Android Development Project Report on Grocery Shopping List Application Using Kotlin in Android Studio

Abstract:

Shopping is an intrinsic part of everyday life. The purpose of creating shopping lists is to efficiently manage time and resources while shopping and not forget something important to buy. The paper-pen based approach is no longer the only or effective means of creating shopping lists.

With the advent of the mobile phone era and the frequent use of mobile apps to perform everyday tasks, the trend to use apps for creating and managing shopping lists is becoming more popular by the day. There are hundreds of “Shopping List” apps for both Android phones and iPhone. Also, the design of the mobile app interface contributes a significant share in human computer interaction (HCI) research. The goal of this paper is to study the transformation from paper-pen based approach to the mobile app approach, analyze user experience with the most popular shopping list apps for both Android and iPhone; and suggest new ways to improve this experience both in terms of usability of app and user interface improvements.

1 Introduction

1.1 Overview

People today are very busy and tend to forget what they need to buy. There are existing apps which allow the user to make a reminder list, but when a user is in the market, he/she needs to look up in the list. This app will give the users an alert of the stores where they can buy the stuff mentioned in the list, thereby reducing the effort to remember and manually search for stores for a given item. This way whenever a user goes to market, he/she will automatically get updates about the stores to shop various items.

1.2 Purpose

The main aim of this project is to list the items so that whenever users go to grocery stores, users will not be able to forget their items and this grocery application helps the users to tackle their day to day chaos more effortlessly. It's not easy for the users to remember every item in this hectic lifestyle, they frequently can't recall their required necessity so we decided to build an app to store the items in the database for their future use. After buying the items users can delete the added items in the database.

2. Literature Survey

2.1 Existing Problem

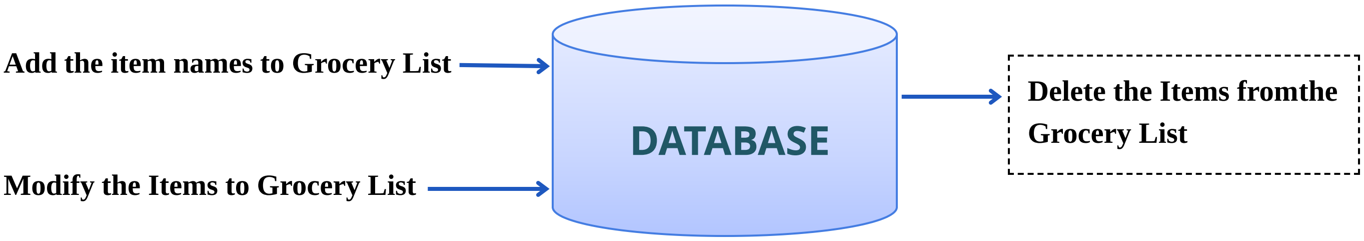
It’s not easy for the users to remember every item in this hectic lifestyle, they frequently can’t recall their required necessity so we decided to build an app to store the items in the database for their future use. After buying the items users can delete the added items in the database.

2.2 Proposed Solution

The main aim of this project is to list the items so that whenever users go to grocery stores, users will not be able to forget their items and this grocery application helps the users to tackle their day to day chaos more effortlessly. The goal of this project is to make an app that stores the user items in a cart and can modify and delete the added item in the list. To develop a reliable system.

3. Theoritical Analysis

3.1 Block Diagram


Add the item names to Grocery List
Modify the Items to Grocery List
DATABASE
Delete the Items fromthe Grocery List

3.2 Hardware / Software designing

SOFTWARE

The Software Package is developed using Kotlin and Android Studio, basic SQL commands are used to store the database.
Operating System: Windows 10
Software: Kotlin and Java
Emulator: Pixel 4 API 30

HARDWARE

- RAM: 16 GB
- ROM: 20 GB
- Hard Disk – 2 GB.
- Processor – Dual Core or Above.
- Mouse.
- Keyboard.
- Monitor.

4. EXPERIMENTAL INVESTIGATIONS

Android Studio

Android Studio is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer's job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an 'S', it automatically senses everything that starts with an 'S' and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK's, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

SQLite :

SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. It is used as a way of storing user data. We are using SQLite to store the data about the lists of the user along with the list items, priority and the type.

Test Plan and Test Activities

Test plan is necessary for any project to plan the testing phase and decide the scope of the project. Test plan involves collecting design specifications about the project, wiring test cases, executing them manually or automatically using automated testing tools. Testing any application is highly important. Test plan is a method of documenting the test cases, specification plans and other basic level details about how the application works. Test Activities for this project includes various testing like:

- Black Box testing: In this project, sample test cases are written and manual testing is done to check the functionality of the application.
- White Box testing: Once the application meets the user requirements and functionalities according to the test cases, its internal logic are completely tested to ensure that the application does not have any logical errors or issues.
- Unit Testing: I have tested all the modules of the application individually by running as a test program
- Integration testing: After testing the modules individually, tested them by integrating all the sub modules, modules into one application.
- System Testing: It refers to checking whether the system in which the application is built meets the necessary requirements like software support. For example: In this project, I have checked whether the device in which the application developed is compatible with the software (Android Studio)
- End to End Testing: Tested the complete environment of application by connecting the device with different machines, installing as an APK file, with the database and in local network.
- Usability Testing: Finally, usability testing is performed by testing the application’s flow, UI design and how flexible and easy the application is easy to use.

SYSTEM REQUIREMENT SPECIFICATION

Data Requirements

The set of data that is involved in any project is defined using data requirements. For this project, the main data required is the login information to register the application and the item’s information. Without this information the application cannot process the transaction.

Functional Requirements

Functional requirements are properties that must exist in the final system. For any mobile application, we need to download the application from the play store. The application could be either free or paid depending upon the store or merchant. To use the application, the user needs to register and login to the application after installing by providing login information. Once, he or she logins into the application, they can use all the features.

Performance Requirements

Response time, scalability, platform dependencies, tolerance are the performance requirements that should be considered when developing any system. The application or system should be able to respond quickly when the user interacts with the application. The application should be developed in such a way that it should be scalable enough to accept new features when we want to expand the application complexity. The application 5 should run in all the specified software and hardware requirements from the design phase of the project. Also, the tolerance rate (fault tolerance) of the application should be at a higher level in case of network issues, connectivity issues, and when the application crashes or stops. It should be able to deliver the information about any of those issues to the user when the system is no longer able to provide results when the user wants.

System Requirements

The application should be installed into a device, system or any machine in such a way that it should have basic requirements like supporting software and hardware of the device, accessing in-built software, say camera for mobile device, internet permissions, and potential security issues such as virus or any malware detection.

Testing and Maintainability Requirements

The application should be able to meet all the possible good and bad test cases under a test environment. Application should be developed in such a way that it does not have any issues or crashes when the user is using the application. It should be able to extend itself when we expand the code or implement any new functions to the existing application.

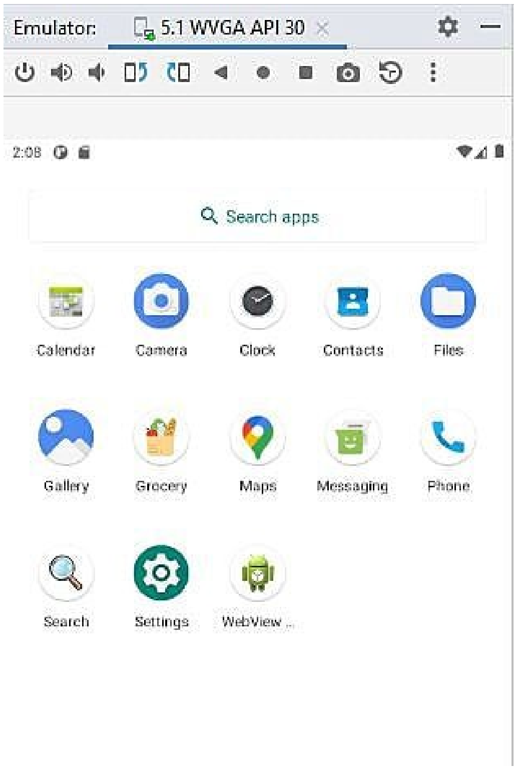
Validation

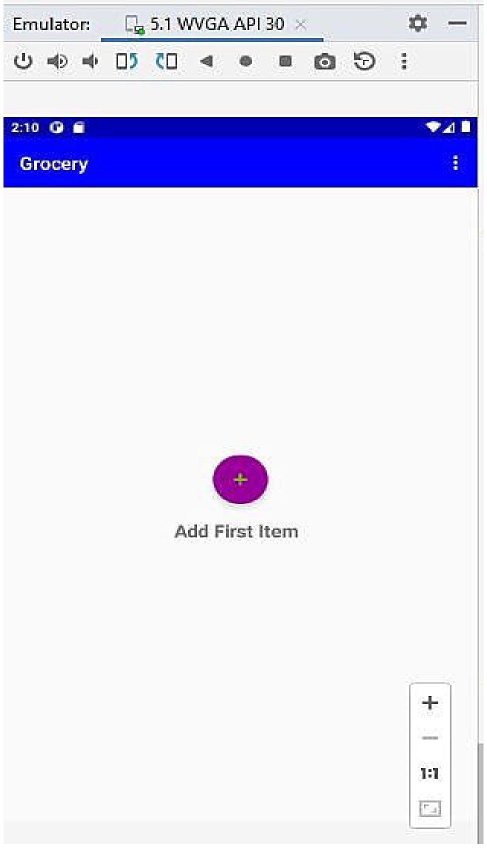
Validating any application is an important criterion before releasing the application to the users. If there is no validation, the information entered by users may be redundant, formatted inappropriately and cannot be maintained

Android Development Tools

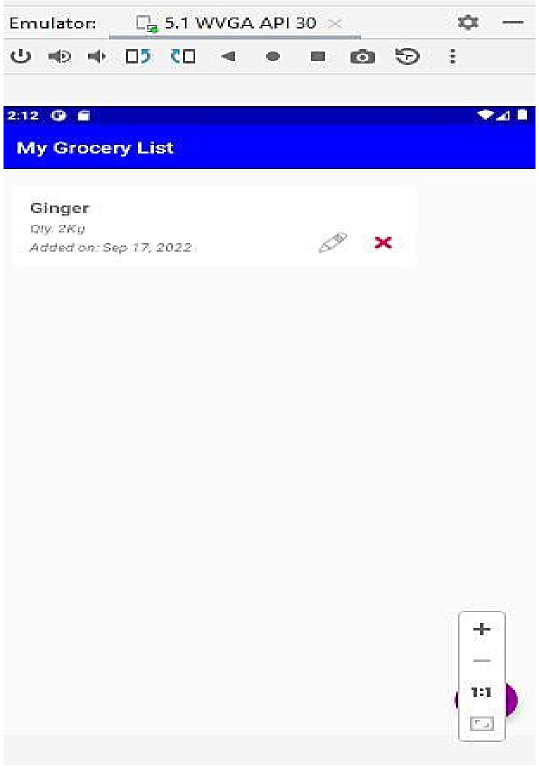
Android is the significant platform for open source mobile applications. It uses Linux kernelbased as default operating system; using Android SDK to build user application in Java. Android is more flexible when it can be developed in both Mac and Windows systems [2]. Using the addon Android SDK, we not only can develop but also can run and test the application on both of the emulators and real Android devices.

Snapshots


Emulator:
5.1 WVGA API 30
2:08
Search apps
Calendar
Camera
Clock
Contacts
Files
Gallery
Grocery
Maps
Messaging
Phone
Search
Settings
WebView ..


Emulator:
5.1 WVGA API 30
2:10
Grocery
+
Add First Item
1:1

z

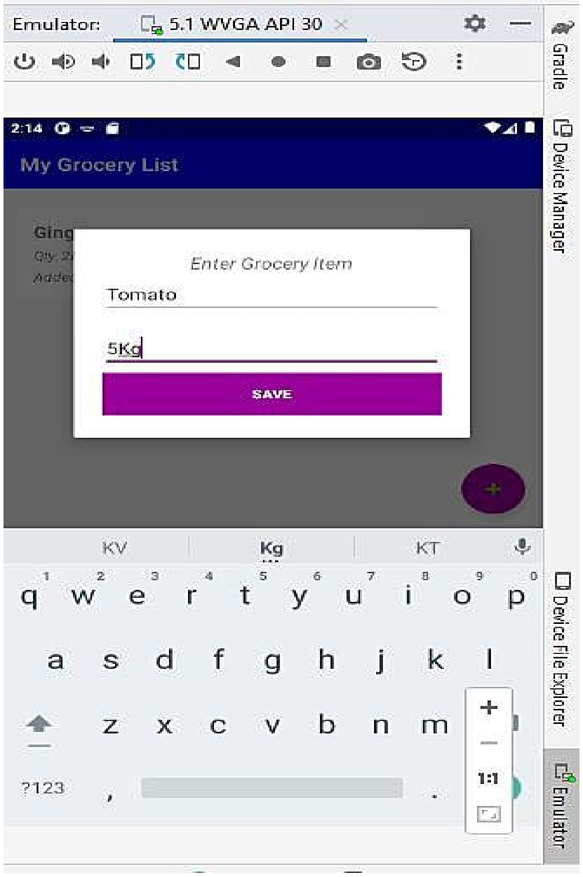


Emulator: 5.1 WVGA API 30

2:12

My Grocery List

Ginger
Qty: 2Kg
Added on: Sep 17, 2022



Emulator: 5.1 WVGA API 30

2:14

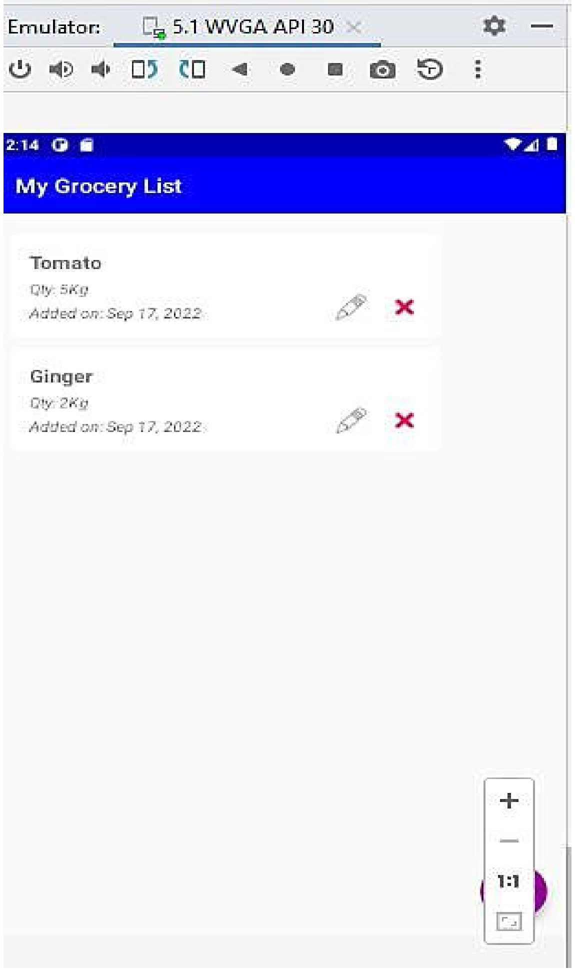
My Grocery List

Enter Grocery Item

Tomato

5Kg

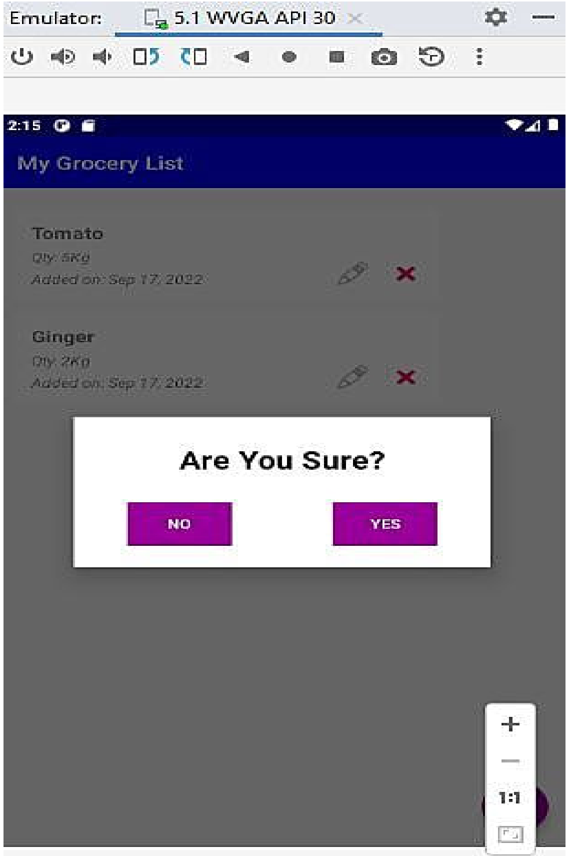
SAVE



My Grocery List

Tomato
Qty: 5Kg
Added on: Sep 17, 2022

Ginger
Qty: 2Kg
Added on: Sep 17, 2022

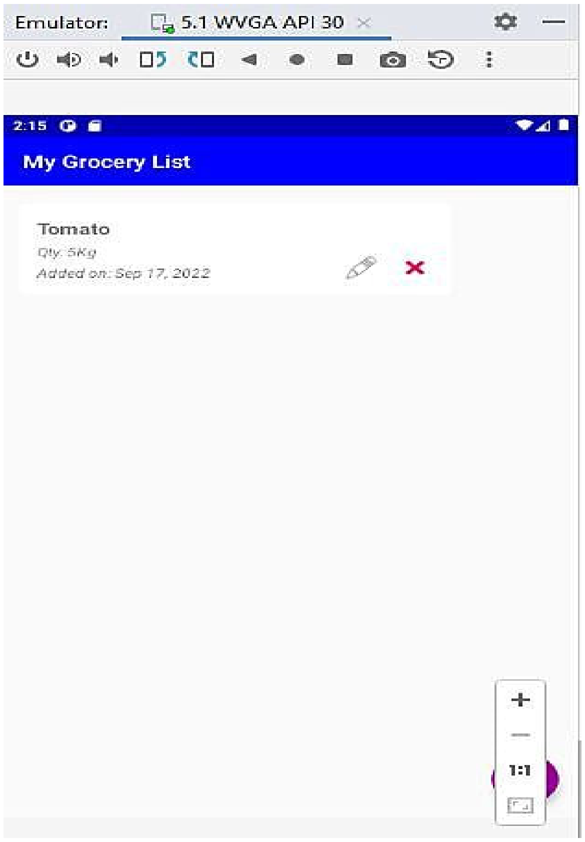


My Grocery List

Tomato
Qty: 5Kg
Added on: Sep 17, 2022

Ginger
Qty: 2Kg
Added on: Sep 17, 2022

Are You Sure?

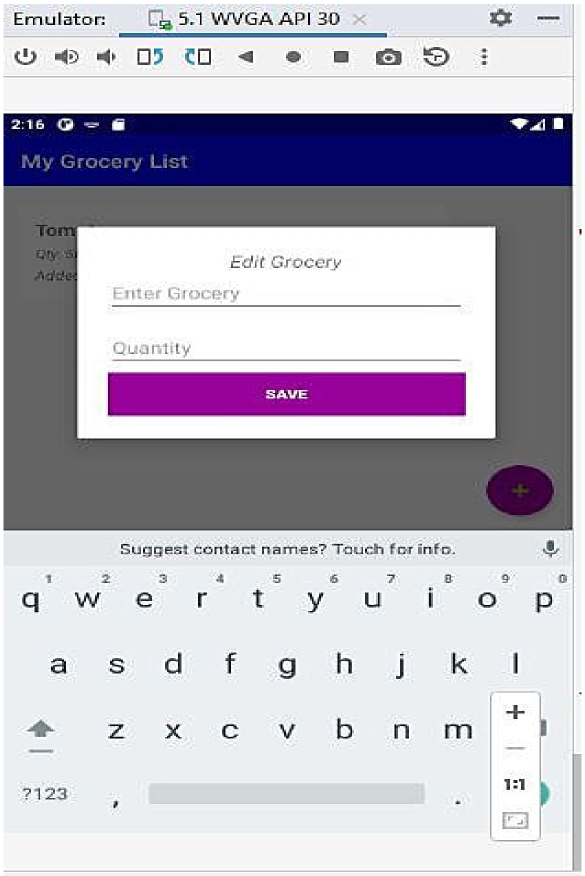


My Grocery List

Tomato

Qty: 5Kg

Added on: Sep 17, 2022



My Grocery List

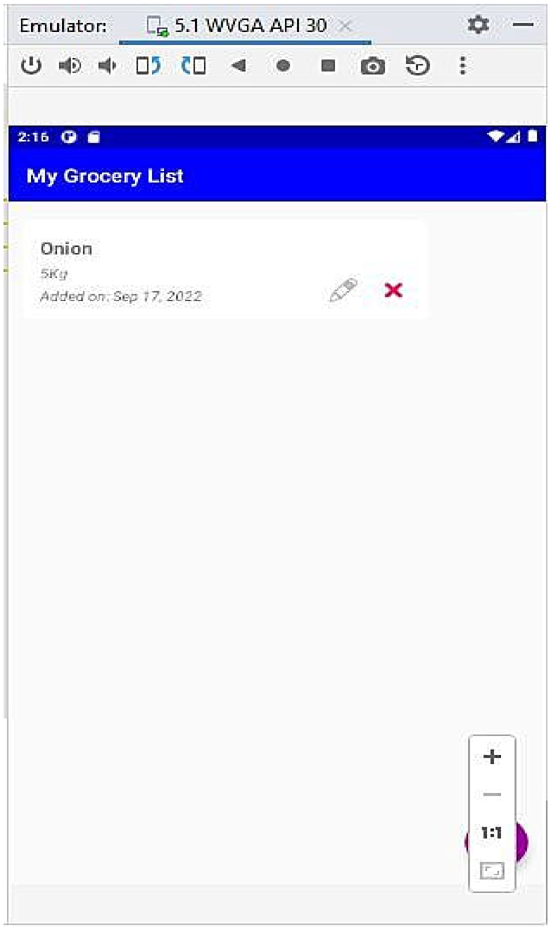
Edit Grocery

Enter Grocery

Quantity

SAVE

Suggest contact names? Touch for info.



My Grocery List

Onion
5Kg
Added on: Sep 17, 2022

Conclusion

I use different applications on my phone every single day, and with the way that these different technologies are revolutionizing society is truly mind blowing. I couldn't even imagine a day where I didn't have a cell phone to check, or an app to play in my free time. This fact alone proves that app development will never be a useless skill to have, because we as a society will only grow to expand its concepts more and more.

I'm very pleased to have taken on this project and to have finished it with results that were above my expectations. Developing apps is a skill that I know will help me become even more versatile when it comes to software development. Learning the basics, successfully implementing learned ideas, and creating a finished product is a process that I will live enjoy for a very long time.

Future Enhancement

Based on the current state of my grocery list app, a future implementation I could create would be to generalize the purpose of the app altogether. For instance, instead of just making the app exclusively for grocery lists, I could broaden the purpose to just general note taking. That way the app could see multiple uses and potentially be more appealing to users. However, if I were to stick with my initial idea, I would want to implement the ability to access databases of all retailers across the United States. Now I know this may not even be possible to implement, but perhaps a simpler version of this idea could be done. I could potentially create my own database that I populate with all different foods from the main food retailers and allow the user to search through that database and manually add the different products they want to their grocery list. I could even save pictures of all the products that could also be included in the grocery list.

BIBILOGRAPHY

1.http://www.vogella.com/tutorials/AndroidSQLite/article.html
2. https://developers.google.com/maps/
3. http://developer.android.com/design/patterns/notifications.html

APPENDIX

A. Source Code

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Activities.DetailsActivity">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="5dp">

    <TextView
        android:id="@+id/itemNameDet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Textview"
        android:textSize="18sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/quantityDet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/itemNameDet"
        android:text="Qty"
        android:textSize="14sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/dateAddedDet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/quantityDet"
        android:text="Date"
        android:textSize="18sp"
```

```
                android:textStyle="bold" />

            <TableRow
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/dateAddedDet"
                android:layout_alignParentRight="true"
                android:layout_marginTop="15dp"
                android:orientation="horizontal">

                <Button
                    android:id="@+id/editButtonDet"
                    android:layout_width="30dp"
                    android:layout_height="30dp"
                    android:layout_marginRight="15dp"
                    android:background="@android:drawable/ic_menu_edit" />

                <Button
                    android:id="@+id/deleteButtonDet"
                    android:layout_width="30dp"
                    android:layout_height="30dp"
                    android:background="@android:drawable/ic_delete" />

            </TableRow>

        </RelativeLayout>


    </android.support.v7.widget.CardView>
</android.support.constraint.ConstraintLayout>
```