

IST 402 Team Project

Summer 2016

Project Name:

TOFU

(Tons of FOOD for U)

Team Name:

IST 402 Team 4

Team Members:

Hunkyung Park

Seojung Nam

Sydnee Woodhouse

I. Mission Statement

- A. To share variety of recipes from all over the world to inspire that everybody can easily cook amazing dishes for their loved ones by simply following step-by-step recipes of TOFU application.

II. Group members with Roles

- A. **Hunkyung Park (Project Manager)** - Head of the project and is responsible for all the actions and works done by the members. Need to set the dates for group meet up, and make sure everyone is doing what they are supposed to do.
- B. **Seojung Nam (Assistant Leader)** - Aiding the project manager, Hunkyung Park, and managing the other members for them to have the best working environment. Moreover, responsible for checking if everything is done completely without any errors.
- C. **Sydnee Woodhouse (System Analyst)** - Analyzing system problems and responsible for complete life-cycle and new modified technology follows the trend.

III. Problem Definition

- A. As a team, we noticed how inconvenient it is to cook meals and dishes. Most people don't like to cook because it's time consuming and they want something quick like fast food. That is why we created this app. We wanted people to see that cooking can be fast, easy and delicious. Also we noticed that people who do cook often browse different websites for the same recipe. We wanted to create an app that allows users to have access to all the different styles of the same recipe at once instead of having to spend more time needed browsing the web.

IV. Solution Statement

- A. To achieve our goal, we will be creating the application that can share food and recipes to the users. The application will provide the users recipes of food that are new, old, interesting, healthy, and more. Furthermore, if we can get far enough, we will add forums where users can share their own recipes with each other.

V. Resources Required

- A. Before creating our actual application, we first need to take an investigation of real users to clear the users' need. Then we will be carefully fixing our plans with the iterations to reduce the scope creeps. We will be creating multiple sample designs to test the users and will be having iterations each time same as the planning phase of our projects. We will make sure that our application fulfill our tasks without any defects and delays.

VI. Project Plan outlining the tasks

A. **Sprint 1:** Project Concept, Overview, Project Plan *due by June, 5*

- Contacting and knowing each other by communication through Canvas mail, skype, Groupme. Selecting the methods to contact and work
- Selecting our project topic through the discussion
- Writing the mission statement based on our goal and the users' needs
- Selecting the roles based on each one's skills and concerns
- Identifying the problem definition by writing the purpose of the project
- Writing the expected solution that we seeks and fits the users that we aimed
- Collecting the resources that are required

B. **Sprint 2:** Statement of Work and Functional and Technical Requirements *due by June 26*

- Clearing the needs and goals that users seek by collecting the stories of real user
- Identifying the functional requirements with the diarams and charts if necessary
- Identifying the technical requirements with the diarams and charts if necessary
- Providing the Scope of Work with project charter, gannt chart, and more if necessary
- Completing the updated project plan with the detailed work breakdown structure, so not only us to work effectively, but also to provide the stakeholders where we are.

C. **Sprint 3:** UI Layout, Style Guide, Class Diagram, Database Design Documents, ERD, UML *due by July 17*

- Designing overall user interface layout and database.

- Creating an entity relationship diagram (ERD) that shows the relationships of entity sets stored in a database.
 - Developing Unified Modeling Language (UML) to visualize the design of an application.
- D. **Sprint 4:** Development: User Interface, Java Code Behind, Database, External Integration, Unit Tests *due by August 7*
- Creating Java and XML code based on the finalized UI layout and database design.
 - Testing the application with Android Studio

VII. User Stories with Results

- 1) To create user stories that can have users' ideas as much as possible, we have made a question simple and discrete, but broad and unconstrained. So the users are free to speak anything including comments and constraints about previous apps and suggestions and wants about future apps that will be developed.
- 2) The question
 - a) "Which words come up to you when you think of food apps?"
 - b) "What is your opinion about the layout of the food applications?"
 - c) "What is your overall impressions about the food applications?"
 - d) "What do you want from future food application?"
 - e) "When will you use the food application the most?"
- 3) Answers from the users
 - a) John Lee
 - i) "I would say the food recipe applications were pretty much complicated and repetitive. I want them to have more buttons so I can move easily through the pages."
 - b) Sungho Kim
 - i) "Hard to read, Don't have much time to read, Lacks variety, Hard to cook, Too much ingredients. So basically, I want the food applications to have more reality that we can think the developers have think about us while making the app."
 - c) Nina Song

- i) “Comparing with eating at the restaurant, it didn’t have much advantage. Cooking is too much of time consuming and food apps don’t give much of help. I want it to be more cost effective where the quality of the food meets the quality of products of the restaurant.”
- d) Kevin Kim
 - i) “Those food applications out there are really useful. I love to look up the recipes from ‘SmartyPans’ whenever I’m cooking for my wife. The overall layout and impression was quite simple and bright. It would be much better if I can see some selected recommended recipes for me.”
- e) Myungkook Lee
 - i) “Not very well-organized, didn’t have information that I wanted, it would be great if there’re choice for us to select professional or trainee. Also I think it would be very helpful if we have a navigation bar or remote control bar which exist in every page and we can use it anytime.”
- f) Jaeseok Yoo
 - i) “My cousins’ family is currently living at the outside of Korea and I wanted to cook their food, but most of the apps had limited selections. So for future applications, I want it to give me various and easy options that the users can pick and cook easily without having hard time.”
- g) Sarah Park
 - i) “Since most of the recipes sound too complicated, I had never actually tried out any. I always ended up cooking the same dishes all the time.”
- h) Todd Simms
 - i) “I don’t care too much about cooking apps. They all have the same repeated recipes! I want something new for once! I rarely user those applications.”
- i) Jason Wood
 - i) “A lot of apps didn’t have dessert options available. I want to surprise my wife and kids with a tasty dessert after their meals.

Also it would be much better if I can find healthy recipes easily.”

4) Results from the user stories

- a) Based on the users stories, we have spoken with Eun Sook Han, an application developer who has experience in making the food application. And through the discussion among the team members, we have decided to focus our goals to “Simple, various levels, various kind, global, neat, and uncomplicate.” To ensure to satisfy users’ want and need, we will keep in touch with the users and the expert as the developement of the project progresses.

VIII. Functional Requirements (what it will do)

- A. Provide step-by-step recipes with pictures
- B. Search recipes with keywords
- C. Display randomly selected tags that help the users browse random recipes.
- D. Have recipes organized by what meals they for. For example, lunch foods will be in the lunch tag, dessert foods will be in the dessert tag, etc.
- E. Provide calorie intake on recipes and highlight healthy recipes.
- F. Calculate total calorie intake by selecting recipes

IX. Technical Requirements (how you will do it)

- A. Hardware Requirements
 - a. Android Devices and related accessories (USB cable, charger, and etc)
 - b. Laptops, desktops, mouse, and earphones
- B. Inner Technical Requirements
 - a. Input recipes, pictures, and related tags.
 - b. Search box that enables users to look up recipes or tags.
 - c. Number of keywords and tags will be selected randomly and displayed when the user first open the application.
 - d. Put recipes in correct meal tag so that users are able to search for recipes based on the meal
 - e. For every recipe put the number of calories and if it is healthy, put a sticker icon saying it’s a healthy choice.
 - f. Put calorie calculator which adds up to the total calorie intake by users’ selection.

C. Software Requirements

- a. Supported Android OS: 4.4 (KitKat) or higher
- b. JAVA(SE 8 Update 92), SE 7 Update 51 (Minimum)
- c. Android Studio 2.1.1
- d. DB/Cloud Service: SQLite, AWS(Amazon Web Services)

X. SOW: Statement of Work - Scope of Work

A. Scope Management Plan

- a. The plan for our project is to create the application that provides various recipes of food around the world. The main goal of our project is to make everyone who are interested in cooking to easily cook their own meals. We will be collecting the requirements through the meeting with users and expert to reduce the gap between the actual application and user's want. On a daily basis, the scope management will be managed by Team 4. If the change is necessary, a change request form will be provided to users or stakeholders.

B. Scope Statement

Project Title: IST 402 Team 4 TOFU

Date: 06/24/2016

Prepared by: Hunkyung Park, Project manager

Project Justification: Our project has been started by our team's hope to give everyone in the world more efficient and various ways of cooking food. So that we can eat healthier, cheap, and adequate food for every meal which will eventually lead to healthy society. We will make food recipe application that can be easily used by anyone.

Product Characteristics and Requirements:

1. Food cooking knowledge requirements
2. Interest in food
3. Worldwide used application

Summary of Project Deliverables

Project management-related deliverables: Sprint 1 (Mission Statement, Group members with roles, Problem Definition, Solution Statement, Resources Required, Project Plan Outlining the task), Sprint 2 (User Stories, Functional and Technical Requirements, SOW), Sprint 3 (Wireframe and Mockups for the project, Style guide, Diagrams, Database Shema, Digital assets, ERD diagram, UML diagrams), updated project plan through all 4 phases.

Product Related deliverables: Sprint 4 (XML markup, java code, Screen shots, Testing, Unit test results)

Project Success Criteria: Our goal is to provide everyone a simple application that can

effectively give recipes about food in various areas. We will aim our success to the benefit of our users the most by thoroughly understanding the process of making application and ways of people interacting with the application.

XI. Updated Project Plan (of the work effort details)

1. **Sprint 3:** UI Layout, Style Guide, Class Diagram, Database Design Documents, ERD, UML *due by July 17*
 - a. Designing overall user interface layout and database.
 - b. Providing a style guide to reference upon.
 - c. Creating an entity relationship diagram (ERD) that shows the relationships of entity sets stored in a database.
 - d. Developing Unified Modeling Language (UML) to visualize the design of an application.
2. **Sprint 4:** Development: User Interface, Java Code Behind, Database, External Integration, Unit Tests *due by August 7*
 - a. Creating Java and XML code based on the finalized UI layout and database design. Testing the application with Android Studio

XII. Wireframe and Mockups for the Project - **Seojung**

- a. Home Page



Welcome to TOFU!

TOFU

Tons of FOOD for U!



Search

Explore and Try Our Random Recipes !

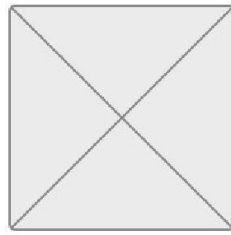


- There will be a search box that enables user to look up recipes with keywords at the center of the home page.
- There will be randomly selected tags or pictures that help the users browse other new recipes at the bottom of the home page.
- There will be a navigation to menu option which will contain: My Favorites, Calorie Calculator, About Us.

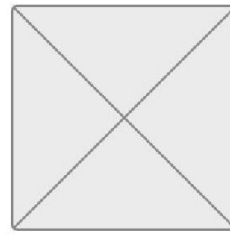
b. Search Result Page



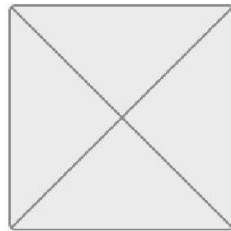
Search Other Recipes!



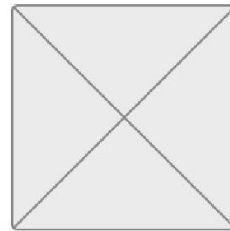
Recipe Name



Recipe Name



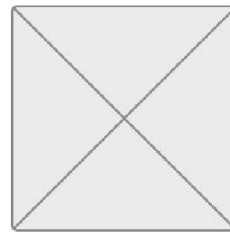
Recipe Name



Recipe Name



Recipe Name



Recipe Name

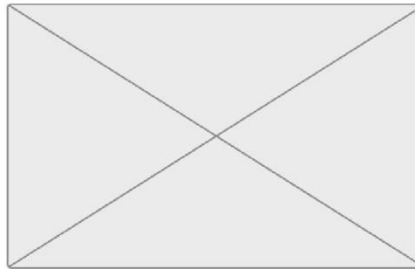
- There will be six search results displayed on one page. Each item contains a picture and a name of the recipe.
- There will be a search box on the top of the page also so that users can explore other recipes at this page.
- Users can navigate back to home page by using back button on the left top corner.
- Users can view full ingredients and recipe by clicking each item.

c. Recipe Page



Search Other Recipes!

Recipe Name



Calorie: __cal



INGREDIENTS

1. 2 lbs ground beef, preferably lean.
2. 1 medium white onion, finely chopped.
3. 2 tablespoons garlic powder.
4. 4 tablespoons Worcestershire sauce.
5. 2 teaspoons salt.
6. 1 teaspoon ground pepper.
7. 3/4 cup quick-cooking oats (more if the mixture seems soggy or won't hold together)
8. 2 eggs.

DIRECTIONS

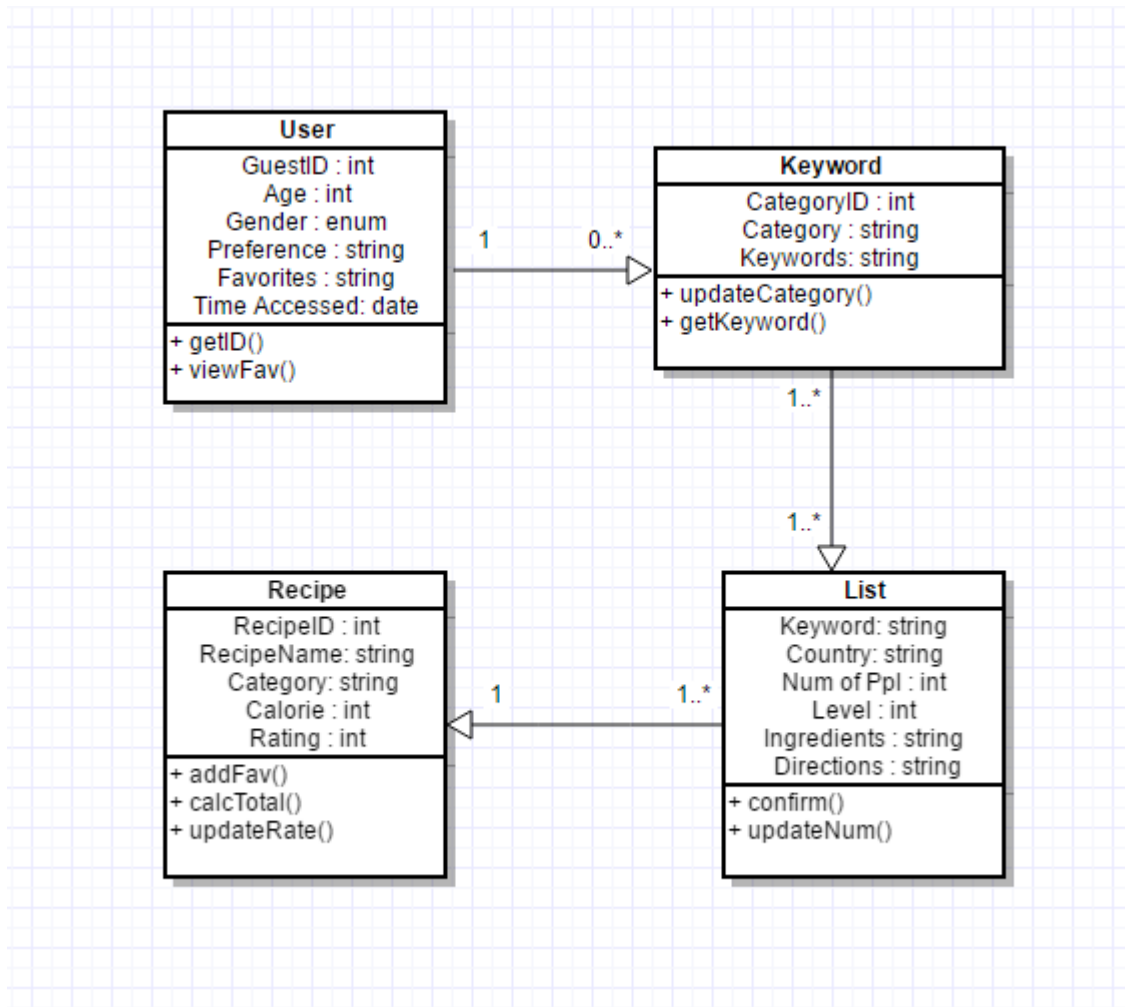
1. Whisk egg in a bowl& add next 6 ingredients.
2. Add any of the "stir-ins" that appeal to you.
3. Crumble in beef& using your hands or a fork, gently mix together.
4. Handle the meat as little as possible – the more you work it, the tougher it gets.
5. Gently shape (don't firmly press) mixture into burgers about 3/4inch thick.
6. Using your thumb, make a shallow depression in the centre of each burger to prevent puffing up during cooking.
7. Place burgers on the grill, close lid& BBQ until NO LONGER PINK INSIDE, turning once, about 6 – 8 minutes per side.
8. An instant read thermometer should read 160F.
9. Don't abuse your burgers by pressing with a spatula, pricking with a fork or turning frequently as precious juices will be lost!
10. Tuck into a warm crusty bun& add your favourite toppings.

- This page will display a name, picture, calorie intake, ingredients, and directions of the recipe.
- There will be a search box on the top of the page also so that users can explore other recipes at this page.
- Users can add calorie intake of selected recipe to their own calorie calculator by using plus(+) button next to the calorie.
- User can add the selected recipe to their favorites menu.
- Users can navigate back to search result page by using back button on the left top corner.

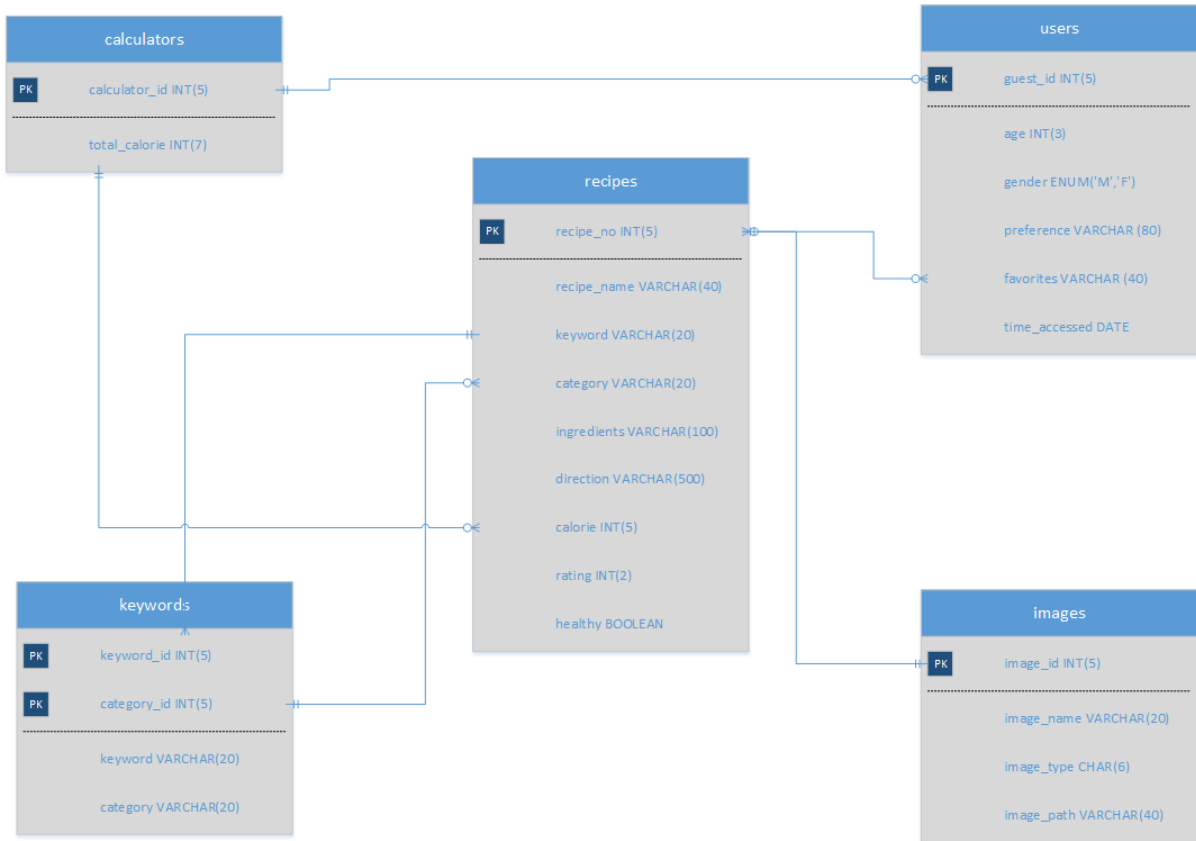
XIII. Style Guide for the Project - Sydnee

a. <https://app.frontify.com/d/mMiFbjNOyc8r/tofu-style-guide>

XIV. Class Diagram - Seojung



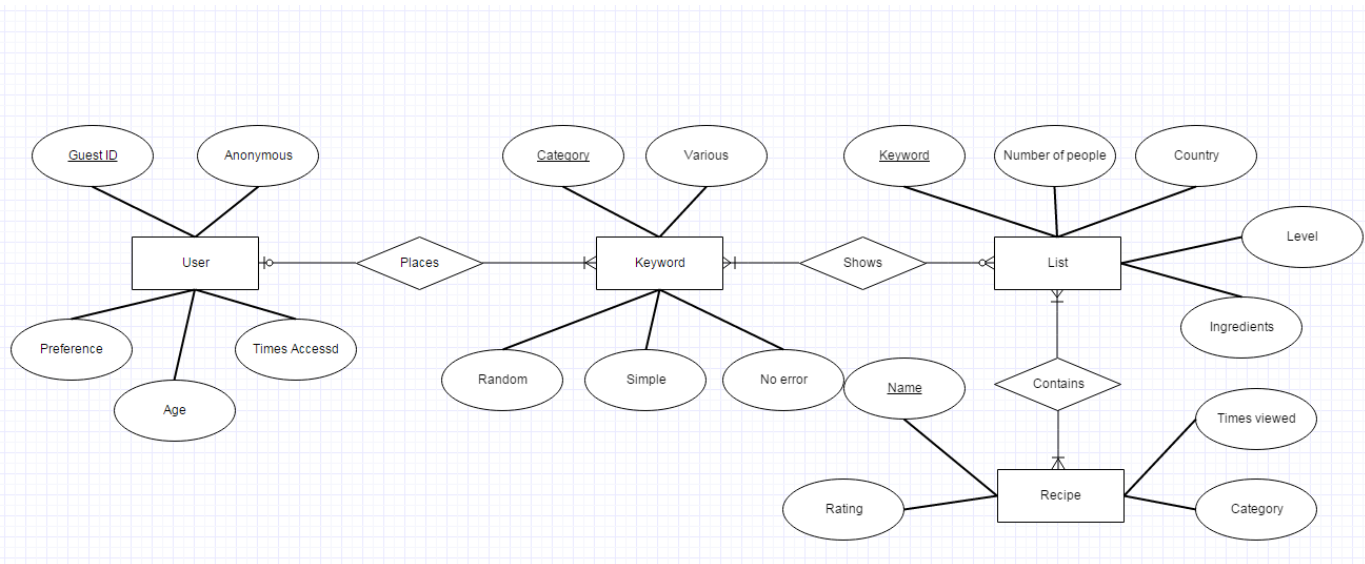
XV. Database Schema - Seojung



XVI. Digital Assets to be used in the project: Images, Fonts - **Sydnee**

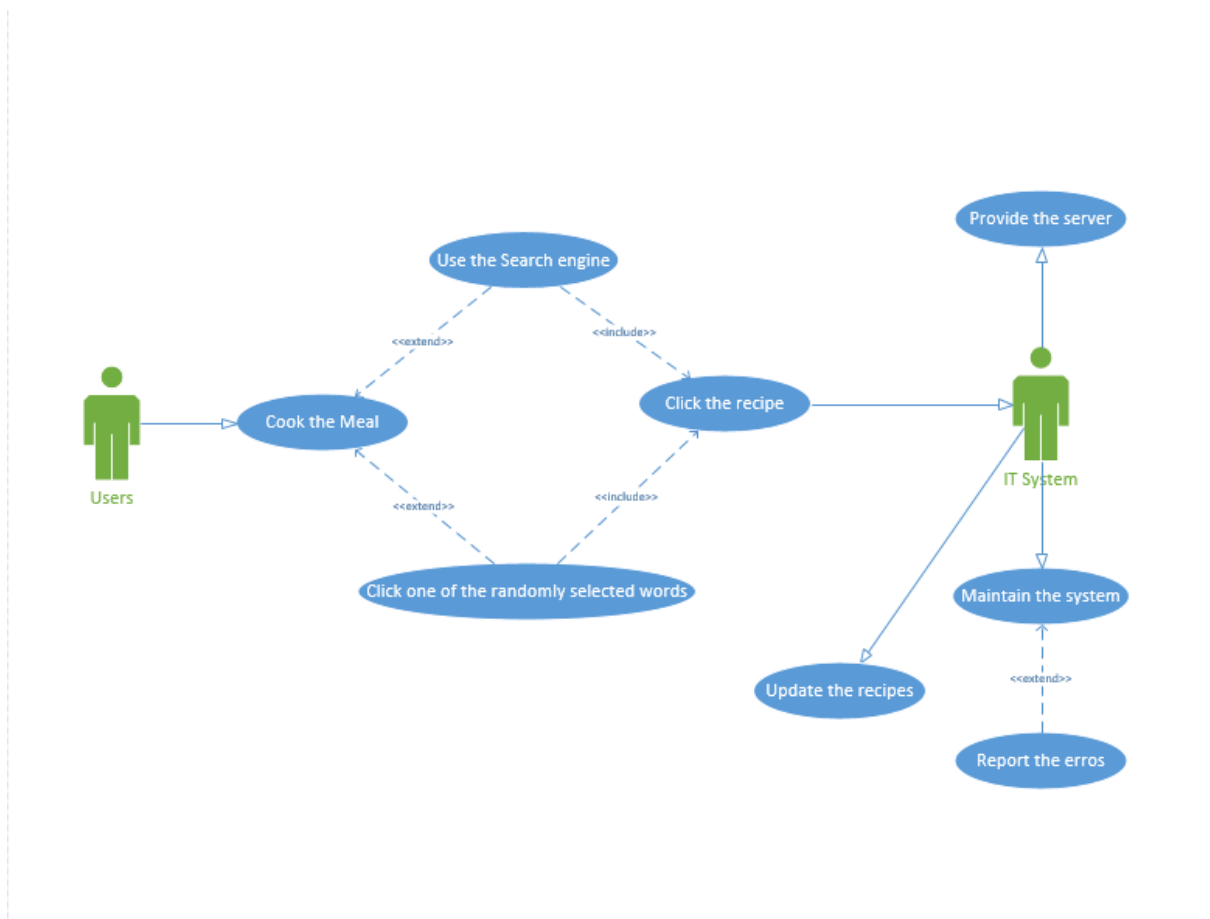
- <https://app.frontify.com/d/mMiFbjNOyc8r/tofu-style-guide>
- We will be using png images found on Google images
- We will be using Arial Narrow for the font

XVII. ERD (Entity Relationship Diagram)- **Hunkyung**

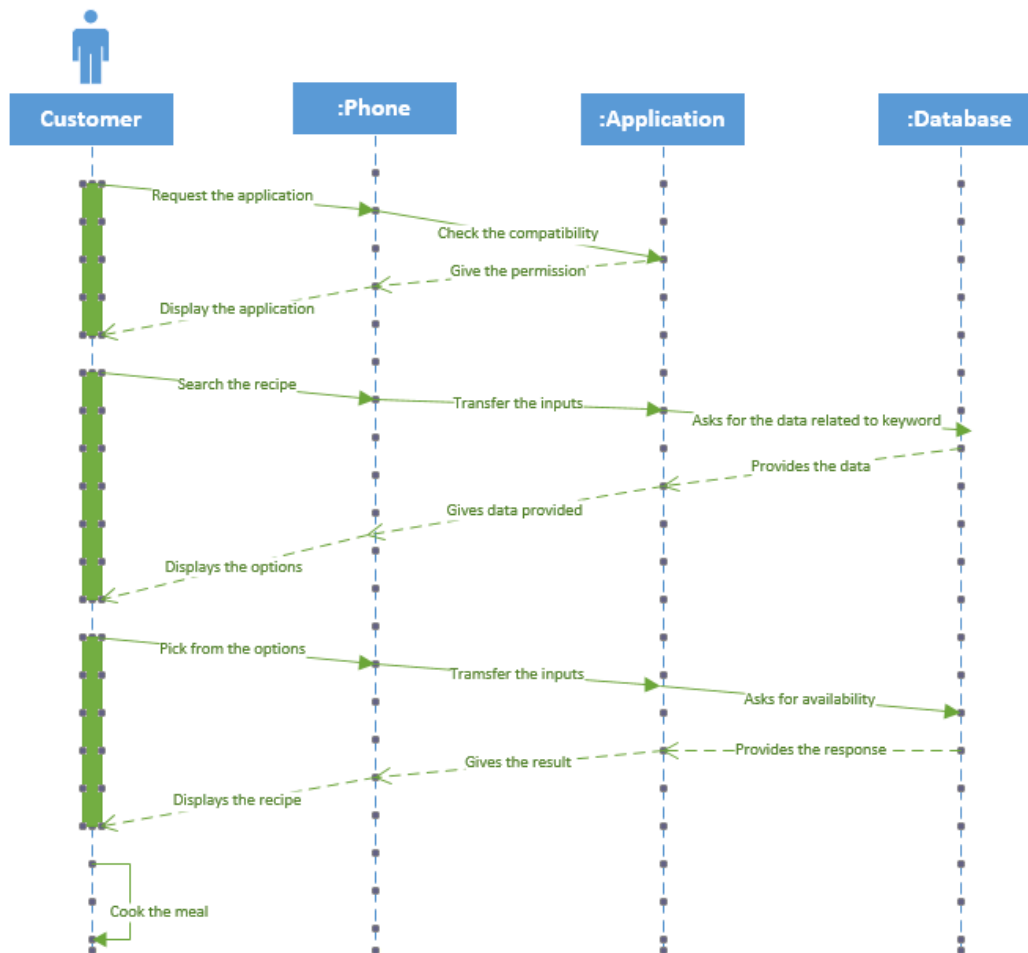


XVIII. UML Diagrams: Use Cases and Sequence Diagrams - **Hunkyung**

a. Use Cases



b. Sequence Diagrams



XIX. Updated Project Plan of the Work Effort Details - **Team**

3. **Sprint 3:** UI Layout, Style Guide, Class Diagram, Database Design Documents, ERD, UML *due by July 17*
 - a. Designing overall user interface layout and database.
 - b. Creating an entity relationship diagram (ERD) that shows the relationships of entity sets stored in a database.
 - c. Developing Unified Modeling Language (UML) to visualize the design of an application.
4. **Sprint 4:** Development: User Interface, Java Code Behind, Database, External Integration, Unit Tests *due by August 7*
 - a. Creating Java and XML code based on the finalized UI layout and database design. Testing the application with Android Studio

SPRINT 4

I. XML markup created for the project with comments

A. step_item.xml - *Hunkyung*

```
1. <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="32dp"
    android:orientation="vertical" >

    <ImageView
      android:id="@+id/stepImageView"
      android:layout_width="fill_parent"
      android:layout_height="300dp"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:layout_marginBottom="16dp"
      android:background="@color/list_image_bg_color"
      android:src="@drawable/ic_noimage" />

    <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_alignBottom="@+id/stepImageView"
      android:layout_alignParentLeft="true"
      android:background="@color/list_text_bg_color"
      android:orientation="vertical" >

      <TextView
        android:id="@+id/textStep"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:textAppearance="@style/RecipeTitleFont" />
```



```
</LinearLayout>
```

```
</RelativeLayout>
```

B. list_item.xml - *Hunkyung*

```
1. <?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="16dp"
    android:orientation="vertical" >

<ImageView
    android:id="@+id/imageView"
    android:layout_width="fill_parent"
    android:layout_height="300dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginBottom="16dp"
    android:background="@color/list_image_bg_color"
    android:src="@drawable/ic_noimage" />

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/imageView"
    android:layout_alignParentLeft="true"
    android:background="@color/list_text_bg_color"
    android:orientation="vertical" >

<TextView
    android:id="@+id/textTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="8dp"
    android:textAppearance="@style/RecipeTitleFont" />
```

```
<TextView
    android:id="@+id/textSummary"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:layout_marginLeft="16dp"
    android:textAppearance="@style/RecipeSummaryFont" />
</LinearLayout>
```

```
</RelativeLayout>
```

C. recipe.xml - Seojung

1. <?xml version="1.0" encoding="utf-8"?>

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

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

```
<ImageView
    android:id="@+id/recipeImageView"
    android:layout_width="fill_parent"
    android:layout_height="300dp"
    android:layout_marginBottom="16dp"
    android:background="@color/list_image_bg_color"
    android:src="@drawable/ic_noimage" />
```

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@color/list_text_bg_color"
    android:orientation="vertical" >
```

```
<TextView
```

```
android:id="@+id/recipeTextTitle"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="8dp"  
android:layout_marginLeft="8dp"  
android:textAppearance="@style/RecipeHugeFont" />
```

```
<TextView  
    android:id="@+id/recipeTextSummary"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="16dp"  
    android:layout_marginLeft="16dp"  
    android:textAppearance="@style/RecipeSummaryFont" />  
</LinearLayout>
```

```
<TextView  
    android:id="@+id/ingredientsHeader"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="8dp"  
    android:layout_marginLeft="8dp"  
    android:text="@string/ingredients"  
    android:textAppearance="@style/RecipeTitleFont"  
    android:visibility="gone" />
```

```
<TextView  
    android:id="@+id/textIngredients"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="8dp"  
    android:layout_marginLeft="8dp"  
    android:textAppearance="@style/RecipeSummaryFont" />
```

```
<TextView  
    android:id="@+id/stepsHeader"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```

        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:text="@string/steps"
        android:textAppearance="@style/RecipeTitleFont"
        android:visibility="gone" />

<LinearLayout
    android:id="@+id/layoutSteps"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:orientation="vertical"
    android:textAppearance="@style/RecipeSummaryFont" >
    </LinearLayout>
</LinearLayout>
</ScrollView>

```

II. Java code created for the project with comments

A. AssetUtils.java - Seojung

1. package com.ist402.android.tofu;

```

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;

```

```

import org.json.JSONException;
import org.json.JSONObject;

```

```

import java.io.IOException;
import java.io.InputStream;

```

```

final class AssetUtils {
    private static final String TAG = "TOFU";

```

```

    public static byte[] loadAsset(Context context, String asset) {
        byte[] buffer = null;
        try {

```

```

        InputStream is = context.getAssets().open(asset);
        int size = is.available();
        buffer = new byte[size];
        is.read(buffer);
        is.close();
    } catch (IOException e) {
        Log.e(TAG, "Failed to load asset " + asset + ": " + e);
    }
    return buffer;
}

```

```

public static JSONObject loadJSONAsset(Context context, String asset)
{
    String jsonString = new String(loadAsset(context, asset));
    JSONObject jsonObject = null;
    try {
        jsonObject = new JSONObject(jsonString);
    } catch (JSONException e) {
        Log.e(TAG, "Failed to parse JSON asset " + asset + ": " + e);
    }
    return jsonObject;
}

```

```

public static Bitmap loadBitmapAsset(Context context, String asset) {
    InputStream is = null;
    Bitmap bitmap = null;
    try {
        is = context.getAssets().open(asset);
        if (is != null) {
            bitmap = BitmapFactory.decodeStream(is);
        }
    } catch (IOException e) {
        Log.e(TAG, e.toString());
    } finally {
        if (is != null) {
            try {
                is.close();
            } catch (IOException e) {

```

```

        Log.e(TAG, "Cannot close InputStream: ", e);
    }
}
}
return bitmap;
}
}

```

B. Constants.java - Seojung

1.

```

package com.ist402.android.tofu;

public final class Constants {
    private Constants() {
    }

    public static final String RECIPE_LIST_FILE = "recipelist.json";
    public static final String RECIPE_NAME_TO_LOAD = "recipe_name";

    public static final String RECIPE_FIELD_LIST = "recipe_list";
    public static final String RECIPE_FIELD_IMAGE = "img";
    public static final String RECIPE_FIELD_INGREDIENTS =
"ingredients";
    public static final String RECIPE_FIELD_NAME = "name";
    public static final String RECIPE_FIELD_SUMMARY = "summary";
    public static final String RECIPE_FIELD_STEPS = "steps";
    public static final String RECIPE_FIELD_TEXT = "text";
    public static final String RECIPE_FIELD_TITLE = "title";
    public static final String RECIPE_FIELD_STEP_TEXT = "step_text";
    public static final String RECIPE_FIELD_STEP_IMAGE = "step_image";

    public static final String EXTRA_RECIPE = "recipe";

    public static final int NOTIFICATION_ID = 0;
    public static final int NOTIFICATION_IMAGE_WIDTH = 280;
    public static final int NOTIFICATION_IMAGE_HEIGHT = 280;
}

```

C. MainActivity.java - Seojung

1. package com.ist402.android.tofu;

```

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ListView;

public class MainActivity extends ListActivity {

    private static final String TAG = "TOFU";
    private RecipeListAdapter mAdapter;

    @Override
    protected void onItemClick(ListView l, View v, int position, long id) {
        if (Log.isLoggable(TAG, Log.DEBUG)) {
            Log.d(TAG, "onItemClick " + position);
        }
        String itemName = mAdapter.getItemName(position);
        Intent intent = new Intent(getApplicationContext(),
RecipeActivity.class);
        intent.putExtra(Constants.RECIPE_NAME_TO_LOAD, itemName);
        startActivity(intent);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(android.R.layout.list_content);

        mAdapter = new RecipeListAdapter(this);
        setListAdapter(mAdapter);
    }
}

```

D. Recipe.java - Hunkyung

1. package com.ist402.android.tofu;

```

import android.content.Context;
import android.os.Bundle;

```

```
import android.os.Parcelable;
import android.util.Log;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class Recipe {
    private static final String TAG = "TOFU";

    public String titleText;
    public String summaryText;
    public String recipelImage;
    public String ingredientsText;

    public static class RecipeStep {
        RecipeStep() { }
        public String stepImage;
        public String stepText;

        public Bundle toBundle() {
            Bundle bundle = new Bundle();
            bundle.putString(Constants.RECIPE_FIELD_STEP_TEXT,
stepText);
            bundle.putString(Constants.RECIPE_FIELD_STEP_IMAGE,
stepImage);
            return bundle;
        }

        public static RecipeStep fromBundle(Bundle bundle) {
            RecipeStep recipeStep = new RecipeStep();
            recipeStep.stepText =
bundle.getString(Constants.RECIPE_FIELD_STEP_TEXT);
            recipeStep.stepImage =
bundle.getString(Constants.RECIPE_FIELD_STEP_IMAGE);
            return recipeStep;
        }
    }
}
```



```

    }
}
ArrayList<RecipeStep> recipeSteps;

public Recipe() {
    recipeSteps = new ArrayList<RecipeStep>();
}

public static Recipe fromJson(Context context, JSONObject json) {
    Recipe recipe = new Recipe();
    try {
        recipe.titleText =
json.getString(Constants.RECIPE_FIELD_TITLE);
        recipe.summaryText =
json.getString(Constants.RECIPE_FIELD_SUMMARY);
        if (json.has(Constants.RECIPE_FIELD_IMAGE)) {
            recipe.recipeImage =
json.getString(Constants.RECIPE_FIELD_IMAGE);
        }
        JSONArray ingredients =
json.getJSONArray(Constants.RECIPE_FIELD_INGREDIENTS);
        recipe.ingredientsText = "";
        for (int i = 0; i < ingredients.length(); i++) {
            recipe.ingredientsText += " - "
                +
ingredients.getJSONObject(i).getString(Constants.RECIPE_FIELD_TEXT)
+ "\n";
        }

        JSONArray steps =
json.getJSONArray(Constants.RECIPE_FIELD_STEPS);
        for (int i = 0; i < steps.length(); i++) {
            JSONObject step = steps.getJSONObject(i);
            RecipeStep recipeStep = new RecipeStep();
            recipeStep.stepText =
step.getString(Constants.RECIPE_FIELD_TEXT);
            if (step.has(Constants.RECIPE_FIELD_IMAGE)) {
                recipeStep.stepImage =

```

```

step.getString(Constants.RECIPE_FIELD_IMAGE);
    }
    recipe.recipeSteps.add(recipeStep);
    }
} catch (JSONException e) {
    Log.e(TAG, "Error loading recipe: " + e);
    return null;
}
return recipe;
}

public Bundle toBundle() {
    Bundle bundle = new Bundle();
    bundle.putString(Constants.RECIPE_FIELD_TITLE, titleText);
    bundle.putString(Constants.RECIPE_FIELD_SUMMARY,
summaryText);
    bundle.putString(Constants.RECIPE_FIELD_IMAGE, recipelImage);
    bundle.putString(Constants.RECIPE_FIELD_INGREDIENTS,
ingredientsText);
    if (recipeSteps != null) {
        ArrayList<Parcelable> stepBundles = new
ArrayList<Parcelable>(recipeSteps.size());
        for (RecipeStep recipeStep : recipeSteps) {
            stepBundles.add(recipeStep.toBundle());
        }

bundle.putParcelableArrayList(Constants.RECIPE_FIELD_STEPS,
stepBundles);
    }
    return bundle;
}

public static Recipe fromBundle(Bundle bundle) {
    Recipe recipe = new Recipe();
    recipe.titleText =
bundle.getString(Constants.RECIPE_FIELD_TITLE);
    recipe.summaryText =
bundle.getString(Constants.RECIPE_FIELD_SUMMARY);
}

```

```

        recipe.recipeImage =
bundle.getString(Constants.RECIPE_FIELD_IMAGE);
        recipe.ingredientsText =
bundle.getString(Constants.RECIPE_FIELD_INGREDIENTS);
        ArrayList<Parcelable> stepBundles =

bundle.getParcelableArrayList(Constants.RECIPE_FIELD_STEPS);
        if (stepBundles != null) {
            for (Parcelable stepBundle : stepBundles) {
                recipe.recipeSteps.add(RecipeStep.fromBundle((Bundle)
stepBundle));
            }
        }
        return recipe;
    }
}

```

E. RecipeActivity.java - *Hunkyung*

1. package com.ist402.android.tofu;

```

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import org.json.JSONObject;

public class RecipeActivity extends Activity {
    private static final String TAG = "TOFU";

```

```
private String mRecipeName;
private Recipe mRecipe;
private ImageView mImageView;
private TextView mTitleTextView;
private TextView mSummaryTextView;
private TextView mIngredientsTextView;
private LinearLayout mStepsLayout;
```

```
@Override
```

```
protected void onStart() {
    super.onStart();
    Intent intent = getIntent();
    mRecipeName =
intent.getStringExtra(Constants.RECIPE_NAME_TO_LOAD);
    if (Log.isLoggable(TAG, Log.DEBUG)) {
        Log.d(TAG, "Intent: " + intent.toString() + " " + mRecipeName);
    }
    loadRecipe();
}
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.recipe);
    mTitleTextView = (TextView) findViewById(R.id.recipeTextTitle);
    mSummaryTextView = (TextView)
findViewById(R.id.recipeTextSummary);
    mImageView = (ImageView) findViewById(R.id.recipeImageView);
    mIngredientsTextView = (TextView)
findViewById(R.id.textIngredients);
    mStepsLayout = (LinearLayout) findViewById(R.id.layoutSteps);
}
```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

```
}
```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()) {  
        case R.id.action_cook:  
            return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

```
private void loadRecipe() {  
    JSONObject jsonObject = AssetUtils.loadJSONAsset(this,  
mRecipeName);  
    if (jsonObject != null) {  
        mRecipe = Recipe.fromJson(this, jsonObject);  
        if (mRecipe != null) {  
            displayRecipe(mRecipe);  
        }  
    }  
}
```

```
private void displayRecipe(Recipe recipe) {  
    Animation fadeIn = AnimationUtils.loadAnimation(this,  
android.R.anim.fade_in);  
    mTitleTextView.setAnimation(fadeIn);  
    mTitleTextView.setText(recipe.titleText);  
    mSummaryTextView.setText(recipe.summaryText);  
    if (recipe.recipeImage != null) {  
        mImageView.setAnimation(fadeIn);  
        Bitmap recipeImage = AssetUtils.loadBitmapAsset(this,  
recipe.recipeImage);  
        mImageView.setImageBitmap(recipeImage);  
    }  
    mIngredientsTextView.setText(recipe.ingredientsText);  
  
    findViewById(R.id.ingredientsHeader).setAnimation(fadeIn);  
    findViewById(R.id.ingredientsHeader).setVisibility(View.VISIBLE);
```

```

findViewById(R.id.stepsHeader).setAnimation(fadeIn);

findViewById(R.id.stepsHeader).setVisibility(View.VISIBLE);

LayoutInflater inf = LayoutInflater.from(this);
mStepsLayout.removeAllViews();
int stepNumber = 1;
for (Recipe.RecipeStep step : recipe.recipeSteps) {
    View view = inf.inflate(R.layout.step_item, null);
    ImageView iv = (ImageView)
view.findViewById(R.id.stepImageView);
    if (step.stepImage == null) {
        iv.setVisibility(View.GONE);
    } else {
        Bitmap stepImage = AssetUtils.loadBitmapAsset(this,
step.stepImage);
        iv.setImageBitmap(stepImage);
    }
    ((TextView) view.findViewById(R.id.textStep)).setText(
        (stepNumber++) + ". " + step.stepText);
    mStepsLayout.addView(view);
}
}
}

```

F. RecipeListAdapter.java - HunKyung

1. package com.ist402.android.tofu;

```

import android.content.Context;
import android.database.DataSetObserver;
import android.graphics.Bitmap;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.ListAdapter;
import android.widget.TextView;

```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class RecipeListAdapter implements ListAdapter {
    private String TAG = "TOFU";

    private class Item {
        String title;
        String name;
        String summary;
        Bitmap image;
    }

    private List<Item> mItems = new ArrayList<Item>();
    private Context mContext;
    private DataSetObserver mObserver;

    public RecipeListAdapter(Context context) {
        mContext = context;
        loadRecipeList();
    }

    private void loadRecipeList() {
        JSONObject jsonObject = AssetUtils.loadJSONAsset(mContext,
Constants.RECIPE_LIST_FILE);
        if (jsonObject != null) {
            List<Item> items = parseJson(jsonObject);
            appendItemsToList(items);
        }
    }

    private List<Item> parseJson(JSONObject json) {
        List<Item> result = new ArrayList<Item>();
        try {

```

```

        JSONArray items =
json.getJSONArray(Constants.RECIPE_FIELD_LIST);
        for (int i = 0; i < items.length(); i++) {
            JSONObject item = items.getJSONObject(i);
            Item parsed = new Item();
            parsed.name =
item.getString(Constants.RECIPE_FIELD_NAME);
            parsed.title = item.getString(Constants.RECIPE_FIELD_TITLE);
            if (item.has(Constants.RECIPE_FIELD_IMAGE)) {
                String imageFile =
item.getString(Constants.RECIPE_FIELD_IMAGE);
                parsed.image = AssetUtils.loadBitmapAsset(mContext,
imageFile);
            }
            parsed.summary =
item.getString(Constants.RECIPE_FIELD_SUMMARY);
            result.add(parsed);
        }
    } catch (JSONException e) {
        Log.e(TAG, "Failed to parse recipe list: " + e);
    }
    return result;
}

private void appendItemsToList(List<Item> items) {
    mItems.addAll(items);
    if (mObserver != null) {
        mObserver.onChangeed();
    }
}

@Override
public int getCount() {
    return mItems.size();
}

@Override
public Object getItem(int position) {

```



```

        return mItems.get(position);
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public int getItemViewType(int position) {
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        View view = convertView;
        if (view == null) {
            LayoutInflater inf = LayoutInflater.from(mContext);
            view = inf.inflate(R.layout.list_item, null);
        }
        Item item = (Item) getItem(position);
        TextView titleView = (TextView) view.findViewById(R.id.textTitle);
        TextView summaryView = (TextView)
view.findViewById(R.id.textSummary);
        ImageView iv = (ImageView) view.findViewById(R.id.imageView);

        titleView.setText(item.title);
        summaryView.setText(item.summary);
        if (item.image != null) {
            iv.setImageBitmap(item.image);
        } else {
            iv.setImageDrawable(mContext.getResources().getDrawable(R.drawable.
ic_noimage));
        }
        return view;
    }

```

```
@Override
public int getViewTypeCount() {
    return 1;
}
```

```
@Override
public boolean hasStableIds() {
    return false;
}
```

```
@Override
public boolean isEmpty() {
    return mItems.isEmpty();
}
```

```
@Override
public void registerDataSetObserver(DataSetObserver observer) {
    mObserver = observer;
}
```

```
@Override
public void unregisterDataSetObserver(DataSetObserver observer) {
    mObserver = null;
}
```

```
@Override
public boolean areAllItemsEnabled() {
    return true;
}
```

```
@Override
public boolean isEnabled(int position) {
    return true;
}
```

```
public String getItemName(int position) {
    return mItems.get(position).name;
}
```

```
}  
}
```

G. RecipeService.java - Seojung

```
1. package com.ist402.android.tofu;
```

```
import android.app.Notification;  
import android.app.Service;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.os.Binder;  
import android.os.IBinder;  
import android.support.v4.app.NotificationManagerCompat;  
import android.support.v4.app.NotificationCompat;
```

```
import java.util.ArrayList;
```

```
public class RecipeService extends Service {  
    private NotificationManagerCompat mNotificationManager;  
    private Binder mBinder = new LocalBinder();  
    private Recipe mRecipe;
```

```
    public class LocalBinder extends Binder {  
        RecipeService getService() {  
            return RecipeService.this;  
        }  
    }  
}
```

```
@Override
```

```
public void onCreate() {  
    mNotificationManager = NotificationManagerCompat.from(this);  
}
```

```
@Override
```

```
public IBinder onBind(Intent intent) {  
    return mBinder;  
}
```

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {

    return START_NOT_STICKY;
}

private void createNotification(Intent intent) {
    mRecipe =
Recipe.fromBundle(intent.getBundleExtra(Constants.EXTRA_RECIPE));
    ArrayList<Notification> notificationPages = new
ArrayList<Notification>();

    int stepCount = mRecipe.recipeSteps.size();

    for (int i = 0; i < stepCount; ++i) {
        Recipe.RecipeStep recipeStep = mRecipe.recipeSteps.get(i);
        NotificationCompat.BigTextStyle style = new
NotificationCompat.BigTextStyle();
        style.bigText(recipeStep.stepText);
        style.setBigContentTitle(String.format(
            getResources().getString(R.string.step_count), i + 1,
stepCount));
        style.setSummaryText("");
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this);
        builder.setStyle(style);
        notificationPages.add(builder.build());
    }

    NotificationCompat.Builder builder = new
NotificationCompat.Builder(this);

    if (mRecipe.recipeImage != null) {
        Bitmap recipeImage = Bitmap.createScaledBitmap(
            AssetUtils.loadBitmapAsset(this, mRecipe.recipeImage),
            Constants.NOTIFICATION_IMAGE_WIDTH,
Constants.NOTIFICATION_IMAGE_HEIGHT, false);
        builder.setLargeIcon(recipeImage);
    }
}

```

```

    }
    builder.setTitle(mRecipe.titleText);
    builder.setText(mRecipe.summaryText);
    builder.setSmallIcon(R.mipmap.ic_notification_recipe);

    Notification notification = builder
        .extend(new NotificationCompat.WearableExtender()
            .addPages(notificationPages))
        .build();
    mNotificationManager.notify(Constants.NOTIFICATION_ID,
        notification);
    }
}

```

iii. JSON Files - *Seojung and Hunkyung*

A. Northern-irish-vegetable-soup.json

```

{
  name: "northern-irish-vegetable-soup.json",
  steps: [
    {
      text: "Place the beef in a large stock pot, cover with water and stew for 1 - 2 hours."
    },
    {
      text: "Allow the stock to cool, skim off any fat."
    },
    {
      text: "Add the soup mix to the stock, bring to the boil and simmer for 1 hour."
    },
    {
      text: "Roughly chop the leeks (green and white parts), onion, carrots, celery and
parsley. Add to the soup. Season well and simmer until the vegetables are soft."
    }
  ],
}

```

summary: "This recipe is apparently unique to Northern Ireland and uses soup/herb celery which is hard to find outside the area, but regular table celery can be substituted (including the leaves).",

title: "Northern irish vegetable soup",

img: "irish-stew.jpg",

serving: [

"Whole boiled potatoes are traditionally placed in the soup at time of serving."

],

ingredients: [

{

text: "2 lbs beef shin or similar beef on bone"

},

{

text: "60g soup mix (30g barley, 15g red lentils, 15g split peas)"

},

{

text: "3 carrots"

},

{

text: "1 white onion"

},

{

text: "field celery or 1 stalk celery, plus any leaves on the bunch"

},

{

text: "2 leeks"

},

{

text: "1 bunch parsley"

}

]

}

B. Guacamole.json

```
{
  name: "guacamole.json",
  steps: [
    {
      text: "Use a spoon to scoop the flesh of the avocados into a bowl."
    },
    {
      text: "Mash with a fork until fairly smooth and creamy. Preserve some small solid
      chunks to add texture."
    },
    {
      text: "Add the juice of the lime. ",
      img: "step-3-lime.jpg"
    },
    {
      text: "Add the cilantro."
    },
    {
      text: "Mix thoroughly."
    }
  ],
  summary: "Some guacamole recipes call for many ingredients and can be a pain to
  prepare. This super simple guac can be thrown together in a couple of minutes and
  tastes great.",
  title: "Super simple guacamole",
  img: "guacamole.jpg",
  serving: "",
  ingredients: [
    {
      text: "2 ripe avocados"
```

```
},  
{  
text: "1 lime"  
},  
{  
text: "2 tbsp cilantro"  
}  
]  
}
```

C. Beef-brisket-chili.json

```
{  
name: "beef-brisket-chili.json",  
steps: [  
{  
text: "Place the cinnamon stick, cumin, paprika and oregano in a pestle and mortar.  
Bash and grind until the cinnamon is in very small flakes. ",  
img: "step-1-grind-spices.jpg"  
},  
{  
text: "Score the beef and rub the spice mix into the cuts. ",  
img: "step-2-score-beef.jpg"  
},  
{  
text: "Season the beef and drizzle with olive oil. Brown in a hot pan. ",  
img: "step-3-brown-beef.jpg"  
},  
{  
text: "Finely slice the onion and fry in a little olive oil together with the chillies, finely  
sliced. You can control the heat of your chili by deseeding some of the chillies. ",  
img: "step-4-fry-onion.jpg"  
},  
]
```



```
{
  text: "Add the browned brisket to the pan with the onions and chillies and fry on a low heat."
},
{
  text: "Place the bell peppers, roughly sliced, the bay leaves and chopped tomatoes in a large covered pot. Bring to the boil and then add the beef, onions and chillies. ",
  img: "step-6-combine.jpg"
},
{
  text: "Bring back to the boil, cover and allow to simmer on a low heat for 4 hours."
},
{
  text: "Remove the brisket from the pot, place on a large plate and use two forks to pull the beef apart into individual pieces. ",
  img: "step-8-pull.jpg"
},
{
  text: "Remove the bay leaves and add the pulled beef and coriander back to the pot, together with the kidney beans."
},
{
  text: "Bring back to the boil and simmer gently for a further 15 - 20 mins."
}
],
summary: "",
title: "Beef brisket chili",
img: "chili.jpg",
serving: [
  "Serve with rice, yoghurt and fresh guacamole. Garnish with the remaining coriander."
],
```

ingredients: [

{

text: "3 lbs beef brisket"

},

{

text: "2 red onions"

},

{

text: "4 jalapeno chillies"

},

{

text: "1 large cinnamon stick"

},

{

text: "1 tbsp ground cumin"

},

{

text: "1 tbsp paprika"

},

{

text: "1 heaped tbsp dried oregano"

},

{

text: "2 fresh bay leaves"

},

{

text: "2 red bell peppers"

},

{

text: "2 green bell peppers"

},

```
{
  text: "800g tin chopped tomatoes"
},
{
  text: "400g tin kidney beans"
},
{
  text: "400ml beef stock"
},
{
  text: "1/2 bunch coriander"
}
]
```

D. Sandwich.json

```
{
  name: "sandwich.json",
  steps: [
    {
      text: "Place the bread on the bottom"
    },
    {
      text: "Put your vegetables, cheese, and ham or turkey in order you want "
    },
    {
      text: "Put the one left slice of bread on the top"
    },
    {
      text: "Enjoy eating!"
    }
  ]
}
```

],

summary: "This recipe is available all around the world. Easy to cook and eat, but in the same time, it contains much ingredients that we need",

title: "Basic Sandwich",

img: "sandwich.jpg",

serving: [

"Served mostly with cup of drink."

],

ingredients: [

{

text: "2 slices of bread"

},

{

text: "Few lettuce leaves"

},

{

text: "Tomatos"

},

{

text: "Some slices of cheese"

},

{

text: "ham or turkey"

},

{

text: "Additional vegetables"

}

]

}

E. Kimchi-soup.json

```
{
  name: "kimchi-soup.json",
  steps: [
    {
      text: "Put pork belly in a bowl. Add garlic, ginger, soy sauce, sesame oil and fish
      sauce. Toss well to coat and let marinate for 10 minutes."
    },
    {
      text: "Set a heavy-bottomed soup pot over medium heat. Melt butter, then add pork
      belly mixture and let it cook gently for 5 minutes. Add onion and cook, stirring, until
      softened, about 5 minutes. Turn heat to medium high and add kimchi, gochujang and
      gochugaru. Let mixture simmer for 2 minutes."
    },
    {
      text: "Add kimchi juice and water (or broth, if using) and bring to a boil. Reduce heat
      to a brisk simmer and cook for 20 minutes. Taste broth and adjust seasoning. "
    },
    {
      text: "Just before serving, add tofu and stir gently to combine. When tofu is heated
      through, ladle into bowls and garnish with scallions."
    }
  ],
  summary: "In Korean cuisine, kimchi is not only a condiment or pickle; it is also used
  as an ingredient in many cooked dishes. Kimchi soup, called Jigae, is a satisfying
  example. The fermented kimchi vegetables quickly provide deep flavor to the broth,
  so the soup can be produced in little more than half an hour. Aged kimchi, which is
  more intensely sour, is the preferred type to use. Surprisingly mellow, the soup is
  neither too spicy nor too salty.",
  title: "Kimchi Soup",
  img: "kimchi.jpg",
  serving: "",
  ingredients: [
    {
```

```
text: "1 pound fresh pork belly, cut in 1/2-inch pieces"
},
{
text: "4 garlic cloves, minced"
},
{
text: "3 tablespoons Korean red pepper paste (gochujang)"
},
{
text: "1 cup kimchi juice"
},
{
text: "2 cups kimchi, aged if possible, squeezed dry and chopped"
}
]
}
```

F. Cake.json

```
{
name: "cake.json",
steps: [
{
text: "Preheat oven to 350 degrees F (175 degrees C). Grease and flour a 9x9 inch
pan or line a muffin pan with paper liners."
},
{
text: "In a medium bowl, cream together the sugar and butter. Beat in the eggs, one
at a time, then stir in the vanilla. Combine flour and baking powder, add to the
creamed mixture and mix well. Finally stir in the milk until batter is smooth. Pour or
spoon batter into the prepared pan."
},
{

```

text: "Bake for 30 to 40 minutes in the preheated oven. For cupcakes, bake 20 to 25 minutes. Cake is done when it springs back to the touch."

},

{

text: "Kids come!"

}

],

summary: "This recipe is to make the basic cake for the desert. Get the loves from the family with the sweetness",

title: "Cake Cake",

img: "cake.jpg",

serving: [

"Served mostly with cup of coffee."

],

ingredients: [

{

text: "1 cup white sugar"

},

{

text: "1/2 cup butter"

},

{

text: "2 eggs"

},

{

text: "2 teaspoons vanilla extract"

},

{

text: "1/2 cup milk"

},

{

text: "1 1/2 cups all-purpose flour"

```
},  
{  
  text: "1 3/4 teaspoons baking powder"  
}  
]  
}
```

G. Pizza.json

```
{  
  name: "pizza.json",  
  steps: [  
    {  
      text: "Place the sugar, salt, olive oil, water, 1 cup of flour, yeast, and remaining cup of  
      flour into a standing mixer's work bowl."  
    },  
    {  
      text: "Tear off a small piece of dough and flatten into a disc."  
    },  
    {  
      text: "Roll the pizza dough into a smooth ball on the countertop."  
    },  
    {  
      text: "Bake!"  
    }  
  ],  
  summary: "",  
  title: "Basic Pizza",  
  img: "pizza.jpg",  
  serving: "8",  
  ingredients: [  
    {  
      text: "1 3/4 teaspoons baking powder"  
    }  
  ]  
}
```



```
{
  text: "800g strong white bread flour"
},
{
  text: "1 clove garlic"
},
{
  text: "85g mozzarella"
},
{
  text: "400g good-quality tinned plum tomatoes"
},
{
  text: "extra virgin olive oil"
}
]
```

H. Budaejjigae.json

```
{
  name: "Budaejjigae.json",
  steps: [
    {
      text: "Put everything together except salt and pepper and boil"
    },
    {
      text: "Season with salt and pepper"
    },
    {
      text: "There you go! Have fun eating!"
    }
  ]
}
```

],

summary: "This recipe is to make Korean traditional soup that is spicy. It is usually good with the alcohols, specially soju, Korean traditional alcohol with 19 degree.",

title: "Budaejjigae",

img: "Budaejjigae.jpg",

serving: [

"Served mostly with alcohol and rice."

],

ingredients: [

{

text: "Ham, sausage, cheese and tofu"

},

{

text: "Minced Garlic, onion, and green onion"

},

{

text: "Sagol soup"

},

{

text: "Salt, pepper"

},

{

text: "Kimchi"

},

{

text: "Water from kimchi"

},

{

text: "Additional: Noodle"

}

]

}

I. Spaghetti.json

```
{
  name: "spaghetti.json",
  steps: [
    {
      text: "Combine ground beef, onion, garlic, and green pepper in a large saucepan.
      Cook and stir until meat is brown and vegetables are tender. Drain grease."
    },
    {
      text: "Stir diced tomatoes, tomato sauce, and tomato paste into the pan. Season with
      oregano, basil, salt, and pepper. Simmer spaghetti sauce for 1 hour, stirring
      occasionally."
    },
    {
      text: "Enjoy eating!"
    }
  ],
  summary: "",
  title: "Spaghetti Sauce with Ground Beef",
  img: "spaghetti.jpg",
  serving: [
    "Served mostly with cup of drink or soda."
  ],
  ingredients: [
    {
      text: "1 pound ground beef"
    },
    {
      text: "1 onion, chopped"
    },
  ],
}
```

```
{
  text: "4 cloves garlic, minced"
},
{
  text: "2 teaspoons dried basil"
}
]
```

J. Ramyun.json

```
{
  name: "ramyun.json",
  steps: [
    {
      text: "Place the water on the pen and turn on the fire"
    },
    {
      text: "Put everything except the egg when the water boils"
    },
    {
      text: "Put the egg in when the noodle is almost ready to eat!"
    },
    {
      text: "Enjoy eating!"
    }
  ],
  summary: "This food is Korean instant noodle that can be cooked and eaten in 10 minutes, but mucho delicioso",
  title: "Ramyun",
  img: "ramyun.jpg",
```

```
    serving: [  
      "Served mostly with Kimchi, Danmooji, and water."  
    ],  
    ingredients: [  
      {  
        text: "Instant noodle"  
      },  
      {  
        text: "Noodle soup sauce"  
      },  
      {  
        text: "Egg"  
      },  
      {  
        text: "Water"  
      }  
    ]  
  }  
}
```

K. Koreandonkatsu.json

```
{  
  name: "koreandonkatsu.json",  
  steps: [  
    {  
      text: "1) Trim off fat from pork chops (usually on the sides), then pound into thin pieces with tenderizer. Season with salt and pepper."  
    },  
    {  
      text: "2) Coat pork with flour, then beaten egg, and lastly panko bread crumbs."  
    },  
  ]  
}
```

```
{
text: "3) Heat vegetable oil in pan or skillet on medium heat."
},
{
text: "4) Fry until golden brown and cooked through (you can cut to check insides).
Usually takes about 8~10 minutes total if fully submerged in oil."
},
{
text: "5) Remove cutlets from the oil and drain on paper towel to remove excess oil
and cool."
},
{
text: "6) Drain shredded cabbage well. Cut each cutlet into small pieces and serve on
plate with cabbage slaw. "
},
{
text: "7) Serve with katsu sauce for dipping and serve with rice."
}
],
summary: "This food is fired pork called Korean Donkatsu",
title: "Korean Donkatsu",
img: "koreandonkatsu.jpg",
serving: [
"Served mostly with Kimchi, Danmooji, and water."
],
ingredients: [
{
text: "4 boneless pork chops (thin cut)"
},
{
text: "2 beaten eggs"
},
```

```
{
  text: "2 cups panko bread crumbs"
},
{
  text: "1/2 carrot, grated or shredded"
},
{
  text: "2 cups panko bread crumbs"
},
{
  text: "1/4 head cabbage, thinly shredded"
},
{
  text: "1/2 cup flour"
},
{
  text: "katsu sauce"
},
{
  text: "vegetable oil"
},
{
  text: "salt and pepper to taste"
}
]
}
```

L. Insalata.json

```
{
  name: "insalata.json",
```

```
steps: [  
  {  
    text: "On a large platter, alternate and overlap the tomato slices, mozzarella cheese  
    slices, and basil leaves. Drizzle with olive oil. Season with sea salt and pepper."  
  },  
  {  
    text: "Enjoy Eating!"  
  }  
],  
summary: " ",  
title: "Insalata Caprese",  
img: "insalata.jpg",  
serving: [  
  "Served mostly with cup of drink."  
],  
ingredients: [  
  {  
    text: "4 large ripe tomatoes, sliced 1/4 inch thick"  
  },  
  {  
    text: "1 pound fresh mozzarella cheese, sliced 1/4 inch thick"  
  },  
  {  
    text: "1/3 cup fresh basil leaves"  
  },  
  {  
    text: "3 tablespoons extra virgin olive oil"  
  },  
  {  
    text: "freshly ground black pepper to taste"  
  },  
  {
```



```
text: "fine sea salt to taste"
}
]
}
```

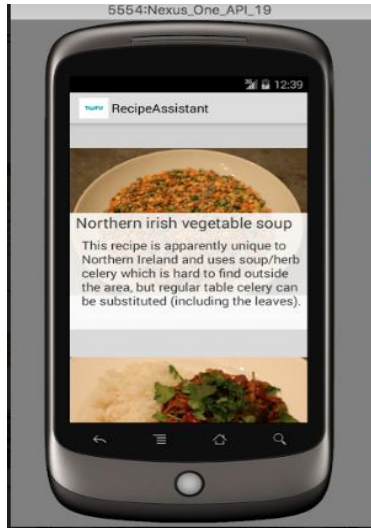
M. Recipelist.json

```
{
  recipe_list:
  [
    {
      name: "guacamole.json",
      summary: "Some guacamole recipes call for many ingredients and can be a pain to
      prepare. This super simple guac can be thrown together in a couple of minutes and
      tastes great.",
      title: "Super simple guacamole",
      img: "guacamole.jpg"
    },
    {
      name: "northern-irish-vegetable-soup.json",
      summary: "This recipe is apparently unique to Northern Ireland and uses soup/herb
      celery which is hard to find outside the area, but regular table celery can be
      substituted (including the leaves).",
      title: "Northern irish vegetable soup",
      img: "irish-stew.jpg"
    },
    {
      name: "beef-brisket-chili.json",
      summary: " ",
      title: "Beef brisket chili",
      img: "chili.jpg"
    },
    {
      name: "sandwich.json",
```

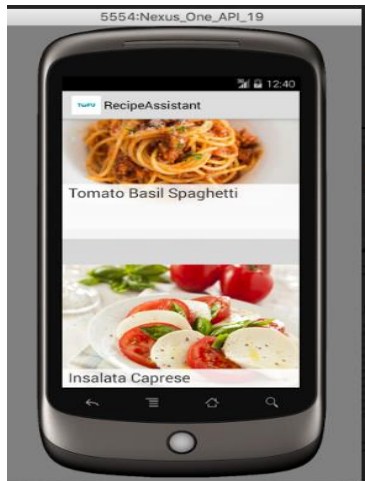
```
summary: "",
title: "Basic Sandwich",
img: "sandwich.jpg"
},
{
name: "pizza.json",
summary: "",
title: "Basic Pizza",
img: "pizza.jpg"
},
{
name: "spaghetti.json",
summary: "",
title: "Tomato Basil Spaghetti",
img: "spaghetti.jpg"
},
{
name: "insalata.json",
summary: " ",
title: "Insalata Caprese",
img: "insalata.jpg"
},
{
name: "cake.json",
summary: "",
title: "Cake",
img: "cake.jpg"
},
{
name: "Budaejjigae.json",
summary: "",
```

```
title: "Budaejjigae",
img: "Budaejjigae.jpg"
},
{
name: "ramyun.json",
summary: "",
title: "Ramyun",
img: "ramyun.jpg"
},
{
name: "koreandonkatsu.json",
summary: "",
title: "Korean Donkatsu",
img: "koreandonkatsu.jpg"
},
{
name: "kimchi-soup.json",
summary: "",
title: "Kimchi Soup",
img: "kimchi.jpg"
}
]
}
```

IV. Screen shots of the project running - *Seojung*



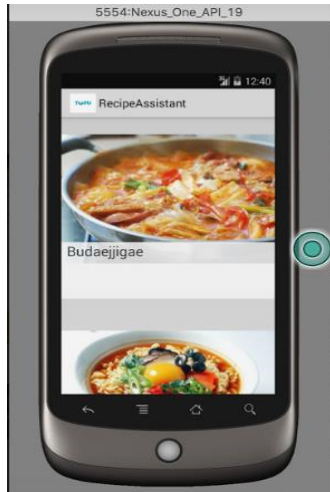
A.



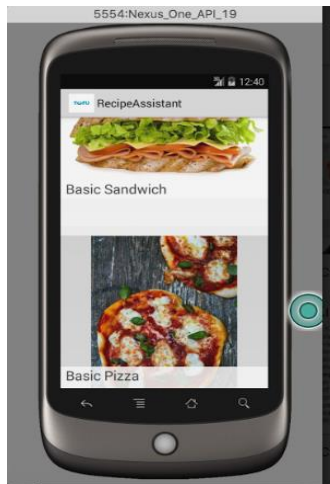
B.



C.



D.



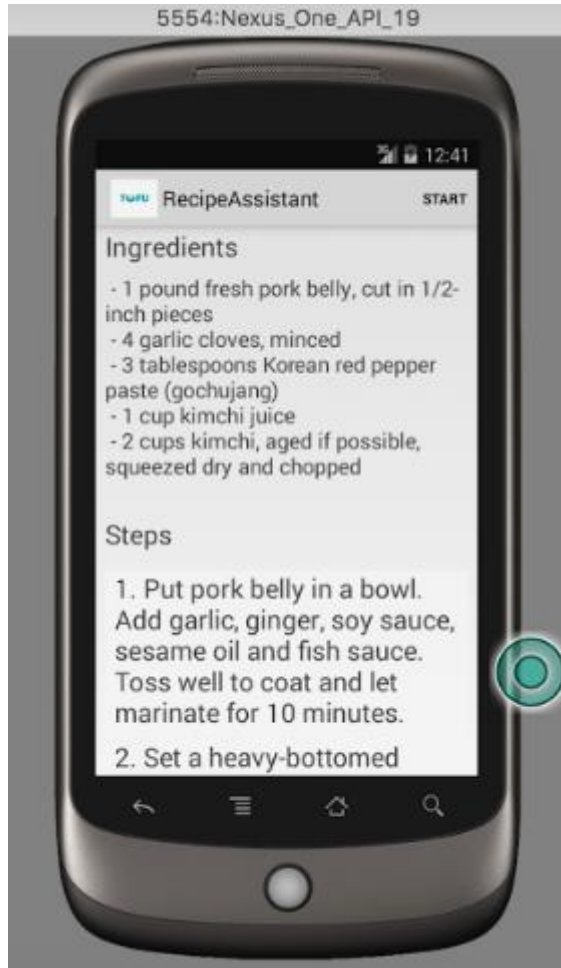
E.



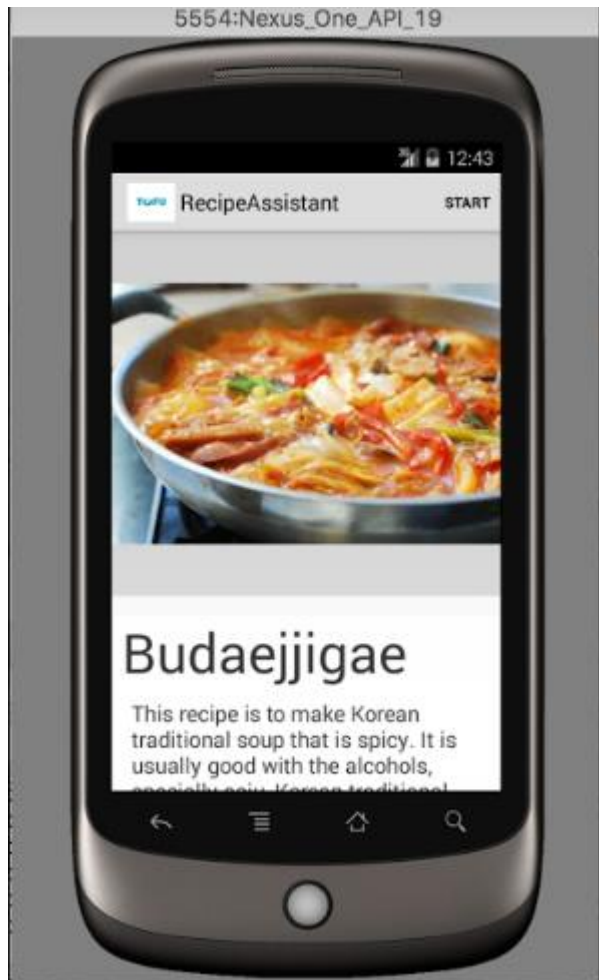
F.



G.



H.



1.



J.

v. Testing plan

- A. Test 1: User clicks photo on the app
- B. Test 2: User wants to go back to previous page
- C. Test 3: User clicks recipe name instead of recipe photo on the app
- D. Test 4: User clicks on white spaces between photos of recipes

vi. Unit Test Results

A.

Test #	Input	Result
Test 1	User tries to click on Beef photo	Negative - Application stops running
Test 1	User tries to click on Sandwich photo	Negative- Application stops running
Test 1	User clicks on Guacamole photo	Positive- sends user to guacamole recipe

Test 2	User uses back button to exit from guacamole recipe	Positive- sends user to previous page
Test 1	User tries to click on soup photo	Negative- Application stops running
Test 1	User clicks on pizza photo	Positive- sends user to pizza recipe
Test 2	User uses back button to exit from pizza recipe	Positive- sends user to previous page
Test 1	User tries to click on spaghetti photo	Positive- sends user to spaghetti recipe
Test 2	User uses back button to exit from spaghetti recipe	Positive- sends user to previous page
Test 1	User tries to click on cake photo	Positive- sends user to cake recipe
Test 2	User uses back button to exit from cake recipe	Positive- sends user to previous page
Test 1	User tries to click on ramyun photo	Positive- sends user to ramyun recipe
Test 2	User uses back button to exit from ramyun recipe	Positive- sends user to previous page
Test 1	User tries to click on kimchi photo	Positive- sends user to kimchi recipe
Test 2	User uses back button to exit from kimchi soup recipe	Positive- sends user to previous page
Test 3	User clicks recipe name 'Kimchi Soup' on the list page	Positive - sends user to the Kimchi Soup recipe page
Test 3	User clicks recipe name 'Sandwich' on the list page instead of recipe photo	Positive - sends user to the Sandwich recipe page
Test 3	User clicks recipe	Positive - sends user

	name 'Insalata' on the list page instead of recipe photo and then clicked back button	to the Kimchi Soup recipe page and then navigate back to the list page successfully
Test 4	User clicks whitespace between recipes	Negative- Application stops

B.

vii. Updated project plan of the work effort details

A. Sprint 4: Development: User Interface, Java Code Behind, Database, External Integration, Unit Tests *due by August 7*

1. Creating Java and XML code based on the finalized UI layout and database design. Testing the application with Android Studio