

“BACHELOR” Android Based Helping Tool for Bachelor People

B.Sc in Computer Science and Engineering
East West University



Prepared by

Rayhan Khan
ID: 2011-1-60-026

Supervised by

K. M. Imtiaz-Ud-Din
Senior Lecturer
Department of Computer Science & Engineering
East West University

DECLARATION

This is to certify that this project was prepared under CSE-497 project course and it was not been submitted elsewhere for the requirement of any degree or diploma or any other purpose.

Rayhan Khan
ID: 2011-1-60-026
Department of Computer Science and Engineering
East West University, Dhaka, Bangladesh.

LETTER OF ACCEPTANCE

I hereby declare that this project is from the student's own work and effort, and all other sources of information used have been acknowledged. This project has been submitted with my approval.

Supervisor:

K. M. Imtiaz-Ud-Din
Senior Lecturer
Department of Computer Science & Engineering
East West University

Chairperson:

Dr. Shamim Hasnat Ripon
Associate Professor
Computer Science & Engineering
East West University

ACKNOWLEDGEMENT

First of all Thanks to ALLAH for the uncountable blessings on me. Thanks to my Supervisor K. M. Imtiaz-Ud-Din for providing me this opportunity to test my skills in the best possible manner. He enlightened, encouraged, and provided me with ingenuity to transform my vision into reality.

Special Thanks to Shikkhok.com from where I learned a lot.

ABSTRACT

Social network based services offer many advantages to the mobile user to connect more people known or unknown. Social networks affect our lives in many ways. Some may say that using social network is nothing but waste of time. People waste there valuable time by chatting and doing unnecessary things in social networks. But if we utilize this vast network for our betterment, some aspect life can be easier for us. Using social network vast amount of people can be connected. This is the motivation of every social network.

Being empowered by the same motivation we tried to do something good for the bachelor community in our society. As currently there is no application or website to help or to provide services to them, we tried to help them a bit by our application. We tried to help them to manage their lifestyle by connecting them and tried to provide them a common platform where they can help each other in finding better housing facilities.

TABLE OF CONTENTS

1. INTRODUCTION.....	1-3
1.1. BACKGROUND.....	1
1.2. AIMS AND GOALS.....	2
1.3. OBJECTIVE.....	2
1.4. MOTIVATION.....	2
1.5. ORGANIZATION OF REPORT.....	3
2. PROPOSED MODEL.....	4-8
2.1. OVERVIEW.....	4
2.2. APPLICATION STRUCTURE AND MODEL.....	4
2.2.1. INTENT DEFINATION.....	4
2.2.2. SERVICE PROVIDER AND COMPONENTS.....	5
2.2.3. COMPONENT OF TWITTER FUNCTION.....	5
2.3. SYSTEM ARCHITECTURE.....	7
2.4. FLOW.....	8
3. IMPLEMENTATION.....	9-12
3.1. INTRODUCTION.....	9
3.2. FUNCTIONAL OVERVIEW.....	9
3.3. TECHNOLOGY USED.....	10
3.3.1. Java.....	10
3.3.2. SQLite.....	11
3.3.3. ANDROID OPERATING SYSTEM.....	11
3.4. SYSTEM EVALUATION AND TESTING.....	12
4. USERS MANUAL.....	13-21
4.1. INTRODUCTION.....	13
4.2. MINIMUM SYSTEM REQUIRMENT.....	13

4.3. INSTALLATION.....	14
4.3.1. STEPS OF INSTALLATION.....	14
4.3.2. GETTING STARTED.....	15
4.4. BACHELOR APPLICATION MANAGER.....	15
4.5. SUMMARY.....	21
5. CONCLUSIONS AND FUTURE WORK.....	22
6. APPENDIX A	23-71

BIBLIOGRAPHY.....	72
-------------------	----

LIST OF TABLES

1. 3.1 API version of Android.....	11
2. 4.1 Required Hardware.....	13

3. 4.2 Required Software.....13

LIST OF FIGURES

1. Fig.2.1 Components.....5
2. Fig.2.2 System Architecture.....7
3. Fig.4.1 Installation14
4. Fig.4.2 Demo “Budget”15
5. Fig.4.3Demo “Today”16
6. Fig.4.4Demo “Toilet”17
7. Fig.4.5 Demo “Twitter Tweet”, “Twitter Search”18
8. Fig.4.6Demo of “Search Results”19

9. Fig.4.7 Demo of "Tweet History"20

1 INTRODUCTION

The idea of using mobile handsets and phones is to provide valuable services except the basic communication that had been started in early 1990s when Internet was added to Voice Telephony service. At the beginning of 2000s small computer like palmtop, small tablet computers were first introduced. And the mobile phone people used to use changed in 1900s changed a lot too. Smart phone was introduced, which were lot more easily to use as easier User Interfaces were added. Their functionality was changed. Now with mobile phone people could stream videos online, create and edit documents, pretty much everything that a Computer can do.

After 2005 when different social networking sites emerged the application of mobile phone increased in an enormous rate. Now a day's almost everybody owns a smart phone and most of them has accounts on Social networking sites like Face book, Twitter, Viber etc.

There are many kind of operating system available in market for available devices. Such as Android, IOs, Windows, Firefox etc. Almost all of the social sites have their own APIs for different operating systems.

Social Network based applications are very easy and fun to use and also effective. Social Networking sites who allow third party applications to pull data from their sites can be used as in Different efficient and effective ways as resource of applications.

we can post on the site.

#we can search for someone.

we can search for specific post, searching for text wrapped with Hash Tags "#" is one of the most efficient to do that.

we can send private messages to our friends.

we can share files with others.

1.1 BACKGROUND

As we all know that bachelor people who leave their family in village and come to cities to work or study have to face a huge accommodation problem. Getting a proper house as a bachelor is more than tough. And there is no specific area for bachelor accommodation. So for the first time when a bachelor guy comes to city area face a lot of trouble. For a new city comer bachelor girl finding proper accommodation is almost impossible.

Cost maintenance is another issue for new city comers. How much money he/she is willing to spend for a month and how much they are really spending is another question to solve?

1.2 AIMS AND GOALS

The "BECHALORS" application is a Twitter API based Android application which is built specially for Bachelor people. Currently we provide 2 main services. Using our application user will be able to keep track of his daily expenditure and maintain a relation between his/her budget for the month. And user can easily Tweet and search Tweets (using # hash tag) for available bachelor To-let Tweets (wanted or available), posted by other users.

1.3 OBJECTIVE

The objective of the Bachelor application can be stated as follows:

1. Take input our budget for the month.
2. Take input and save the entire thing we buy.
3. Compare between our budget and our total expenditure.
4. Notify we at the end of every month.
5. Take in the specification of our available/wanted accommodation.
6. Tweet on our Twitter timeline on Behalf of us.
7. Search To-let related tweets for we.

1.4 MOTIVATION

In this project we use android device because of android cellular phones can be used to carry not only voice but also data traffic, such as text messages, pictures and video clips from anywhere at any time. Cell phones now emulate computers with enhanced graphical user interfaces, integrated Global Positioning Systems, wireless data connectivity, efficient batteries, powerful central processing units (CPU) and expended storage capabilities. Advanced communication protocols, databases, and software development environment ensure these and

system devices are connected to wireless cellular networks and can interact with many hosts and services via the internet.

Similarly, hardware independent programming languages allow the development of applications that can run on any of these devices and exchanged information to and from other clients, servers, and specialized databases.

Firstly-to learn a new that will likely find its way to the top of the mobile segment within the next few years.

Secondly-to get an experience to work with a real project

1.5 Organization of Report

The rest of this report is organized as follows, Chapter 2 discusses about the proposed model. Chapter 3 Technical Infrastructure related to our project & description of location-Based Technologies and software description which used in my project Chapter 4 includes the User Manual in my project, & finally, describes conclude and project overview in Chapter 5.

2 PROPOSED MODEL

This section contains Overview, Application Structure and Model, Service Provider and Components, System Architecture and Flow.

2.1 OVERVIEW

Android is software stack for mobile devices that includes an operating system, middleware and applications. Android is powered by Linux kernel initially developed by Google and later the Open Handset Alliance. It allows developers to write managed codes in java language, controlling the device via Google developed java library named Android Software Development Kit (SDK).

2.2 APPLICATION STRUCTURE AND MODEL

2.2.1 INTENT DEFINATION

Proposal of an integrated android application based on simple database (SQLITE) and Twitter API. The discussions in the previous section motivate us to adopt principle that the following functions are realized "on the spot".

- 1) Installation Process.
- 2) System takes input of the monthly budget.
- 3) User input daily bazaar items.
- 4) System compares total expenditure with budget.
- 5) Twitter function helps user to post or search To-let tweets for Bachelors.

2.2.2 SERVICE PROVIDER AND COMPONENTS

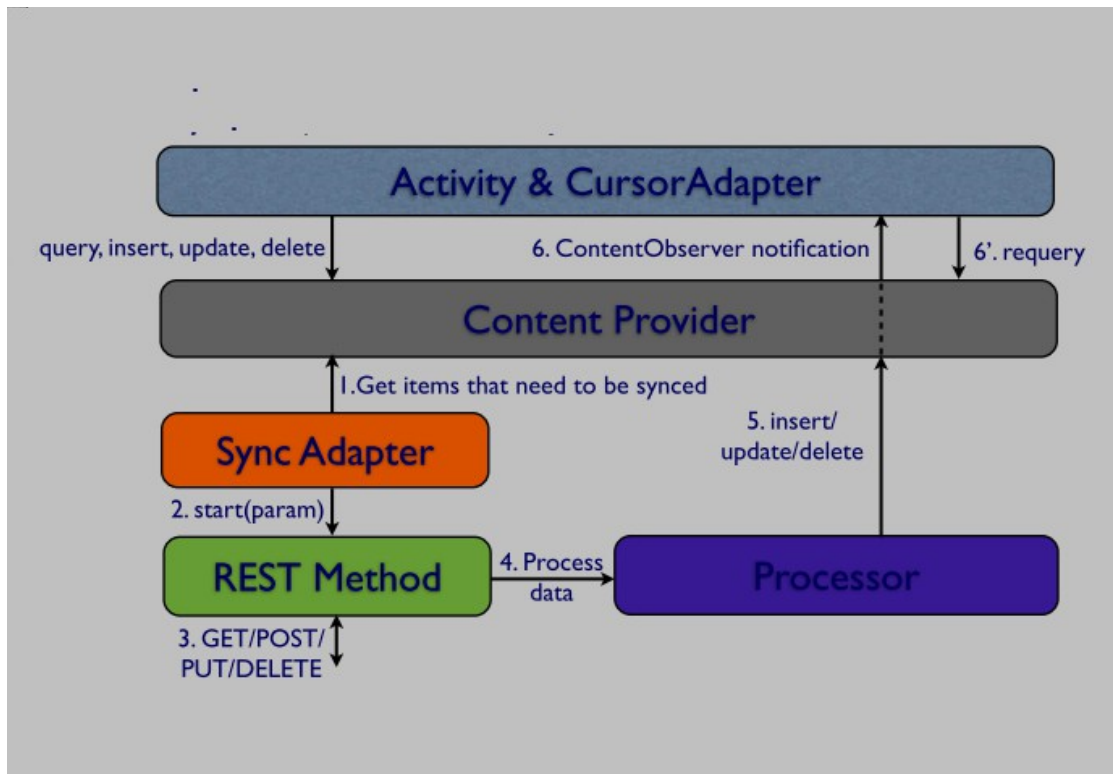


Fig. 2.1: Components

2.2.3 COMPONENT OF TWITTER FUNCTION:

Twitter offers a platform with a number of different ways to interact with it. Web Intents, Tweet Button and Follow Button are the simplest way to bring basic Twitter functionality to our site or APP. It provides features like the ability to tweet, retweet, or follow using basic HTML and JavaScript. We can also embed individual tweets. More complex integrations can utilize our [alias:/docs/api, title="REST"], Search The Search. API provides relevant results to ad-hoc user queries from a limited corpus of recent tweets. The REST API allows access to the nouns and verbs of Twitter such as reading timelines, tweeting, and following. To use the REST and Streaming API, we should register an application and get to know the ways of OAuth and Access token.

➤ OAuth

OAuth is an authentication protocol that allows users to approve application to act on their behalf without sharing their password. OAuth allows we to share our private resources (photos, videos, contact list, bank accounts) stored on one site with another site without having to hand out our username and password. There are many reasons why one should not share their

private credentials. Giving our email account password to a social network site so they can look up our friends is the same thing as going to dinner and giving our ATM card and PIN code to the waiter when it's time to pay. Any restaurant asking for our PIN code will go out of business, but when it comes to the web, users put themselves at risk sharing the same private information. OAuth to the rescue.

➤ Access tokens

In order to make authorized calls to Twitter's APIs, our application must first obtain an OAuth access token on behalf of a Twitter user or we could issue Application-only authenticated requests when user context is not required. The way we will obtain such tokens will depend on our use case.

Once we have an access token and token secret, the Twitter API is our oyster! By following the steps described in Authorizing a request, we can issue authorized requests to the REST API and the Streaming APIs. If the OAuth process sound like it is beyond the scope of our integration, consider using Web Intents, which do not need to use access tokens to interact with the Twitter API.

➤ REST API

There are currently two active versions of the [alias:/docs/api, title="Twitter REST API"]. [alias:/docs/api/1.1, title="REST API version 1.1"] is the most recent version of the API. To unifying versioning across the platform, version 1.1 also encompasses the Search and Streaming APIs. [alias:/docs/api/1, title="REST API version 1"] is now deprecated and will cease functioning in June 2013. Move to version 1.1 as soon as possible. The concept of "version 1" of the API includes unversioned APIs like the Search API.

Works Done by REST API:

- _Prepares the HTTP URL & HTTP request body
- Executes the HTTP transaction
- Processes the HTTP response

➤ REST API over JSON-P

The REST API supports a callback parameter on nearly all methods. In API v1.1 all requests require authentication. Because of this, most JSON-P use cases are actively discouraged as it is rarely possible to perform without exposing our client credentials.

2.3 SYSTEM ARCHITECTURE

The system architecture provides better understanding of the system in a detailed form as shown

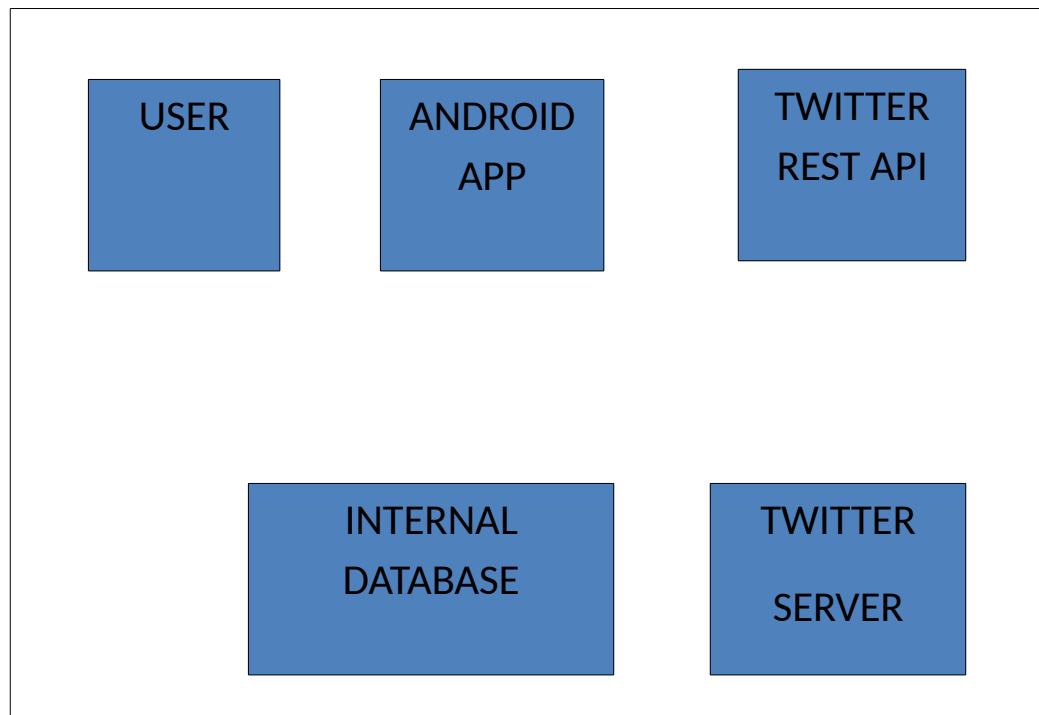


Fig.2.2: System Architecture

2.4 FLOW

The Flow shows how the application works.

1. User makes Twitter request or input/request data to the application
2. The application stores or retrieves the data in/from internal database.
3. Provides feedback to the user.
4. Or forms the request and pass it to the Twitter REST API.
5. The REST API add the access token and OAuth with the request and through internet pass send it to the Twitter Server.
6. The server sends data in JSON form.
7. REST Client process the data and provide to the application.

3 IMPLEMENTATION

This Section Contains INTRODUCTION, FUNCTIONAL OVERVIEW, TECHNOLOGY USED.

3.1 INTRODUCTION

Having developed several trial application consisting DATABASE and REST CLIENT API we started to develop our project application, BACHELOR. In the Introduction section we will discuss environment we used for the application building and different properties of the application.

The environment we used is one of the most famous IDE for java and Android development named Eclipse IDE for Java Developers Version: Luna Service Release 1a (4.4.1).

Properties of the Application:

Name: Bachelor

Minimum SDK Version: 14 (Ice Cream Sandwich)

Permissions: android.permission.INTERNET, android.permission.ACCESS_NETWORK_STATE.

3.2 FUNCTIONAL OVERVIEW

There are different classes in our project. In this section we will discuss the purpose of each of the classes.

- ✓ AlertDialogManage: Purpose of this class is to prompt an error message if the device is not connected with Internet when user tries to log in into Twitter.
- ✓ BazarHistory: This class is an Intent class. Which means it contains a XML Graphical Interface to interact with user. BazarHistory class shows all the shopping done by the user.
- ✓ BazarItem: The purpose of the class is to create objects.
- ✓ ConnectionDetector: Alerts if the device is connected with Internet or not. If not this class calls AlertDialogManage class.
- ✓ CustomAdapterforBazarHistory: As we said BazarHistory is an intent class, it contains a ListView. CustomAdapterforBazarHistory is the adapter for that ListView.
- ✓ CustomAdapterForPostHistory: This is another adapter for PostHistory Class.
- ✓ DBHelper: This class handles all the works done our database. Like Insert or search.
- ✓ MainActivity: This is an Intent class. This class is the entry point of our application.

- ✓ PostHistory: Purpose of this class is to show all the post we made to Twitter using our application.
- ✓ TodayActivity: This is another Intent class. The purpose of this class to take the data in (what a User bought Today) and save it to the Database.
- ✓ ToletActivity: Take in properties of a To-let post and save them in the database. Also pass the data to the Twitter class
- ✓ ToletClass: Purpose of this class is Create To-let object. , which contains the properties of a To-let post which will be posted in Twitter.
- ✓ Twitter: This is an Intent class, containing login, post, search and logout function. This class handles all the requests to the Twitter server.

3.3 TECHNOLOGY USED

We used most popular technology available

3.3.1 Java

Java is one the most language for the developers all over the world. Only java is supported by Android Operating System, means java is must to develop android apps.

3.3.2 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.

3.3.3 ANDROID OPERATING SYSTEM

Android is an open source and Linux based operating system designed foe touch screen mobile device such as smart phones and tablet computers. This open source code and permissive

licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Android has a growing selection of third party applications, which can be acquired by users either through an app store such as Google Play or the Amazon App store, or by downloading in the Java language using the Android software development kit (SDK).

Different versions of Android devices are given below:

Version	Codename	API
2.2	Froyo	8
2.3.3 - 2.3.7	Gingerbread	10
4.0.3 - 4.0.4	Ice Cream Sandwich	15
4.1.x	Jelly Bean	16
4.2.x		17
4.3		18
4.4	KitKat	19
5.0	Lollipop	21
5.1		22

Table 3.1: API Versions of Android

The project we developed for Gingerbread version using API level 15.

3.4 SYSTEM EVALUATION AND TESTING

I tested my application both in emulator and on a mobile device. Then I tested the following scenario.

1. I checked if the applications run properly and does not crash.
2. When the application is installed and ready to use I checked every button and activity response properly.
3. User can not make any empty request.
4. User can not provide any empty input.
5. Checked the amount of request user makes so that Time Out exception does not happen.

I also tested the mobile application on Android covering API levels 14,18,19,21. And the application was tested using Samsung Galaxy S and Sony Ericson handsets.

4 USERS MANUAL

This section contains Introduction, Installation, Application Limitation and Summary.

4.1 INTRODUCTION

The name of the application that we developed is BACHELOR. BACHELOR is a social network based helping tool for all the bachelor people out there. Using the Twitter platform we provide services. Any bachelor man or woman can use this application. It is very helpful to find accommodation or a suitable roommate. There is no need for registration. It's a great way of keeping track of your daily or monthly expenses.

4.2 MINIMUM SYSTEM REQUIREMENT

Hardware Requirement		
Device	Ram	Disk Space/ Memory
Any Android Device	64 MB or High	5 MB

Table 4.1: Required Hardware

Software Requirement		
Operating System	Version	Database
Android OS	Minimum 14 or Higher	SQLite

Table 4.2: Required Software

4.3 INSTALLATION

To install the application user need to import the installation package in his android device and run it.

“Bachelor_1.0.apk”

It is a 1.444 MB apk file. Generally the installed file stored in your phone memory but you can move it to your external memory.

Installation processes are very easy and user friendly. You just need to click on the installation package (apk file) to complete the installation.

4.3.1 STEPS OF INSTALLATION

1. Import the installation package, **Bachelor_1.0.apk**.
2. Tap the folder where you saved it.
3. Tap the application Package.
4. The process will ask you for your permission.
5. If you permit the package will be installed in your device.

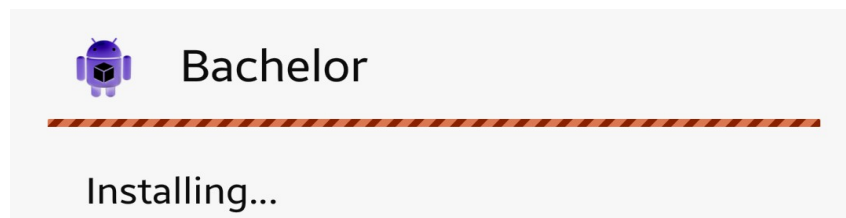


Fig.4.1: Installation

4.3.2 GETTING STARTED

Getting started with the BACHELOR app is very easy and simple. After completing the installation user will find the application icon in their Apps section.

Now user needs to configure some necessary setting in their respective device.

1. Turn on or enable internet connection or wifi on the respective device.

4.4 BACHELOR APPLICATION MANAGER

User needs to tap on BACHELOR icon. This will launch the 1st Screen. And there will be a toast message to update user's budget for the month.

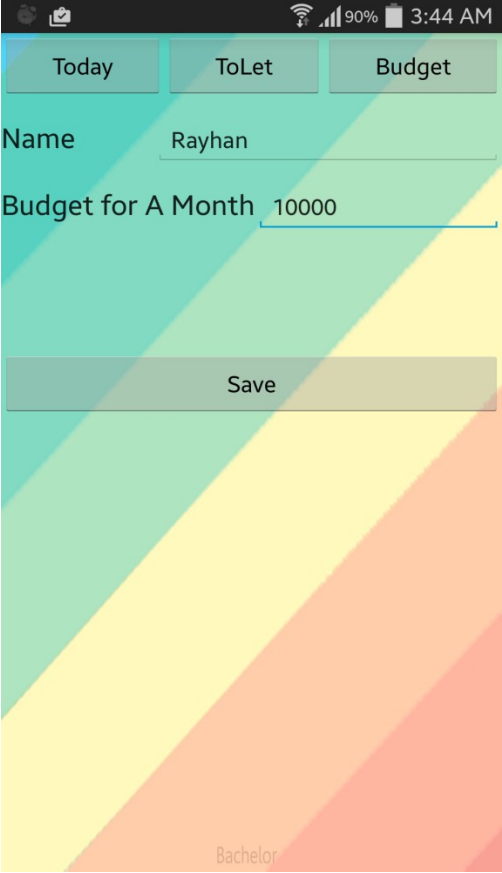


Fig.4.2: Demo “Budget”

Tap on to the Budget Button up there on the right. It will take you to the Budget page. Enter your name and the amount of the budget for the month. And press the save button. The data will be saved in a SQLite Database. You can no keep any field empty.

Now if you press back button you exit from the application. But if you tap the “**Today**” button up there on the left Today page will open. This is the page where you will be able to input all your expenses according to their type.

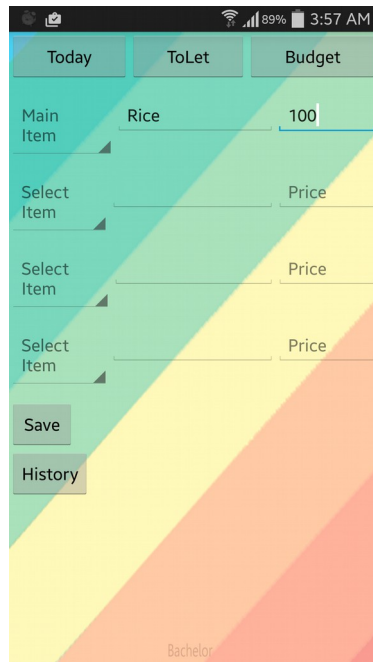
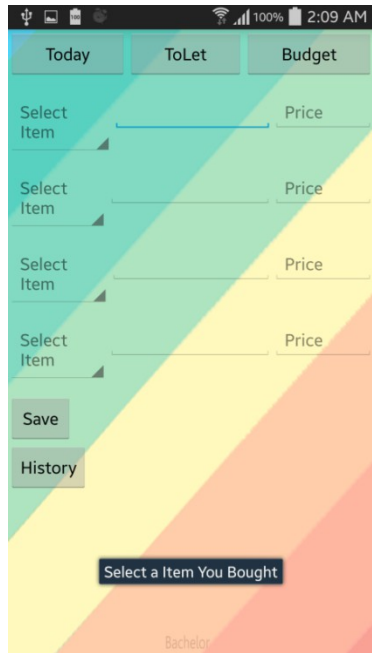




Fig.4.3: Demo of “Today”

There are 4 types of Items you can put in Main Item (Rice, Flour etc), Curry Item (Meat, Chicken, Fish, and Potato etc), Masala Item (all masalas), and Other Items.

Now suppose you bought Rice today, which is under the Main Item. Select Main Item, type the name of the item Rice (Auto complete text view will suggest you). Then type the price.

Now if you press “**Save**” button, the data you have typed will be saved in the database.

If you tap on “**History**” button you see the Item you bought earlier.

Now if you Need any roommate or need a room in a bachelor mess you simply tap on the “**Tolet**” button. This will open new screen.

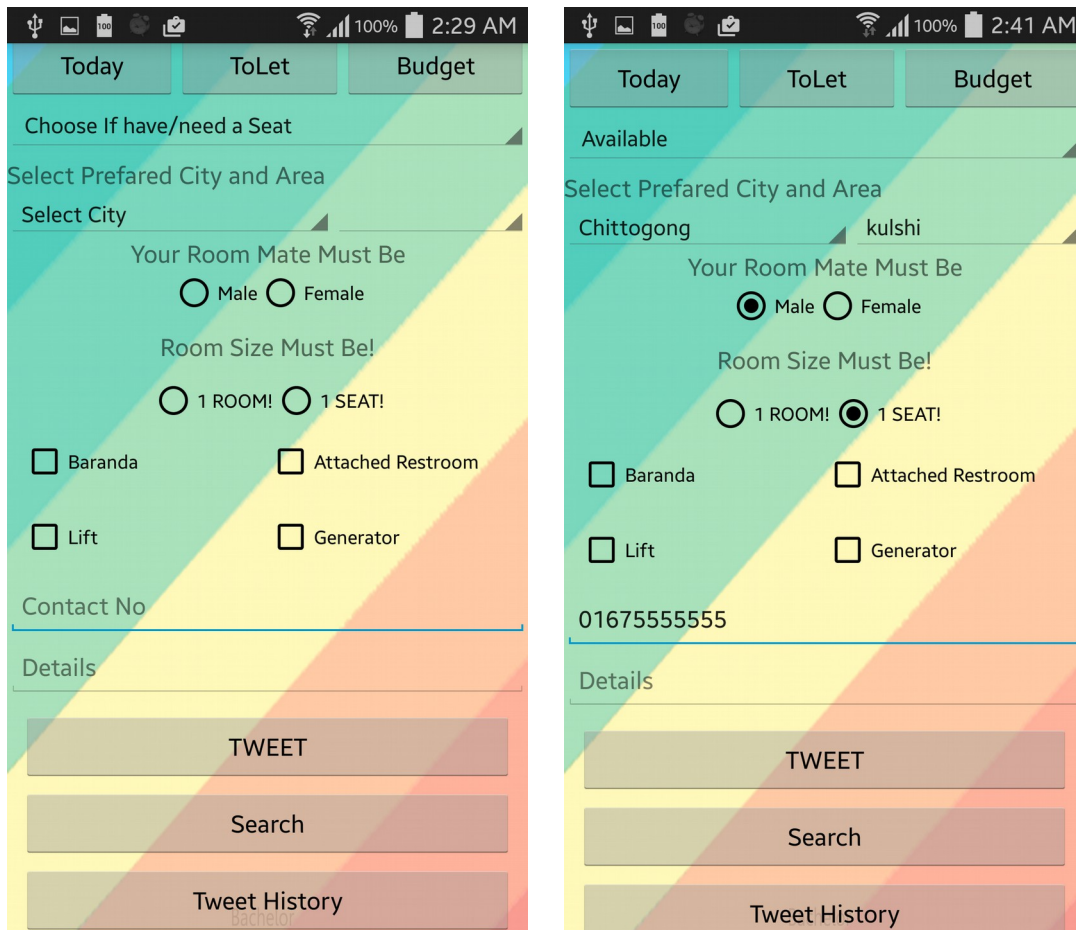


Fig.4.4: Demo of “Tolet”

Now suppose you have seat available in your mess and you want to let other bachelor people to know you simply just need to fill the form on this screen and tap on “**TWEET**” button. It will open one of your browser’s window and ask you to login into your Twitter account after it’s done the app will open a new page.

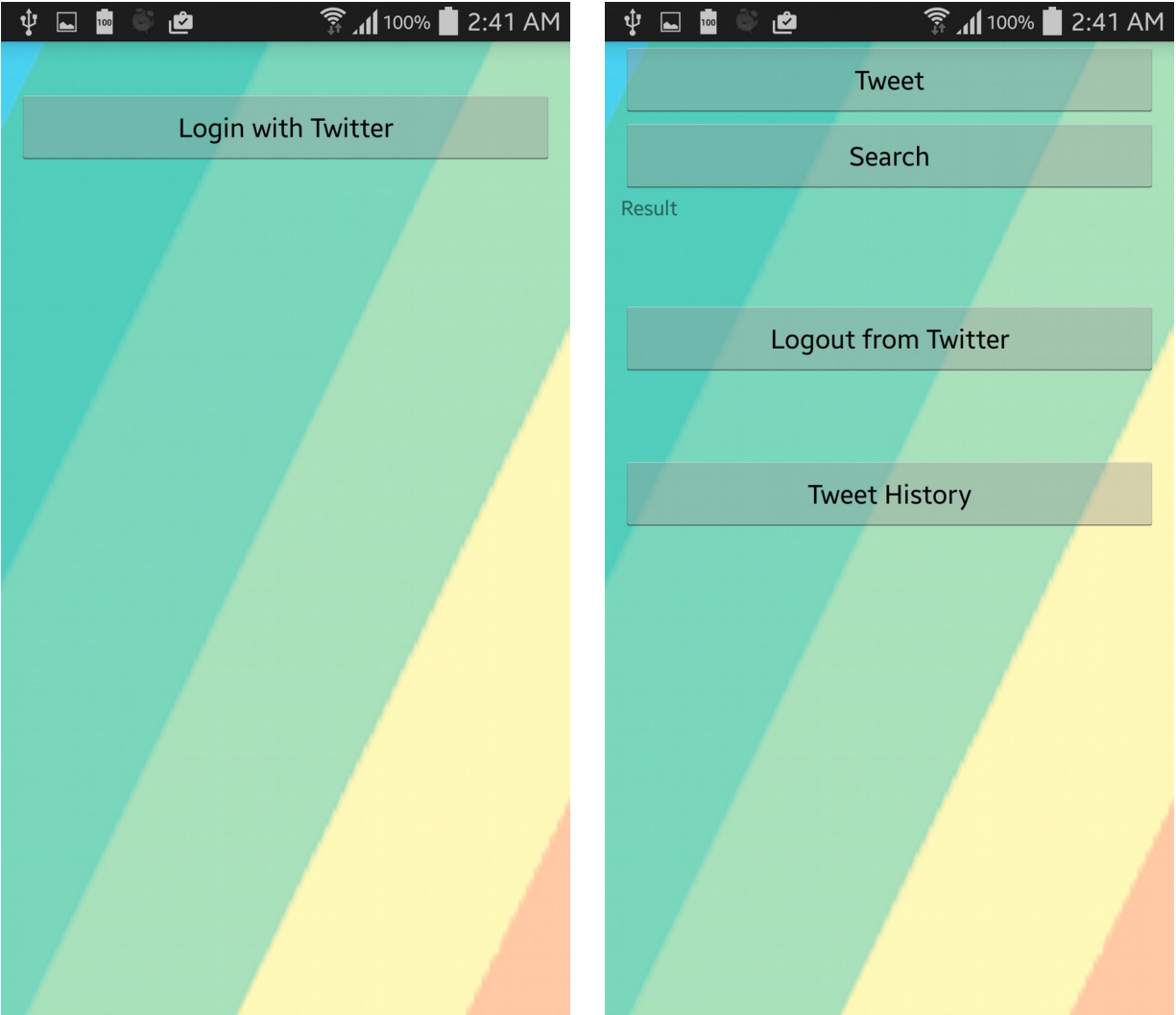


Fig.4.5: Demo of Twitter Tweet, Twitter Search

If you tap on **“Tweet”** your tweet will be posted.

If you tap on **“Search”** button this application will search on twitter for available posts and will show you 10 most recent posts.



Fig.4.6: Demo of “Search Results”

You can see here some posts I have made earlier.

Now if you tap on **“Tweet History”** you will find out all the old tweets you posted through application.

If you tap on **“Logout from Twitter”** you will be logged out from twitter get back to the Log in page.

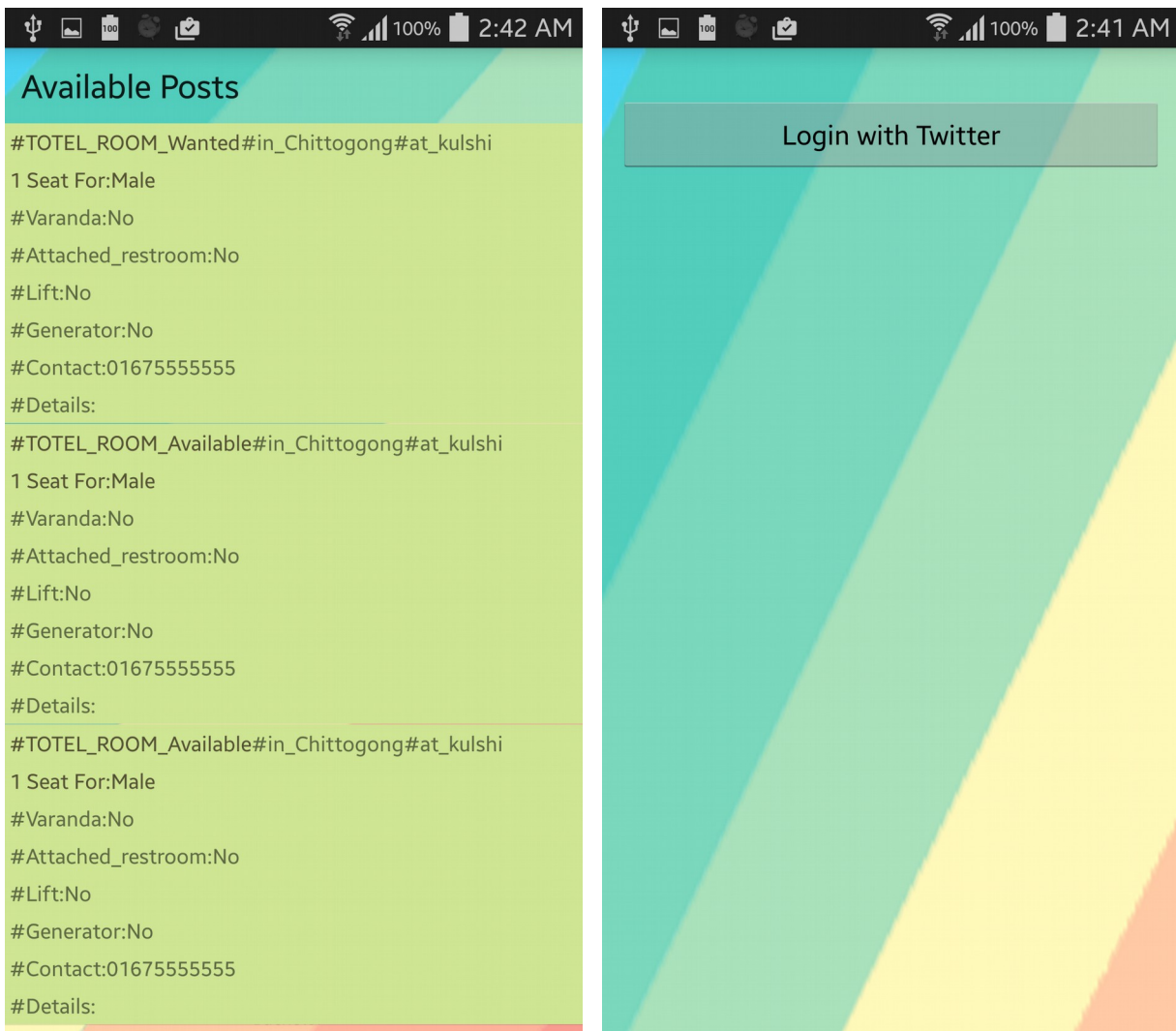


Fig.4.7: Demo of “Tweet History”

You can also search or check out you previous Tweets from **“Toilet”** Page. But to search on Twitter you login first.

4.5 SUMMARY

The User manual represents the basic overview on BACHELOR. It shows how to get familiar with the app. Although this is very easy to use its recommended that user must have some Idea how android application works.

5 CONCLUSIONS AND FUTURE WORK

There is no systematic and easy way to find accommodation for a bachelor here around in Bangladesh. And there is no efficient way to keep track of the expenses. It will make life easier for the bachelors. This application we have developed keeping focus on the way we can help people using existing social networks.

➤ **APPLICATION LIMITATION**

- Internet must be on if user want to post something or search from Twitter.
- Android version must be above of 2.2.3.
- User must have a Twitter account.

We hope that our application will help many Bachelor people around the world.

The project was completed based on only Android Operation System. Our vision is to develop the same application for IOS and Windows operating systems. We are also planning to make website based on the same Idea. Obviously we will integrate more social networks to our app in future.

6 APPENDIX A

This Section contains all the Source Codes.

6.1 Source Code

➤ MainActivity.java

```
package com.rayhan.bachelor;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
    TextView tvName;
    TextView tvBudget;
    TextView tvEx;
    TextView tvRe;

    ArrayList<BudgetClass> allbudget = new ArrayList<BudgetClass>();
    ArrayList<BazarClass> allbazar = new ArrayList<BazarClass>();

    DBHelper db = new DBHelper(this);

    int INTb;
    int ex;
    int Total_expense = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvName = (TextView) findViewById(R.id.tvName);
        tvBudget = (TextView) findViewById(R.id.tvBudget);
        tvEx = (TextView) findViewById(R.id.tvTotal);
        tvRe = (TextView) findViewById(R.id.tvRemain);

        allbudget = db.getAllBudget();
        allbazar = db.getAllBazarItem();
    }
}
```

```

if (allbudget != null && allbudget.size() > 0 && allbazar != null
    && allbazar.size() > 0) {

    int k = allbudget.size();
    BudgetClass bud = allbudget.get(k - 1);
    String budget = bud.getBudget();

    INTb = Integer.parseInt(budget);

    for (int i = 0; i < allbazar.size(); i++) {
        BazarClass bazar = allbazar.get(i);
        String e = bazar.getItemPrice();
        int ex = Integer.parseInt(e);
        Total_expense = Total_expense + ex;
    }
    int rem = INTb - Total_expense;

    tvName.setText("WELCOME");
    tvBudget.setText("Your Budget is: " + INTb + " TK");
    tvEx.setText("Total Expense of this month is: " + Total_expense
        + " TK");
    tvRe.setText("Remaining: " + rem);

} else {
    Toast.makeText(getApplicationContext(),
        "Please Update Your Budget For the Month",
        Toast.LENGTH_LONG).show();
}

}

public void today(View v) {
    Intent intent = new Intent(this, TodayActivity.class);
    startActivity(intent);
    finish();
}

public void tolet(View v) {
    Intent intent = new Intent(this, ToletActivity.class);
    startActivity(intent);
    finish();
}

public void budget(View v) {
    Intent intent = new Intent(this, BudgetActivity.class);
    startActivity(intent);
    finish();
}

```

```
}}  
➤ TodayActivity.java
```

```
package com.rayhan.bachelor;  
  
import java.util.Calendar;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.AutoCompleteTextView;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
public class TodayActivity extends Activity {  
    // data source  
    private String mainItem[] = { "Main Item", "Rice", "Flour", "Chal", "Moyda" };  
    private String curryItem[] = { "Curry Item",  
"Fish", "Potato", "Tomato", "Vegitable", "Chicken", "Beef" };  
    private String masalaItem[] = { "Masalas", "Onion", "Ginger", "Coryender" };  
    private String otherItem[] = { "Other Item" };  
    private String SpnSource[] = { "Select Item", "Main Item (Rice,Flour...)",  
"Curry(Fish,Meat,Vegs....)",  
    "Masala (Ginger,Onion.....)", "Other" };  
    EditText price1, price2, price3, price4;  
  
    DBHelper dbHelper;  
  
    // view object of Auto  
  
    private Spinner spn1;  
    private Spinner spn2;  
    private Spinner spn3;  
    private Spinner spn4;  
  
    private AutoCompleteTextView Auto1;  
    private AutoCompleteTextView Auto2;  
    private AutoCompleteTextView Auto3;  
    private AutoCompleteTextView Auto4;  
  
    // adapter
```

```
ArrayAdapter<String> forspn1;  
ArrayAdapter<String> forspn2;  
ArrayAdapter<String> forspn3;  
ArrayAdapter<String> forspn4;
```

```
ArrayAdapter<String> forAuto1;  
ArrayAdapter<String> forAuto2;  
ArrayAdapter<String> forAuto3;  
ArrayAdapter<String> forAuto4;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_today);  
  
    // initialize view object  
    price1 = (EditText) findViewById(R.id.etPrice1);  
    price2 = (EditText) findViewById(R.id.etPrice2);  
    price3 = (EditText) findViewById(R.id.etPrice3);  
    price4 = (EditText) findViewById(R.id.etPrice4);  
  
    spn1 = (Spinner) findViewById(R.id.spinner1);  
    spn2 = (Spinner) findViewById(R.id.spinner2);  
    spn3 = (Spinner) findViewById(R.id.spinner3);  
    spn4 = (Spinner) findViewById(R.id.spinner4);  
  
    Auto1 = (AutoCompleteTextView) findViewById(R.id.actmainItem);  
    Auto2 = (AutoCompleteTextView) findViewById(R.id.actcurryItem);  
    Auto3 = (AutoCompleteTextView) findViewById(R.id.actmasalaItem);  
    Auto4 = (AutoCompleteTextView) findViewById(R.id.actotherItem);  
  
    // adapter initialization  
    forspn1 = new ArrayAdapter<String>(TodayActivity.this,  
        android.R.layout.simple_list_item_1, SpnSource);  
    forspn2 = new ArrayAdapter<String>(TodayActivity.this,  
        android.R.layout.simple_list_item_1, SpnSource);  
    forspn3 = new ArrayAdapter<String>(TodayActivity.this,  
        android.R.layout.simple_list_item_1, SpnSource);  
    forspn4 = new ArrayAdapter<String>(TodayActivity.this,  
        android.R.layout.simple_list_item_1, SpnSource);  
  
    spn1.setAdapter(forspn1);  
    spn2.setAdapter(forspn2);  
    spn3.setAdapter(forspn3);  
    spn4.setAdapter(forspn4);  
  
    spn1.setOnItemClickListener(spn1Listener);  
    spn2.setOnItemClickListener(spn2Listener);
```

```

        spn3.setOnItemSelectedListener(spn3Listener);
        spn4.setOnItemSelectedListener(spn4Listener);
        dbHelper = new DBHelper(this);
    }

    OnItemSelectedListener spn1Listener = new OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> parent, View v, int position,
            long id) {

            if (spn1.getSelectedItemId() == 1) {

                forAuto1 = new ArrayAdapter<String>(TodayActivity.this,
                    android.R.layout.simple_list_item_1, mainItem);

            } else if (spn1.getSelectedItemId() == 2) {
                forAuto1 = new ArrayAdapter<String>(TodayActivity.this,
                    android.R.layout.simple_list_item_1, curryItem);
            } else if (spn1.getSelectedItemId() == 3) {
                forAuto1 = new ArrayAdapter<String>(TodayActivity.this,
                    android.R.layout.simple_list_item_1, masalaItem);
            } else if (spn1.getSelectedItemId() == 4) {
                forAuto1 = new ArrayAdapter<String>(TodayActivity.this,
                    android.R.layout.simple_list_item_1, otherItem);
            } else {
                Toast.makeText(getApplicationContext(),
                    "Select a Item You Bought",
                    Toast.LENGTH_SHORT).show();
            }
            Auto1.setAdapter(forAuto1);
            Auto1.setOnItemSelectedListener(Auto1SelectedListener);

        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }

    };

    OnItemSelectedListener spn2Listener = new OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> parent, View v, int position,

```

```

        long id) {

        if (spn2.getSelectedItemId() == 1) {

            forAuto2 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, mainItem);

        } else if (spn2.getSelectedItemId() == 2) {
            forAuto2 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, curryItem);
        } else if (spn2.getSelectedItemId() == 3) {
            forAuto2 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, masalaItem);
        } else if (spn2.getSelectedItemId() == 4) {
            forAuto2 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, otherItem);
        } else {
            Toast.makeText(getApplicationContext(),
                "Select a Item You Bought",
                Toast.LENGTH_LONG).show();
        }
        Auto2.setAdapter(forAuto4);
        Auto2.setOnItemClickListener(Auto2SelectedListener);

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

};

OnItemSelectedListener spn3Listener = new OnItemSelectedListener() {

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

        if (spn3.getSelectedItemId() == 1) {

            forAuto3 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, mainItem);

        } else if (spn3.getSelectedItemId() == 2) {
            forAuto3 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, curryItem);
        }
    }
};

```

```

        } else if (spn3.getSelectedItemId() == 3) {
            forAuto3 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, masalaItem);

        } else if (spn3.getSelectedItemId() == 4) {
            forAuto3 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, otherItem);

        } else {
            Toast.makeText(getApplicationContext(),
                "Select a Item You Bought",
                Toast.LENGTH_LONG).show();
        }
        Auto3.setAdapter(forAuto3);
        Auto3.setOnItemClickListener(Auto3SelectedListener);

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

};
OnItemSelectedListener spn4Listener = new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View v, int position,
        long id) {

        if (spn4.getSelectedItemId() == 1) {

            forAuto4 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, mainItem);

        } else if (spn4.getSelectedItemId() == 2) {
            forAuto4 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, curryItem);

        } else if (spn4.getSelectedItemId() == 3) {
            forAuto4 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, masalaItem);

        } else if (spn4.getSelectedItemId() == 4) {
            forAuto4 = new ArrayAdapter<String>(TodayActivity.this,
                android.R.layout.simple_list_item_1, otherItem);

        } else {

```

```

        Toast.makeText(getApplicationContext(),
                        "Select a Item You Bought",
Toast.LENGTH_LONG).show();
    }
    Auto4.setAdapter(forAuto4);
    Auto4.setOnItemSelectedListener(Auto4SelectedListener);

}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}

};

OnItemSelectedListener Auto4SelectedListener = new OnItemSelectedListener() {

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

        Log.i("Selected item ", otherItem[position]);

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

};

OnItemSelectedListener Auto3SelectedListener = new OnItemSelectedListener() {

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

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

};

OnItemSelectedListener Auto2SelectedListener = new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View view,

```



```

        int position, long id) {
            String pos = (String) parent.getItemAtPosition(position);
            Toast.makeText(getApplicationContext(), "item: " + pos,
                Toast.LENGTH_LONG).show();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
};

OnItemSelectedListener Auto1SelectedListener = new OnItemSelectedListener() {

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

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
};

public void today(View v) {
    Toast.makeText(getApplicationContext(),
        "You Are in this page currently", Toast.LENGTH_LONG).show();
}

public void tolet(View v) {

    Intent intent = new Intent(this, ToletActivity.class);
    startActivity(intent);
    finish();
}

public void budget(View v) {
    Intent intent = new Intent(this, BudgetActivity.class);
    startActivity(intent);
    finish();
}

```

```

}

public void save(View v) {

    if (spn1.getSelectedItemPosition() > 0) {
        String itemType = SpnSource[spn1.getSelectedItemPosition()];
        String itemName = Auto1.getEditableText().toString();
        String itemPrice = price1.getText().toString();
        String purchaseDate = java.text.DateFormat.getDateInstance()
            .format(Calendar.getInstance().getTime());

        BazarClass bazarClass = new BazarClass(itemType, itemName, itemPrice,
            purchaseDate);

        Toast.makeText(getApplicationContext(), bazarClass.toString(),
            Toast.LENGTH_SHORT).show();

        if (itemType != null && itemName != null && itemPrice != null) {
            long inserted = dbHelper.insertBazarTable(bazarClass);
            if (inserted >= 0) {
                Toast.makeText(getApplicationContext(), "saved",
                    Toast.LENGTH_SHORT).show();
                spn1.setSelection(0);
                Auto1.setText("");
                price1.setText("");
            } else {
                Toast.makeText(getApplicationContext(), "failed...",
                    Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(getApplicationContext(),
                "Please insert The Field properly...",
                Toast.LENGTH_SHORT).show();
        }
    }

    if (spn2.getSelectedItemPosition() > 0) {
        String itemType = SpnSource[spn2.getSelectedItemPosition()];
        String itemName = Auto2.getEditableText().toString();
        String itemPrice = price2.getText().toString();
        String purchaseDate = java.text.DateFormat.getDateInstance()
            .format(Calendar.getInstance().getTime());

        BazarClass bazarClass = new BazarClass(itemType, itemName, itemPrice,
            purchaseDate);

        Toast.makeText(getApplicationContext(), bazarClass.toString(),
            Toast.LENGTH_SHORT).show();
    }
}

```

```

        if (itemType != null && itemName != null && itemPrice != null) {
            long inserted = dbHelper.insertBazarTable(bazarClass);
            if (inserted >= 0) {
                Toast.makeText(getApplicationContext(), "saved",
                    Toast.LENGTH_SHORT).show();
                spn2.setSelection(0);
                Auto2.setText("");
                price2.setText("");
            } else {
                Toast.makeText(getApplicationContext(), "failed...",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}

if (spn3.getSelectedItemPosition() > 0) {
    String itemType = SpnSource[spn3.getSelectedItemPosition()];
    String itemName = Auto3.getEditableText().toString();
    String itemPrice = price3.getText().toString();
    String purchaseDate = java.text.DateFormat.getDateTimeInstance()
        .format(Calendar.getInstance().getTime());

    BazarClass bazarClass = new BazarClass(itemType, itemName, itemPrice,
        purchaseDate);

    Toast.makeText(getApplicationContext(), bazarClass.toString(),
        Toast.LENGTH_SHORT).show();
    if (itemType != null && itemName != null && itemPrice != null) {
        long inserted = dbHelper.insertBazarTable(bazarClass);
        if (inserted >= 0) {
            Toast.makeText(getApplicationContext(), "saved",
                Toast.LENGTH_SHORT).show();
            spn3.setSelection(0);
            Auto3.setText("");
            price3.setText("");
        } else {
            Toast.makeText(getApplicationContext(), "failed...",
                Toast.LENGTH_SHORT).show();
        }
    }
}

if (spn4.getSelectedItemPosition() > 0) {
    String itemType = SpnSource[spn4.getSelectedItemPosition()];
    String itemName = Auto4.getEditableText().toString();
    String itemPrice = price4.getText().toString();
    String purchaseDate = java.text.DateFormat.getDateTimeInstance()
        .format(Calendar.getInstance().getTime());

```

```

        BazarClass bazarClass = new BazarClass(itemType, itemName, itemPrice,
            purchaseDate);

        Toast.makeText(getApplicationContext(), bazarClass.toString(),
            Toast.LENGTH_SHORT).show();
        if (itemType != null && itemName != null && itemPrice != null) {
            long inserted = dbHelper.insertBazarTable(bazarClass);
            if (inserted >= 0) {
                Toast.makeText(getApplicationContext(), "saved",
                    Toast.LENGTH_SHORT).show();
                spn4.setSelection(0);
                Auto4.setText("");
                price4.setText("");
            } else {
                Toast.makeText(getApplicationContext(), "failed...",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }

}

public void history(View v) {
    Intent intent = new Intent(this, BazarHistory.class);
    startActivity(intent);
}
}

```

➤ ToletActivity.java

```
package com.rayhan.bachelor;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.Spinner;
import android.widget.Toast;

public class ToletActivity extends Activity implements OnCheckedChangeListener {

    // data source
    private String[] strChose = { "Choose If have/need a Seat", "Available",
        "Wanted" };

    private static String[] strCities = { "Select City", "Dhaka", "Chittogong",
        "Comilla", "sylhet", "Rajshahi", "Rangpur", "Barishal" };

    private static String[] strDhaka = { "Select Area", "Badda", "Gulshan",
        "Rampura", "Banasree", "Khilgao", "Dhanmondi", "Kolabagan" };

    private static String[] strCtg = { "kulshi", "lalbag", "HaliShohor",
        "Agrabad", "CU Area", "Chuet Area" };

    private static String[] strRaj = { "Near RU", "Near RUET", "Rani Nagar",
        "Padmar Par", "T Badh", "New Market" };

    private static String[] strCom = { "Ranir Dighi", "Taltola", "Doulotpur",
        "Mughol Tuli", "Amtoli", "Race Course", "Tomchom Bridge" };

    private static String[] strSyl = { "Jalalabad", "Topobon", "Akhalia",
        "Ambar Khana", "Modina Market", "Bandar B.", "Rakabi B." };

    // view object
    private Spinner spnpostOption;
    private Spinner spnCities;
    private Spinner spnAres;
```

```

// adapter
ArrayAdapter<String> optionAdapter;
ArrayAdapter<String> cityAdapter;
ArrayAdapter<String> areaAdapter;

// class
static ToletClass tolet = new ToletClass("", "", "", "", "", "", "", "", "",
                                         "", "", "");
// checkBox

EditText etPhone;
EditText etDetail;

CheckBox box1;
CheckBox box2;
CheckBox box3;
CheckBox box4;

DBHelper dbHelper;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tolet);
    // check box
    box1 = (CheckBox) findViewById(R.id.chkbox1);
    box2 = (CheckBox) findViewById(R.id.chkbox2);
    box3 = (CheckBox) findViewById(R.id.chkbox3);
    box4 = (CheckBox) findViewById(R.id.chkbox4);

    // editText
    etPhone = (EditText) findViewById(R.id.txtphn);
    etDetail = (EditText) findViewById(R.id.txtdetails);

    // initialize view object
    spnpostOption = (Spinner) findViewById(R.id.spnChoose);
    // initialize adapter
    optionAdapter = new ArrayAdapter<String>(ToletActivity.this,
                                           android.R.layout.simple_spinner_item, strChose);
    // binding
    spnpostOption.setAdapter(optionAdapter);
    spnpostOption.setOnItemSelectedListener(typelistener);
    spnCities = (Spinner) findViewById(R.id.spnCity);
    spnAres = (Spinner) findViewById(R.id.spnArea);

    // initialize Adapter
    cityAdapter = new ArrayAdapter<String>(ToletActivity.this,
                                           android.R.layout.simple_spinner_item, strCities);

```

```

// binding adapter and view
spnCities.setAdapter(cityAdapter);
spnCities.setOnItemSelectedListener(citylistener);

// radio group
RadioGroup Gunit1 = (RadioGroup) findViewById(R.id.rdo1);
Gunit1.setOnCheckedChangeListener(this);
RadioGroup Gunit2 = (RadioGroup) findViewById(R.id.rdo2);
Gunit2.setOnCheckedChangeListener(this);

dbHelper = new DBHelper(this);

}

OnItemSelectedListener citylistener = new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View view,
        int position, long id) {
        int pos = spnCities.getSelectedItemId();
        if (pos > 0) {
            String city = strCities[pos];
            toilet.setCity(city);
            Toast.makeText(getApplicationContext(), "City:" + city,
                Toast.LENGTH_SHORT).show();
        }
        if (spnCities.getSelectedItemId() == 1) {
            areaAdapter = new ArrayAdapter<String>(ToiletActivity.this,
                android.R.layout.simple_spinner_item, strDhaka);

        } else if (spnCities.getSelectedItemId() == 2) {

            areaAdapter = new ArrayAdapter<String>(ToiletActivity.this,
                android.R.layout.simple_spinner_item, strCtg);

        } else if (spnCities.getSelectedItemId() == 3) {
            areaAdapter = new ArrayAdapter<String>(ToiletActivity.this,
                android.R.layout.simple_spinner_item, strCom);

        } else if (spnCities.getSelectedItemId() == 4) {
            areaAdapter = new ArrayAdapter<String>(ToiletActivity.this,
                android.R.layout.simple_spinner_item, strSyl);
        } else if (spnCities.getSelectedItemId() == 5) {
            areaAdapter = new ArrayAdapter<String>(ToiletActivity.this,
                android.R.layout.simple_spinner_item, strRaj);
        }
        spnAres.setAdapter(areaAdapter);
    }
}

```

```

        spnAres.setOnItemSelectedListener(arealistener);
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
};
OnItemSelectedListener typelistener = new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View v, int position,
        long id) {
        int pos = spnpostOption.getSelectedItemId();
        if (pos > 0) {
            String postType = strChose[pos];
            tolet.setPostType(postType);

        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
};
OnItemSelectedListener arealistener = new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View v, int position,
        long id) {

        String area = (String) spnAres.getSelectedItem();
        tolet.setArea(area);

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
};

@Override
public void onCheckedChanged(RadioGroup group, int checkedId) {
    if (checkedId == R.id.rdoBtn1_1) {

```



```

        toilet.setGender("Male");
    } else if (checkedId == R.id.rdoBtn1_2) {
        toilet.setGender("Female");
    }
    if (checkedId == R.id.rdoBtn2_1) {
        toilet.setRoomSize("1 Room");
    } else if (checkedId == R.id.rdoBtn2_2) {
        toilet.setRoomSize("1 Seat");
    }
}

public void post(View v) {
    if (box1.isChecked() == true) {
        toilet.setVaranda("Yes");
    }
    if (box1.isChecked() == false) {
        toilet.setVaranda("No");
    }
    if (box2.isChecked() == true) {
        toilet.setRestroom("Yes");
    }
    if (box2.isChecked() == false) {
        toilet.setRestroom("No");
    }
    if (box3.isChecked() == true) {
        toilet.setLift("Yes");
    }
    if (box3.isChecked() == false) {
        toilet.setLift("No");
    }
}

```

```

    }
    if (box4.isChecked() == true) {

        toilet.setGenerator("yes");

    }
    if (box4.isChecked() == false) {

        toilet.setGenerator("No");

    }

    String phoneNo = etPhone.getText().toString();
    String detail = etDetail.getText().toString();
    toilet.setPhoneNo(phoneNo);
    toilet.setDetail(detail);

    String postType = toilet.getPostType().toString();
    String city = toilet.getCity().toString();
    String area = toilet.getArea().toString();
    String generator = toilet.getGenerator();
    String gender = toilet.getGender();
    String roomSize = toilet.getRoomSize();
    String restroom = toilet.getRestroom();
    String lift = toilet.getLift();
    String varanda = toilet.getVaranda();
    String PhoneNo = toilet.getPhoneNo();
    String Detail = toilet.getDetail();

    ToletClass toilet_class = new ToletClass(postType, city, area, gender,
        roomSize, varanda, restroom, lift, generator, PhoneNo, Detail);

    long inserted = dbHelper.insertTolet_Class(toilet_class);
    if (inserted >= 0) {
        Toast.makeText(getApplicationContext(), "saved", Toast.LENGTH_SHORT)
            .show();

    } else {
        Toast.makeText(getApplicationContext(), "failed...",
            Toast.LENGTH_SHORT).show();
    }
    Intent intent = new Intent(this, TwitterActivity.class);
    startActivity(intent);

}

public static ToletClass getObject() {

```

```

String postType = toilet.getPostType().toString();
String city = toilet.getCity().toString();
String area = toilet.getArea().toString();
String generator = toilet.getGenerator();
String gender = toilet.getGender();
String roomSize = toilet.getRoomSize();
String restroom = toilet.getRestroom();
String lift = toilet.getLift();
String varanda = toilet.getVaranda();
String PhoneNo = toilet.getPhoneNo();
String Detail = toilet.getDetail();

ToiletClass toilet_class = new ToiletClass(postType, city, area, gender,
roomSize, varanda, restroom, lift, generator, PhoneNo, Detail);
return toilet_class;

}

// listener for search
public void search(View v) {
    Intent intent = new Intent(this, TwitterActivity.class);
    startActivity(intent);
}

// listener for Tweet History Button
public void show(View v) {

    Intent intent = new Intent(this, TweetHistory.class);
    startActivity(intent);
}

public void today(View v) {
    Intent intent = new Intent(this, TodayActivity.class);
    startActivity(intent);
    finish();
}

public void toilet(View v) {

    Toast.makeText(getApplicationContext(),
        "You Are in this page currently", Toast.LENGTH_LONG).show();
}

public void budget(View v) {
    Intent intent = new Intent(this, BudgetActivity.class);

```

```

        startActivity(intent);
        finish();
    }
}

```

➤ ToletClass.java

```

package com.rayhan.bachelor;

import android.app.Activity;

public class ToletClass extends Activity{

    private int id;
    private String postType;
    private String city;
    private String area;
    private String gender;
    private String roomSize;
    private String varanda;
    private String restroom;
    private String lift;
    private String generator;
    private String phoneNo;
    private String detail;
    public ToletClass(String postType, String city, String area,
        String gender, String roomSize, String varanda, String restroom,
        String lift, String generator, String phoneNo, String detail) {
        super();
        this.postType = postType;
        this.city = city;
        this.area = area;
        this.gender = gender;
        this.roomSize = roomSize;
        this.varanda = varanda;
        this.restroom = restroom;
        this.lift = lift;
        this.generator = generator;
        this.phoneNo = phoneNo;
        this.detail = detail;
    }
    public int getId() {
        return id;
    }
}

```

```

}
public void setId(int id) {
    this.id = id;
}
public String getPostType() {
    return postType;
}
public void setPostType(String postType) {
    this.postType = postType;
}
public String getCity() {
    return city;
}
public void setCity(String city) {
    this.city = city;
}
public String getArea() {
    return area;
}
public void setArea(String area) {
    this.area = area;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public String getRoomSize() {
    return roomSize;
}
public void setRoomSize(String roomSize) {
    this.roomSize = roomSize;
}
public String getVaranda() {
    return varanda;
}
public void setVaranda(String varanda) {
    this.varanda = varanda;
}
public String getRestroom() {
    return restroom;
}
public void setRestroom(String restroom) {
    this.restroom = restroom;
}
public String getLift() {
    return lift;
}

```

```

    }
    public void setLift(String lift) {
        this.lift = lift;
    }
    public String getGenerator() {
        return generator;
    }
    public void setGenerator(String generator) {
        this.generator = generator;
    }
    public String getPhoneNo() {
        return phoneNo;
    }
    public void setPhoneNo(String phoneNo) {
        this.phoneNo = phoneNo;
    }
    public String getDetail() {
        return detail;
    }
    public void setDetail(String detail) {
        this.detail = detail;
    }
    @Override
    public String toString() {
        return "Toilet_Class [postType=" + postType + ", city=" + city
            + ", area=" + area + ", gender=" + gender + ", roomSize="
            + roomSize + ", varanda=" + varanda + ", restroom=" + restroom
            + ", lift=" + lift + ", generator=" + generator + ", phoneNo="
            + phoneNo + ", detail=" + detail + "];"
    }
}
}

```

➤ BazarClass.java

```
package com.rayhan.bachelor;

public class BazarClass {
    private String itemType;
    private String itemName;
    private String itemPrice;
    private String purchaseDate;
    public BazarClass(String itemType, String itemName, String itemPrice,
        String purchaseDate) {
        super();
        this.itemType = itemType;
        this.itemName = itemName;
        this.itemPrice = itemPrice;
        this.purchaseDate = purchaseDate;
    }
    @Override
    public String toString() {
        return "BazarItem [itemType=" + itemType + ", itemName=" + itemName
            + ", itemPrice=" + itemPrice + ", purchaseDate=" + purchaseDate
            + "]";
    }
    public String getItemType() {
        return itemType;
    }
    public void setItemType(String itemType) {
        this.itemType = itemType;
    }
    public String getItemName() {
        return itemName;
    }
    public void setItemName(String itemName) {
        this.itemName = itemName;
    }
    public String getItemPrice() {
        return itemPrice;
    }
    public void setItemPrice(String itemPrice) {
        this.itemPrice = itemPrice;
    }
    public String getPurchaseDate() {
        return purchaseDate;
    }
}
```

```

    }
    public void setPurcheDate(String purcheDate) {
        this.purchaseDate = purcheDate;
    }
}

```

➤ BazarHistory.java

```

package com.rayhan.bachelor;

import java.util.ArrayList;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ListView;

public class BazarHistory extends Activity {
    DBHelper dbHelper;
    // Declare View
    ListView lvBazaritem;
    // Declare Adapter
    CustomAdapterforBazarHistory adapter;

    // Data Source

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.bazar_history);
        dbHelper = new DBHelper(this);
        lvBazaritem = (ListView) findViewById(R.id.lvBazar);

        ArrayList<BazarClass> bazarClass = dbHelper.getAllBazarItem();

        if (bazarClass != null && bazarClass.size() > 0) {

            adapter = new CustomAdapterforBazarHistory(this, bazarClass);
            lvBazaritem.setAdapter(adapter);

        }

    }
}

```



```
}
```

➤ BudgetActivity.java

```
package com.rayhan.bachelor;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class BudgetActivity extends Activity {

    EditText etBudget;
    String bSum;
    BudgetClass budget;

    DBHelper db = new DBHelper(this);

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

        etBudget = (EditText) findViewById(R.id.etBudget);
    }

    public void today(View v) {
        Intent intent = new Intent(this, TodayActivity.class);
        startActivity(intent);
    }

    public void toilet(View v) {
        Intent intent = new Intent(this, ToiletActivity.class);
        startActivity(intent);
    }

    public void budget(View v) {
        Toast.makeText(getApplicationContext(),
            "You Are in this page currently", Toast.LENGTH_LONG).show();
    }
}
```

```

}

public void save(View v) {

    String Budget = etBudget.getText().toString();

    ArrayList<BudgetClass> a = db.getAllBudget();
    int i = a.size();
    if (i > 0) {
        BudgetClass b = a.get(i - 1);
        String s = b.getBudget().toString();
        int k = Integer.parseInt(s);
        int j = Integer.parseInt(Budget);
        bSum = Integer.toString(k + j);
        Toast.makeText(getApplicationContext(), bSum, Toast.LENGTH_LONG)
            .show();
        budget = new BudgetClass(bSum);
    } else {
        budget = new BudgetClass(Budget);
    }

    if (Budget != null) {
        long inserted = db.insertBudgetTable(budget);

        if (inserted >= 0) {
            Toast.makeText(getApplicationContext(), "Saved",
                Toast.LENGTH_LONG).show();
            etBudget.setText("");
        }
    } else {
        Toast.makeText(getApplicationContext(),
            "Fill up the Fields properly", Toast.LENGTH_LONG).show();
    }
}
}
}

```

➤ BudgetClass.java

```
package com.rayhan.bachelor;

public class BudgetClass {
    private String Budget;

    public String getBudget() {
        return Budget;
    }

    public void setBudget(String budget) {
        Budget = budget;
    }

    public BudgetClass(String budget) {
        super();
        Budget = budget;
    }

    @Override
    public String toString() {
        return "BudgetClass [Budget=" + Budget + " ]";
    }
}
```

➤ TwitterActivity.java

```
package com.rayhan.bachelor;

import java.util.ArrayList;

import twitter4j.Query;
import twitter4j.QueryResult;
import twitter4j.Status;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.User;
import twitter4j.auth.AccessToken;
import twitter4j.auth.RequestToken;
import twitter4j.conf.Configuration;
import twitter4j.conf.ConfigurationBuilder;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.content.pm.ActivityInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class TwitterActivity extends Activity {

    static String TWITTER_CONSUMER_KEY = "w171QPVom8pj2BhrQjVdzeMVg";
    static String TWITTER_CONSUMER_SECRET =
"Kx2xBR3lfMBw2SOjNORYOdZ5D4uDZDj5Wa7UsZ0PWbNN7Y2cGZ";

    static String PREFERENCE_NAME = "twitter_oauth";
    static final String PREF_KEY_OAUTH_TOKEN = "oauth_token";
    static final String PREF_KEY_OAUTH_SECRET = "oauth_token_secret";
```

```

static final String PREF_KEY_TWITTER_LOGIN = "isTwitterLoggedIn";

static final String TWITTER_CALLBACK_URL = "oauth://t4jsample";
public static String tweetString;

static final String URL_TWITTER_AUTH = "auth_url";
static final String URL_TWITTER_OAUTH_VERIFIER = "oauth_verifier";
static final String URL_TWITTER_OAUTH_TOKEN = "oauth_token";

// Login button
Button btnLoginTwitter;
// Update status button
Button btnUpdateStatus;
// Logout button
Button btnLogoutTwitter;
// Search button
Button btnSearch;
// show button
Button btnShow;

TextView tvResult;

// Progress dialog
ProgressDialog pDialog;
// int
public int k = 0;
// Object
ToiletClass toilet;

// Twitter
private static twitter4j.Twitter twitter;
private static RequestToken requestToken;

// Shared Preferences
private static SharedPreferences mSharedPreferences;

// Internet Connection detector
private ConnectionDetector cd;

// Alert Dialog Manager
AlertDialogManager alert = new AlertDialogManager();

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.twitter_activity);
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

cd = new ConnectionDetector(getApplicationContext());
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
    .permitAll().build();
StrictMode.setThreadPolicy(policy);
// Check if Internet present
if (!cd.isConnectingToInternet()) {
    // Internet Connection is not present
    alert.showAlertDialog(TwitterActivity.this, "Internet Connection Error",
        "Please connect to working Internet connection", false);
    // stop executing code by return
    return;
}

// Check if twitter keys are set
if (TWITTER_CONSUMER_KEY.trim().length() == 0
    || TWITTER_CONSUMER_SECRET.trim().length() == 0) {
    // Internet Connection is not present
    alert.showAlertDialog(TwitterActivity.this, "Twitter oAuth tokens",
        "Please set your twitter oauth tokens first!", false);
    // stop executing code by return
    return;
}

btnLoginTwitter = (Button) findViewById(R.id.btnLoginTwitter);
btnUpdateStatus = (Button) findViewById(R.id.btnUpdateStatus);
btnSearch = (Button) findViewById(R.id.btnSearch);
btnLogoutTwitter = (Button) findViewById(R.id.btnLogoutTwitter);
tvResult = (TextView) findViewById(R.id.tvResult);
btnShow = (Button) findViewById(R.id.btnshow);

// Shared Preferences
mSharedPreferences = getApplicationContext().getSharedPreferences(
    "MyPref", 0);

/**
 * Twitter login button click event will call loginToTwitter() function
 * */
btnLoginTwitter.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // Call login twitter function
        loginToTwitter();
    }
}

```

```

});
btnSearch.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View arg0) {
        // Call Search twitter function
        search();
    }
});

/**
 * Button click event to Update Status, will call updateTwitterStatus()
 * function
 * */
btnUpdateStatus.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // Call update status function
        // Get the status from EditText
        // String status = txtUpdate.getText().toString();
        ToletClass tolet = ToletActivity.getObject();

        String postType = tolet.getPostType().toString();
        String city = tolet.getCity().toString();
        String area = tolet.getArea().toString();
        String generator = tolet.getGenerator();
        String gender = tolet.getGender();
        String roomSize = tolet.getRoomSize();
        String restroom = tolet.getRestroom();
        String lift = tolet.getLift();
        String varanda = tolet.getVaranda();
        String PhoneNo = tolet.getPhoneNo();
        String Detail = tolet.getDetail();

        String status = "#Bachelor_Toilet " + roomSize + ":" + postType
            + " in:" + city + "," + area + " " + "For:" + gender
            + " " + "AttachedToilet:" + restroom + " "
            + "Varendra:" + varanda + " " + "Lift:" + lift + " "
            + "Generator:" + generator + " " + "Phone:" + PhoneNo;

        // Check for blank text
        if (status.trim().length() > 85 && status.trim().length() <= 140) {
            // update status
            new updateTwitterStatus().execute(status);
        } else {

```

```

        // EditText is empty
        Toast.makeText(
           (getApplicationContext(),
                "Message should be more than 0 and less than
140 Charecter",
                Toast.LENGTH_SHORT).show();
    }
}
});

/**
 * Button click event for logout from twitter
 */
btnLogoutTwitter.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // Call logout twitter function
        logoutFromTwitter();
    }
});

/**
 * This if conditions is tested once is redirected from twitter page.
 * Parse the uri to get oAuth Verifier
 */
if (!isTwitterLoggedInAlready()) {
    Uri uri = getIntent().getData();
    if (uri != null && uri.toString().startsWith(TWITTER_CALLBACK_URL)) {
        // oAuth verifier
        String verifier = uri
            .getQueryParameter(URL_TWITTER_OAUTH_VERIFIER);

        try {
            // Get the access token
            AccessToken accessToken = twitter.getOAuthAccessToken(
                requestToken, verifier);

            // Shared Preferences
            Editor e = mSharedPreferences.edit();

            // After getting access token, access token secret
            // store them in application preferences
            e.putString(PREF_KEY_OAUTH_TOKEN, accessToken.getToken());
            e.putString(PREF_KEY_OAUTH_SECRET,
                accessToken.getTokenSecret());
            // Store login status - true
            e.putBoolean(PREF_KEY_TWITTER_LOGIN, true);

```



```

        e.commit(); // save changes

        Log.e("Twitter OAuth Token", "> " + accessToken.getToken());

        // Hide login button
        btnLoginTwitter.setVisibility(View.GONE);

        // Show Update Twitter

        btnUpdateStatus.setVisibility(View.VISIBLE);
        btnLogoutTwitter.setVisibility(View.VISIBLE);
        btnSearch.setVisibility(View.VISIBLE);
        tvResult.setVisibility(View.VISIBLE);
        btnShow.setVisibility(View.VISIBLE);

    } catch (Exception e) {
        // Check log for login errors
        Log.e("Twitter Login Error", "> " + e.getMessage());
    }
}

}

}

/**
 * Function to login twitter
 */

private void loginToTwitter() {
    // Check if already logged in
    if (!isTwitterLoggedInAlready()) {
        ConfigurationBuilder builder = new ConfigurationBuilder();
        builder.setOAuthConsumerKey(TWITTER_CONSUMER_KEY);
        builder.setOAuthConsumerSecret(TWITTER_CONSUMER_SECRET);
        Configuration configuration = builder.build();

        TwitterFactory factory = new TwitterFactory(configuration);
        twitter = factory.getInstance();

        try {
            requestToken = twitter
                .getOAuthRequestToken(TWITTER_CALLBACK_URL);
            this.startActivity(new Intent(Intent.ACTION_VIEW, Uri

```

```

                .parse(requestToken.getAuthenticationURL()));
        } catch (TwitterException e) {
            e.printStackTrace();
        }
    } else {
        // user already logged into twitter
        Toast.makeText(getApplicationContext(),
            "Already Logged into twitter", Toast.LENGTH_LONG).show();
        // Hide login button
        btnLoginTwitter.setVisibility(View.GONE);

        // Show Update Twitter

        btnUpdateStatus.setVisibility(View.VISIBLE);
        btnLogoutTwitter.setVisibility(View.VISIBLE);
        btnSearch.setVisibility(View.VISIBLE);
        tvResult.setVisibility(View.VISIBLE);
        btnShow.setVisibility(View.VISIBLE);
    }
}

/**
 * Function to update status
 */
class updateTwitterStatus extends AsyncTask<String, String, String> {

    /**
     * Before starting background thread Show Progress Dialog
     */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(TwitterActivity.this);
        pDialog.setMessage("Updating to twitter...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    /**
     * getting Places JSON
     */
    protected String doInBackground(String... args) {
        Log.d("Tweet Text", "> " + args[0]);
        String status = args[0];
        try {
            ConfigurationBuilder builder = new ConfigurationBuilder();
            builder.setOAuthConsumerKey(TWITTER_CONSUMER_KEY);

```

```

        builder.setOAuthConsumerSecret(TWITTER_CONSUMER_SECRET);

        // Access Token
        String access_token = mSharedPreferences.getString(
            PREF_KEY_OAUTH_TOKEN, "");
        // Access Token Secret
        String access_token_secret = mSharedPreferences.getString(
            PREF_KEY_OAUTH_SECRET, "");

        AccessToken accessToken = new AccessToken(access_token,
            access_token_secret);
        twitter4j.Twitter twitter = new TwitterFactory(builder.build())
            .getInstance(accessToken);

        // Update status
        twitter4j.Status response = (twitter).updateStatus(status);

        Log.d("Status", "> " + response.getText());
    } catch (TwitterException e) {
        // Error in updating status
        Log.d("Twitter Update Error", e.getMessage());
    }
    return null;
}

/**
 * After completing background task Dismiss the progress dialog and show
 * the data in UI Always use runOnUiThread(new Runnable()) to update UI
 * from background thread, otherwise you will get error
 * **/
protected void onPostExecute(String file_url) {
    // dismiss the dialog after getting all products
    pDialog.dismiss();
    // updating UI from Background Thread
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(getApplicationContext(),
                "Status tweeted successfully",
                Toast.LENGTH_SHORT)
                .show();
            // Clearing EditText field
        }
    });
}
}
}

```

```

/**
 * Function to logout from twitter It will just clear the application shared
 * preferences
 */
private void logoutFromTwitter() {
    // Clear the shared preferences
    Editor e = mSharedPreferences.edit();
    e.remove(PREF_KEY_OAUTH_TOKEN);
    e.remove(PREF_KEY_OAUTH_SECRET);
    e.remove(PREF_KEY_TWITTER_LOGIN);
    e.commit();

    // After this take the appropriate action
    // I am showing the hiding/showing buttons again
    // You might not needed this code
    btnLogoutTwitter.setVisibility(View.GONE);
    btnUpdateStatus.setVisibility(View.GONE);
    btnSearch.setVisibility(View.GONE);
    tvResult.setVisibility(View.GONE);
    tvResult.setText("");
    btnShow.setVisibility(View.GONE);

    btnLoginTwitter.setVisibility(View.VISIBLE);
}

public void search() {
    ConfigurationBuilder builder = new ConfigurationBuilder();
    builder.setOAuthConsumerKey(TWITTER_CONSUMER_KEY);
    builder.setOAuthConsumerSecret(TWITTER_CONSUMER_SECRET);

    // Access Token
    String access_token = mSharedPreferences.getString(
        PREF_KEY_OAUTH_TOKEN, "");
    // Access Token Secret
    String access_token_secret = mSharedPreferences.getString(
        PREF_KEY_OAUTH_SECRET, "");

    AccessToken accessToken = new AccessToken(access_token,
        access_token_secret);
    twitter4j.Twitter twitter = new TwitterFactory(builder.build())
        .getInstance(accessToken);

    // search
    Query query = new Query("#bachelor_tolet");

    long lastID = Long.MAX_VALUE;
    ArrayList<Status> tweets = new ArrayList<Status>();

```

```

query.setCount(10);

try {
    QueryResult result = twitter.search(query);
    tweets.addAll(result.getTweets());
    System.out.println("Gathered " + tweets.size() + " tweets" + "\n");
    for (Status t : tweets)
        if (t.getId() < lastID)
            lastID = t.getId();

} catch (TwitterException te) {
    System.out.println("Couldn't connect: " + te);
}
;tvResult.setText("");

for (int i = 0; i < tweets.size(); i++) {
    Status t = (Status) tweets.get(i);

    String user = t.getUser().getScreenName();
    String msg = t.getText();

    String previousTweet = tvResult.getText().toString();
    tvResult.setText(previousTweet + "\n" + i + "@" + user
        + " wrote: " + msg + "\n");
}

}

// Check user already logged in your application using twitter Login flag is

private boolean isTwitterLoggedInAlready() {
    // return twitter login status from Shared Preferences
    return mSharedPreferences.getBoolean(PREF_KEY_TWITTER_LOGIN, false);
}

protected void onResume() {
    super.onResume();
}

// listener for Tweet History Button
public void show(View v) {

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

}

```

```

public static String postItem(ToletClass tolet_class) {
    String postType = tolet_class.getPostType();
    String Varanda = tolet_class.getVaranda();
    String RoomSize = tolet_class.getRoomSize();
    String area = tolet_class.getArea();
    String City = tolet_class.getCity();
    String Details = tolet_class.getDetail();
    String Gender = tolet_class.getGender();
    String Generator = tolet_class.getGenerator();
    String Lift = tolet_class.getLift();
    String phoneString = tolet_class.getPhoneNo();
    String restroom = tolet_class.getRestroom();

    tweetString = "#bachelor_tolet" + postType + " #in_" + City + " #at"
        + area + "\n" + RoomSize + " For:" + Gender + "\n" + "Varanda:"
        + Varanda + "\n" + "Attached Rest Room:" + restroom + "\n"
        + "Lift:" + Lift + "\n" + "Genarator:" + Generator + "\n"
        + "Details:" + Details + "\n" + "Contact No:" + phoneString;
    return tweetString;
}
}

```

➤ TweetHistory.java

```

package com.rayhan.bachelor;

import java.util.ArrayList;

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

public class TweetHistory extends Activity {
    DBHelper dbHelper;
    ListView lvPost;
    CustomAdapterForPostHistory adapterForPost;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tweet_history);
    }
}

```

```

        lvPost = (ListView) findViewById(R.id.lvPost);
        dbHelper = new DBHelper(this);

        ArrayList<ToiletClass> allpost = dbHelper.getAllPosts();
        if (allpost != null && allpost.size() > 0) {

                adapterForPost = new CustomAdapterForPostHistory(this, allpost);
                lvPost.setAdapter(adapterForPost);

        }

    }

    public void tweet(View v) {
        Intent intent = new Intent(this, TwitterActivity.class);
        startActivity(intent);
    }
}

```

➤ DBHelper.java

```

package com.rayhan.bachelor;

import java.util.ArrayList;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBHelper extends SQLiteOpenHelper {

    public static String DB_NAME = "beachelor";
    public static int DB_VIRSION = 1;
    public static String TABLE_NAME = "bazar";
    public static String TABLE_NAME1 = "rentPosts";

    public static String ID = "_id";
    public static String POSTTYPE = "postType";
    public static String CITY = "city";
    public static String AREA = "area";
}

```

```

public static String GENDER = "gender";
public static String ROOMSIZE = "roomSize";
public static String VARANDA = "varanda";
public static String RESTROOM = "restroom";
public static String LIFT = "lift";
public static String GENARATOR = "generator";
public static String PHONENO = "phoneNo";
public static String DETAIL = "detail";

public static String DATE_FIELD = "date";
public static String ID_FIELD = "Id";
public static String ITEM_FIELD = "item";
public static String TYPE_FIELD = "type";
public static String PRICE_FIELD = "price";

public static String BUDGET = "budget";
public static String TABLE_NAME2 = "budget_table";

public static final String BUDGET_TABLE_SQL = "CREATE TABLE " + TABLE_NAME2
        + " ( " + ID + " INTEGER PRIMARY KEY, " + BUDGET
        + " TEXT DEFAULT 0)";

public static final String BAZAR_TABLE_SQL = "CREATE TABLE " + TABLE_NAME
        + " ( " + ID_FIELD + " INTEGER PRIMARY KEY, " + DATE_FIELD
        + " TEXT, " + ITEM_FIELD + " TEXT, " + TYPE_FIELD + " TEXT, "
        + PRICE_FIELD + " TEXT )";

public static final String RENT_POST_TABLE_SQL = "CREATE TABLE "
        + TABLE_NAME1 + " ( " + ID + " INTEGER PRIMARY KEY, " + POSTTYPE
        + " TEXT, " + CITY + " TEXT, " + AREA + " TEXT, " + GENDER
        + " TEXT, " + ROOMSIZE + " TEXT, " + VARANDA + " TEXT, "
        + RESTROOM + " TEXT, " + LIFT + " TEXT, " + GENARATOR + " TEXT, "
        + DETAIL + " TEXT, " + PHONENO + " TEXT )";

public DBHelper(Context context) {
    super(context, DB_NAME, null, DB_VIRSION);
}

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL(BAZAR_TABLE_SQL);
    db.execSQL(RENT_POST_TABLE_SQL);
    db.execSQL(BUDGET_TABLE_SQL);
    Log.e("Table Created", BAZAR_TABLE_SQL);
    Log.e("Table Created", RENT_POST_TABLE_SQL);
    Log.e("Table Created", BUDGET_TABLE_SQL);
}

```



```

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub

}

public long insertBazarTable(BazarClass bazarClass) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(DATE_FIELD, bazarClass.getPurcheDate());
    values.put(TYPE_FIELD, bazarClass.getItemType());
    values.put(ITEM_FIELD, bazarClass.getItemName());
    values.put(PRICE_FIELD, bazarClass.getItemPrice());
    long inserted = db.insert(TABLE_NAME, null, values);

    db.close();
    return inserted;
}

public long insertBudgetTable(BudgetClass budgetoftheMonth) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(BUDGET, budgetoftheMonth.getBudget());

    long inserted = db.insert(TABLE_NAME2, null, values);

    db.close();
    return inserted;
}

public long insertTolet_Class(ToletClass tolet_class) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values1 = new ContentValues();

    values1.put(POSTTYPE, tolet_class.getPostType());
    values1.put(CITY, tolet_class.getCity());
    values1.put(AREA, tolet_class.getArea());
    values1.put(GENDER, tolet_class.getGender());
    values1.put(ROOMSIZE, tolet_class.getRoomSize());
    values1.put(VARANDA, tolet_class.getVaranda());
    values1.put(RESTROOM, tolet_class.getRestroom());
}

```

```

        values1.put(LIFT, tolet_class.getLift());
        values1.put(GENARATOR, tolet_class.getGenerator());
        values1.put(PHONENO, tolet_class.getPhoneNo());
        values1.put(DETAIL, tolet_class.getDetail());
        long inserted = db.insert(TABLE_NAME1, null, values1);

        db.close();
        return inserted;
    }

    public ArrayList<BazarClass> getAllBazarItem() {
        ArrayList<BazarClass> allBazarItem = new ArrayList<BazarClass>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db
            .query(TABLE_NAME, null, null, null, null, null, null);
        if (cursor != null && cursor.getCount() > 0) {
            cursor.moveToFirst();
            for (int i = 0; i < cursor.getCount(); i++) {
                String PurchaseDate = cursor.getString(cursor
                    .getColumnIndex(DATE_FIELD));
                String ItemType = cursor.getString(cursor
                    .getColumnIndex(TYPE_FIELD));
                String ItemName = cursor.getString(cursor
                    .getColumnIndex(ITEM_FIELD));
                String ItemPrice = cursor.getString(cursor
                    .getColumnIndex(PRICE_FIELD));

                BazarClass b = new BazarClass(ItemType, ItemName, ItemPrice,
                    PurchaseDate);
                allBazarItem.add(b);
                cursor.moveToNext();
            }
        }
        cursor.close();
        db.close();

        return allBazarItem;
    }

    public ArrayList<BudgetClass> getAllBudget() {
        ArrayList<BudgetClass> allBudget = new ArrayList<BudgetClass>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_NAME2, null, null, null, null, null,
            null);
        if (cursor != null && cursor.getCount() > 0) {
            cursor.moveToFirst();
            for (int i = 0; i < cursor.getCount(); i++) {

```

```

        String budget = cursor.getString(cursor.getColumnIndex(BUDGET));

        BudgetClass b = new BudgetClass(budget);
        allBudget.add(b);
        cursor.moveToNext();
    }
}
cursor.close();
db.close();

return allBudget;
}

public ArrayList<ToiletClass> getAllPosts() {
    ArrayList<ToiletClass> allpost = new ArrayList<ToiletClass>();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_NAME1, null, null, null, null, null,
        null);
    if (cursor != null && cursor.getCount() > 0) {
        cursor.moveToFirst();
        for (int i = 0; i < cursor.getCount(); i++) {
            String postType = cursor.getString(cursor
                .getColumnIndex(POSTTYPE));
            String city = cursor.getString(cursor.getColumnIndex(CITY));
            String area = cursor.getString(cursor.getColumnIndex(AREA));
            String gender = cursor.getString(cursor.getColumnIndex(GENDER));
            String roomSize = cursor.getString(cursor
                .getColumnIndex(ROOMSIZE));
            String varanda = cursor.getString(cursor
                .getColumnIndex(VARANDA));
            String restroom = cursor.getString(cursor
                .getColumnIndex(RESTROOM));
            String lift = cursor.getString(cursor.getColumnIndex(LIFT));
            String generator = cursor.getString(cursor
                .getColumnIndex(GENARATOR));
            String phoneNo = cursor.getString(cursor
                .getColumnIndex(PHONENO));
            String detail = cursor.getString(cursor.getColumnIndex(DETAIL));

            ToiletClass toilet_class = new ToiletClass(postType, city, area,
                gender, roomSize, varanda, restroom, lift, generator,
                phoneNo, detail);
            allpost.add(toilet_class);
            cursor.moveToNext();
        }
    }
}

```

```
        cursor.close();
        db.close();

        return allpost;
    }
}
```

➤ AlertDialogManager.java

```
package com.rayhan.bachelor;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;

public class AlertDialogManager {
    public void showAlertDialog(Context context, String title, String message,
        Boolean status) {
        AlertDialog alertDialog = new AlertDialog.Builder(context).create();

        // Setting Dialog Title
        alertDialog.setTitle(title);

        // Setting Dialog Message
        alertDialog.setMessage(message);

        if(status != null)
            // Setting alert dialog icon
            alertDialog.setIcon((status) ? R.drawable.success : R.drawable.fail);

        // Setting OK Button
        alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }
}
```

➤ ConnectionDetector.java

```
package com.rayhan.bachelor;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

public class ConnectionDetector {

    private Context _context;

    public ConnectionDetector(Context context){
        this._context = context;
    }

    public boolean isConnectingToInternet(){
        ConnectivityManager connectivity = (ConnectivityManager)
        _context.getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivity != null)
        {
            NetworkInfo[] info = connectivity.getAllNetworkInfo();
            if (info != null)
                for (int i = 0; i < info.length; i++)
                    if (info[i].getState() == NetworkInfo.State.CONNECTED)
                    {
                        return true;
                    }
        }
        return false;
    }
}
```

➤ CustomAdapterforBazarHistory.java

```
package com.rayhan.bachelor;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.TextView;

public class CustomAdapterforBazarHistory extends ArrayAdapter<BazarClass> {
    Activity con;
    ArrayList<BazarClass> bazarList;

    public CustomAdapterforBazarHistory(Context context, ArrayList<BazarClass> bazarClass) {
        super(context, R.layout.list_view, bazarClass);
        this.con = (Activity) context;
        this.bazarList = bazarClass;
    }

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

        final ViewHolder mHolder;
        View v = null;
        LayoutInflater inflater = con.getLayoutInflater();
        // generate a view
        mHolder = new ViewHolder();
        v = inflater.inflate(R.layout.list_view, null);

        mHolder.txtPurchasedate = (TextView) v.findViewById(R.id.txtDate);
        mHolder.txtType = (TextView) v.findViewById(R.id.txtType);
        mHolder.txtName = (TextView) v.findViewById(R.id.txtName);
        mHolder.txtPrice = (TextView) v.findViewById(R.id.txtPrice);

        v.setTag(mHolder);
        BazarClass b = bazarList.get(position);
        mHolder.txtPurchasedate.setText("Purchase Date:"
```



```

        this.con=(Activity)context;
        this.toilet_Classes=allpost;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        final ViewHolder1 mHolder;
        View v = null;
        LayoutInflater inflater = con.getLayoutInflater();

        // generate a view
        mHolder = new ViewHolder1();
        v = inflater.inflate(R.layout.list_view_post, null);

        mHolder.tvpostType=(TextView)v.findViewById(R.id.tvpostoption);
        mHolder.tvcity=(TextView)v.findViewById(R.id.tvcity);
        mHolder.tvarea=(TextView)v.findViewById(R.id.tvarea);
        mHolder.tvgender=(TextView)v.findViewById(R.id.tvgender);
        mHolder.tvroomSize=(TextView)v.findViewById(R.id.tvroomsize);
        mHolder.tvvaranda=(TextView)v.findViewById(R.id.tvvaranda);
        mHolder.tvrestroom=(TextView)v.findViewById(R.id.tvrestroom);
        mHolder.tvlift=(TextView)v.findViewById(R.id.tvlift);
        mHolder.tvgenerator=(TextView)v.findViewById(R.id.tvgenerator);
        mHolder.tvphoneNo=(TextView)v.findViewById(R.id.tvphone);
        mHolder.tvdetail=(TextView)v.findViewById(R.id.tvdetail);

        v.setTag(mHolder);
        ToiletClass p=toilet_Classes.get(position);

        mHolder.tvpostType.setText("#TOTELE_ROOM_"+p.getPostType());
        mHolder.tvcity.setText("#in_"+p.getCity());
        mHolder.tvarea.setText("#at_"+p.getArea());
        mHolder.tvgender.setText(" For:"+p.getGender());
        mHolder.tvroomSize.setText(p.getRoomSize());
        mHolder.tvvaranda.setText("#Varanda:"+p.getVaranda());
        mHolder.tvrestroom.setText("#Attached_restroom:"+p.getRestroom());
        mHolder.tvlift.setText("#Lift:"+p.getLift());
        mHolder.tvgenerator.setText("#Generator:"+p.getGenerator());
        mHolder.tvphoneNo.setText("#Contact:"+p.getPhoneNo());
        mHolder.tvdetail.setText("#Details:"+p.getDetail());

        return v;
    }

```



```
private class ViewHolder1 {  
  
    private TextView tvpostType;;  
    private TextView tvcity;  
    private TextView tvarea;  
    private TextView tvgender;  
    private TextView tvroomSize;  
    private TextView tvvaranda;  
    private TextView tvrestroom;  
    private TextView tvlift;  
    private TextView tvgenerator;  
    private TextView tvphoneNo;  
    private TextView tvdetail;  
    }  
}
```

BIBLIOGRAPHY

1. <https://dev.twitter.com/rest/public/search>
2. <https://stackoverflow.com>
3. <https://shikkhok.com>