VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

A Mobile Application Development Mini-Project Report

On

GROCERY LIST APP

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF DEGREE

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

ADITHI SATISH (1JB19IS003)

Under the Guidance of

Mr. Jeevaraj R

Assistant Professor

Dept. of ISE, SJBIT

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY,

KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2021 – 2022

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Information Science & Engineering

CERTIFICATE

Certified that the Mini-project work entitled "**Grocery List App",** is a bonafide work carried out by **ADITHI SATISH(1JB19IS003),** student of **SJB Institute of Technology,** in partial fulfilment for 6th semester **Mobile Application Development Laboratory with mini project** in **INFORMATION SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, **BELAGAVI** during the academic year **2021-22.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project prescribed for the said degree.

Mr. JEEVARAJ R **Asst. Professor** **Dept. of ISE, SJBIT**	**Dr. REKHA B** **Professor & Head** **Dept. of ISE, SJBIT**

EXTERNAL VIVA

NAME OF THE EXAMINERS	SIGNATURE WITH DATE
1.	
2.	



ACKNOWLEDGEMENT

I would like to express my profound gratitude to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha MahaSwamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha MahaSwamiji** for providing me an opportunity to complete my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, **Managing Director**, SJB Institute of Technology, for his continuous support in providing amenities to carry out this project in this admired institution.

I express my gratitude to **Dr. K V Mahendra Prashanth**, **Principal**, SJB Institute of Technology, for providing me an excellent facilities and academic ambience; which has helped me in satisfactory completion of project work.

I extend my sincere thanks to **Dr. Rekha B**, **Professor & Head**, Department of Information Science and Engineering; for providing me an invaluable support throughout the period of my project work.

I wish to express my heartfelt gratitude to the **mini-project coordinator, Mr. Jeevaraj R, Asst. Prof,** Department of Information Science and Engineering for his valuable guidance, suggestions and cheerful encouragement during the entire period of my project work.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents,Teaching & Non-teaching staff of the department, and all my friends, who have directly or indirectly supported me during the period of my project work.

ADITHI SATISH(1JB19IS003)

ABSTRACT

Grocery list app is a product application that is intended for the users to list all the grocery items they would require at any given moment. It provides an easier interface for the users to list the required items. It also gives provision to mention the amount of item required. So this is a much important thing of life must be very easily accessible and easily provided to the customer. here if we talk about getting it online, we have a system that can maintain the whole objects of grocery in a very classified manner and make it easy to list out the products fast. It also gives an option to modify the items and the amount of product required which is very convenient for the usage.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 What is Mobile App Development?

An application is software that is used to accomplish specific requirements of user. For eg. To read PDFfiles,creating documents, gaming applications, applications to play audio and video files etc. For all these typesof different requirements different applications needs to be developed. Application development is the processof designing, building, and implementing software applications. It can be done by massive organizations with large teams working on projects, or by a single freelance developer. Application development defines the process of how the application is made, and generally follows a standard methodology.

Mobile application development is the process to making software for smartphones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used forthis kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transactions and access information. Today, mobile devices—and the mobileapplications that unlock their value—are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications thattheir customers, partners and employee's demand.

Yet mobile application development might seem daunting. Once you've selected the OS platform or platforms, you need to overcome the limitations of mobile devices and usher your app all the way past the potential hurdles of distribution.

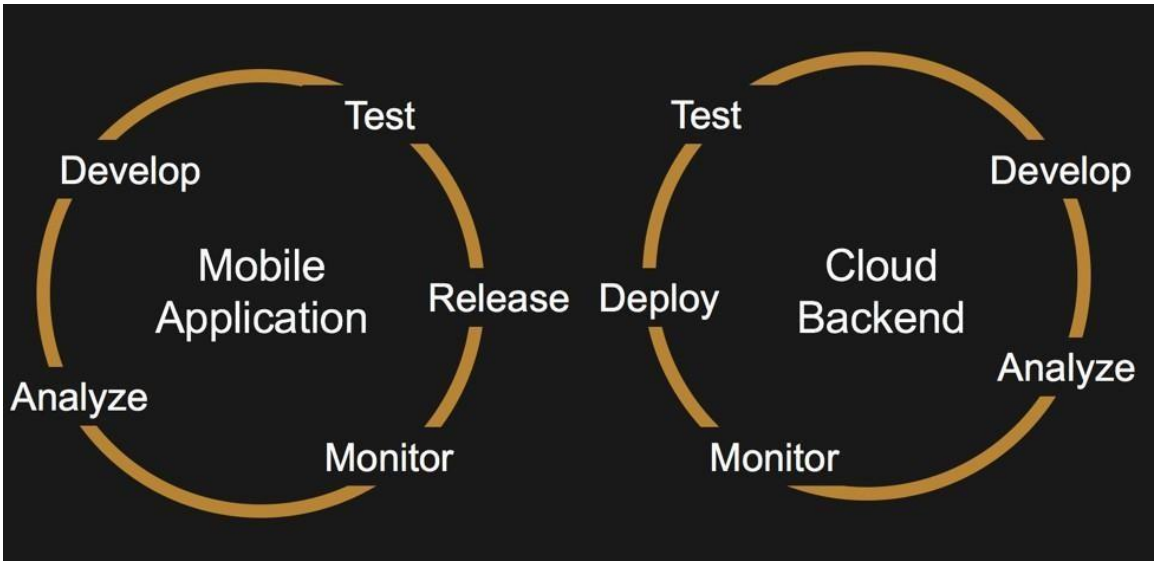


Figure 1.1 Mobile Application Development Lifecycle

1.2 Grocery List App

Having a well-planned grocery list gets you in and out of the store quickly and helps you stick to your healthy eating plan.

Use these tips and in just a few minutes, you'll have a blueprint for a cart full of groceries that won't bust your budget or diet.

Organize your grocery shopping list by aisle. Follow these tips for filling that list with the healthiest foods from each aisle.

1.3 Project Structure

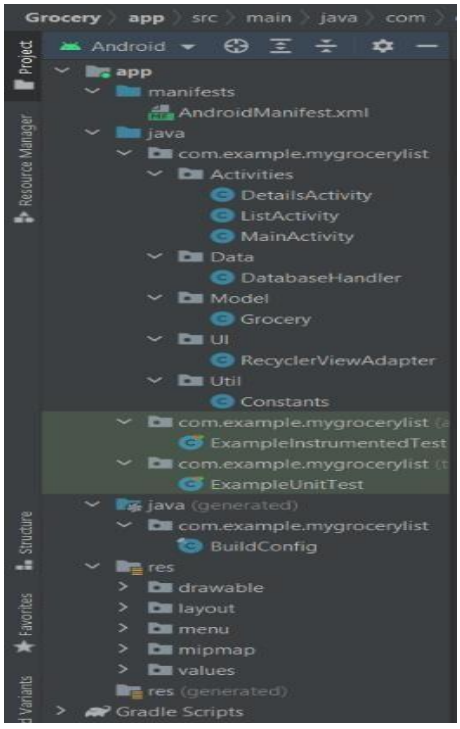


Fig.1.2: Project window

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

- **manifests**: Contains the AndroidManifest.xml file.
- **java**: Contains the Java source code files, including JUnit test code.
- **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it's showing as **Android**).

You can also customize the view of the project files to focus on specific aspects of your Development.

Chapter 2

LITERATURE SURVEY

Android Studio is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.

Android Studio was announced on 16th May 2013 at the Google I/O conference as an official IDE for Android app development. It started its early access preview from version 0.1 in May 2013. The first stable builtversion was released in December 2014, starts from version 1.0. Since 7th May 2019, Kotlin is Google's preferred language for Android application development. Besides this, other programming languages such as Java are supported by Android Studio.

2.1 Android Studio

Android Studio is an integrated development environment (IDE) for developing for the Android platform. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

2.2 Android Architecture

We studied the Android system architecture. Android system is a Linux-based system, Use of the software stack architecture design patterns . As shown in Figure 1, the Android architecture consists of four layers: Linux kernel, Libraries and Android runtime, Application framework and Applications [5-8]. Each layer of the lower encapsulation, while providing call interface to the upper.

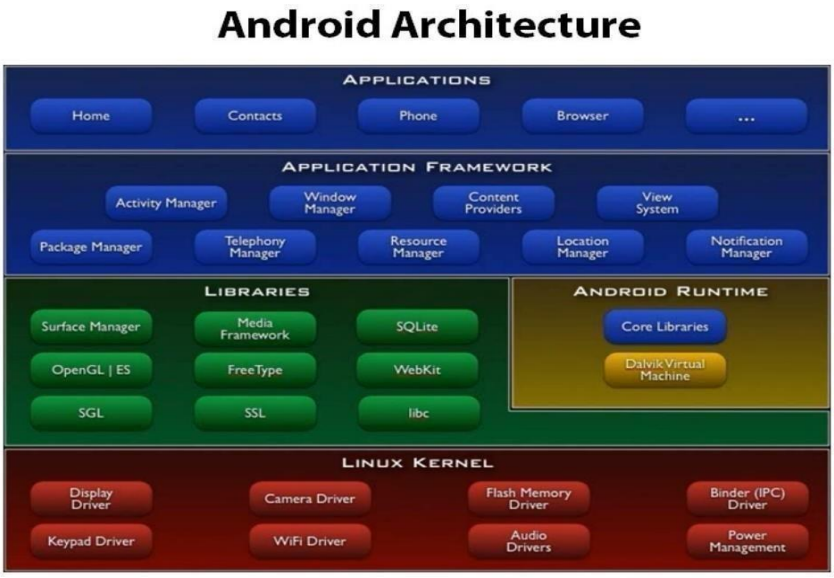


Figure-2.1: Android Architecture

2.3 Methodology

This project is made by using Android studio, Virtual emulator and Photoshop. The programming languages used for building the application are Java and XML .User interface is handled using XML codes. Backend programming is handled mainly through set of java codes.

The other libraries required are:

- Android SDK tools
- Android SDK build tools
- SDK platform
- Google APIs
- Google APIs ARM EABI v7a System Image
- GPU Debugging tools
- Android Support Library
- Google USB drivers
- Google Web drivers

2.4 Features of Android Studio

- It has a flexible Gradle-based build system.
- It has a fast and feature-rich emulator for app testing.
- Android Studio has a consolidated environment where we can develop for all Android devices.
- Apply changes to the resource code of our running app without restarting the app.
- Android Studio provides extensive testing tools and frameworks.
- It supports Java, C++ and NDK.
- It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google CloudMessaging and App Engine.

2.5 Android Studio Project Structure

The Android Studio project contains one or more modules with resource files and source code files.

These include different types of modules-

2.4.1 Android app modules

2.4.2 Library modules

2.4.3 Google App Engine modules

By default, Android Studio displays our project files in the Android project view, as shown in the below

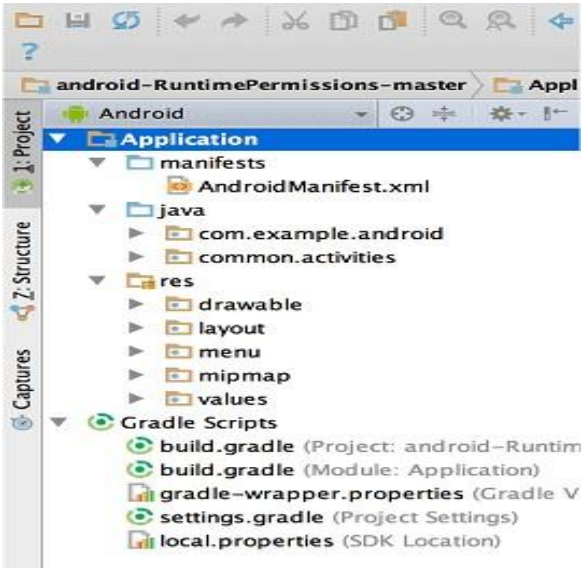


Figure 2.2 Android Project Structure

These build files are visible to the top-level under Gradle Scripts. And the app module contains the following folders:

2.4.4 manifests: It contains the AndroidManifest.xml file.

2.4.5 java: It contains the source code of Java files, including the JUnit test code.

2.4.6 res: It contains all non-code resources, UI strings, XML layouts, and bitmap images.

We will see the actual file structure of the project by selecting the Project from the Project dropdown.

2.6 Android Studio User Interface

The Android Studio main window contains the several logical areas which are shown in the below Figure 2.3:

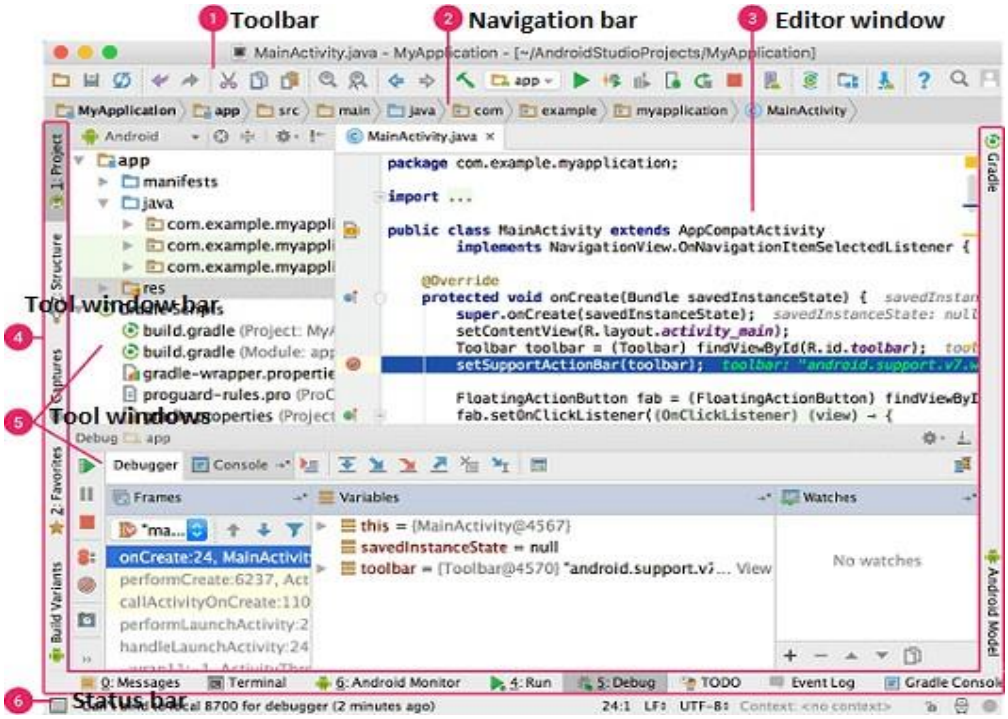


Figure 2.3 Android Studio Main Window

The **toolbar** provides us a wide range of actions, which includes running apps and launching Androidtools.

i. The **navigation bar** helps in navigating our project and open files for editing. It gives a compact viewof structure visible in the Project window.

ii. The **editor window** is a space where we can create and modify our code. On the basis of the current filetype, the editor can change. While viewing a layout file, the editor displays the Layout Editor.

iii. The **tool window bar** runs around the outside the IDE window and contains buttons that allow as toexpand and collapse individual tool windows.

iv. The **tool windows** provide us access specific tasks like search, project management, version control,and more. We can able expand and collapse them.

v. The **status bar** displays the status of our project and IDE itself, as well as any messages or warnings.

2.7 Gradle Build System

Gradle build used as the foundation of the build system in Android Studio. It uses more Android-specificcapabilities provided by the Android plugin for Gradle. This build system runs independently from the commandline and integrated tool from the Android Studio menu. We can use build features for the following purpose:

2.6.1 Configure, customize, and extend the build process.

2.6.2 We can create multiple APKs from our app, with different features using the same project and modules.

2.6.3 Reuse resource and code across source sets.

2.8 Install and Setup Android Studio and Java JDK

i.JDK installation

- **Step 1**: Download the JDK by using below link according to your Operating system. https://www.oracle.com/java/technologies/javase-jdk16-downloads.html
- **Step 2**: Install the downloaded JDK in PC.
- **Step 3**: After Installing, Right Click on This PC or My Computer on Desktop
- **Step 4**: Click on Properties
- **Step 5**: Click on Advance System Settings
- **Step 6**: Click on Environment Variables in the bottom
- **Step 7**: Double Click on path in the second dialog box

- **Step 8**: Click on New to add path
- **Step 9**: Go to JDK path in C Drive
- **Step 10**: Copy and paste the JDK path in system settings
- **Step 11**: Go to JRE path in C drive
- **Step 12**: Copy and paste the JRE path in system settings
- **Step 13**: For verification, Go to Command Prompt and press javac command. If it is getting executed, then you have successfully installed the Java JDK.

ii.Android Studio Setup

- **Step 1**: Open Android Studio
- **Step 2**: Click on create new project
- **Step 3**: Select Empty Activity
- **Step 4**: You can name your application of your own choice in Name field and you can select your language either Java/Kotlin, and you can also select the Minimum SDK- The lower the version you select as a developer, your app will run approx. in all the existing devices. Then press finish to start programming.
- **Step 5**: It takes time to build gradle which depends on your Internet connectivity and system configuration. (Approx. 1-3 minutes)
- **Step 6**: After gradle building, you will get the Android Studio Main Window.
- **Step 7**: Run the program by clicking on Run command.
- **Step 8**: Output will be showed in Virtual Emulator as your mobile

Chapter 3

SYSTEM REQUIREMENTS

Requirements analysis is critical for project development. Requirements must be documented, actionable, measurable, testable and defined to a level of detail sufficient for system design. Requirements canbe architectural, structural, behavioural, functional, and non-functional. A software requirements specification (SRS) is a comprehensive description of the intended purpose and the environment for software under development.

3.1 Hardware Requirements

- Minimum of 8GB RAM for efficient working
- 2 GB of available disk space minimum, 4 GB recommended
- Preferably SSD (Solid state drive) instead of HDD
- Minimum 512MB graphics card for better performance of virtual emulator
- 1280 x 800 minimum screen resolution
- Keyboard
- Mouse
- Display Unit
- Dual-Core or AMD with minimum of 1.5GHz speed

3.2 Software Requirement

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- Android Studio Bundle
- Java JDK

Chapter 4

DESIGN AND IMPEMENTATION

4.1 System Design

Once the app is installed in the android mobile phone. the user will be able to see the home screen whichhas a button add item which will allow the user to add the grocery item to the list along with the quantity .There is a button to save the entered grocery item. There are buttons to edit the grocery items and the quantity of the item and also allows you to delete the item from the list.

.

4.2 Design Using XML

4.2.1 Main activity

The below code represents the home page of the app.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout 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.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>
```

```
    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_input_add" />

</android.support.design.widget.CoordinatorLayout>
```

4.2.2 Item list

It shows all the grocery items you have added to your list.

```
    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_list" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_input_add" />

</android.support.design.widget.CoordinatorLayout>
```

4.2.3 Main Activity onCreate function snippet

```
public class MainActivity extends AppCompatActivity {
    private AlertDialog.Builder dialogBuilder;
    private AlertDialog dialog;
    private EditText groceryItem;
    private EditText quantity;
    private Button saveButton;
    private DatabaseHandler db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        db = new DatabaseHandler( context: this);
        byPassActivity();

        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {

            createPopupDialog();
        });
    }
```

4.2.4 Main activity 2

Function to handle the functionality of the pop up dialog

```
private void createPopupDialog() {

    dialogBuilder = new AlertDialog.Builder( context: this);
    View view = getLayoutInflater().inflate(R.layout.popup,  root: null);
    groceryItem = (EditText) view.findViewById(R.id.groceryItem);
    quantity = (EditText) view.findViewById(R.id.groceryQty);
    saveButton = (Button) view.findViewById(R.id.saveButton);
    dialogBuilder.setView(view);
    dialog = dialogBuilder.create();
    dialog.show();
    saveButton.setOnClickListener((v) -> {
            if (!groceryItem.getText().toString().isEmpty()
                    && !quantity.getText().toString().isEmpty()) {
                saveGroceryToDB(v);
            }
    });
}
```

4.2.5 detailsActivity.java

```
package com.example.mygrocerylist.Activities;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import com.example.mygrocerylist.R;

public class DetailsActivity extends AppCompatActivity {
    private TextView itemName;
    private TextView quantity;
    private TextView dateAdded;
    private int groceryId;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);
        itemName = (TextView) findViewById(R.id.itemNameDet);
        quantity = (TextView) findViewById(R.id.quantityDet);
        dateAdded = (TextView) findViewById(R.id.dateAddedDet);
        Bundle bundle = getIntent().getExtras();
        if ( bundle != null ) {
            itemName.setText(bundle.getString( key: "name"));
            quantity.setText(bundle.getString( key: "quantity"));
            dateAdded.setText(bundle.getString( key: "date"));
            groceryId = bundle.getInt( key: "id");
        }
    }
}
```

4.2.6 Grocery class

```
package com.example.mygrocerylist.Model;

public class Grocery {
    private String name;
    private String quantity;
    private String dateItemAdded;
    private int id;

    public Grocery() {
    }
    public Grocery(String name, String quantity, String dateItemAdded, int id) {
        this.name = name;
        this.quantity = quantity;
        this.dateItemAdded = dateItemAdded;
        this.id = id;
    }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getQuantity() { return quantity; }
    public void setQuantity(String quantity) { this.quantity = quantity; }
    public String getDateItemAdded() { return dateItemAdded; }
    public void setDateItemAdded(String dateItemAdded) { this.dateItemAdded = dateItemAdded; }
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
}
```

4.2.7 Confirmation dialog 1

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

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="23dp">
            <TextView
                android:id="@+id/textAlert"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:paddingTop="5dp"
                android:text="Are You Sure?"
                android:textSize="17sp"
                android:textStyle="bold" />
            <Button
```

Confirmation dialog 2

```
            <Button
                android:id="@+id/noButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/textAlert"
                android:layout_marginTop="19dp"
                android:background="@color/colorAccent"
                android:text="No"
                android:textColor="@android:color/white" />
            <Button
                android:id="@+id/yesButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/textAlert"
                android:layout_alignParentRight="true"
                android:layout_marginTop="19dp"
                android:background="@color/colorAccent"
                android:text="Yes"
                android:textColor="@android:color/white" />
        </RelativeLayout>
    </android.support.v7.widget.CardView>
</android.support.constraint.ConstraintLayout>
```

4.2.8 Function to save item to database

```
private void saveGroceryToDB(View v) {
    Grocery grocery = new Grocery();
    String newGrocery = groceryItem.getText().toString();
    String newGroceryQuantity = quantity.getText().toString();
    grocery.setName(newGrocery);
    grocery.setQuantity(newGroceryQuantity);
    //Save to DB
    db.addGrocery(grocery);
    Snackbar.make(v, text: "Item Saved!", Snackbar.LENGTH_LONG).show();
    new Handler().postDelayed(() -> {
            dialog.dismiss();
            //start a new activity
            startActivity(new Intent( packageContext: MainActivity.this, ListActivity.class));
    }, delayMillis: 1200); //  1 second.

}
```

4.2.9 Function to get items from the database

```
public Grocery getGrocery(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.query(Constants.TABLE_NAME, new String[]{Constants.KEY_ID,
                    Constants.KEY_GROCERY_ITEM, Constants.KEY_QTY_NUMBER, Constants.KEY_DATE_NAME},
            selection: Constants.KEY_ID + "=?",
            new String[]{String.valueOf(id)}, groupBy: null, having: null, orderBy: null, limit: null);
    if (cursor != null)
        cursor.moveToFirst();
    Grocery grocery = new Grocery();
    grocery.setId(Integer.parseInt(cursor.getString(cursor.getColumnIndex(Constants.KEY_ID))));
    grocery.setName(cursor.getString(cursor.getColumnIndex(Constants.KEY_GROCERY_ITEM)));
    grocery.setQuantity(cursor.getString(cursor.getColumnIndex(Constants.KEY_QTY_NUMBER)));
    java.text.DateFormat dateFormat = java.text.DateFormat.getDateInstance();
    String formatedDate = dateFormat.format(new Date(cursor.getLong(cursor.getColumnIndex
            (Constants.KEY_DATE_NAME)))
            .getTime());
    grocery.setDateItemAdded(formatedDate);
    return grocery;
}
```

```
public void deleteGrocery(int id){
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(Constants.TABLE_NAME, whereClause: Constants.KEY_ID + " = ?",
            new String[] {String.valueOf(id)});

    db.close();

}
//Get Count
public int getGroceriesCount(){
    String countQuery = "SELECT * FROM " + Constants.TABLE_NAME;
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.rawQuery(countQuery, selectionArgs: null);

    return cursor.getCount();

}
```

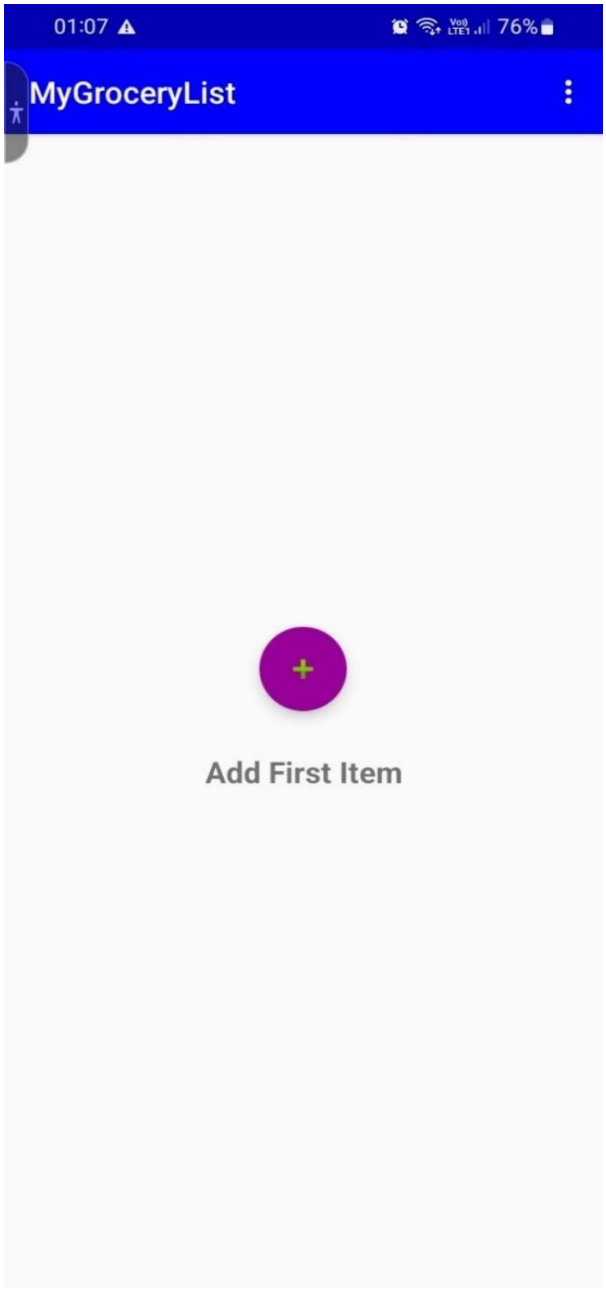
4.2.10 Constants

```java
package com.example.mygrocerylist.Util;

public class Constants {
    public static final int DB_VERSION = 1;
    public static final String DB_NAME = "groceryListDB";
    public static final String TABLE_NAME = "groceryTBL";
    public static final String KEY_ID = "id";
    public static final String KEY_GROCERY_ITEM = "grocery_item";
    public static final String KEY_QTY_NUMBER = "quantity_number";
    public static final String KEY_DATE_NAME = "date_added";
}
```

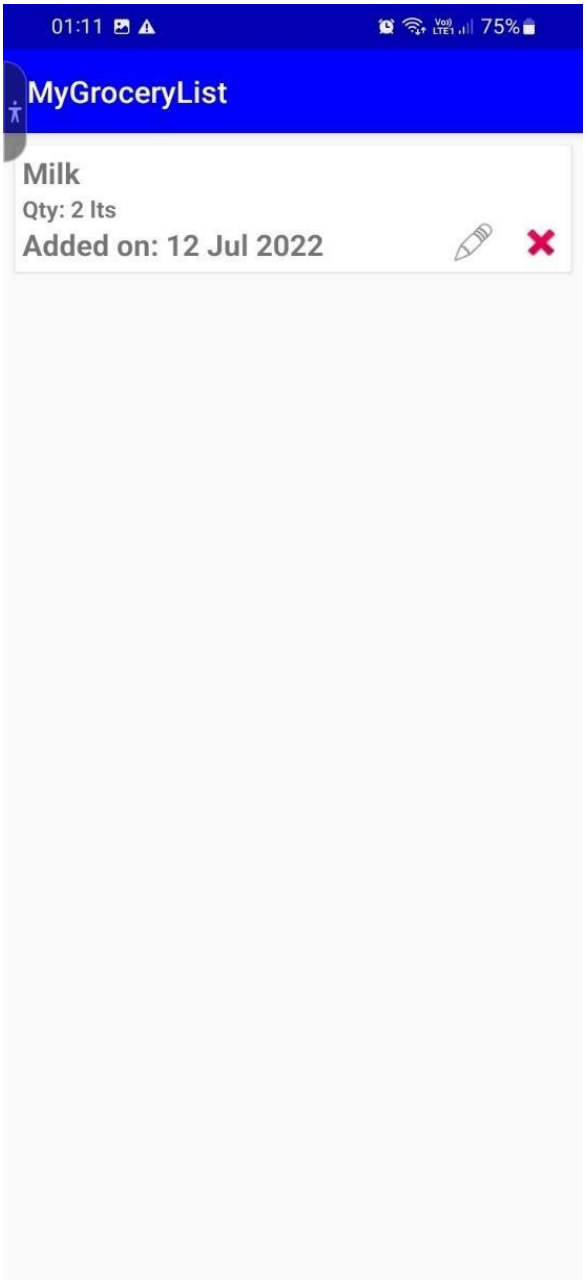
Chapter 5

SNAPSHOTS



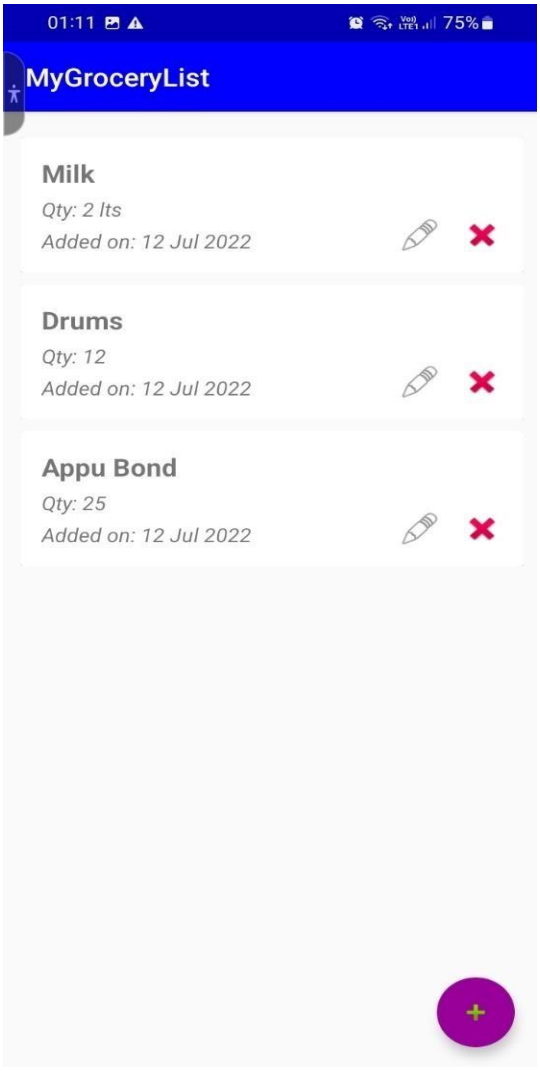
5.1 Home page

The home page of the application contains a button to add items to the list



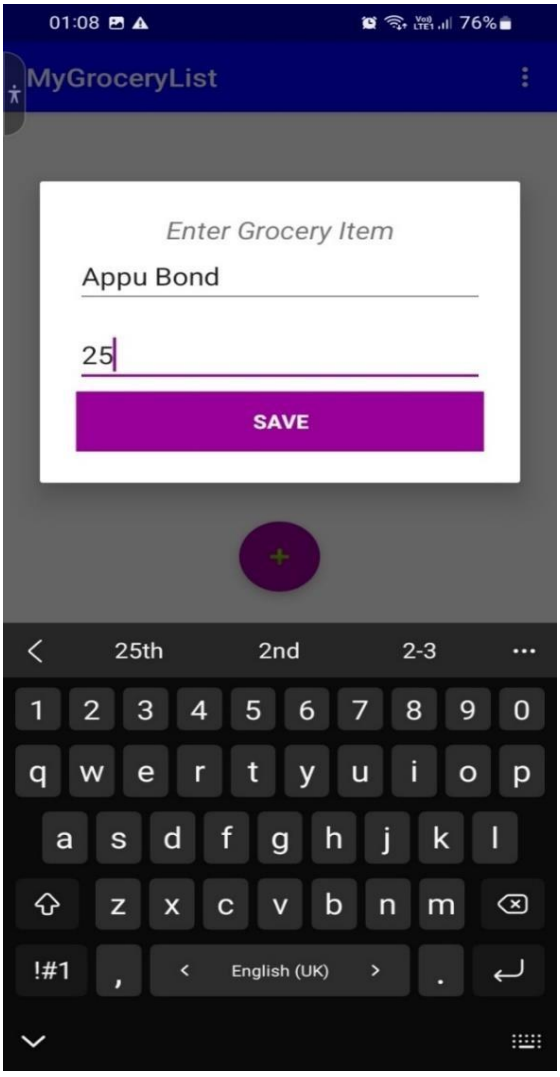
5.2 Individual items

This page displays the individual items present in the list.



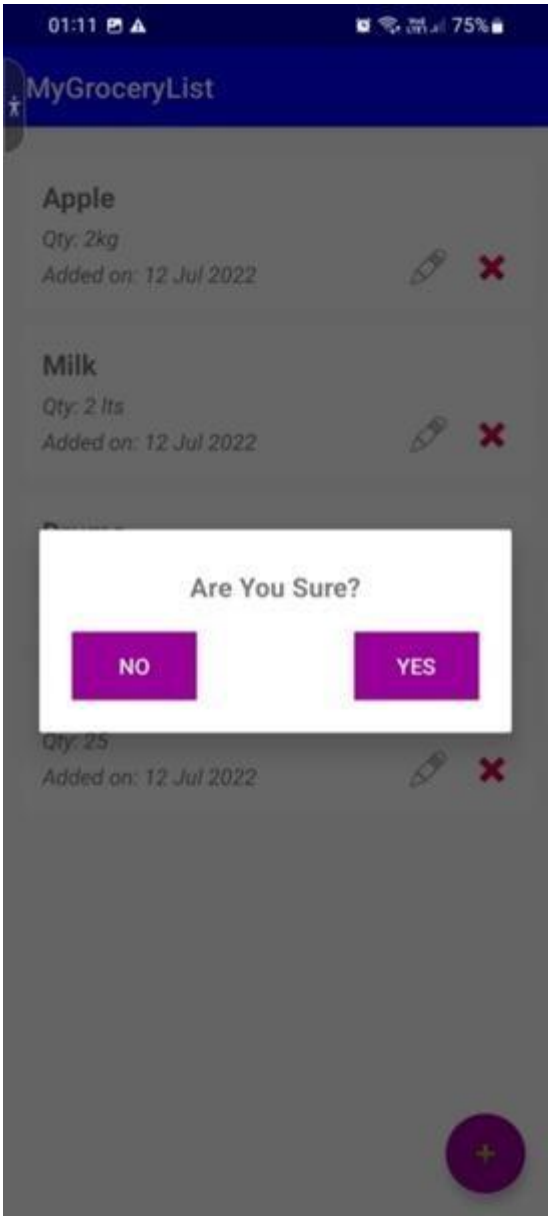
5.3 List of items

This screen displays the items added to the list, the user can edit the items by clicking on the edit button.



5.4 Adding item

This screen displays how the user can add items to the list.



5.5 Deleting item

The delete button lets the user delete the item from the list by clicking on the delete button, the application asks for a final confirmation before the item is deleted.

CONCLUSION

The system is highly scalable and user friendly. Almost all the system objectives have beenmet. The system has been tested under all criteria. The design of the database is flexible ensuring that the systemcan be implemented. It is implemented and gone through all validation. All phases of development were conceived using methodologies. User with little training can get the required report

FUTURE ENHANCEMENT

- Further in the future the grocery store app can be updated to never user requirements and feedback. The Code has been designed in such an organized and disconnected manner than adding new features to existing features is an easy task.
- The app can be improved more to function more efficiently and effectively.
- A real time price of the items can be included by providing a signup feature for admin registration.
- The feature of mail can be included to mail the list produced to the shopkeeper.

BIBLIOGRAPHY

[i] https://developer.android.com/studio

[ii] https://www.researchgate.net/

[iii] https://www.javatpoint.com/android-tutorial

[iv] https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/

[v] Android Programming: Pushing the Limits- Book by Erik Hellman

[vi] Head First Android Development, 3rd Edition- by Dawn Griffiths, David Griffiths

[vii] Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)- by Bill Phillip