Android Application Design Documentation

Jose Manriquez
Patrick Rock
Yaofei Li

December 2013

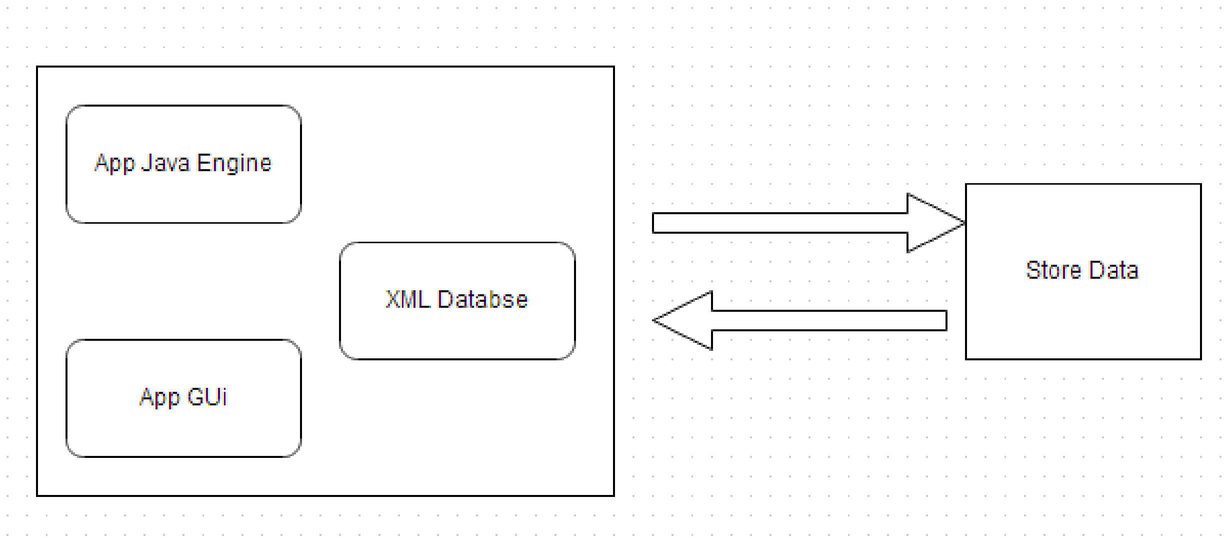
1 Introduction

Android is a mobile operating system used by cell phones and tablets. The purpose of this project is to create an Android application that uses scientific data and stores it according to the application's use. The application will be written in Java using Eclipse, Robotium, and the Android SDK development kit. The project will be broken down into 3 parts: user interface code, core algorithm implementation, and the final version. Furthermore, the project will be completed using a test-driven development approach.

2 Application

The application will be a food and nutrition calculator. The calculator will take in the user's input for each meal every day and then calculate the amount of calories and how healthy the user is eating. The application will store the user's progress each day and report the overall information when the user inquires it. The basic operations are: logging in, adding a food item, viewing the daily total, and viewing a long term progress report.

3 High Level Design

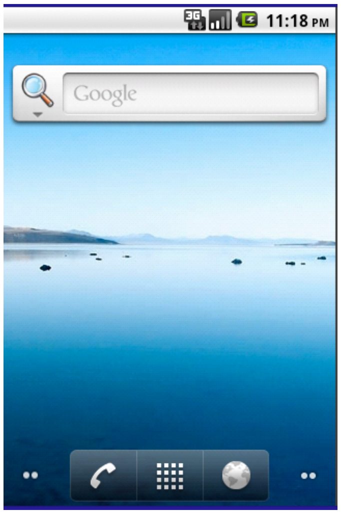


This is a diagram of the high level design. The application will include Java Engine, GUI, and XML database. For the store data, user data will be transferred and stored to the SD card.

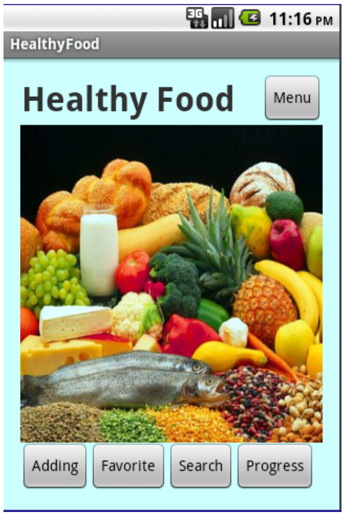
4 Low Level Design

4.1 App GUI

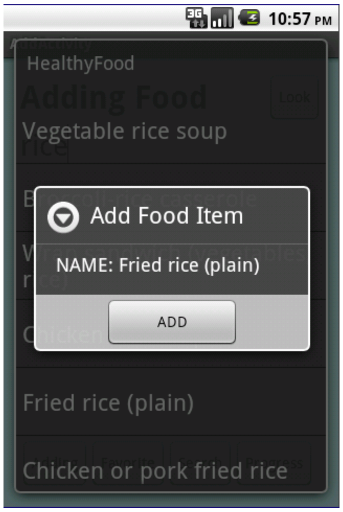
The design of the application will be the following: Main Screen, Adding Screen, Favorites Screen, Progress Screen, Settings Screen, and Search Screen.



Default Android Emulator



Main Screen Menu



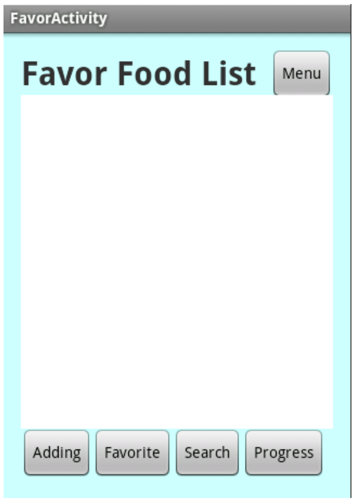
Vegetable rice soup

NAME: Fried rice (plain)

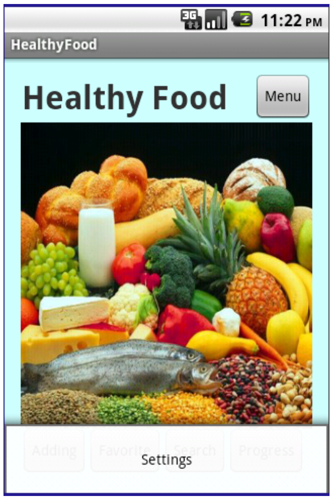
Fried rice (plain)

Chicken or pork fried rice

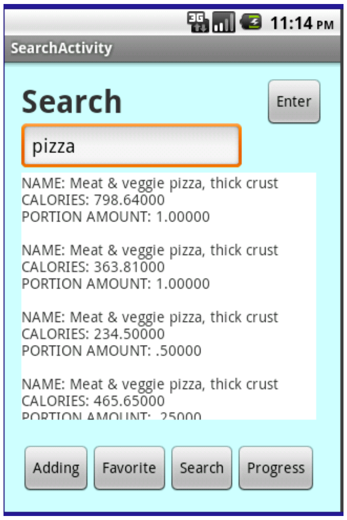
Adding Screen Menu



Favorites Menu Screen



Settings Screen Menu



pizza

```
NAME: Meat & veggie pizza, thick crust
CALORIES: 798.64000
PORTION AMOUNT: 1.00000

NAME: Meat & veggie pizza, thick crust
CALORIES: 363.81000
PORTION AMOUNT: 1.00000

NAME: Meat & veggie pizza, thick crust
CALORIES: 234.50000
PORTION AMOUNT: .50000

NAME: Meat & veggie pizza, thick crust
CALORIES: 465.65000
PORTION AMOUNT: .25000
```

Search Screen Menu

4.2 XML Design

The following is a rough XML table that will be used in the application.

```
    <User>
        <Day_Number>
            <Day_Calorie_Total>
            <Meal_Number>
                <Food_Items></Food_Items>
                <Meal_Calorie></Meal_Calorie>
                <Saturated_Fats></Saturated_Fats>
            </Meal_Number>
        </Day_Number>
    </User>
```

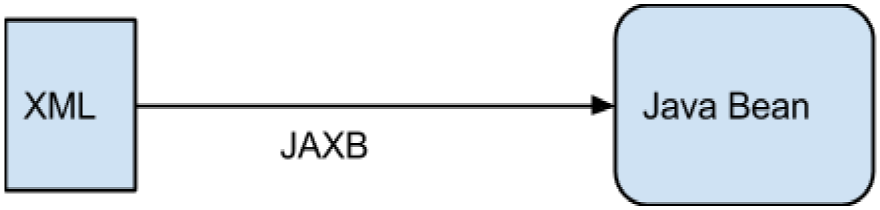
```
  <Food_Code></Food_Code>
    <Display_Name></Display_Name>
         <Portion_Default></Portion_Default>
         <Portion_Amount></Portion_Amount>
         <Portion_Display_Name></Portion_Display_Name>
         <Factor></Factor>
        <Increment></Increment>
         <Multiplier></Multiplier>
         <Grains></Grains>
         <Whole_Grains></Whole_Grains>
         <Vegetables></Vegetables>
         <Orange_Vegetables></Orange_Vegetables>
         <Drkgreen_Vegetables></Drkgreen_Vegetables>
         <Starchy_vegetables></Starchy_vegetables>
         <Other_Vegetables></Other_Vegetables>
         <Fruits></Fruits>
         <Milk></Milk>
         <Meats></Meats>
         <Soy></Soy>
         <Drybeans_Peas></Drybeans_Peas>
         <Oils></Oils>
         <Solid_Fats></Solid_Fats>
         <Added_Sugars></Added_Sugars>
         <Alcohol></Alcohol>
         <Calories></Calories>
         <Saturated_Fats></Saturated_Fats>
</Food_Display_Row>
```

4.3 Data

The data used by the application will be obtain from data.gov which is an official website maintained by the US goverment and contains a detailed list of 4,682 food items. Information like portions, calorie count, and nutritional value will be used to calculate the user's calorie and nutritional intake. The three xml files that will be used in the application are Food_Display_Table, Foods_Needing_Condiments_Table, and lu_Condiment_Food_Table. The data can be found following the URL below.

We will use an XML marshaller to load the data into a Java bean. A bean is an object with private fields, and a public getter and setter for each field. We will use JAXB to map XML to java objects.

http://www.cnpp.usda.gov/Innovations/DataSource/MyFoodapediaData.zip

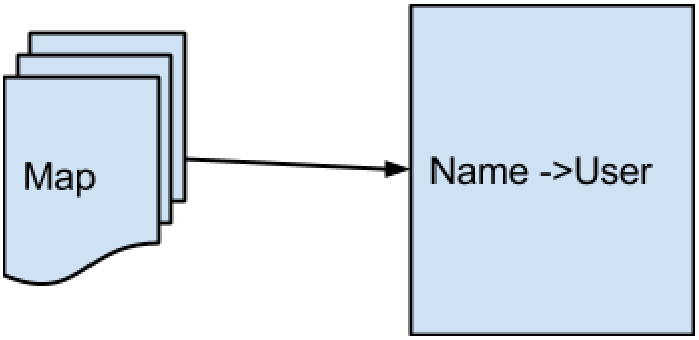


4.4 Object Structure

Each user in the system will be represented with a user object. A user object contains all of the data associated with that user. The Activities responsible for displaying progress and updating eating habits will query and update the user object. A user object contains a list of identifiers that map to food items. These identifiers should be grouped into days. The scientific XML data representing nutrition data should be global and immutable. The activities will query the user for the name of an XML item, and then marshall it into a bean. The application's static state is the scientific XML, which will be an android resource. The applications dynamic state will be encapsulated within the user object.

We will need a range of utilities that interact with user objects. Given a user we should be able to calculate the total caloric intake, the daily average intake, most common food. We can add utilities to generate common statistical metrics such as the mean, median and range. These utilities will be used by the activity responsible for displaying data about the user's eating habits.

To support login we will need to map user-names to user-objects. When a user logs in, the corresponding object is retrieved and transmitted via an intent to the following activities. The login activity will maintain this map. When a user creates a new account, the map will be updated with a new object.



5 Conclusion

The benefits of the project:

- the use of XML for data interchange
- mobile phone application development concepts
- mobile phone application development tools
- test-driven development concepts and practices
- presenting your work to others
- Friendly GUI and easy to use.

Assumptions:

- Android device’s version should between from 2.2 to 4.3
- Android device can connect to internet
- Android device has touch screen
- There is enough memory for the data
- The Android Emulator will correctly run the application
- Assuming we will be able to implement a multi-user system