Menu Planner Application

MIS 571 Final Project

Group 5

Ruoxi Huang

Scott Judson

Isha Srivastava

Yan Tan

1. Business Problem

1.1 Problem Statement

Creating a shopping list for household or for an event that requires reviewing recipes for each dish to be prepared and computing the quantity of each ingredient required.

Several challenges arise in this process:

- A variety of different units are used for the same ingredients in different recipes.
- Calculating the ingredients required to serve a group when different recipes have different numbers of servings.
- Ingredients are frequently sold and packaged using different units than are used by recipes.

The errors caused by these challenges include:

- Insufficient ingredients to complete a recipe.
- Purchasing more ingredients than are required.

The solution that exists now is to either:

- Manually compute the total quantity of each of the recipes, and estimate the purchase units from the total recipe units, or
- Compile the data into a spreadsheet (this method is not feasible for day to day use, but has proven invaluable when planning large events).

1.2 Proposed Solution

A database backed application would be far more functional, where user can give their details and use/store recipes for future use, thereby providing a better user interface for data entry and a view of the final sorted shopping list while performing the essential functions of scaling quantities and converting them into standard units.

1.3 Key Product Features:

a) A library of recipes
b) A library where ingredient lists are stored
c) Users defines one or more meals/events with a menu, specifying the number of participants (or servings required)
d) Users can add a meal to a shopping trip (or generate a weekly shopping trip from the meals for that week)
e) A shopping list that is generated for one or more meals/events based on:
    - Scaling the recipe based on the servings per batch and the number of servings required (recipes are scaled to nearest integer value- no fractional batches allowed).
    - Converts the ingredients for multiple recipes with overlapping ingredients into common units.
    - Groups the ingredients by grocery department.

1.4 Business Rules

a) Recipes:
    - have one or many ingredients
    - is contributed by a user or the application administrator
    - may be used by any user
b) Ingredients:
    - May be measured in units of weight or volume for recipes
    - Have a density to conversion between volumetric and weight/mass units
    - Have purchase units (e.g., lbs. for flour, sugar, butter);
c) Meals/Events:
    - Have one or more recipes (via Menu composite table)
    - has a name and date

- is created by a user
- Have number guests/attendees

d) Menus:
   - Are part of a single meal/event
   - an event may have one or more menu items
   - A recipe will appear on the menu for an event once
   - Recipes will appear on many different menus

e) Shopping List
   - Have one or more meals/events
   - Is created by a user
   - Are grouped by ingredients
   - Ingredient quantities are expressed in the units that are commonly used to purchase that ingredient.

2. Database Design

2.1 Entity-relationship Diagram

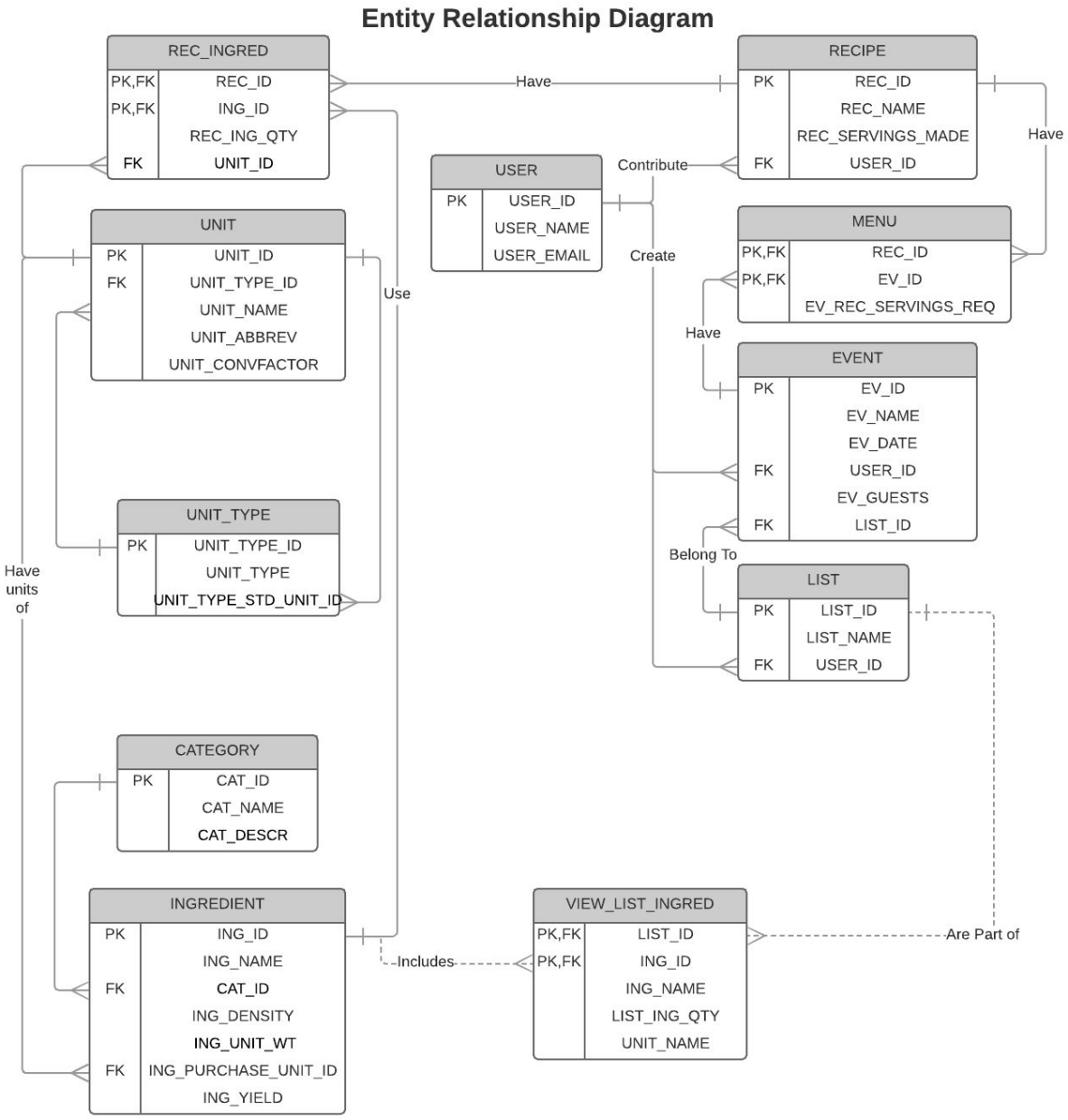

Figure 1 - Entity Relationship Diagram

2.2 Normalized 3NF Relations:

USER (**USER_ID**, USER_NAME, USER_EMAIL)

CATEGORY (**CAT_ID**, CAT_NAME, CAT_DESCR)

UNIT_TYPE (**UNIT_TYPE_ID,** UNIT_TYPE, UNIT_TYPE_STD_UNIT_ID)

UNIT (**UNIT_ID**, UNIT_TYPE_ID, UNIT_NAME, UNIT_ABBREV, UNIT_CONVFACTOR)

LIST (**LIST_ID**, LIST_NAME, USER_ID)

EVENT (**EV_ID**, EV_NAME, EV_DATE, USER_ID, EV_GUESTS, LIST_ID)

RECIPE (**REC_ID**, REC_NAME, REC_SERVINGS_MADE, USER_ID)

MENU (**REC_ID**, **EV_ID**, EV_REC_SERVINGS_REQ)

REC_INGRED (**REC_ID**, **ING_ID**, REC_ING_QTY, UNIT_ID)

INGREDIENT (**ING_ID**, ING_NAME, CAT_ID, ING_DENSITY, ING_UNIT_WT,

ING_PURCHASE_UNIT_ID, ING_YIELD)

VIEW_LIST_INGRED (**LIST_ID**, **ING_ID**, ING_NAME, LIST_ING_QTY, UNIT_NAME)

2.2 Data Dictionary

Table Name: USER

The user table is used to store the information of the users, such as their ID, name and email.

Attributes	**Format**	**Description**
USER_ID (PK)	INTEGER	User identification key
USER_NAME	VARCHAR(35)	User name
USER_EMAIL	VARCHAR(50)	User email address

Table Name: CATEGORY

Each ingredient will be assigned a certain category that corresponds to a department in the grocery store where the ingredient is typically located.

Attributes	**Format**	**Description**
CAT_ID(PK)	INTEGER	Category identification key
CAT_NAME	VARCHAR(25)	Category name
CAT_DESCR	VARCHAR(50)	Category description

Table Name: UNIT_TYPE

Each units of measure are classified into one of three unit types: count, weight, or volume.

Attributes	**Format**	**Description**
UNIT_TYPE_ID(PK)	INTEGER	Unit Type identification key
UNIT_TYPE	VARCHAR(10)	Type name e.g. Count, Volume, or Weight
UNIT_TYPE_STD_UNIT_ID (FK)	INTEGER	The UNIT_ID for the standard units

Table Name: UNIT

This table stores information regarding the units of measures used for the ingredients in recipes and shopping list, including the name and abbreviation, and the conversion factors into standard units for the units type (pieces for count, grams for weight, and milliliters for volume).

Attributes	**Format**	**Description**
UNIT_ID (PK)	INTEGER	Unit identification key
UNIT_TYPE_ID (FK)	INTEGER	Type of unit
UNIT_NAME	VARCHAR(15)	Name of the unit
UNIT_ABBREV	CHAR(3)	Abbreviation for the Unit Name
UNIT_CONVFACTOR	NUMERIC(9.4)	Conversion factor into standard units

Table Name: LIST

The list table contains the name of each list and the user ID of the creator of the list.

Attributes	**Format**	**Description**
LIST_ID (PK)	INTEGER	List identification key
LIST_NAME	VARCHAR(35)	Name of the list
USER_ID (FK)	INTEGER	User identification key (i.e., who created this list)

Table Name: EVENT

This table contains the information for each event, including the name, date, number of guests, as well as the lis that the event is assigned to (if any) and the user ID of the creator of the event.

Attributes	**Format**	**Description**
EV_ID (PK)	INTEGER	Event identification key
EV_NAME	VARCHAR(35)	Event name
EV_Date	DATE	Event date
USER_ID (FK)	INTEGER	User identification key (i.e., the user that created the event)
EV_GUESTS	INTEGER	No. of guests in the event
LIST_ID (FK)	INTEGER	List identification key

Table Name: RECIPE

The recipe table contains high level information about the recipe, including the name, the number of servings made by the recipe and the user ID of the person who contributed the recipe.

Attributes	**Format**	**Description**
REC_ID (PK)	INTEGER	Recipe Identification key
REC_NAME	VARCHAR(50)	Name of the receipt
REC_SERVINGS_MADE	INTEGER	No. of servings made by the recipe
USER_ID (FK)	INTEGER	User identification key (i.e., who contributed this recipe)

Table Name: MENU

This table stores the information about which recipes will be used for each event, as well as the number of servings of the recipe that is required for the event.

Attributes	**Format**	**Description**
REC_ID (PK,FK)	INTEGER	Recipe Identification key
EV_ID (PK,FK)	INTEGER	Event identification key
EV_REC_SERVINGS_REQ	INTEGER	No. of servings of the recipe required

Table Name: RECIPE_INGRED

This table acts as a bridge table between the recipe table and the ingredient table. It stores the recipe ID, ingredient ID, and the quantity and units of the ingredient.

Attributes	**Format**	**Description**
REC_ID (PK,FK)	INTEGER	Recipe identification key
ING_ID (PK,FK)	INTEGER	Ingredient identification key
REC_ING_QTY	REAL	The quantity of ingredient required in the recipe
UNIT_ID (FK)	INTEGER	Unit identification key for the units of measure for REC_ING_QTY

Table Name: INGREDIENT

All the information of the ingredients will be stored in this table, including the name, category, density, unit weight, purchase units and the yield.

Attributes	**Format**	**Description**
ING_ID (PK)	INTEGER	Ingredient identification key
ING_NAME	VARCHAR(35)	Name of the ingredient
CAT_ID (FK)	INTEGER	Category of Ingredient
ING_DENSITY	NUMERIC(6.4)	Density in which ingredient is measured (in grams/milliliter)
ING_UNIT_WT	NUMERIC(6.4)	Weight per unit for items sold by count
ING_PURCHASE_UNIT_ID (FK)	INTEGER	The unit ID for the units of measure in which ingredient will be purchased
ING_YIELD	NUMERIC(4.3)	The yield of the ingredient (i.e., the amount of usable material available after it is prepared for use prior to cooking).

Virtual Table Name: VIEW_LIST_INGRED

The ER Diagram includes a View_List_Ingredient entity that was intended to be generated dynamically for users when a detailed shopping list was requested. In developing the prototype, it was determined that a simple select query couple accomplish the essentially the same result without the added complexity of creating and managing the virtual table. The

select query is similar to the query that the virtual table would use, calculating the aggregate quantity of the ingredients require buy the recipes link to the list via the EVENT and MENU entities , but omits the primary key List_ID and ING_ID attributes. Given the importance of this query, the information regarding the proposed virtual table is included here for reference.

Attributes	**Format**	**Description**
LIST_ID (PK,FK)	INTEGER	List identification key
ING_ID (PK,FK)	INTEGER	Ingredient identification key
ING_NAME	VARCHAR(35)	Ingredient Name
LIST_ING_QTY	NUMERIC(6.4)	The final quantity of ingredient needed (Calculated)
UNIT_NAME	VARCHAR(15)	The name of the unit in which ingredient will be purchased

List of indices

Indices will be created to improve the performance of searches using the name of users, user email addresses, recipe names, event names, and list names. The size of the database in the the prototype does not require the use of indexes, but they will be essential as the database grows in size.

Index Key Attribute	Table	Why these attributes?
USER_EMAIL	USER	User email addresses are used to log into the program.
USER_NAME	USER	User name is frequently used to track the information of a user.
RECIPE_NAME	RECIPE	Recipes search and lookup functions use Recipe Name as the index key.
ING_NAME	INGREDIENT	Ingredient search and lookup functions use Ingredient Name as the index key.
EVENT_NAME	EVENT	Event name is frequently used to lookup a specific event.
LIST_NAME	LIST	List name is frequently used to lookup a specific list.
USER_ID	RECIPE	Users will be able to lookup the recipes they contribute. Indexing the Recipe table using User_ID will facilitate this search.
USER_ID	EVENT	Indexing Event and List tables using User_ID as the key will facilitate restricting users to accessing only their own events and lists.
USER_ID	LIST	

2.3 Relationships between Entities

USER:EVENT(1:M)

A user can create many events; an event is only created by a certain user. Thus, the two entities have a one-to-many relationship.

USER:RECIPE(1:M)

A user can contribute many recipes; a recipe is only contributed by a certain user. Thus, the two entities have a one-to-many relationship.

USER:LIST(1:M)

A user can create many shopping lists; a shopping list is only created by a certain user. Thus, the two entities have a one-to-many relationship.

EVENT:RECIPE (M:N) -→EVENT:MENU(1:M), RECIPE:MENU(1:M)

An event will have several recipes, while recipes may be used for many different events. This many to many relations is converted into two separate one to many relationships using the Menu entity. Each Menu instance represents a unique combination of one recipe and one event. An event will have many entries in the menu table (one to many) and a recipe may appear in many menu table instances (one to many).

RECIPE:INGREDIENT(M:N)→RECIPE:REC_INGRED (1:M),
INGREDIENT:REC_INGRED(1:M)

A recipe may have many ingredients; an ingredient will appear on many different recipes. Thus, the two entities have a many-to-many relationship. By adding a REC_INGRED entity, the many-to-many relationship is converted to two one-to-many relationships.

CATEGORY:INGREDIENT(1:M)

A category can contain many different ingredients; an ingredient can only be classified into a certain category. Hence, the two entities have a one-to-many relationship.

UNIT:INGREDIENT(1:M), UNIT:REC_INGRED(1:M)

An ingredient (INGREDIENT) can only have a single unit of measure used for purchasing that ingredient, and the ingredient in a recipe (REC_INGRED) can only have one unit of measure. A unit of measure can be used for many different ingredients and in many different recipes[[[[[. Hence, the two entities have a one-to-many relationship.

UNIT_TYPE:UNIT(1:M)

A unit type can contain many different units; a unit can only be classified into a unit type. Hence, the two entities have a one-to-many relationship.

LIST:VIEW_LIST_INGRED (1:M), INGREDIENT:VIEW_LIST_INGRED (1:M)

The VIEW_LIST_INGREDIENT entity is an aggregated view that selects specific columns from a table created by a series of joins across the Event, Menu, Recipe, Rec_Ingred, Ingredient, Unit and Unit Type entities. Each instance in VIEW_LIST_INGRED is linked to a single List, and a single ingredient, while lists and ingredients will appear in many instances of the VIEW_LIST_INGRED entity (i.e., they a one to many relationships with VIEW_LIST_INGRED).

2.4 Create Table and Insert SQL Output

CREATE Tables Output

```
PRAGMA foreign_keys = ON;
DROP TABLE IF EXISTS User;
DROP TABLE IF EXISTS Event;
DROP TABLE IF EXISTS List;
DROP TABLE IF EXISTS Recipe;
DROP TABLE IF EXISTS Menu;
DROP TABLE IF EXISTS Rec_Ingred;
DROP TABLE IF EXISTS Ingredient;
DROP TABLE IF EXISTS Unit;
DROP TABLE IF EXISTS Unit_Type;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS View_List_Ingred;
CREATE TABLE User
(User_ID INTEGER,
User_Name VARCHAR(35) NOT NULL,
User_Email VARCHAR(50) NOT NULL,
User_Password VARCHAR(50) NOT NULL,
CONSTRAINT User_User_ID_PK PRIMARY KEY (User_ID));
CREATE TABLE Category
(Cat_ID INTEGER,
Cat_Main VARCHAR(15) NOT NULL,
Cat_Descr VARCHAR(50) NOT NULL,
CONSTRAINT Category_Cat_ID_PK PRIMARY KEY (Cat_ID));
CREATE TABLE Unit_Type
(Unit_Type_ID INTEGER,
Unit_Type VARCHAR(10) NOT NULL,
CONSTRAINT Unit_Type_Ut_ID_PK PRIMARY KEY (Unit_Type_ID));
CREATE TABLE Unit
(Unit_ID INTEGER,
Unit_Type_ID INTEGER NOT NULL,
Unit_Name VARCHAR(15) NOT NULL,
Unit_Abbrev CHAR(3) NOT NULL,
Unit_Convfactor REAL NOT NULL,
CONSTRAINT Unit_Unit_ID_PK PRIMARY KEY (Unit_ID),
CONSTRAINT Unit_Unittype_ID_FK FOREIGN KEY (Unit_Type_ID) REFERENCES
Unit_Type(Unit_Type_ID));
CREATE TABLE List
```

```
(List_ID INTEGER,
List_Name VARCHAR(35) NOT NULL,
User_ID INTEGER NOT NULL,
CONSTRAINT List_List_ID_PK PRIMARY KEY (List_ID),
CONSTRAINT List_User_ID_FK FOREIGN KEY (User_ID) REFERENCES User(User_ID));
CREATE TABLE Event
(Ev_ID INTEGER,
Ev_Name VARCHAR(35) NOT NULL,
Ev_Date DATE,
User_ID INTEGER NOT NULL,
Ev_Guests INTEGER NOT NULL DEFAULT(0),
List_ID INTEGER,
CONSTRAINT Event_Ev_ID_PK PRIMARY KEY (Ev_ID),
CONSTRAINT Event_User_ID_FK FOREIGN KEY (User_ID) REFERENCES User(User_ID),
CONSTRAINT Event_List_ID_FK FOREIGN KEY (List_ID) REFERENCES List(List_ID)
ON DELETE SET NULL);
CREATE TABLE Recipe
(Rec_ID INTEGER,
Rec_Name VARCHAR(50) NOT NULL,
Rec_Servings_Made INTEGER NOT NULL DEFAULT(0),
User_ID INTEGER NOT NULL,
CONSTRAINT Recipe_Rec_ID_PK PRIMARY KEY (Rec_ID),
CONSTRAINT Recipe_User_ID_FK FOREIGN KEY (User_ID) REFERENCES
User(User_ID));
CREATE TABLE Menu
(Rec_ID INTEGER,
Ev_ID INTEGER,
Ev_Rec_Servings_Req INTEGER NOT NULL DEFAULT(0),
CONSTRAINT Menu_Rec_ID_FK FOREIGN KEY (Rec_ID) REFERENCES Recipe(Rec_ID) ON
DELETE CASCADE,
CONSTRAINT Menu_Ev_ID_FK FOREIGN KEY (Ev_ID) REFERENCES Event(Ev_ID) ON
DELETE CASCADE,
CONSTRAINT Menu_Rec_Ev_ID_PK PRIMARY KEY (Rec_ID, Ev_ID));
CREATE TABLE Rec_Ingred
(Rec_ID INTEGER,
Ing_ID INTEGER,
Rec_Ing_Qty REAL NOT NULL,
Unit_ID INTEGER NOT NULL,
CONSTRAINT Rec_Ingred_Rec_ID_FK FOREIGN KEY (Rec_ID) REFERENCES
Recipe(Rec_ID) ON DELETE CASCADE,
CONSTRAINT Rec_Ingred_Ing_ID_FK FOREIGN KEY (Ing_ID) REFERENCES
Ingredient(Ing_ID) ON DELETE CASCADE,
```

```
CONSTRAINT Rec_Ingred_Unit_ID_FK FOREIGN KEY (Unit_ID) REFERENCES
Unit(Unit_ID),
CONSTRAINT Rec_Ingred_Rec_Ing_ID_PK PRIMARY KEY (Rec_ID, Ing_ID));
CREATE TABLE Ingredient
(Ing_ID INTEGER,
Ing_Name VARCHAR(35) NOT NULL,
Cat_ID INTEGER NOT NULL,
Ing_Density REAL NOT NULL DEFAULT(0),
Ing_Unit_Wt REAL NOT NULL DEFAULT(0),
Ing_Purchase_Unit_ID INTEGER NOT NULL,
Ing_Yield REAL NOT NULL DEFAULT(1),
CONSTRAINT Ingredient_Ing_ID_PK PRIMARY KEY (Ing_ID),
CONSTRAINT Ingredient_Ingcat_ID_FK FOREIGN KEY (Cat_ID) REFERENCES
Category(Cat_ID),
CONSTRAINT Ingredient_Ingpur_Unit_ID_FK FOREIGN KEY (Ing_Purchase_Unit_ID)
REFERENCES Unit(Unit_ID));
```

INSERT Statement Output (Representative Sample)

```
INSERT INTO Category (Cat_ID, Cat_Main,Cat_Descr)
VALUES (1,"Coffee & Tea","");

INSERT INTO Unit_Type (Unit_Type_ID, Unit_Type)
VALUES (3,    "Weight");

INSERT INTO Unit (Unit_ID, Unit_Name, Unit_Abbrev, Unit_Convfactor,
Unit_Type_ID)
VALUES (1,"pieces","pcs",1,1);

INSERT INTO Ingredient (Ing_Name, Cat_ID,
Ing_Density,Ing_Unit_Wt,Ing_Purchase_Unit_ID, Ing_Yield)
VALUES ("Allspice, Ground",9,0.4058,0.0000,13,1.0000);

INSERT INTO User (User_ID, User_Name, User_Email,User_Password)
VALUES (1, "Admin","Admin@MenuPlanner.us.co","password1");

INSERT INTO List (List_ID, List_Name, User_ID)
VALUES (1,"Admin List",1);

INSERT INTO Event (Ev_ID, Ev_Name,Ev_Date,User_ID,Ev_Guests,List_ID)
VALUES (1, "Company Launch Party", "2018-04-24",1,30,1);
```

```
INSERT INTO Recipe (Rec_ID, Rec_Name, Rec_Servings_Made, User_ID)
VALUES (1, "Scrambled Eggs", 2, 1);
INSERT INTO Rec_Ingred (Rec_ID, Ing_ID, Rec_Ing_Qty, Unit_ID)
VALUES (1,71,4,1);
INSERT INTO Rec_Ingred (Rec_ID, Ing_ID, Rec_Ing_Qty, Unit_ID)
VALUES (1,102,2,7);
INSERT INTO Rec_Ingred (Rec_ID, Ing_ID, Rec_Ing_Qty, Unit_ID)
VALUES (1,22,1,7);

INSERT INTO Menu (Rec_ID, Ev_ID, Ev_Rec_Servings_Req)
VALUES (4,1,30);
```

3. User Interface and Query Documentation

3.1 Query Documentation

There are two groups of queries:

- Single table queries that are used to populate the interactive portions of the user interface and handle routine functions, including adding, updating, and deleting specific instances of an entity, searching for an instance of an entity based on the primary key or descriptive name, etc.
- Multi table queries that perform two primary tasks
    - Generating the user readable lists of Ingredients and Recipes for the Recipe and Menu pages respectively
    - Generating a shopping list in a common set of units, aggregated by ingredient, for a specific list of events.

The table below describes the purpose of the various single table queries and which tables these queries apply to, and a description of the multi-table queries follows the table. The SQL statements for each query, as well as the SQLite output for select queries that produce results, follow these explanations.

Single Table Queries

	TABLES							
QUERY TYPE	USER	EVENT	MENU	RECIPE	REC_INGRED	INGREDIENT	LIST	UNIT
SELECT NAMES (non PK Attribute) FOR ALL INSTANCES These queries are used to show available values for lookup and autocomplete user interface elements, including: ● User email addresses (Landing page) ● Recipe, ingredient, event, list, and unit names	●	●		●		●	●	●
SELECT INSTANCE BY NAME (non PK attribute) This is used to lookup the details of an instance based		●		●			●	●

on user input, and to populate interactive UI elements, including: ● Event guests and date on the Event Page ● Recipe servings, Ingredient name, and Unit name on the Recipe page								
SELECT SUBSET OF ATTRIBUTES FOR SINGLE INSTANCE BY FOREIGN KEY This is used to create a list of events with the number of guests and event dates to the show to the user on the List page		●						
SELECT ID (PK Attribute) BASED ON NAME (non PK Attrribute) This is used to lookup a ingredient ID based on the ingredient name						●		
UPDATE INSTANCE BY NAME OR ID Changes one or more attributes in a row. Use to change: ● Number of guests or event date in EVENTS ● Number of servings required for a recipe in MENU ● Number of servings made in RECIPE ● Quantity of ingredients for a recipe in REC_INGRED		●	●	●	●			
DELETE INSTANCE Removes an existing instance from a table. ● Remove an Event, Recipe or List from the corresponding page ● Remove an ingredients from a recipe (via Rec_Ingred) ● Remove a recipes from an events (via Menu)		●	●	●	●		●	
ADD/REMOVE EVENT TO/FROM LIST Events are assigned to lists by setting the List_ID FK reference in the Event instance to the corresponding List_ID. Events are removed from lists by setting the FK to NULL.		●						

The SQL statements to insert new rows are show in Section 2.4 of the report, and are not listed in this table, or in the SQL statement and output provided below.

The update and delete queries (including the query used to add and remove events from a list) only echo the query and do not generate any other output when run. The SQL output will be omitted for these queries.

Multi-Table Queries

There are three multi-table queries which are used to generate lists of data related to events, recipes, and lists:

- SELECT_EVENT_MENU - This query generates a list of recipes that are on the menu for a specific event, and is used on the Event page. The query joins the Menu and Recipe tables, and selects only the recipe name (from Recipe) and servings required (from Menu).

- SELECT_RECIPE_INGREDIENT_DETAIL - This query generates a list of recipe ingredients for a specific recipe and is used on the Recipe page. This query joins the Rec_ingred, Ingredient, and Unit to generate a list showing the ingredient name (from Ingredient), the quantity required (from Rec_ingred) and the unit of measure (from Unit) for the specified recipe.

- CREATE_SHOPPING_LIST - This query is critical to the application, generating the detailed shopping list by:
    - Joining the Ingredient, Rec_Ingred, Unit, Recipe, Menu, Event and List tables for those instances in the joined table corresponding the selected List_ID
    - Translating the recipe ingredient quantities and units into the ingredient purchase quantities (as QtyReqd) and units (as PurchaseUnits)
        - Calculating the number of batches of a recipe required based on the servings required from Menu and servings made from Recipe
        - Converting recipe units into the standard units
        - Performing density or unit weight conversions based on the recipe and purchase units as needed
        - Converting the quantity in standard units into purchase units
    - Aggregating the ingredient quantities by ingredient
    - Return the list of ingredient names, quantities required and purchase units, sorted by ingredient category.

SQL QUERY STATEMENTS

SELECT NAMES FOR ALL INSTANCES

```
--SELECT_USER_EMAIL
SELECT User_Email FROM User;

--SELECT_EVENT
SELECT Ev_Name FROM Event;

--SELECT_RECIPE
SELECT Rec_Name FROM Recipe;

--SELECT_INGREDIENT
SELECT Ing_Name FROM Ingredient;

--SELECT_LIST
SELECT List_Name FROM List;

--SELECT_UNIT
SELECT Unit_Name FROM Unit;
```

SELECT INSTANCE BY NAME

```
--SELECT_EVENT_BY_NAME
SELECT * FROM Event WHERE Ev_Name = "Dinner at Roxies";

--SELECT_RECIPE_BY_NAME
SELECT * FROM Recipe WHERE Rec_Name = "Apple Pie";


--SELECT_LIST_BY_NAME
SELECT * FROM List
WHERE List_Name = "Yan List";

--SELECT_UNIT_BY_NAME
SELECT * FROM Unit WHERE Unit_Name = "pound";
```

SELECT SUBSET OF ATTRIBUTES BY FOREIGN KEY

```
--SELECT_EVENT_LIST
SELECT Ev_Name,Ev_Guests,Ev_Date FROM Event WHERE Event.List_ID = 1;
```

SELECT ID BY NAME

```
--SELECT_INGREDIENT_BY_NAME
SELECT Ing_Id FROM Ingredient WHERE Ing_Name = "Wasabi Powder";
```

UPDATE INSTANCE BY NAME (Non PK Atrribute) OR ID (PK Attribute)

```
--UPDATE_EVENT
UPDATE Event SET Ev_Guests = "4", Ev_Date ="2018-08-21"
WHERE Ev_Name = "Dinner at Roxies";

--UPDATE_RECIPE_SERVINGS
UPDATE Recipe SET Rec_Servings_Made = 72 WHERE Rec_ID = 2;


--UPDATE_REC_ING_QTY_UNIT
UPDATE Rec_Ingred SET Rec_Ing_Qty = 2, Unit_ID = 2
WHERE Rec_ID = 2
AND Ing_ID = 32;
```

DELETE INSTANCE

```
--DELETE_EVENT
DELETE FROM Event WHERE Ev_ID = 99;

--DELETE_RECIPE
DELETE FROM Recipe WHERE Rec_ID= 1;

--DELETE_RECIPE_FROM_MENU
DELETE FROM Menu WHERE Ev_ID =99 AND Rec_ID = 1;

--DEL_ING_FROM_RECIPE
DELETE FROM Rec_Ingred WHERE Rec_ID= 2 AND Ing_ID = 32;

--DELETE_LIST
DELETE FROM List WHERE List_ID=99;
```

ADD/REMOVE EVENT TO LIST

```
--ADD_EVENT_TO_LIST
UPDATE Event SET List_ID = 3 WHERE EV_ID = 4;

--DELETE_EVENT_ FROM_LIST
UPDATE Event SET List_ID = NULL WHERE EV_ID = 2;
```

MULTI-TABLE QUERIES

```
--SELECT_EVENT_MENU
SELECT Rec_Name,Ev_Rec_Servings_Req
FROM Menu,Recipe
WHERE Menu.Rec_ID = Recipe.Rec_ID
AND  Ev_ID = 2 ;

--SELECT_RECIPE_INGREDIENT_DETAIL
SELECT Ing_Name,Rec_Ing_Qty,Unit_Name
FROM Rec_Ingred,Ingredient,Unit
WHERE Rec_Ingred.Ing_Id = Ingredient.Ing_Id
AND Rec_Ingred.Unit_Id = Unit.Unit_Id
AND Rec_ID = 2;

--CREATE_SHOPPING_LIST
SELECT  Ing.Ing_Name,
  SUM(ROUND(
  CASE
  -- Case 1-1, 2-2, 3-3, Same UnitType for Rec_Ingred and Purchase Units
  WHEN UP.Unit_Type_ID = UR.Unit_Type_ID
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*UR.Unit_Convfactor/UP.Unit_Convfactor)

  -- Case 1-2 Purchase Unit Type = Count && Rec_Ingred Unit Type = VOL
  WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*UR.Unit_Convfactor*Ing.Ing_Density)/Ing.Ing_Unit_Wt)

  -- Case 1-3 Purchase Unit Type = Count && Rec_Ingred Unit Type = WEIGHT
```

```
  WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*UR.Unit_Convfactor)/Ing.Ing_Unit_Wt

  -- Case 2-1 - Purchase Unit Type = Volume, Rec_ingred Unit type = Count
  WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 1
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*Ing.Ing_Unit_Wt/Ing.Ing_Density)/UP.Unit_Convfactor)

  -- Case 2-3 - Purchase Unit Type = Volume, Rec-Ingred Unit Type = Weight
  WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 2
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*UR.Unit_Convfactor/Ing.Ing_Density)/UP.Unit_Convfactor)

  -- Case 3-1 - Purchase Unit Type = Wt, Rec_ingred Unit type = Count
  WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 1
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*Ing.Ing_Unit_Wt/UP.Unit_Convfactor)

  -- Case 3-2 - Purchase Unit Type = Wt, Rec_ingred Unit type = Volume
  WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 2
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*(RI.REC_ING_QTY*UR.Unit_Convfactor*Ing.Ing_Density)/UP.Unit_Convf
actor)
     ELSE "Bad Code"

  END,2)) AS QtyReqd, UP.Unit_Abbrev AS PurchaseUnits

  FROM Ingredient AS Ing
  JOIN Unit     AS UP ON UP.Unit_ID = Ing.Ing_Purchase_Unit_ID
  JOIN Rec_Ingred AS RI ON Ing.Ing_ID = RI.Ing_ID
  JOIN Unit     AS UR ON RI.Unit_ID = UR.Unit_ID
  JOIN Recipe   AS R  ON RI.Rec_ID =R.Rec_ID
  JOIN Menu     AS M  ON R.Rec_ID = M.Rec_ID
  JOIN Event    AS E  ON M.Ev_ID = E.Ev_ID
  JOIN List     AS L  ON E.List_ID = L.List_ID
  WHERE L.List_ID = 5
  GROUP BY Ing.Ing_ID
```

```
ORDER BY Ing.Cat_ID ASC
;
```

SQLITE OUTPUT

SELECT NAMES FOR ALL INSTANCES

```
SELECT User_Email FROM User;
User_Email
Admin@MenuPlanner.us.co
Isha@MenuPlanner.us.co
Roxie@MenuPlanner.us.co
Scott@MenuPlanner.us.co
Yan@MenuPlanner.us.co


SELECT Ev_Name FROM Event;
Ev_Name
Brunch at Yans
Company Launch Party
Dessert at Scotts
Dinner at Roxies
Drinks at Ishas
Test Event

SELECT Rec_Name FROM Recipe;
Rec_Name
Apple Pie
Chocolate Cake
Noodle Kugel
Pancakes
Quiche

SELECT Ing_Name FROM Ingredient;
Ing_Name
Allspice, Ground
Almond extract
Anise Seed, Whole
Apple, Whole Fruit
Apples, Golden Delicious
Apples, Granny Smith
Avocado, Whole
…
Wasabi Powder
Watercress
```

```
Whipping Cream
Yeast - Active
Yogurt, Greek

SELECT List_Name FROM List;
List_Name
Admin List
Isha List
Roxie List
Scott List
Yan List

SELECT Unit_Name FROM Unit;
Unit_Name
pieces
milliliters
liters
…
ounce
pound
```

SELECT INSTANCE BY NAME

```
SELECT * FROM Event WHERE Ev_Name = "Dinner at Roxies";
Ev_ID  |  Ev_Name  |  Ev_Date  |  User_ID  |  Ev_Guests  |  List_ID
3  |  Dinner at Roxies  |  2018-08-21  |  3  |  4  |  3


SELECT * FROM Recipe WHERE Rec_Name = "Apple Pie";
Rec_ID  |  Rec_Name  |  Rec_Servings_Made  |  User_ID
5  |  Apple Pie  |  6  |  5


SELECT * FROM List
 WHERE List_Name = "Yan List";
List_ID  |  List_Name  |  User_ID
2  |  Yan List  |  2

SELECT * FROM Unit WHERE Unit_Name = "pound";
Unit_ID  |  Unit_Type_ID  |  Unit_Name  |  Unit_Abbrev  |  Unit_Convfactor
14  |  3  |  pound  |  lbm  |  453.592
```

SELECT SUBSET OF ATTRIBUTES BY FOREIGN KEY

```
SELECT Ev_Name,Ev_Guests,Ev_Date FROM Event WHERE Event.List_ID = 1;
Ev_Name  |  Ev_Guests  |  Ev_Date
Company Launch Party  |  30  |  2018-04-24
Test Event  |  30  |  2018-04-24
```

SELECT ID BY NAME

```
SELECT Ing_Id FROM Ingredient WHERE Ing_Name = "Wasabi Powder";
Ing_ID
182
```

MULTI-TABLE QUERIES

```
SELECT Rec_Name,Ev_Rec_Servings_Req
FROM Menu,Recipe
WHERE Menu.Rec_ID = Recipe.Rec_ID
AND  Ev_ID = 2 ;
Rec_Name  |  Ev_Rec_Servings_Req
Quiche  |  4
Pancakes  |  2

SELECT Ing_Name,Rec_Ing_Qty,Unit_Name
FROM Rec_Ingred,Ingredient,Unit
WHERE Rec_Ingred.Ing_Id = Ingredient.Ing_Id
AND Rec_Ingred.Unit_Id = Unit.Unit_Id
AND Rec_ID = 2;
Ing_Name  |  Rec_Ing_Qty  |  Unit_Name
Butter  |  8.0  |  tablespoon
Eggs, Large, Whole (in shell)  |  12.0  |  pieces
Milk  |  1.0  |  cup
Mozzarella Cheese  |  2.0  |  cup
```

```
SELECT  Ing.Ing_Name,
  SUM(ROUND(
  CASE
  -- Case 1-1, 2-2, 3-3, Same UnitType for Rec_Ingred and Purchase Units
  WHEN UP.Unit_Type_ID = UR.Unit_Type_ID
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*UR.Unit_Convfactor/UP.Unit_Convfactor)

  -- Case 1-2 Purchase Unit Type = Count && Rec_Ingred Unit Type = VOL
  WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*UR.Unit_Convfactor*Ing.Ing_Density)/Ing.Ing_Unit_Wt)

  -- Case 1-3 Purchase Unit Type = Count && Rec_Ingred Unit Type = WEIGHT
  WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*UR.Unit_Convfactor)/Ing.Ing_Unit_Wt

  -- Case 2-1 - Purchase Unit Type = Volume, Rec_ingred Unit type = Count
  WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 1
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*Ing.Ing_Unit_Wt/Ing.Ing_Density)/UP.Unit_Convfactor)

  -- Case 2-3 - Purchase Unit Type = Volume, Rec-Ingred Unit Type = Weight
  WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 2
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*
  (RI.REC_ING_QTY*UR.Unit_Convfactor/Ing.Ing_Density)/UP.Unit_Convfactor)

  -- Case 3-1 - Purchase Unit Type = Wt, Rec_ingred Unit type = Count
  WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 1
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*RI.REC_ING_QTY*Ing.Ing_Unit_Wt/UP.Unit_Convfactor)

  -- Case 3-2 - Purchase Unit Type = Wt, Rec_ingred Unit type = Volume
  WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 2
```

```
  THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
REAL)/CAST(R.Rec_Servings_Made AS
REAL),0)*(RI.REC_ING_QTY*UR.Unit_Convfactor*Ing.Ing_Density)/UP.Unit_Convfac
tor)
      ELSE "Bad Code"

  END,2)) AS QtyReqd, UP.Unit_Abbrev AS PurchaseUnits

  FROM Ingredient AS Ing
  JOIN Unit      AS UP ON UP.Unit_ID = Ing.Ing_Purchase_Unit_ID
  JOIN Rec_Ingred AS RI ON Ing.Ing_ID = RI.Ing_ID
  JOIN Unit      AS UR ON RI.Unit_ID = UR.Unit_ID
  JOIN Recipe    AS R  ON RI.Rec_ID =R.Rec_ID
  JOIN Menu      AS M  ON R.Rec_ID = M.Rec_ID
  JOIN Event     AS E  ON M.Ev_ID = E.Ev_ID
  JOIN List      AS L  ON E.List_ID = L.List_ID
  WHERE L.List_ID = 5
  GROUP BY Ing.Ing_ID
  ORDER BY Ing.Cat_ID ASC
  ;
Ing_Name  |  QtyReqd  |  PurchaseUnits
Butter  |  1.51  |  lbm
Flour, All Purpose  |  0.55  |  lbm
Sugar, Brown, Dark  |  1.32  |  lbm
Sugar, Granulated  |  2.2  |  lbm
Noodles, wide  |  2.0  |  lbm
Apple, Whole Fruit  |  2.73  |  lbm
```

3.2. User interface

Overview

The figure below provides an overview of the navigation through the various user interface screens.

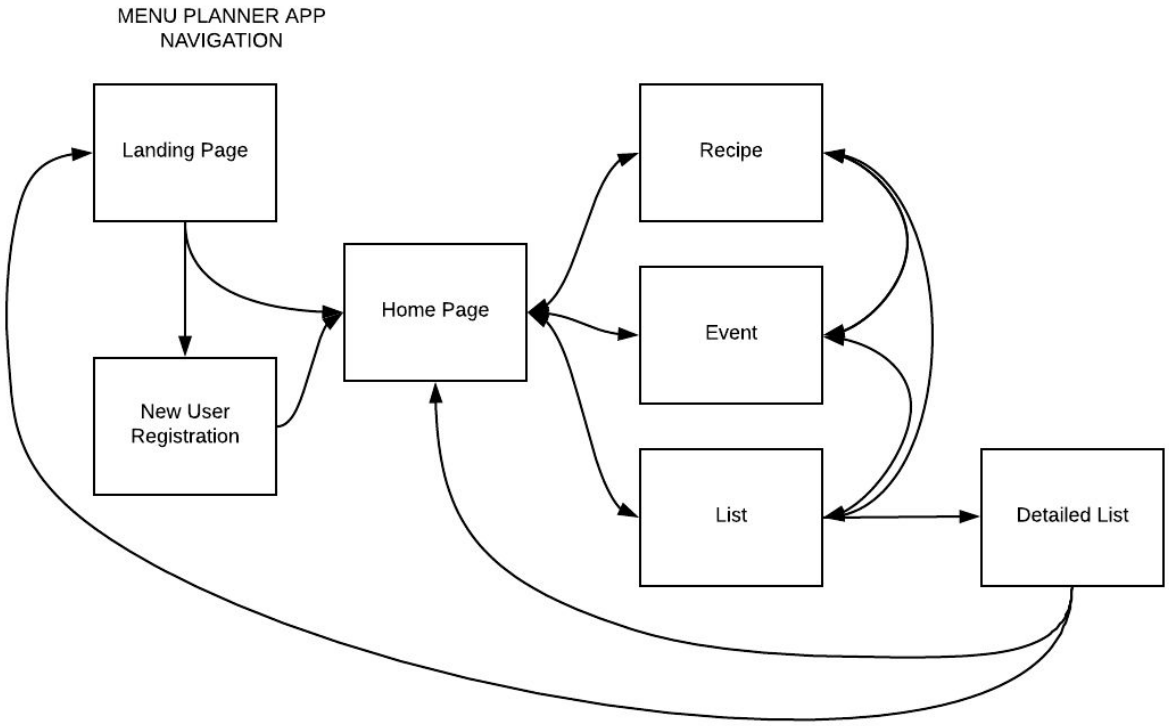

Figure 2 - Application User Interface Navigation

User Interface Screens

The application starts at the Landing Page, where the user is prompted for an email address.

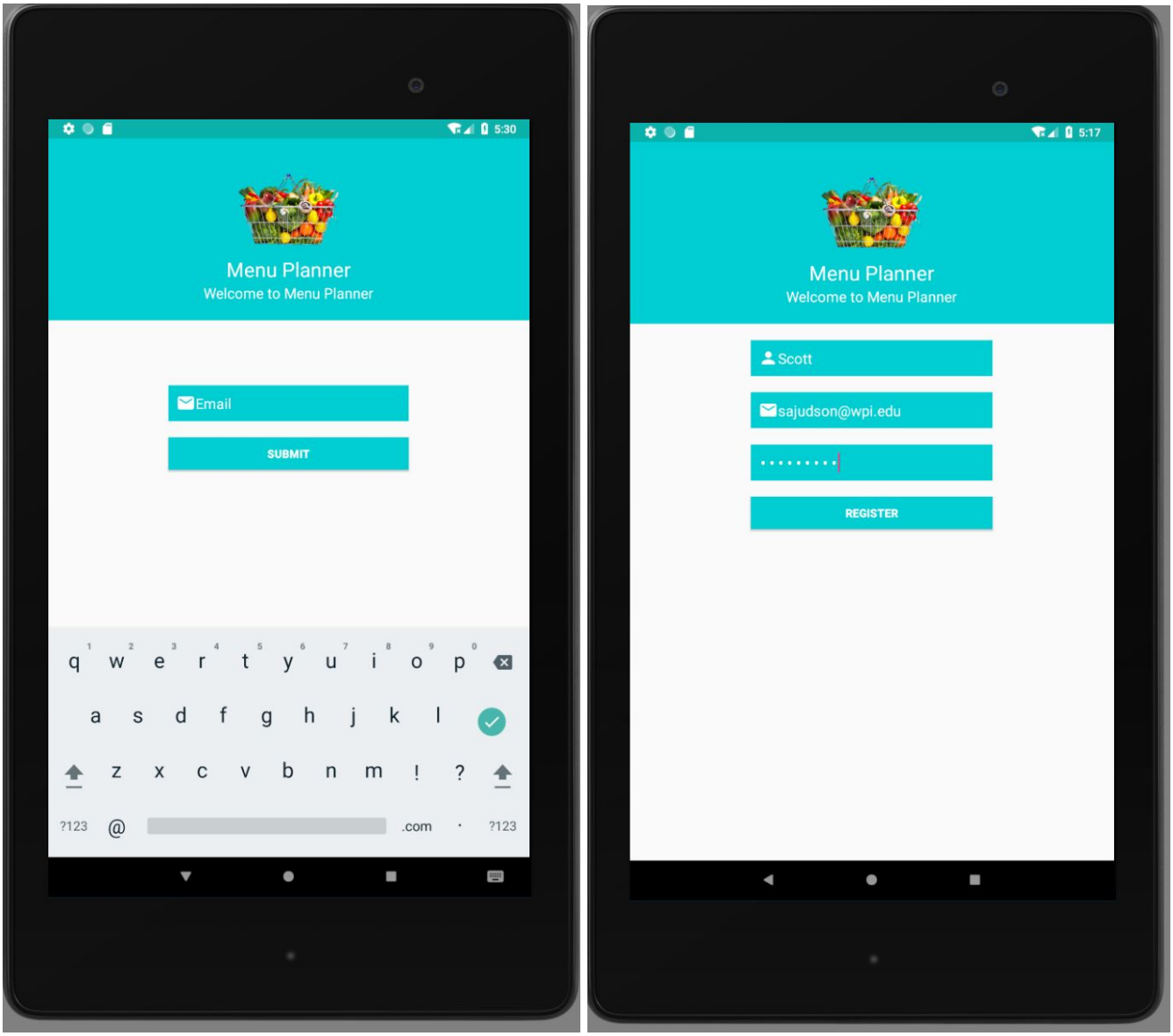

Figure 3 - Landing and New User Registration Pages

If the users email address does not exist in the user database, the user is sent to the New User Registration page. If the User does exist, they are sent to the application Home Page.

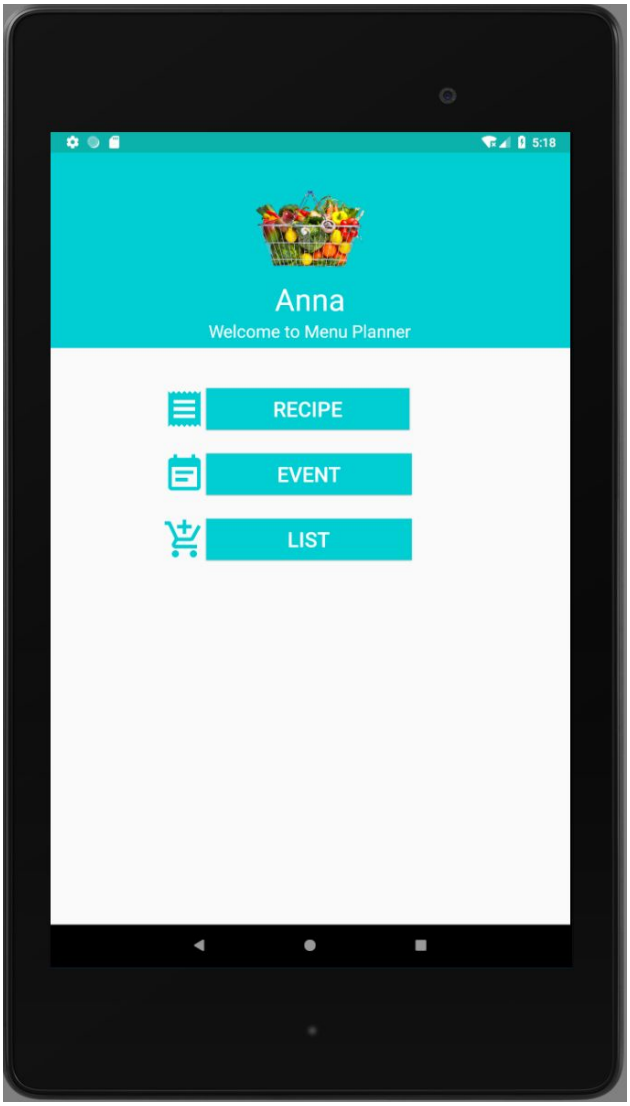

Figure 4 Home Page

At the application Home Page, the user can navigate to the Recipe, Event and List user interface screens

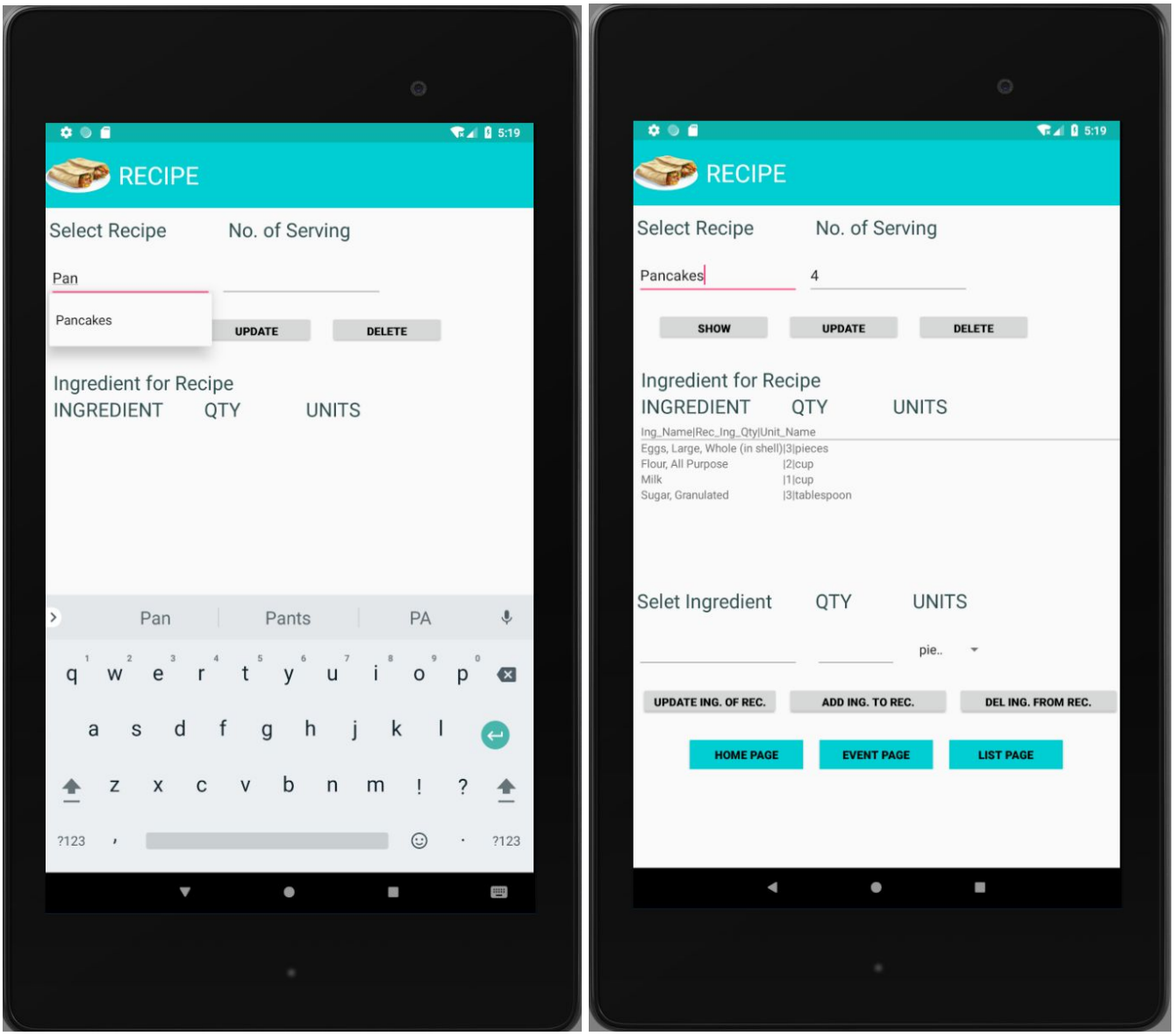

Figure 5a-b Recipe Page a) Select Recipe, b) View Recipe Ingredients

On the Recipe UI screen, the user can select a recipe (using autocomplete), update the servings made field, or delete the recipe. They can also view the recipe ingredients, and add or delete ingredients, and update ingredient quantities. The page also includes buttons to navigate back to the home page, or directly to the event and list pages.

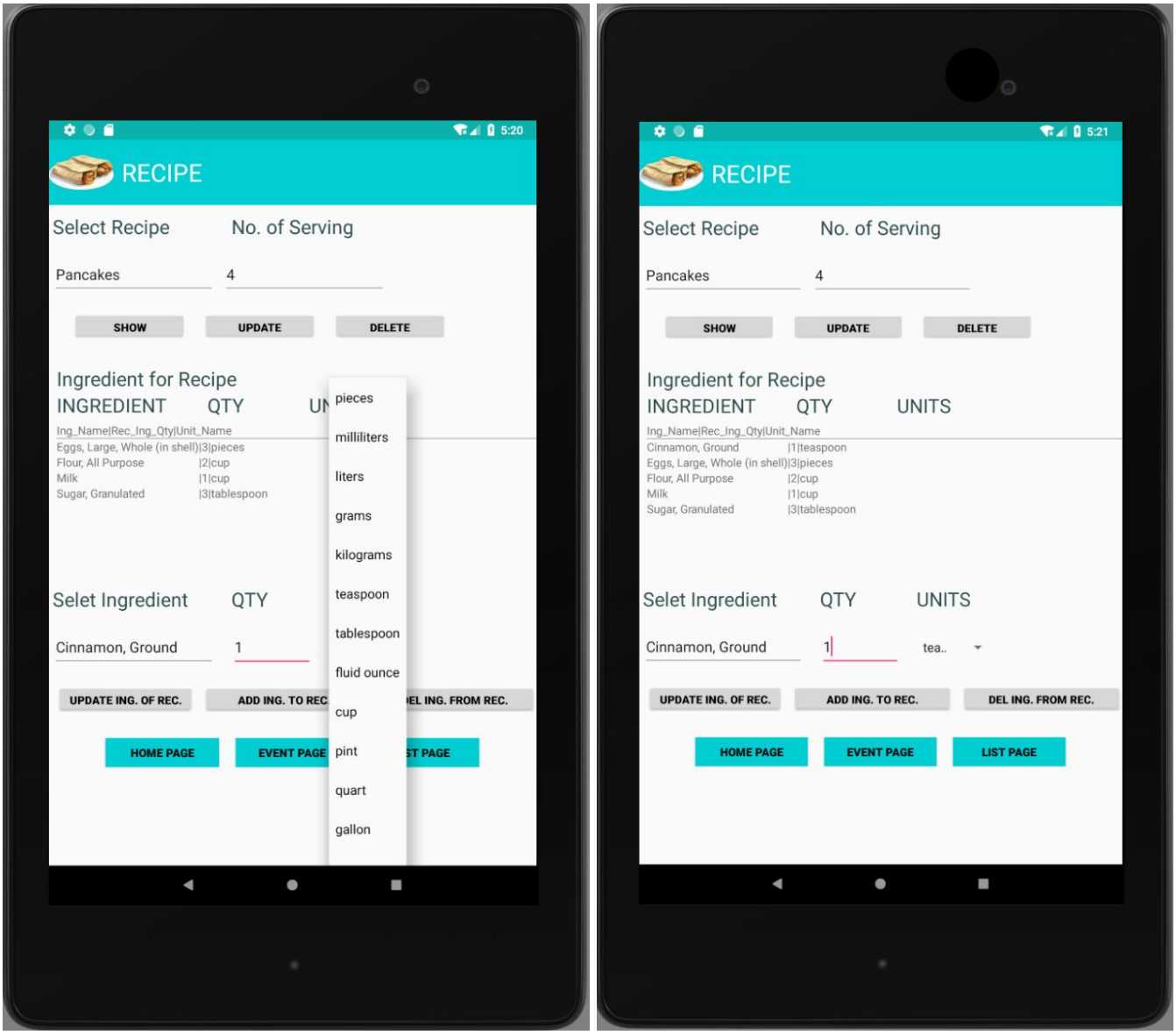

Figure 5c-d Recipe Page c) Select recipe ingredient units of measure, d) Ingredient added to recipe.

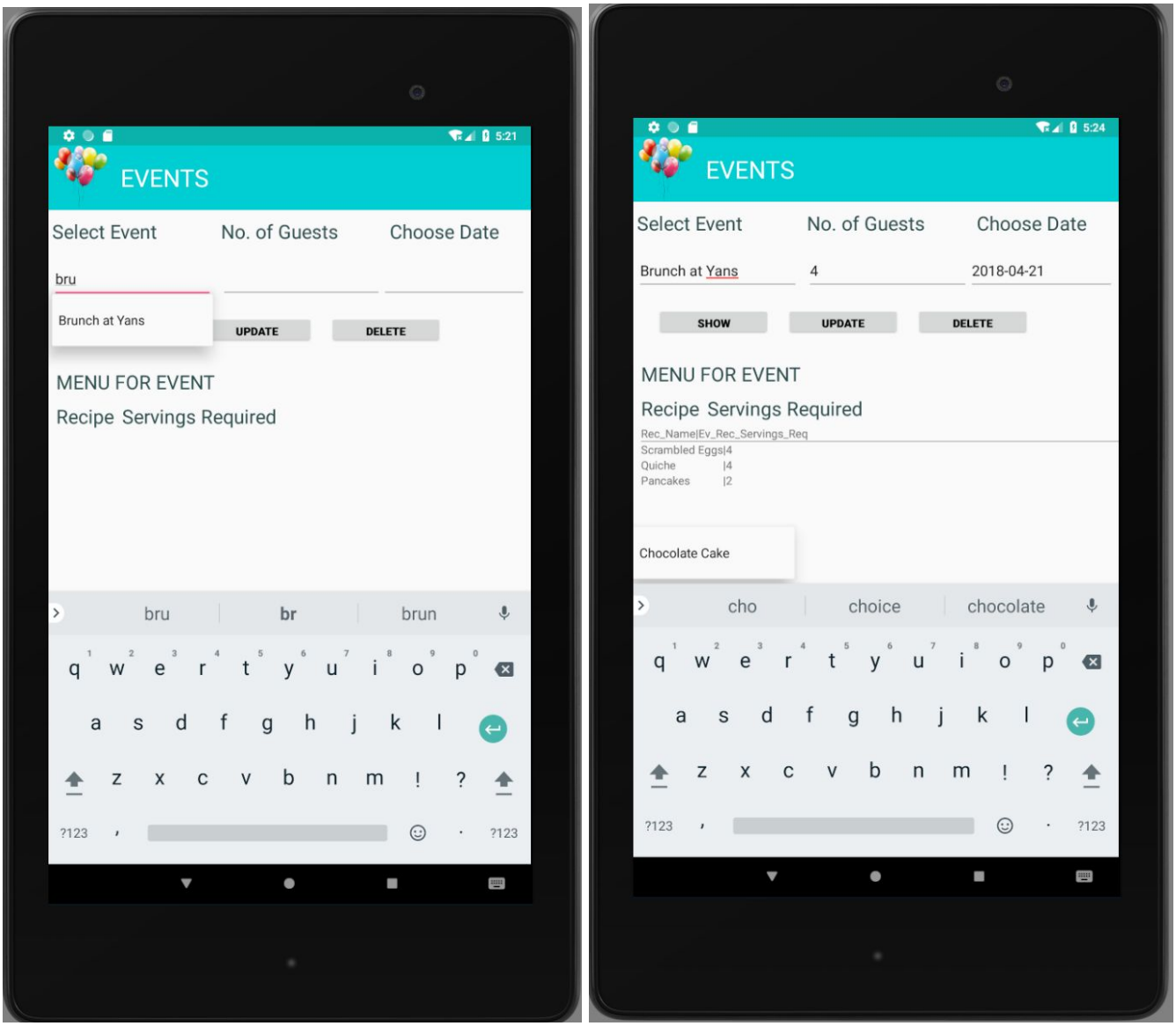

Figure 6 Event Page a) Select event, b) Select new recipe

On the Event UI screen, the user can select an event (using autocomplete to list potential matches), update the number of guests at the event, or delete the event. The user can also view the recipes on the event's menu, add recipes to the event's menu, update the number of servings required or delete a recipe from the event's menu.

The page also includes buttons to navigate back to the home page, or directly to the recipe and list pages.

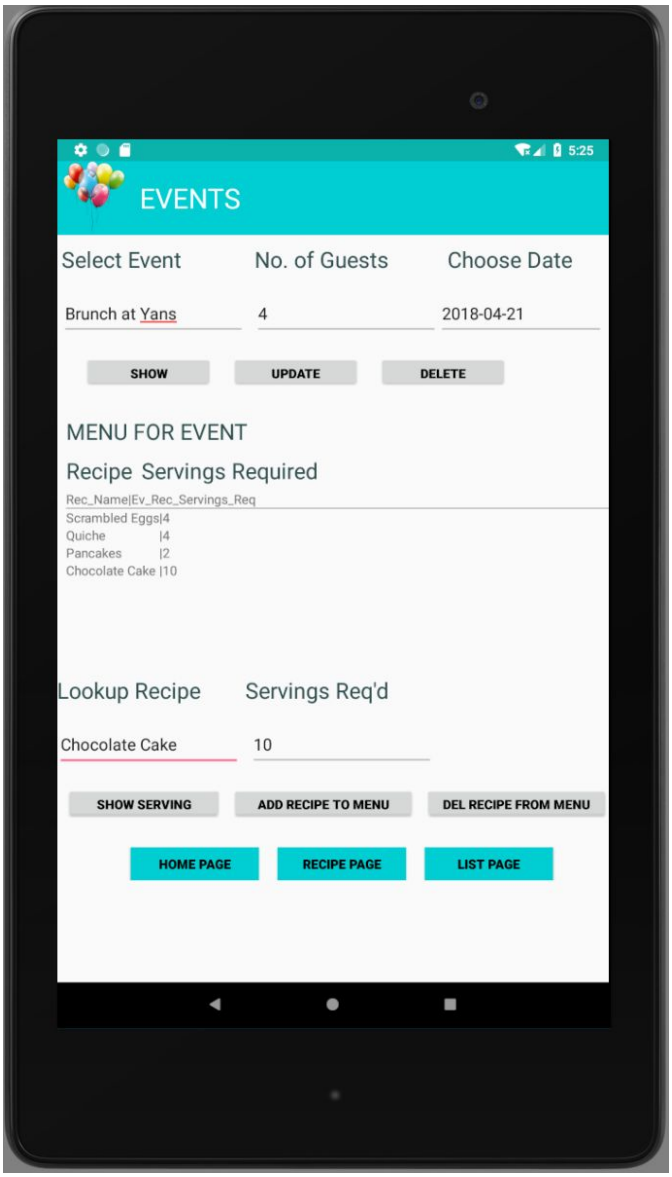

Figure 6c Event Page c) New recipe added

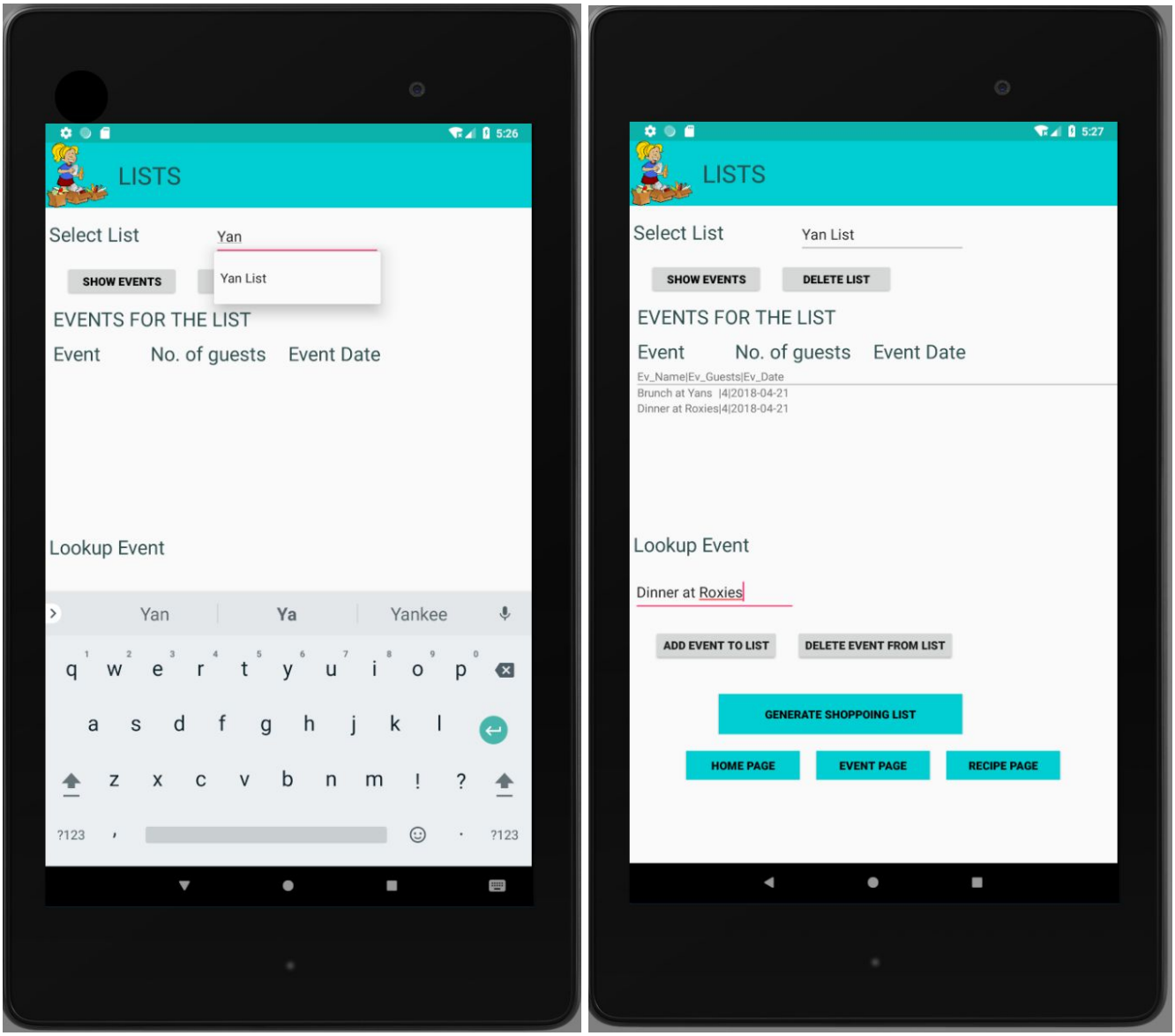

Figure 7 List Page a) Select List, b) Add Event to List

On the Event UI screen, the user can select an list (using autocomplete), show the events associated with that list, or delete the list. The user can also add or delete events from the list.

The page also includes a link to the detailed shopping list page, as well as buttons to navigate back to the home page, or directly to the event and recipe pages.

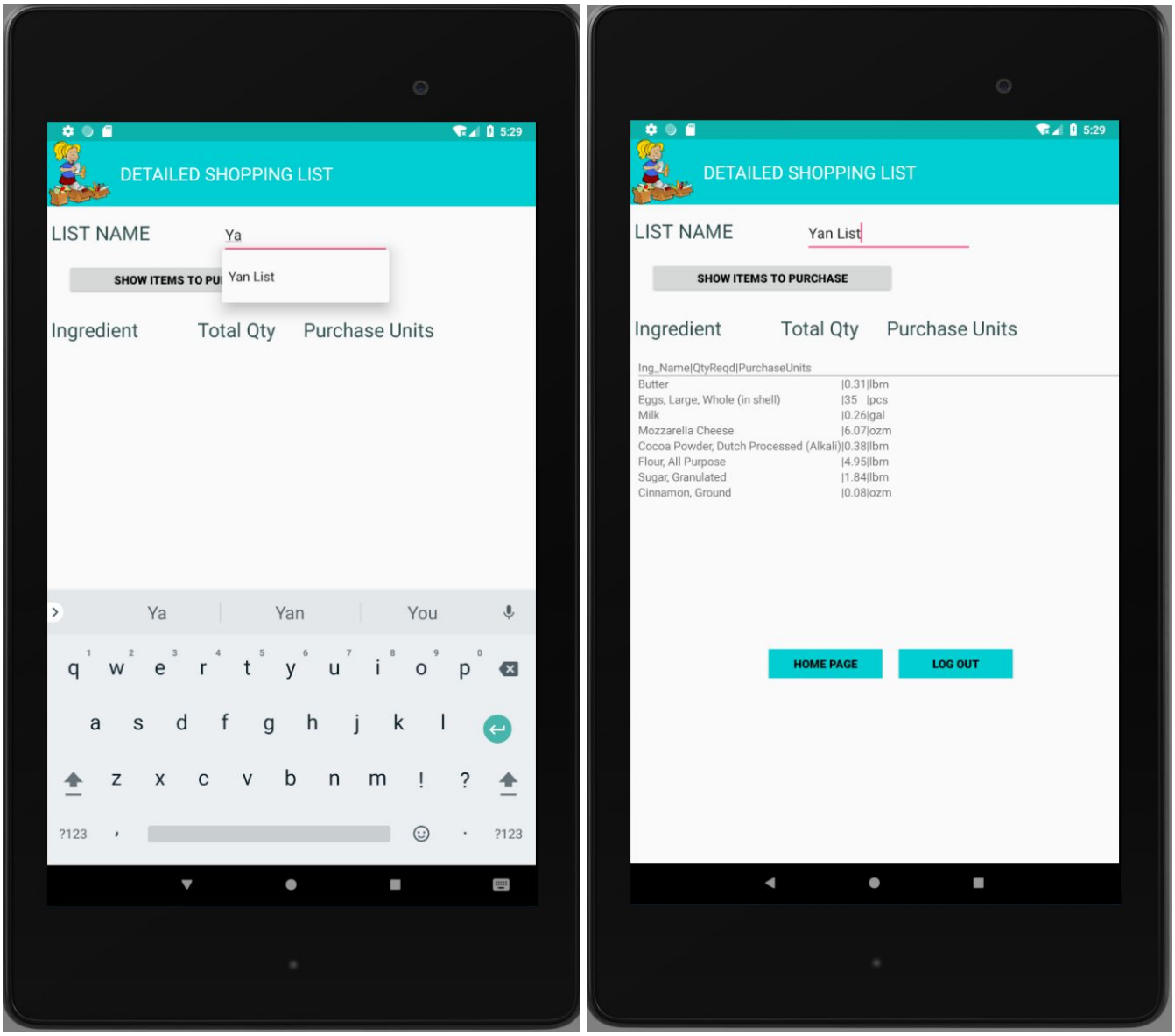

Figure 8 Detailed Shopping List Page a) Select list to view, b) Generate a detailed list of ingredients to be purchased.

The Detailed Shopping List pages allows the use to lookup a list and generate a detailed list of ingredients to be purchased. It also contains buttons to access the home page and to logout.

4. Technical documentation and Implementation

4.1 Space estimates

The current and future space requirements are shown in the table below.

TABLE	Row Size	Number of Rows in Table		Size of Database (MB)	
		Year 1	Year 5	Year 1	Year 5
USER	87	2,000	100,000	0.166	8.3
EVENT	12	30,000	12,500,000	0.343	143.1
LIST	6	10,000	4,166,667	0.057	23.8
RECIPE	6	11,000	3,500,000	0.063	20.0
MENU	6	150,000	87,500,000	0.858	500.7
REC_INGRED	14	88,000	35,000,000	1.175	467.3
UNIT	12	14	20	0.000	0.0
INGREDIENT	30	250	2,500	0.007	0.1
UNIT_TYPE	4	3	3	0.000	0.0
CATEGORY	2	25	40	0.000	0.0
			TOTAL	***2.67***	***1,163***

4.2 Backup & Security

Backup

While the prototype uses the SQLITE database in Android to store all of the data, the full version of the software would store a copy of the recipe and recipe ingredient tables along with user specific event, menu, and list data on the mobile device, and synchronize this to an

cloud based database which contains the data of all users. The cloud based database would be distributed across multiple physical locations (e.g,. AWS data center on the east and west coast). The size of the database is anticipated to be small enough that full replication of all rows of the database tables could be replicated at each physical location. This provide data redundancy and ensure the availability of all users and application data in the event of a network, server, or storage failure at any one location. In addition, full backups of the entire database would be made at monthly, and stored location separate from the distributed cloud database instances locations, with incremental backups made daily and stored at the same location as the full backup. At least three full monthly backups would be retained in addition to current months incremental backup.

Security

While the nature of the data in this application is not sensitive, nonetheless significant attention must be paid to ensure that personal information is not disclosed without the users conset. The full version of the application will implement the following security features:

- User passwords will be stored as a salted hash in both the cloud based and mobile databases. Only the user of the device will have their password stored on the mobile device
- Communication between the mobile application and the cloud based database will be encrypyted (i.e., data encrypted in motion).
- User specific data, including user name, user email address, and all event, menu and list data, will be encrypted to prevent the disclosure of any personally identifiable information (data encrypted at rest). A user specific encryption key will be used to encrypt this information.
- Users will only be able to view their own events, menus, and lists.
- Individual user data will not be shared with third parties. Aggregated data may be shared with the user’s consent.

4.3 Potential data problems and solutions

There are several data quality issues that will need to be addressed in the full version:

- Integrity of Ingredient data. The ingredient data contains information regarding the ingredient density and unit weights that are critical to the operation of the program
    - All ingredient characteristics need to be validated against at least one, and preferably two, independent sources
    - The characteristics need to be screened to ensure that the values are appropriate for the ingredient. Specifically, ingredients that may be sold by the piece (rather than weight or volume) will in most cases require a unit weight.

- Integrity of Recipe data. Users will be able to see and use recipes contributed by others, but there are number of data integrity issues that need to be considered and addressed:
    - Users should only be allowed to edit their own recipes. Users could be allowed to "open source" their recipes, allowing other users to make contributions or "fork" a recipe to make their own version.
    - Any edits or updates will need to be tracked and may required a mechanism to notify other users of the recipes of changes. Depending on the scale of this issue, some form of version control may be required.
    - Users deleting their own recipes will cause problems for other users who have added those recipes to one of their menus, with previously added recipes silently disappearing from their menus. Options include:
        - Allow deleting a recipe to cascade to the menus with that recipe (ON DELETE CASCADE) would maintain the referential integrity of the database, but a notification process similar to that required by updates would be needed. That could take the form of a data validation routine that screens for deleted recipes when a menu is reviewed, warning people that the recipe is no longer available, and perhaps suggesting an alternative dish that uses similar ingredients.

        - Prevent people who have not used the recipe from accessing it, but allowing existing users continued access for at least some period of time.

- Quality of recipes contributed by users. The quality of recipes shared by users is a concern. We have considered several options for ensuring that contributed recipes are of good quality:
    - User ratings and comments to allow users to to provide feedback on the quality of the recipes they use. This feature is straightforward to implement, and will require only moderate effort to maintain.
    - A mechanism to identify potential errors in recipes, and to report errors identified by end users
    - Editorial review of contributed recipe would prevent at least some poor quality recipes to be removed. This would require significant effort, and would have to rely on user ratings and comments to focus editors attention on the recipes that are marginal to be cost effective.
    - Recipes contributed by sponsors (e.g., user ratings and comments will provide a source of vetted, high quality recipes.

  User ratings and comments would be implemented in the full version, and we would seek sponsored recipes to generate revenue to support the application operating and development costs. The need for editorial review will be evaluated during the first year of operation.

4.4. Source Code Listing

Java Code for Android Application (/MenuPlanner_Group5/app/src/main/java/)

```java
package library.group5;

import android.os.Bundle;
import library.group5.util.DBOperator;
import library.group5.view.TableView;
import library.group5.constant.SQLCommand;
import android.app.Activity;
import android.database.Cursor;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.AutoCompleteTextView;
import android.widget.ArrayAdapter;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;


/**
 * Created by Isha on 06-04-2018.
 */

public class EventActivity extends Activity implements OnClickListener{

    private ArrayList<String> eventNameString=new ArrayList<String>();
    private ArrayList<String> recipeNameString=new ArrayList<String>();
    private AutoCompleteTextView autoTextView;
    private AutoCompleteTextView autoTextView1;
    private ArrayAdapter<String>arrayAdapter;
    private ArrayAdapter<String>arrayAdapter1;

    AutoCompleteTextView eventName,recName;
    EditText noOfGuest,eventDate,servReq;
    Button showBtn,updateBtn,deleteBtn,showServ,addRec,delRec,homeBtn,listBtn,recBtn;
    ScrollView menuView;

    String eventID=null;
    String recID=null;

```

```
43
44      @Override
45      public void onCreate(Bundle savedInstanceState) {
46          super.onCreate(savedInstanceState);
47          setContentView(R.layout.event);
48          showBtn = (Button) this.findViewById(R.id.show_btn);
49          showBtn.setOnClickListener(this);
50          updateBtn = (Button) this.findViewById(R.id.update_btn
    );
51          updateBtn.setOnClickListener(this);
52          deleteBtn = (Button) this.findViewById(R.id.delete_btn
    );
53          deleteBtn.setOnClickListener(this);
54          homeBtn = (Button) this.findViewById(R.id.home_btn);
55          homeBtn.setOnClickListener(this);
56          listBtn = (Button) this.findViewById(R.id.list_btn);
57          listBtn.setOnClickListener(this);
58          recBtn = (Button) this.findViewById(R.id.rec_btn);
59          recBtn.setOnClickListener(this);
60          showServ = (Button) this.findViewById(R.id.show_serv);
61          showServ.setOnClickListener(this);
62          addRec = (Button) this.findViewById(R.id.add_rec);
63          addRec.setOnClickListener(this);
64          delRec = (Button) this.findViewById(R.id.del_rec);
65          delRec.setOnClickListener(this);
66          menuView = (ScrollView) this.findViewById(R.id.
    menu_view);
67
68
69          noOfGuest = (EditText) this.findViewById(R.id.
    No_of_Guests);
70          eventDate = (EditText) this.findViewById(R.id.
    eventDate);
71          servReq = (EditText) this.findViewById(R.id.Serv_Req);
72
73          eventName = (AutoCompleteTextView) this.findViewById(R
    .id.EventAutoComp);
74          recName = (AutoCompleteTextView) this.findViewById(R.
    id.recAutoComp);
75
76
77          Cursor cursor = DBOperator.getInstance().execQuery(
78                  SQLCommand.SELECT_EVENT);
79          while (cursor.moveToNext()) {
80              eventNameString.add(cursor.getString(0));
81          }
```

```
 82          Cursor cursor_one = DBOperator.getInstance().
    execQuery(
 83                  SQLCommand.SELECT_RECIPE);
 84          while (cursor_one.moveToNext()) {
 85              recipeNameString.add(cursor_one.getString(0));
 86          }
 87          initialize();
 88
 89      }
 90
 91      private void initialize() {
 92          autoTextView = (AutoCompleteTextView) this
 93                  .findViewById(R.id.EventAutoComp);
 94
 95
 96          // use ArrayAdpater to add content to
    AutoCompleteTextView
 97          arrayAdapter = new ArrayAdapter<String>(this, android
    .R.layout.simple_dropdown_item_1line, eventNameString);
 98          autoTextView.setAdapter(arrayAdapter);
 99
100          autoTextView1 = (AutoCompleteTextView) this
101                  .findViewById(R.id.recAutoComp);
102
103
104          // use ArrayAdpater to add content to
    AutoCompleteTextView
105          arrayAdapter1 = new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line, recipeNameString
    );
106          autoTextView1.setAdapter(arrayAdapter1);
107      }
108
109      public void onClick(View v) {
110          int id = v.getId();
111
112          String[] args = null;
113          args = new String[1];
114          args[0]=eventName.getText().toString();
115
116          String[] argsUpdate = null;
117          argsUpdate = new String[3];
118
119          String[] argsMenu = null;
120          argsMenu = new String[1];
121
```

```
122         String[] argsServ = null;
123         argsServ = new String[1];
124         argsServ[0]=recName.getText().toString();
125
126         String[] argsRec = null;
127         argsRec = new String[3];
128         argsRec[0]=recID;
129         argsRec[1]=eventID;
130         argsRec[2]=servReq.getText().toString();
131
132         String[] argsDel = null;
133         argsDel = new String[2];
134
135         if (id == R.id.show_btn) {
136            menuView.removeAllViews();
137
138             Cursor cursor = DBOperator.getInstance().
    execQuery(
139                     SQLCommand.SELECT_EVENT_BY_NAME,args);
140
141             while (cursor.moveToNext()) {
142                 noOfGuest.setText(cursor.getString(4),
    TextView.BufferType.EDITABLE);
143                 eventDate.setText(cursor.getString(2),
    TextView.BufferType.EDITABLE);
144                 eventID=cursor.getString(0);
145             }
146             argsMenu[0]=eventID;
147
148             Cursor cursor_three = DBOperator.getInstance().
    execQuery(
149                     SQLCommand.SELECT_EVENT_MENU,argsMenu);
150             menuView.addView(new TableView(this.
    getBaseContext(),cursor_three));
151
152         }
153         if (id == R.id.update_btn) {
154             argsUpdate[0]=noOfGuest.getText().toString();
155             argsUpdate[1]=eventDate.getText().toString();
156             argsUpdate[2]=eventID;
157             DBOperator.getInstance().execSQL(SQLCommand.
    UPDATE_EVENT,argsUpdate);
158             Toast.makeText(getBaseContext(), "Event Updated
    successfully", Toast.LENGTH_SHORT).show();
159
160         }
```

```
161         if (id == R.id.delete_btn) {
162             argsMenu[0]=eventID;
163             DBOperator.getInstance().execSQL(SQLCommand.
    DELETE_EVENT,argsMenu);
164             Toast.makeText(getBaseContext(), "Event Deleted
    successfully", Toast.LENGTH_SHORT).show();
165 
166         }
167         if (id == R.id.show_serv) {
168             Cursor cursor = DBOperator.getInstance().
    execQuery(
169                     SQLCommand.SELECT_RECIPE_BY_NAME,argsServ
    );
170 
171             while (cursor.moveToNext()) {
172                 servReq.setText(cursor.getString(2), TextView
    .BufferType.EDITABLE);
173                 recID=cursor.getString(0);
174             }
175         }
176         if (id == R.id.add_rec) {
177             DBOperator.getInstance().execSQL(SQLCommand.
    ADD_RECIPE_MENU,argsRec);
178             Toast.makeText(getBaseContext(), "Recipe for
    event inserted", Toast.LENGTH_SHORT).show();
179 
180         }
181         if (id == R.id.del_rec) {
182             argsDel[0]=eventID;
183             argsDel[1]=recID;
184 
185             DBOperator.getInstance().execSQL(SQLCommand.
    DELETE_RECIPE_FROM_MENU,argsDel);
186             Toast.makeText(getBaseContext(), "Recipe from
    event deleted", Toast.LENGTH_SHORT).show();
187 
188         }
189         if (id == R.id.home_btn) {
190             Intent intent = new Intent(this, HomePageActivity
    .class);
191             this.startActivity(intent);
192         }
193         if(id == R.id.rec_btn){
194             Intent intent = new Intent(this, RecipeActivity.
    class);
195             this.startActivity(intent);
```

```
196
197         }
198         if(id == R.id.list_btn){
199             Intent intent = new Intent(this,
    ManageListActivity.class);
200             this.startActivity(intent);
201
202         }
203
204     }
205
206 }
207
```

```java
package library.group5;

import android.os.Bundle;
import library.group5.util.DBOperator;
import library.group5.constant.SQLCommand;
import android.app.Activity;
import android.database.Cursor;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class group5Activity extends Activity implements OnClickListener {

    Button submitBtn;
    EditText emailIdEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.landingpage);
        emailIdEdit = (EditText) this.findViewById(R.id.email_id);
        submitBtn = (Button) this.findViewById(R.id.submit_btn);
        submitBtn.setOnClickListener(this);
        try{
            DBOperator.copyDB(getBaseContext());
        }catch(Exception e){
            e.printStackTrace();
        }

    }

    public void onClick(View v) {
        int b=0;
        int id = v.getId();
        String emailValue=emailIdEdit.getText().toString();

        if (id == R.id.submit_btn) {
            Cursor cursor = DBOperator.getInstance().execQuery(
                    SQLCommand.SELECT_USER_EMAIL);
            while (cursor.moveToNext()) {
```

```
43                  String cursorValue = cursor.getString(0);
44                  if (cursorValue.equals(emailValue)) {
45                      b=1;
46                  }
47
48              }
49              if (b == 1){
50                  Intent intent = new Intent(this,
    HomePageActivity.class);
51                  this.startActivity(intent);
52              }
53              else{
54                  Intent intent = new Intent(this,
    NewCustomerActivity.class);
55                  this.startActivity(intent);
56              }
57
58          }
59
60      }
61
62
63  }
64
```

```java
package library.group5;

import android.os.Bundle;
import library.group5.util.DBOperator;
import library.group5.view.TableView;
import library.group5.constant.SQLCommand;
import android.app.Activity;
import android.database.Cursor;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.AutoCompleteTextView;
import android.widget.ArrayAdapter;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;

/**
 * Created by Isha on 06-04-2018.
 */

public class RecipeActivity extends Activity implements OnClickListener{
    private ArrayList<String> recipeNameString=new ArrayList<String>();
    private ArrayList<String> ingNameString=new ArrayList<String>();
    private ArrayList<String> unitNameString=new ArrayList<String>();


    private AutoCompleteTextView autoTextView;
    private AutoCompleteTextView autoTextView1;
    private Spinner spinner;
    private ArrayAdapter<String>arrayAdapter;
    private ArrayAdapter<String>arrayAdapter1;
    private ArrayAdapter<String>arrayAdapter2;

    AutoCompleteTextView recName,ingName;
    EditText noOfServ,qty;
    Button showIngBtn,updateRecBtn,deleteRecBtn,updateIng,addIng,delIng,homeBtn,listBtn,eventBtn;
```

```
42      ScrollView recView;
43      Spinner unitSpinner;
44
45      String recID=null,ingID=null,unitID=null;
46
47
48      @Override
49      public void onCreate(Bundle savedInstanceState) {
50          super.onCreate(savedInstanceState);
51          setContentView(R.layout.recipe);
52
53          showIngBtn = (Button) this.findViewById(R.id.
    showIng_btn);
54          showIngBtn.setOnClickListener(this);
55          updateRecBtn = (Button) this.findViewById(R.id.
    updateRec_btn);
56          updateRecBtn.setOnClickListener(this);
57          deleteRecBtn = (Button) this.findViewById(R.id.
    deleteRec_btn);
58          deleteRecBtn.setOnClickListener(this);
59          updateIng = (Button) this.findViewById(R.id.
    updateIng_btn);
60          updateIng.setOnClickListener(this);
61          addIng = (Button) this.findViewById(R.id.addIng_btn);
62          addIng.setOnClickListener(this);
63          delIng = (Button) this.findViewById(R.id.delIng_btn);
64          delIng.setOnClickListener(this);
65          homeBtn = (Button) this.findViewById(R.id.homePage_btn
    );
66          homeBtn.setOnClickListener(this);
67          listBtn = (Button) this.findViewById(R.id.listPage_btn
    );
68          listBtn.setOnClickListener(this);
69          eventBtn = (Button) this.findViewById(R.id.
    eventPage_btn);
70          eventBtn.setOnClickListener(this);
71          recView = (ScrollView) this.findViewById(R.id.rec_view
    );
72
73
74          noOfServ = (EditText) this.findViewById(R.id.
    no_of_serv);
75          qty = (EditText) this.findViewById(R.id.ingQty);
76
77          recName = (AutoCompleteTextView) this.findViewById(R.
    id.recAutoComp);
```

```
 78          ingName = (AutoCompleteTextView) this.findViewById(R.
    id.ingAutoComp);
 79          unitSpinner = (Spinner) this.findViewById(R.id.
    unitId_spinner);
 80
 81
 82          Cursor cursor_one = DBOperator.getInstance().
    execQuery(
 83                  SQLCommand.SELECT_RECIPE);
 84          while (cursor_one.moveToNext()) {
 85              recipeNameString.add(cursor_one.getString(0));
 86          }
 87
 88          Cursor cursor_two = DBOperator.getInstance().
    execQuery(
 89                  SQLCommand.SELECT_INGREDIENT);
 90          while (cursor_two.moveToNext()) {
 91              ingNameString.add(cursor_two.getString(0));
 92          }
 93
 94          Cursor cursor_three = DBOperator.getInstance().
    execQuery(
 95                  SQLCommand.SELECT_UNIT);
 96          while (cursor_three.moveToNext()) {
 97              unitNameString.add(cursor_three.getString(0));
 98          }
 99          initialize();
100
101      }
102
103      private void initialize() {
104          autoTextView = (AutoCompleteTextView) this
105                  .findViewById(R.id.recAutoComp);
106
107          // use ArrayAdpater to add content to
    AutoCompleteTextView
108          arrayAdapter = new ArrayAdapter<String>(this, android
    .R.layout.simple_dropdown_item_1line, recipeNameString);
109          autoTextView.setAdapter(arrayAdapter);
110
111          autoTextView1 = (AutoCompleteTextView) this
112                  .findViewById(R.id.ingAutoComp);
113          arrayAdapter1 = new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line, ingNameString);
114          autoTextView1.setAdapter(arrayAdapter1);
115
```

```
116         spinner = (Spinner) this
117                 .findViewById(R.id.unitId_spinner);
118
119         // set items list and textview style of items in
    ArrayAdapter
120         arrayAdapter2 = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, unitNameString);
121         // set the style of drop down views
122         arrayAdapter2.setDropDownViewResource(android.R.
    layout.simple_spinner_dropdown_item);
123         // set the ArrayAdapter into Spinner
124         spinner.setAdapter(arrayAdapter2);
125
126     }
127
128
129     public void onClick(View v) {
130         int id = v.getId();
131         String[] argsRec = null;
132         argsRec = new String[1];
133         argsRec[0]=recName.getText().toString();
134
135         String[] args = null;
136         args = new String[1];
137
138         String[] argsRecUpdate = null;
139         argsRecUpdate = new String[2];
140
141         String[] argsIng = null;
142         argsIng = new String[1];
143         argsIng[0]=ingName.getText().toString();
144
145         String[] argsUnit = null;
146         argsUnit = new String[1];
147         argsUnit[0]=unitSpinner.getSelectedItem().toString();
148
149         String[] argsIngUpdate = null;
150         argsIngUpdate = new String[4];
151
152         String[] argsIngDel = null;
153         argsIngDel = new String[2];
154
155
156         if (id == R.id.showIng_btn) {
157
158             recView.removeAllViews();
```

```
159             Cursor cursor = DBOperator.getInstance().
    execQuery(
160                     SQLCommand.SELECT_RECIPE_BY_NAME,argsRec)
    ;
161
162             while (cursor.moveToNext()) {
163                 noOfServ.setText(cursor.getString(2),
    TextView.BufferType.EDITABLE);
164                 recID=cursor.getString(0);
165             }
166             args[0]=recID;
167             Cursor cursor_one = DBOperator.getInstance().
    execQuery(
168                     SQLCommand.
    SELECT_RECIPE_INGREDIENT_DETAIL,args);
169             recView.addView(new TableView(this.getBaseContext
    (),cursor_one));
170
171         }
172
173         else if (id == R.id.updateRec_btn) {
174             argsRecUpdate[1]=recID;
175             argsRecUpdate[0]=noOfServ.getText().toString();;
176             DBOperator.getInstance().execSQL(SQLCommand.
    UPDATE_RECIPE_SERVINGS,argsRecUpdate);
177             Toast.makeText(getBaseContext(), "Recipe Updated
    successfully", Toast.LENGTH_SHORT).show();
178         }
179
180         else if (id == R.id.deleteRec_btn) {
181             args[0]=recID;
182             DBOperator.getInstance().execSQL(SQLCommand.
    DELETE_RECIPE,args);
183             Toast.makeText(getBaseContext(), "Recipe Deleted
    successfully", Toast.LENGTH_SHORT).show();
184         }
185
186         else if (id == R.id.updateIng_btn) {
187             Cursor cursor = DBOperator.getInstance().
    execQuery(
188                     SQLCommand.SELECT_INGREDIENT_BY_NAME,
    argsIng);
189
190             while (cursor.moveToNext()) {
191                 ingID=cursor.getString(0);
192             }
```

```
193
194             Cursor cursor_two = DBOperator.getInstance().
    execQuery(
195                     SQLCommand.SELECT_UNIT_BY_NAME,argsUnit);
196
197             while (cursor_two.moveToNext()) {
198                 unitID=cursor_two.getString(0);
199             }
200
201             argsIngUpdate[0]=qty.getText().toString();
202             argsIngUpdate[1]=unitID;
203             argsIngUpdate[2]=recID;
204             argsIngUpdate[3]=ingID;
205
206             DBOperator.getInstance().execSQL(SQLCommand.
    UPDATE_REC_ING_QTY_UNIT,argsIngUpdate);
207             Toast.makeText(getBaseContext(), "Recipe
    Ingredient Updated successfully", Toast.LENGTH_SHORT).show();
208
209         }
210         else if (id == R.id.addIng_btn) {
211             Cursor cursor = DBOperator.getInstance().
    execQuery(
212                     SQLCommand.SELECT_INGREDIENT_BY_NAME,
    argsIng);
213
214             while (cursor.moveToNext()) {
215                 ingID=cursor.getString(0);
216             }
217
218             Cursor cursor_two = DBOperator.getInstance().
    execQuery(
219                     SQLCommand.SELECT_UNIT_BY_NAME,argsUnit);
220
221             while (cursor_two.moveToNext()) {
222                 unitID=cursor_two.getString(0);
223             }
224             argsIngUpdate[0]=recID;
225             argsIngUpdate[1]=ingID;
226             argsIngUpdate[2]=qty.getText().toString();
227             argsIngUpdate[3]=unitID;
228
229             DBOperator.getInstance().execSQL(SQLCommand.
    ADD_ING_TO_REC,argsIngUpdate);
230             Toast.makeText(getBaseContext(), "Recipe
    Ingredient Added successfully", Toast.LENGTH_SHORT).show();
```

```
231
232
233         }
234         else if (id == R.id.delIng_btn) {
235
236             Cursor cursor = DBOperator.getInstance().
    execQuery(
237                     SQLCommand.SELECT_INGREDIENT_BY_NAME,
    argsIng);
238
239             while (cursor.moveToNext()) {
240                 ingID=cursor.getString(0);
241             }
242             argsIngDel[0]=recID;
243             argsIngDel[1]=ingID;
244
245             DBOperator.getInstance().execSQL(SQLCommand.
    DEL_ING_FROM_RECIPE,argsIngDel);
246             Toast.makeText(getBaseContext(), "Recipe
    Ingredient Deleted successfully", Toast.LENGTH_SHORT).show();
247
248         }
249         else if (id == R.id.homePage_btn) {
250             Intent intent = new Intent(this, HomePageActivity
    .class);
251             this.startActivity(intent);
252         }
253         else if (id == R.id.listPage_btn) {
254             Intent intent = new Intent(this,
    ManageListActivity.class);
255             this.startActivity(intent);
256         }
257         else if (id == R.id.eventPage_btn) {
258             Intent intent = new Intent(this, EventActivity.
    class);
259             this.startActivity(intent);
260         }
261
262     }
263
264 }
265
```

```java
package library.group5;

import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.app.Activity;

/**
 * Created by Isha on 06-04-2018.
 */

public class HomePageActivity extends Activity implements OnClickListener{

    Button recipeBtn,eventBtn,listBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.homepage);
        recipeBtn = (Button) this.findViewById(R.id.recipe_btn);
        recipeBtn.setOnClickListener(this);
        eventBtn = (Button) this.findViewById(R.id.event_btn);
        eventBtn.setOnClickListener(this);
        listBtn = (Button) this.findViewById(R.id.list_btn);
        listBtn.setOnClickListener(this);
    }
    public void onClick(View v) {
        int id = v.getId();

        if (id == R.id.recipe_btn) {
            Intent intent = new Intent(this, RecipeActivity.class);
            this.startActivity(intent);
        }
        else if(id == R.id.event_btn){
            Intent intent = new Intent(this, EventActivity.class);
            this.startActivity(intent);

        }
        else if(id == R.id.list_btn){
            Intent intent = new Intent(this,
```

```
42 ManageListActivity.class);
43             this.startActivity(intent);
44
45         }
46
47     }
48
49 }
50
```

```java
package library.group5;

import android.os.Bundle;
import library.group5.util.DBOperator;
import library.group5.view.TableView;
import library.group5.constant.SQLCommand;
import android.app.Activity;
import android.database.Cursor;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.AutoCompleteTextView;
import android.widget.ArrayAdapter;
import android.widget.ScrollView;
import android.widget.Toast;
import java.util.ArrayList;
/**
 * Created by Isha on 06-04-2018.
 */

public class ManageListActivity extends Activity implements
OnClickListener{

        private ArrayList<String> listNameString=new ArrayList
<String>();
        private ArrayList<String> eventNameString=new
ArrayList<String>();
        private AutoCompleteTextView autoTextView;
        private AutoCompleteTextView autoTextView1;
        private ArrayAdapter<String>arrayAdapter;
        private ArrayAdapter<String>arrayAdapter1;

       AutoCompleteTextView listName,eventName;
       Button showEvent,delList,addEvent,delEvent,homeBtn,
recBtn,eventBtn,ShopBtn;
       ScrollView eventView;
       String listID=null, eventID=null;



    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.managelist);

```

```
43          showEvent = (Button) this.findViewById(R.id.show_event
    );
44          showEvent.setOnClickListener(this);
45          delList = (Button) this.findViewById(R.id.del_list);
46          delList.setOnClickListener(this);
47          addEvent = (Button) this.findViewById(R.id.add_event);
48          addEvent.setOnClickListener(this);
49          delEvent = (Button) this.findViewById(R.id.del_event);
50          delEvent.setOnClickListener(this);
51          homeBtn = (Button) this.findViewById(R.id.home_Btn);
52          homeBtn.setOnClickListener(this);
53          recBtn = (Button) this.findViewById(R.id.rec_Btn);
54          recBtn.setOnClickListener(this);
55          eventBtn = (Button) this.findViewById(R.id.event_btn);
56          eventBtn.setOnClickListener(this);
57          ShopBtn = (Button) this.findViewById(R.id.shop_Btn);
58          ShopBtn.setOnClickListener(this);
59
60          eventView = (ScrollView) this.findViewById(R.id.
    event_view);
61
62          listName = (AutoCompleteTextView) this.findViewById(R.
    id.listAutoComp);
63          eventName = (AutoCompleteTextView) this.findViewById(R
    .id.eventAutoComp);
64
65
66          Cursor cursor_one = DBOperator.getInstance().execQuery
    (
67                  SQLCommand.SELECT_LIST);
68          while (cursor_one.moveToNext()) {
69              listNameString.add(cursor_one.getString(0));
70          }
71
72          Cursor cursor_two = DBOperator.getInstance().execQuery
    (
73                  SQLCommand.SELECT_EVENT);
74          while (cursor_two.moveToNext()) {
75              eventNameString.add(cursor_two.getString(0));
76          }
77
78          initialize();
79      }
80
81      private void initialize() {
82          autoTextView = (AutoCompleteTextView) this
```

```
 83                 .findViewById(R.id.listAutoComp);
 84
 85         // use ArrayAdpater to add content to
    AutoCompleteTextView
 86         arrayAdapter = new ArrayAdapter<String>(this, android
    .R.layout.simple_dropdown_item_1line, listNameString);
 87         autoTextView.setAdapter(arrayAdapter);
 88
 89         autoTextView1 = (AutoCompleteTextView) this
 90                 .findViewById(R.id.eventAutoComp);
 91
 92
 93         arrayAdapter1 = new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line, eventNameString)
    ;
 94         autoTextView1.setAdapter(arrayAdapter1);
 95     }
 96
 97     public void onClick(View v) {
 98         int id = v.getId();
 99
100         String[] args = null;
101         args = new String[1];
102         args[0]=listName.getText().toString();
103
104         String[] argsId = null;
105         argsId = new String[1];
106         argsId[0]=eventName.getText().toString();
107
108         String[] argList = null;
109         argList = new String[1];
110
111         String[] argEvent = null;
112         argEvent = new String[2];
113
114         String[] argDel = null;
115         argDel = new String[1];
116
117
118         if (id == R.id.show_event) {
119             eventView.removeAllViews();
120
121             Cursor cursor = DBOperator.getInstance().
    execQuery(
122                     SQLCommand.SELECT_LIST_BY_NAME,args);
123
```

```
124             while (cursor.moveToNext()) {
125                 listID=cursor.getString(0);
126             }
127             argList[0]=listID;
128 
129             Cursor cursor_one = DBOperator.getInstance().
    execQuery(
130                     SQLCommand.SELECT_EVENT_LIST,argList);
131             eventView.addView(new TableView(this.
    getBaseContext(),cursor_one));
132 
133 
134         }
135         else if(id == R.id.del_list){
136             argList[0]=listID;
137             DBOperator.getInstance().execSQL(SQLCommand.
    DELETE_LIST,argList);
138             Toast.makeText(getBaseContext(), "List Deleted
    successfully", Toast.LENGTH_SHORT).show();
139 
140         }
141         else if(id == R.id.add_event){
142             Cursor cursor = DBOperator.getInstance().
    execQuery(
143                     SQLCommand.SELECT_EVENT_BY_NAME,argsId);
144             while (cursor.moveToNext()) {
145                 eventID=cursor.getString(0);
146             }
147             argEvent[0]=listID;
148             argEvent[1]=eventID;
149             DBOperator.getInstance().execSQL(SQLCommand.
    ADD_EVENT_TO_LIST,argEvent);
150             Toast.makeText(getBaseContext(), "Event Added to
    the list successfully", Toast.LENGTH_SHORT).show();
151 
152         }
153         else if(id == R.id.del_event){
154             Cursor cursor = DBOperator.getInstance().
    execQuery(
155                     SQLCommand.SELECT_EVENT_BY_NAME,argsId);
156             while (cursor.moveToNext()) {
157                 eventID=cursor.getString(0);
158             }
159             argDel[0]=eventID;
160             DBOperator.getInstance().execSQL(SQLCommand.
    DELETE_EVENT_FROM_LIST,argDel);
```

```
161             Toast.makeText(getBaseContext(), "Event deleted
    from the list successfully", Toast.LENGTH_SHORT).show();
162
163         }
164         else if(id == R.id.shop_Btn){
165             Intent intent = new Intent(this,
    ShoppingListActivity.class);
166             this.startActivity(intent);
167         }
168         else if(id == R.id.home_Btn){
169             Intent intent = new Intent(this, HomePageActivity
    .class);
170             this.startActivity(intent);
171         }
172         else if(id == R.id.event_btn){
173             Intent intent = new Intent(this, EventActivity.
    class);
174             this.startActivity(intent);
175         }
176         else if(id == R.id.rec_Btn){
177             Intent intent = new Intent(this, RecipeActivity.
    class);
178             this.startActivity(intent);
179         }
180
181
182     }
183 }
184
```

```java
package library.group5;
import android.content.Intent;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Calendar;

import library.group5.util.DBOperator;
import library.group5.constant.SQLCommand;

/**
 * Created by Isha on 06-04-2018.
 */

public class NewCustomerActivity extends Activity implements
OnClickListener {

    Button registerBtn;
    EditText userNameEdit,passwordEdit,emailIdEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.newcustomer);
        emailIdEdit = (EditText) this.findViewById(R.id.
user_email);
        userNameEdit = (EditText) this.findViewById(R.id.
user_name);
        passwordEdit = (EditText) this.findViewById(R.id.
user_pwd);
        registerBtn = (Button) this.findViewById(R.id.
register_btn);
        registerBtn.setOnClickListener(this);
    }


    public void onClick(View v) {
        int id = v.getId();
        String[] args = null;
        args = new String[3];
```

```
42
43          args[0]=userNameEdit.getText().toString();
44          args[1]=emailIdEdit.getText().toString();
45          args[2]=passwordEdit.getText().toString();
46
47          if (id == R.id.register_btn) {
48              DBOperator.getInstance().execSQL(SQLCommand.
    INSERT_USER,args);
49              Toast.makeText(getBaseContext(), "inserted", Toast
    .LENGTH_SHORT).show();
50
51              Intent intent = new Intent(this, HomePageActivity.
    class);
52              this.startActivity(intent);
53
54          }
55
56      }
57
58
59  }
60
```

```java
package library.group5;

/**
 * Created by Isha on 16-04-2018.
 */
import android.os.Bundle;
import library.group5.util.DBOperator;
import library.group5.view.TableView;
import library.group5.constant.SQLCommand;
import android.app.Activity;
import android.database.Cursor;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.AutoCompleteTextView;
import android.widget.ArrayAdapter;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;


public class ShoppingListActivity extends Activity implements OnClickListener {

    private ArrayList<String> listNameString=new ArrayList<String>();
    private AutoCompleteTextView autoTextView;
    private ArrayAdapter<String>arrayAdapter;
    AutoCompleteTextView listName;
    Button showItem,homeBtn,logOutBtn;
    ScrollView ingView;
    String listID=null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.shoppinglist);

        showItem = (Button) this.findViewById(R.id.show_item);
        showItem.setOnClickListener(this);
        homeBtn = (Button) this.findViewById(R.id.shop_homebtn);
        homeBtn.setOnClickListener(this);
```

```
44          logOutBtn = (Button) this.findViewById(R.id.log_out);
45          logOutBtn.setOnClickListener(this);
46
47          ingView = (ScrollView) this.findViewById(R.id.ing_view
    );
48          listName = (AutoCompleteTextView) this.findViewById(R.
    id.list_auto_comp);
49
50          Cursor cursor_one = DBOperator.getInstance().execQuery
    (
51                  SQLCommand.SELECT_LIST);
52          while (cursor_one.moveToNext()) {
53              listNameString.add(cursor_one.getString(0));
54          }
55
56          initialize();
57
58      }
59
60      private void initialize() {
61          autoTextView = (AutoCompleteTextView) this
62                  .findViewById(R.id.list_auto_comp);
63
64          // use ArrayAdpater to add content to
    AutoCompleteTextView
65          arrayAdapter = new ArrayAdapter<String>(this, android.
    R.layout.simple_dropdown_item_1line, listNameString);
66          autoTextView.setAdapter(arrayAdapter);
67      }
68
69      public void onClick(View v) {
70          int id = v.getId();
71
72          String[] args = null;
73          args = new String[1];
74          args[0]=listName.getText().toString();
75
76          String[] argList = null;
77          argList = new String[1];
78
79          if (id == R.id.show_item) {
80              ingView.removeAllViews();
81
82              Cursor cursor = DBOperator.getInstance().execQuery
    (
83                      SQLCommand.SELECT_LIST_BY_NAME,args);
```

```
 84
 85              while (cursor.moveToNext()) {
 86                  listID=cursor.getString(0);
 87              }
 88              argList[0]=listID;
 89
 90              Cursor cursor_one = DBOperator.getInstance().
    execQuery(
 91                      SQLCommand.CREATE_SHOPPING_LIST,argList);
 92              ingView.addView(new TableView(this.getBaseContext
    (),cursor_one));
 93
 94          }
 95          else if (id == R.id.shop_homebtn) {
 96              Intent intent = new Intent(this, HomePageActivity
    .class);
 97              this.startActivity(intent);
 98          }
 99          else if (id == R.id.log_out) {
100              Intent intent = new Intent(this, group5Activity.
    class);
101              this.startActivity(intent);
102          }
103
104
105      }
106
107 }
108
```

```java
package library.group5.util;

/**
 * Created by Isha on 05-04-2018.
 */

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import library.group5.constant.DBConstant;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class DBOperator {
    private static DBOperator instance = null;
    private SQLiteDatabase db;

    private DBOperator()
    {
//path of database file
        String path = DBConstant.DATABASE_PATH + "/" + DBConstant.DATABASE_FILE;
        System.out.println("path"+path);
        db = SQLiteDatabase.openDatabase(path, null, SQLiteDatabase.OPEN_READWRITE);

    }
    /*
     * Singleton Pattern
     * Why should we avoid multiple instances here?
     */

    public static DBOperator getInstance()
    {
        if (instance==null) instance = new DBOperator();
        return instance;
    }
    /**
     * Copy database file
```

```
45      * From assets folder (in the project) to android folder (
    on device)
46      */
47     public static void copyDB(Context context) throws
    IOException,FileNotFoundException{
48         String path = DBConstant.DATABASE_PATH + "/" +
    DBConstant.DATABASE_FILE;
49         File file = new File(path);
50         if(!file.exists()){
51             DBOpenHelper dbhelper = new DBOpenHelper(context,
    path ,1);
52             dbhelper.getWritableDatabase();
53             InputStream is =context.getAssets().open(
    DBConstant.DATABASE_FILE);
54             OutputStream os = new FileOutputStream(file);
55             byte[] buffer = new byte[1024];
56             int length;
57             while ((length = is.read(buffer))>0){
58                 os.write(buffer, 0, length);
59             }
60             is.close();
61             os.flush();
62             os.close();
63         }
64     }
65
66     /**
67      * execute sql without returning data, such as alter
68      * @paramsql
69      */
70     public void execSQL(String sql) throws SQLException
71     {
72         db.execSQL(sql);
73     }
74     /**
75      * execute sql such as update/delete/insert
76      * @paramsql
77      * @paramargs
78      * @throwsSQLException
79      */
80     public void execSQL(String sql, Object[] args) throws
    SQLException
81     {
82         db.execSQL(sql, args);
83     }
84     /**
```

```
 85      * execute sqlquery
 86      * @paramsql
 87      * @paramselectionArgs
 88      * @return cursor
 89      * @throwsSQLException
 90      */
 91     public Cursor execQuery(String sql,String[] selectionArgs
    ) throws SQLException
 92     {
 93         return db.rawQuery(sql, selectionArgs);
 94     }
 95     /**
 96      * execute query without arguments
 97      * @paramsql
 98      * @return
 99      * @throwsSQLException
100      */
101     public Cursor execQuery(String sql) throws SQLException
102     {
103         return this.execQuery(sql, null);
104     }
105     /**
106      * close database
107      */
108     public void closeDB()
109     {
110         if (db!=null) db.close();
111     }
112
113 }
114
```

```java
package library.group5.util;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

/**
 * Created by Isha on 05-04-2018.
 */

public class DBOpenHelper  extends SQLiteOpenHelper {
    public DBOpenHelper(Context context, String path, int version){
        super(context, path, null, version);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }

}
```

```java
package library.group5.view;

import android.content.Context;
import android.database.Cursor;
import android.view.View;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import library.group5.util.DBOperator;


/**
 * Created by Isha on 12-04-2018.
 */

public class TableView extends TableLayout {

    public TableView(Context context, String tableName)
    {
        super(context);
        String sql = "select * from " + tableName + ";";
        DBOperator op = DBOperator.getInstance();
        Cursor cursor = op.execQuery(sql, null);
        this.extractData(context, cursor);
    }
    public TableView(Context context, Cursor cursor)
    {
        super(context);
        this.extractData(context, cursor);
    }
    /*
     * fill data in table view with a cursor
     */
    private void extractData(Context context,Cursor cursor)
    {
        TextView textView;
        TableRow row;
        boolean first = true;
        while (cursor.moveToNext())
        {
            /*
 * Before displaying the first row,
 * display column names as a header
 */
            if (first){
                textView = new TextView(context);
```

```
47                  String[] columnNames = cursor.getColumnNames()
    ;
48                  StringBuilder strBuilder = new StringBuilder()
    ;
49                  for (int i=0;i<columnNames.length;i++){
50                      if (i>0) strBuilder.append("|");
51                      strBuilder.append(columnNames[i]);
52                  }
53                  textView.setText(strBuilder);
54                  this.addView(textView);
55                  //show separation line
56                  View line = new View(context);
57                  line.setLayoutParams(new TableLayout.
    LayoutParams(TableLayout.LayoutParams.FILL_PARENT,2));
58                  line.setBackgroundColor(0xFF909090);
59                  this.addView(line);
60                  first = false;
61              }
62  //show values in a row
63              row = new TableRow(context);
64              int length = cursor.getColumnCount();
65              for (int i=0;i<length;i++)
66              {
67                  if (i>0){
68                      textView = new TextView(context);
69                      textView.setText("|");
70                      row.addView(textView);
71                  }
72                  textView = new TextView(context);
73                  textView.setText(cursor.getString(i));
74                  row.addView(textView);
75              }
76              this.addView(row);
77          }
78  /*
79   * Do not forget to close the cursor!
80   * Otherwise database exceptions will be thrown
81   */
82          cursor.close();
83      }
84  }
85
```

```java
package library.group5.constant;

/**
 * Created by Isha on 05-04-2018.
 */

public abstract class DBConstant {

    //database file directory
    public static String DATABASE_PATH ="/data/data/library.group5/databases";
    //database file name
    public static String DATABASE_FILE = "Menu_Planner.db";
    //database version
    public static int DATABASE_VERSION = 1;
}

```

```
 1 package library.group5.constant;
 2
 3 /**
 4  * Created by Isha on 05-04-2018.
 5  */
 6
 7 public abstract class SQLCommand {
 8
 9     //Ouery for Landing Page
10     public static String SELECT_USER_EMAIL = "select
   User_Email from User";
11
12     //Query to insert new user
13     public static String INSERT_USER = "insert into User(
   User_Name,User_Email,User_Password) values(?,?,?)";
14
15     //Query for event page
16     public static String SELECT_EVENT = "select Ev_Name from
   Event";
17
18     public static String SELECT_EVENT_BY_NAME ="SELECT * FROM
   Event  WHERE Ev_Name = ?";
19
20     //Select, Add, Delete, Update  Event
21
22     //public static String AddEvent =" INSERT INTO Event (
   Ev_Name, Ev_Date, User_ID, Ev_Guests) " +
23     //"VALUES (?, ?, ?, ?)";
24
25     //public static String DeleteEvent ="DELETE FROM Event
   WHERE Ev_ID =?";
26
27     public static String UPDATE_EVENT ="UPDATE Event " +
28             "SET Ev_Guests = ?, Ev_Date = ? WHERE Ev_ID = ?";
29
30     public static String DELETE_EVENT ="DELETE FROM Event
   WHERE Ev_ID =?";
31
32     //Select, Add, Delete, Update  Menu Recipes
33
34     public static String SELECT_EVENT_MENU ="SELECT Rec_Name,
   Ev_Rec_Servings_Req FROM Menu,Recipe " +
35             "WHERE Menu.Rec_ID = Recipe.Rec_ID and  Ev_ID
    = ? ";
36
37     public static String SELECT_RECIPE = "select Rec_Name from
```

```
37  Recipe";
38
39      public static String SELECT_RECIPE_BY_NAME ="SELECT * FROM
     Recipe WHERE Rec_Name = ?";
40
41      public static String ADD_RECIPE_MENU ="INSERT INTO Menu (
    Rec_ID, Ev_ID, Ev_Rec_Servings_Req) " +
42              "VALUES (?, ?, ?)";
43
44      public static String DELETE_RECIPE_FROM_MENU ="DELETE " +
45              "FROM Menu " +
46              "WHERE Ev_ID =? AND Rec_ID = ?";
47
48      //public static String UpdateMenuRecipeServings ="UPDATE
    Menu " +
49             // "SET Ev_Rec_Servings_Req = ? " +
50             // "WHERE Ev_ID = ? AND Rec_ID = ?";
51
52      //Query for list Page
53
54
55      //Select, Add, Delete Lists
56
57      public static String SELECT_LIST = "select List_Name from
    List";
58
59
60      public static String SELECT_LIST_BY_NAME= "SELECT * " +
61              "FROM List  " +
62              "WHERE List_Name = ?";
63
64      //public static String AddList ="INSERT INTO List (
    List_Name, User_ID) " +
65             // "VALUES (?, ?)";
66
67      public static String DELETE_LIST ="DELETE " +
68             "FROM List " +
69              "WHERE List_ID=?";
70
71      //Select, Add, Delete Events to/from Lists
72
73      public static String SELECT_EVENT_LIST ="SELECT Ev_Name,
    Ev_Guests,Ev_Date " +
74              "FROM Event " +
75              "WHERE Event.List_ID = ?";
76
```

```
 77      public static String ADD_EVENT_TO_LIST ="UPDATE Event " +
 78              "SET List_ID = ? " +
 79              "WHERE EV_ID = ?";
 80
 81      public static String DELETE_EVENT_FROM_LIST ="UPDATE
    Event " +
 82              "SET List_ID = NULL " +
 83              "WHERE EV_ID = ?";
 84
 85
 86      //Query for Recipe Page
 87
 88      //Select, Add, Delete, Update  Recipe
 89
 90     // public static String AddRecipe ="INSERT INTO Recipe
    VALUES(?, ?, ?, ?)";
 91
 92      public static String DELETE_RECIPE ="DELETE FROM Recipe
    WHERE Rec_ID= ?";
 93
 94      public static String UPDATE_RECIPE_SERVINGS ="UPDATE
    Recipe " +
 95              "SET Rec_Servings_Made = ? " +
 96               "WHERE Rec_ID = ?";
 97
 98      //Select, Add, Delete, Update Recipe Ingredients
 99
100      public static String SELECT_INGREDIENT ="SELECT Ing_Name
    FROM Ingredient";
101
102      public static String SELECT_RECIPE_INGREDIENT ="SELECT *
    FROM Rec_Ingred WHERE Rec_ID = ?";
103
104      public static String SELECT_RECIPE_INGREDIENT_DETAIL ="
    SELECT Ing_Name,Rec_Ing_Qty,Unit_Name FROM Rec_Ingred,
    Ingredient,Unit "+
105              "WHERE Rec_Ingred.Ing_Id = Ingredient.Ing_Id "+
106              "and Rec_Ingred.Unit_Id = Unit.Unit_Id "+
107              "and Rec_ID = ?";
108
109
110      public static String ADD_ING_TO_REC ="INSERT INTO
    Rec_Ingred(Rec_ID, Ing_ID, Rec_Ing_Qty, Unit_id) " +
111              "VALUES (?, ?, ?, ?)";
112
113      public static String UPDATE_REC_ING_QTY_UNIT ="UPDATE
```

```
113 Rec_Ingred " +
114             "SET Rec_Ing_Qty = ?, Unit_ID = ? " +
115             "WHERE Rec_ID = ? AND Ing_ID = ?";
116
117     public static String DEL_ING_FROM_RECIPE ="DELETE " +
118             "FROM Rec_Ingred " +
119             "WHERE Rec_ID= ? AND Ing_ID = ?";
120
121     //Select Unit in the Recipe Page
122
123     public static String SELECT_UNIT = "select Unit_Name from
     Unit";
124
125     public static String SELECT_UNIT_BY_NAME ="SELECT * FROM
    Unit WHERE Unit_Name = ?";
126
127     public static String SELECT_INGREDIENT_BY_NAME ="SELECT
    Ing_Id FROM Ingredient WHERE Ing_Name = ?";
128
129     public static String CREATE_SHOPPING_LIST ="SELECT  Ing.
    Ing_Name, " +
130             "SUM(ROUND(" +
131             "CASE WHEN UP.Unit_Type_ID = UR.Unit_Type_ID " +
132             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*RI.REC_ING_QTY*UR.
    Unit_Convfactor/UP.Unit_Convfactor) " +
133             "WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
     " +
134             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*(RI.REC_ING_QTY*UR
    .Unit_Convfactor*Ing.Ing_Density)/Ing.Ing_Unit_Wt) " +
135             "WHEN UP.Unit_Type_ID = 1 AND UR.Unit_Type_ID = 3
     " +
136             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*(RI.REC_ING_QTY*UR
    .Unit_Convfactor)/Ing.Ing_Unit_Wt) " +
137             "WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 1
     " +
138             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*(RI.REC_ING_QTY*
    Ing.Ing_Unit_Wt/Ing.Ing_Density)/UP.Unit_Convfactor) " +
139             "WHEN UP.Unit_Type_ID = 2 AND UR.Unit_Type_ID = 2
     " +
140             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*(RI.REC_ING_QTY*UR
    .Unit_Convfactor/Ing.Ing_Density)/UP.Unit_Convfactor) " +
```

```
141             "WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 1
    " +
142             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*RI.REC_ING_QTY*Ing
    .Ing_Unit_Wt/UP.Unit_Convfactor) " +
143             "WHEN UP.Unit_Type_ID = 3 AND UR.Unit_Type_ID = 2
    " +
144             "THEN (ROUND(.4999+CAST(M.Ev_Rec_Servings_Req AS
    REAL)/CAST(R.Rec_Servings_Made AS REAL),0)*(RI.REC_ING_QTY*UR
    .Unit_Convfactor*Ing.Ing_Density)/UP.Unit_Convfactor) " +
145             "ELSE \"Bad Code\" " +
146             "END,2)) AS QtyReqd, UP.Unit_Abbrev AS
    PurchaseUnits " +
147             "FROM Ingredient AS Ing " +
148             "JOIN Unit       AS UP ON UP.Unit_ID = Ing.
    Ing_Purchase_Unit_ID " +
149             "JOIN Rec_Ingred AS RI ON Ing.Ing_ID = RI.Ing_ID
    " +
150             "JOIN Unit       AS UR ON RI.Unit_ID = UR.Unit_ID
    " +
151             "JOIN Recipe     AS R  ON RI.Rec_ID =R.Rec_ID " +
152             "JOIN Menu       AS M  ON R.Rec_ID = M.Rec_ID " +
153             "JOIN Event      AS E  ON M.Ev_ID = E.Ev_ID " +
154             "JOIN List       AS L  ON E.List_ID = L.List_ID "
    +
155             "WHERE L.List_ID = ? " +
156             "GROUP BY Ing.Ing_ID " +
157             "ORDER BY Ing.Cat_ID ASC";
158
159
160 }
161
```

XML Layout Code for Android Application (/MenuPlanner_Group5/app/src/main/res/layout)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:background="@color/colorPrimary"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center"
                android:src="@drawable/party" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginLeft="10dp"
                android:text="EVENTS"
                android:textColor="#fff"
                android:textSize="30sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:orientation="vertical">

            <LinearLayout
```

```
47                  android:layout_width="match_parent"
48                  android:layout_height="wrap_content"
49                  android:orientation="horizontal"
50                  android:paddingLeft="5dp"
51                  android:weightSum="1">
52
53                  <TextView
54                      android:layout_width="200dp"
55                      android:layout_height="50dp"
56                      android:text="@string/Sel_Event"
57                      android:textColor="#2F4F4F"
58                      android:textSize="24sp" />
59
60                  <TextView
61                      android:layout_width="200dp"
62                      android:layout_height="50dp"
63                      android:layout_marginLeft="10dp"
64                      android:text="@string/No_of_Guest"
65                      android:textColor="#2F4F4F"
66                      android:textSize="24sp" />
67
68                  <TextView
69                      android:layout_width="wrap_content"
70                      android:layout_height="wrap_content"
71                      android:layout_marginLeft="10dp"
72                      android:text="@string/Choose_Date"
73                      android:textColor="#2F4F4F"
74                      android:textSize="24sp" />
75
76
77              </LinearLayout>
78
79              <LinearLayout
80                  android:layout_width="match_parent"
81                  android:layout_height="wrap_content"
82                  android:orientation="horizontal"
83                  android:paddingLeft="5dp"
84                  android:weightSum="1">
85
86                  <AutoCompleteTextView
87                      android:id="@+id/EventAutoComp"
88                      android:layout_width="200dp"
89                      android:layout_height="50dp" />
90
91
92                  <EditText
```

```
 93                      android:id="@+id/No_of_Guests"
 94                      android:layout_width="200dp"
 95                      android:layout_height="50dp"
 96                      android:layout_marginLeft="10dp" />
 97
 98                  <EditText
 99                      android:id="@+id/eventDate"
100                      android:layout_width="180dp"
101                      android:layout_height="50dp" />
102
103              </LinearLayout>
104
105              <LinearLayout
106                  android:layout_width="match_parent"
107                  android:layout_height="wrap_content"
108                  android:layout_marginBottom="20dp"
109                  android:layout_marginTop="20dp"
110                  android:orientation="horizontal"
111                  android:paddingLeft="10dp"
112                  android:weightSum="1">
113
114                  <Button
115                  android:id="@+id/show_btn"
116                  android:layout_width="140dp"
117                  android:layout_height="36dp"
118                  android:layout_marginLeft="20dp"
119                  android:text="@string/Show"
120                  android:textColor="@android:color/black"
121                  android:textSize="14sp"
122                  android:textStyle="bold" />
123
124              <Button
125                  android:id="@+id/update_btn"
126                  android:layout_width="140dp"
127                  android:layout_height="36dp"
128                  android:layout_marginLeft="20dp"
129                  android:text="@string/Update"
130                  android:textColor="@android:color/black"
131                  android:textSize="14sp"
132                  android:textStyle="bold" />
133
134              <Button
135                  android:id="@+id/delete_btn"
136                  android:layout_width="140dp"
137                  android:layout_height="36dp"
138                  android:layout_marginLeft="20dp"
```

```
139                 android:text="@string/Delete"
140                 android:textColor="@android:color/black"
141                 android:textSize="14sp"
142                 android:textStyle="bold" />
143
144         </LinearLayout>
145
146             <LinearLayout
147                 android:layout_width="match_parent"
148                 android:layout_height="wrap_content"
149                 android:layout_marginTop="10dp"
150                 android:orientation="horizontal"
151                 android:paddingLeft="10dp"
152                 android:weightSum="1">
153
154                 <TextView
155                     android:layout_width="wrap_content"
156                     android:layout_height="wrap_content"
157                     android:text="MENU FOR EVENT"
158                     android:textColor="#2F4F4F"
159                     android:textSize="24sp" />
160
161             </LinearLayout>
162
163             <LinearLayout
164                 android:layout_width="match_parent"
165                 android:layout_height="wrap_content"
166                 android:layout_marginTop="10dp"
167                 android:orientation="horizontal"
168                 android:paddingLeft="10dp"
169                 android:weightSum="1">
170
171                 <TextView
172                     android:layout_width="wrap_content"
173                     android:layout_height="wrap_content"
174                     android:text="Recipe"
175                     android:textColor="#2F4F4F"
176                     android:textSize="24sp" />
177
178                 <TextView
179                     android:layout_width="wrap_content"
180                     android:layout_height="wrap_content"
181                     android:layout_marginLeft="10dp"
182                     android:text="Servings Required"
183                     android:textColor="#2F4F4F"
184                     android:textSize="24sp" />
```

```
185             </LinearLayout>
186
187             <LinearLayout
188                 android:layout_width="match_parent"
189                 android:layout_height="wrap_content"
190                 android:layout_marginTop="5dp"
191                 android:orientation="horizontal"
192                 android:paddingLeft="10dp"
193                 android:weightSum="1">
194
195                 <ScrollView
196                     android:id="@+id/menu_view"
197                     android:layout_width="match_parent"
198                     android:layout_height="200dp"
199                     android:textColor="#2F4F4F" />
200
201             </LinearLayout>
202
203             <LinearLayout
204                 android:layout_width="match_parent"
205                 android:layout_height="wrap_content"
206                 android:layout_gravity="center"
207                 android:orientation="vertical">
208
209                 <LinearLayout
210                     android:layout_width="match_parent"
211                     android:layout_height="wrap_content"
212                     android:orientation="horizontal"
213                     android:weightSum="1">
214
215                     <TextView
216                         android:layout_width="200dp"
217                         android:layout_height="50dp"
218                         android:text="Lookup Recipe"
219                         android:textColor="#2F4F4F"
220                         android:textSize="24sp" />
221
222                     <TextView
223                         android:id="@+id/textView4"
224                         android:layout_width="200dp"
225                         android:layout_height="50dp"
226                         android:paddingLeft="5dp"
227                         android:text="Servings Req'd"
228                         android:textColor="#2F4F4F"
229                         android:textSize="24sp" />
230
```

```
231                 </LinearLayout>
232
233                 <LinearLayout
234                     android:layout_width="match_parent"
235                     android:layout_height="wrap_content"
236                     android:orientation="horizontal"
237                     android:weightSum="1">
238
239                     <AutoCompleteTextView
240                         android:id="@+id/recAutoComp"
241                         android:layout_width="200dp"
242                         android:layout_height="50dp" />
243
244                     <EditText
245                         android:id="@+id/Serv_Req"
246                         android:layout_width="200dp"
247                         android:layout_height="50dp"
248                         android:layout_marginLeft="10dp" />
249
250
251                 </LinearLayout>
252
253                 <LinearLayout
254                     android:layout_width="match_parent"
255                     android:layout_height="wrap_content"
256                     android:layout_marginBottom="10dp"
257                     android:layout_marginTop="20dp"
258                     android:orientation="horizontal"
259                     android:paddingLeft="10dp"
260                     android:weightSum="1">
261
262
263                     <Button
264                         android:id="@+id/show_serv"
265                         android:layout_width="170dp"
266                         android:layout_height="36dp"
267                         android:text="@string/Show_serv"
268                         android:textColor="@android:color/
    black"
269                         android:textSize="14sp"
270                         android:textStyle="bold" />
271
272                     <Button
273                         android:id="@+id/add_rec"
274                         android:layout_width="200dp"
275                         android:layout_height="36dp"
```

```
276                         android:layout_marginLeft="10dp"
277                         android:text="@string/Add_Rec"
278                         android:textColor="@android:color/
    black"
279                         android:textSize="14sp"
280                         android:textStyle="bold" />
281
282                     <Button
283                         android:id="@+id/del_rec"
284                         android:layout_width="200dp"
285                         android:layout_height="36dp"
286                         android:layout_marginLeft="10dp"
287                         android:text="@string/Del_Rec"
288                         android:textColor="@android:color/
    black"
289                         android:textSize="14sp"
290                         android:textStyle="bold" />
291
292                 </LinearLayout>
293
294                 <LinearLayout
295                     android:layout_width="match_parent"
296                     android:layout_height="wrap_content"
297                     android:layout_marginBottom="20dp"
298                     android:layout_marginLeft="50dp"
299                     android:layout_marginTop="20dp"
300                     android:orientation="horizontal"
301                     android:paddingLeft="10dp"
302                     android:weightSum="1">
303
304                     <Button
305                         android:id="@+id/home_btn"
306                         android:layout_width="140dp"
307                         android:layout_height="36dp"
308                         android:layout_marginLeft="20dp"
309                         android:background="@color/
    colorPrimary"
310                         android:text="@string/Home_Page"
311                         android:textColor="@android:color/
    black"
312                         android:textSize="14sp"
313                         android:textStyle="bold" />
314
315                     <Button
316                         android:id="@+id/rec_btn"
317                         android:layout_width="140dp"
```

```
318                         android:layout_height="36dp"
319                         android:layout_marginLeft="20dp"
320                         android:background="@color/
    colorPrimary"
321                         android:text="@string/Recipe_Page"
322                         android:textColor="@android:color/
    black"
323                         android:textSize="14sp"
324                         android:textStyle="bold" />
325
326                     <Button
327                         android:id="@+id/list_btn"
328                         android:layout_width="140dp"
329                         android:layout_height="36dp"
330                         android:layout_marginLeft="20dp"
331                         android:background="@color/
    colorPrimary"
332                         android:text="@string/List_Page"
333                         android:textColor="@android:color/
    black"
334                         android:textSize="14sp"
335                         android:textStyle="bold" />
336
337                 </LinearLayout>
338
339
340             </LinearLayout>
341
342
343         </LinearLayout>
344     </LinearLayout>
345 </LinearLayout>
346
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:background="@color/colorPrimary"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center"
                android:src="@drawable/recipe" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginLeft="10dp"
                android:text="RECIPE"
                android:textColor="#fff"
                android:textSize="30sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:orientation="vertical">

            <LinearLayout
```

```
47                  android:layout_width="match_parent"
48                  android:layout_height="wrap_content"
49                  android:orientation="horizontal"
50                  android:paddingLeft="5dp"
51                  android:weightSum="1">
52
53                  <TextView
54                      android:layout_width="200dp"
55                      android:layout_height="50dp"
56                      android:text="Select Recipe"
57                      android:textColor="#2F4F4F"
58                      android:textSize="24sp" />
59
60                  <TextView
61                      android:layout_width="200dp"
62                      android:layout_height="50dp"
63                      android:layout_marginLeft="20dp"
64                      android:text="No. of Serving"
65                      android:textColor="#2F4F4F"
66                      android:textSize="24sp" />
67
68
69              </LinearLayout>
70
71              <LinearLayout
72                  android:layout_width="match_parent"
73                  android:layout_height="wrap_content"
74                  android:orientation="horizontal"
75                  android:paddingLeft="5dp"
76                  android:weightSum="1">
77
78                  <AutoCompleteTextView
79                      android:id="@+id/recAutoComp"
80                      android:layout_width="200dp"
81                      android:layout_height="50dp" />
82
83                  <EditText
84                      android:id="@+id/no_of_serv"
85                      android:layout_width="200dp"
86                      android:layout_height="50dp"
87                      android:layout_marginLeft="10dp" />
88
89              </LinearLayout>
90
91              <LinearLayout
92                  android:layout_width="match_parent"
```

```
 93                  android:layout_height="wrap_content"
 94                  android:layout_marginBottom="20dp"
 95                  android:layout_marginTop="20dp"
 96                  android:orientation="horizontal"
 97                  android:paddingLeft="10dp"
 98                  android:weightSum="1">
 99
100                  <Button
101                      android:id="@+id/showIng_btn"
102                      android:layout_width="140dp"
103                      android:layout_height="36dp"
104                      android:layout_marginLeft="20dp"
105                      android:text="@string/Show"
106                      android:textColor="@android:color/black"
107                      android:textSize="14sp"
108                      android:textStyle="bold" />
109
110                  <Button
111                      android:id="@+id/updateRec_btn"
112                      android:layout_width="140dp"
113                      android:layout_height="36dp"
114                      android:layout_marginLeft="20dp"
115                      android:text="@string/Update"
116                      android:textColor="@android:color/black"
117                      android:textSize="14sp"
118                      android:textStyle="bold" />
119
120                  <Button
121                      android:id="@+id/deleteRec_btn"
122                      android:layout_width="140dp"
123                      android:layout_height="36dp"
124                      android:layout_marginLeft="20dp"
125                      android:text="@string/Delete"
126                      android:textColor="@android:color/black"
127                      android:textSize="14sp"
128                      android:textStyle="bold" />
129
130              </LinearLayout>
131
132              <LinearLayout
133                  android:layout_width="match_parent"
134                  android:layout_height="wrap_content"
135                  android:layout_marginTop="10dp"
136                  android:orientation="horizontal"
137                  android:paddingLeft="10dp"
138                  android:weightSum="1">
```

```
139
140                 <TextView
141                     android:layout_width="wrap_content"
142                     android:layout_height="wrap_content"
143                     android:text="@string/Ing_recipe"
144                     android:textColor="#2F4F4F"
145                     android:textSize="24sp" />
146
147             </LinearLayout>
148
149             <LinearLayout
150                 android:layout_width="match_parent"
151                 android:layout_height="wrap_content"
152                 android:orientation="horizontal"
153                 android:paddingLeft="10dp"
154                 android:weightSum="1">
155
156                 <TextView
157                     android:id="@+id/textView2"
158                     android:layout_width="wrap_content"
159                     android:layout_height="wrap_content"
160                     android:text="INGREDIENT"
161                     android:textColor="#2F4F4F"
162                     android:textSize="24sp" />
163
164                 <TextView
165                     android:id="@+id/textView5"
166                     android:layout_width="wrap_content"
167                     android:layout_height="wrap_content"
168                     android:layout_marginLeft="50dp"
169                     android:text="QTY"
170                     android:textColor="#2F4F4F"
171                     android:textSize="24sp" />
172
173                 <TextView
174                     android:id="@+id/textView7"
175
176                     android:layout_width="wrap_content"
177                     android:layout_height="wrap_content"
178                     android:layout_marginLeft="80dp"
179                     android:text="UNITS"
180                     android:textColor="#2F4F4F"
181                     android:textSize="24sp" />
182
183             </LinearLayout>
184
```

```
185             <LinearLayout
186                 android:layout_width="match_parent"
187                 android:layout_height="wrap_content"
188                 android:layout_marginTop="5dp"
189                 android:orientation="horizontal"
190                 android:paddingLeft="10dp"
191                 android:weightSum="1">
192
193                 <ScrollView
194                     android:id="@+id/rec_view"
195                     android:layout_width="match_parent"
196                     android:layout_height="200dp"
197                     android:textColor="#2F4F4F" />
198
199             </LinearLayout>
200
201             <LinearLayout
202                 android:layout_width="match_parent"
203                 android:layout_height="wrap_content"
204                 android:layout_gravity="center"
205                 android:orientation="vertical">
206
207                 <LinearLayout
208                     android:layout_width="match_parent"
209                     android:layout_height="wrap_content"
210                     android:orientation="horizontal"
211                     android:paddingLeft="5dp"
212                     android:weightSum="1">
213
214                     <TextView
215                         android:layout_width="200dp"
216                         android:layout_height="50dp"
217                         android:text="Selet Ingredient"
218                         android:textColor="#2F4F4F"
219                         android:textSize="24sp" />
220
221                     <TextView
222                         android:layout_width="100dp"
223                         android:layout_height="50dp"
224                         android:layout_marginLeft="20dp"
225                         android:text="QTY"
226                         android:textColor="#2F4F4F"
227                         android:textSize="24sp" />
228
229                     <TextView
230                         android:layout_width="100dp"
```

```
231                         android:layout_height="50dp"
232                         android:layout_marginLeft="20dp"
233                         android:text="UNITS"
234                         android:textColor="#2F4F4F"
235                         android:textSize="24sp" />
236
237                 </LinearLayout>
238
239                 <LinearLayout
240                     android:layout_width="match_parent"
241                     android:layout_height="wrap_content"
242                     android:orientation="horizontal"
243                     android:paddingLeft="5dp"
244                     android:weightSum="1">
245
246                     <AutoCompleteTextView
247                         android:id="@+id/ingAutoComp"
248                         android:layout_width="200dp"
249                         android:layout_height="50dp" />
250
251
252                     <EditText
253                         android:id="@+id/ingQty"
254                         android:layout_width="100dp"
255                         android:layout_height="50dp"
256                         android:layout_marginLeft="20dp" />
257
258                     <Spinner
259                         android:id="@+id/unitId_spinner"
260                         android:layout_width="100dp"
261                         android:layout_height="50dp"
262                         android:layout_marginLeft="20dp" />
263
264                 </LinearLayout>
265
266                 <LinearLayout
267                     android:layout_width="match_parent"
268                     android:layout_height="wrap_content"
269                     android:layout_marginBottom="10dp"
270                     android:layout_marginTop="20dp"
271                     android:orientation="horizontal"
272                     android:paddingLeft="10dp"
273                     android:weightSum="1">
274
275
276                     <Button
```

```
277                         android:id="@+id/updateIng_btn"
278                         android:layout_width="170dp"
279                         android:layout_height="36dp"
280                         android:text="@string/Update_Ing"
281                         android:textColor="@android:color/
    black"
282                         android:textSize="14sp"
283                         android:textStyle="bold" />
284
285                     <Button
286                         android:id="@+id/addIng_btn"
287                         android:layout_width="200dp"
288                         android:layout_height="36dp"
289                         android:layout_marginLeft="10dp"
290                         android:text="@string/Add_Ing"
291                         android:textColor="@android:color/
    black"
292                         android:textSize="14sp"
293                         android:textStyle="bold" />
294
295                     <Button
296                         android:id="@+id/delIng_btn"
297                         android:layout_width="200dp"
298                         android:layout_height="36dp"
299                         android:layout_marginLeft="10dp"
300                         android:text="@string/Del_Ing"
301                         android:textColor="@android:color/
    black"
302                         android:textSize="14sp"
303                         android:textStyle="bold" />
304
305                 </LinearLayout>
306
307
308             </LinearLayout>
309
310             <LinearLayout
311                 android:layout_width="match_parent"
312                 android:layout_height="wrap_content"
313                 android:layout_marginLeft="50dp"
314                 android:layout_marginTop="20dp"
315                 android:orientation="horizontal">
316
317                 <Button
318                     android:id="@+id/homePage_btn"
319                     android:layout_width="140dp"
```

```
                    android:layout_height="36dp"
                    android:layout_marginLeft="20dp"
                    android:background="@color/colorPrimary"
                    android:text="@string/Home_Page"
                    android:textColor="@android:color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />

                <Button
                    android:id="@+id/eventPage_btn"
                    android:layout_width="140dp"
                    android:layout_height="36dp"
                    android:layout_marginLeft="20dp"
                    android:background="@color/colorPrimary"
                    android:text="@string/Event_Page"
                    android:textColor="@android:color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />

                <Button
                    android:id="@+id/listPage_btn"
                    android:layout_width="140dp"
                    android:layout_height="36dp"
                    android:layout_marginLeft="20dp"
                    android:background="@color/colorPrimary"
                    android:text="@string/List_Page"
                    android:textColor="@android:color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="225dp"
            android:background="@color/colorPrimary"
            android:orientation="vertical">

            <ImageView
                android:layout_width="125dp"
                android:layout_height="125dp"
                android:layout_gravity="center"
                android:layout_marginTop="20dp"
                android:src="@drawable/profilephoto" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="Anna"
                android:textColor="#fff"
                android:textSize="35sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/Weclome"
                android:textColor="#fff"
                android:textSize="20sp" />

        </LinearLayout>

        <LinearLayout
```

```
47              android:layout_width="360dp"
48              android:layout_height="wrap_content"
49              android:layout_gravity="center"
50              android:layout_marginTop="45dp"
51              android:orientation="vertical">
52
53              <LinearLayout
54                  android:layout_width="match_parent"
55                  android:layout_height="wrap_content"
56                  android:orientation="horizontal"
57                  android:paddingLeft="10dp"
58                  android:weightSum="1">
59
60                  <ImageView
61                      android:layout_width="50dp"
62                      android:layout_height="50dp"
63                      android:src="@drawable/
   ic_receipt_black_24dp" />
64
65                  <Button
66                      android:id="@+id/recipe_btn"
67                      android:layout_width="235dp"
68                      android:layout_height="wrap_content"
69                      android:layout_gravity="center_vertical"
70                      android:background="#00CED1"
71                      android:text="@string/Recipe"
72                      android:textColor="#fff"
73                      android:textSize="24sp" />
74              </LinearLayout>
75
76              <LinearLayout
77                  android:layout_width="match_parent"
78                  android:layout_height="wrap_content"
79                  android:layout_marginTop="25dp"
80                  android:orientation="horizontal"
81                  android:paddingLeft="10dp"
82                  android:weightSum="1">
83
84                  <ImageView
85                      android:layout_width="50dp"
86                      android:layout_height="50dp"
87                      android:src="@drawable/
   ic_event_note_black_24dp" />
88
89                  <Button
90                      android:id="@+id/event_btn"
```

```
 91                      android:layout_width="210dp"
 92                      android:layout_height="wrap_content"
 93                      android:layout_gravity="center_vertical"
 94                      android:layout_weight="0.31"
 95                      android:background="#00CED1"
 96                      android:text="@string/Event"
 97                      android:textColor="#fff"
 98                      android:textSize="24sp" />
 99              </LinearLayout>
100
101              <LinearLayout
102                  android:layout_width="match_parent"
103                  android:layout_height="wrap_content"
104                  android:layout_marginTop="25dp"
105                  android:orientation="horizontal"
106                  android:paddingLeft="10dp"
107                  android:weightSum="1">
108
109                  <ImageView
110                      android:layout_width="50dp"
111                      android:layout_height="50dp"
112                      android:src="@drawable/
    ic_add_shopping_cart_black_24dp" />
113
114                  <Button
115                      android:id="@+id/list_btn"
116                      android:layout_width="50dp"
117                      android:layout_height="wrap_content"
118                      android:layout_gravity="center_vertical"
119                      android:layout_weight="0.75"
120                      android:background="#00CED1"
121                      android:text="@string/list"
122                      android:textColor="#fff"
123                      android:textSize="24sp" />
124              </LinearLayout>
125
126          </LinearLayout>
127      </LinearLayout>
128 </LinearLayout>
129
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:background="@color/colorPrimary"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center"
                android:src="@drawable/shopping" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginLeft="10dp"
                android:text="LISTS"
                android:textColor="#2F4F4F"
                android:textSize="30sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:orientation="vertical">

            <LinearLayout
```

```
47                  android:layout_width="match_parent"
48                  android:layout_height="wrap_content"
49                  android:orientation="horizontal"
50                  android:paddingLeft="5dp"
51                  android:weightSum="1">
52
53                  <TextView
54                      android:layout_width="200dp"
55                      android:layout_height="50dp"
56                      android:layout_weight="0.02"
57                      android:text="Select List"
58                      android:textColor="#2F4F4F"
59                      android:textSize="24sp" />
60                  <AutoCompleteTextView
61                      android:id="@+id/listAutoComp"
62                      android:layout_width="200dp"
63                      android:layout_height="50dp"
64                      android:layout_weight="0.03" />
65
66
67              </LinearLayout>
68
69              <LinearLayout
70                  android:layout_width="match_parent"
71                  android:layout_height="wrap_content"
72                  android:layout_marginTop="10dp"
73                  android:orientation="horizontal"
74                  android:paddingLeft="5dp"
75                  android:weightSum="1">
76
77                  <Button
78                      android:id="@+id/show_event"
79                      android:layout_width="140dp"
80                      android:layout_height="40dp"
81                      android:layout_marginLeft="20dp"
82                      android:text="Show Events"
83                      android:textColor="@android:color/black"
84                      android:textSize="14sp"
85                      android:textStyle="bold" />
86
87                  <Button
88                      android:id="@+id/del_list"
89                      android:layout_width="140dp"
90                      android:layout_height="40dp"
91                      android:layout_marginLeft="20dp"
92                      android:text="Delete List"
```

```
 93                      android:textColor="@android:color/black"
 94                      android:textSize="14sp"
 95                      android:textStyle="bold" />
 96
 97
 98              </LinearLayout>
 99
100              <LinearLayout
101                  android:layout_width="match_parent"
102                  android:layout_height="wrap_content"
103                  android:layout_marginTop="10dp"
104                  android:orientation="horizontal"
105                  android:paddingLeft="10dp"
106                  android:weightSum="1">
107
108                  <TextView
109                      android:layout_width="wrap_content"
110                      android:layout_height="wrap_content"
111                      android:text="EVENTS FOR THE LIST"
112                      android:textColor="#2F4F4F"
113                      android:textSize="24sp" />
114
115              </LinearLayout>
116
117              <LinearLayout
118                  android:layout_width="match_parent"
119                  android:layout_height="wrap_content"
120                  android:layout_marginTop="10dp"
121                  android:orientation="horizontal"
122                  android:paddingLeft="10dp"
123                  android:weightSum="1">
124
125                  <TextView
126                      android:layout_width="100dp"
127                      android:layout_height="wrap_content"
128                      android:text="Event"
129                      android:textColor="#2F4F4F"
130                      android:textSize="24sp" />
131
132                  <TextView
133                      android:layout_width="150dp"
134                      android:layout_height="wrap_content"
135                      android:layout_marginLeft="20dp"
136                      android:text="No. of guests"
137                      android:textColor="#2F4F4F"
138                      android:textSize="24sp" />
```

```
139
140                 <TextView
141                     android:layout_width="wrap_content"
142                     android:layout_height="wrap_content"
143                     android:layout_marginLeft="20dp"
144                     android:text="Event Date"
145                     android:textColor="#2F4F4F"
146                     android:textSize="24sp" />
147             </LinearLayout>
148
149             <LinearLayout
150                 android:layout_width="match_parent"
151                 android:layout_height="wrap_content"
152                 android:layout_marginTop="5dp"
153                 android:orientation="horizontal"
154                 android:paddingLeft="10dp"
155                 android:weightSum="1">
156
157                 <ScrollView
158                     android:id="@+id/event_view"
159                     android:layout_width="match_parent"
160                     android:layout_height="200dp"
161                     android:textColor="#2F4F4F" />
162
163             </LinearLayout>
164
165             <LinearLayout
166                 android:layout_width="match_parent"
167                 android:layout_height="wrap_content"
168                 android:layout_gravity="center"
169                 android:orientation="vertical">
170
171                 <LinearLayout
172                     android:layout_width="match_parent"
173                     android:layout_height="wrap_content"
174                     android:orientation="horizontal"
175                     android:paddingLeft="5dp"
176                     android:weightSum="1">
177
178                     <TextView
179                         android:layout_width="200dp"
180                         android:layout_height="50dp"
181                         android:text="Lookup Event"
182                         android:textColor="#2F4F4F"
183                         android:textSize="24sp" />
184
```

```
185                 </LinearLayout>
186
187                 <LinearLayout
188                     android:layout_width="match_parent"
189                     android:layout_height="wrap_content"
190                     android:orientation="horizontal"
191                     android:paddingLeft="5dp"
192                     android:weightSum="1">
193
194                     <AutoCompleteTextView
195                         android:id="@+id/eventAutoComp"
196                         android:layout_width="200dp"
197                         android:layout_height="50dp" />
198                 </LinearLayout>
199                 <LinearLayout
200                     android:layout_width="match_parent"
201                     android:layout_height="wrap_content"
202                     android:layout_marginBottom="20dp"
203                     android:layout_marginTop="20dp"
204                     android:orientation="horizontal"
205                     android:paddingLeft="10dp"
206                     android:weightSum="1">
207
208                     <Button
209                         android:id="@+id/add_event"
210                         android:layout_width="wrap_content"
211                         android:layout_height="40dp"
212                         android:layout_marginLeft="20dp"
213                         android:text="Add Event to List"
214                         android:textStyle="bold" />
215                     <Button
216                         android:id="@+id/del_event"
217                         android:layout_width="wrap_content"
218                         android:layout_height="40dp"
219                         android:layout_marginLeft="20dp"
220                         android:text="Delete Event from List"
221                         android:textStyle="bold" />
222                 </LinearLayout>
223
224             </LinearLayout>
225
226             <LinearLayout
227                 android:layout_width="match_parent"
228                 android:layout_height="wrap_content"
229                 android:orientation="horizontal"
230                 android:paddingLeft="10dp"
```

```
231                 android:weightSum="1">
232
233                 <Button
234                     android:id="@+id/shop_Btn"
235                     android:layout_width="300dp"
236                     android:layout_height="50dp"
237                     android:layout_marginLeft="100dp"
238                     android:layout_marginTop="20dp"
239                     android:background="@color/colorPrimary"
240                     android:gravity="center"
241                     android:text="Generate shoppoing list"
242                     android:textColor="@android:color/
    background_dark"
243                     android:textStyle="bold" />
244
245             </LinearLayout>
246
247         </LinearLayout>
248
249         <LinearLayout
250             android:layout_width="match_parent"
251             android:layout_height="wrap_content"
252             android:layout_marginLeft="50dp"
253             android:layout_marginTop="20dp"
254             android:orientation="horizontal">
255
256             <Button
257                 android:id="@+id/home_Btn"
258                 android:layout_width="140dp"
259                 android:layout_height="36dp"
260                 android:layout_marginLeft="20dp"
261                 android:background="@color/colorPrimary"
262                 android:text="@string/Home_Page"
263                 android:textColor="@android:color/black"
264                 android:textSize="14sp"
265                 android:textStyle="bold" />
266
267             <Button
268                 android:id="@+id/event_btn"
269                 android:layout_width="140dp"
270                 android:layout_height="36dp"
271                 android:layout_marginLeft="20dp"
272                 android:background="@color/colorPrimary"
273                 android:text="@string/Event_Page"
274                 android:textColor="@android:color/black"
275                 android:textSize="14sp"
```

```
276                 android:textStyle="bold" />
277
278             <Button
279                 android:id="@+id/rec_Btn"
280                 android:layout_width="140dp"
281                 android:layout_height="36dp"
282                 android:layout_marginLeft="20dp"
283                 android:background="@color/colorPrimary"
284                 android:text="@string/Recipe_Page"
285                 android:textColor="@android:color/black"
286                 android:textSize="14sp"
287                 android:textStyle="bold" />
288         </LinearLayout>
289     </LinearLayout>
290 </LinearLayout>
291
292
```

File - /Users/sajudson/Dropbox/WPI/MIS571/G5Project/Scott/MenuPlanner_Group5/app/src/main/res/layout/landingpage.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="225dp"
            android:orientation="vertical"
            android:background="@color/colorPrimary">

            <ImageView
                android:id="@+id/img_1"
                android:layout_width="125dp"
                android:layout_height="125dp"
                android:layout_gravity="center"
                android:layout_marginTop="20dp"
                android:src="@drawable/profilephoto" />

            <TextView
                android:id="@+id/title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/Title"
                android:textColor="#fff"
                android:textSize="25sp" />

            <TextView
                android:id="@+id/heading"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/Weclome"
                android:textColor="#fff"
                android:textSize="18sp" />

        </LinearLayout>
```

```
47          <LinearLayout
48              android:orientation="vertical"
49              android:layout_width="match_parent"
50              android:layout_height="match_parent">
51
52              <EditText
53                  android:id="@+id/email_id"
54                  android:layout_width="300dp"
55                  android:layout_height="wrap_content"
56                  android:layout_gravity="center"
57
58                  android:layout_marginTop="80dp"
59                  android:background="@color/colorPrimary"
60                  android:drawableLeft="@drawable/
   ic_email_black_24dp"
61                  android:drawableStart="@drawable/
   ic_email_black_24dp"
62                  android:hint="@string/Email"
63                  android:inputType="textEmailAddress"
64                  android:padding="10dp"
65                  android:textColor="#fff"
66                  android:textColorHint="#fff" />
67
68              <Button
69                  android:id="@+id/submit_btn"
70                  android:layout_width="300dp"
71                  android:layout_height="40dp"
72                  android:layout_gravity="center"
73                  android:layout_marginTop="20dp"
74                  android:background="@color/colorPrimary"
75                  android:text="@string/Submit"
76                  android:textColor="#fff"
77                  android:textStyle="bold" />
78          </LinearLayout>
79
80
81
82      </LinearLayout>
83  </LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="225dp"
            android:orientation="vertical"
            android:background="@color/colorPrimary">
            <ImageView
                android:layout_marginTop="20dp"
                android:src="@drawable/profilephoto"
                android:layout_gravity="center"
                android:layout_width="125dp"
                android:layout_height="125dp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/Title"
                android:textColor="#fff"
                android:textSize="25sp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/Weclome"
                android:textColor="#fff"
                android:textSize="18sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
```

```
47              android:orientation="vertical">
48
49              <EditText
50                  android:id="@+id/user_name"
51                  android:layout_width="300dp"
52                  android:layout_height="wrap_content"
53                  android:layout_gravity="center"
54                  android:layout_marginTop="20dp"
55                  android:background="@color/colorPrimary"
56                  android:drawableLeft="@drawable/
    ic_person_black_24dp"
57                  android:drawableStart="@drawable/
    ic_person_black_24dp"
58                  android:hint="@string/User"
59                  android:padding="10dp"
60                  android:textColor="#fff"
61                  android:textColorHint="#fff" />
62
63              <EditText
64                  android:id="@+id/user_email"
65                  android:layout_width="300dp"
66                  android:layout_height="wrap_content"
67                  android:layout_gravity="center"
68
69                  android:layout_marginTop="20dp"
70                  android:background="@color/colorPrimary"
71                  android:drawableLeft="@drawable/
    ic_email_black_24dp"
72                  android:drawableStart="@drawable/
    ic_email_black_24dp"
73                  android:hint="@string/Email"
74                  android:inputType="textEmailAddress"
75                  android:padding="10dp"
76                  android:textColor="#fff"
77                  android:textColorHint="#fff" />
78
79              <EditText
80                  android:id="@+id/user_pwd"
81                  android:layout_width="300dp"
82                  android:layout_height="wrap_content"
83                  android:layout_gravity="center"
84
85                  android:layout_marginTop="20dp"
86                  android:background="@color/colorPrimary"
87                  android:hint="@string/Password"
88                  android:inputType="textPassword"
```

```
 89                  android:padding="10dp"
 90                  android:textColor="#fff"
 91                  android:textColorHint="#fff" />
 92
 93              <Button
 94                  android:id="@+id/register_btn"
 95                  android:layout_width="300dp"
 96                  android:layout_height="40dp"
 97                  android:layout_gravity="center"
 98                  android:layout_marginTop="20dp"
 99                  android:background="@color/colorPrimary"
100                  android:text="@string/Register"
101                  android:textColor="#fff"
102                  android:textStyle="bold" />
103          </LinearLayout>
104
105
106
107      </LinearLayout>
108  </LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:background="@color/colorPrimary"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center"
                android:src="@drawable/shopping" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginLeft="10dp"
                android:text="DETAILED SHOPPING LIST"
                android:textColor="#fff"
                android:textSize="22sp" />

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:orientation="vertical">

            <LinearLayout
```

```
47                  android:layout_width="match_parent"
48                  android:layout_height="wrap_content"
49                  android:orientation="horizontal"
50                  android:paddingLeft="5dp"
51                  android:weightSum="1">
52
53                  <TextView
54                      android:layout_width="200dp"
55                      android:layout_height="50dp"
56                      android:text="LIST NAME"
57                      android:textColor="#2F4F4F"
58                      android:textSize="24sp" />
59
60                  <AutoCompleteTextView
61                      android:id="@+id/list_auto_comp"
62                      android:layout_width="200dp"
63                      android:layout_height="50dp"
64                      android:layout_marginLeft="10dp"
65                      android:layout_weight="0.03" />
66
67
68              </LinearLayout>
69
70              <LinearLayout
71                  android:layout_width="match_parent"
72                  android:layout_height="wrap_content"
73                  android:layout_marginTop="10dp"
74                  android:orientation="horizontal"
75                  android:paddingLeft="5dp"
76                  android:weightSum="1">
77
78                  <Button
79                      android:id="@+id/show_item"
80                      android:layout_width="300dp"
81                      android:layout_height="40dp"
82                      android:layout_marginLeft="20dp"
83                      android:text="Show Items to purchase"
84                      android:textColor="@android:color/black"
85                      android:textSize="14sp"
86                      android:textStyle="bold" />
87
88
89              </LinearLayout>
90
91              <LinearLayout
92                  android:layout_width="match_parent"
```

```
 93                  android:layout_height="wrap_content"
 94                  android:layout_marginTop="25dp"
 95                  android:orientation="horizontal"
 96                  android:paddingLeft="5dp"
 97                  android:weightSum="1">
 98
 99                  <TextView
100                      android:id="@+id/textView2"
101                      android:layout_width="150dp"
102                      android:layout_height="50dp"
103                      android:text="Ingredient"
104                      android:textColor="#2F4F4F"
105                      android:textSize="24sp" />
106
107                  <TextView
108                      android:id="@+id/textView4"
109                      android:layout_width="100dp"
110                      android:layout_height="50dp"
111                      android:layout_marginLeft="30dp"
112                      android:text="Total Qty"
113                      android:textColor="#2F4F4F"
114                      android:textSize="24sp" />
115
116                  <TextView
117                      android:id="@+id/textView5"
118                      android:layout_width="200dp"
119                      android:layout_height="50dp"
120                      android:layout_marginLeft="30dp"
121                      android:text="Purchase Units"
122                      android:textColor="#2F4F4F"
123                      android:textSize="24sp" />
124
125              </LinearLayout>
126
127              <LinearLayout
128                  android:layout_width="match_parent"
129                  android:layout_height="wrap_content"
130                  android:layout_marginTop="5dp"
131                  android:orientation="horizontal"
132                  android:paddingLeft="10dp"
133                  android:weightSum="1">
134
135                  <ScrollView
136                      android:id="@+id/ing_view"
137                      android:layout_width="match_parent"
138                      android:layout_height="335dp"
```

```
                    android:textColor="#2F4F4F" />

            </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="150dp"
                android:layout_marginTop="20dp"
                android:orientation="horizontal"
                android:weightSum="1">

                <Button
                    android:id="@+id/shop_homebtn"
                    android:layout_width="140dp"
                    android:layout_height="36dp"
                    android:layout_marginLeft="20dp"
                    android:background="@color/colorPrimary"
                    android:text="@string/Home_Page"
                    android:textColor="@android:color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />

                <Button
                    android:id="@+id/log_out"
                    android:layout_width="140dp"
                    android:layout_height="36dp"
                    android:layout_marginLeft="20dp"
                    android:background="@color/colorPrimary"
                    android:text="@string/Logout"
                    android:textColor="@android:color/black"
                    android:textSize="14sp"
                    android:textStyle="bold" />

            </LinearLayout>


        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="
http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="library.group5.group5Activity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

```

XML Values Code for Android Application (/MenuPlanner_Group5/app/src/main/res/values)

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#00ced1</color>
    <color name="colorPrimaryDark"> #20B2AA</color>
    <color name="colorAccent">#FF4081</color>
    <color name="gray">#B0C4DE</color>
    <color name="green">#008B8B</color>
    <color name="DarkSlateGray">#2F4F4F</color>
</resources>

```

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>

```

```xml
<resources>
    <string name="app_name">MenuPlanner_Group5</string>
    <string name="Title">Menu Planner</string>
    <string name="Weclome">Welcome to Menu Planner</string>
    <string name="Submit">Submit</string>
    <string name="Register">Register</string>
    <string name="Recipe">Recipe</string>
    <string name="Event">Event</string>
    <string name="list">List</string>
    <string name="Password">Password</string>
    <string name="Email">Email</string>
    <string name="User">User Name</string>
    <string name="Sel_Event">Select Event</string>
    <string name="No_of_Guest">No. of Guests</string>
    <string name="Show">Show</string>
    <string name="Update">Update</string>
    <string name="Delete">Delete</string>
    <string name="Add_Rec">Add Recipe to Menu</string>
    <string name="Show_serv">Show Serving</string>
    <string name="Del_Rec">Del Recipe from Menu</string>
    <string name="Home_Page">Home Page</string>
    <string name="List_Page">List Page</string>
    <string name="Recipe_Page">Recipe Page</string>
    <string name="Event_Page">Event Page</string>
    <string name="Choose_Date">Choose Date</string>
    <string name="Ing_recipe">Ingredient for Recipe</string>
    <string name="Add_Ing">Add Ing. to Rec.</string>
    <string name="Update_Ing">Update Ing. of Rec.</string>
    <string name="Del_Ing">Del Ing. from Rec.</string>
    <string name="Logout">Log Out</string>



</resources>

```