KITCHENETTE RECIPES

Technical Manual

Ailish Kavanagh
C00206130@itcarlow.ie
Supervisor: Paul Barry
Paul.barry@itcarlow.ie

Abstract

The purpose of this project is to create an Android application where the user can mark foods that they have in the house and based on these foods the app should suggest recipes they can cook. The application should present the user the option to add their own custom ingredients and recipes. If a user selects, they have cooked a certain recipe, the app should automatically update the contents list to remove the ingredients they used. If a user wants to make a recipe, but does not have all the necessary ingredients, the app should present the user with a "shopping list" of ingredients to buy.

Table of Contents

Section 1 - Introduction

The following document contains all code for the application. The Kitchenette application can be run on any Android device or Android Virtual Machine. To install the application at present, Android Studio or a similar tool would be required.

The Kotlin code for this document is broken down into the sections of database files, List Adapter code and Activity classes. Some code is omitted from this document if it is auto generated or repetitive. To see the full extent of the code please see the GitHub account linked below.

The XML code is broken down into sections of content layout, list adapters, and navigation units. A lot of the auto generated XML code for Android, though necessary, can be very repetitive and as such is not included in this document. Please see the GitHub account for all other source code.

Finally, this document will include the Gradle dependency files and the Android Manifest. The generated JSON Firebase file is not included in this document as that is entirely created and downloaded by the Firebase platform itself and just inserted into the code.

All code can be found at the GitHub link below:
https://github.com/kavanagha/Kitchenette

Section 3 - Kotlin Code

Section 3.1 – Database Files

3.1.2 – Diet.kt

File - C:\Users\ailis\AndroidStudioProjects\Kitchenette_v2\app\src\main\java\com\kitchenette\kitchenette\Diet.kt

```
 1 package com.kitchenette.kitchenette
 2
 3 class Diet{
 4     var id: Int = 0
 5     var name: String = ""
 6     var recipeID: Long = 0
 7
 8
 9     constructor(name:String, recipeID:Long){
10         this.name = name
11         this.recipeID=recipeID
12     }
13
14     constructor(){}
15 }
```

3.1.2 – Food.kt

File - C:\Users\ailis\AndroidStudioProjects\Kitchenette_v2\app\src\main\java\com\kitchenette\kitchenette\Food.kt

```
 1 package com.kitchenette.kitchenette
 2
 3 import android.graphics.Bitmap
 4
 5 class Food {
 6     var id: Int = 0
 7     var name: String = ""
 8     var category: String = ""
 9     var cupboard: Int = 0 //bool
10     var favourite: Int = 0  //bool
11     var shoppingList: Int = 0   //bool
12     var quantity: Double = 0.0
13     var measurement: String = ""
14     var bought: Int = 0 //bool
15     var photo: Bitmap? = null
16
17     constructor(name:String, category:String, photo :
   Bitmap){
18         this.name = name
19         this.category = category
20         this.photo = photo
21     }
22
23     constructor()
24
25 }
```

3.1.3 – Barcode.kt

```
class Barcodes{
    var id:Int =0
    var barcode:String = ""
    var type:String = ""
    var foodID: Int? = null
    var brand:String = ""
    var quantity:Double = 0.0
    var measurement: String = ""

    constructor(barcode:String){
        this.barcode= barcode
    }

    constructor(){}

}
```

3.1.4 – Recipe.kt

```
class Recipe{
    var id: Int = 0
    var name: String = ""
    var mealType: String = ""
    var cuisine: String = ""
    //var diet: Boolean = false
    var description: String = ""
    var method:String = ""
    var favourite: Int = 0
    var photo: Bitmap? = null
    var servings: Int = 0


    constructor(name:String, mealType:String, cuisine:String, servings:Int,
                description:String, method:String, photo:Bitmap){
        this.name = name
        this.mealType = mealType
        this.cuisine = cuisine
       // this.diet = diet
        this.description = description
        this.method = method
      //  this.favourite = favourite
        this.servings = servings
        this.photo = photo
    }

    constructor(){}
}
```

3.1.5 – Ingredients.kt

File - C:\Users\ailis\AndroidStudioProjects\Kitchenette_v2\app\src\main\java\com\kitchenette\kitchenette\Ingredients.kt

```
 1 package com.kitchenette.kitchenette
 2
 3 class Ingredients{
 4     var id: Int = 0
 5     var recipeID : Long = 0
 6     var foodID : Int = 0
 7     var quantity: Double = 0.0
 8     var measurement: String = ""
 9
10     constructor(recipeID:Long, foodID:Int, quantity:Double
   , measurement:String){
11         this.recipeID=recipeID
12         this.foodID=foodID
13         this.quantity=quantity
14         this.measurement=measurement
15     }
16
17     constructor(){}
18 }
```

3.1.6 – DatabaseHandler.kt

```
package com.kitchenette.kitchenette

import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.graphics.BitmapFactory
import android.widget.Toast
import com.readystatesoftware.sqliteasset.SQLiteAssetHelper
import java.util.ArrayList
import android.graphics.Bitmap
import java.io.ByteArrayOutputStream

const val DATABASE_NAME = "kitchenette.db"

const val TABLE_FOOD = "food"
const val COL_FOOD_ID = "id"
const val COL_FOOD_NAME = "name"
const val COL_FOOD_CATEGORY = "category"
const val COL_FOOD_CUPBOARD = "cupboard"
const val COL_FOOD_FAVOURITE = "favourite"
const val COL_FOOD_SHOPPING = "shoppingList"
const val COL_FOOD_QUANTITY = "quantity"
const val COL_FOOD_MEASUREMENT = "measurement"
const val COL_FOOD_BOUGHT = "bought"
const val COL_FOOD_PHOTO = "photo"

const val TABLE_BARCODE = "barcodes"
const val COL_BARCODE_ID = "id"
const val COL_BARCODE_BARCODE = "barcode"
const val COL_BARCODE_SCANNED = "scanned"
const val COL_BARCODE_TYPE = "type"
const val COL_BARCODE_FOODID = "foodID"
const val COL_BARCODE_BRAND = "brand"
const val COL_BARCODE_QUANTITY = "quantity"
const val COL_BARCODE_MEASUREMENT = "measurement"

const val TABLE_RECIPE = "recipes"
const val COL_RECIPE_ID = "id"
const val COL_RECIPE_NAME = "name"
const val COL_RECIPE_MEAL = "mealType"
const val COL_RECIPE_CUISINE = "cuisine"
const val COL_RECIPE_DESCRIPTION = "description"
const val COL_RECIPE_METHOD = "method"
const val COL_RECIPE_FAVOURITE = "favourite"
const val COL_RECIPE_PHOTO =  "photo"
const val COL_RECIPE_SERVINGS = "servings"

const val TABLE_INGREDIENT = "ingredients"
const val COL_INGREDIENT_ID = "id"
```

```
const val COL_INGREDIENT_RECIPE = "recipeID"
const val COL_INGREDIENT_FOOD = "foodID"
const val COL_INGREDIENT_QUANTITY = "quantity"
const val COL_INGREDIENT_MEASUREMENT = "measurement"

const val TABLE_DIET = "diet"
const val COL_DIET_ID = "id"
const val COL_DIET_NAME = "name"
const val COL_DIET_RECIPE = "recipeID"

class DataBaseHandler (var context: Context) : SQLiteAssetHelper(context, DATABASE_NAME, null, 1){

   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {    }

   /****************** FOOD TABLE ******************/

   fun insertFood(food: Food) : Long? {
      val db = this.writableDatabase

      val bos = ByteArrayOutputStream()
      food.photo?.compress(Bitmap.CompressFormat.JPEG, 10, bos)
      val bArray = bos.toByteArray()

      val cv = ContentValues()
      cv.put(COL_FOOD_NAME, food.name)
      cv.put(COL_FOOD_CATEGORY, food.category)
      cv.put(COL_FOOD_PHOTO, bArray)
      cv.put(COL_FOOD_MEASUREMENT, "kg")

      val result = db.insert(TABLE_FOOD,null,cv)
      return if(result == (-1).toLong()) {
         Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
         null
      } else {
         Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
         result
      }
   }
   fun editFood(foodName:String, category : String, bitmap: Bitmap, id:Int){
      val db = this.writableDatabase

      val bos = ByteArrayOutputStream()
      bitmap.compress(Bitmap.CompressFormat.JPEG, 30, bos)
      val bArray = bos.toByteArray()

      val cv = ContentValues()
      cv.put(COL_FOOD_NAME, foodName)
      cv.put(COL_FOOD_CATEGORY, category)
      cv.put(COL_FOOD_PHOTO, bArray)

      val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
```

```
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
fun readFoodData() : MutableList<Food>{
    val list : MutableList<Food> = ArrayList()

    val db = this.readableDatabase
    val query = "SELECT * FROM $TABLE_FOOD ORDER BY $COL_FOOD_NAME"
    val result = db.rawQuery(query, null)
    if(result.moveToFirst()){
        do {
            val food = Food()
            food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
            food.name = result.getString(result.getColumnIndex(COL_FOOD_NAME))
            list.add(food)
        }while (result.moveToNext())
    }
    result.close()
    db.close()
    return list
}
fun findFood(id : Int) : Food? {
    val db = this.readableDatabase

    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_ID = ?"
    db.rawQuery(query, arrayOf(id.toString())).use{
        if (it.moveToFirst()){
            val food = Food()
            food.name=it.getString(it.getColumnIndex(COL_FOOD_NAME))
            food.category=it.getString(it.getColumnIndex(COL_FOOD_CATEGORY))
            food.shoppingList=it.getString(it.getColumnIndex(COL_FOOD_SHOPPING)).toInt()
            food.favourite=it.getString(it.getColumnIndex(COL_FOOD_FAVOURITE)).toInt()
            food.quantity = it.getString(it.getColumnIndex(COL_FOOD_QUANTITY)).toDouble()
            food.measurement = it.getString(it.getColumnIndex(COL_FOOD_MEASUREMENT))
            val image = it.getBlob(it.getColumnIndex(COL_FOOD_PHOTO))
            if (image!=null)
                food.photo = BitmapFactory.decodeByteArray(image, 0, image.size )
            return food
        }
    }
    db.close()
    return null
}
fun findFoodName(name:String): Int? {
    val db = this.readableDatabase

    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_NAME = ?"
```

```
    db.rawQuery(query, arrayOf(name)).use{
        if (it.moveToFirst()){
            val food = Food()
            food.id=it.getString(it.getColumnIndex(COL_FOOD_ID)).toInt()
            return food.id
        }
    }
    db.close()
    return null
}

fun addFoodShopping(id:Int) {
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_SHOPPING, "1")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
fun removeFoodShopping(id:Int){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_SHOPPING, "0")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
fun readShopping():MutableList<Food>{
    val list : MutableList<Food> = ArrayList()
    val db = this.readableDatabase
    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_SHOPPING =\"1\" " +
            "ORDER BY $COL_FOOD_CATEGORY, $COL_FOOD_NAME"
    val result = db.rawQuery(query, null)
    if(result.moveToFirst()){
        do {
            val food = Food()
            food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
            list.add(food)
        }while (result.moveToNext())
```

```
    }
    result.close()
    db.close()
    return list
}
fun readShoppingCategory(category:String):MutableList<Food>{
    val list : MutableList<Food> = ArrayList()
    val db = this.readableDatabase
    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_SHOPPING =\"1\" " +
            "AND $COL_FOOD_CATEGORY = \"$category\"" +
            "ORDER BY $COL_FOOD_CATEGORY, $COL_FOOD_NAME"
    val result = db.rawQuery(query, null)
    if(result.moveToFirst()){
        do {
            val food = Food()
            food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
            list.add(food)
        }while (result.moveToNext())
    }
    result.close()
    db.close()
    return list
}

fun addFoodBought(id:Int){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_BOUGHT, "1")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
fun removeFoodBought(id:Int){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_BOUGHT, "0")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
```

```
    }
    fun readBought():MutableList<Food>{
        val list : MutableList<Food> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_BOUGHT =\"1\" ORDER BY
$COL_FOOD_NAME"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val food = Food()
                food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
                list.add(food)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }
    fun readBoughtCategory(category: String):MutableList<Food>{
        val list : MutableList<Food> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_BOUGHT =\"1\" " +
                "AND $COL_FOOD_CATEGORY = \"$category\"" +
                "ORDER BY $COL_FOOD_CATEGORY, $COL_FOOD_NAME"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val food = Food()
                food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
                list.add(food)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }

    fun addFoodFavourites(id:Int){
        val db = this.writableDatabase

        val cv = ContentValues()
        cv.put(COL_FOOD_FAVOURITE, "1")

        val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
```

```
    fun removeFoodFavourites(id:Int){
        val db = this.writableDatabase

        val cv = ContentValues()
        cv.put(COL_FOOD_FAVOURITE, "0")

        val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun readFoodFavourites():MutableList<Food>{
        val list : MutableList<Food> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_FAVOURITE =\"1\" ORDER BY
$COL_FOOD_NAME"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val food = Food()
                food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
                list.add(food)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }

    fun readFoodCategory(cat:String):MutableList<Food>{
        val list : MutableList<Food> = ArrayList()
        val db = this.readableDatabase

        val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_CATEGORY = \"$cat\" ORDER BY
$COL_FOOD_NAME"

        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val food = Food()
                food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
                list.add(food)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }
```

```
fun addFoodCupboard(id:Int){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_CUPBOARD, "1")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
private fun removeFoodCupboard(id:Int){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_FOOD_CUPBOARD, "0")

    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    db.close()
}
fun readFoodCupboard() :MutableList<Food>{
    val list : MutableList<Food> = ArrayList()
    val db = this.readableDatabase
    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_CUPBOARD =\"1\" " +
            "ORDER BY $COL_FOOD_CATEGORY, $COL_FOOD_NAME"
    val result = db.rawQuery(query, null)
    if(result.moveToFirst()){
        do {
            val food = Food()
            food.id = result.getString(result.getColumnIndex(COL_FOOD_ID)).toInt()
            food.name = result.getString(result.getColumnIndex(COL_FOOD_NAME))
            food.quantity = result.getString(result.getColumnIndex(COL_FOOD_QUANTITY)).toDouble()
            food.measurement = result.getString(result.getColumnIndex(COL_FOOD_MEASUREMENT))
            list.add(food)
        }while (result.moveToNext())
    }
    result.close()
    db.close()
    return list
}

private fun findFoodQuantity(id : Int) : Food? {
```

```
    val db = this.readableDatabase

    val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_ID = ?"
    db.rawQuery(query, arrayOf(id.toString())).use{
      if (it.moveToFirst()){
        val food = Food()
        food.name=it.getString(it.getColumnIndex(COL_FOOD_NAME))
        food.category=it.getString(it.getColumnIndex(COL_FOOD_CATEGORY))
        food.quantity = it.getString(it.getColumnIndex(COL_FOOD_QUANTITY)).toDouble()
        food.measurement = it.getString(it.getColumnIndex(COL_FOOD_MEASUREMENT))
        return food
      }
    }
    db.close()
    return null
  }
  fun addFoodQuantity(id:Int, qty:Double, msr:String){
    val db = this.writableDatabase
    val cv = ContentValues()

    val food : Food? = findFoodQuantity(id)
    val m : String? = findMeasurement(id)

    val oldqty : Double = food?.quantity!!.toDouble()
    val oldG = Measurements(oldqty, m!!, "grams")
    oldG.convert()
    val newG = Measurements(qty, msr, "grams")
    newG.convert()
    val newqty : Double = oldG.quantity + newG.quantity
    val resultQty = Measurements(newqty, "grams", m)
    resultQty.convert()

    cv.put(COL_FOOD_QUANTITY,resultQty.quantity)
    val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
    if(result >=1 ) {
      Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
      Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    addFoodCupboard(id)
    db.close()
  }
  fun delFoodQuantity(id:Int, qty:Double, msr:String){
    val db = this.writableDatabase
    val cv = ContentValues()

    val food : Food? = findFoodQuantity(id)
    val m : String? = findMeasurement(id)

    val oldqty : Double = food?.quantity!!
    val oldG = Measurements(oldqty, m!!, "grams")
```

```
        oldG.convert()
        val newG = Measurements(qty, msr, "grams")
        newG.convert()
        val newqty : Double = oldG.quantity - newG.quantity
        val resultQty = Measurements(newqty, "grams", m)
        resultQty.convert()

        if (resultQty.quantity <= 0.0) {
            setFoodQuantity(id,0.0)
            removeFoodCupboard(id)
        }
        else{
            cv.put(COL_FOOD_QUANTITY,resultQty.quantity)
            db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
        }
        db.close()
    }
    private fun checkDelFoodQuantity(id:Int, qty:Double, msr:String) : Boolean{
        val food : Food? = findFoodQuantity(id)
        val m : String? = findMeasurement(id)

        val oldqty : Double = food?.quantity!!
        val oldG = Measurements(oldqty, m!!, "grams")
        oldG.convert()
        val newG = Measurements(qty, msr, "grams")
        newG.convert()
        val newqty : Double = oldG.quantity - newG.quantity
        val resultQty = Measurements(newqty, "grams", m)
        resultQty.convert()

        return resultQty.quantity >= 0.0
    }

    private fun setFoodQuantity(id:Int, qty:Double){
        val db = this.writableDatabase

        val cv = ContentValues()
        cv.put(COL_FOOD_QUANTITY, qty)

        val result = db.update(TABLE_FOOD, cv, "$COL_FOOD_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    private fun findMeasurement(id : Int) : String?{
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_FOOD WHERE $COL_FOOD_ID = ?"
        db.rawQuery(query, arrayOf(id.toString())).use{
```

```
        if (it.moveToFirst()){
            return it.getString(it.getColumnIndex(COL_FOOD_MEASUREMENT))
        }
    }
    db.close()
    return null
}


/****************** RECIPES TABLE ********************/
fun insertRecipe(recipe:Recipe) : Long?{
    val db = this.writableDatabase
    val bos = ByteArrayOutputStream()

    recipe.photo?.compress(Bitmap.CompressFormat.JPEG, 10, bos)
    val bArray = bos.toByteArray()
    val cv = ContentValues()

    cv.put(COL_RECIPE_NAME, recipe.name)
    cv.put(COL_RECIPE_METHOD,recipe.method)
    cv.put(COL_RECIPE_CUISINE, recipe.cuisine)
    cv.put(COL_RECIPE_DESCRIPTION, recipe.description)
    cv.put(COL_RECIPE_MEAL, recipe.mealType)
    cv.put(COL_RECIPE_PHOTO, bArray)
    cv.put(COL_RECIPE_SERVINGS, recipe.servings)

    val result = db.insert(TABLE_RECIPE, null, cv)
    return if(result == (-1).toLong()) {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        null
    } else {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        result
    }
}
fun updateRecipe(name:String, method:String, cuisine:String, description:String,
                 mealType:String, bitmap: Bitmap, servings: Int, id:Int){
    val db = this.writableDatabase
    val bos = ByteArrayOutputStream()
    bitmap.compress(Bitmap.CompressFormat.JPEG, 30, bos)
    val bArray = bos.toByteArray()
    val cv = ContentValues()

    cv.put(COL_RECIPE_NAME, name)
    cv.put(COL_RECIPE_METHOD,method)
    cv.put(COL_RECIPE_CUISINE, cuisine)
    cv.put(COL_RECIPE_DESCRIPTION, description)
    cv.put(COL_RECIPE_MEAL, mealType)
    cv.put(COL_RECIPE_PHOTO, bArray)
    cv.put(COL_RECIPE_SERVINGS, servings)
```

```
        val result = db.update(TABLE_RECIPE, cv, "$COL_RECIPE_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun readRecipeData() : MutableList<Recipe>{
        val list : MutableList<Recipe> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_RECIPE ORDER BY $COL_RECIPE_FAVOURITE,
$COL_RECIPE_NAME"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val recipe = Recipe()
                recipe.id = result.getString(result.getColumnIndex(COL_RECIPE_ID)).toInt()
                recipe.name = result.getString(result.getColumnIndex(COL_RECIPE_NAME))
                list.add(recipe)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }
    fun findRecipe(id:Int): Recipe?{
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_RECIPE WHERE $COL_RECIPE_ID = ?"
        db.rawQuery(query, arrayOf(id.toString())).use{
            if (it.moveToFirst()){
                val recipe = Recipe()
                recipe.name=it.getString(it.getColumnIndex(COL_RECIPE_NAME))
                recipe.mealType = it.getString(it.getColumnIndex(COL_RECIPE_MEAL))
                recipe.cuisine = it.getString(it.getColumnIndex(COL_RECIPE_CUISINE))
                recipe.description = it.getString(it.getColumnIndex(COL_RECIPE_DESCRIPTION))
                recipe.method = it.getString(it.getColumnIndex(COL_RECIPE_METHOD))
                recipe.favourite = it.getString(it.getColumnIndex(COL_RECIPE_FAVOURITE)).toInt()
                recipe.servings = it.getString(it.getColumnIndex(COL_RECIPE_SERVINGS)).toInt()
                val image = it.getBlob(it.getColumnIndex(COL_RECIPE_PHOTO))
                if (image!=null)
                    recipe.photo = BitmapFactory.decodeByteArray(image, 0, image.size )
                return recipe
            }
        }
        db.close()
        return null
    }
    fun findRecipeName(name:String): Int? {
        val db = this.readableDatabase
```

```
        val query = "SELECT * FROM $TABLE_RECIPE WHERE $COL_RECIPE_NAME = ?"
        db.rawQuery(query, arrayOf(name)).use{
            if (it.moveToFirst()){
                val recipe = Recipe()
                recipe.id=it.getString(it.getColumnIndex(COL_RECIPE_ID)).toInt()
                return recipe.id
            }
        }
        db.close()
        return null
    }

    fun addRecipeFavourites(id:Int){
        val db = this.writableDatabase

        val cv = ContentValues()
        cv.put(COL_RECIPE_FAVOURITE, "1")

        val result = db.update(TABLE_RECIPE, cv, "$COL_RECIPE_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun removeRecipeFavourites(id:Int){
        val db = this.writableDatabase

        val cv = ContentValues()
        cv.put(COL_RECIPE_FAVOURITE, "0")

        val result = db.update(TABLE_RECIPE, cv, "$COL_RECIPE_ID = $id", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun readRecipeFavourites():MutableList<Recipe>{
        val list : MutableList<Recipe> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_RECIPE WHERE $COL_RECIPE_FAVOURITE =\"1\" ORDER BY
$COL_FOOD_NAME"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val recipe = Recipe()
                recipe.id = result.getString(result.getColumnIndex(COL_RECIPE_ID)).toInt()
                list.add(recipe)
```

```
        }while (result.moveToNext())
    }
    result.close()
    db.close()
    return list
}

fun suggestRecipes():MutableList<Recipe>{
    val cupboard : MutableList<Food> = readFoodCupboard()
    val suggested : MutableList<Recipe> = ArrayList()
    val recipes : MutableList<Recipe> = readRecipeData()

    for(i in 0..(recipes.size-1)){
        val recipe : Recipe = recipes[i]
        val ingredientList : MutableList<Ingredients> = readIngredients(recipe.id)
        var valid = false
        for( j in 0..(ingredientList.size-1)){
            val ingredients : Ingredients =ingredientList[j]
            var found = false
            for (x in 0..(cupboard.size-1)){
                if (ingredients.foodID == cupboard[x].id){
                    val msr = Measurements(cupboard[x].quantity,cupboard[x].measurement,ingredients.measurement)
                    msr.convert()
                    val new = msr.quantity - ingredients.quantity
                    if(new >= 0.0) {
                        found = true
                        break
                    }
                    else{
                        found = false
                        break
                    }
                }
            }
            if(!found){
                valid = false
                break
            }
            else
                valid = true
        }
        if(valid)
            suggested.add(recipe)
    }
    return suggested
}

/********************** INGREDIENTS TABLE ******************************/
fun insertIngredient(ingredients: Ingredients){
    val db = this.writableDatabase
```

```
        val cv = ContentValues()

        cv.put(COL_INGREDIENT_RECIPE, ingredients.recipeID)
        cv.put(COL_INGREDIENT_FOOD, ingredients.foodID)
        cv.put(COL_INGREDIENT_QUANTITY, ingredients.quantity)
        cv.put(COL_INGREDIENT_MEASUREMENT, ingredients.measurement)

        db.insert(TABLE_INGREDIENT, null, cv)
    }
    fun readIngredients(id:Int) : MutableList<Ingredients>{
        val list : MutableList<Ingredients> = ArrayList()
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_INGREDIENT WHERE $COL_INGREDIENT_RECIPE = ?"
        val result = db.rawQuery(query, arrayOf(id.toString()))
        if (result.moveToFirst()) {
            do {
                val ingredient = Ingredients()
                ingredient.id = result.getString(result.getColumnIndex(COL_INGREDIENT_ID)).toInt()
                ingredient.foodID =
result.getString(result.getColumnIndex(COL_INGREDIENT_FOOD)).toInt()
                list.add(ingredient)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }
    fun findIngredient(id:Int) : Ingredients?{
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_INGREDIENT WHERE $COL_INGREDIENT_ID = ?"
        db.rawQuery(query, arrayOf(id.toString())).use{
            if (it.moveToFirst()){
                val ingredient = Ingredients()
                ingredient.quantity =
it.getString(it.getColumnIndex(COL_INGREDIENT_QUANTITY)).toDouble()
                ingredient.measurement =
it.getString(it.getColumnIndex(COL_INGREDIENT_MEASUREMENT))
                return ingredient
            }
        }
        db.close()
        return null
    }
    fun updateIngredient(id: Int, qty: Double, msr:String){
        val db = this.writableDatabase
        val cv = ContentValues()

        cv.put(COL_INGREDIENT_QUANTITY, qty)
        cv.put(COL_INGREDIENT_MEASUREMENT, msr)

        val result = db.update(TABLE_INGREDIENT, cv, "$COL_INGREDIENT_ID = $id", null)
```

```
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
    }
    fun removeIngredient(id:Int){
        this.writableDatabase.apply {
            delete(TABLE_INGREDIENT, "$COL_INGREDIENT_ID = $id", null)
            close()
        }
    }

    fun removeQuantityCupboard(iId:Int, fId:Int){
        val db = this.readableDatabase
        val ingredient = findIngredient((iId))

        val iQuantity = ingredient!!.quantity
        val msr : String? = ingredient.measurement
        delFoodQuantity(fId, iQuantity, msr!!)

        db.close()
    }

    /************************ DIET TABLE ********************************/
    fun insertDiet(diet: Diet){
        val db = this.writableDatabase
        val cv = ContentValues()

        cv.put(COL_DIET_NAME, diet.name)
        cv.put(COL_DIET_RECIPE, diet.recipeID)

        db.insert(TABLE_DIET, null, cv)
        db.close()
    }
    fun findDietName(rId:Int) : String?{
        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_DIET WHERE $COL_DIET_RECIPE = ?"
        db.rawQuery(query, arrayOf(rId.toString())).use{
            if (it.moveToFirst()){
                return it.getString(it.getColumnIndex(COL_DIET_NAME))
            }
        }
        db.close()
        return null
    }
    fun updateDiet(name:String, rId : Int){
        val db = this.writableDatabase
        val cv = ContentValues()

        cv.put(COL_DIET_NAME, name)
```

```
    val result = db.update(TABLE_DIET, cv, "$COL_DIET_RECIPE = $rId", null)
    if(result >=1 ) {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
}

/************************* BARCODE TABLE ************************************/

fun insertBarcode(barcode: Barcodes){
    val db = this.writableDatabase

    val cv = ContentValues()
    cv.put(COL_BARCODE_BARCODE, barcode.barcode)
    cv.put(COL_BARCODE_SCANNED, System.currentTimeMillis())

    val result = db.insert(TABLE_BARCODE,null,cv)
    if(result == (-1).toLong()) {
        Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
    }
    else {
        Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
    }
}
fun findBarcode(id:Int):Barcodes?{
    val db = this.readableDatabase
    val query = "SELECT * FROM $TABLE_BARCODE WHERE $COL_BARCODE_ID = ?"
    db.rawQuery(query, arrayOf(id.toString())).use{
        if (it.moveToFirst()){
            val barcode = Barcodes()
            barcode.barcode = it.getString(it.getColumnIndex(COL_BARCODE_BARCODE))
            barcode.type = it.getString(it.getColumnIndex(COL_BARCODE_TYPE))
            barcode.brand = it.getString(it.getColumnIndex(COL_BARCODE_BRAND))
            if(it.getString(it.getColumnIndex(COL_BARCODE_FOODID)) != null)
                barcode.foodID = it.getString(it.getColumnIndex(COL_BARCODE_FOODID)).toInt()
            else
                barcode.foodID = null
            barcode.quantity = it.getString(it.getColumnIndex(COL_BARCODE_QUANTITY)).toDouble()
            barcode.measurement = it.getString(it.getColumnIndex(COL_BARCODE_MEASUREMENT))
            return barcode
        }
    }
    db.close()
    return null
}
fun checkBarcode(barcode:String) : Boolean{
    val db = this.readableDatabase

    val query = "SELECT * FROM " + TABLE_BARCODE + " WHERE " +
```

```
            COL_BARCODE_BARCODE + " = ?"

        db.rawQuery(query, arrayOf(barcode)).use{
            if(it.count > 0)
                return true
        }
        db.close()
        return false
    }
    fun updateLastScan(barcode : String){
        val db = this.writableDatabase
        val cv = ContentValues()
        cv.put(COL_BARCODE_SCANNED,System.currentTimeMillis())
        val result = db.update(TABLE_BARCODE, cv, "$COL_BARCODE_BARCODE = $barcode", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun updateBarcode(bId : Int, fID : Int, qty : Double, msr : String, brand : String){
        val db = this.writableDatabase
        val cv = ContentValues()

        cv.put(COL_BARCODE_FOODID,fID)
        cv.put(COL_BARCODE_QUANTITY, qty)
        cv.put(COL_BARCODE_MEASUREMENT, msr)
        cv.put(COL_BARCODE_BRAND, brand)

        val result = db.update(TABLE_BARCODE, cv, "$COL_BARCODE_ID = $bId", null)
        if(result >=1 ) {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        db.close()
    }
    fun findBarcodeName(barcode : String) : Int?{
        val db = this.readableDatabase

        val query = "SELECT * FROM " + TABLE_BARCODE + " WHERE " +
                COL_BARCODE_BARCODE + " = ?"

        db.rawQuery(query, arrayOf(barcode)).use{
            if (it.moveToFirst()){
                val barcodes = Barcodes()
                barcodes.id=it.getString(it.getColumnIndex(COL_BARCODE_ID)).toInt()
                return barcodes.id
            }
        }
```

```
        db.close()
        return null
    }
    fun readBarcodeData(): MutableList<Barcodes>{
        val list : MutableList<Barcodes> = ArrayList()

        val db = this.readableDatabase
        val query = "SELECT * FROM $TABLE_BARCODE ORDER BY $COL_BARCODE_SCANNED DESC"
        val result = db.rawQuery(query, null)
        if(result.moveToFirst()){
            do {
                val barcode = Barcodes()
                barcode.id = result.getString(result.getColumnIndex(COL_BARCODE_ID)).toInt()
                barcode.barcode = result.getString(result.getColumnIndex(COL_BARCODE_BARCODE))
                list.add(barcode)
            }while (result.moveToNext())
        }
        result.close()
        db.close()
        return list
    }
}
```

Section 3.2 – Adapters

The full list of adapters in the project are:

- AddIngredientsAdapter.kt
- AutoCompleteFoodAdapter.kt
- AutoCompleteREcipeAdapter.kt
- BarcodeAdapter.kt
- BoughtAdapter.kt
- CupboardAdapter.kt
- FoodAdapter.kt
- IngredientsAdapter.kt
- RecipeAdapter.kt
- ShoppingAdapter.kt
- ShoppingPopupAdapter.kt

Most are similar in functionality, though they correspond to different layouts. Included in this document is the AutoCoompleteFoodAdapter.kt, the FoodAdapter.kt, the RecipeAdapter.kt, and the ShoppingPopupAdapter.kt.

3.2.1 – AutoCompleteFoodAdapter.kt

```
package com.kitchenette.search

import android.app.Activity
import android.content.Context
import android.graphics.Bitmap
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import com.kitchenette.kitchenette.DataBaseHandler
import com.kitchenette.kitchenette.Food
import com.kitchenette.kitchenette.R

import java.util.*

class AutoCompleteFoodAdapter(private val context: Activity, private var foodItems : ArrayList<Food>)
   : ArrayAdapter<Food>(context, R.layout.list_item_food, foodItems) {


   private var resultList: MutableList<Food> = ArrayList()

   override fun getCount(): Int {
      return resultList.size
   }
   override fun getItem(index: Int): Food {
      return resultList[index]
   }
   override fun getItemId(position: Int): Long {
      return position.toLong()
   }

   override fun getView(position: Int, view: View?, parent: ViewGroup): View {
      var view = view
      if (view == null) {
         val inflater = context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater
         view = inflater.inflate(R.layout.list_item_food, parent, false)
      }

      val db = DataBaseHandler(context)

      val food : Food? = db.findFood(resultList[position].id)

      val tvFood = view!!.findViewById<View>(R.id.tv_food) as TextView
      val image = view.findViewById<View>(R.id.image_food_icon) as ImageView
      tvFood.text = food?.name
      if(food?.photo!=null){
         val bitmap: Bitmap? = food.photo
```

```
            image.setImageBitmap(bitmap)
        }

        return view
    }

    override fun getFilter() = filter

    private var filter: Filter = object : Filter() {

        override fun performFiltering(constraint: CharSequence?): Filter.FilterResults {
            val results = FilterResults()

            val query = if (constraint != null && constraint.isNotEmpty())
autocomplete(constraint.toString())
            else arrayListOf()

            results.values = query
            results.count = query.size

            return results
        }

        private fun autocomplete(input: String): ArrayList<Food> {
            val results = arrayListOf<Food>()

            for (food in foodItems) {
                if (food.name.toLowerCase().contains(input.toLowerCase())) results.add(food)
            }

            return results
        }

        override fun publishResults(constraint: CharSequence?, results: Filter.FilterResults) {
            resultList = results.values as ArrayList<Food>
            notifyDataSetInvalidated()
        }

        override fun convertResultToString(result: Any) = (result as Food).name
    }
}
```

3.2.2 – FoodAdapter.kt

```
package com.kitchenette.kitchenette

import android.content.Context
import android.graphics.Bitmap
import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import kotlinx.android.synthetic.main.list_item_food.view.*


class FoodAdapter(private val items : ArrayList<String>, val context: Context) : RecyclerView.Adapter<ViewHolder>() {


    override fun onBindViewHolder(p0: ViewHolder, p1: Int) {
        TODO("not implemented") //To change body of created functions use File | Settings | File Templates.
    }

    override fun getItemCount(): Int {
        return items.size
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        return ViewHolder(LayoutInflater.from(context).inflate(R.layout.list_item_food, parent, false))
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int, payloads: MutableList<Any>) {
        val db = DataBaseHandler(context)

        val food : Food? = db.findFood(items[position].toInt())

        holder.tvFoodItem.text = food?.name
        if(food?.photo!= null){
            val bitmap: Bitmap? = food?.photo
            holder.image.setImageBitmap(bitmap)
        }
    }
}

class ViewHolder (view: View) : RecyclerView.ViewHolder(view) {
    val tvFoodItem = view.tv_food!!
    val image = view.image_food_icon!!
}
```

3.2.3 – RecipeAdapter.kt

```
package com.kitchenette.kitchenette

import android.content.Context
import android.graphics.Bitmap
import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import kotlinx.android.synthetic.main.list_item_recipe.view.*


class RecipeAdapter(private val items : ArrayList<String>, val context: Context) : RecyclerView.Adapter<RecipeView>() {


    override fun onBindViewHolder(p0: RecipeView, p1: Int) {
        TODO("not implemented") //To change body of created functions use File | Settings | File Templates.
    }

    override fun getItemCount(): Int {
        return items.size
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecipeView {
        return RecipeView(LayoutInflater.from(context).inflate(R.layout.list_item_recipe, parent, false))
    }

    override fun onBindViewHolder(holder: RecipeView, position: Int, payloads: MutableList<Any>) {
        val db = DataBaseHandler(context)

        val recipe : Recipe? = db.findRecipe(items[position].toInt())

        holder.tvRecipeItem.text = recipe?.name

        if(recipe?.photo!= null){
            val bitmap: Bitmap? = recipe?.photo
            holder.image.setImageBitmap(bitmap)
        }
    }
}

class RecipeView (view: View) : RecyclerView.ViewHolder(view) {
    val tvRecipeItem = view.tv_recipe!!
    val image = view.image_recipe_icon!!
}
```

3.2.4 – ShoppingPopupAdapter.kt

```
package com.kitchenette.kitchenette

import android.content.Context
import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import kotlinx.android.synthetic.main.list_item_shopping_popup.view.*


class ShoppingPopupAdapter(private val items : ArrayList<String>,val context: Context)
    : RecyclerView.Adapter<ShoppingPopupHolder>() {

    val adapter = this

    override fun onBindViewHolder(p0: ShoppingPopupHolder, p1: Int) {
        TODO("not implemented") //To change body of created functions use File | Settings | File Templates.
    }

    override fun getItemCount(): Int {
        return items.size
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ShoppingPopupHolder {
        return ShoppingPopupHolder(
            LayoutInflater.from(context).inflate(R.layout.list_item_shopping_popup,
                parent, false))
    }

    override fun onBindViewHolder(holder: ShoppingPopupHolder, position: Int, payloads: MutableList<Any>) {
        val db = DataBaseHandler(context)

        val food : Food? = db.findFood(items[position].toInt())
        holder.tvFoodItem.text = food?.name
        holder.quantity.text = food?.quantity.toString()
        holder.measurement.text = food?.measurement

        holder.addBtn.setOnClickListener {
            db.addFoodShopping(items[position].toInt())
            items.remove(items[position])
            adapter.notifyDataSetChanged()
        }

        db.close()
    }
}
```

```
class ShoppingPopupHolder (view: View) : RecyclerView.ViewHolder(view) {
    val tvFoodItem = view.tv_food!!
    val quantity = view.tv_quantity!!
    val measurement = view.tv_measure!!
    val addBtn = view.add_shop!!
}
```

Section 3.3 – Activity Files

3.3.1 – AddFoodActivity.kt

```
package com.kitchenette.kitchenette

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_add_food.*
import kotlinx.android.synthetic.main.app_bar_add_food.*
import kotlinx.android.synthetic.main.content_add_food.*

class AddFoodActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener,
    AdapterView.OnItemSelectedListener {

    var categoryList = arrayOf("Baking & Grains","Beans & Legumes","Beverages",
        "Broths & Soups","Condiments & Sauces","Dairy","Dairy Alternatives",
        "Desserts & Snacks","Fruit","Meat & Poultry","Nuts & Seeds","Oils","Seafood & Fish",
        "Spices, Herbs, Seasonings","Sweeteners","Vegetables")
    var categorySelected : String? = null

    var bitmap: Bitmap? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_food)
        setSupportActionBar(toolbar)

        val context = this
        val db = DataBaseHandler(context)

        /************************** SPINNER ************************************/

        foodCategory!!.onItemSelectedListener = this
        val aa = ArrayAdapter(this, android.R.layout.simple_spinner_item, categoryList)
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
```

```
foodCategory!!.adapter = aa

/************************** UPLOAD IMAGE ****************************/
upload_image.setOnClickListener {
  //check runtime permission
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
    if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
      PackageManager.PERMISSION_DENIED){
      //permission denied
      val permissions = arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE)
      //show popup to request runtime permission
      requestPermissions(permissions, PERMISSION_CODE)
    }
    else{
      pickImageFromGallery() //permission already granted
    }
  }
  else{
    pickImageFromGallery() //system OS is < Marshmallow
  }
}

/*********************** FLOATING ACTION BUTTON ************************/

fab.setOnClickListener {
  if (foodName.text.toString().isNotEmpty() &&
    categorySelected!!.isNotEmpty() &&
        bitmap != null
  ) {
    val food = Food(foodName.text.toString(), categorySelected!!, bitmap!!)
    val newID = db.insertFood(food)
    val message = newID.toString()
    val intent = Intent(this@AddFoodActivity, FoodItemActivity::class.java)
    intent.putExtra("food", message)
    startActivity(intent)
  } else {
    Toast.makeText(context, "Please Fill Out All details", Toast.LENGTH_SHORT).show()
  }

}

/************************ NAVIGATION DRAWER ******************************/

val toggle = ActionBarDrawerToggle(
  this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
)
drawer_layout.addDrawerListener(toggle)
toggle.syncState()

nav_view.setNavigationItemSelectedListener(this)
```

```
}

/************************NAV DRAWER METHODS *************************/

override fun onBackPressed() {
    if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
    } else {
        super.onBackPressed()
    }
}

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    // Inflate the menu; this adds items to the action bar if it is present.
    menuInflater.inflate(R.menu.settings_menu, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_settings -> true
        else -> super.onOptionsItemSelected(item)
    }
}

override fun onNavigationItemSelected(item: MenuItem): Boolean {
    // Handle navigation view item clicks here.
    when (item.itemId) {
        R.id.nav_cupboard -> {
            val menuIntent = Intent(this@AddFoodActivity, MainActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_cookbook -> {
            val menuIntent = Intent(this@AddFoodActivity, CookbookActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_shopping -> {
            val menuIntent = Intent(this@AddFoodActivity, ShoppingListActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_favourite -> {
            val menuIntent = Intent(this@AddFoodActivity, FavouritesActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_barcode -> {
            val menuIntent = Intent(this@AddFoodActivity, ScanBarcodeActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_share -> {
```

```
        }
      }

      drawer_layout.closeDrawer(GravityCompat.START)
      return true
    }

    /************************** SPINNER METHODS ********************************/
    override fun onItemSelected(arg0: AdapterView<*>, arg1: View, position: Int, id: Long) {
      categorySelected = categoryList[position]
    }

    override fun onNothingSelected(arg0: AdapterView<*>) {
    }

    /************************** UPLOAD IMAGE METHODS ***************************************/
    private fun pickImageFromGallery() {
      //Intent to pick image
      val intent = Intent(Intent.ACTION_PICK)
      intent.type = "image/*"
      startActivityForResult(intent, IMAGE_PICK_CODE)
    }

    companion object {
      //image pick code
      private const val IMAGE_PICK_CODE = 1000
      //Permission code
      internal const val PERMISSION_CODE = 1001
    }

    //handle requested permission result
    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
      when(requestCode){
        PERMISSION_CODE -> {
          if (grantResults.isNotEmpty() && grantResults[0] ==
            PackageManager.PERMISSION_GRANTED){
            //permission from popup granted
            pickImageFromGallery()
          }
          else{
            //permission from popup denied
            Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
          }
        }
      }
    }

    //handle result of picked image
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
```

```
        if (resultCode == Activity.RESULT_OK && requestCode == IMAGE_PICK_CODE){
            image.setImageURI(data?.data)
            image.cropToPadding
            bitmap = MediaStore.Images.Media.getBitmap(this.contentResolver, data?.data)
        }
    }
}
```

3.3.2 – AddRecipeActivity.kt

```
package com.kitchenette.kitchenette

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.Gravity
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.*
import com.kitchenette.search.AutoCompleteFoodAdapter
import kotlinx.android.synthetic.main.activity_add_recipe.*
import kotlinx.android.synthetic.main.app_bar_add_recipe.*
import kotlinx.android.synthetic.main.content_add_recipe.*
import kotlinx.android.synthetic.main.content_add_recipe.root_layout


class AddRecipeActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener,
    AdapterView.OnItemSelectedListener {

    val list : ArrayList<Food> = ArrayList()
    private val ingredients : ArrayList<String> = ArrayList()
    private val quantityList : ArrayList<Double> = ArrayList()
    private val measureList : ArrayList<String> = ArrayList()
    var bitmap: Bitmap? = null
    private val mealTypeList = arrayOf("Breakfast", "Lunch", "Dinner", "Desserts & Snacks", "Other")
    var selected : String? = null
    private val measurements = arrayOf("cup","dessertspoon","fl. oz",
        "grams","kg","litres","ml","oz","pint","tbsp","tsp", "whole")
    var s : String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_recipe)
        setSupportActionBar(toolbar)

        /********************* FLOATING ACTION BUTTON ******************************/
        fab.setOnClickListener {
```

```
        val context = this
        val db = DataBaseHandler(context)

        if(name.text.toString().isNotEmpty() &&
                description.text.toString().isNotEmpty() &&
                method.text.toString().isNotEmpty()&&
                servings.text.toString().isNotEmpty() &&
                cuisine.text.toString().isNotEmpty() &&
                bitmap!= null &&
                ingredients.isNotEmpty() &&
                quantityList.isNotEmpty() &&
                measureList.isNotEmpty()){

            val recipe = Recipe(name.text.toString(),selected.toString(), cuisine.text.toString(),
                servings.text.toString().toInt(),description.text.toString(),method.text.toString(), bitmap!!)
            val newRecipeID = db.insertRecipe(recipe)

            if (newRecipeID!=null){
                for (i in 0..(ingredients.size-1)) {
                    val ing = Ingredients(newRecipeID, ingredients[i].toInt(), quantityList[i], measureList[i])
                    db.insertIngredient(ing)
                }
                if(diet.text.toString().isNotEmpty()){
                    val diet = Diet(diet.text.toString(), newRecipeID)
                    db.insertDiet(diet)
                }
            }
            val message = newRecipeID.toString()
            val intent = Intent(this@AddRecipeActivity, RecipeItemActivity::class.java)
            intent.putExtra("recipe", message)
            startActivity(intent)
        }else
            Toast.makeText(context, "Please Fill Out All details", Toast.LENGTH_SHORT).show()
    }

    /********************** NAVIGATION *********************************/
    val toggle = ActionBarDrawerToggle(
        this,          drawer_layout,          toolbar,          R.string.navigation_drawer_open,
R.string.navigation_drawer_close
    )
    drawer_layout.addDrawerListener(toggle)
    toggle.syncState()

    nav_view.setNavigationItemSelectedListener(this)

    /************************** UPLOAD IMAGE ******************************/
    upload_image.setOnClickListener {
        //check runtime permission
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
            if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
                PackageManager.PERMISSION_DENIED){
```

```
            //permission denied
            val permissions = arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE)
            //show popup to request runtime permission
            requestPermissions(permissions, AddFoodActivity.PERMISSION_CODE)
          }
          else{
            pickImageFromGallery() //permission already granted
          }
        }
        else{
          pickImageFromGallery() //system OS is < Marshmallow
        }
      }

      /************************** SPINNER ************************************/

      meal_type!!.onItemSelectedListener = this
      val aa = ArrayAdapter(this, android.R.layout.simple_spinner_item, mealTypeList)
      aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
      meal_type!!.adapter = aa

      /************************** SEARCH METHODS
**********************************/

      addFoodItems()

      val adapter = AutoCompleteFoodAdapter(this, list)
      autocompletetextview?.threshold=1
      autocompletetextview?.setAdapter(adapter)
      autocompletetextview?.setOnFocusChangeListener {
          _, _ ->
        autocompletetextview.setOnItemClickListener { _, _, _, _ ->
          val db = DataBaseHandler(this)
          val message = db.findFoodName(autocompletetextview.text.toString()).toString()

          ingredientPopup(message.toInt())
          autocompletetextview.text.clear()
          val item = findViewById<RecyclerView>(R.id.ingredient_list)
          item.layoutManager = LinearLayoutManager(this)
          item.adapter = AddIngredientAdapter(ingredients, quantityList,measureList, this)
        }
      }
    }

    /************************** NAVIGATION METHODS
****************************************/
    override fun onBackPressed() {
      if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
      } else {
        super.onBackPressed()
```

```
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@AddRecipeActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@AddRecipeActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@AddRecipeActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_favourite -> {
                val menuIntent = Intent(this@AddRecipeActivity, FavouritesActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_barcode -> {
                val menuIntent = Intent(this@AddRecipeActivity, ScanBarcodeActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_share -> {

            }
        }

        drawer_layout.closeDrawer(GravityCompat.START)
        return true
    }

    /************************* UPLOAD IMAGE METHODS
*****************************************/
    private fun pickImageFromGallery() {
        val intent = Intent(Intent.ACTION_PICK)
        intent.type = "image/*"
```

```
        startActivityForResult(intent, IMAGE_PICK_CODE)
    }

    companion object {
        private const val IMAGE_PICK_CODE = 1000
        private const val PERMISSION_CODE = 1001
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
        when(requestCode){
            PERMISSION_CODE -> {
                if (grantResults.isNotEmpty() && grantResults[0] ==
                    PackageManager.PERMISSION_GRANTED){
                    pickImageFromGallery()
                }
                else
                    Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
            }
        }
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        if (resultCode == Activity.RESULT_OK && requestCode == IMAGE_PICK_CODE){
            image.setImageURI(data?.data)
            image.cropToPadding
            bitmap = MediaStore.Images.Media.getBitmap(this.contentResolver, data?.data)
        }
    }

    /************************** SPINNER METHODS ********************************/
    override fun onItemSelected(parent: AdapterView<*>, view: View, position: Int, id: Long) {
        if(parent.id == R.id.meal_type)
            selected = mealTypeList[position]
        else if(parent.id == R.id.enter_measurement)
            s = measurements[position]
    }

    override fun onNothingSelected(arg0: AdapterView<*>) {}

    /************************** ADD FOOD ITEMS TO LIST METHODS
**********************************/
    private fun addFoodItems() {
        val context = this
        val db = DataBaseHandler(context)

        val data = db.readFoodData()

        for(i in 0..(data.size-1)){
            list.add(data[i])
        }
```

```
    }

    /*************************** POPUP METHODS*********************************/
    private fun ingredientPopup(id:Int){
        val context = this
        val db = DataBaseHandler(context)
        val window = PopupWindow(context)
        val view = layoutInflater.inflate(R.layout.popup_add_ingredient,null)

        window.isFocusable = true
        window.isOutsideTouchable = true
        window.update()
        window.width = LinearLayout.LayoutParams.MATCH_PARENT

        window.contentView = view

        view.findViewById<Spinner>(R.id.enter_measurement)!!.onItemSelectedListener = this
        val a = ArrayAdapter(this,
            android.R.layout.simple_spinner_item, this.measurements)
        a.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        view.findViewById<Spinner>(R.id.enter_measurement)!!.adapter = a


        val food : Food? = db.findFood(id)
        val foodLabel = view.findViewById<TextView>(R.id.food)
        foodLabel.text = food?.name

        val qty = view.findViewById<EditText>(R.id.enter_quantity)


        val add = view.findViewById<ImageButton>(R.id.add_qty_btn)
        add.setOnClickListener{
            ingredients.add(id.toString())
            measureList.add(s!!)
            val amt  = qty!!.text.toString().toDouble()
            quantityList.add(amt)
            window.dismiss()
        }

        val close  = view.findViewById<ImageButton>(R.id.cancel)
        close.setOnClickListener {
            window.dismiss()
        }
        db.close()
        window.showAtLocation(root_layout, Gravity.CENTER,0,0)
    }
}
```

3.3.3 – BarcodeHistoryActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.view.Menu
import android.view.MenuItem
import kotlinx.android.synthetic.main.activity_barcode_history.*
import kotlinx.android.synthetic.main.app_bar_barcode_history.*
import kotlinx.android.synthetic.main.content_barcode_history.*

class BarcodeHistoryActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private val list : ArrayList<String> = ArrayList()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_barcode_history)
        setSupportActionBar(toolbar)

        fab.setOnClickListener {
            val intent = Intent(this@BarcodeHistoryActivity, ScanBarcodeActivity::class.java)
            startActivity(intent)
        }

        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)

        addBarcodeItems()
        item.layoutManager = LinearLayoutManager(this)
        item.adapter = BarcodeAdapter(list, this, this)
    }

    /********************** NAV DRAWER METHODS ***************************/
    override fun onBackPressed() {
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
```

```
    }
  }

  override fun onCreateOptionsMenu(menu: Menu): Boolean {
    menuInflater.inflate(R.menu.settings_menu, menu)
    return true
  }

  override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
      R.id.action_settings -> true
      else -> super.onOptionsItemSelected(item)
    }
  }

  override fun onNavigationItemSelected(item: MenuItem): Boolean  {
    when (item.itemId) {
      R.id.nav_cupboard -> {
        val menuIntent = Intent(this@BarcodeHistoryActivity, MainActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_cookbook -> {
        val menuIntent = Intent(this@BarcodeHistoryActivity, CookbookActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_shopping -> {
        val menuIntent = Intent(this@BarcodeHistoryActivity, ShoppingListActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_favourite -> {
        val menuIntent = Intent(this@BarcodeHistoryActivity, FavouritesActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_barcode -> {
        val menuIntent = Intent(this@BarcodeHistoryActivity, ScanBarcodeActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_share -> {

      }
    }

    drawer_layout.closeDrawer(GravityCompat.START)
    return true
  }

  /********************** VIEW BARCODE HISTORY METHODS ********************/
  private fun addBarcodeItems()
  {
    val context = this
    val db = DataBaseHandler(context)
```

```
        val data = db.readBarcodeData()

        for(i in 0..(data.size-1))
            list.add(data[i].id.toString())
    }
}
```

3.3.4 – CookbookActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.design.widget.TabLayout
import android.support.v4.app.Fragment
import android.support.v4.app.FragmentManager
import android.support.v4.app.FragmentPagerAdapter
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.*
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.AutoCompleteTextView
import android.widget.Spinner
import com.kitchenette.search.AutoCompleteFoodAdapter
import kotlinx.android.synthetic.main.activity_cookbook.*
import kotlinx.android.synthetic.main.app_bar_cookbook.*


class CookbookActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private var mSectionsPagerAdapter: CookbookActivity.SectionsPagerAdapter? = null
    private val list : ArrayList<Recipe> = ArrayList()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_cookbook)
        setSupportActionBar(toolbar)

        /************************ TAB  ACTIVITY ************************/
        mSectionsPagerAdapter = SectionsPagerAdapter(supportFragmentManager)

        container.adapter = mSectionsPagerAdapter

        container.addOnPageChangeListener(TabLayout.TabLayoutOnPageChangeListener(tabs))
        tabs.addOnTabSelectedListener(TabLayout.ViewPagerOnTabSelectedListener(container))

        /******************FLOATING ACTION BUTTON ************************/
        fab.setOnClickListener { view ->
            val menuIntent = Intent(this@CookbookActivity, AddRecipeActivity::class.java)
            startActivity(menuIntent)
        }

        /************************ NAVIGATION DRAWER ************************/
```

```
        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)

        /************************** SEARCH METHODS **********************************/
        addRecipeItems()
        val autocompletetextview = findViewById<AutoCompleteTextView>(R.id.autocompletetextview)
        val adapter = AutoCompleteRecipeAdapter(this, list)
        autocompletetextview?.threshold=1
        autocompletetextview?.setAdapter(adapter)
        autocompletetextview?.setOnFocusChangeListener {
            _, _ ->
            autocompletetextview.setOnItemClickListener { _, _, _, _ ->
                val db = DataBaseHandler(this)
                val message = db.findRecipeName(autocompletetextview.text.toString()).toString()
                val intent = Intent(this@CookbookActivity, RecipeItemActivity::class.java)
                intent.putExtra("recipe", message)
                this.startActivity(intent)
            }
        }

    }


    /**************** NAVIGATION METHODS *****************/
    override fun onBackPressed() {
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        // Inflate the menu; this adds items to the action bar if it is present.
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }
```

```
override fun onNavigationItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        R.id.nav_cupboard -> {
            val menuIntent = Intent(this@CookbookActivity, MainActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_cookbook -> {
            val menuIntent = Intent(this@CookbookActivity, CookbookActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_shopping -> {
            val menuIntent = Intent(this@CookbookActivity, ShoppingListActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_favourite -> {
            val menuIntent = Intent(this@CookbookActivity, FavouritesActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_barcode -> {
            val menuIntent = Intent(this@CookbookActivity, ScanBarcodeActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_share -> {

        }
    }

    drawer_layout.closeDrawer(GravityCompat.START)
    return true
}

/************************ SEARCH METHODS ***************************/
private fun addRecipeItems(){
    val context = this
    val db = DataBaseHandler(context)

    val data = db.readRecipeData()

    for(i in 0..(data.size-1))
        list.add(data[i])
}

/**************** TAB ACTIVITY METHODS *****************/
inner class SectionsPagerAdapter(fm: FragmentManager) : FragmentPagerAdapter(fm) {

    override fun getItem(position: Int): Fragment {
        return PlaceholderFragment.newInstance(position + 1)
    }

    override fun getCount(): Int { return 2 }
```

```
}

class PlaceholderFragment : Fragment() {

    private val list : ArrayList<String> = ArrayList()
    private var mealList : Array<String> = arrayOf("All","Breakfast", "Lunch",
        "Dinner","Desserts & Snacks", "Other")
    private var cuisineList : ArrayList<String> = ArrayList()
    private var dietList : ArrayList<String> = ArrayList()
    private var selectMeal : String? = null
    private var selectCuisine : String? = null
    private var selectDiet : String? = null

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {

        return if(arguments?.getInt(ARG_SECTION_NUMBER)==1) {
            inflater.inflate(R.layout.fragment_cookbook_suggested, container, false)
        } else{
            inflater.inflate(R.layout.fragment_cookbook_all, container, false)
        }
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        val recipeItem = view.findViewById(R.id.recipeItem) as RecyclerView
        fillCuisineList()
        fillDietList()

        if(arguments?.getInt(ARG_SECTION_NUMBER)==1) {
            val spinnerMeal = view.findViewById(R.id.spinner_meal_type) as Spinner
            val spinnerCuisine = view.findViewById(R.id.spinner_cuisine) as Spinner
            val spinnerDiet = view.findViewById(R.id.spinner_diet) as Spinner
            addSuggestItems()


            /** MEAL TYPE FILTER **/
            spinnerMeal.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
                override fun onItemSelected(
                    adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
                ) {
                    addSuggestItems()
                    selectMeal = mealList[position]
                    if (selectMeal != null) {
                        if(selectMeal!="All")
                            removeAllList()
                        recipeItem.layoutManager = LinearLayoutManager(activity)
                        recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
                    }
                }
```

```
    override fun onNothingSelected(arg0: AdapterView<*>) { }
}

val am = ArrayAdapter(activity!!.applicationContext,
    android.R.layout.simple_spinner_item, mealList)
am.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
spinnerMeal.adapter = am

/** CUISINE FILTER **/

spinnerCuisine.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
    override fun onItemSelected(
        adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
    ) {
        addSuggestItems()
        selectCuisine = cuisineList[position]
        if (selectCuisine != null) {
            if(selectCuisine!="All")
                removeAllList()
            recipeItem.layoutManager = LinearLayoutManager(activity)
            recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
        }
    }

    override fun onNothingSelected(arg0: AdapterView<*>) { }
}

val adapter = ArrayAdapter<String>(activity!!.applicationContext,
    android.R.layout.simple_spinner_item, cuisineList)
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
spinnerCuisine.adapter = adapter

/** DIET FILTER **/

spinnerDiet.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
    override fun onItemSelected(
        adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
    ) {
        addSuggestItems()
        selectDiet = dietList[position]
        if (selectDiet != null) {
            if(selectDiet!="All")
                removeAllList()
            recipeItem.layoutManager = LinearLayoutManager(activity)
            recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
        }
    }

    override fun onNothingSelected(arg0: AdapterView<*>) { }
}
```

```
            val adapterDiet = ArrayAdapter<String>(activity!!.applicationContext,
                android.R.layout.simple_spinner_item, dietList)

adapterDiet.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
            spinnerDiet.adapter = adapterDiet

        } else{
            val spinnerMeal = view.findViewById(R.id.spinner_meal_type) as Spinner
            val spinnerCuisine = view.findViewById(R.id.spinner_cuisine) as Spinner
            val spinnerDiet = view.findViewById(R.id.spinner_diet) as Spinner
            addAllItems()

            /** MEAL TYPE FILTER **/
            spinnerMeal.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
                override fun onItemSelected(
                    adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
                ) {
                    addAllItems()
                    selectMeal = mealList[position]
                    if (selectMeal != null) {
                        if(selectMeal!="All")
                            removeAllList()
                        recipeItem.layoutManager = LinearLayoutManager(activity)
                        recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
                    }
                }

                override fun onNothingSelected(arg0: AdapterView<*>) { }
            }

            val am = ArrayAdapter(activity!!.applicationContext,
                android.R.layout.simple_spinner_item, mealList)
            am.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
            spinnerMeal.adapter = am

            /** CUISINE FILTER **/

            spinnerCuisine.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
                override fun onItemSelected(
                    adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
                ) {
                    addAllItems()
                    selectCuisine = cuisineList[position]
                    if (selectCuisine != null) {
                        if(selectCuisine!="All")
                            removeAllList()
                        recipeItem.layoutManager = LinearLayoutManager(activity)
                        recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
                    }
                }
```

```
            override fun onNothingSelected(arg0: AdapterView<*>) { }
        }

        val adapter = ArrayAdapter<String>(activity!!.applicationContext,
            android.R.layout.simple_spinner_item, cuisineList)
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        spinnerCuisine.adapter = adapter

        /** DIET FILTER **/

        spinnerDiet.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
            override fun onItemSelected(
                adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
            ) {
                addAllItems()
                selectDiet = dietList[position]
                if (selectDiet != null) {
                    if(selectDiet!="All")
                        removeAllList()
                    recipeItem.layoutManager = LinearLayoutManager(activity)
                    recipeItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
                }
            }

            override fun onNothingSelected(arg0: AdapterView<*>) { }
        }

        val adapterDiet = ArrayAdapter<String>(activity!!.applicationContext,
            android.R.layout.simple_spinner_item, dietList)

adapterDiet.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        spinnerDiet.adapter = adapterDiet

    }

    /** ADD ON CLICK LISTENER TO RECYCLER VIEW ITEM **/
    var message:String?
    recipeItem.addOnItemTouchListener(
        RecyclerItemClickListener(
            activity!!.applicationContext,
            object : RecyclerItemClickListener.OnItemClickListener {
                override fun onItemClick(view: View, position: Int) {
                    message = list[position]
                    val intent = Intent(activity!!.applicationContext, RecipeItemActivity::class.java)
                    intent.putExtra("recipe", message)
                    startActivity(intent)
                }
            })
    )
}
```

```
private fun addSuggestItems() {
    val context = activity!!.applicationContext
    val db = DataBaseHandler(context)
    val data: MutableList<Recipe>
    list.clear()
    data = db.suggestRecipes()

    for (i in 0..(data.size - 1))
        list.add(data[i].id.toString())
}
private fun addAllItems(){
    val context = activity!!.applicationContext
    val db = DataBaseHandler(context)
    val data : MutableList<Recipe>
    list.clear()
    data = db.readRecipeData()

    for(i in 0..(data.size-1))
        list.add(data[i].id.toString())
}
private fun removeAllList(){
    val context = activity!!.applicationContext
    val db = DataBaseHandler(context)

    val temp : ArrayList<String> = ArrayList()
    for(i in 0..(list.size-1))
      temp.add(list[i])

    list.clear()

    /** MEAL TYPE FILTER **/

    for(i in 0..(temp.size-1)){
        val recipe = db.findRecipe(temp[i].toInt())
        if (selectMeal!= null && selectMeal!= "All"){
            if ( recipe?.mealType == selectMeal)
                list.add(temp[i])
        } else
            list.add(temp[i])
    }

    /** CUISINE FILTER **/
    temp.clear()

    for(i in 0..(list.size-1))
        temp.add(list[i])

    list.clear()

    for(i in 0..(temp.size-1)){
```

```
            val recipe = db.findRecipe(temp[i].toInt())
            if (selectCuisine!= null && selectCuisine!= "All"){
                if ( recipe?.cuisine == selectCuisine)
                    list.add(temp[i])
            } else
                list.add(temp[i])
        }

        /** DIET FILTER **/
        temp.clear()

        for(i in 0..(list.size-1))
            temp.add(list[i])

        list.clear()

        for(i in 0..(temp.size-1)){
            val diet = db.findDietName(temp[i].toInt())
            if (selectDiet!= null && selectDiet!= "All"){
                if ( diet == selectDiet)
                    list.add(temp[i])
            } else
                list.add(temp[i])
        }


    }
    private fun fillCuisineList() {
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        addAllItems()
        cuisineList.add("All")

        for (i in 0..(list.size - 1)) {
            val recipe = db.findRecipe(list[i].toInt())
            var found = false
            for (i in 0..(cuisineList.size - 1)) {
                if (cuisineList[i] == recipe?.cuisine) {
                    found = true
                    break
                }
            }
            if (!found)
                cuisineList.add(recipe!!.cuisine)
        }
    }
    private fun fillDietList(){
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        addAllItems()
        dietList.add("All")
```

```
            for (i in 0..(list.size - 1)) {
                var found = false
                val name = db.findDietName(list[i].toInt())
                for (i in 0..(dietList.size - 1)) {
                    if (dietList[i]==name) {
                        found = true
                        break
                    }
                }
                if (!found && name!=null)
                    dietList.add(name)
            }
        }

        companion object {

            private const val ARG_SECTION_NUMBER = "section_number"

            fun newInstance(sectionNumber: Int): PlaceholderFragment {
                val fragment = PlaceholderFragment()
                val args = Bundle()
                args.putInt(ARG_SECTION_NUMBER, sectionNumber)
                fragment.arguments = args
                return fragment
            }

        }
    }
}
```

3.3.5 – CupboardActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.view.*
import android.widget.AutoCompleteTextView
import com.kitchenette.search.AutoCompleteFoodAdapter
import kotlinx.android.synthetic.main.activity_cupboard.*
import kotlinx.android.synthetic.main.app_bar_cupboard.*
import kotlinx.android.synthetic.main.content_cupboard.*

class CupboardActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private val list : ArrayList<String> = ArrayList()
    private val foodList : ArrayList<Food> = ArrayList()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_cupboard)
        setSupportActionBar(toolbar)

        /*************************** NAV DRAWER ***********************************/

        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)

        /*************************** RECYCLER VIEW ***********************************/
        addFoodItemsId()
        foodItem.layoutManager = LinearLayoutManager(this)
        foodItem.adapter = CupboardAdapter(list, this)

        /*************************** SEARCH METHODS
***********************************/
        addFoodItems()
        val autocompletetextview = findViewById<AutoCompleteTextView>(R.id.autocompletetextview)
        val adapter = AutoCompleteFoodAdapter(this, foodList)
        autocompletetextview?.threshold=1
        autocompletetextview?.setAdapter(adapter)
```

```
        autocompletetextview?.setOnFocusChangeListener {
            _, _ ->
          autocompletetextview.setOnItemClickListener { _, _, _, _ ->
            val db = DataBaseHandler(this)
            val message = db.findFoodName(autocompletetextview.text.toString()).toString()
            val intent = Intent(this@CupboardActivity, FoodItemActivity::class.java)
            intent.putExtra("food", message)
            this.startActivity(intent)
          }
        }

    }

    /*****************************  GENERAL  METHODS
**********************************/
    override fun onBackPressed() {
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    /*****************************  NAVIGATION  DRAWER
**********************************/
    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        // Handle navigation view item clicks here.
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@CupboardActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@CupboardActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@CupboardActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
```

```
        }
        R.id.nav_favourite -> {
            val menuIntent = Intent(this@CupboardActivity, FavouritesActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_barcode -> {
            val menuIntent = Intent(this@CupboardActivity, ScanBarcodeActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_share -> {

        }
      }

      drawer_layout.closeDrawer(GravityCompat.START)
      return true
    }

    /*************************** RECYLER VIEW METHODS
**********************************/
    private fun addFoodItems(){
      val context = this
      val db = DataBaseHandler(context)

      val data = db.readFoodCupboard()

      for(i in 0..(data.size-1))
        foodList.add(data[i])
    }
    private fun addFoodItemsId(){
      val context = this
      val db = DataBaseHandler(context)

      val data = db.readFoodCupboard()

      for(i in 0..(data.size-1))
        list.add(data[i].id.toString())
    }
}
```

3.3.6 – EditFoodActivity.kt

```
package com.kitchenette.kitchenette

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_edit_food.*
import kotlinx.android.synthetic.main.app_bar_edit_food.*
import kotlinx.android.synthetic.main.content_edit_food.*

class EditFoodActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener,
    AdapterView.OnItemSelectedListener{

    var categoryList = arrayOf("Baking & Grains","Beans & Legumes","Beverages",
        "Broths & Soups","Condiments & Sauces","Dairy","Dairy Alternatives",
        "Desserts & Snacks","Fruit","Meat & Poultry","Nuts & Seeds","Oils","Seafood & Fish",
        "Spices, Herbs, Seasonings","Sweeteners","Vegetables")
    var categorySelected : String? = null

    var bitmap: Bitmap? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_edit_food)
        setSupportActionBar(toolbar)

        val context = this
        val db = DataBaseHandler(context)

        val foodMessage: String = intent.getStringExtra("food")
        val id = foodMessage.toInt()

        val food : Food? = db.findFood(foodMessage.toInt())

        foodName.setText(food?.name)
```

```
    if(food?.photo != null){
        bitmap = food.photo
        image.setImageBitmap(bitmap)
    }
    categorySelected = food?.category

    /************************** SPINNER ***********************************/

    foodCategory!!.onItemSelectedListener = this
    val aa = ArrayAdapter(this, android.R.layout.simple_spinner_item, categoryList)
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    foodCategory!!.adapter = aa

    /************************** UPLOAD IMAGE ******************************/
    upload_image.setOnClickListener {
        //check runtime permission
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
            if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
                PackageManager.PERMISSION_DENIED){
                //permission denied
                val permissions = arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE)
                //show popup to request runtime permission
                requestPermissions(permissions, PERMISSION_CODE)
            }
            else{
                pickImageFromGallery() //permission already granted
            }
        }
        else{
            pickImageFromGallery() //system OS is < Marshmallow
        }
    }

    /********************** FLOATING ACTION BUTTON ************************/

    fab.setOnClickListener {
        if(foodName.text.toString().isNotEmpty() && categorySelected!!.isNotEmpty() && bitmap!=
null){
            db.editFood(foodName.text.toString(),categorySelected!!, bitmap!!, id)
            val message = id.toString()
            val intent = Intent(this@EditFoodActivity, FoodItemActivity::class.java)
            intent.putExtra("food", message)
            startActivity(intent)
        }else
            Toast.makeText(context, "Please Fill Out All details", Toast.LENGTH_SHORT).show()
    }

    /************************ NAVIGATION DRAWER ********************************/

    val toggle = ActionBarDrawerToggle(
```

```
        this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
      )
      drawer_layout.addDrawerListener(toggle)
      toggle.syncState()

      nav_view.setNavigationItemSelectedListener(this)
    }

    /************************** NAVIGATION METHODS
*********************************/
    override fun onBackPressed() {
      if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
      } else {
        super.onBackPressed()
      }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
      menuInflater.inflate(R.menu.settings_menu, menu)
      return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
      return when (item.itemId) {
        R.id.action_settings -> true
        else -> super.onOptionsItemSelected(item)
      }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
      when (item.itemId) {
        R.id.nav_cupboard -> {
          val menuIntent = Intent(this@EditFoodActivity, MainActivity::class.java)
          startActivity(menuIntent)
        }
        R.id.nav_cookbook -> {
          val menuIntent = Intent(this@EditFoodActivity, CookbookActivity::class.java)
          startActivity(menuIntent)
        }
        R.id.nav_shopping -> {
          val menuIntent = Intent(this@EditFoodActivity, ShoppingListActivity::class.java)
          startActivity(menuIntent)
        }
        R.id.nav_favourite -> {
          val menuIntent = Intent(this@EditFoodActivity, FavouritesActivity::class.java)
          startActivity(menuIntent)
        }
        R.id.nav_barcode -> {
          val menuIntent = Intent(this@EditFoodActivity, ScanBarcodeActivity::class.java)
```

```
            startActivity(menuIntent)
        }
        R.id.nav_share -> {

        }
      }

      drawer_layout.closeDrawer(GravityCompat.START)
      return true
    }

    /************************** SPINNER METHODS *******************************/
    override fun onItemSelected(arg0: AdapterView<*>, arg1: View, position: Int, id: Long) {
      categorySelected = categoryList[position]
    }

    override fun onNothingSelected(arg0: AdapterView<*>) {}

    /************************** UPLOAD IMAGE METHODS ****************************************/
    private fun pickImageFromGallery() {
      //Intent to pick image
      val intent = Intent(Intent.ACTION_PICK)
      intent.type = "image/*"
      startActivityForResult(intent, IMAGE_PICK_CODE)
    }

    companion object {
      //image pick code
      private const val IMAGE_PICK_CODE = 1000
      //Permission code
      internal const val PERMISSION_CODE = 1001
    }

    //handle requested permission result
    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
      when(requestCode){
        PERMISSION_CODE -> {
          if (grantResults.isNotEmpty() && grantResults[0] ==
            PackageManager.PERMISSION_GRANTED){
            //permission from popup granted
            pickImageFromGallery()
          }
          else{
            //permission from popup denied
            Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
          }
        }
      }
    }
```

```
    //handle result of picked image
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        if (resultCode == Activity.RESULT_OK && requestCode == IMAGE_PICK_CODE){
            image.setImageURI(data?.data)
            image.cropToPadding
            bitmap = MediaStore.Images.Media.getBitmap(this.contentResolver, data?.data)
        }
    }
}
```

3.3.7 – EditRecipeActivty.kt

```
package com.kitchenette.kitchenette

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.Gravity
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.*
import com.kitchenette.search.AutoCompleteFoodAdapter
import kotlinx.android.synthetic.main.activity_edit_recipe.*
import kotlinx.android.synthetic.main.app_bar_edit_recipe.*
import kotlinx.android.synthetic.main.content_edit_recipe.*

class EditRecipeActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener,
    AdapterView.OnItemSelectedListener{

    val list : ArrayList<Food> = ArrayList()
    private val ingredients : ArrayList<String> = ArrayList()
    private val quantityList : ArrayList<Double> = ArrayList()
    private val measureList : ArrayList<String> = ArrayList()
    var bitmap: Bitmap? = null
    private val mealTypeList = arrayOf("Breakfast", "Lunch", "Dinner", "Desserts & Snacks", "Other")
    var selected : String? = null
    private val measurements = arrayOf("cup","dessertspoon","fl. oz",
        "grams","kg","litres","ml","oz","pint","tbsp","tsp", "whole")
    var s : String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_edit_recipe)
        setSupportActionBar(toolbar)

        /****************************** SET RECIPE ITEM
**********************************/
        val context = this
        val db = DataBaseHandler(context)
```

```
val id: String = intent.getStringExtra("recipe")
val recipe = db.findRecipe(id.toInt())

name.setText(recipe?.name)
description.setText(recipe?.description)
method.setText(recipe?.method)
servings.setText(recipe?.servings.toString())
cuisine.setText(recipe?.cuisine)
if(recipe?.photo != null){
   bitmap = recipe.photo
   image.setImageBitmap(bitmap)
}
selected = recipe?.mealType
val d = db.findDietName(id.toInt())
diet.setText(d!!)

addIngredients(id.toInt())
ingredient_list.layoutManager = LinearLayoutManager(this)
ingredient_list.adapter = AddIngredientAdapter(ingredients, quantityList,measureList, this)

/******************** FLOATING ACTION BUTTON ******************************/
fab.setOnClickListener {
   if(name.text.toString().isNotEmpty() && description.text.toString().isNotEmpty() &&
      method.text.toString().isNotEmpty()&& servings.text.toString().isNotEmpty() &&
      cuisine.text.toString().isNotEmpty() && bitmap!= null){

      db.updateRecipe(name.text.toString(), method.text.toString(), cuisine.text.toString(),
         description.text.toString(), selected!!, bitmap!!, servings.text.toString().toInt(), id.toInt())
      db.updateDiet(diet.text.toString(),id.toInt())

      /********************** UPDATE INGREDIENTS ******************************/
      val oldIngredientList = db.readIngredients(id.toInt())

      for (i in 0.. (oldIngredientList.size-1)){
         var found = false
         for(j in 0..(ingredients.size-1)){
            if(ingredients[j].toInt() == oldIngredientList[i].foodID){
               found = true
               break
            }
         }
         if(!found){
            db.removeIngredient(oldIngredientList[i].id)
         }
      }

      for (i in 0..(ingredients.size-1)) {
         var found = false
         for (j in 0.. (oldIngredientList.size-1)){
            if(ingredients[i].toInt() == oldIngredientList[j].foodID){
               db.updateIngredient(oldIngredientList[j].id, quantityList[i], measureList[i])
```

```
                found = true
                break
            }
        }
        if(!found){
            val ing = Ingredients(id.toLong(), ingredients[i].toInt(), quantityList[i], measureList[i])
            db.insertIngredient(ing)
        }

    }

    /****************** OPEN RECIPE ITEM PAGE WITH UPDATED RECIPE ********************/
        val intent = Intent(this@EditRecipeActivity, RecipeItemActivity::class.java)
        intent.putExtra("recipe", id)
        startActivity(intent)
    }else
        Toast.makeText(context, "Please Fill Out All details", Toast.LENGTH_SHORT).show()
}

/************************ NAVIGATION *********************************/
val toggle = ActionBarDrawerToggle(
    this, drawer_layout, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close
)
drawer_layout.addDrawerListener(toggle)
toggle.syncState()
nav_view.setNavigationItemSelectedListener(this)

/*************************** UPLOAD IMAGE ******************************/
upload_image.setOnClickListener {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        if (checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
            PackageManager.PERMISSION_DENIED){
            val permissions = arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE)
            requestPermissions(permissions, PERMISSION_CODE)
        } else
            pickImageFromGallery()
    } else
        pickImageFromGallery()
}

/***************************** SPINNER ***********************************/

meal_type!!.onItemSelectedListener = this
val aa = ArrayAdapter(this, android.R.layout.simple_spinner_item, mealTypeList)
aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
meal_type!!.adapter = aa

/***************************** SEARCH METHODS *********************************/
```

```
        addFoodItems()
        val adapter = AutoCompleteFoodAdapter(this, list)
        autocompletetextview?.threshold=1
        autocompletetextview?.setAdapter(adapter)
        autocompletetextview?.setOnFocusChangeListener {
            _, _ ->
            autocompletetextview.setOnItemClickListener { _, _, _, _ ->
                val message = db.findFoodName(autocompletetextview.text.toString()).toString()
                ingredientPopup(message.toInt())
                autocompletetextview.text.clear()
                val item = findViewById<RecyclerView>(R.id.ingredient_list)
                item.layoutManager = LinearLayoutManager(this)
                item.adapter = AddIngredientAdapter(ingredients, quantityList,measureList, this)
            }
        }
    }

    /************************** NAVIGATION METHODS
****************************************/
    override fun onBackPressed() {
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@EditRecipeActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@EditRecipeActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@EditRecipeActivity, ShoppingListActivity::class.java)
```

```
            startActivity(menuIntent)
        }
        R.id.nav_favourite -> {
            val menuIntent = Intent(this@EditRecipeActivity, FavouritesActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_barcode -> {
            val menuIntent = Intent(this@EditRecipeActivity, ScanBarcodeActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_share -> {

        }
    }

    drawer_layout.closeDrawer(GravityCompat.START)
    return true
  }

  /************************* UPLOAD IMAGE METHODS *************************************/
  private fun pickImageFromGallery() {
    val intent = Intent(Intent.ACTION_PICK)
    intent.type = "image/*"
    startActivityForResult(intent, IMAGE_PICK_CODE)
  }

  companion object {
    private const val IMAGE_PICK_CODE = 1000
    private const val PERMISSION_CODE = 1001
  }

  override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
    when(requestCode){
      PERMISSION_CODE -> {
        if (grantResults.isNotEmpty() && grantResults[0] ==
          PackageManager.PERMISSION_GRANTED){
          pickImageFromGallery()
        }
        else
          Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
      }
    }
  }

  override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    if (resultCode == Activity.RESULT_OK && requestCode == IMAGE_PICK_CODE){
      image.setImageURI(data?.data)
      image.cropToPadding
      bitmap = MediaStore.Images.Media.getBitmap(this.contentResolver, data?.data)
```

```
    }
  }

  /************************** SPINNER METHODS ********************************/
  override fun onItemSelected(parent: AdapterView<*>, view: View, position: Int, id: Long) {
    if(parent.id == R.id.meal_type)
      selected = mealTypeList[position]
    else if(parent.id == R.id.enter_measurement)
      s = measurements[position]
  }

  override fun onNothingSelected(arg0: AdapterView<*>) {}

  /**************************** ADD FOOD ITEMS TO LIST METHODS
**********************************/
  private fun addFoodItems() {
    val context = this
    val db = DataBaseHandler(context)
    val data = db.readFoodData()
    for(i in 0..(data.size-1))
      list.add(data[i])
  }

  private fun addIngredients(id:Int){
    val context = this
    val db = DataBaseHandler(context)
    val data = db.readIngredients(id)

    for(i in 0..(data.size-1)) {
      ingredients.add(data[i].foodID.toString())
      val ingredient = db.findIngredient(data[i].id)
      quantityList.add(ingredient?.quantity!!)
      measureList.add(ingredient.measurement)
    }
  }

  /************************** POPUP METHODS*********************************/
  private fun ingredientPopup(id:Int){
    val context = this
    val db = DataBaseHandler(context)
    val window = PopupWindow(context)
    val view = layoutInflater.inflate(R.layout.popup_add_ingredient,null)

    window.isFocusable = true
    window.isOutsideTouchable = true
    window.update()
    window.width = LinearLayout.LayoutParams.MATCH_PARENT

    window.contentView = view

    view.findViewById<Spinner>(R.id.enter_measurement)!!.onItemSelectedListener = this
```

```
        val a = ArrayAdapter(this,
            android.R.layout.simple_spinner_item, this.measurements)
        a.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        view.findViewById<Spinner>(R.id.enter_measurement)!!.adapter = a


        val food : Food? = db.findFood(id)
        val foodLabel = view.findViewById<TextView>(R.id.food)
        foodLabel.text = food?.name

        val qty = view.findViewById<EditText>(R.id.enter_quantity)


        val add = view.findViewById<ImageButton>(R.id.add_qty_btn)
        add.setOnClickListener{
            ingredients.add(id.toString())
            measureList.add(s!!)
            val amt  = qty!!.text.toString().toDouble()
            quantityList.add(amt)
            window.dismiss()
        }

        val close  = view.findViewById<ImageButton>(R.id.cancel)
        close.setOnClickListener {
            window.dismiss()
        }
        db.close()
        window.showAtLocation(root_layout, Gravity.CENTER,0,0)
    }
}
```

3.3.8- FavouritesActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.design.widget.TabLayout
import android.support.v4.app.Fragment
import android.support.v4.app.FragmentManager
import android.support.v4.app.FragmentPagerAdapter
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.*
import kotlinx.android.synthetic.main.activity_favourites.*
import kotlinx.android.synthetic.main.app_bar_favourites.*


class FavouritesActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private var mSectionsPagerAdapter: FavouritesActivity.SectionsPagerAdapter? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_favourites)
        setSupportActionBar(toolbar)

        /************************ TAB ACTIVITY *************************/
        mSectionsPagerAdapter = SectionsPagerAdapter(supportFragmentManager)
        container.adapter = mSectionsPagerAdapter
        container.addOnPageChangeListener(TabLayout.TabLayoutOnPageChangeListener(tabs))
        tabs.addOnTabSelectedListener(TabLayout.ViewPagerOnTabSelectedListener(container))

        /************************ NAVIGATION DRAWER ***********************/

        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)
    }

    /************************ NAV DRAWER METHODS *************************/

    override fun onBackPressed() {
```

```
    if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
    } else {
        super.onBackPressed()
    }
}

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    menuInflater.inflate(R.menu.settings_menu, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_settings -> true
        else -> super.onOptionsItemSelected(item)
    }
}

override fun onNavigationItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
        R.id.nav_cupboard -> {
            val menuIntent = Intent(this@FavouritesActivity, MainActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_cookbook -> {
            val menuIntent = Intent(this@FavouritesActivity, CookbookActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_shopping -> {
            val menuIntent = Intent(this@FavouritesActivity, ShoppingListActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_favourite -> {
            val menuIntent = Intent(this@FavouritesActivity, FavouritesActivity::class.java)
            startActivity(menuIntent)

        }
        R.id.nav_barcode -> {
            val menuIntent = Intent(this@FavouritesActivity, ScanBarcodeActivity::class.java)
            startActivity(menuIntent)
        }
        R.id.nav_share -> {

        }
    }

    drawer_layout.closeDrawer(GravityCompat.START)
    return true
}
```

```
/************************** TAB ACTIVITY METHODS ****************************/
inner class SectionsPagerAdapter(fm: FragmentManager) : FragmentPagerAdapter(fm) {

    override fun getItem(position: Int): Fragment {
        return FavouritesActivity.PlaceholderFragment.newInstance(position + 1)
    }

    override fun getCount(): Int {
        return 2
    }
}

class PlaceholderFragment : Fragment() {

    private val list : ArrayList<String> = ArrayList()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val rootView = inflater.inflate(R.layout.content_favourites, container, false)
        val favItem = rootView.findViewById(R.id.favItem) as RecyclerView
        var message:String?
        var intent : Intent

        if (arguments?.getInt(ARG_SECTION_NUMBER) == 1) {
            addRecipeItems()
            favItem.layoutManager = LinearLayoutManager(activity)
            favItem.adapter = RecipeAdapter(list, activity!!.applicationContext)
            favItem.addOnItemTouchListener(
                RecyclerItemClickListener(activity!!.applicationContext,
                    object : RecyclerItemClickListener.OnItemClickListener{
                        override fun onItemClick(view: View, position: Int) {
                            message = list[position]
                            intent = Intent(activity!!.applicationContext, RecipeItemActivity::class.java)
                            intent.putExtra("recipe",message)
                            startActivity(intent)
                        }
                    })
            )
        }
        else{
            addFoodItems()
            favItem.layoutManager = LinearLayoutManager(activity)
            favItem.adapter = FoodAdapter(list, activity!!.applicationContext)
            favItem.addOnItemTouchListener(
                RecyclerItemClickListener(activity!!.applicationContext,
                    object : RecyclerItemClickListener.OnItemClickListener {
                        override fun onItemClick(view: View, position: Int) {
                            message = list[position]
                            intent = Intent(activity!!.applicationContext, FoodItemActivity::class.java)
```

```
                    intent.putExtra("food", message)
                    startActivity(intent)
                }
            })
        )
    }
    return rootView
}

private fun addFoodItems(){
    val context = activity!!.applicationContext
    val db = DataBaseHandler(context)
    val data = db.readFoodFavourites()

    for(i in 0..(data.size-1))
        list.add(data[i].id.toString())
    db.close()
}
private fun addRecipeItems(){
    val context = activity!!.applicationContext
    val db = DataBaseHandler(context)
    val data = db.readRecipeFavourites()

    for(i in 0..(data.size-1))
        list.add(data[i].id.toString())
    db.close()
}

companion object {
    private const val ARG_SECTION_NUMBER = "section_number"

    fun newInstance(sectionNumber: Int): FavouritesActivity.PlaceholderFragment {
        val fragment = FavouritesActivity.PlaceholderFragment()
        val args = Bundle()
        args.putInt(ARG_SECTION_NUMBER, sectionNumber)
        fragment.arguments = args
        return fragment
    }
}
  }
}
```

3.3.9 – FoodItemActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.graphics.Bitmap
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_food_item.*
import kotlinx.android.synthetic.main.app_bar_food_item.*
import kotlinx.android.synthetic.main.content_food_item.*

import android.graphics.Color
import android.view.*
import android.widget.*


class FoodItemActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener,
    AdapterView.OnItemSelectedListener {

    var list = arrayOf("cup","dessertspoon","fl. oz",
        "grams","kg","litres","ml","oz","pint","tbsp","tsp", "whole")
    var s : String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_food_item)
        setSupportActionBar(toolbar)

        /********************************** NAV DRAWER
****************************************/
        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)

        /********************************** SET FOOD ITEM
****************************************/

        val context = this
        val db = DataBaseHandler(context)

        val foodMessage: String = intent.getStringExtra("food")
        val id = foodMessage.toInt()
```

```
    var food : Food? = db.findFood(foodMessage.toInt())

    foodName.text = food?.name
    foodItem_category.text = food?.category
    val bitmap: Bitmap? = food?.photo
    image.setImageBitmap(bitmap)
    old_qty.text = food?.quantity.toString()
    old_msr.text = food?.measurement

    if(food?.shoppingList==1)
      shoppingButton.setColorFilter(Color.argb(255, 0, 191, 255))
    else
      shoppingButton.setColorFilter(Color.argb(0, 0, 0, 0))

    if(food?.favourite==1)
      favButton.setColorFilter(Color.argb(255, 0, 191, 255))
    else
      favButton.setColorFilter(Color.argb(0, 0, 0, 0))

    /**********************************                                    BUTTONS
****************************************/

    shoppingButton.setOnClickListener {
      food = if(food?.shoppingList==0) {
        db.addFoodShopping(foodMessage.toInt())
        shoppingButton.setColorFilter(Color.argb(255, 0, 191, 255))
        db.findFood(foodMessage.toInt())
      } else {
        db.removeFoodShopping(foodMessage.toInt())
        shoppingButton.setColorFilter(Color.argb(0, 0, 0, 0))
        db.findFood(foodMessage.toInt())
      }
    }

    favButton.setOnClickListener {
      food = if(food?.favourite==0) {
        db.addFoodFavourites(foodMessage.toInt())
        favButton.setColorFilter(Color.argb(255, 0, 191, 255))
        db.findFood(foodMessage.toInt())
      } else {
        db.removeFoodFavourites(foodMessage.toInt())
        favButton.setColorFilter(Color.argb(0, 0, 0, 0))
        db.findFood(foodMessage.toInt())
      }
    }

    edit_button.setOnClickListener {
      val intent = Intent(this@FoodItemActivity, EditFoodActivity::class.java)
      intent.putExtra("food", foodMessage)
      startActivity(intent)
    }
```

```
addCupboard.setOnClickListener {
    val window = PopupWindow(context)
    val view = layoutInflater.inflate(R.layout.popup_add_quantity,null)

    window.isFocusable = true
    window.isOutsideTouchable = true
    window.update()

    window.contentView = view

    val old_qty = view.findViewById<TextView>(R.id.old_qty)
    old_qty.text  = food?.quantity.toString()
    val old_msr = view.findViewById<TextView>(R.id.old_msr)
    old_msr.text  = food?.measurement

    val spinner = view.findViewById<Spinner>(R.id.enter_measurement)

    spinner!!.onItemSelectedListener = this
    val aa = ArrayAdapter(context, android.R.layout.simple_spinner_item, list)
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
    spinner.adapter = aa

    val qty = view.findViewById<EditText>(R.id.enter_quantity)

    val add = view.findViewById<ImageButton>(R.id.add_qty_btn)
    add.setOnClickListener{
        if(qty.text.toString().isNotEmpty() &&
            s!!.isNotEmpty()){
            db.addFoodQuantity(id, qty.text.toString().toDouble(),s.toString())
            db.addFoodCupboard(id)
            this.recreate()
            window.dismiss()
        }
    }
    window.showAtLocation(root_layout, Gravity.CENTER,0,0)
}

removeCupboard.setOnClickListener {
    val window = PopupWindow(context)
    val view = layoutInflater.inflate(R.layout.popup_remove_quantity,null)

    window.isFocusable = true
    window.isOutsideTouchable = true
    window.update()

    window.contentView = view

    val old_qty = view.findViewById<TextView>(R.id.old_qty)
    old_qty.text  = food?.quantity.toString()
    val old_msr = view.findViewById<TextView>(R.id.old_msr)
```

```
        old_msr.text  = food?.measurement

        val spinner = view.findViewById<Spinner>(R.id.enter_measurement)

        spinner!!.onItemSelectedListener = this
        val aa = ArrayAdapter(context, android.R.layout.simple_spinner_item, list)
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        spinner.adapter = aa

        val qty = view.findViewById<EditText>(R.id.enter_quantity)

        val add = view.findViewById<ImageButton>(R.id.add_qty_btn)
        add.setOnClickListener{
          if(qty.text.toString().isNotEmpty() &&
            s!!.isNotEmpty()){
            db.delFoodQuantity(id, qty.text.toString().toDouble(),s.toString())
            this.recreate()
            window.dismiss()
          }
        }
        window.showAtLocation(root_layout, Gravity.CENTER,0,0)
      }


    }

    /*********************************** NAVIGATION METHODS
*****************************************/
    override fun onBackPressed() {
      if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
      } else {
        super.onBackPressed()
      }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
      // Inflate the menu; this adds items to the action bar if it is present.
      menuInflater.inflate(R.menu.settings_menu, menu)
      return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
      return when (item.itemId) {
        R.id.action_settings -> true
        else -> super.onOptionsItemSelected(item)
      }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
      // Handle navigation view item clicks here.
```

```
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@FoodItemActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@FoodItemActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@FoodItemActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_favourite -> {
                val menuIntent = Intent(this@FoodItemActivity, FavouritesActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_barcode -> {
                val menuIntent = Intent(this@FoodItemActivity, ScanBarcodeActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_share -> {

            }
        }

        drawer_layout.closeDrawer(GravityCompat.START)
        return true
    }

    /*********************************** SPINNER METHODS
*****************************************/
    override fun onItemSelected(arg0: AdapterView<*>, arg1: View, position: Int, id: Long) {
        s = list[position]
    }
    override fun onNothingSelected(arg0: AdapterView<*>) {    }

}
```

3.3.10 – MainActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.content_main.*
import kotlinx.android.synthetic.main.app_bar_main.*

class MainActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(toolbar)


        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)


        loadSearchPage.setOnClickListener {
            val intent = Intent(this@MainActivity, SearchFoodActivity::class.java)
            startActivity(intent)
        }

        loadAddPage.setOnClickListener {
            val intent = Intent(this@MainActivity, AddFoodActivity::class.java)
            startActivity(intent)
        }

        viewCupboardButton.setOnClickListener{
            val intent = Intent(this@MainActivity, CupboardActivity::class.java)
            startActivity(intent)
        }
    }

    override fun onBackPressed() {
```

```
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        // Inflate the menu; this adds items to the action bar if it is present.
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        // Handle navigation view item clicks here.
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@MainActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@MainActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@MainActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_favourite -> {
                val menuIntent = Intent(this@MainActivity, FavouritesActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_barcode -> {
                val menuIntent = Intent(this@MainActivity, ScanBarcodeActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_share -> {

            }
        }

        drawer_layout.closeDrawer(GravityCompat.START)
        return true
    }
```

3.3.11 – Measurements.kt

```
package com.kitchenette.kitchenette

class Measurements{
    var quantity : Double = 0.0
    var type : String = ""
    var desired : String = ""

    private val cupToMl : Double = 250.0
    private val pintToMl : Double = 570.0
    private val litreToMl : Double = 1000.0
    private val flOzToMl : Double = 29.6
    private val tspToMl : Double = 5.0
    private val tbspToMl : Double = 15.0
    private val dessertspoonToMl : Double = 10.0

    private val mlToGrams : Double = 1.0
    private val ozToGrams : Double = 30.0
    private val kgToGrams : Double = 1000.0


    constructor(quantity:Double, type:String, desired:String){
        this.quantity = quantity
        this.type = type
        this.desired = desired
    }

    fun convert(): Double?{
        return when {
            this.desired == "litres" -> toLitres()
            this.desired == "kg" -> toKG()
            this.desired == "grams" -> convertGrams()
            this.desired == "ml" -> convertMl()
            this.desired == "whole" -> whole()
            else -> null
        }
    }

    private fun whole(): Double? {
        return this.quantity
    }

    private fun toKG(): Double? {
        this.quantity = this.convertGrams()!!
        this.quantity = this.quantity / kgToGrams
        this.type = "kg"
        return this.quantity
    }

    private fun toLitres(): Double? {
        this.quantity = this.convertGrams()!!
```

```
        this.quantity = this.quantity / litreToMl
        this.type = "litres"
        return this.quantity
    }


    private fun convertMl():Double?{
        return when {
            this.type == "cups" -> cupml(this.quantity)
            this.type == "pint" -> pintml(this.quantity)
            this.type == "litres" -> litreml(this.quantity)
            this.type == "tsp" -> tspml(this.quantity)
            this.type == "tbsp" -> tbspml(this.quantity)
            this.type == "fl. oz" -> flozml(this.quantity)
            this.type == "dessertspoon" -> dspml(this.quantity)
            this.type == "grams" -> gramsml(this.quantity)
            else -> null
        }
    }

    private fun gramsml(quantity: Double): Double? {
        this.quantity = quantity * mlToGrams
        this.type = "ml"
        return this.quantity
    }

    private fun convertGrams():Double?{
        return when {
            this.type == "ml" -> mlgrams(this.quantity)
            this.type == "oz" -> ozgrams(this.quantity)
            this.type == "kg" -> kggrams(this.quantity)
            else -> convertMl()
        }
    }

    private fun kggrams(quantity: Double): Double? {
        this.quantity = quantity * kgToGrams
        this.type = "grams"
        return this.quantity
    }

    private fun ozgrams(quantity: Double): Double? {
        this.quantity = quantity * ozToGrams
        this.type = "grams"
        return this.quantity
    }

    private fun mlgrams(quantity: Double): Double? {
        this.quantity = quantity * mlToGrams
        this.type = "grams"
        return this.quantity
```

```
    }

    private fun dspml(quantity: Double): Double? {
        this.quantity = quantity * dessertspoonToMl
        this.type = "ml"
        return this.quantity
    }

    private fun flozml(quantity: Double): Double? {
        this.quantity = quantity * flOzToMl
        this.type = "ml"
        return this.quantity
    }

    private fun tbspml(quantity: Double): Double? {
        this.quantity = quantity * tbspToMl
        this.type = "ml"
        return this.quantity
    }

    private fun tspml(quantity: Double): Double? {
        this.quantity = quantity * tspToMl
        this.type = "ml"
        return this.quantity
    }

    private fun litreml(quantity: Double): Double? {
        this.quantity = quantity * litreToMl
        this.type = "ml"
        return this.quantity
    }

    private fun pintml(quantity: Double): Double? {
        this.quantity = quantity * pintToMl
        this.type = "ml"
        return this.quantity
    }

    private fun cupml(quantity:Double):Double?{
        this.quantity = quantity * cupToMl
        this.type = "ml"
        return this.quantity
    }
}
```

3.3.12 – RecipeItemActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.graphics.Bitmap
import android.graphics.Color
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.*
import android.widget.*
import kotlinx.android.synthetic.main.activity_recipe_item.*
import kotlinx.android.synthetic.main.app_bar_recipe_item.*
import kotlinx.android.synthetic.main.content_recipe_item.*

class RecipeItemActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    private val foodList : ArrayList<String> = ArrayList()
    private val idList : ArrayList<String> = ArrayList()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_recipe_item)
        setSupportActionBar(toolbar)

        val context = this
        val db = DataBaseHandler(context)
        val id: String = intent.getStringExtra("recipe")
        addIngredients(id.toInt())

        /***************FLOATING ACTION BUTTONS ******************/
        // "Make This" Button
        fab.setOnClickListener {
            makeThisPopup(id.toInt())
        }
        /********************** NAVIGATION **********************/
        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)

        /*************  FILL PAGE ******************************/
```

```
    var recipe : Recipe? = db.findRecipe(id.toInt())
    recipeName.text = recipe?.name
    serving.text = recipe?.servings.toString()
    cuisine.text = recipe?.cuisine
    description.text = recipe?.description
    method.text = recipe?.method

    val bitmap: Bitmap? = recipe?.photo
    image.setImageBitmap(bitmap)

    ingredientItem.layoutManager = LinearLayoutManager(this)
    ingredientItem.adapter = IngredientAdapter(foodList, idList, this)

    /************************* Buttons ***************************/

    if(recipe?.favourite==1)
      favButton.setColorFilter(Color.argb(255, 0, 191, 255))
    else
      favButton.setColorFilter(Color.argb(0, 0, 0, 0))

    shoppingButton.setOnClickListener {
      shoppingPopup()
    }

    favButton.setOnClickListener {
      recipe = if(recipe?.favourite==0) {
        db.addRecipeFavourites(id.toInt())
        favButton.setColorFilter(Color.argb(255, 0, 191, 255))
        db.findRecipe(id.toInt())
      } else {
        db.removeRecipeFavourites(id.toInt())
        favButton.setColorFilter(Color.argb(0, 0, 0, 0))
        db.findRecipe(id.toInt())
      }
    }

    edit_button.setOnClickListener {
      val intent = Intent(this@RecipeItemActivity, EditRecipeActivity::class.java)
      intent.putExtra("recipe", id)
      startActivity(intent)
    }

    db.close()
  }

  /*************** NAVIGATION METHODS *****************/
  override fun onBackPressed() {
    if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
      drawer_layout.closeDrawer(GravityCompat.START)
    } else {
      super.onBackPressed()
```

```
    }
  }

  override fun onCreateOptionsMenu(menu: Menu): Boolean {
    menuInflater.inflate(R.menu.settings_menu, menu)
    return true
  }

  override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
      R.id.action_settings -> true
      else -> super.onOptionsItemSelected(item)
    }
  }

  override fun onNavigationItemSelected(item: MenuItem): Boolean {
    when (item.itemId) {
      R.id.nav_cupboard -> {
        val menuIntent = Intent(this@RecipeItemActivity, MainActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_cookbook -> {
        val menuIntent = Intent(this@RecipeItemActivity, CookbookActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_shopping -> {
        val menuIntent = Intent(this@RecipeItemActivity, ShoppingListActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_favourite -> {
        val menuIntent = Intent(this@RecipeItemActivity, FavouritesActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_barcode -> {
        val menuIntent = Intent(this@RecipeItemActivity, ScanBarcodeActivity::class.java)
        startActivity(menuIntent)
      }
      R.id.nav_share -> {

      }
    }
    drawer_layout.closeDrawer(GravityCompat.START)
    return true
  }

  /*************** INGREDIENTS METHODS ****************/
  private fun addIngredients(id:Int){
    val context = this
    val db = DataBaseHandler(context)
    val data = db.readIngredients(id)
```

```
        for(i in 0..(data.size-1)) {
            foodList.add(data[i].foodID.toString())
            idList.add(data[i].id.toString())
        }
    }

    private fun deleteQuantity(id:Int){
        val context = this
        val db = DataBaseHandler(context)
        val data = db.readIngredients(id)
        for(i in 0..(data.size-1)) {
            db.removeQuantityCupboard(data[i].id, data[i].foodID)
        }
    }

    /**************** POPUP METHODS *****************/

    private fun shoppingPopup(){
        val context = this
        val db = DataBaseHandler(context)
        val window = PopupWindow(context)
        val view = layoutInflater.inflate(R.layout.popup_add_shopping,null)

        window.isFocusable = true
        window.isOutsideTouchable = true
        window.width = LinearLayout.LayoutParams.MATCH_PARENT
        window.update()

        window.contentView = view

        val foodItem = view.findViewById<RecyclerView>(R.id.ingredient_list)
        foodItem.layoutManager = LinearLayoutManager(this)
        foodItem.adapter = ShoppingPopupAdapter(foodList,  this)

        //window.dismiss()
        val add = view.findViewById<ImageButton>(R.id.add_all)
        add.setOnClickListener{
            for(i in 0..(foodList.size-1))
                db.addFoodShopping(foodList[i].toInt())
            window.dismiss()
        }

        val close  = view.findViewById<ImageButton>(R.id.add_none)
        close.setOnClickListener {
            window.dismiss()
        }
        db.close()
        window.showAtLocation(root_layout, Gravity.CENTER,0,0)
    }

    private fun makeThisPopup(id:Int){
```

```
        val context = this
        val db = DataBaseHandler(context)
        val window = PopupWindow(context)
        val view = layoutInflater.inflate(R.layout.popup_make_this,null)

        window.isFocusable = true
        window.isOutsideTouchable = true
        window.update()

        window.contentView = view

        val make = view.findViewById<ImageButton>(R.id.make)
        make.setOnClickListener{
            deleteQuantity(id)
            window.dismiss()
            shoppingPopup()
        }

        val close  = view.findViewById<ImageButton>(R.id.no_make)
        close.setOnClickListener {
            window.dismiss()
        }
        db.close()
        window.showAtLocation(root_layout, Gravity.CENTER,0,0)
    }
}
```

3.3.13 – SearchFoodActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Context
import android.content.Intent
import android.content.res.Resources
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.v4.app.Fragment
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.support.v7.widget.ThemedSpinnerAdapter
import android.view.*
import android.widget.*
import com.kitchenette.search.AutoCompleteFoodAdapter
import kotlinx.android.synthetic.main.activity_search_food.*
import kotlinx.android.synthetic.main.app_bar_search_food.*
import kotlinx.android.synthetic.main.list_item.view.*

class SearchFoodActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {

    val list : ArrayList<Food> = ArrayList()


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_search_food)
        setSupportActionBar(toolbar)

        /************************** TAB ACTIVITY **********************************/
        supportActionBar?.setDisplayShowTitleEnabled(false)

        spinner.adapter = MyAdapter(
            toolbar.context,
            arrayOf("All","Baking & Grains","Beans & Legumes","Beverages",
                "Broths & Soups","Condiments & Sauces","Dairy","Dairy Alternatives",
                "Desserts & Snacks","Fruit","Meat & Poultry","Nuts & Seeds","Oils","Seafood & Fish",
                "Spices, Herbs & Seasonings","Sweeteners","Vegetables")
        )

        spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
            override fun onItemSelected(parent: AdapterView<*>, view: View?, position: Int, id: Long) {
                supportFragmentManager.beginTransaction()
                    .replace(R.id.container, PlaceholderFragment.newInstance(position + 1))
                    .commit()
            }
```

```
        override fun onNothingSelected(parent: AdapterView<*>) {}
    }

    /**************************** FLOATING BUTTON ***********************************/

    fab.setOnClickListener {
        val intent = Intent(this@SearchFoodActivity, AddFoodActivity::class.java)
        startActivity(intent)
    }

    /**************************** NAV DRAWER ***********************************/

    val toggle = ActionBarDrawerToggle(
        this, drawer_layout, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close
    )
    drawer_layout.addDrawerListener(toggle)
    toggle.syncState()

    nav_view.setNavigationItemSelectedListener(this)

    /**************************** SEARCH METHODS ***********************************/

    addFoodItems()

    val adapter = AutoCompleteFoodAdapter(this, list)
    autocompletetextview?.threshold=1
    autocompletetextview?.setAdapter(adapter)
    autocompletetextview?.setOnFocusChangeListener {
        _, _ ->
        autocompletetextview.setOnItemClickListener { _, _, _, _ ->
            val db = DataBaseHandler(this)
            val message = db.findFoodName(autocompletetextview.text.toString()).toString()
            val intent = Intent(this@SearchFoodActivity, FoodItemActivity::class.java)
            intent.putExtra("food", message)
            this.startActivity(intent)
        }
    }

}

/**************************** STANDARD METHODS ***********************************/
override fun onBackPressed() {
    if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
    } else {
        super.onBackPressed()
    }
```

```
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        // Inflate the menu; this adds items to the action bar if it is present.
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    /**************************** NAV DRAWER METHODS ********************************/
    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        // Handle navigation view item clicks here.
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@SearchFoodActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@SearchFoodActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@SearchFoodActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_favourite -> {
                val menuIntent = Intent(this@SearchFoodActivity, FavouritesActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_barcode -> {
                val menuIntent = Intent(this@SearchFoodActivity, ScanBarcodeActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_share -> {

            }
        }

        drawer_layout.closeDrawer(GravityCompat.START)
        return true
    }

    /****************************ADD FOOD ITEMS TO LIST METHODS*********************************/
```

```
    private fun addFoodItems() {
        val context = this
        val db = DataBaseHandler(context)

        val data = db.readFoodData()

        for(i in 0..(data.size-1)){
            list.add(data[i])
        }
    }

    /**************************** ADAPTER CLASS *********************************/
    private class MyAdapter(context: Context, objects: Array<String>) :
        ArrayAdapter<String>(context, R.layout.list_item, objects), ThemedSpinnerAdapter {
        private val dropDownHelper: ThemedSpinnerAdapter.Helper =
ThemedSpinnerAdapter.Helper(context)

        override fun getDropDownView(position: Int, convertView: View?, parent: ViewGroup): View {
            val view: View

            view = if (convertView == null) {
                // Inflate the drop down using the helper's LayoutInflater
                val inflater = dropDownHelper.dropDownViewInflater
                inflater.inflate(R.layout.list_item, parent, false)
            } else {
                convertView
            }

            view.text1.text = getItem(position)

            return view
        }

        override fun getDropDownViewTheme(): Resources.Theme? {
            return dropDownHelper.dropDownViewTheme
        }

        override fun setDropDownViewTheme(theme: Resources.Theme?) {
            dropDownHelper.dropDownViewTheme = theme
        }
    }

    /**************************** FRAGMENT CLASS *********************************/

    class PlaceholderFragment : Fragment() {

        val list : ArrayList<String> = ArrayList()

        override fun onCreateView(
            inflater: LayoutInflater, container: ViewGroup?,
            savedInstanceState: Bundle?
```

```
    ): View? {
        val rootView = inflater.inflate(R.layout.content_search_food, container, false)
        val foodItem = rootView.findViewById(R.id.foodItem) as RecyclerView
        val foodMessage1 = rootView.findViewById(R.id.all_food_message1) as TextView
        val foodMessage2 = rootView.findViewById(R.id.all_food_message2) as TextView
        val foodMessage3 = rootView.findViewById(R.id.all_food_message3) as TextView
        val foodMessage4 = rootView.findViewById(R.id.all_food_message4) as TextView
        foodItem.layoutManager = LinearLayoutManager(activity)
        foodItem.setHasFixedSize(true)

        val categoryArray = arrayOf("All","Baking & Grains","Beans & Legumes","Beverages",
            "Broths & Soups","Condiments & Sauces","Dairy","Dairy Alternatives",
            "Desserts & Snacks","Fruit","Meat & Poultry","Nuts & Seeds","Oils","Seafood & Fish",
            "Spices, Herbs & Seasonings","Sweeteners","Vegetables")

        if(arguments?.getInt(SearchFoodActivity.PlaceholderFragment.ARG_SECTION_NUMBER)!=1) {
            val position =
(arguments?.getInt(SearchFoodActivity.PlaceholderFragment.ARG_SECTION_NUMBER)!! -1)
            val category = categoryArray[position]
            addCategoryFoodItems(category)
            foodMessage1.visibility = LinearLayout.GONE
            foodMessage2.visibility = LinearLayout.GONE
            foodMessage3.visibility = LinearLayout.GONE
            foodMessage4.visibility = LinearLayout.GONE
        }

        foodItem.adapter = FoodAdapter(list, activity!!.applicationContext)

        var message:String?
        foodItem.addOnItemTouchListener(
            RecyclerItemClickListener(
                activity!!.applicationContext,
                object : RecyclerItemClickListener.OnItemClickListener {
                    override fun onItemClick(view: View, position: Int) {
                        message = list[position]
                        val intent = Intent(activity!!.applicationContext, FoodItemActivity::class.java)
                        intent.putExtra("food", message)
                        startActivity(intent)
                    }
                })
        )

        return rootView
    }

    /********************** ADD FOOD ITEMS TO LIST METHODS
*****************************/

    private fun addCategoryFoodItems(cat:String) {
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
```

```
            val data = db.readFoodCategory(cat)

            for(i in 0..(data.size-1)){
                list.add(data[i].id.toString())
            }
        }

        companion object {

            private const val ARG_SECTION_NUMBER = "section_number"

            fun newInstance(sectionNumber: Int): PlaceholderFragment {
                val fragment = PlaceholderFragment()
                val args = Bundle()
                args.putInt(ARG_SECTION_NUMBER, sectionNumber)
                fragment.arguments = args
                return fragment
            }
        }
    }
}
```

3.3.14 – ShoppingListActivity.kt

```
package com.kitchenette.kitchenette

import android.content.Intent
import android.os.Bundle
import android.support.design.widget.NavigationView
import android.support.design.widget.TabLayout
import android.support.v4.app.Fragment
import android.support.v4.app.FragmentManager
import android.support.v4.app.FragmentPagerAdapter
import android.support.v4.view.GravityCompat
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.view.*
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Spinner
import kotlinx.android.synthetic.main.activity_shopping.*
import kotlinx.android.synthetic.main.app_bar_shopping_list.*
import android.R.attr.country
import android.widget.Toast
import kotlinx.android.synthetic.main.content_barcode_history.*


class ShoppingListActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener
{

    private var sectionsPagerAdapter: ShoppingListActivity.SectionsPagerAdapter? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_shopping)
        setSupportActionBar(toolbar)

        /************************** TAB ACTIVITY *********************************/
        sectionsPagerAdapter = SectionsPagerAdapter(supportFragmentManager)
        container.adapter = sectionsPagerAdapter

        container.addOnPageChangeListener(TabLayout.TabLayoutOnPageChangeListener(tabs))
        tabs.addOnTabSelectedListener(TabLayout.ViewPagerOnTabSelectedListener(container))

        /************************** NAVIGATION DRAWER
**********************************/

        val toggle = ActionBarDrawerToggle(
            this, drawer_layout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close
        )
```

```
        drawer_layout.addDrawerListener(toggle)
        toggle.syncState()

        nav_view.setNavigationItemSelectedListener(this)
    }

    /*************************** NAVIGATION DRAWER METHODS
**********************************/

    override fun onBackPressed() {
        if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
            drawer_layout.closeDrawer(GravityCompat.START)
        } else {
            super.onBackPressed()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
        menuInflater.inflate(R.menu.settings_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_settings -> true
            else -> super.onOptionsItemSelected(item)
        }
    }

    override fun onNavigationItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.nav_cupboard -> {
                val menuIntent = Intent(this@ShoppingListActivity, MainActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_cookbook -> {
                val menuIntent = Intent(this@ShoppingListActivity, CookbookActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_shopping -> {
                val menuIntent = Intent(this@ShoppingListActivity, ShoppingListActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_favourite -> {
                val menuIntent = Intent(this@ShoppingListActivity, FavouritesActivity::class.java)
                startActivity(menuIntent)
            }
            R.id.nav_barcode -> {
                val menuIntent = Intent(this@ShoppingListActivity, ScanBarcodeActivity::class.java)
                startActivity(menuIntent)
            }
```

```
        R.id.nav_share -> {

        }
      }

      drawer_layout.closeDrawer(GravityCompat.START)
      return true
    }


    /**************************** TAB SECTIONS CLASS ***********************************/
    inner class SectionsPagerAdapter(fm: FragmentManager) : FragmentPagerAdapter(fm) {

      override fun getItem(position: Int): Fragment {
        return PlaceholderFragment.newInstance(position + 1)
      }

      override fun getCount(): Int { return 2 }
    }

    /**************************** FRAGMENT CLASS ***********************************/
    class PlaceholderFragment : Fragment() {

      private var list : ArrayList<String> = ArrayList()
      private var categoryList : Array<String>  = arrayOf("All","Baking & Grains",
        "Beans & Legumes","Beverages", "Broths & Soups","Condiments & Sauces",
        "Dairy","Dairy Alternatives", "Desserts & Snacks","Fruit","Meat & Poultry",
        "Nuts & Seeds","Oils","Seafood & Fish", "Spices, Herbs & Seasonings","Sweeteners","Vegetables")
      private var selected: String? = ""

      override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
      ): View? {
        return if(arguments?.getInt(ARG_SECTION_NUMBER)==1) {
          inflater.inflate(R.layout.fragment_shopping_list, container, false)
        } else{
          inflater.inflate(R.layout.fragment_bought_list, container, false)
        }
      }

      override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        if(arguments?.getInt(ARG_SECTION_NUMBER)==1) {
          list.clear()
          val foodItem = view.findViewById(R.id.foodItem) as RecyclerView
          val spinner = view.findViewById(R.id.spinner_shop) as Spinner

          spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
```

```
            override fun onItemSelected(
                adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
            ) {
                selected = categoryList[position]
                if (selected != null) {
                    if(selected.equals("All")){
                        addShoppingItems()
                    }
                    else{
                        addShoppingCategoryItems()
                    }
                    foodItem.layoutManager = LinearLayoutManager(activity)
                    foodItem.adapter = ShoppingAdapter(list, activity!!.applicationContext)
                }
            }

            override fun onNothingSelected(arg0: AdapterView<*>) { }
        }

        val aa = ArrayAdapter(activity!!.applicationContext, android.R.layout.simple_spinner_item,
categoryList)
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        spinner.adapter = aa
    }
    else{
        list.clear()
        val foodItem = view.findViewById(R.id.foodItem) as RecyclerView
        val spinner = view.findViewById(R.id.spinner_bought) as Spinner

        spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
            override fun onItemSelected(
                adapterView: AdapterView<*>, arg1: View, position: Int, id: Long
            ) {
                selected = categoryList[position]
                if (selected != null) {
                    if(selected.equals("All")){
                        addBoughtItems()
                    }
                    else{
                        addBoughtCategoryItems()
                    }
                    foodItem.layoutManager = LinearLayoutManager(activity)
                    foodItem.adapter = BoughtAdapter(list, activity!!.applicationContext)
                }
            }

            override fun onNothingSelected(arg0: AdapterView<*>) { }
        }

        val aa = ArrayAdapter(activity!!.applicationContext, android.R.layout.simple_spinner_item,
categoryList)
```

```
            aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
            spinner.adapter = aa
        }
    }

    private fun addBoughtCategoryItems(){
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        val data : MutableList<Food>
        data = db.readBoughtCategory(selected!!)
        list.clear()

        for(i in 0..(data.size-1))
            list.add(data[i].id.toString())
    }

    private fun addBoughtItems(){
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        val data : MutableList<Food>
        data = db.readBought()
        list.clear()

        for(i in 0..(data.size-1))
            list.add(data[i].id.toString())
    }

    private fun addShoppingCategoryItems() {
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        val data: MutableList<Food>
        data = db.readShoppingCategory(selected!!)
        list.clear()

        for (i in 0..(data.size - 1))
            list.add(data[i].id.toString())
    }

    private fun addShoppingItems() {
        val context = activity!!.applicationContext
        val db = DataBaseHandler(context)
        val data: MutableList<Food>
        data = db.readShopping()
        list.clear()

        for (i in 0..(data.size - 1))
            list.add(data[i].id.toString())
    }

    companion object {
        private const val ARG_SECTION_NUMBER = "section_number"
```

```
        fun newInstance(sectionNumber: Int): PlaceholderFragment {
            val fragment = PlaceholderFragment()
            val args = Bundle()
            args.putInt(ARG_SECTION_NUMBER, sectionNumber)
            fragment.arguments = args
            return fragment
        }

    }
  }

}
```

Section 4 – XML Code

This section covers the code for the graphical user interface (GUI) design of the application. In some cases, multiple identical files for each activity had to be created. Only one of each of these files will be found in this document. All remaining source code can be found at the GitHub link in the description.

Section 4.1 – Navigation and Menu Files

4.1.1 – main activity file

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

  <include
      layout="@layout/app_bar_add_recipe"
      android:layout_width="match_parent"
      android:layout_height="match_parent"/>

  <android.support.design.widget.NavigationView
      android:id="@+id/nav_view"
      android:layout_width="wrap_content"
      android:layout_height="match_parent"
      android:layout_gravity="start"
      android:fitsSystemWindows="true"
      app:headerLayout="@layout/nav_header"
      app:menu="@menu/activity_drawer"/>

</android.support.v4.widget.DrawerLayout>
```

4.1.2 – App Bar

```
<?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=".SearchFoodActivity">

  <android.support.design.widget.AppBarLayout
      android:layout_height="wrap_content"
      android:layout_width="match_parent"
      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"/>
    <Spinner
        android:id="@+id/spinner"
        app:popupTheme="@style/AppTheme.PopupOverlay"
        android:backgroundTint="@color/secondaryTextColor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_gravity="center">
      <ImageView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginStart="5dp"
          android:layout_marginTop="5dp"
          android:padding="10dp"
          android:tint="@color/secondaryTextColor"
          android:src="@android:drawable/ic_menu_search"
          android:background="@null"/>
      <AutoCompleteTextView
          android:id="@+id/autocompletetextview"
          android:layout_margin="10dp"
          android:padding="10dp"
          android:textColor="@android:color/primary_text_light"
          android:hint="Search Food"
          android:background="@android:color/background_light"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"/>
```

```
        </LinearLayout>


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

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

    <android.support.v4.widget.NestedScrollView
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"/>

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

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

4.1.3 – Nav Header

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@color/primaryLightColor"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    android:orientation="vertical"
    android:gravity="bottom">
  <ImageView
      android:layout_width="96dp"
      android:layout_height="96dp"
      android:paddingTop="@dimen/nav_header_vertical_spacing"
      app:srcCompat="@mipmap/ic_launcher_round"
      android:contentDescription="@string/nav_header_desc"
      android:id="@+id/imageView"/>
  <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingTop="@dimen/nav_header_vertical_spacing"
      android:text="@string/nav_header_title"
      android:textAppearance="@style/TextAppearance.AppCompat.Body1"/>
  <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/nav_header_subtitle"
      android:id="@+id/textView"/>
</LinearLayout>
```

4.1.4 – Activity Drawer

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

  <group android:checkableBehavior="single">
    <item
        android:id="@+id/nav_cupboard"
        android:icon="@drawable/ic_menu_cupboard"
        android:title="Cupboard"/>
    <item
        android:id="@+id/nav_cookbook"
        android:icon="@drawable/ic_menu_cookbook"
        android:title="Cookbook"/>
    <item
        android:id="@+id/nav_shopping"
        android:icon="@drawable/ic_menu_shopping"
        android:title="Shopping List"/>
    <item
        android:id="@+id/nav_favourite"
        android:icon="@drawable/ic_menu_favourite"
        android:title="Favourites"/>
    <item
        android:id="@+id/nav_barcode"
        android:icon="@drawable/ic_menu_barcode"
        android:title="Scan a Barcode"/>
  </group>

  <item android:title="Communicate">
    <menu>
      <item
          android:id="@+id/nav_share"
          android:icon="@drawable/ic_menu_share"
          android:title="Share"/>
    </menu>
  </item>

</menu>
```

4.1.5 – List Item

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
          android:orientation="vertical"
          android:layout_width="match_parent"
          android:layout_height="wrap_content">
   <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="2">
      <de.hdodenhof.circleimageview.CircleImageView
            android:id="@+id/image_food_icon"
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:layout_margin="10dp"
            android:layout_weight="0.01"
            android:src="@mipmap/ic_launcher_round" />

      <TextView
            android:id="@+id/tv_foodShop"
            android:textColor="@color/primaryTextColor"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:padding="20dp"
            android:layout_weight="0.8"
            android:text="FOOD"/>
      <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="2"
            android:layout_weight="1.2">
        <ImageButton
              android:id="@+id/shopCheck"
              android:tint="@color/secondaryDarkColor"
              android:layout_width="match_parent"
              android:layout_height="wrap_content"
              android:padding="10dp"
              android:layout_margin="10dp"
              android:layout_weight="1"
              android:src="@drawable/ic_check"
              android:background="@null"/>
        <ImageButton
              android:id="@+id/removeList"
              android:tint="@color/secondaryDarkColor"
              android:layout_width="match_parent"
              android:layout_height="wrap_content"
              android:padding="10dp"
              android:layout_margin="10dp"
```

```
            android:layout_weight="1"
            android:src="@drawable/ic_remove"
            android:background="@null"/>
        </LinearLayout>


    </LinearLayout>
</LinearLayout>
```

Section 4.2 – Content Files

4.2.1 – Add Food Content

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_add_food"
    tools:context=".AddFoodActivity"
    android:orientation="vertical">
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical"
      android:weightSum="100">
    <de.hdodenhof.circleimageview.CircleImageView
        android:layout_width="match_parent"
        android:layout_height="150pt"
        android:id="@+id/image"
        android:layout_weight="60"
        android:scaleType="centerCrop"
        android:src="@mipmap/ic_launcher"/>
    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/upload_image"
        android:layout_gravity="right"
        android:padding="10dp"
        android:layout_marginRight="10dp"
        android:background="@null"
        android:src="@android:drawable/ic_menu_camera"/>
    <EditText
        android:id="@+id/foodName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="10"
        android:layout_margin="10dp"
        android:padding="10dp"
        android:hint="Food Name"/>
    <Spinner
        android:id="@+id/foodCategory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="20"
        android:layout_margin="10dp"
        android:padding="10dp"/>
```

```
    </LinearLayout>

</ScrollView>
```

4.2.2 – Add Recipe Content

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/root_layout"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_add_recipe"
    tools:context=".AddRecipeActivity">
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical">
    <de.hdodenhof.circleimageview.CircleImageView
        android:layout_width="match_parent"
        android:layout_height="100pt"
        android:id="@+id/image"
        android:scaleType="centerCrop"
        android:src="@mipmap/ic_launcher"/>
    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/upload_image"
        android:layout_gravity="right"
        android:padding="10dp"
        android:layout_marginRight="10dp"
        android:background="@null"
        android:src="@android:drawable/ic_menu_camera"/>
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
        android:hint="Recipe Name"/>
    <EditText
        android:id="@+id/description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
        android:hint="Description"/>
    <EditText
        android:id="@+id/method"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
        android:hint="Method"/>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
        android:weightSum="3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
      <EditText
          android:id="@+id/servings"
          android:hint="Servings"
          android:inputType="number"
          android:padding="10dp"
          android:layout_weight="1"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"/>
      <EditText
          android:id="@+id/cuisine"
          android:padding="10dp"
          android:layout_weight="1"
          android:hint="Cuisine"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"/>
      <EditText
          android:id="@+id/diet"
          android:padding="10dp"
          android:layout_weight="1"
          android:hint="Diet Type"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
      <TextView
          android:text="Category: "
          android:textSize="18sp"
          android:layout_marginRight="10dp"
          android:layout_marginLeft="10dp"
          android:layout_marginTop="5dp"
          android:padding="10dp"
          android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"/>
        <Spinner
            android:id="@+id/meal_type"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="5dp"
            android:padding="10dp"/>
    </LinearLayout>
    <TextView
        android:text="Add Your Ingredients: "
        android:textSize="20sp"
        android:textColor="@color/secondaryDarkColor"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="5dp"
            android:padding="10dp"
            android:tint="@color/secondaryDarkColor"
            android:src="@android:drawable/ic_menu_search"
            android:background="@null"/>
        <AutoCompleteTextView
            android:id="@+id/autocompletetextview"
            android:layout_marginRight="10dp"
            android:layout_marginTop="5dp"
            android:padding="10dp"
            android:textColor="@android:color/primary_text_light"
            android:hint="Search Food"
            android:background="@android:color/background_light"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </LinearLayout>
    <android.support.v7.widget.RecyclerView
        android:id="@+id/ingredient_list"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="15dp"
        android:padding="10dp"
```

```
            android:scrollbarAlwaysDrawVerticalTrack="true"
            android:layout_width="match_parent"
            android:layout_height="200dp" />
    </LinearLayout>
</ScrollView>
```

4.2.3 - Food Item Content

```
<?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"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      app:layout_behavior="@string/appbar_scrolling_view_behavior"
      tools:showIn="@layout/app_bar_food_item"
      android:id="@+id/root_layout"
      tools:context=".FoodItemActivity">
   <LinearLayout
         android:orientation="vertical"
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:weightSum="100">
      <de.hdodenhof.circleimageview.CircleImageView
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_weight="30"
            android:id="@+id/image"
            android:src="@mipmap/ic_launcher"/>
      <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="end">
         <Button
               android:id="@+id/edit_button"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:text = "Edit"
               android:layout_gravity="top"
               android:background="@null"/>
         <ImageButton
               android:id="@+id/shoppingButton"
               android:layout_marginTop="10dp"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_marginEnd="30dp"
               android:src="@drawable/ic_menu_shopping"
               android:background="@null"/>
         <ImageButton
               android:id="@+id/favButton"
               android:layout_marginTop="10dp"
               android:layout_width="wrap_content"
               android:layout_height="wrap_content"
               android:layout_marginEnd="30dp"
               android:src="@drawable/ic_menu_favourite"
```

```
        android:background="@null"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp">

      <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:id="@+id/foodName"
          android:layout_marginStart="10dp"
          android:layout_marginTop="20dp"
          android:text="Food Name"
          android:textSize="40sp"
          android:textColor="@color/secondaryDarkColor"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp">
      <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_margin="10dp"
          android:id="@+id/foodItem_category_label"
          android:textSize="20sp"
          android:layout_marginRight="15dp"
          android:textColor="@color/secondaryDarkColor"
          android:text="Category:"/>
      <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_margin="10dp"
          android:id="@+id/foodItem_category"
          android:textSize="20sp"
          android:textColor="@color/primaryTextColor"
          android:text="Category here"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp">
      <TextView
          android:id="@+id/cupboard_label"
          android:layout_margin="10dp"
          android:text= "In your cupboard: "
          android:textColor="@color/secondaryDarkColor"
```

```
            android:textStyle="bold"
            android:layout_width="wrap_content"
            android:textSize="20sp"
            android:layout_marginRight="10dp"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/old_qty"
            android:layout_margin="10dp"
            android:textColor="@color/primaryTextColor"
            android:text="0"
            android:textSize="20sp"
            android:layout_marginRight="10dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/old_msr"
            android:layout_margin="10dp"
            android:textColor="@color/primaryTextColor"
            android:text="grams"
            android:textSize="20sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:orientation="horizontal">
        <ImageButton
            android:id="@+id/addCupboard"
            android:layout_margin="10dp"
            android:padding="10dp"
            android:src="@drawable/ic_add_circle"
            android:background="@null"
            android:tint="@color/secondaryDarkColor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <ImageButton
            android:id="@+id/removeCupboard"
            android:layout_margin="10dp"
            android:padding="10dp"
            android:src="@drawable/ic_remove_circle"
            android:background="@null"
            android:tint="@color/secondaryDarkColor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
  </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

4.2.4 – Main Content

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main"
    tools:context=".MainActivity">
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:layout_marginTop="100dp">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical"
        android:weightSum="3"
        android:layout_gravity="center_horizontal">
      <ImageButton
          android:id="@+id/viewCupboardButton"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:tint="@color/secondaryDarkColor"
          android:src="@drawable/ic_food"
          android:background="@null"
          android:padding="10dp"
          android:layout_margin="10dp"
          android:layout_weight="2"/>
      <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:textColor="@color/primaryTextColor"
          android:layout_weight="1"
          android:text="My Cupboard"
          android:padding="10dp"
          android:layout_margin="10dp"
          android:textSize="20sp"
          android:labelFor="@+id/viewCupboardButton"
          android:textAlignment="center"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
```

```
    android:weightSum="2"
    android:orientation="horizontal">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:weightSum="3"
    android:orientation="vertical"
    android:layout_gravity="center_horizontal">
  <ImageButton
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/loadSearchPage"
      android:tint="@color/secondaryDarkColor"
      android:layout_weight="2"
      android:layout_margin="10dp"
      android:padding="10dp"
      android:background="@null"
      android:src="@drawable/ic_search"/>
  <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:labelFor="@+id/loadSearchPage"
      android:textColor="@color/primaryTextColor"
      android:textAlignment="center"
      android:text="Search All Food"
      android:layout_margin="10dp"
      android:padding="10dp"
      android:layout_weight="1"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:weightSum="3"
    android:orientation="vertical"
    android:layout_gravity="center_horizontal">
  <ImageButton
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/loadAddPage"
      android:tint="@color/secondaryDarkColor"
      android:layout_weight="2"
      android:layout_margin="10dp"
      android:padding="10dp"
      android:src="@drawable/ic_add_circle"
      android:background="@null"/>
  <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:labelFor="@+id/loadAddPage"
```

```
                android:textColor="@color/primaryTextColor"
                android:textAlignment="center"
                android:text="Add Custom Food"
                android:layout_margin="10dp"
                android:padding="10dp"
                android:layout_weight="1"/>

        </LinearLayout>

    </LinearLayout>

  </LinearLayout>

</LinearLayout>
```

4.2.5 – Recipe Item Content

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:fillViewport="false"
    android:id="@+id/root_layout"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_recipe_item"
    tools:context=".RecipeItemActivity">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
      <de.hdodenhof.circleimageview.CircleImageView
          android:layout_width="match_parent"
          android:layout_height="200dp"
          android:id="@+id/image"
          android:scaleType="centerCrop"
          android:src="@mipmap/ic_launcher"/>
      <LinearLayout
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:orientation="horizontal"
          android:layout_gravity="right">
        <Button
            android:id="@+id/edit_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text = "Edit"
            android:layout_gravity="top"
            android:background="@null"/>
        <ImageButton
            android:id="@+id/shoppingButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="40dp"
            android:layout_marginTop="10dp"
            android:src="@drawable/ic_menu_shopping"
            android:background="@null"/>
        <ImageButton
            android:id="@+id/favButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="40dp"
            android:layout_marginTop="10dp"
```

```
            android:src="@drawable/ic_menu_favourite"
            android:background="@null"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="10dp">
      <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/recipeName"
            android:layout_marginStart="10dp"
            android:layout_marginTop="20dp"
            android:text="Recipe Name"
            android:textSize="40sp"
            android:textColor="@color/secondaryDarkColor"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:layout_marginTop="10dp">
      <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:id="@+id/serving_label"
            android:textSize="20sp"
            android:textColor="@color/secondaryDarkColor"
            android:text="Serves:"/>
      <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:id="@+id/serving"
            android:textSize="20sp"
            android:textColor="@color/primaryTextColor"
            android:text="0"/>
      <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:id="@+id/cuisine_label"
            android:textSize="20sp"
            android:textColor="@color/secondaryDarkColor"
            android:text="|  Cuisine:"/>
      <TextView
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:id="@+id/cuisine"
            android:textSize="20sp"
            android:textColor="@color/primaryTextColor"
            android:text="Irish"/>
    </LinearLayout>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:padding="10dp"
        android:id="@+id/description"
        android:textSize="20sp"
        android:textColor="@color/primaryTextColor"
        android:text="Description Here"/>
    <TextView
        android:id="@+id/ingredient_label"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:padding="10dp"
        android:textColor="@color/secondaryDarkColor"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:text="Ingredients:"/>
    <android.support.v7.widget.RecyclerView
        android:id = "@+id/ingredientItem"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"/>
    <TextView
        android:id="@+id/method_label"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:padding="10dp"
        android:textColor="@color/secondaryDarkColor"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:text="Method:"/>
    <TextView
        android:id="@+id/method"
        android:textColor="@color/primaryTextColor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:padding="20dp"
        android:layout_margin="10dp"
```

```
            android:text="METHOD HERE"/>
    </LinearLayout>
</ScrollView>
```

4.2.6 – Popup Add Quantity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android./apk/res-auto"
    android:id="@+id/linear_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="10dp"
    android:background="@color/primaryColor">

  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:padding="10dp"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp">
    <TextView
        android:id="@+id/cupboard_label"
        android:text= "How much should you add? "
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:textColor="@color/secondaryTextColor"
        android:textSize="20sp"
        android:layout_marginRight="10dp"
        android:layout_height="wrap_content"/>
  </LinearLayout>
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:padding="10dp"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp">
    <TextView
        android:id="@+id/food"
        android:text="Food Name Here"
        android:textColor="@color/secondaryColor"
        android:textStyle="bold"
        android:textSize="20sp"
        android:layout_marginRight="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

  </LinearLayout>
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:orientation="horizontal"
        android:padding="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">
    <TextView
            android:text="Quantity: "
            android:textColor="@color/secondaryColor"
            android:textSize="20sp"
            android:labelFor="@+id/enter_quantity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    <EditText
            android:id="@+id/enter_quantity"
            android:textSize="20sp"
            android:textColor="@color/secondaryTextColor"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="numberDecimal"/>
</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">
    <TextView
            android:text="Measurement: "
            android:textSize="20sp"
            android:textColor="@color/secondaryColor"
            android:labelFor="@+id/enter_measurement"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    <Spinner
            android:id="@+id/enter_measurement"
            android:textColor="@color/secondaryTextColor"
            android:backgroundTint="@color/secondaryTextColor"
            android:textSize="20sp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp"
        android:layout_gravity="center_horizontal"
        android:weightSum="1">
    <ImageButton
```

```
            android:id="@+id/add_qty_btn"
            android:tint="@color/secondaryTextColor"
            android:src="@drawable/ic_check_circle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="20dp"
            android:background="@null"/>
        <ImageButton
            android:id="@+id/cancel"
            android:tint="@color/secondaryTextColor"
            android:src="@drawable/ic_cancel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@null"/>

    </LinearLayout>

</LinearLayout>
```

4.2.7 – Popup Add to Shopping List

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android./apk/res-auto"
    android:id="@+id/linear_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:background="@color/primaryColor">
  <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp"
      android:orientation="horizontal"
      android:layout_gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryColor"
        android:text="Add to Shopping List?"/>
  </LinearLayout>
  <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:layout_gravity="center"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="center"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">
      <ImageButton
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:src="@drawable/ic_add_circle"
          android:background="@null"
          android:layout_gravity="center"
          android:padding="10dp"
          android:tint="@color/secondaryTextColor"
          android:id="@+id/add_all"/>
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:textSize="20sp"
            android:textColor="@color/secondaryTextColor"
            android:textStyle="bold"
            android:text="Add All"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="center"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">
        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:src="@drawable/ic_cancel"
            android:tint="@color/secondaryTextColor"
            android:background="@null"
            android:padding="10dp"
            android:id="@+id/add_none"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:textSize="20sp"
            android:textColor="@color/secondaryTextColor"
            android:textStyle="bold"
            android:text="Close"/>
    </LinearLayout>
  </LinearLayout>
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/ingredient_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
  </LinearLayout>
</LinearLayout>
```

4.2.8 – Popup Save Barcode

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android./apk/res-auto"
    android:id="@+id/linear_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:background="@color/primaryColor">
  <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp"
      android:orientation="horizontal"
      android:layout_gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryTextColor"
        android:text="Barcode: "/>
    <TextView
        android:id="@+id/tv_barcode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryTextColor"
        android:text="xxxxxxxxxxxx "/>
  </LinearLayout>
  <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginRight="10dp"
      android:layout_marginLeft="10dp"
      android:orientation="horizontal"
      android:layout_gravity="center">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:padding="10dp"
```

```
        android:tint="@color/secondaryTextColor"
        android:src="@android:drawable/ic_menu_search"
        android:background="@null"/>
    <AutoCompleteTextView
        android:id="@+id/autocompletetextview"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryTextColor"
        android:hint="Search Food"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:orientation="horizontal"
    android:layout_gravity="center">
    <TextView
        android:id="@+id/tv_food"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryColor"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginBottom="10dp"
    android:orientation="horizontal"
    android:layout_gravity="center">
    <TextView
        android:text="Quantity: "
        android:textColor="@color/secondaryColor"
        android:textSize="20sp"
        android:labelFor="@+id/enter_quantity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/tv_qty"
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
```

```
        android:inputType="numberDecimal"
        android:layout_marginEnd="10dp"
        android:textColor="@color/secondaryTextColor"/>
    <Spinner
        android:id="@+id/tv_measure"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textColor="@color/secondaryTextColor"
        android:backgroundTint="@color/secondaryTextColor"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:orientation="horizontal"
    android:layout_gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textColor="@color/secondaryColor"
        android:text="Brand: "/>
    <EditText
        android:id="@+id/brand"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryTextColor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="10dp"
    android:orientation="horizontal"
    android:layout_gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/secondaryTextColor"
        android:text="Do you want to save?"/>
</LinearLayout>
```

```
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_gravity="center"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">

      <ImageButton
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:src="@drawable/ic_check_circle"
          android:background="@null"
          android:layout_gravity="center"
          android:padding="10dp"
          android:tint="@color/secondaryTextColor"
          android:id="@+id/save"/>
      <ImageButton
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_gravity="center"
          android:src="@drawable/ic_cancel"
          android:tint="@color/secondaryTextColor"
          android:background="@null"
          android:padding="10dp"
          android:id="@+id/no_save"/>
    </LinearLayout>
</LinearLayout>
```

Section 5 – Manifest and Gradle Files

Section 5.1 – Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.kitchenette.kitchenette">

    <uses-permission android:name="android.permission.INTERNET"/>

    <uses-permission-sdk-23 android:name="android.permission.INTERNET"/>

    <uses-permission android:name="android.permission.CAMERA"/>

    <uses-permission-sdk-23 android:name="android.permission.CAMERA"/>

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
      <activity
          android:name=".EditRecipeActivity"
          android:label="Edit Recipe"
          android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
          android:name=".EditFoodActivity"
          android:label="Edit Food"
          android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
          android:name=".AddRecipeActivity"
          android:label="Add New Recipe"
          android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
          android:name=".RecipeItemActivity"
          android:label="View Recipe"
          android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
          android:name=".CookbookActivity"
          android:label="My Cookbook"
          android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
          android:name=".CupboardActivity"
```

```
        android:label="My Cupboard"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".FavouritesActivity"
        android:label="@string/title_activity_favourites"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".ShoppingListActivity"
        android:label="@string/title_activity_shopping_list"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".BarcodeHistoryActivity"
        android:label="@string/title_activity_barcode_history"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".ScanBarcodeActivity"
        android:label="@string/title_activity_scan_barcode"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".AddFoodActivity"
        android:label="@string/title_activity_add_food"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".FoodItemActivity"
        android:label="@string/title_activity_food_item"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".SearchFoodActivity"
        android:label="@string/title_activity_search_food"
        android:theme="@style/AppTheme.NoActionBar">
    </activity>
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:theme="@style/AppTheme.NoActionBar">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>

    <meta-data
        android:name="com.google.android.gms.vision.DEPENDENCIES"
```

```
            android:value="barcode"/>
    </application>

</manifest>
```

Section 5.2 – Project Build Gradle

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.kitchenette.kitchenette">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission-sdk-23 android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.CAMERA"/>

    <uses-permission-sdk-23 android:name="android.permission.CAMERA"/>

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
      <activity
            android:name=".EditRecipeActivity"
            android:label="Edit Recipe"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
            android:name=".EditFoodActivity"
            android:label="Edit Food"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
            android:name=".AddRecipeActivity"
            android:label="Add New Recipe"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
            android:name=".RecipeItemActivity"
            android:label="View Recipe"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
            android:name=".CookbookActivity"
            android:label="My Cookbook"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
            android:name=".CupboardActivity"
            android:label="My Cupboard"
            android:theme="@style/AppTheme.NoActionBar">
      </activity>
      <activity
```

```
            android:name=".FavouritesActivity"
            android:label="@string/title_activity_favourites"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".ShoppingListActivity"
            android:label="@string/title_activity_shopping_list"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".BarcodeHistoryActivity"
            android:label="@string/title_activity_barcode_history"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".ScanBarcodeActivity"
            android:label="@string/title_activity_scan_barcode"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".AddFoodActivity"
            android:label="@string/title_activity_add_food"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".FoodItemActivity"
            android:label="@string/title_activity_food_item"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".SearchFoodActivity"
            android:label="@string/title_activity_search_food"
            android:theme="@style/AppTheme.NoActionBar">
        </activity>
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
          <intent-filter>
            <action android:name="android.intent.action.MAIN"/>

            <category android:name="android.intent.category.LAUNCHER"/>
          </intent-filter>
        </activity>

        <meta-data
            android:name="com.google.android.gms.vision.DEPENDENCIES"
            android:value="barcode"/>
    </application>

</manifest>
```

Section 5.3 – App Build Gradle

```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'


android {
   compileSdkVersion 28
   defaultConfig {
      applicationId "com.kithcenette.kitchenette_v2"
      minSdkVersion 21
      targetSdkVersion 28
      versionCode 1
      versionName "1.0"
      testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
   }
   buildTypes {
      release {
         minifyEnabled false
         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
      }
   }
}

dependencies {
   implementation 'com.google.firebase:firebase-core:16.0.8'
   implementation fileTree(include: ['*.jar'], dir: 'libs')
   implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
   implementation 'com.android.support:appcompat-v7:28.0.0'
   implementation 'com.android.support:support-v4:28.0.0'
   implementation 'com.android.support:design:28.0.0'
   implementation 'com.android.support.constraint:constraint-layout:1.1.3'
   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.android.gms:play-services-vision:17.0.2'
   implementation 'com.readystatesoftware.sqliteasset:sqliteassethelper:+'
   implementation 'de.hdodenhof:circleimageview:3.0.0'
   implementation                                    files('libs/CircleImageView-master/CircleImageView-
master/gradle/wrapper/gradle-wrapper.jar')
}

apply plugin: 'com.google.gms.google-services'
```