPE OF TESTING	11
OBJECTIVES	11
LOGGING TEST AND PROCEDURES	12
WHAT WE LEARNED DURING TESTING	14
TECHNICAL ELEMENTS	14
OVERVIEW	14
TECHNICAL ARCHITECTURE DIAGRAM	15
APPLICATION	15
DATABASE	15
USER INTERFACE SCREENS	16
CONCLUSION	20
BACK MATTER	21
REFERENCES	21
APPENDIXES	22

LIST OF FIGURES

FIGURE 1: USER PROFILE (SHOPPER)	5
FIGURE 2: USER PROFILE (ADMIN)	6
FIGURE 3: USE CASE DIAGRAM	7
TABLE 1: BUDGET TABLE	8
FIGURE 4: GANTT CHART	9
TABLE 2: LAUNCH TEST	12
TABLE 3: STABILITY TEST	13
TABLE 4: FUNCTIONALITY TEST	14
TABLE 5: USER ACCEPTANCE TEST.....	14
FIGURE 5: TECHNICAL ARCH DIAGRAM.....	15
FIGURE 6: LOGIN SCREEN.....	16
FIGURE 7: CREATE ACCOUNT SCREEN.....	16
FIGURE 8: STORE SELECTION SCREEN.....	17
FIGURE 9: INVENTORY SCREEN	17
FIGURE 10: MY LIST SCREEN	18
FIGURE 11: STORE MAP SCREEN	19
FIGURE 12: CODE FOR CATEGORY QUERIES	22
FIGURE 13: GOCART IT EXPO POSTER	23

Abstract

The use of the GoCart application will have the ability to enhance grocery shoppers' experience by taking the shopping list and generating the most efficient shopping path. According to the Time Use Institute, the average shopping trip takes 45 minutes. In part, the time spent is attributed to confusing store layout, congested aisles, and uninformed staff (Marks, 2015). Results from our grocery shopping experience has shown that more time is directed towards roaming through the aisles searching for the items. There are applications dealing with allowing us to create a virtual list, but none directed towards the location of the items. Along with GoCart creating the most productive shopping path with the virtual list, there will also be a map with the store layout that pinpoints each item. Having GoCart will make the next trip to the grocery store quick and painless.

Introduction

Problem Statement

According to the Time Use Institute, the average shopping trip takes 41 minutes. In part, the time spent is attributed to confusing store layout, congested aisles, and uninformed staff.

(Marks, 2015)

Grocery stores are becoming larger and their inventories are increasing making it harder for newer customer to efficiently shop. Customers do not often have the time to search through aisles for the item they want, and often grow frustrated when the labeled sections do not accurately represent the items they hold. Another issue arises from the general layout of the store, markets are not always laid out in a way that makes it easy for customers. Customers are not able to follow their shopping list, from one item to the other, in the most optimal path.

Solution

GoCart is an Android application that allows shoppers to connect their shopping list with the best route for their items in a timely manner. The items will be generated on a map, pinpointing all the items on the list that are available. The items targeted will be created by the user with keywords associated to that specific item requested. Then, a separate page will appear with a list of the items generated along with their availability.

GoCart is a grocery store application like other applications out on the market now that help the user organize their items and reduce the stress of shopping. Many other similar applications created focus on allowing the user to manage their grocery lists to help him/her to be organized and reduce their time spent at the market. Grocery Pal is just one of the applications provided through iOS and Android platforms. It allows shoppers to create their own grocery list and find the best possible deals and coupons offered at local supermarkets. GoCart is taking the same

organizational method and expanding it with not only the list of items, but a route to reduce time spent shopping.

The focus of GoCart is towards a local grocery store, The Clifton Markets, located near The University of Cincinnati. The Web application for this grocery store was unorganized and hard to navigate around. There was a feature that allowed the user to create a list on a notepad but did not show the inventory or availability of the item. GoCart would be able to target the audience that is unfamiliar with the market and items it contains. Once a user has entered in the items from the store, a map layout of the store will be generated with drop pins of the items along with the best route. The app will estimate the amount of time it will take to get all the selected groceries in the store and checkout. This will be done by using a time estimate from the number of items in your list. The focus of this app is to make sure that users can have a list and route of their items that they are trying to acquire to shorten time and limit confusion in the store.

Project Goals

The goal of developing GoCart is to provide a way for grocery shoppers to save time and effort when traveling to the store. It will take the shoppers' list and transform that list into another list that will be listed in the most efficient path along with the store map layout with pinpoint locations of those items as well.

Overview

The remaining sections of this report describe how GoCart was completed from start to finish. Within this report, the following sections are detailed: design logic, testing methodology, timeline, budget, lessons learned, and problems encountered.

Discussion

Project Concept/Solution

We came up with the project idea based on our experiences in grocery stores and knowing how difficult it can be to find items on your grocery list if your unfamiliar with the store layout that you are shopping at. Our main purpose behind this application is to alleviate the stress of locating items within the store by creating a shopping list and generating a path of which to shop for your items.

Design Objectives

Our goal was to make an application that will cut the amount of time that is spent within a grocery store looking for items on your shopping list. This app will transform your shopping list into the most efficient path/list along with the items on a map of the store via pinpoints. There are no features/goals that we had to deviate from because we made significant progress throughout the project.

Methodology/Technical Approach

The goal is to make an application that will make it easier for grocery shoppers to get in and out of the store without having the difficulty of finding items on their shopping list, and utilize an easy to navigate user interface. The first approach is to use Firebase to store the store's inventory where the user will have the ability to search for the items that they would like to add to their shopping list. After the user is finished with creating their list, then they will be able to click a button that generate a new list that will be the most efficient path through the store. Along with that new list there is a map of the store layout with pinpoints of each of the item's locations within the store. The user will be able to click the pinpoints to gather more details of the item,

such as the item name, row number, and section number. The user can switch between the pages to see their list or the store map with the pinpoints of the items that are in their list.

User Profile

Figure 1: Grocery Shopper User Profile, illustrates the grocery shopper user profile. It depicts the users of the application, related experiences with other apps, and the expected tasks that can be completed with using GoCart.

<p>APPLICATION/USER TYPE: GoCart - Shopper</p>
<p>POTENTIAL USERS:</p> <ul style="list-style-type: none"> • People that go to the grocery store and are looking for a way to save time shopping
<p>SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: GoCart users should have experience with using Android applications along with using web applications online. The features of both parts of our application will be simple to use and will be easy for people to navigate through the application.</p>
<p>EXPERIENCE WITH SIMILAR APPLICATIONS:</p> <ul style="list-style-type: none"> • Kroger • Target • Walmart <p>These apps are similar in some respects in terms of in those apps you can create shopping lists or search for where the items are located, but not necessarily transforming your shopping list into the most efficient shopping path.</p>
<p>TASK EXPERIENCE: When the user initially opens the application, they will be directed to a login page where they can either login or create an account. After login they will have the ability to add items to their shopping list and when they are finished with their list, they can then generate the most efficient shopping path with the click of the button. This will output the most efficient list in list form along with a store map with pinpoints. Along with that there will be an estimated time listed on the screen which will tell the customer about how long it will take to get all the items on their list. As the user goes through their route, they can mark off what items have been collected allowing the next item to show up at the top of the list. By clicking on the pinpointed item, the user can expand the details showing the item name along with the row number listed.</p>
<p>FREQUENCY OF USE: For the mobile app you will need to install it, which is only done once. The app will be used whenever a customer is wanting to go to the store, so they will create their list and be able to shop more efficiently/quicker than without the application.</p>

KEY INTERFACE DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Easy to navigate UI with simple structured platform
- Total time to use the app/generate most efficient shopping path is minimal
- Visually appealing application
- Map with pinpoints of each item on list corresponding with the store map
- Generated list in order of most efficient path will list the corresponding row of the item and the item name

Figure 1: Grocery Shopper User Profile

Figure 2: Admin User Profile, illustrates the administrator user profile. It depicts the users of the application, related experiences with other apps, and the expected tasks that can be completed with using GoCart.

APPLICATION/USER TYPE: GoCart - Admin
POTENTIAL USERS: <ul style="list-style-type: none">• Grocery store management that wants to manage the inventory and to look at user accounts to see how many users are registered for the app
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: GoCart administrators should have experience with using Android applications along with using web applications online. The features of both parts of our application will be simple to use and will be easy for people to navigate through the application.
EXPERIENCE WITH SIMILAR APPLICATIONS: <ul style="list-style-type: none">• Kroger• Target• Walmart <p>These apps are similar in some respects in terms of in those apps you can create shopping lists or search for where the items are located, but not necessarily transforming your shopping list into the most efficient shopping path.</p>
TASK EXPERIENCE: It will open to a home page for the app just like before, but when you login with the administrator credentials then you will be able to access the inventory for the ability to add or remove items from the stored inventory. There will be another page with information regarding how many users along with user information for store purposes for the ability to send weekly ads, etc. There will be a tips page to help the admins use the application in terms of working with the inventory.
FREQUENCY OF USE: For the mobile app you will need to install it, which is only done once. The app will be used whenever a customer is wanting to go to the store, so they will create their list and be able to shop more efficiently/quicker than without the application.

KEY INTERFACE DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Easy to navigate UI with simple structured platform
- Simple design to add items or remove items from the inventory list
- Easy monitoring of the number of users of the application

Figure 2: Admin User Profile

Use Case Diagram

The following diagram, Figure 3: Use Case diagram, displays the case use for the GoCart application. This diagram depicts all the users along with the corresponding tasks each user will interact within the application.

Web Application/Android Mobile Application

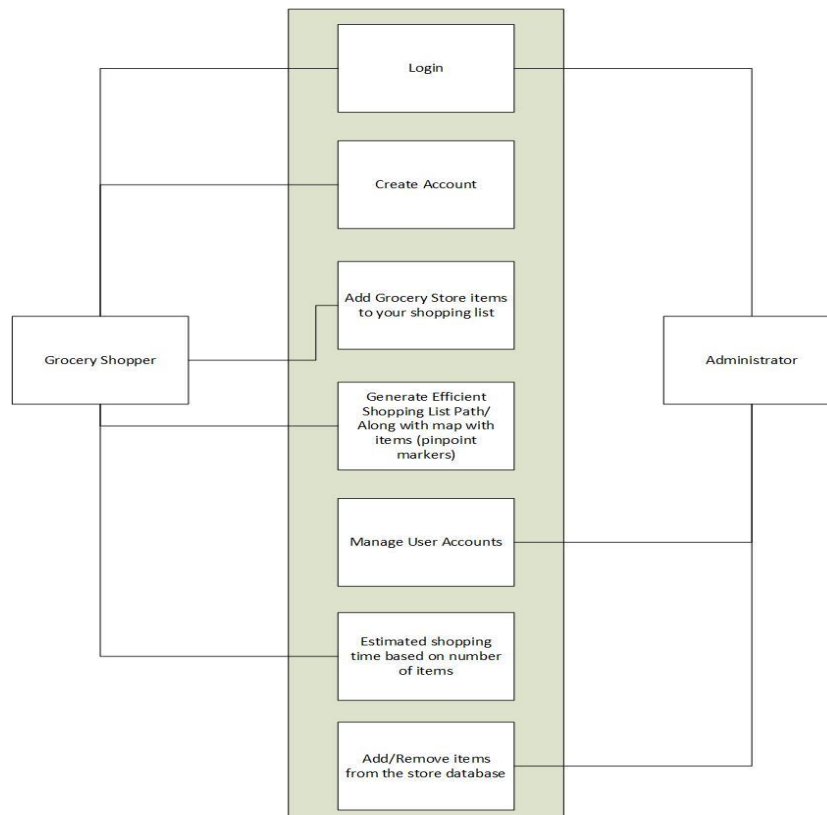


Figure 3: Use Case Diagram

Budget

In Table 1: Budget Table shows the budget expenses distributed throughout the project. The first half of the chart shows the budget regarding the software and hardware framework that was used. All the hardware used for the development of the project were free of expenses. The framework used to run and test the applications were all open source and free of charge. The second half shows the amounts projected from labor based off the current market. The wage cost was estimated using the average minimum wage of a developer of \$25 an hour. Given that there were three programmers and the average hourly wage, the expected cost was calculated \$20,000. The actual cost was totaled to \$0 since this project was created towards a Bachelor's of Science Degree in Information Technology.

<i>Category</i>	<i>Item</i>	<i>Description</i>	<i>Expected Cost</i>	<i>Actual Cost</i>
Materials	Hardware	Computer/ Android platforms used to test and run the application.	\$0	\$0
	Software	The development framework and tools used to develop the application.	\$0	\$0
Labor	Wage Cost	The USD hourly wage off a developer of \$25 an hour considering the current market.	\$20,000	\$20,000
Totals (Excluding Wage Cost)			\$0	\$0

Table 1: Budget Table

Gantt Chart

In Figure 4: Gantt Chart, there is an overview of our deadlines that we set forth for our project and the planned course of action throughout the project process of our application. The chart

includes the dates for assignments along with project tasks.

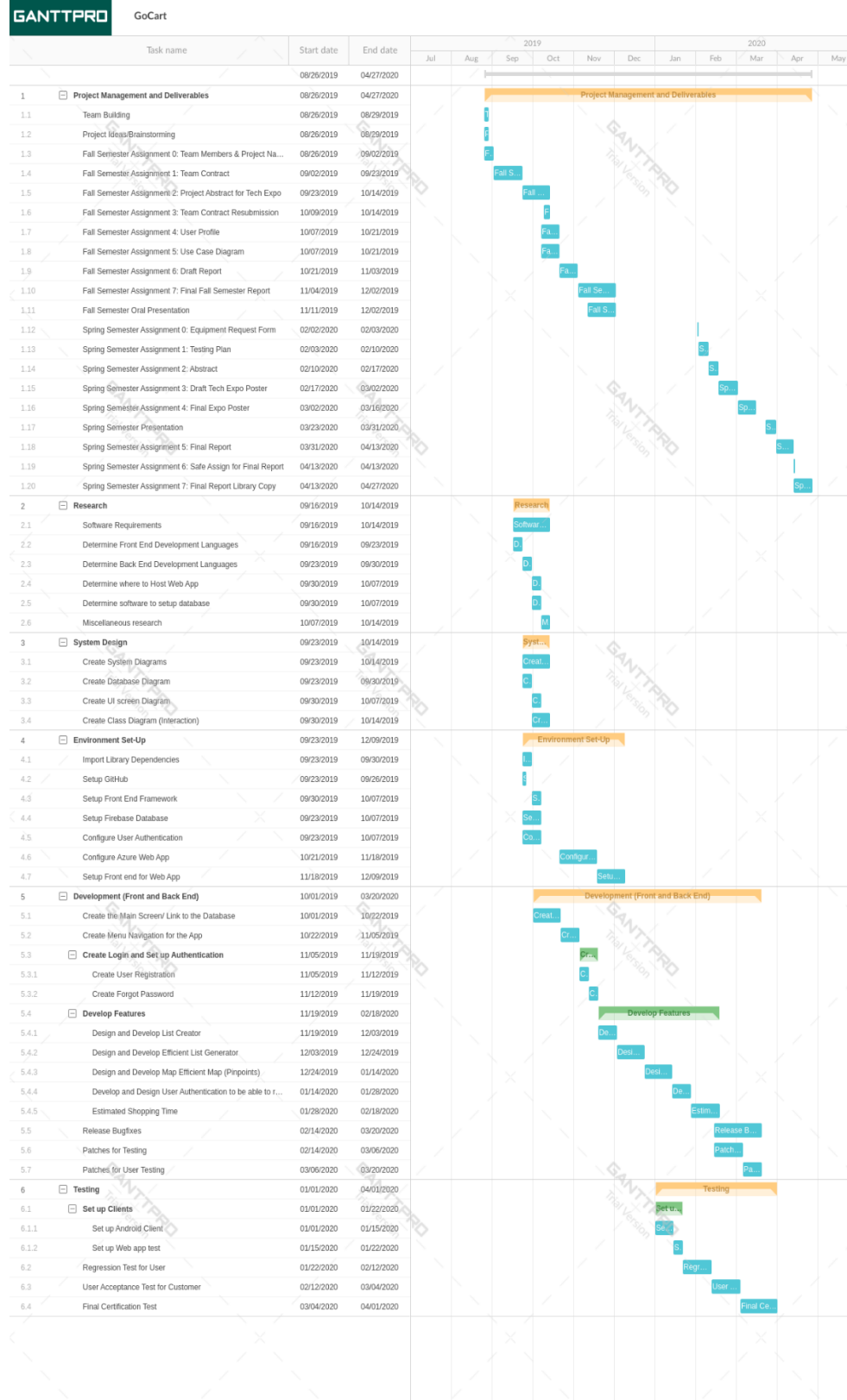


Figure 4: GANTT CHART

Problems Encountered and Analysis of Problems Solved

Problems encountered with the project included having issues connecting our database to our application. This problem was resolved by fixing various dependencies within the dependency file. Another issue that arose regarded enabling authentication with firebase, so we did research to find out the best way to go about using this method for authentication within our app. We faced issues with the map function of the application. Initially it was hard to connect the “user created list” to the store map and have the appropriate pins appear. This issue potential took the most time to solve because it took a lot of trial and error. To get a better understanding, we did countless hours of research and communication to help resolve the issue. After reviewing numerous examples, we successfully implemented this feature.

Future Recommendation for Improvement

There are a few things we could do with this application in the future to improve it. The first and most obvious being store expansion; the inclusion of more markets and larger grocery stores would make this app far more valuable. Partnering with larger supermarket chains and expanding the company would make this application beneficial to a bigger crowd due to more store options. The layout of the application can be altered to have more of a 3D or virtual view as well instead of just a screen layout of the store. Additionally, having more interactive features as well as making small improvements to the UI could help take this application to the next level.

Testing Plan

Overview/Methodology:

This section will go through the process that we have taken in terms of testing our GoCart mobile application, which is developed for Android devices. The different groups that will use this app throughout the testing portion of the project are the beta users (grocery shoppers) and

our team ourselves. The testing approach that we have taken is the refactoring approach by reviewing our source code and improving our features of our application for the app improve efficiency and eliminate the bugs that we may have within our application.

We will be conducting User Acceptance Testing where the different users will use the application to determine whether the application lives up to the requirements set forth by us. Also, this will determine whether the features are working properly like they are intended to work.

Scope of Testing:

The test cases that we have created will cover all the main features that we have created for our application. The cases will cover testing for each User Profile that will be for each type of user that would use our application. We are testing both profiles because it is important that all users have a good experience while using the application because if you only focus on one particular user type, then the other user types will most likely discontinue using your application if it is not fully functional.

Objectives:

The goal of this testing is to determine whether all our features in our application is working properly. The end equality is very important to satisfy user needs and making them want to use the app repeatedly when they go to the store based on the ease of use with a functional mobile application.

Strategy for Testing:

- 1.) Launch Testing to determine if the app can launch properly without it crashing.
- 2.) Stability Test to make sure the application runs without any issues while navigation throughout the app.

- 3.) Functionality Testing to make sure each one of the applications features is working properly.
- 4.) User Acceptance Testing to make sure the app is user-friendly and to confirm that there are no bugs or unexpected crashes that we did not see during our testing.

Logging Test and Procedures:

We will conduct our launch, stability, functionality, and user acceptance testing in the tables below that will have the results from our testing. For the functionality section we are testing each one of our main features in our application to ensure that the features are bug free by the time IT Expo comes around. Each of these tests will have one of two outcomes, which will be either pass or fail depending on the result of the test. Another test that we are conducting is allowing the end user to test the app to see if they encounter any issues/problems with the application that they would like to see changes within the application. This test will ensure a quality application for the different types of end users.

In Table 2 is the results from our launch test, which is a test to see if the application has any issues opening from the android mobile device. If the times successful is equivalent to the amount of times we have tested, then these tests will be deemed to pass.

Launch Test	Times Tested	Times Successful	Results
Android Emulator	25	25	No Issues
Google Pixel 2XL	25	25	No Issues

Table 2: Launch Test

In Table 3, we are testing to see how stable the application is in terms of if using the app over a long period of time to see if there is any crashes or problems. If the application does not experience any problems, then it will pass the test.

Stability Test	Hours Tested	Times Tested	Results
Android Emulator	2	2	No Issues
Google Pixel 2XL	2	2	No Issues

Table 3: Stability Test

The information from the Stability Test will help make sure we have an application that is bug-free along with not crashing when in use.

Table 4 is showing the results from our functionality testing, which is where we will be testing the different features in our application to ensure that they are working properly. This is key to ensure that our app is bug and crash free. If all the features pass this test, then our application will be deemed successful.

Feature Name	Issues	Results
Login	- None	Passed
List Creator	- None	Passed
Efficient Shopping Path List Generator/Map	- None	Passed
Estimated Shopping Time	- None	Passed

Table 4: Functionality Test

For Table 5 we are conducting test results from our users in our beta release to determine whether there are bugs or crashes that occur when their using the mobile app. These users are going to be grocery shoppers along with grocery store administrators. The following table has the results that we have gathered from the testing.

User	Issues Found	Passed/Failed
Grocery Shopper User 1	- None	Passed
Grocery Shopper User 2	- UI	Passed (Fixed UI Recommendations)
Administrator User 1	- None	Passed
Administrator User 2	- None	Passed

Table 5: User Acceptance Test

What we learned during Testing:

We have learned throughout the testing process that the best feedback of your application is from the end user and/or company that you're working with, which in our case that company is Clifton Market. This is crucial in determining whether the application that we have built meets the needs of the end users in terms of functionality. Some of the feedback that we had received during our testing is some of the user interface needed to be redone to ensure a better experience for the users of the application.

Technical Elements

Overview

We put a lot of focus on our design/layout of our application to ensure a good user experience when they use our application. A good design will attract users to an app because people are visually appealed to things and not necessarily just on the functionality of what the application itself does. Firebase, which is a service created by google will serve as our backend framework in terms of user authentication along with storing the store's inventory in a database.

Technical Architecture Diagram

In figure 5 below will show the architecture diagram for the application. This diagram will show the interactions that occur during the use of the application. The following interactions are with firebase authentication and firebase database.

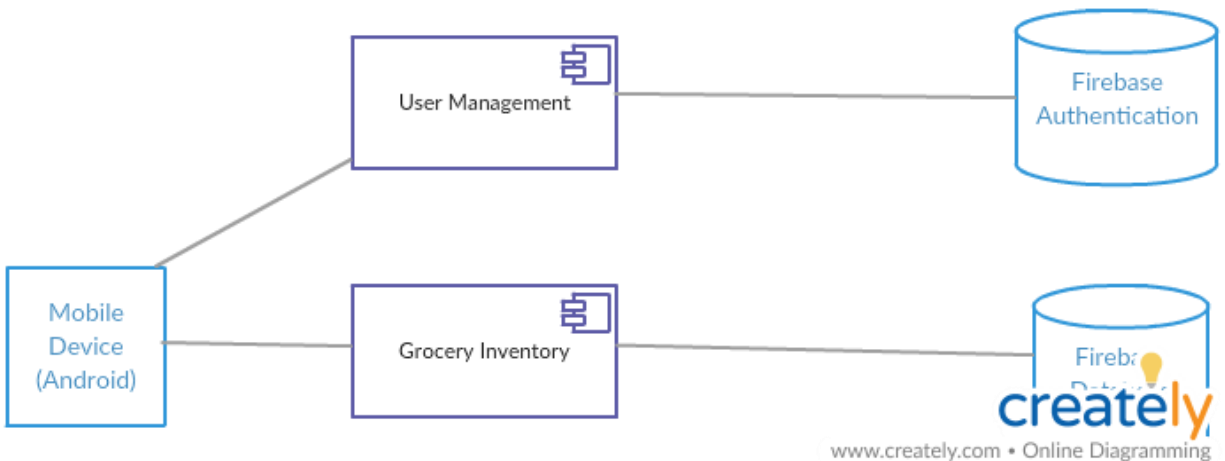


Figure 5: Technical Architecture Diagram

Application

For our application, we will be using Java for the main framework of the project. The project will implement Firebase to use for the authentication of users along with storing the database. The service is easy to incorporate into the project, which is why we chose this service to use with our application. The data for the inventory for the store will be stored in JSON objects from Firebase.

Database

The application will use Firebase which is a NoSQL database. Our data has a basic structure that is an easy to read from the database for the user to search items that they would like to add to their grocery shopping list. After the user completes their shopping list, then they will click a button which will then transform that list into the most efficient shopping path based

upon additional fields within the database for the items, such as row number along with row location in terms of what section of the row the item is located.

User Interface Screens

LOGIN

In figure 6 below shows what the login screen looks like when the user logs in to the app.

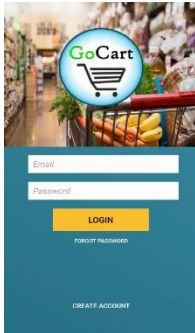


Figure 6: Login Screen

On this page the user can login to the app or create a new account. Login authentication is handled by Firebase. The image below shows what the login screen looks like for the application.

CREATE ACCOUNT

In figure 7 below the user can register for an account to use the application.



Figure 7: Create Account Screen

Users can use this page to create an account if they have not already. The authentication for the application is handled by Firebase. The password and confirm password would notify you if the

passwords entered do not match. Below is an image that shows what the create account screen looks like.

STORE SELECTION

As shown in figure 8 is where the user can select a store to use while using the app.

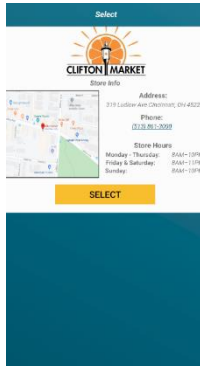


Figure 8: Store Selection Screen

The user can select which store they will be shopping from. (Clifton Market is currently the only store available. Store information such as address, and hours of operation are displayed.

INVENTORY

In figure 9 is where the inventory for the store is listed where the user can select different grocery item categories



Figure 9: Inventory Screen

On this page the inventory for the selected store will be displayed. The user can sort through different categories and/or search items by name. The user can then click on the item they want, and it will be added to their personal shopping list. The top menu displays icons to navigate

between pages. The navigation menu (from left to right) includes links to Store Map, My List, Store Selection, and Logout (returns user to login screen).

MY LIST

In figure 10 is where the users completed list is shown with the row and section numbers of the selected items are shown.

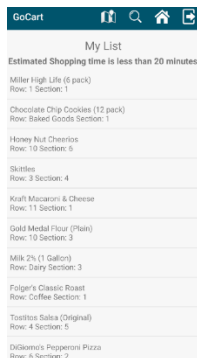


Figure 10: My List Screen

Here, all the items that the user selected will be displayed, along with their row and section number. An estimation of shopping time will be shown, this is based on the number of items the user has added. The top menu displays icons to navigate between pages. The navigation menu (from left to right) includes links to Store Map, Inventory, Store Selection, and Logout (returns user to login screen).

STORE MAP

In figure 11 below is where the store layout map is shown with the pinpoints of the different items that are in their shopping list.



Figure 11: Store Map Screen

An image of the store layout will be displayed. Pinpoints will be placed on this map that represent the location of each item the user has in their list. The user can click on the pinpoints to view information about that item. The top menu displays icons to navigate between pages. The navigation menu includes links to, Inventory, My List, Store Selection, and Logout (returns user to login screen).

Conclusion

During the fall semester of our project we learned how to effectively work together as a team along with following deadlines set by our Gantt chart. We learned how vital it is to form a good process of what you are trying to accomplish with your idea. Along with that there were speedbumps along the way once we started coding our application.

For the spring semester, we had learned a lot about working well together as a team and the ability to meet deadlines set forth by our Gantt chart. Maintaining a consistent pace is another aspect that we have strived in the project and thus far have retained. We have gained significant knowledge in Android application development, web app development, along with using Firebase as our database/hosting platform. We learned a lot about Firebase because we used it as our authentication method for users to login to our application. We found ways to overcome difficulties in the design process and adapted as the project evolved. We successfully created the product we wanted to, and the functionality and ease of use of this application match the expectations we set at the beginning.

Back Matter

References

- Lake, Rebecca. "Grocery Shopping Statistics: 23 Fun Size Facts to Know."
<https://www.creditdonkey.com/grocery-shopping-statistics.html>. Accessed November 10, 2019. <https://www.creditdonkey.com/grocery-shopping-statistics.html>.
- Marks, T. "12 Reasons Americans Hate Grocery Shopping." April 10, 2015.
<https://www.consumerreports.org/cro/news/2015/04/12-reasons-americans-hate-grocery-shopping/index.htm>.

Appendixes

Below is an image of our data query to filter out the different types of grocery categories from our firebase database. This will compare the category value in the database and if it is equal to that value, then in the drop-down selection of the category will retrieve each item that is equal to that value.

```
//Bakery Query to retrieve items that have a category value of Bakery
bakery.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

        for (DataSnapshot itemSnapshot : dataSnapshot.getChildren()) {

            String itemName = itemSnapshot.child("itemName").getValue(String.class);

            bakeryAdapter.add(itemName);
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
});
```

Figure 12: Code for Category Queries

In the picture below is our projects poster that we used for the IT Expo. This poster shows a basic understanding on what our mobile application accomplishes along with showing examples of what the application would look like on your phone.



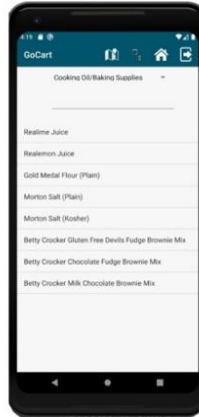
41
Minutes **Average shopping time according to Time Use Institute.**

THE PROBLEM

- Customers having problem locating items needed
- Inefficiency in shopping by not getting their items in the most optimal path in the store
- Too much unnecessary time spent

THE SOLUTION

- Personalized grocery list
- Store map
- Detailed inventory database
- Accurate in store item locations



ABOUT

- GoCart can enhance grocery shoppers' experience by taking the shopping list and generating the most efficient shopping path.

TECHNOLOGIES



PARTNER



Figure 13: GoCart IT Expo Poster