OPTIMUM SHOPPING APP

By

Samuel Yilema

Senior Project

Submitted to the Faculty of the

Department of Computer Science

in partial fulfillment of the requirements

for the degree of

BACHELOR OF SCIENCE

in

Computer Science

AUG 10, 2018

Nashville, Tennessee

Approved:

Technical Advisor:	Dr.Kamal Al Nasr	_____
Course Instructor:	Dr. Ali Sekmen	_____
Department Chair:	Dr. Ali Sekmen	_____

To all TSU family,

ABSTRACT

Usually customers in retail stores wasted a lot of time looking for specific of item and trying to remember their shopping list items. As in this modern and fastidious world, each minute is valuable for us so, the Optimum Shopping application saves our estimable time by making the picking and checkout process faster. This paper composed of a new Optimum Shop Application practiced for an Android Mobile, meant for the customer of some retail stores. In this Android application, data is stored using Firebase database which is a type of synchronization database and is stored within a google cloud.

ACKNOWLEDGMENTS

This project work itself an acknowledgement to the desperation, sincerity, passion and technical assistance contributed to it by all our group members and our talented teachers. With immense gratitude we acknowledge our indebtedness to all the persons whose support and guidance have helped us in carrying out this project work dynamically and up to the standards of our prestigious institution.

We wish to express our deep respect and thankfulness to Dr. Ali Sekmen and Dr.Kamal Al Nasr, our project guide, without their guidance, insightful comments, smart encouragement and valuable help this work would not have been a success.

We have great pleasure in acknowledging the help we received from all those who favored in giving shape to this project, a sincere thanks to: All the teacher's who post video on YouTube and UDACITY.com. The Smart people out there who invented the Internet and Google. And Obviously our great teachers in TSU.

TABLE OF CONTENTS

Page

DEDICATION ii

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF FIGURES vii

Chapter

1. INTRODUCTION 1

    1.1 Problem Statement 2
    1.2 Literature Review 2
    1.3 PROPOSED APPROACH 4
        1.3.1 Conceptual framework 4
        1.3.2 Grouping data by cluster Analysis 4
        1.3.3 Graph Cellular Automata 5
    1.4 Research Goal and Objectives 5
        1.4.1 Goal 5
        1.4.2 Associated Objectives 5
    1.5 CONCLUSION 6
    1.6 Project Organization 6
    1.7 Project Organization 6

2. REQUIREMENTS ANALYSIS 7

    2.1 Functional Requirements 7
    2.2 Non-Functional Requirements 8

3. DESIGN AND IMPLEMENTATION 10

    3.1 Architectural Design 10
    3.2 Detailed Design 11
        3.2.1 System 11
        3.2.2 User interaction Architecture 12
        3.2.3 Database design 12

4. IMPLEMENTATION 14

4.1 Database 14
4.2 First Activity 17
4.3 Second Activity 18
4.4 Third activity 21

5. RESULTS 23

5.1 Database tables. 23
5.2 Store selection. 24
5.3 User selected item and Location. 25

6. CONCLUSIONS 27

LIST OF FIGURES

1.1 A shopping list application. 4

3.1 The overall system context design. 10
3.2 System Architecture. 11
3.3 User-case design architecture. 12
3.4 Database design architecture. 13

4.1 Code to select Kroger table. 15
4.2 Snapshot of Realtime database. 15
4.3 Snapshot of Realtime database. 16
4.4 Dependencies. 17
4.5 Dependencies. 17
4.6 Xml layout code. 18
4.7 Listview with item add from database. 19
4.8 Code that add file to listview. 19
4.9 Code for onClick method. 20
4.10 Code for onClick method. 21
4.11 Code to compare location of the item. 22
4.12 code result XML file. 22

5.1 database. 23
5.2 welcome page and Store selection menu. 24
5.3 Available item in store selection. 25
5.4 Selected item and location. 26

CHAPTER 1

INTRODUCTION

As its the world of technology and everyone wants easy way to do their daily activities. People need shortcut and smart ways with the help of technology. As the demand of technology is increasing day by day, specialists are find more advanced and optimized ways to serve people. Optimum shopping applications is one of the advanced method to do shopping list. Basically, optimum means the finest possible results within a particular set of circumstances. And optimum shopping list provides us best possible results of shopping for our daily needs; from milk to deodorant whatever a person needs in his/her daily life. Optimum shopping app provides us simple list of items that an individuals needs to purchase in local retail stores and it also enables to create quick list of items. By this application a person can save his/her time by avoiding regular visits to the local stores. When we look into the examples of optimum shopping we can see that since 2000, digital business of buying and selling in China has entered in a time of speedy development. From optimum shopping large organizations are earning million dollars. In contrast, small manufacturers faces a low performance. If someone wants to promote the healthy progress of optimum shopping system, it is essential to learn the stability of application users from the perception of customers behavior. Though customers certainly incorporates online and offline shopping, vendors are still discovering the potentials of optimal combination. The consumer journey no longer visits a single retail store for shopping more than once. For example, according to a survey, while a customer making shopping in local store, 80 percent of customers first trying remember what they want purchase in advance before buying the item. This ratio is even greater in different countries for example, in United States its 89 percent, in United Kingdom its 90 percent and in China its 92 percent. These optimum applications provide effective and desired access to customers according to their interest.

1.1 Problem Statement

Shoppers wasting a lot of time looking for specific item location and trying to remember a shopping list in stores. Its very important to make a shopping list before we head to grocery store. You will save a lot of time and money. The shopping list will give access to customer with an organized approach to get in and out of the store without wasting time looking for unwanted item. In addition, it will give the shopper enough confident in the list to know items not needed. The shopping list can reduce your food expense by eliminating unnecessary items. In this project, mobile application will be developed to make easy shopping list note and locate specific item in the store. According increasing demand of technology in shopping systems customers wants more optimized solutions and ease of managing their shopping list. But as application advances there are greater chances of increasing complications and problem in development of systems as well as in use of systems. When look toward the requirements and basic needs of customers, we can clearly analyze that every customer and retailer wants a manageable and easy to use application, with lesser time to operate and save a lot of time for customer when they pick their order. Specialists trying their best to modify already existing and new application to develop according but they also face different problem as well. Such as, shopping list application may be difficult to develop, time consuming, Listing is a process which attempts to optimally assign number of item to a group of activities that should be executed within an indicated order list while sustaining multiple restrictions. Optimized shop listing is a type of conventional listing task need to be done, in which, it has been comprehensively examined in various fields of technology and engineering. The core purpose in the shop listing problem is to discover an optimal arrangement of all items on all retail stores while attaining an optimized independent function, such as loss of time, and purchasing unnecessary items.

1.2 Literature Review

In recent years, various number of Poles took benefit of the electronic shopping applications. The reasons behind the partial rejection of taking list note in an old-fashioned method in favor of mo2

bile shopping applications are suitable, easy and easy and a time saver to use. The growth in the number of optimum shop applications consumers also contributes to various advertising activities prearranged by optimum shopping Apps. Concerned for their reliable consumers and obtaining latest ones is a challenge for them. Publications of industry recommends that an assessment of target customer groups is important for the success of the ongoing market existence of an organization. The current application that provides the easiness of taking note applications to the localshopping stores, latest application are being developed for local retailer to interpret the benefits of local shopping stores to the customer. For example, the social characteristics of shopping can be provided share information with other shoppers and family member. However, various number of shopping store still appear to be looking for how to integrate best shopping list and easy pick items location, numerous examples previously exist of retailer trying to optimize this incorporation. For example, the shopping chain of Hema, has established a phone application for shopping that permits customers to check the production and goods of various outlets. The ware house has investigated with, between other things, inspirations as a way to provide mobile application item finder. Their objective was to present more personal services to clients in the local located stores by making their customer and buying information noticeable to the sales officials every time a customer uses their application. As per the research shows, a ratio of customers would be happy to obtain other task such as texting calling during shopping. Products and variety of goods are most essential part of any shopping application and the multi-use of application is also a way to impress the customers, because its the easiness which matters most to the customers. The application structural design is a complicated design activity that leaves impacts on all consequent customer and retailers phases such as adding items, prototyping, and removing items. An extensively established description of application design is gathering and analyzing customer needed. Application with fundamental structural design are developed with an arrangement of item list to purchase. Figure 1.1 shows example of some application that already been developed.

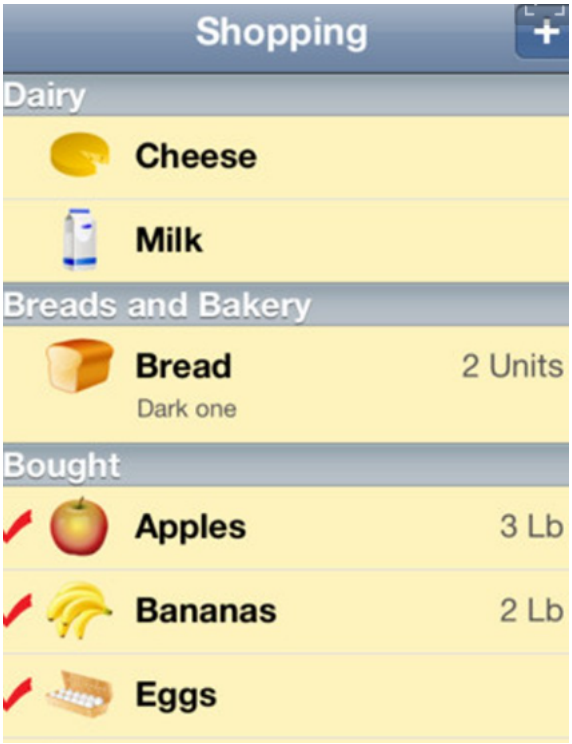

Figure 1.1: A shopping list application.

1.3 PROPOSED APPROACH

Following are the proposed approaches for the best optimum shopping solutions such as:

1.3.1 Conceptual framework

The model of the frame-work is a multifaceted arrangement and it contains information layer (data layer), technology of application, presentation layer and methodological layer of outcomes. The procedure used to organize optimum shopping store consumers are based on the cluster assessment and graphic cellular automation sustained by the data base device and the operational store.

1.3.2 Grouping data by cluster Analysis

Analysis of cluster is an idea of information assessment and machine learning, gathered from the wider idea of unattended learning. This is a group of method used to segregate consistent subcategories of populace objects. It is also known as clustering. The reason behind the cluster

analysis is to distribute a space into groups/clusters that objects inside a specified subsection are as comparable as possible, but then they vary as much as entities from other subgroups. Formerly, clustering was directed in a particular method, established on the observation and conclusion of the researcher.

1.3.3 Graph Cellular Automata

The reason of the study was to generate a typology of optimized application consumers based on the mobile app they frequently uses so we can easily improves the way of getting more visitors to increase the optimization of mobile app shopping list.

1.4 Research Goal and Objectives

1.4.1 Goal

The research goal of this project is to develop a mobile device application that help customer to organize a shopping list and will give them shortest route to pick they grocery items in store.

1.4.2 Associated Objectives

The specific objectives are listed as:

1. Create first project and get familiars with android studio. We will learn detail about android studio.

2. Create fire-base connectivity to store relevant information about out app and also design our data structure

3. Develop android application by using android studio.

4. Design and hard code widgets and layouts, UI Events, Event Listeners.

1.5 CONCLUSION

The research and study shows that as the demand of technology is increasing rapidly, professionals are trying to find more advanced and optimized ways for shopping list application. Basically, optimum shopping list solutions provides us best possible results of shopping and help us to shop with ease. The reasons behind the partial rejection of taking list note in an old-fashioned method in favor of mobile application shopping

1.6 Project Organization

list are suitable, easy and a time saver to use. The growth in the number of optimum shop applications consumers also contributes to various advertising activities prearranged by optimum shopping Apps. We have also stated different research of researchers and also provided methods to improvise shopping list solutions such as, conceptual framework, data collection and Graph cellular algorithms. As the researchers are finding out more optimized ways by using different experiments and methods, soon we will be in the world of advanced application and we will surely able to have a unbelievable experience of most optimized shopping application.

1.7 Project Organization

Chapter 2 will provide a comprehensive requirement analysis that includes functional and nonfunctional requirements. Chapter 3 will give architectural and detailed designs and it will be followed by implementation in Chapter 4. Chapter 5 will present the experimental results and Chapter 6 will give conclusions and talk about possible extensions of the project.

CHAPTER 2

REQUIREMENTS ANALYSIS

This chapter discusses the functional and non-functional requirements for our optimum shopping app. The functional requirements second will explain the functionality of the system itself and the predicted functions that need to be implemented to complete it. The non-functional requirements will discuss the development issues and limits of the system itself. These requirements will not explicitly define how the system must be used, just how it works.

2.1 Functional Requirements

1. The system shall enable a user to create a new shopping list menu:
    - user shall be able to enter item name by typing and the system shall save it to item list view menu.
2. A user shall be able to interact with a app vocally to enter item name. The the system shall enter item name to system:
    - The system shall have speaker input when user call item names the system shall type the item name and save it item list view.
3. The user shall have add item option that where they type item name entered to list view.
    - If a user want add item name when ever they remember this option should be very help full.
4. User should have auto suggestion of item name on the keyboard
    - If the user type wrong item the system should give the suggestion should be very convenient especially for type on the go.

5. User shall have auto pick item option

    - Auto pick is an option button that will give the user shortest route to pick their item at any store.

6. A user shall have delete item option.

    - user can edit or remove item that already add to their shopping list by using this option, sometime a user change their mind about their shopping list they like to edit or delete so this option will give a user control of their shopping list.

    - Any item that deleted by user the system should remove it from the shopping list or update it to item name that user entered.

7. The system shall remove item name on the list when a user pick the item.

    - when a user uses auto pick option they system will mark the item as picked or move it to history activity list.

8. User shall be able to select store name where to shop on the menu.

    - A user can choose any store like to shop their grocery list. For example will have list of store name Wal-mart, Kroger, Pulix a user can shop at any of those store our system will be connected to those store database to located item location.

9. User shall be able to search item they already entered to list.

    - When user entered a lot of of item its very difficult to search item name on big list so it is to search it by name.

2.2 Non-Functional Requirements

1. The system will be developed using Android studio

Rationale: The system will be developed using a Android studio by java code. Since this android studio implementation works only for android phone users, we are limited to the android users.

2. The system shall support any android user. Rationale: The system is expected to be used by any android user around the world. Based on the size of android user, this system should be useful to many users. However, the system should also support ios users in future.

CHAPTER 3

DESIGN AND IMPLEMENTATION

This chapter will discuss, in detail, the architectural build of the optimum shopping list that it will use. The architectural design will give a view of the overall system, including its subsystems. The detailed design section includes system architecture and software architecture and database architecture, in addition to be included coupon system and GPS system in the future.

3.1 Architectural Design

Figure 3.1 shows that overall idea of Context design of our application. A user interacts with the system to select shopping store and make item list. The item list should be listed on display menu as well as the location of the item that a user selected. The system processes the user selection and get correct information form the database to display items information to user by mean android code implemented to create database connection and to access database. The database helps the system to store and retrieve data as requested from the user. Our application has many functionalities that will interact with a user to access different activities

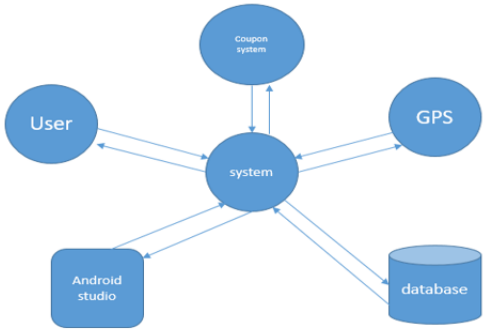

Figure 3.1: The overall system context design.

3.2 Detailed Design

3.2.1 System

Figures 3.2 shows the system have many functionality and how they interact each other. We have four functions that system uses to interact with customer such as select store, add item, find location and setting. Select store option will give access to the customer to choose their preference store where they want to shop their items and the system will display it to their shopping list menu. This option is very useful when you trying to shop in different location. Sometimes customer refer to shop specific store for many reason for example one store have many option than other store so selecting store option will give us option to choose any store on the list. Add items option helps customer to enter any item list to shopping list. This option gives access to user to add any time they would like to shop and easy to use. The other option user can use is Find location button this button helps customer find the items location very quickly and it makes shopping at big store much easier, faster as you use this application. This option could save a lot of time customer that spend searching one item in the store. last option we have on display menu is setting this button specific set up of a system and customer can select any color of their application.

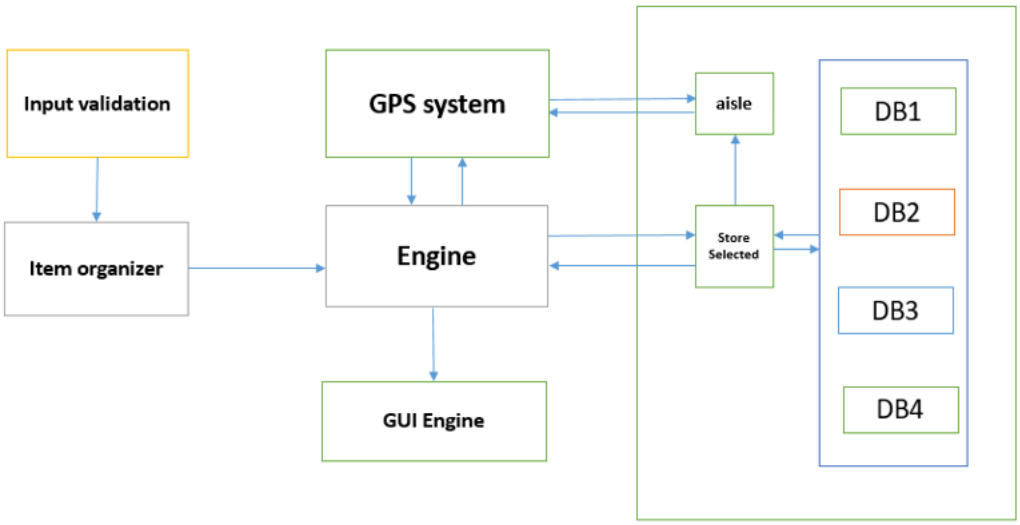

Figure 3.2: System Architecture.

3.2.2 User interaction Architecture

Figure 3.3 Shows how a user can add item list to their shopping list. First user need to type the item name and press add button so the item it will display on view list. When use user add list of items a delete button should automatically create with item list. Type item is where you a user write item name to add to the list and click add button to save it. Delete option will create when the user enters new item to the system. When user entered wrong item, user will have option to go back and change the item list or if the item on the list did not want it any more user can remove or update at any time. Android studio code will be implemented to handle all these activities. In the future search item list option will be included at this we are only focusing to implement Type item, add item and Delete.

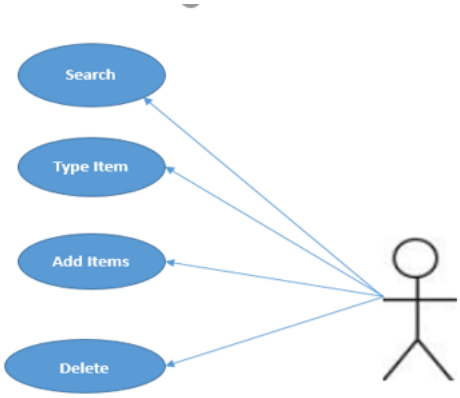

Figure 3.3: User-case design architecture.

3.2.3 Database design

Figure 3.4 shows the database set up and how we access data from the database. The database helps a user to store data file and to save same set ups. Whenever a user opens the application user will get same information that was entered previously and display all activity history on the file. JDBC is built in method to connect android studio and SQLite database. When user read or write information from the database the user data manipulating language (DML). DML helps to create query that will read or write data to database. This connection helps us to access database file at

any movement when the application run. On the other hand database schema will help the database to organize and manage the database. A database can have many multiple schemas in the database but the user can only use same schema associated with the same name of the user. basically, any user who created object in the database it just created on its own schema unless user specifically directed to create another one. The last one is data, data is where the actual data is being stored in SQLite. This data is translated in to efficiently movement process and saved as binary digits.

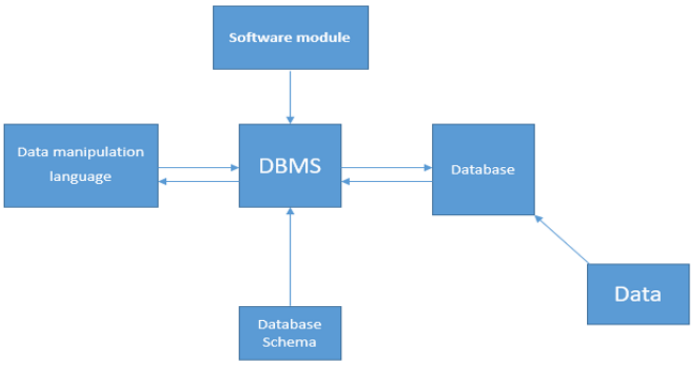

Figure 3.4: Database design architecture.

CHAPTER 4

IMPLEMENTATION

This chapter will present the implementation details of each activity described in Chapter3. Some partial code will be shown for clarity.

4.1 Database

Firebase database is backend database that will allow user to sore file build application. Normally we use database connection to read file from server. But on firebase only need synchronization method that allow to get real-time activity in our database. on the other hand, the performance is very fast reliable to access. Create firebase project and download configuration and register your app id to download file the google-services.json file. This file provided connection between the database and android application. Database is very important for this project because item list name and location need to be store in database file in order let user pick items from listview. In this database we have three different tables that hold different data information for three different stores. The first table name called Kroger table in this table contains items Name and Location. Each item has unique name and location assigned so that item can be identify quick when user selected to find item location stores. When a user selected Kroger table to add their item list to this table the list view will show them all available item in this table and for each selected item that user want to add to shopping list this code below will show the item name and location in list view.

```
database = FirebaseDatabase.getInstance();
ref = database.getReference("Kroger");
list = new ArrayList<>();
adapter = new ArrayAdapter<String>(this, R.layout.user, R.id.userinfo, list);
ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for (DataSnapshot ds : dataSnapshot.getChildren()) {
            user1 = ds.getValue(User.class);
            list.add(user1.getName().toString() + " ");
        }
        listView.setAdapter(adapter);

    }

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

    }
});
```

Figure 4.1: Code to select Kroger table.

The database contains some other tables such as Publix and Walmart but all tables has similar code to retrieve data from database. The firebase database is fast and easy to access when you make change in database file any change that we make in database file it will show it in Realtime activity of the database change. The below figure the snap shot of our Realtime database with different type of tables name and each table contain different information.

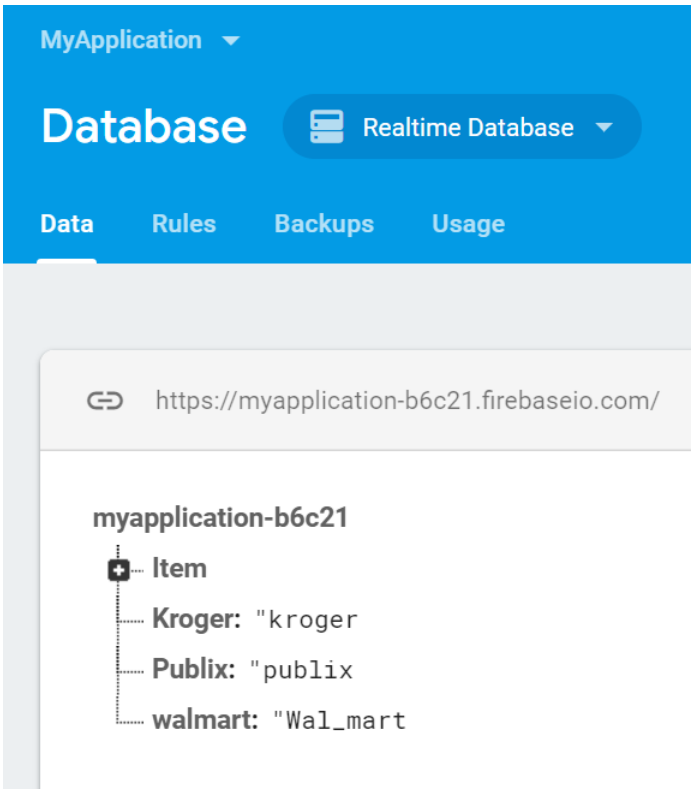

Figure 4.2: Snapshot of Realtime database.

Once all the table created in database we pass Json code from google to android studio to synchronize our database and android studio. Then this code will help us to connect our project to database and get database information.

Figure 4.3: Snapshot of Realtime database.

Then adding google services classpath and maven URL this help our project dependencies provides Google Analytics for Firebase functionality. Functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users. Figure 4.4 show the dependencies code.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.google.firebase:firebase-core:16.0.1'
}
```

Figure 4.4: Dependencies.

4.2 First Activity

First activity is first page of our application that contains welcome page and drop-down button that give options to user where to shop their items. In this java class we drop-down button code let user to pick store on the list and add their items to shopping cart. the XML activity contains types of button on this page has text view to show welcome massage and ADD item button to move to next page of the application. This button has activity listener and let user to pass the store selection to next activity.Figure 4.5 below shows our drop down code and next activity button.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Spinner mySpinner = (Spinner) findViewById(R.id.spinner1);
    ArrayAdapter<String> myAdapter = new ArrayAdapter<~>( context: MainActivity.this,android.R.layout.
    myAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    mySpinner.setAdapter(myAdapter);
}
public void Additemslist(View v)
{
    if (v.getId() == R.id.button)
    {
        Intent i = new Intent( packageContext: MainActivity.this, main_menu.class);
        startActivity(i);
    }
```

Figure 4.5: Dependencies.

The Android platform uses XML files in projects for many purposes, from providing basic configuration of the application in the Manifest File, to using XML Layout Files to define the user interface. Elements uses these standard files in the same way they are used when working with the Java language, so Elements developers have access to the same controls and UI capabilities as all other Android developers, fully natively. There are many ways for working with XML Layout files in your Android projects: You can edit the files in XML format using the regular code editor in Fire and Visual Studio. This option is favored by many Android developers, and gives you full control about your UI design down to the most minute detail.Figure 4.6 shows first page Xml code.

```
<TextView
    android:layout_width="150dp"
    android:layout_height="45dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="114dp"
    android:text="SELECT STORE"
    android:textAppearance="@android:style/TextAppearance.DeviceDefault.DialogWindowTitle"
    app:layout_constraintBottom_toTopOf="@+id/spinner1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:textStyle="bold"/>

<Spinner
    android:id="@+id/spinner1"
    android:layout_width="match_parent"
    android:layout_height="84dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="202dp"
    android:dropDownWidth="match_parent"
    android:spinnerMode="dialog"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="243dp" />

<Button
    android:id="@+id/button"
```

Figure 4.6: Xml layout code.

4.3 Second Activity

On this activity user can perform a lot of activity such as check item, search item, unchecked items and pass selected item to next activity. Check item button is important to add item to shopping list. This will allow user to add selected item in shopping list and removed item from the shopping list by checking and unchecking the checkbox. Searching item help user to find item that they want add check to add to their shopping list. Search item option will become handy when data table have to many items on the list its easier to just find item by the name.

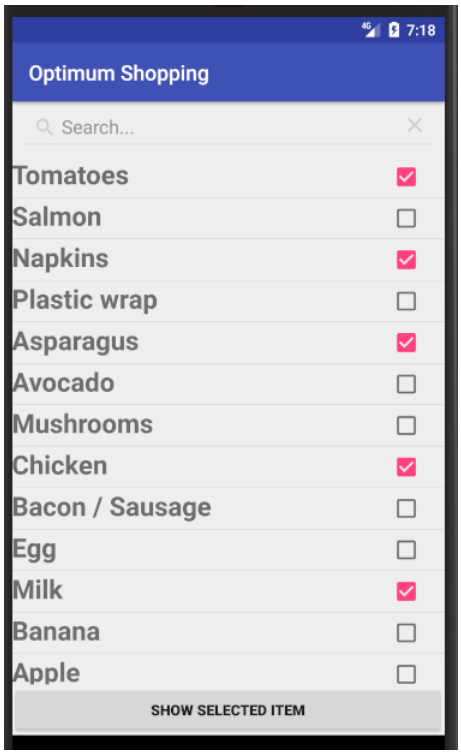

Figure 4.7: Listview with item add from database.

Also there is a java code that make this view work when user click it. We use Onclick listener to act when user select those any of items. This item list get data from firebase database. the java code to get this application. We create database reference to access data from firebase and arrayadapter to show our file to listview.Figure 4.8 shows code for that add item listview.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mainmenu);

    button = (Button) findViewById(R.id.button2);

    listView = (ListView) findViewById(R.id.tvItemSelected);

    listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    database = FirebaseDatabase.getInstance();
    ref = database.getReference( s: "Item");
    list = new ArrayList<>();
    adapter = new ArrayAdapter<String>( context: this, R.layout.user, R.id.userinfo, list);
```

Figure 4.8: Code that add file to listview.

```
public void onClick(View v) {
    SparseBooleanArray checked = listView.getCheckedItemPositions();
    ArrayList<String> selectedItems = new ArrayList<String>();
    for (int i = 0; i < checked.size(); i++) {
        // Item position in adapter
        int position = checked.keyAt(i);
        // Add sport if it is checked i.e.) == TRUE!
        if (checked.valueAt(i))
            selectedItems.add(adapter.getItem(position));
    }

    String[] outputStrArr = new String[selectedItems.size()];

    for (int i = 0; i < selectedItems.size(); i++) {
        outputStrArr[i] = selectedItems.get(i);
    }
```

Figure 4.9: Code for onClick method.

For this activity we use XML file for layout view for this application. Any changes you make to your XML Layouts in Android Studio will automatically sync back into your project, and elements defined in your layouts and the other XML files will be available via the static.

```
<SearchView
    android:id="@+id/searchview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:queryHint="Search..."
    android:layout_alignParentLeft="true"
    />

<Button
    android:id="@+id/button2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:text="@string/submit"
    android:textStyle="bold"/>
<ListView
    android:id="@+id/tvItemSelected"
    android:layout_width="fill_parent"
    android:layout_height="411dp"
    android:layout_above="@id/button2"
    android:layout_alignParentTop="true"
```

Figure 4.10: Code for onClick method.

4.4 Third activity

This activity contain the last page of our application and selected items list that user selected from preview list. This activity contains listview and button. The listview shows selected item that user need to shop in the store with item location where located so that user can pick the item in store without wasting anytime. This code below that allow the user to find closet item to pick first. To find shortest route in database first you need to find item location and compare it all other item location and return closet location to the user.

```
public static Comparator<String> StringAscComparator = new Comparator<String>() {

public int compare(String app1, String app2) {

String stringName1 = app1;
String stringName2 = app2;

return stringName1.compareToIgnoreCase(stringName2);
}
```

Figure 4.11: Code to compare location of the item.

The XML layout we use to show last result of our project in Listview that contain item name and location. The listview on this layout more organize and has a lot of information about the item information. Figure 4.12 shows our layout XML file.

```
<?xml version="1.0" encoding="utf-8"?>
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/outputList"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
```

Figure 4.12: code result XML file.

CHAPTER 5

RESULTS

This chapter will present the results through a series of screen shots of the programs written during the development of the project. It will also include the database file result.

5.1 Database tables.

Figure 5.1 shows database information. Firebase Realtime Database uses data synchronization-every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code. This database has three different table each table have different item and location and when user select store name the system will get information from this selected table and provide output information to user. each store has different style of organizing their location and items so we created table for each store, so they can store any information to help user to find item location in store.

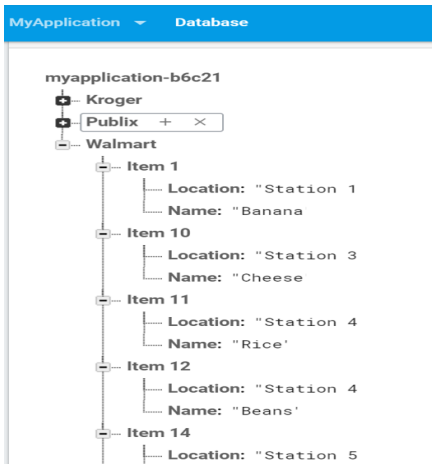

Figure 5.1: database.

5.2 Store selection.

Figure 5.2 shows welcome page and store selection. This page has to many functionalities that user need to move to next screen and to select their preference store location where they want shop. Shopping is most complicated if you do not know what you need and where to shop. This application will solve many issues that a user may have to find item location in sore. User can select their sore and add items to their shopping list to any sore they like to shop. The dropdown menu is to select Store name that available in the Store and there is button that will move user to next activity when clicked.

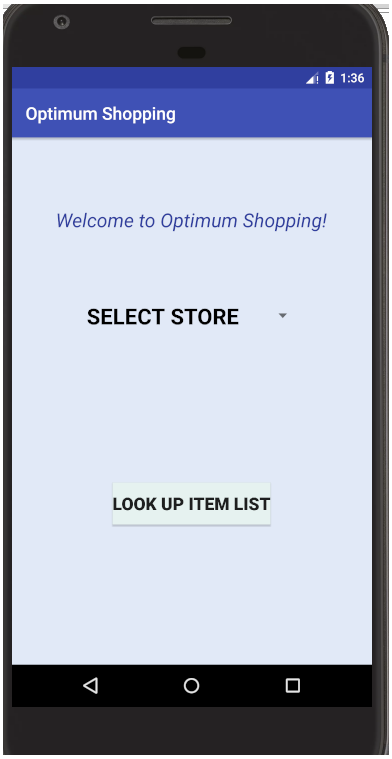

Figure 5.2: welcome page and Store selection menu.

Figure 5.3 this image show a list of item that stored in the database with checkbox button to select the item. Whenever the database file updated it should reflected on list view. The database contains many type of tables each table have specific information of items. Whenever the table update items name and location it will show the real-time change on list view as well. Search box

is field accepts typed-in text in order to look up item or launch a search for related item name in listview. When the table have to many items is not easy to find items name by scrolling down its best to use search item by name. when you search item by name its very easy to find the item name we want.

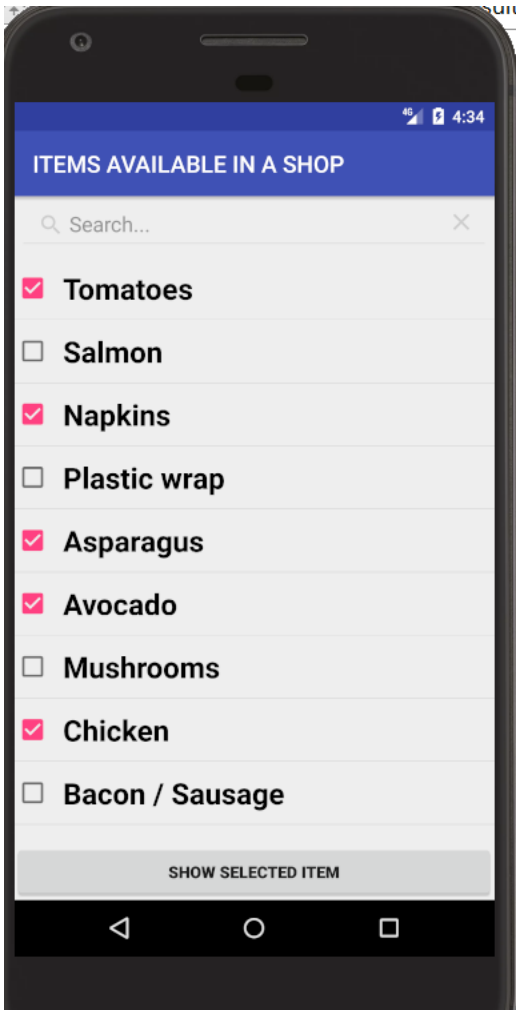

Figure 5.3: Available item in store selection.

5.3 User selected item and Location.

Figure 5.4This figure shows list of items that the user already selected on the previous page and it will organized it on order from the nearest aisle that already organized on the database by putting different tables, each table represent a different store under each table the Item number with de-

scription of the location of this item and the name of it so when the user selected it gives him/her the shortest way inside the store for example:table Walmart ¿ Item 8 ¿ Name Fish Location Aisle5.

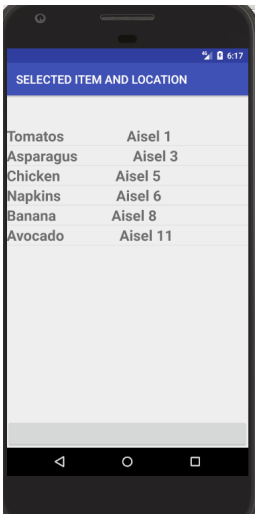

Figure 5.4: Selected item and location.

CHAPTER 6

CONCLUSIONS

During this project we created mobile application using android studio and Firebase database. We also created firebase database synchronizationevery time data changes, any connected device receives that update within milliseconds. The database contains three tables and each table also contain item name and location. These location and item name application uses to provide available items in the store. The optimum shopping apps does help user by organizing item list and provide the shorts route to pick item in the store. The application ask a user to select shopping store to add they items and when user select store our java code will get selected store id and provide selected store table information to listview. The listview provide check box button with item name so user can add any item on the list to shopping list when they ready to check out they click checkout button and the next page will show selected item and their locations. This page will allow a customer to pick their item fast in the store. we successfully completed this work through the course of the project. How ever there is are many advances that can be made with in this project. For example, adding bar code reader to this application will give a user more access and save more time by checking out their own items on the go. Improve sharing a item list, you can decide whether to let people edit or just view it. To control who gets to share a list item, you can invite certain people or everyone. This project can be improved even more ways as we mentioned on the example above.