Towards an Enhanced Shopping Experience: Design and Implementation of a Location-Based Android App

GRADUATE PROJECT PROPOSAL

Submitted to the Faculty of the School of Engineering and Computing Sciences Texas A&M University-Corpus Christi Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

by

Ilker Okan Aker
August 2012

Committee Members

Dr. Ahmed Mahdy
Committee Chairperson

Dr. Scott King
Committee Member

Dr. John Fernandez
Committee Member
ABSTRACT

As technology improves, the world of mobile devices has been continuously progressing. Mobile devices are getting more powerful and functional, providing endless opportunities to create diverse applications in different domains, such as gaming, education, entertainment, health & fitness, transportation, and travel. iOS, Windows Mobile, RIM, and Android are currently the major operating systems used in smartphones. They represent more than 90% of the market. Among these mobile operating systems, Android tops the market with almost a 50% share as of 2012. Android is an open source mobile operating system that uses a Linux kernel. Even though the Android platform is fairly new, numerous applications spanning a wide array of different categories have already been developed for this operating system. This project presents a new location-based shopping application that, to the best of knowledge, has not been developed before. The developed application enhances the overall shopping experience on different levels by providing users with ongoing special deals, locating items of interest, finding the shortest route, and comparing prices with other online stores by scanning the barcode. Users of this app can select the preferred store among other settings. By using this app, users should be able to save time and effort spent on grocery shopping.
# TABLE OF CONTENTS

Abstract ........................................................................................................................................ ii

Table of Contents ....................................................................................................................... iii

List of Figures ............................................................................................................................. iv

1. Background and Rationale ...................................................................................................... 1

2. Narrative .................................................................................................................................. 3

   2.1. Implementation of the Application .................................................................................. 5

       2.1.1. Native Side .............................................................................................................. 5

       2.1.2. Server Side ............................................................................................................ 7

2.2. Database ............................................................................................................................. 8

2.3. Web Server .......................................................................................................................... 10

2.4. Admin Login Panel ............................................................................................................. 12

3. Requirements Definition ....................................................................................................... 14

   3.1. Structured Requirements .............................................................................................. 14

4. System Design ....................................................................................................................... 17

   4.1. Design Rationale ............................................................................................................ 17

   4.2. Use Cases ..................................................................................................................... 18

   4.3. Database Design ............................................................................................................ 24

5. System Implementation ......................................................................................................... 25

6. Evaluation and Results ......................................................................................................... 32

7. Conclusion and Future Works ............................................................................................. 34

Bibliography ............................................................................................................................... 36

Appendix A ................................................................................................................................. 37-89
LIST OF FIGURES

Figure 1. Block Diagram of the Application`s User Side .................................................. 4
Figure 2. ER Diagram of Shopping App DB ................................................................. 9
Figure 3. Representation of Server-Client Communication in the Application ................. 10
Figure 4. Connecting to Web Server via Internet Service ............................................. 11
Figure 5. A Sample Wrapper for MySQL Database ..................................................... 12
Figure 6. Block Diagram of the Application`s Customer Side ..................................... 13
Figure 7. Shopping List Application`s System Design .................................................. 17
Figure 8. Use Case Diagram of Admin ...................................................................... 18
Figure 9. Use Case Diagram of User .......................................................................... 20
Figure 10. Use Case Diagram of Super Admin ............................................................ 23
Figure 11. Database Table List on the Server .............................................................. 24
Figure 12. Home Screen ............................................................................................. 26
Figure 13. Navigation ................................................................................................. 26
Figure 14. Search Screen ........................................................................................... 26
Figure 15. Admin Login Screen .................................................................................. 27
Figure 16. Shopping List Screen .................................................................................. 27
Figure 17. Select Store Screen .................................................................................... 28
Figure 18. Value of the Day Screen ............................................................................. 28
Figure 19. Add Products Screen .................................................................................. 28
Figure 20. Clean My List Screen .................................................................................. 29
Figure 21. Show My List and Total Cost Screen ......................................................... 29
Figure 22. Final Page and Remove Items Screen .................................................................30

Figure 23. Pictures of H-E-B Staples’s Store Map ............................................................32
1. BACKGROUND AND RATIONALE

After great technology improvements, the mobile phone industry has made huge breakthroughs. Today we see mobile devices used everywhere. Smart mobile phones are so popular and widely used all around the world. IOS, Android, BIM and Symbian are the mobile operating systems which make up almost 90% of the app market. Android has been growing rapidly since the time that Android Inc. was purchased by Google in 2007. So many developers went into this field and built their projects for Android.

Human beings have to satisfy many needs by shopping. Shopping is an activity that has to be done in a given period of time. Most of the time regained depends on the number of items that are to be purchased. There are a few other factors which influence the time, such as not being familiar with the store or having a hard time remembering which items need to be purchased. The problem is that not everybody in the world likes to go shopping because of the time it takes. Time became a major problem after large super-markets were built. Even though managers try to organize their items, the logistical complexity has never been solved completely. Unfortunately, there has been no solution to reduce the overall shopping time in super-markets up to now. The intent and purpose of this project is to come up with a solution to solve this problem.

Unfortunately, nobody has built an android application that proposes to solve this issue. Some popular grocery applications exist in the market. One of the authors of Macworld.com Rob Griffiths, tested quite a few grocery applications and narrowed down into three most useful ones for users. They are SplashShopper List Manager, ShopShop or Grocery Gadget Shopping List. He stated that “It’s not feature rich by any measure, but it’s very simple to use and works quite well for handling basic grocery list needs.” [7]
As this quote means, what they all do is basically displaying the shopping list that has been selected by the user. They do not connect to the databases of supermarkets to list the location of searched products. What the proposed application will do;

- Users can search a product in a particular store.
- Users can see value of the day of a particular store.
- Users can add searched product or the value of the day into their shopping list.
- Users can create a shopping list.
- Users can continue from their last shopping list.
- Users can see their shopping list.
- Users can see the total cost of the list and also how many products they have in their list.
- Users can take a picture of something, such as an aisle or a product, to let him/her know about something via e-mail.
- Users can compare prices by scanning the barcode. This feature requires another app (any barcode reader app available in the market) to be installed in the device.
- Users will have to select a store to be able to do all above steps.

There is the web-site extension of this app which helps store managers to insert the products into database. Basically, a manager will be responsible for a particular store. The credentials will be given, so that once logged in to the system; the manager will be able to add products to his store. Once the page is submitted, users will be able to see that store in the app. The main point of this app is to help users reduce their shopping time.
2. NARRATIVE

Since the era of mobile phone technology, more breakthroughs have been made. Information gathering and communication is getting easier thanks to current technology. People with mobile devices are capable of doing what they want just like what they do with their computers. Many applications have been written for mobile devices since the beginning of mobile operating systems. Different needs exist for different people in the world. However, there is a common issue that many people wish to have which is spending less time on a job. That way the time saved can be used on some other jobs. That is the logic behind technology, making peoples` lives easier. The idea of application that will be built comes from the same logic.

This project proposes an application which helps people take advantage of technology in terms of the time they spend on shopping by using Android devices. Many people do not like to go shopping just because of the time it takes. The idea is to build a mobile application to be used in supermarkets. The application is planned to be used for any type of supermarkets all around the world. Since more than 95% of the people in the world use mobile phones, there is definitely a need for such an application. This is the driving force why this proposed application will be created.

The must feature of proposed application will allow users to create their shopping list and then once the user clicks on done button, it will display the locations of searched products by applying the shortest path algorithm for that particular supermarket.
This project consists of two sides. The first side is the user, who uses the application, and the other side is the customer who purchases the application, such as HEB or Wal-Mart. Basically, the customer side will be more dominant as the project requires. Based on the ideas that have been gathered from different people, two flowcharts were drawn for the user and the customer side of the application. Figure 1 below shows the general overview of the user side. It gives a brief idea to understand how the system works.

Figure 1: Block Diagram of the Application`s User Side
This project consists of four parts,

2.1. Implementation of the Application (with JAVA)

   2.1.1. Native Side

   2.1.2. Server Side

2.2. Database (with MySQL)

2.3. Web Server (with Php)

2.4. Admin Login Panel (with Php)

2.1. Implementation of the Application

Due to lack of knowledge in JAVA and the Android SDK, the initial phase was met with complications because of a limited amount of knowledge of apps and codes. Therefore, the plan was to break the app into many small pieces so that it would be easy to implement and debug.

2.1.1. Native Side

The activities that are used in this section;

- **Home Activity.java**: Selected store text, background image, spinner and a search button is shown in this activity.

- **Admin Login Activity.java**: This activity was built for store managers to login and create their stores.

- **Loading Screen Activity.java**: This activity works like an animation page. Instead of seeing a black screen when users click on a button, this activity shows them a “Please Wait” screen.
- **Array Adapter Activity.java:** In this activity, a custom adapter was created for the Select Category page.

- **Search Activity.java:** This activity involves the products’ list to show. It allows you to filter the products in the list by typing a letter.

- **Search Activity 2.java:** In this activity, users see the detailed information such as where the product is in that store, how much the regular price is, how much today’s price is and the name of product. It also asks you if you want to add the item into your shopping list.

- **Select Categories Activity.java:** Users select the categories and then they are directed to the selected categories’ products.

- **Select Store Activity.java:** Users have to select store to be able to do any action in the app. This activity shows the name of available stores in the database.

- **Select Product Activity.java:** In this activity, products are shown based on the selected category.

- **Value of the Day Activity.java:** As it is understood from the name, this activity shows value of the day in a format that has old price, today’s price and the aisle number. It also asks you if you want to add the item into your shopping list.

- **Shopping List Activity.java:** In this activity, users see two buttons, a search button, and a spinner with an image. User either can click on “Create New List” button or “Continue from last Shopping List”. Or else, they can search a product.
- **Show Final Page Activity.java**: This is the activity that the final result is shown after the shortest path algorithm is applied.

- **Show My List Activity.java**: Users can see their shopping list in this activity. Also in this activity, they can see the total cost, if they wish they can remove product(s) from the list.

- CustomHttpClient (This is for connecting the remote DB)

**2.1.2. Server Side**

PHP is used to communicate with the native code (JAVA). The PHP code that has been written is to query to the DB and return the results to my JAVA code. The PHP files that are created are:

- **Db.php**: This is the file where the credentials are stored to connect to the database. Instead of having the credentials in every page, this file was created. Therefore, in each PHP file, this db.php needs to be included. A sample code can be found in the Appendix A.

- **Adminlogin.php**: In this file, the credentials are checked and result is returned by echo. The query can be found in Appendix A.

- **Done.php**: The most important PHP file is this one due to the reason shortest path algorithm is run here based on the string gotten from Show Final Page.java. The query can be found in Appendix A.

- **Search.php**: This file consists of a db.php, as the others, and query. It returns the products that are saved in the database. The query can be found in Appendix A.
- **Search2.php**: This file does not return all the products in the database. Instead, it returns the selected product’s information which are aisle number, product old price and regular price. The query can be found in Appendix A.

- **Selectcategories.php**: As we can understand from the name, this file returns categories. The query can be found in Appendix A.

- **Selectproduct.php**: This file returns products based on selected categories. The query can be found in Appendix A.

- **Selectstore.php**: This file returns available stores in the database. The query can be found in Appendix A.

- **Total_cost.php**: This file returns total cost of the shopping list of a user. The query can be found in Appendix A.

- **Valueoftheday.php**: Every day, only one item is on sale at any store. This file returns the value of the day. The query can be found in Appendix A.

2.2. Database

The app requires a connection to the database where all the product names, category names, product prices, and credentials are stored. The following ER diagram shows the details about the data types and table names in the DB, as well as the relationships between tables.
Figure 2: ER Diagram of Shopping App DB
Figure 3 demonstrates how the Server-Client communication has been created. As well as, it shows what a super admin deals with. He is responsible from maintaining the database on the server.

2.3. Web Server

It is not possible for an android client to talk to database server without having a web server assistance. In this application, a server has been rented from GoDaddy.com to build the web server. The server allowed the application to run using PHP. Required was a submission of an HTTP request to the server from mobile device, thereafter there was a waiting period for the web server to return the value. The code handles the job talking to the database. It returns the value that has been queried. The HTTP request had two parameters. The first one is the URL, and the second one is the parameters that SQL query needed. In the figure below, you can see the executeHttpPost function written in JAVA.
After the PHP web server gets Post parameters from the mobile client, it submits the related SQL query to the MySQL database server and then the results are returned to JAVA. The following code represents a sample PHP code that takes a parameter and returns a result.

```java
public static String executeHttpPost(String url, StringNameValuePair[] postParameters) throws Exception {
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpPost request = new HttpPost(url);
        UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(postParameters);
        request.setEntity(formEntity);
        HttpResponse response = client.execute(request);
        in = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));
        StringBuffer sb = new StringBuffer("\n");
        String line = "\n";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();
        String result = sb.toString();
        return result;
    } finally {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

**Figure 4: Connecting to Web Server via Internet Service**
2.4. Admin Login Panel

An Admin Login module is required for this application; and it has been created specifically for this app. Initially, the admin login module was built internally for the app but based on the feedback, this module was created separately. Store managers need to log into the system and create their store for users to see their stores in the app. Figure 6 shows a general view about the web site of Admin Login Panel.

Figure 5: A Sample Wrapper for MySQL Database
Figure 6: Block Diagram of the Application’s Customer Side

- Functions, if the database already exists for a particular admin
- Functions, if the database does not exist yet for a particular admin
3. REQUIREMENT DEFINITION

This section provides the user requirements in detail and also gives the structural view of those requirements.

- The managers of the stores must login to the system and create their stores by filling out the form online.
- The users must select a store to search an item.
- The users must select a store to create a shopping list.
- The users must select a store to use “Let Admin Know” module.
- The users must select a store to see value of the day.
- The users must have a barcode reader app to be able to use “Compare Prices” module.
- The users must select a store to use “Compare Prices” module.
- The users must be connected to internet.
- The device that this app will be run must have android version 2.3.1 or higher for best performance.
- The device that this app will be run must have a camera to use “Compare Prices” and “Let Admin Know” modules.

3.1 Structured Requirements

1. Shopping List: Users can click on shopping list from navigation.

   1.1. Create New Shopping List: Users can create their shopping list.
       
       1.1.1. Add Items: Users can add an item based on the selected category.
1.1.2. Remove Items: Users can remove an item or items with gesture effects.

1.1.3. Show My List: Users can click on a button and see the current shopping list.

1.1.4. Total Cost: Users can see the total cost of their shopping list.

1.1.5. Total Number of Items: Users can see how many items they have in their list.

1.2. Continue from Last Shopping List: Users can continue from their last shopping list.

1.2.1. Add Items: Users can add an item based on the selected category.

1.2.2. Remove Items: Users can remove an item or items with gesture effects.

1.2.3. Show My List: Users can click on a button and see the current shopping list.

1.2.4. Total Cost: Users can see the total cost of their shopping list.

1.2.5. Total Number of Items: Users can see how many items they have in their list.

2. Value of the Day: Users can select value of the day from navigation. The page shows value of that day in a list.

2.1. Add Item into List: Users can add value of the day item into their shopping list, if they wish.

3. Select Store: Users can select or change the store.

4. Admin Login: This module is available online at
4.1. Login Page: A manager of a store can login to the system with a valid credential.

4.2. Submit Store: A manager needs to fill out the form based on his/her store then submit it.

4.3. Edit Existing Store: Not built yet. However the developer of the app can always edit from server.

4.4. Remove Store: Not built yet. However the developer of the app can always remove the store from server.

5. Let Admin Know: Users can take a picture of an aisle, a product price or anything related to the store and send it via email quickly.

6. Compare Prices: Users can take a picture of a barcode of an item, and check the prices of the same item in another store.
4. SYSTEM DESIGN

This section focuses on the design issues of the app. It includes Design Rationale, Use Cases, and Database Design.

4.1 Design Rationale

![Diagram of Shopping List Application's System Design]

Figure 7: Shopping List Application’s System Design

While designing the system, client-server model was followed as shown in the Figure 4.1. Whenever the user makes a database connection request, the result is brought from the remote server where PHP web server and MySQL DB are located. The app is not available in the market yet. However, it will be uploaded when all the modules are fully tested.
4.2 Use Cases

UML is a must for almost all projects in computer science field. By analyzing these diagrams, the project will be more clear and easy to maintain for any programmer in the future.

Figure 8: Use Case Diagram of Admin

<table>
<thead>
<tr>
<th><strong>TITLE</strong></th>
<th>Log In</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary</strong></td>
<td>Admin of the system types his/her credentials to login</td>
</tr>
<tr>
<td><strong>Actors</strong></td>
<td>Admin</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>Super Admin must give a valid credential to the admin</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Main Successful Scenario</strong></td>
<td>-Admin types his/her credentials in the admin login page and it goes through.</td>
</tr>
<tr>
<td><strong>Alternative Scenarios</strong></td>
<td>-Admin types his/her credentials in the admin login page</td>
</tr>
</tbody>
</table>
and it does not go through.

- Admin may forget his/her username or password.

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Create DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Admin fills out the form online and click on the submit button.</td>
</tr>
<tr>
<td>Actors</td>
<td>Admin</td>
</tr>
<tr>
<td>Preconditions</td>
<td>A successful login to the system is required</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td>Main Successful Scenario</td>
<td>- Admin fills out the form online and click on the submit button. No connection loss occurs.</td>
</tr>
</tbody>
</table>
| Alternative Scenarios | - Internet connection might be down before the admin click on the submit button. - Godaddy server might be down.
Figure 9: Use Case Diagram of User

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Create Shopping List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>User creates his/her shopping list</td>
</tr>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Application has to be installed on the device and selected store must be available.</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td>Main Successful Scenario</td>
<td>-User clicks on the Create New Shopping List button and starts selecting products by clicking related categories.</td>
</tr>
<tr>
<td>Alternative Scenarios</td>
<td>-Application may crash because of the internet speed at a certain time due to the maintenance on Godaddy servers.</td>
</tr>
<tr>
<td></td>
<td>-Very slow wireless internet connection or no internet access on the phone.</td>
</tr>
<tr>
<td>TITLE</td>
<td>Edit Shopping List</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Summary</td>
<td>User edits his/her shopping list</td>
</tr>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Application has to be installed on the device and selected store must be available.</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td><strong>Main Successful Scenario</strong></td>
<td>-User clicks on the Continue from Last Shopping List button and starts editing his/her list.</td>
</tr>
<tr>
<td><strong>Alternative Scenarios</strong></td>
<td>-Application may crash because of the internet speed at a certain time due to the maintenance on Godaddy servers.</td>
</tr>
<tr>
<td></td>
<td>-Very slow wireless internet connection or no internet access on the phone.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Search Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>User can search a product and see the detail about it</td>
</tr>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Application has to be installed on the device and selected store must be available.</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td><strong>Main Successful Scenario</strong></td>
<td>-User clicks on the Search icon image and starts typing the name of searched product. Then user clicks on the product name appears in the list.</td>
</tr>
<tr>
<td><strong>Alternative Scenarios</strong></td>
<td>-Application may crash because of the internet speed at a</td>
</tr>
<tr>
<td>TITLE</td>
<td>View List</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Summary</td>
<td>User can see his/her products in the shopping list.</td>
</tr>
<tr>
<td>Actors</td>
<td>User</td>
</tr>
</tbody>
</table>
| Preconditions | -Application has to be installed on the device and selected store must be available.  
               -There has to be products in the list. |
| Flow of Events: |                   |
| Main Successful Scenario | -User clicks on the Continue from Last Shopping List button and then another click on Show My List button. |
| Alternative Scenarios | -Application may crash because of the internet speed at a certain time due to the maintenance on Godaddy servers.  
                          -Very slow wireless internet connection or no internet access on the phone. |
**Figure 10: Use Case Diagram of Super Admin**

<table>
<thead>
<tr>
<th><strong>TITLE</strong></th>
<th>Give Credentials to the Administrators</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary</strong></td>
<td>Super Admin assigns credentials to the Administrators</td>
</tr>
<tr>
<td><strong>Actors</strong></td>
<td>Super Admin</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>Super Admin must successfully be logged in to the web server.</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Main Successful Scenario</strong></td>
<td>-Super admin assigns usernames and passwords by manually typing in the database.</td>
</tr>
<tr>
<td><strong>Alternative Scenarios</strong></td>
<td>-Super Admin may forget his database server credentials.</td>
</tr>
<tr>
<td></td>
<td>-Godaddy servers might be down for maintenance.</td>
</tr>
</tbody>
</table>
**Maintain App**

**Summary**
Super Admin maintains the application

**Actors**
Super Admin

**Preconditions**
Super Admin must successfully be logged in to the web server.

**Flow of Events:**

**Main Successful Scenario**
- Based on requirement changes, super admin changes some code in the app and updates it.

**Alternative Scenarios**
- The main code of the app might be lost so that application cannot be maintained.

---

**4.3 Database Design**

All the data are stored in MySQL database on GoDaddy.com servers. The application connects to the database to get the data. As of now, database design does not support for all stores to be shown in the app. The existing data is just a test data. It can be enhanced to support all stores by changing the design.

![Database Table List on the Server](image)

*Figure 11: Database Table List on the Server*
5. SYSTEM IMPLEMENTATION

The code has been written in Android and 27 different “.java” files with the total number of lines of code are almost 3000. The project has been created by using Eclipse IDE with Android SDK. On the server side, PHP has been used and there are 16 different files with nearly 1000 lines of code. Overall, almost 4000 lines of code took two months to write excluding testing.

As the operating system, Windows 7 was chosen. To make sure there is no lack of hardware, the app was built on my personal computer which has 8 GB of RAM with Intel Core I7 and a 2GB of ATI GPU. For the server side of the app, the server was rented from Godaddy.com which provides a high quality and fast connection. The app is made up for mainly seven parts including the home screen.

- Home
- Shopping List
- Value of the Day
- Search
- Admin Login
- Let Admin Know
- Compare Prices
Figure 12: Home Screen

Figure 13: Navigation

Figure 14: Search Screen
Figure 15: Admin Login Screen

Figure 16: Shopping List Screen
Figure 17: Select Store Screen

Figure 18: Value of the Day Screen

Figure 19: Add Products Screen
Figure 20: Clean My List Screen

Figure 21: Show My List and Total Cost Screen
In the app, all screens have been built in such a way that look simple and user friendly. As it is seen in the figures above, buttons and interface are designed as simple and useful. When a user first downloads the app, s/he does not have to do signing up. It will be good to go if the store s/he is looking for is available, created by the store manager on the server.

In figure 16, shopping list screen, three buttons were placed at the bottom of ListView, so that the user does not need to click on the menu button each and every time to see those buttons.

Not all screens but some of them also work on landscape mode. The reason why this feature was restricted for all screens is because background image had to be a fixed size on the screen. However, this was not possible with two buttons in Shopping List Screen.
In some screens, users see the Android title bar remains. For instance, Figure 16, see the differences between two screens. Android has inbuilt activities such as ACTION_CALL for calling and ACTION_SEND for email. Since the current emulator did provide neither the camera nor the email activity, all screenshots could not be taken to show here.
6. EVALUATION AND RESULTS

First of all, all the product data that have been put for test purposes are real. It has been gathered from H-E-B Staples. After a professional explanation of this project to the store managers, they did not see any security issue to take a picture of the stores` boards where all the category names are written. A small map of the store was also provided. The real pictures that were taken in the store can be seen in the Figure 19 below.

![Figure 23: Pictures of H-E-B Staples’s Store Map](image-url)
After the test data were uploaded to the server, many tests were made to find bugs in the app. As well as that, different people tested the app on variety of mobile devices. Quite a few feedbacks were provided. Based on those feedbacks, some features were added or removed from the app and made it more useful.

The most important function of the app is the one that returns the result at the final page. It is called shortest path finder. It basically calculates the result based on the priority of each product selected in the list. As a reminder, just to make it clear, the app does not show the shortest path to reach product depending on user’s location in the store. It always calculates it as if users start their shopping from the main entrance of the store. It has been also assumed that there is only one entrance of any store. Otherwise, it would be so difficult to implement the algorithm. This app was built as the prototype and worked just perfect in the store H-E-B Staples.
7. CONCLUSION AND FUTURE WORK

Android is a new platform that keeps growing every year. Too many developers have been creating applications for Android devices. There are some reasons behind why it became so popular on mobile devices rapidly. The main reason is because it is open-source. The idea of writing code for an open-source platform came so nice to developers. That way money is not required to either build an application or upload it to the market. Another reason is that it provides the capability of using all available sources of the phone. Therefore, developers like to use these capabilities as much as they can. Last but not least, it gives high performance. The application has a user-friendly interface so that it can be used easily by the customers. It is mainly separated by two sides. The first one is the user side, and the second one is the customer side. Due to the nature of the project, customer side is the most dominant part. This paper presented an application proposal for Android devices such that it helps customers to reduce the overall shopping time when they go to supermarkets. In another word, the main purpose of building this application is to help people in their daily lives in terms of the time they spend finding items in supermarkets.

The first task at the top of future work is that applying the app to work for all supermarkets in the world. It is not challenging at all as it seems. What is needed to be done is adapting the SQL queries with a little bit change in the design of the database. Second task would be the performance improvement. As of now, every time the user wants to get data, app requests a new connection. Instead of that, requesting connection of database only once when the user opens up the app would be a better way in terms of the performance improvement.
Another future work might be increasing the image quality. As of now, the images that have been chosen are not high quality. It can be realized when the app is opened up on a big screen Android device such as a tablet. Last but not least, the admin login panel works only to add store. It needs to be enhanced to remove and edit stores.
BIBLIOGRAPHY


<http://www.macworld.com/appguide/article.html?article=136247> 
Last visited: August 5, 2012.
APPENDIX A

DB.php

```php
<?php
   //connect to the db
   $mysql_hostname = "50.63.244...";
   $mysql_user = "ilkerokanaker";
   $mysql_password = ".......";
   $mysql_database = "ilkerokanaker";
   $bd = mysql_connect($mysql_hostname, $mysql_user, $mysql_password)
       or die("Opps some thing went wrong");
   mysql_select_db($mysql_database, $bd) or die("Opps some thing went wrong");
?>
```

Adminlogin.php

```sql
SELECT admin_id,admin_status
FROM admin
WHERE admin_name= '$u_name' AND admin_password = '$u_password'
```

Done.php

```sql
SELECT a.aisle_priority, p.product_name
FROM aisle a, product p
WHERE a.aisle_id=p.aisle_id
AND p.product_name IN (" . $p_name . ")
ORDER BY a.aisle_priority
```

Search.php

```sql
SELECT product_name
FROM product
ORDER BY product_name ASC
```

Search2.php

```sql
SELECT product_name, product_regular_price, aisle_id
FROM product
WHERE product_name= '$u_item'
```
Selectcategories.php

    SELECT category_name
    FROM category
    ORDER BY category_name ASC

Selectproduct.php

    SELECT product_name
    FROM product
    WHERE category_id='c_id'
    ORDER BY product_name ASC

Selectstore.php

    SELECT store_name
    FROM store
    ORDER BY store_id

Total_cost.php

    SELECT SUM( product_regular_price )
    FROM product
    WHERE product_name IN ('$p_name'.)

Valueoftheday.php

    SELECT product_name, product_new_price, product_regular_price, aisle_id
    FROM product
    WHERE value_of_the_day='$u_day'

AdminLoginActivity.java

package com.okan;

import java.util.ArrayList;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.AdapterView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
public class AdminLoginActivity extends Activity implements LocationListener{
    EditText edt,name,password;
    String un,un2;
    Button connect;
    TextView tx;
    String[] tokens;
    Spinner snp;
    boolean flag=true;
    ArrayAdapter <CharSequence> adapter;
    Uri mCapturedImageURI;

    private static final int CAMERA_PIC_REQUEST = 1337;
    private LocationManager locationManager;
    private String provider;
    float lat;
    float lng;

    public void myDatabaseConnection()
    {
        ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
postParameters.add(new BasicNameValuePair("username", name.getText().toString().trim()));
postParameters.add(new BasicNameValuePair("password", password.getText().toString().trim()));

String response = null;
try {
    response = CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/okan_final_project/adminlogin.php", postParameters);
    //JSON not used. Simple string parsing
    String res = response.toString();
    res = res.replaceAll(" ", "");
    res = res.replaceAll("PARSE_", " ");
    res = res.replaceAll("\n", "");
    String delims = "[ ]";
    tokens = res.split(delims);
} catch (Exception e) {
    // TODO: handle exception
}
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.adminlogin);
edt = (EditText) findViewById(R.id.editText1);
edt.setSingleLine();
name = (EditText) findViewById(R.id.editText2);
password = (EditText) findViewById(R.id.editText3);
connect = (Button) findViewById(R.id.button1);
tx = (TextView) findViewById(R.id.textView5);
LoadPreferences();
if(un2.equalsIgnoreCase(""))
    flag = false;
else
    {
        if(un2.equalsIgnoreCase("None"))
            flag = false;
        else
            
40
flag =true;
}
edt.setOnClickListener(new OnClickListner() {

public void onClick(View v) {

Intent fullscreen = new Intent(AdminLoginActivity.this, SearchTestActivity.class);
startActivity(fullscreen);
});
snp = (Spinner) findViewById (R.id.spinnerTest1);
adapter = ArrayAdapter.createFromResource(this, R.array.spinnerTestArray,
android.R.layout.simple_spinner_item);
adapter.setDropDownViewResource(
android.R.layout.simple_spinner_dropdown_item);
snp.setAdapter(adapter);
int position = adapter.getPosition("Admin Login");
snp.setSelection(position);
snp.setOnItemSelectedListener(new MyOnItemSelectedListener());
connect.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
myDatabaseConnection();
SavePreferences("admin_id", tokens[0]);
try{
if(tokens[1].equals("active"))
{

tx.setText("This admin`s status is: active ");
//ONCEDEN GIRIS YAPTIĞI İÇİN BURDA
EDIT OLUNACAK SAYFAYA YONLendirceZ
}
else
{

tx.setText("This admin`s status is: passive ");

Intent intent=new Intent(AdminLoginActivity.this,LoadingScreen_6.class);
startActivity(intent);
}
}
catch (Exception e) {
    // TODO: handle exception
    tx.setText("Check your credentials");
}

// Get the location manager
locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
Criteria criteria = new Criteria();
provider = locationManager.getBestProvider(criteria, false);
//provider = locationManager.GPS_PROVIDER;
Location location = locationManager.getLastKnownLocation(provider);
//Location location = locationManager.getLastKnownLocation(provider);

// Initialize the location fields
if (location != null) {
    //System.out.println("Provider " + provider + " has been selected.");
    lat = (float) (location.getLatitude());
    lng = (float) (location.getLongitude());
} else {
    //Toast.makeText(PictureDemo.this, "Provider not selected"
    ,Toast.LENGTH_LONG).show();
}

@override
public void onResume() {
    super.onResume();
    int position = adapter.getPosition("Admin Login");
    snp.setSelection(position);
    locationManager.requestLocationUpdates(provider, 30000, 5, this);
}

@override
public void onPause() {
    super.onPause();
    locationManager.removeUpdates(this);
}
public class MyOnItemSelectedListener implements OnItemSelectedListener {
    
    public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
        if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Home"))
        {
            SavePreferences("pos", parent.getItemAtPosition(pos).toString());
            Intent intent=new Intent(AdminLoginActivity.this,FinalProjectActivity.class);
            startActivity(intent);
        }
        else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Shopping List"))
        {
            SavePreferences("pos", parent.getItemAtPosition(pos).toString());
            Intent intent=new Intent(AdminLoginActivity.this,ShoppingListActivity.class);
            startActivity(intent);
        }
        else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Value of the Day"))
        {
            if(flag){
                SavePreferences("pos", parent.getItemAtPosition(pos).toString());
                Intent intent=new Intent(AdminLoginActivity.this,LoadingScreen_2.class);
                startActivity(intent);
            } else{
                Toast.makeText(getApplicationContext(), "Select a Store First", Toast.LENGTH_SHORT).show();
                int position = adapter.getPosition("Select Store");
                snp.setSelection(position);
            }
        }
        else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Select Store"))
        {
            SavePreferences("pos", parent.getItemAtPosition(pos).toString());
            Intent intent=new Intent(AdminLoginActivity.this,LoadingScreen_5.class);
            startActivity(intent);
        }
    }
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Let Admin Know"))
{
    String fileName = "temp.jpg";
    ContentValues values = new ContentValues();
    values.put(MediaStore.Images.Media.TITLE, fileName);
    mCapturedImageURI = getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);

    if(flag)
    {
        SavePreferences("shopping", un2);
    }
    else
    {
        Toast.makeText(getApplicationContext(), "Select a Store First", Toast.LENGTH_SHORT).show();
        int position = adapter.getPosition("Admin Login");
        snp.setSelection(position);
    }
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Admin Login"))
{
    //edt.setText("Admin Login");
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Compare Prices"))
{
    try {
        Intent qr = new Intent("com.google.zxing.client.android.SCAN");
        startActivityForResult(qr, 0);
    } catch (Exception e) {

        }
AlertDialog.Builder builder = new AlertDialog.Builder(AdminLoginActivity.this);
builder.setMessage("You must have a barcode reader app to start this activity!"").setCancelable(false).setPositiveButton("Okay", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        dialog.cancel();
    }
});
AlertDialog alert = builder.create();
alert.show();

else {
edt.setText("Failure");
}
}

public void onNothingSelected(AdapterView parent){
    //Do nothing
}
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode != RESULT_OK) {
        return;
    }

    switch (requestCode) {
    case CAMERA_PIC_REQUEST:
    Toast.makeText(AdminLoginActivity.this, "Picture is attached to your email", Toast.LENGTH_SHORT).show();
    Intent i = new Intent(Intent.ACTION_SEND);
    i.setType("text/plain");
    i.setType("image/png");
    }
i.putExtra(Intent.EXTRA_STREAM, mCapturedImageURI);
i.putExtra(Intent.EXTRA_EMAIL, new String[] {"ilkerokanaker@hotmail.com"});
i.putExtra(Intent.EXTRA_SUBJECT, "Topic: " + un2);
String message = String.format("\n\nThe Location: \n Longitude: %1$s \n Latitude: %2$s",
    lng, lat);
String message2 = "\n\nhttp://maps.google.com/maps?q="+lat+","+lng +
message;
i.putExtra(Intent.EXTRA_TEXT, message2);

final PackageManager pm = getPackageManager();
final List<ResolveInfo> matches = pm.queryIntentActivities(i, 0);
ResolveInfo best = null;
for (final ResolveInfo info : matches)
    if (info.activityInfo.packageName.endsWith(".gm") ||
        info.activityInfo.name.toLowerCase().contains("gmail")) best = info;
if (best != null)
    i.setClassName(best.activityInfo.packageName, best.activityInfo.name);
try {
    startActivity(i);
    //startActivity(Intent.createChooser(i, "Send Via.."));
}
} catch (android.content.ActivityNotFoundException ex) {
    Toast.makeText(AdminLoginActivity.this, "There are no email clients installed.", Toast.LENGTH_SHORT).show();
}
break;

default:
    break;
}

public void onLocationChanged(Location location) {
    lat = (float)(location.getLatitude());
    lng = (float)(location.getLongitude());
}

public void onProviderDisabled(String provider) {
    //Toast.makeText(this, "Disabled provider " +
    provider, Toast.LENGTH_SHORT).show();
}
public void onProviderEnabled(String provider) {
    //Toast.makeText(this, "Enabled new provider " + provider, Toast.LENGTH_SHORT).show();
}

public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}

private void SavePreferences(String key, String value){
    SharedPreferences sharedPreferences =getSharedPreferences("mypref", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

private void LoadPreferences(){
    SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
    un= sharedPreferences.getString("pos", "");
    un2= sharedPreferences.getString("pos2", "");
}

FinalProjectActivity.java

package com.okan;

import java.util.List;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.graphics.Bitmap;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Window;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class FinalProjectActivity extends Activity implements LocationListener{

    Spinner snp;
    EditText edt;
    String un;
    TextView tx;
    boolean flag=true;
    ArrayAdapter <CharSequence> adapter;
    Bitmap screenshot;

    Button search_button;

    Uri mCapturedImageURI;

    private static final int CAMERA_PIC_REQUEST = 1337;
    private LocationManager locationManager;
    private String provider;
    float lat;
    float lng;

    Intent intent;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.main);
        LoadPreferences();
    }

    // Other methods...
}
search_button = (Button) findViewById(R.id.search_button);

tx = (TextView) findViewById(R.id.textView2);
snp = (Spinner) findViewById(R.id.spinnerTest1);

if(un.equalsIgnoreCase("")){
    tx.setText(" Please select a store first");
    flag = false;
}
else{
    tx.setText(" Selected Store: " +un);
    if(un.equalsIgnoreCase("None"))
        flag = false;
    else
        flag = true;
}

search_button.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {

        if(flag)
        {
            Intent fullscreen = new Intent(FinalProjectActivity.this,
            LoadingScreen_4.class);
            startActivity(fullscreen);
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Select a Store First",
            Toast.LENGTH_SHORT).show();
            Intent fullscreen = new Intent(FinalProjectActivity.this,
            LoadingScreen_5.class);
            startActivity(fullscreen);
        }
    }
});

adapter = ArrayAdapter.createFromResource(this, R.array.spinnerTestArray,
    android.R.layout.simple_spinner_item);
adaptor.setDropDownViewResource(  
    android.R.layout.simple_spinner_dropdown_item);
snp.setAdapter(adapter);
int position = adapter.getPosition("Home");
snp.setSelection(position);
snp.setOnItemSelectedListener(new MyOnItemSelectedListener());
// Get the location manager
locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

Criteria criteria = new Criteria();
provider = locationManager.getBestProvider(criteria, false);
//provider = locationManager.GPS_PROVIDER;

Location location = locationManager.getLastKnownLocation(provider);
//Location location = locationManager.getLastKnownLocation(provider);

// Initialize the location fields
if (location != null) {
    //System.out.println("Provider " + provider + " has been selected.");
    lat = (float) (location.getLatitude());
    lng = (float) (location.getLongitude());

} else {
    //Toast.makeText(PictureDemo.this, "Provider not selected"
    ,Toast.LENGTH_LONG).show();
}

// @Override
// public void onBackPressed() {
//     AlertDialog.Builder builder = new AlertDialog.Builder(this);
//     builder.setMessage("Do you really want to quit?")
//        .setCancelable(false)
//        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
//        public void onClick(DialogInterface dialog, int id) {
//            System.exit(0);
//        }
//        })
//        .setNegativeButton("No", new DialogInterface.OnClickListener() {
//        public void onClick(DialogInterface dialog, int id) {
//            dialog.cancel();
//        }
//        });
//}
// AlertDialog alert = builder.create();
// alert.show();
//
//@Override
public void onResume(){
    super.onResume();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
    locationManager.requestLocationUpdates(provider, 30000, 5, this);
}

//@Override
public void onPause() { 
    super.onPause();
    locationManager.removeUpdates(this);
}

//@Override
public void onDestroy() {
    super.onDestroy();
}

public class MyOnItemSelectedListener implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent, View view, int pos, 
    long id)
    {
        if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Home"))
        {
            //edt.setText("HOME");
        }
        else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Shopping List"))
        {
            SavePreferences("pos", parent.getItemAtPosition(pos).toString());
            Intent intent=new Intent(FinalProjectActivity.this,ShoppingListActivity.class);
            startActivity(intent);
        } 
        else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Value of the Day"))
        {
            if(flag){
                
            }
        }
    }
}
}
SavePreferences("pos",
parent.getItemAtPosition(pos).toString());
    Intent intent=new
Intent(FinalProjectActivity.this,LoadingScreen_2.class);
    startActivity(intent);
}
else{
    Toast.makeText(getApplicationContext(), "Select a Store First",
Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Select Store");
    snp.setSelection(position);
}
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Select Store"))
{
    SavePreferences("pos", parent.getItemAtPosition(pos).toString());
    Intent fullscreen = new Intent(FinalProjectActivity.this,
LoadingScreen_5.class);
    startActivity(fullscreen);
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Admin Login"))
{
    SavePreferences("pos", parent.getItemAtPosition(pos).toString());
    Intent intent=甦new Intent(FinalProjectActivity.this,AdminLoginActivity.class);
    startActivity(intent);
}
else if(parent.getItemAtPosition(pos).toString().equalsIgnoreCase("Let Admin Know"))
{
    //====================================
    String fileName = "temp.jpg";
    ContentValues values = new ContentValues();
    values.put(MediaStore.Images.Media.TITLE, fileName);
    mCapturedImageURI =
getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
values);
    //====================================
    //====================================
    //====================================
    if(flag)
    {
        SavePreferences("shopping", un);
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

52
intent.putExtra(MediaStore.EXTRA_OUTPUT, mCapturedImageURI);
startActivityForResult(intent, CAMERA_PIC_REQUEST);
} else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
} else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);
}
else {
    Toast.makeText(getApplicationContext(), "Select a Store First",
    Toast.LENGTH_SHORT).show();
    int position = adapter.getPosition("Home");
    snp.setSelection(position);

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 0) {
        //COMPARE PRICES
        if (resultCode == RESULT_OK) {
            String contents = data.getStringExtra("SCAN_RESULT");
            String format = data.getStringExtra("SCAN_RESULT_FORMAT");
            // Handle successful scan
            Toast.makeText(FinalProjectActivity.this, contents, Toast.LENGTH_SHORT).show();
        }
    }

    if (resultCode != RESULT_OK) {
        Toast.makeText(FinalProjectActivity.this, "Resim cekilemedi", Toast.LENGTH_SHORT).show();
        return;
    }

    switch (requestCode) {
    case CAMERA_PIC_REQUEST:
        Toast.makeText(FinalProjectActivity.this, "Picture is attached to your email", Toast.LENGTH_SHORT).show();
        Intent i = new Intent(Intent.ACTION_SEND);
        i.setType("text/plain");
        i.setType("image/png");
        i.putExtra(Intent.EXTRA_STREAM, mCapturedImageURI);
        i.putExtra(Intent.EXTRA_EMAIL, new String[]{"ilkerokanaker@hotmail.com"});
        i.putExtra(Intent.EXTRA_SUBJECT, "Topic: " + un);
        String message = String.format("The Location: 
Longitude: %1$s 
Latitude: %2$s", lng, lat);
        String message2 = "http://maps.google.com/maps?q=+%1$s+,+%2$s";
        i.putExtra(Intent.EXTRA_TEXT, message2);
        i.putExtra(Intent.EXTRA_EMAIL, new String[]{"ilkerokanaker@hotmail.com"});
        i.putExtra(Intent.EXTRA_SUBJECT, "Topic: " + un);
        String message = String.format("The Location: 
Longitude: %1$s 
Latitude: %2$s", lng, lat);
        String message2 = "http://maps.google.com/maps?q=+%1$s+,+%2$s";
        i.putExtra(Intent.EXTRA_TEXT, message2);
    }
final PackageManager pm = getPackageManager();
final List<ResolveInfo> matches = pm.queryIntentActivities(i, 0);
ResolveInfo best = null;
for (final ResolveInfo info : matches)
    if (info.activityInfo.packageName.endsWith(".gm") ||
        info.activityInfo.name.toLowerCase().contains("gmail")) best = info;
if (best != null)
    i.setClassName(best.activityInfo.packageName, best.activityInfo.name);
    try {
        startActivity(i);
        //startActivity(Intent.createChooser(i, "Send Via.."));
    } catch (android.content.ActivityNotFoundException ex) {
        Toast.makeText(FinalProjectActivity.this, "There are no email clients installed.", Toast.LENGTH_SHORT).show();
    }
    break;
    default:
        break;
    }
public void onLocationChanged(Location location) {
    String message = String.format("New Location
Longitude: %1$s
Latitude: %2$s", location.getLongitude(), location.getLatitude());
    Toast.makeText(PoliceDepartmentActivity_3.this, message, Toast.LENGTH_LONG).show();
    lat = (float)(location.getLatitude());
    lng = (float)(location.getLongitude());
}
public void onProviderDisabled(String provider) {
    //Toast.makeText(this, "Disabled provider " + provider,Toast.LENGTH_SHORT).show();
}
public void onProviderEnabled(String provider) {
    //Toast.makeText(this, "Enabled new provider " + provider,Toast.LENGTH_SHORT).show();
}
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}

private void SavePreferences(String key, String value) {
    SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

private void LoadPreferences() {
    SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
    un = sharedPreferences.getString("pos2", "");
}

MySimpleArrayAdapter.java

package com.okan;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
public class MySimpleArrayAdapter extends ArrayAdapter<String> {
    private final Context context;
    private final String[] values;
    public MySimpleArrayAdapter(Context context, String[] values) {
        super(context, R.layout.selectcategories, values);
        this.context = context;
        this.values = values;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

    }
}
LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
View rowView = inflater.inflate(R.layout.selectcategories, parent, false);
TextView textView = (TextView) rowView.findViewById(R.id.label);
ImageView imageView = (ImageView)
rowView.findViewById(R.id.icon);
textView.setText(values[position]);
imageView.setImageResource(R.drawable.arrow);
return rowView;

SearchTestActivity.java

package com.okan;

import java.util.ArrayList;
import org.apache.http.NameValuePair;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
public class SearchTestActivity extends Activity {
  /** Called when the activity is first created. */
  private ListView lv1;
  private EditText ed;
  private ArrayList<String> arr_sort= new ArrayList<String>();
  int textlength=0;
  String[] tokens;
  String selectedFromList;
  public void myDatabaseConnection()
  {
ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
String response = null;

try {

    response = CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/okan_final_project/search.php", postParameters);

    // JSON not used. Simple string parsing
    String res = response.toString();
    res = res.replaceAll(" ", "");
    res = res.replaceAll("PARSE_", " ");
    res = res.replaceAll("\n","");
    String delims = "[ ]";
    tokens = res.split(delims);

} catch (Exception e) {
    // TODO: handle exception
}

@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.searchscreen);
    lv1 = (ListView) findViewById(R.id.ListView01);
    ed = (EditText) findViewById(R.id.EditText01);
    ed.setSingleLine();
    myDatabaseConnection();

    getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);

    lv1.setAdapter(new ArrayAdapter<String>(SearchTestActivity.this, android.R.layout.simple_list_item_1, tokens));
    ed.addTextChangedListener(new TextWatcher() {

        public void afterTextChanged(Editable s) {

        }
    })

}
public void beforeTextChanged(CharSequence s, int start, int count, int after) {
}

public void onTextChanged(CharSequence s, int start, int before, int count) {
    textlength = ed.getText().length();
    arr_sort.clear();
    for (int i = 0; i < tokens.length; i++)
        if (textlength <= tokens[i].length())
            if (ed.getText().toString().equalsIgnoreCase((String) tokens[i].subSequence(0, textlength)))
                arr_sort.add(tokens[i]);

    lv1.setAdapter(new ArrayAdapter<String>(SearchTestActivity.this, android.R.layout.simple_list_item_1, arr_sort));
}

lv1.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        selectedFromList = (String) (lv1.getItemAtPosition(position));
        //Toast.makeText(getApplicationContext(), selectedFromList, Toast.LENGTH_LONG).show();
        SavePreferences("pos3", selectedFromList);
        //SavePreferences("last_visited", "search_1");
        Intent intent = new Intent(SearchTestActivity.this, SearchTestActivity_2.class);
        startActivity(intent);
    }
});
@Override
public void onBackPressed() {
    Intent intent=new Intent(SearchTestActivity.this,FinalProjectActivity.class);
    startActivity(intent);
}

private void SavePreferences(String key, String value){
    SharedPreferences sharedPreferences =getSharedPreferences("mypref", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

SelectCategoriesActivity.java

package com.okan;

import java.util.ArrayList;
import org.apache.http.NameValuePair;
import android.app.AlertDialog;
import android.app.ListActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class SelectCategoriesActivity extends ListActivity {
    String[] tokens;
    String selectedFromList_2,
    str_from_products,
    str_from_categories,
cleared_list_from_categories,
str_from_MyList;

String added_item;
String last_visited_activity;
String str_in_show_my_list;
String flag_deletion_status;
String clicked_button_in_shopping_list;

Button show_my_shopping_list;
Button clean_my_list;
Button done;

private ListView list;
boolean clear_list_flag=false;

ArrayList<String> listItems=new ArrayList<String>();
ArrayAdapter<String> adapter;

private static final int SHOW_MY_SHOPPING_LIST = Menu.FIRST ;
private static final int CLEAN_LIST = Menu.FIRST +1;
private static final int DONE = Menu.FIRST +2;
private static final int GO_HOME = Menu.FIRST ;

public void myDatabaseConnection()
{
    ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();

    String response = null;

    try {
        response=CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/oka
un_final_project/selectcategories.php", postParameters);

        String res = response.toString();
        res = res.replaceAll(" ","");
        res = res.replaceAll("PARSE_"," ");
        res = res.replaceAll("\n"," ");
        String delims = "[ ]";
        tokens = res.split(delims);

    }catch (Exception e) {
        Toast.makeText(SelectCategoriesActivity.this, "Hosting Server is
not responding" ,Toast.LENGTH_LONG).show();
}
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.select_categories_new);
    LoadPreferences();

    setTitle("Select a Category to See Products");
    myDatabaseConnection();

    show_my_shopping_list = (Button) findViewById(R.id.show_my_list);
    clean_my_list = (Button) findViewById(R.id.clean_my_list);
    done = (Button) findViewById(R.id.done);

    //------------------------------------------------------------
    show_my_shopping_list.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            SavePreferences("str_from_products_in_categories",
            str_from_products);
            Intent intent2=new Intent(SelectCategoriesActivity.this,ShowMyList.class);
            startActivity(intent2);
        }
    });
    //------------------------------------------------------------
    clean_my_list.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            AlertDialog.Builder builder = new AlertDialog.Builder(SelectCategoriesActivity.this);
            builder.setMessage("Do you really want to clean?")
            .setCancelable(false)
            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    Toast.makeText(getApplicationContext(), "Cleaned", Toast.LENGTH_SHORT).show();
                    clear_list_flag=true;
                    SavePreferences("cleared_list_from_categories", "true");
                }
            });
        }
    });
SavePreferences("str_from_MyList", "empty");

str_from_products=""

})
).setNegativeButton("No", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}
});
AlertDialog alert = builder.create();
alert.show();

//-------------------------------------------------------------
done.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

str_from_categories = str_from_products;
SavePreferences("str_from_categories", str_from_categories);

Intent intent3 = new Intent(SelectCategoriesActivity.this, ShowFinalPage.class);
startActivity(intent3);

}
});
//-------------------------------------------------------------

list = getListView();
MySimpleArrayAdapter adapter = new MySimpleArrayAdapter(this, tokens);
setListAdapter(adapter);

list.setOnItemClickListener(new OnItemClickListener() {

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

selectedFromList_2 = (String) (list.getItemAtPosition(position));
SavePreferences("selected_category", selectedFromList_2);

if(clear_list_flag)
    SavePreferences("cleared_list_from_categories", "true");


else
    SavePreferences("cleared_list_from_categories", "false");

Intent intent=new Intent(SelectCategoriesActivity.this,SelectProductActivity.class);
    startActivity(intent);

}

});

});

/*@Override
    public void onResume(){
        super.onResume();

        LoadPreferences();

            if(str_from_MyList.equalsIgnoreCase("empty"))
            {
                if(!last_visited_activity.equalsIgnoreCase("search") ||
                    !last_visited_activity.equalsIgnoreCase("value_of_the_day"))
                    {
                        str_from_products="";
                        SavePreferences("cleared_list_from_categories", "true");
                    }
            }

            //===================================================================

            if(last_visited_activity.equalsIgnoreCase("search") ||
                last_visited_activity.equalsIgnoreCase("value_of_the_day"))
            {
                if(!str_from_products.contains(added_item))
                    {
                        str_from_products= str_from_products + added_item + ";";
                        SavePreferences("str_from_products_in_categories", str_from_products);
                        Toast.makeText(getApplicationContext(), added_item+ " added into your list", Toast.LENGTH_SHORT).show();
                    }
                else
                    {
                        Toast.makeText(getApplicationContext(), added_item+ " is already in your list", Toast.LENGTH_LONG).show();
                        Intent intent=new Intent(SelectCategoriesActivity.this,SearchTestActivity.class); //========yeni ekledim
startActivity(intent);

//Toast.makeText(getApplicationContext(), "En son Search`ten geldim", Toast.LENGTH_SHORT).show();

else if(last_visited_activity.equalsIgnoreCase("product"))
{
    SavePreferences("str_from_products_in_categories", str_from_products);
    //Toast.makeText(getApplicationContext(), "En son Product`tan geldim", Toast.LENGTH_SHORT).show();
}
else if(last_visited_activity.equalsIgnoreCase("shopping_list"))
{
    if(clicked_button_in_shopping_list.equalsIgnoreCase("create_new_list"))
    {
        str_from_products = "";
        clear_list_flag=true;
    }
    else
    {
        //nothing
    }
}
else if(last_visited_activity.equalsIgnoreCase("show_my_list"))
{
    str_from_products = str_in_show_my_list;
    SavePreferences("str_from_products", str_from_products);
    SavePreferences("str_from_products_in_categories", str_from_products);
    //Toast.makeText(getApplicationContext(), "En son Show My List`ten geldim", Toast.LENGTH_SHORT).show();
    if(flag_deletion_status.equalsIgnoreCase("true"))
    {
        Intent intent=new Intent(SelectCategoriesActivity.this,ShowMyList.class);
        startActivity(intent);
    }
}

//===================================
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    //    menu.add(0, SHOW_MY_SHOPPING_LIST,0,"Show My Shopping List");  
    //    menu.add(1, CLEAN_LIST,1,"Clean My List");  
    //    menu.add(2, DONE,2,"DONE");  
    menu.add(0, GO_HOME,0,"Go to Home Screen?");  
    return(super.onCreateOptionsMenu(menu));  
}  

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        //        case SHOW_MY_SHOPPING_LIST:  
        //            SavePreferences("str_from_products_in_categories",  
        //            str_from_products);  
        //            //Toast.makeText(getApplicationContext(), "Your  
        //            Shopping List: "+str_from_products, Toast.LENGTH_LONG).show();  
        //            //Intent intent2=new  
        //            Intent(SelectCategoriesActivity.this,ShowMyList.class);  
        //            //startActivity(intent2);  
        //            //            break;  
        //            case CLEAN_LIST:  
        //            Toast.makeText(getApplicationContext(), "Cleaned",  
        //            Toast.LENGTH_SHORT).show();  
        //            clear_list_flag=true;  
        //            //            SavePreferences("cleared_list_from_categories", "true");  
        //            SavePreferences("str_from_MyList", "empty");  
        //            //            str_from_products="";  
        //            //            break;  
        //            case DONE:  
        //            //            str_from_categories = str_from_products;  
    
}
SavePreferences("str_from_categories",
str_from_categories);

Intent intent3=new Intent(SelectCategoriesActivity.this, ShowFinalPage.class);
startActivity(intent3);

break;
case GO_HOME:
Intent intent4=new Intent(SelectCategoriesActivity.this, FinalProjectActivity.class);
startActivity(intent4);
break;
}
return(super.onOptionsItemSelected(item));

private void SavePreferences(String key, String value){
SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
SharedPreferences.Editor editor = sharedPreferences.edit();
editor.putString(key, value);
editor.commit();
}

private void LoadPreferences(){
SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
str_from_products= sharedPreferences.getString("str_from_products", "");
str_from_MyList= sharedPreferences.getString("str_from_MyList", "");

//==================================
last_visited_activity = sharedPreferences.getString("last_visited_activity", "");
added_item = sharedPreferences.getString("added_item", "");
//==================================
str_in_show_my_list = sharedPreferences.getString("str_in_show_my_list", ");
flag_deletion_status = sharedPreferences.getString("flag_deletion_status", "");
clicked_button_in_shopping_list = sharedPreferences.getString("clicked_button_in_shopping_list", ");
}

SelectProductActivity.java
package com.okan;

import java.util.ArrayList;
import java.util.List;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.AlertDialog;
import android.app.ListActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.util.SparseBooleanArray;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
public class SelectProductActivity extends ListActivity {
    String[] tokens;
    String category, store_name;
    String str_from_products;
    String cleared_list_from_categories;
    String my_sel_items;
    String str_from_MyList;
    String str_from_products_in_categories;
    //ArrayAdapter<Model> adapter;
    ArrayAdapter<String> adapter;
    ArrayList<String> savedItemsList;
    ListView listView;
    int position;
    int checkedItemsCount;

    //ArrayAdapter<Model> adapter;
    ArrayAdapter<String> adapter;
    ArrayList<String> savedItemsList;
    ListView listView;

    int position;
    int checkedItemsCount;

}
int count=0;
boolean flag=true;

Button save_and_go_back;
Button clean_my_list;

// private static final int SAVE_AND_GO_BACK = Menu.FIRST ;
// private static final int CLEAN_LIST = Menu.FIRST +1;

public void myDatabaseConnection()
{
    ArrayList<NameValuePair> postParameters = new
    ArrayList<NameValuePair>();
    postParameters.add(new BasicNameValuePair("category_name",
    category.toString()));
    String response = null;
    try {
        response=CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/oka
        n_final_project/selectproduct.php", postParameters);
        String res = response.toString();
        res = res.replaceAll(" ","");
        res = res.replaceAll("PARSE_"," ");
        res = res.replaceAll("n","n");
        String delims = ":[ ]
        tokens = res.split(delims);

    }catch (Exception e) {
        // TODO: handle exception
    }
}

@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.selectproduct);
    LoadPreferences();
    clean_my_list = (Button) findViewById(R.id.clean_my_list);
    save_and_go_back = (Button) findViewById(R.id.save_and_go_back);
    //------------------------------------------------------------
save_and_go_back.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        try{
            my_sel_items = my_sel_items.replaceAll("Selected Items: \n","");
        }
        catch (Exception e) {
            my_sel_items="";
        }
        str_from_products = str_from_products + my_sel_items;
        SavePreferences("str_from_MyList", str_from_products);
        Toast.makeText(getApplicationContext(), "Your Shopping List: \n"+str_from_products, Toast.LENGTH_LONG).show();
        SavePreferences("str_from_products", str_from_products);
        Intent intent=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);
        startActivity(intent);
    }
});

//---------------------------------------------------------------------------
clean_my_list.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(SelectProductActivity.this);
        builder.setMessage("Do you really want to clean?")
            .setCancelable(false)
            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    str_from_products="";
                    Toast.makeText(SelectProductActivity.this, "Cleaned \n"+str_from_products, Toast.LENGTH_LONG).show();
                    SavePreferences("str_from_products", str_from_products);
                    Intent intent3=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);
                    startActivity(intent3);
                }
            })
            .setNegativeButton("No", new DialogInterface.OnClickListener() {
                //
            });

    }
});

public void onClick(View v) {

    AlertDialog.Builder builder = new AlertDialog.Builder(SelectProductActivity.this);
    builder.setMessage("Do you really want to clean?")
        .setCancelable(false)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                str_from_products="";
                Toast.makeText(SelectProductActivity.this, "Cleaned \n"+str_from_products, Toast.LENGTH_LONG).show();
                SavePreferences("str_from_products", str_from_products);
                Intent intent3=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);
                startActivity(intent3);
            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            //
        });

    Toast.makeText(getApplicationContext(), "Your Shopping List: \n"+str_from_products, Toast.LENGTH_LONG).show();
    SavePreferences("str_from_products", str_from_products);
    Intent intent=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);
    startActivity(intent);
}
});
public void onClick(DialogInterface dialog, int id) {
    dialog.cancel();
}

AlertDialog alert = builder.create();
alert.show();

//Load icindekini kapattigim icin Acik burasi
if(cleared_list_from_categories.equalsIgnoreCase("true")){
    str_from_products = "";
}

//else
str_from_products = str_from_products_in_categories;

new Handler().postDelayed(new Runnable() {
    public void run() {
        openOptionsMenu();
    }
}, 1000);

//Toast.makeText(getApplicationContext(), str_from_products + " alindi", Toast.LENGTH_LONG).show();
SavePreferences("str_from_products", str_from_products);

setTitle( category+" in "+store_name);
myDatabaseConnection();

SavePreferences("last_visited_activity","product");

//ADAPTER
MySimpleArrayAdapter adapter = new MySimpleArrayAdapter(this, tokens);
setListAdapter(adapter);
View header = getLayoutInflater().inflate(R.layout.header, null);
//View footer = getLayoutInflater().inflate(R.layout.footer, null);
listView = getListView();
listView.addHeaderView(header);
//listView.addFooterView(footer);

//=============================================ADAPTER
2=============================================ADAPTER
// adapter = new InteractiveArrayAdapter(this, getModel());
// setListAdapter(adapter);
//=============================================ADAPTER

//=============================================ADAPTER
3=============================================ADAPTER
// listView.setAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_multiple_choice, tokens));
// listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);

adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_multiple_choice, tokens);
listView.setAdapter(adapter);
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);

//listView.setItemChecked(2, true);
listView.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        // TODO Auto-generated method stub
        my_sel_items=new String("Selected Items: \n");
        SparseBooleanArray a = listView.getCheckedItemPositions();
        checkedItemsCount = a.size();
        for(int i = 0; i < checkedItemsCount ; i++)
        {
            if (a.valueAt(i))
            {
                position = a.keyAt(i);

                // marginal code
            }
        }
    }
});
if(str_from_products.contains(listView.getAdapter().getItem(position).toString().trim()))
    Toast.makeText(getApplicationContext(), (String) listView.getAdapter().getItem(position)+" is already in the list, Uncheck it to add other items", Toast.LENGTH_SHORT).show();
else
    my_sel_items = my_sel_items + (String) listView.getAdapter().getItem(position)+ ",";
}

//Toast.makeText(getApplicationContext(), ""+my_sel_items, Toast.LENGTH_SHORT).show();
}
);

listView.setOnItemLongClickListener(new OnItemLongClickListener() {
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {

        // SavePreferences("last_visited", "show_my_list");
        SavePreferences("pos3", (String) listView.getAdapter().getItem(position));
        Intent intent=new Intent(SelectProductActivity.this,SearchTestActivity_2_from_Shopping_List.class);
        startActivity(intent);

        return true;
    }
});

//==============================================================

@Override
public void onResume(){
    super.onResume();

    // SavePreferences("last_visited_activity","product");
    //=======================================
}

private List<Model> getModel() {
    List<Model> list = new ArrayList<Model>();
    for(int i=0; i<tokens.length; i++)
        list.add(get(tokens[i]));
    //list.get(0).setSelected(true);
    return list;
}

private Model get(String s) {
    return new Model(s);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0, SAVE_AND_GO_BACK,0,"Save & Go Back");
    menu.add(1, CLEAN_LIST,1,"Clean My List");
    return(super.onCreateOptionsMenu(menu));
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case SAVE_AND_GO_BACK:
        try{
            my_sel_items = my_sel_items.replaceAll("Selected Items: \n","");
            }
        catch (Exception e) {
            my_sel_items="";
        }
        str_from_products = str_from_products + my_sel_items;
        SavePreferences("str_from_MyList", str_from_products);
        Toast.makeText(getApplicationContext(), "Your Shopping List: "+str_from_products, Toast.LENGTH_LONG).show();
        SavePreferences("str_from_products", str_from_products);
        Intent intent=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);
        return false;
    }
}

startActivity(intent);

break;

case CLEAN_LIST:

str_from_products="";

Toast.makeText(SelectProductActivity.this, "Cleaned "+str_from_products, Toast.LENGTH_LONG).show();

SavePreferences("str_from_products", str_from_products);

Intent intent3=new Intent(SelectProductActivity.this,SelectCategoriesActivity.class);

startActivity(intent3);

break;

return(super.onOptionsItemSelected(item));

private void SavePreferences(String key, String value){
    SharedPreferences sharedPreferences =getSharedPreferences("mypref", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

private void LoadPreferences(){
    SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
    category= sharedPreferences.getString("selected_category", "");
    store_name= sharedPreferences.getString("pos2", "");
    str_from_products= sharedPreferences.getString("str_from_products", "");
    cleared_list_from_categories= sharedPreferences.getString("cleared_list_from_categories", "");
    str_from_products_in_categories = sharedPreferences.getString("str_from_products_in_categories", "");

    //===================================
    str_from_products_in_categories = sharedPreferences.getString("str_from_products_in_categories", "");

    //=====================================================================================
package com.okan;

import java.util.ArrayList;
import java.util.List;
import org.apache.http.NameValuePair;
import android.app.ListActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class SelectStoreActivity_ListView extends ListActivity {
    String[] tokens;
    String item;
    StringBuilder str = new StringBuilder();
    TextView tx;

    public void myDatabaseConnection()
    {
        ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
        String response = null;
        try {
            response=CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/okan_final_project/selectstore.php", postParameters);
            String res = response.toString();
            res = res.replaceAll(" ", 
            res = res.replaceAll("PARSE_", " ");
            String delims = "[ ]";
            tokens = res.split(delims);
        }
    }
}
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setTitle("Select Store Screen");
    myDatabaseConnection();
}

//================================ADAPTER=================================
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, tokens);
setListAdapter(adapter);

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    item = (String) getListAdapter().getItem(position);
    Toast.makeText(this, item + " selected", Toast.LENGTH_LONG).show();
    SavePreferences("pos2", item);
    Intent intent=new Intent(SelectStoreActivity_ListView.this,FinalProjectActivity.class);
    startActivity(intent);
}

private void SavePreferences(String key, String value){
    SharedPreferences sharedPreferences =getSharedPreferences("mypref", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

ShowFinalPage.java

package com.okan;
public class ShowFinalPage extends ListActivity {
    String[] tokens;
    // String[] tokens2 = { "Product Name:", "Today`s Price:", "Aisle#:");
    String[] final_token;
    String str_from_categories;
    private ListView lv1;

    public void myDatabaseConnection() {
        ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
        postParameters.add(new BasicNameValuePair("string", str_from_categories.trim()));
        String response = null;
        try {
            response = CustomHttpClient.executeHttpPost("http://www.ilkerokanaker.info/okan_final_project/done_test.php", postParameters);
            String res = response.toString();
            res = res.replaceAll("\s",""),"n\n",""n";
            String delims = "[ ]";
        } catch (Exception e) { ...}
    }
}
tokens = res.split(delims);

} catch (Exception e) {
    // TODO: handle exception
}

@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    LoadPreferences();
    setTitle("Fastest Way to Reach Products (In Order)");
    //lv1=(ListView)findViewById(R.id.ListView01);
    myDatabaseConnection();

    // View header = getLayoutInflater().inflate(R.layout.header_2, null);
    // View footer = getLayoutInflater().inflate(R.layout.footer, null);
    // listView = getListView();
    // listView.addHeaderView(header);

    //==================================ADAPTER
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, tokens);
    setListAdapter(adapter);
    if(tokens[0].equalsIgnoreCase("Empty"))
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Your Shopping List is Empty")
            .setCancelable(false)
            .setPositiveButton("Okay", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                ShowFinalPage.this.finish();
            }
        });
        .setNegativeButton("No", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        AlertDialog alert = builder.create();
alert.show();

//====================ADAPTER
2===================================================
// ArrayList<Map<String, String>> list = buildData();
// String[] from = { "name", "purpose" };
// int[] to = { android.R.id.text1, android.R.id.text2, android.R.id.icon);
//
// SimpleAdapter adapter = new SimpleAdapter(this, list,
// android.R.layout.simple_list_item_2, from, to);
// setListAdapter(adapter);

//====================ADAPTER
3===================================================
// MySimpleArrayAdapter_NEW adapter = new
MySimpleArrayAdapter_NEW(this, tokens, tokens2);
// setListAdapter(adapter);

//====================ADAPTER
4===================================================
// ArrayAdapter<Model> adapter2 = new
InteractiveArrayAdapter(this, getModel());
// setListAdapter(adapter2);

@Override
public void onResume(){
    super.onResume();

    }

private List<Model> getModel() {
    List<Model> list = new ArrayList<Model>();
    for(int i=0; i<tokens.length; i++)
        list.add(get(tokens[i]));

    // Initially select one of the items
    // list.get(1).setSelected(true);
    return list;
}

private Model get(String s) {
    return new Model(s);
private ArrayList<Map<String, String>> buildData() {
    ArrayList<Map<String, String>> list = new ArrayList<Map<String, String>>();
    list.add(putData(tokens2[0], tokens[0]));
    list.add(putData(tokens2[1], tokens[1]));
    list.add(putData(tokens2[2], tokens[2]));
    return list;
}

private HashMap<String, String> putData(String name, String purpose) {
    HashMap<String, String> item = new HashMap<String, String>();
    item.put("name", name);
    item.put("purpose", purpose);
    return item;
}

private void LoadPreferences(){
    SharedPreferences sharedPreferences = getSharedPreferences("mypref", MODE_WORLD_READABLE);
    String str_from_categories = sharedPreferences.getString("str_from_categories", ".");
}

ShowMyList.java

package com.okan;

import java.util.ArrayList;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.AlertDialog;
import android.app.ListActivity;
import android.view.View;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.gesture.Gesture;
import android.gesture.GestureLibraries;
import android.gesture.GestureLibrary;
import android.gesture.Prediction;
public class ShowMyList extends ListActivity implements OnGesturePerformedListener
{
    private GestureLibrary mLibrary;
    String[] tokens, test_tokens;
    String[] final_token;
    String str_from_products_in_categories, copy_of_str_from_products_in_categories, str;
    String my_sel_items_in_ShowMyList;
    int position;
    int checkedItemsCount;
    int count = 0;
    private ListView listView;
    String selectedFromList;
    int number_of_products;

    ArrayAdapter<String> adapter;
    String res;

    private static final int INFO = Menu.FIRST;
    // private static final int TOTAL_COST = Menu.FIRST +1;

    Button info;
    Button total_cost;
}
public void myDatabaseConnection_total_cost()
{
    ArrayList<NameValuePair> postParameters = new
    ArrayList<NameValuePair>();
    postParameters.add(new BasicNameValuePair("string",
        str_from_products_in_categories.trim()));
    
    String response = null;
    
    try {
        response=CustomHttpClient.executeHttpPost("http://www.ilkero
        kanaker.info/okan_final_pro
        ject/total_cost.php", postParameters);
        res = response.toString();
    }
    catch (Exception e) {
        Toast.makeText(ShowMyList.this, "Hosting Server is not
        responding", Toast.LENGTH_LONG).show();
        // TODO: handle exception
    }
    
    @Override
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.gesture_list_activity);
    
    SavePreferences("flag_deletion_status", "false");
    SavePreferences("last_visited_activ
    ity","show_my_list");
    
    LoadPreferences();
    setTitle(" Show My List");
    
    myDatabaseConnection_total_cost();
    
    info = (Button) findViewById(R.id.delete_product);
    total_cost = (Button) findViewById(R.id.total_cost);
    
    res=res.trim();
    total_cost.setText("Total Cost is: $"+res);
    
    //---------------------------------------------------------------------
    info.setOnClickListener(new OnClickListener() {

public void onClick(View v) {
    AlertDialog.Builder builder = new AlertDialog.Builder(ShowMyList.this);
    builder.setMessage("Total # of items: \+number_of_products")
            .setCancelable(false)
            .setPositiveButton("Okay", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            });
    AlertDialog alert = builder.create();
    alert.show();
    // Intent intent=new Intent(ShowMyList.this,ImageDialog.class);
    // startActivity(intent);
}
//
// new Handler().postDelayed(new Runnable() {
//    public void run() {
//        openOptionsMenu();
//    }
//}, 1000);
SavePreferences("str_in_show_my_list", str_from_products_in_categories);

copy_of_str_from_products_in_categories = str_from_products_in_categories;
str_from_products_in_categories =
str_from_products_in_categories.replaceAll(",", "");
final_token = str_from_products_in_categories.split(" ");

number_of_products = final_token.length;
info.setText("Total # of items: "+number_of_products);

if(final_token[0].equalsIgnoreCase(""))
{
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Your Shopping List is Empty")
        .setCancelable(false)
        .setPositiveButton("Okay", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        ShowMyList.this.finish();
    }
});
//
    .setNegativeButton("No", new
DialogInterface.OnClickListener() {
    //
        public void onClick(DialogInterface dialog, int id) {
    //
        dialog.cancel();
    //
    //
    });
    AlertDialog alert = builder.create();
    alert.show();
}

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, final_token);
setListAdapter(adapter);

list = getListView();
list.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
    //
}
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    selectedFromList = (String) (list.getItemAtPosition(position));
    Toast.makeText(ShowMyList.this, selectedFromList, Toast.LENGTH_LONG).show();
}

//==============================================================
==============
//========================ADAPTER
2==========================
listView = getListView();
adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_multiple_choice, final_token);
listView.setAdapter(adapter);
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
listView.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        // TODO Auto-generated method stub

        my_sel_items_in_ShowMyList = new String(""");
        SparseBooleanArray a = listView.getCheckedItemPositions();
        checkedItemsCount = a.size();

        for (int i = 0; i < checkedItemsCount; i++) {
            if (a.valueAt(i)) {
                position = a.keyAt(i);
                my_sel_items_in_ShowMyList = my_sel_items_in_ShowMyList +
                    (String) listView.getAdapter().getItem(position) + ",";
            }
        }

        //Toast.makeText(getApplicationContext(), "" + my_sel_items_in_ShowMyList, Toast.LENGTH_SHORT).show();
    }
});
listView.setOnItemLongClickListener(new OnItemLongClickListener() {
    public boolean onItemLongClick(AdapterView<?> parent, View view,
    int position, long id) {

        //
        SavePreferences("last_visited", "show_my_list");
        SavePreferences("pos3", (String) listView.getAdapter().getItem(position));
        Intent intent=new Intent(ShowMyList.this,SearchTestActivity_2_from_Shopping_List.class);
        startActivity(intent);

        return true;
    }
});

//==============================================================
//======================
if(final_token[0].equalsIgnoreCase("")){
    SavePreferences("str_from_MyList", "empty");
} else{
    SavePreferences("str_from_MyList", "full");
}

mLibrary = GestureLibraries.fromRawResource(this, R.raw.actions);
if (!mLibrary.load()) {
    finish();
}

GestureOverlayView gestures = (GestureOverlayView) findViewById(R.id.gestures);
gestures.addOnGesturePerformedListener(this);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0, INFO,0,"How to remove a product?");
    return(super.onCreateOptionsMenu(menu));
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case INFO:
        Intent intent = new Intent(ShowMyList.this, ImageDialog.class);
        startActivity(intent);
        break;
    }
    return (super.onOptionsItemSelected(item));
}

public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture) {
    ArrayList<Prediction> predictions = mLibrary.recognize(gesture);
    if (predictions.size() > 0) {
        Prediction prediction = predictions.get(0);
        String action = predictions.get(0).name;
        /*Toast.makeText(this, prediction.name,
        Toast.LENGTH_SHORT).show();
        try {
        if ("action_refresh".equalsIgnoreCase(action))
        {
            //Toast.makeText(this, "Deleted",
            Toast.LENGTH_SHORT).show();
        }
        if(!my_sel_items_in_ShowMyList.equalsIgnoreCase("")){
            copy_of_str_from_products_in_categories =
            copy_of_str_from_products_in_categories.replace(my_sel_items_in_ShowMyList, "");
            Toast.makeText(this, "Selected Item(s)
            removed", Toast.LENGTH_SHORT).show();
            SavePreferences("str_in_show_my_list",
            copy_of_str_from_products_in_categories);
            SavePreferences("flag_deletion_status", "true");
        }*/
    }
}
Intent intent = new Intent(ShowMyList.this, SelectCategoriesActivity.class);
    startActivity(intent);
    } else {
    SavePreferences("str_in_show_my_list",
    str_from_products_in_categories);
    Toast.makeText(this, "No item selected to remove", Toast.LENGTH_SHORT).show();
    }
} catch (Exception e) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("No Item Selected to Remove !")
    .setCancelable(false)
    .setPositiveButton("Okay", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        dialog.cancel();
    } });
    AlertDialog alert = builder.create();
    alert.show();
}
}
private void SavePreferences(String key, String value) {
    SharedPreferences sharedPreferences = getSharedPreferences("mypref",
    MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}
private void LoadPreferences() {
    SharedPreferences sharedPreferences = getSharedPreferences("mypref",
    MODE_WORLD_READABLE);
    str_from_products_in_categories = sharedPreferences.getString("str_from_products_in_categories", "");
}