ChihYung Wu

4.19.2016

Assignment 5

The deliverables are:

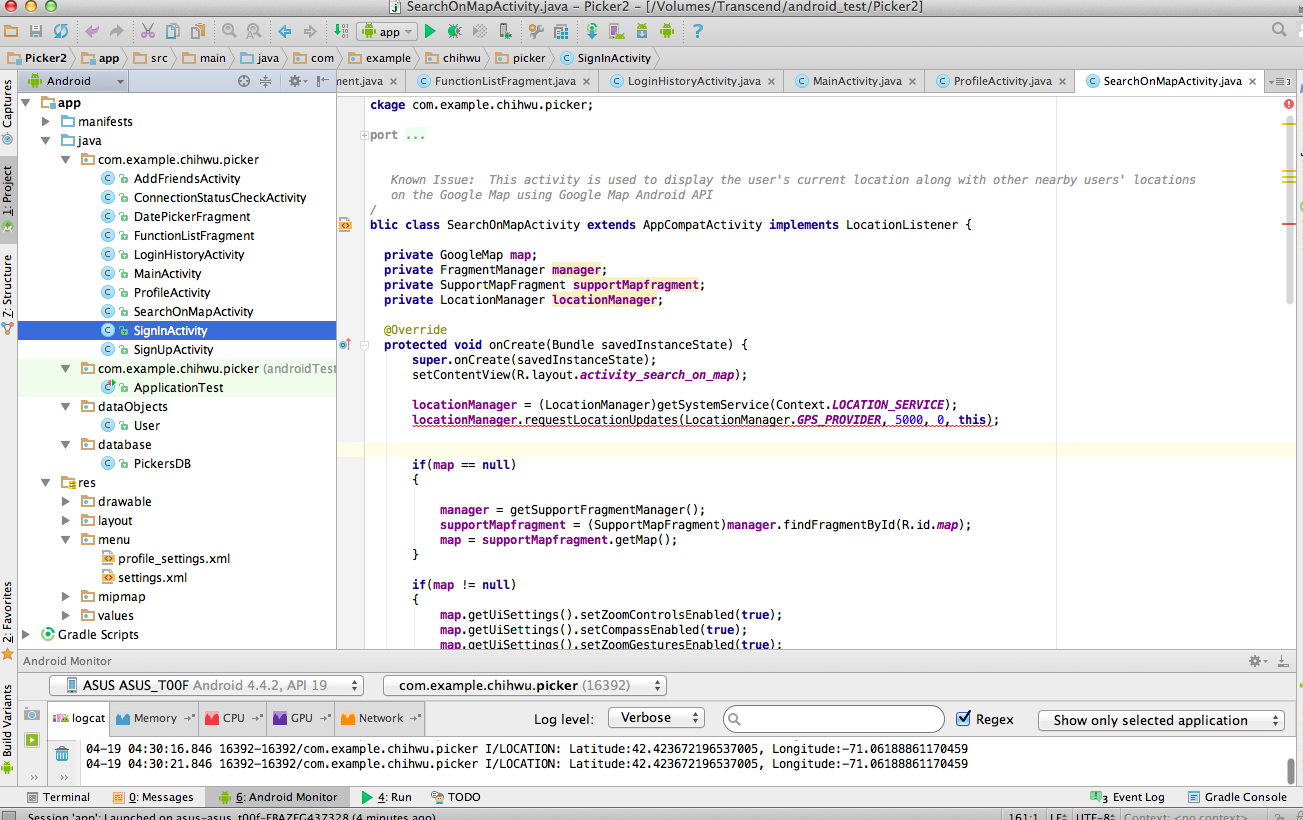
# Part 1—Overall Project Description:

Provide a paragraph (no more) outlining the Android application you’d like to build for this course. Don’t worry about being overly ambitious or not completely specific to begin with: we hold you only to the specific requirements in Part 3.

# This will be a cool-friend-making application that will allow its users to quickly find and make friends with someone in their proximity. In order to be able to use this application, the user will have to create an account by inputting their username, password, date-of-birth, introduction. The results will display the info of all the users in that user’s proximity.

# Currently, users can use Google Map Android API to locate his/her location on the map. Some static data of other users have been entered to display the friend locating functions of the app. Later on, the data of other users will be saved on the database for a more practical purpose.

# Part 2—Screenshot of the directory structure (project or package):

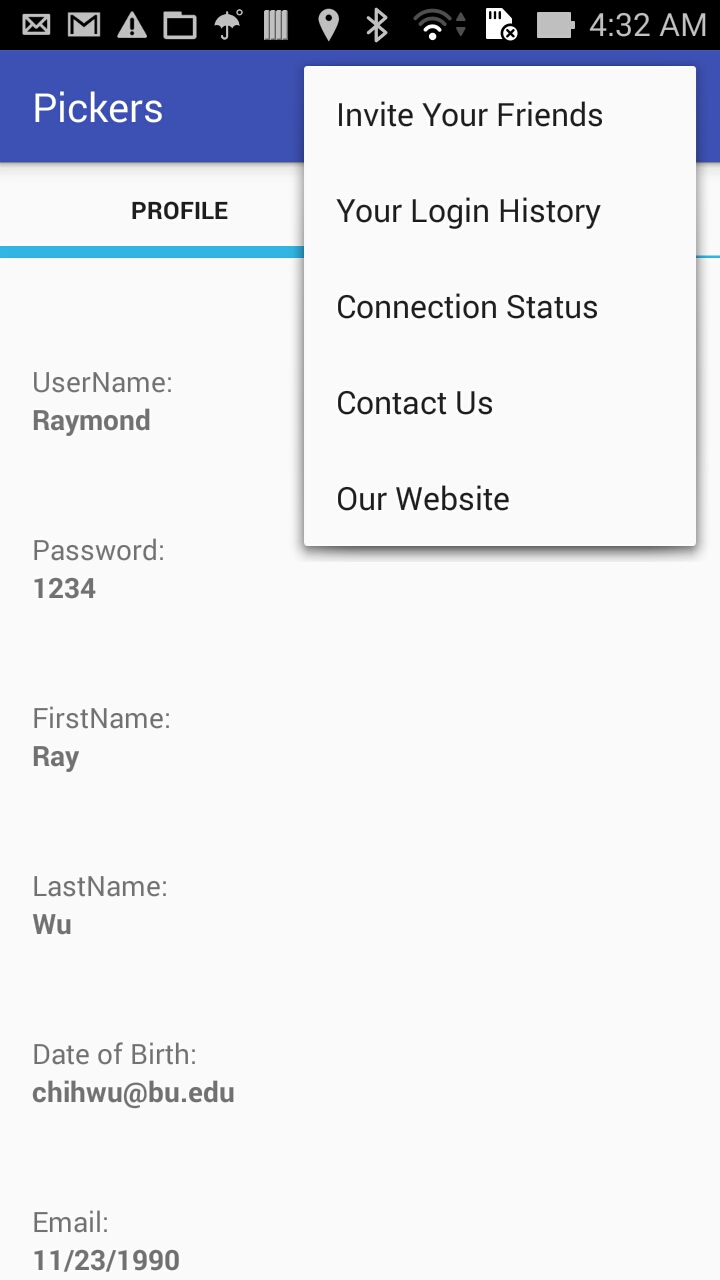
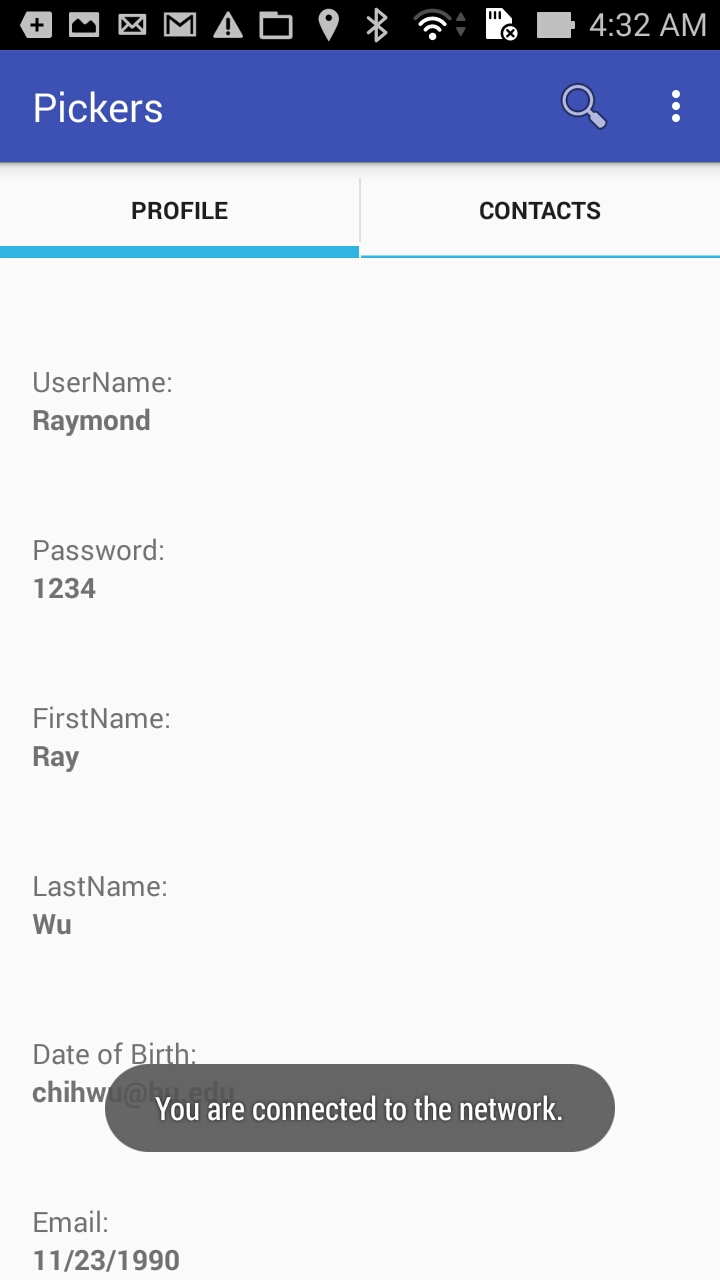


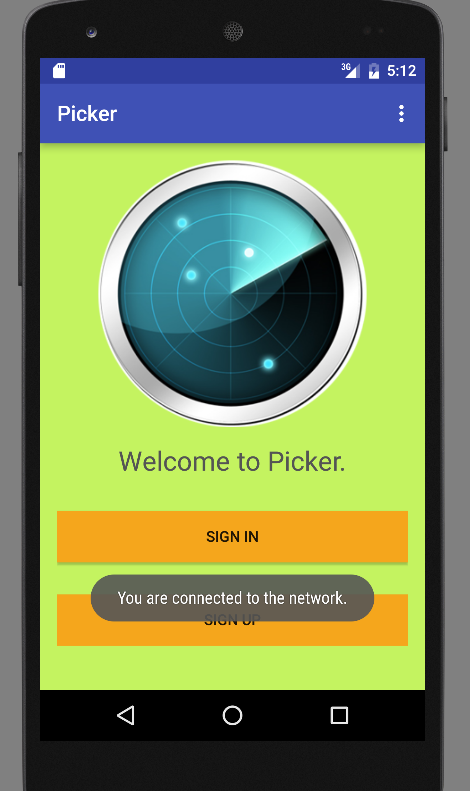
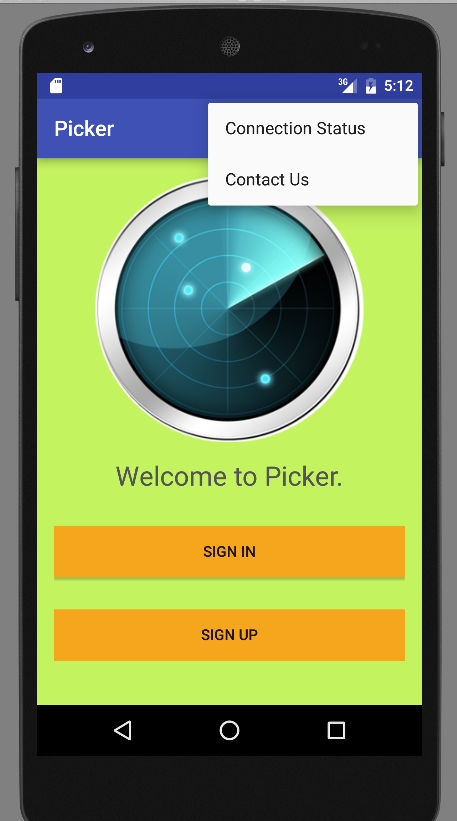
# Part 3—Application Features with output and tests: *(maximum 2 pages of 12-point text, including figures)*

Specify the features of your application that you implemented. The priority is to get practice with as many as possible of the Android constructs covered in Module 1. Number your features 1, 2, 3, … . Each feature must accompanied by output or screen shot that show you accomplished it.

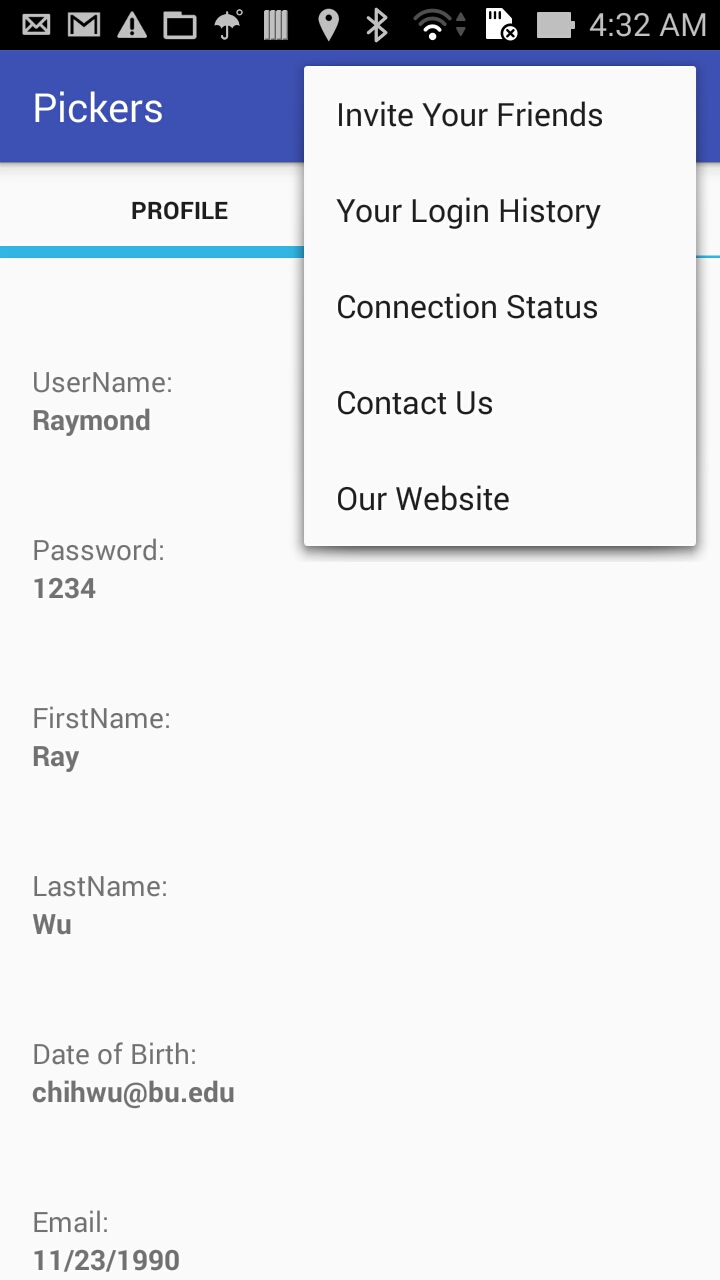
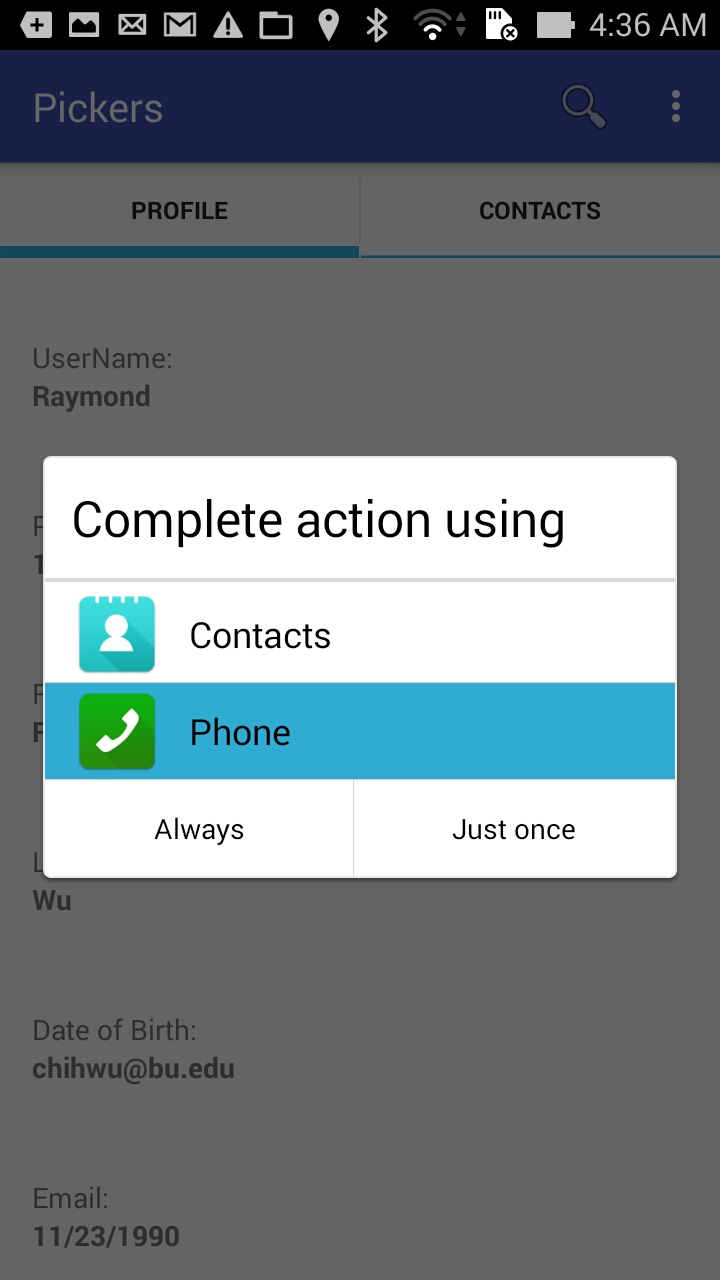
(New Features based on elements covered in this week’s module)

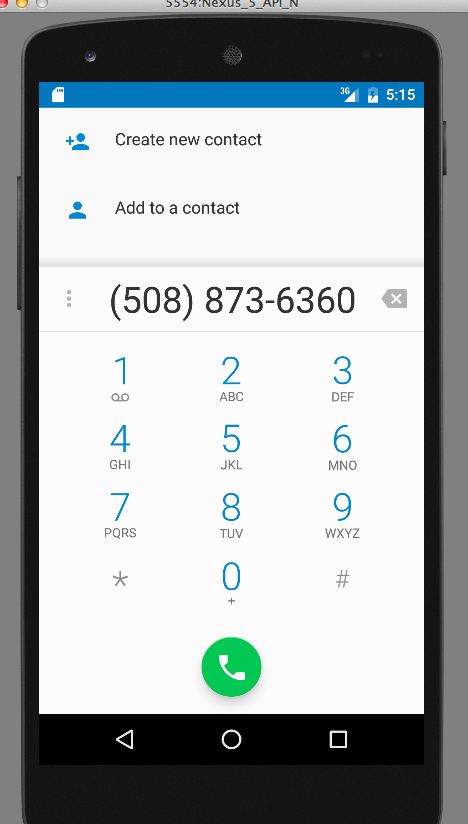
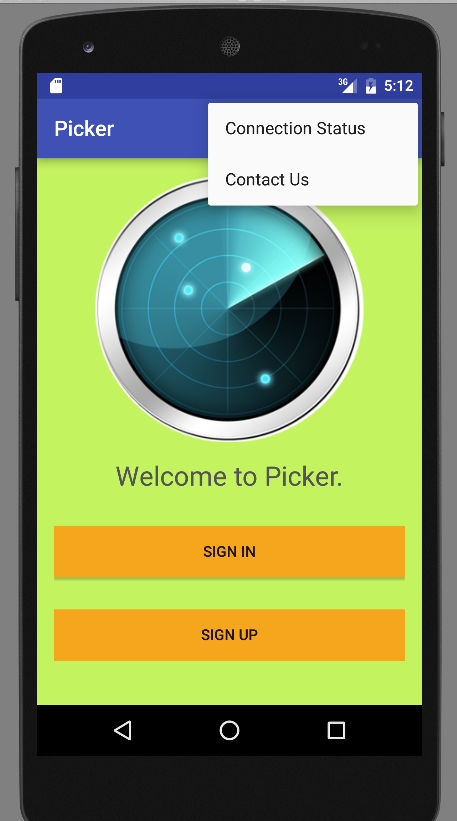
Application Feature 1: (Users now can check their network access condition both on the Home page and on the profile page after they log in when they click “Connection Status”.)

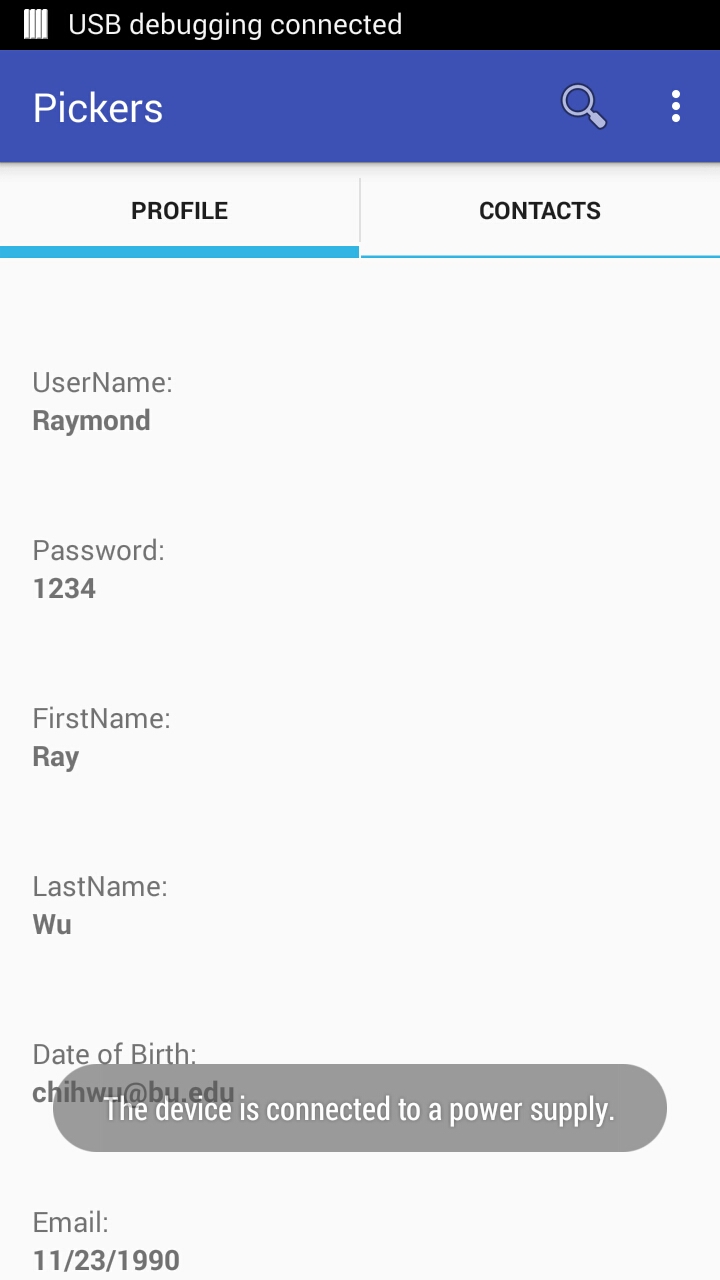


Application Feature 2: (Users now can check their telephony access condition both on the Home page and on the profile page after they log in and dial my phone number when they click “Contact Us”)

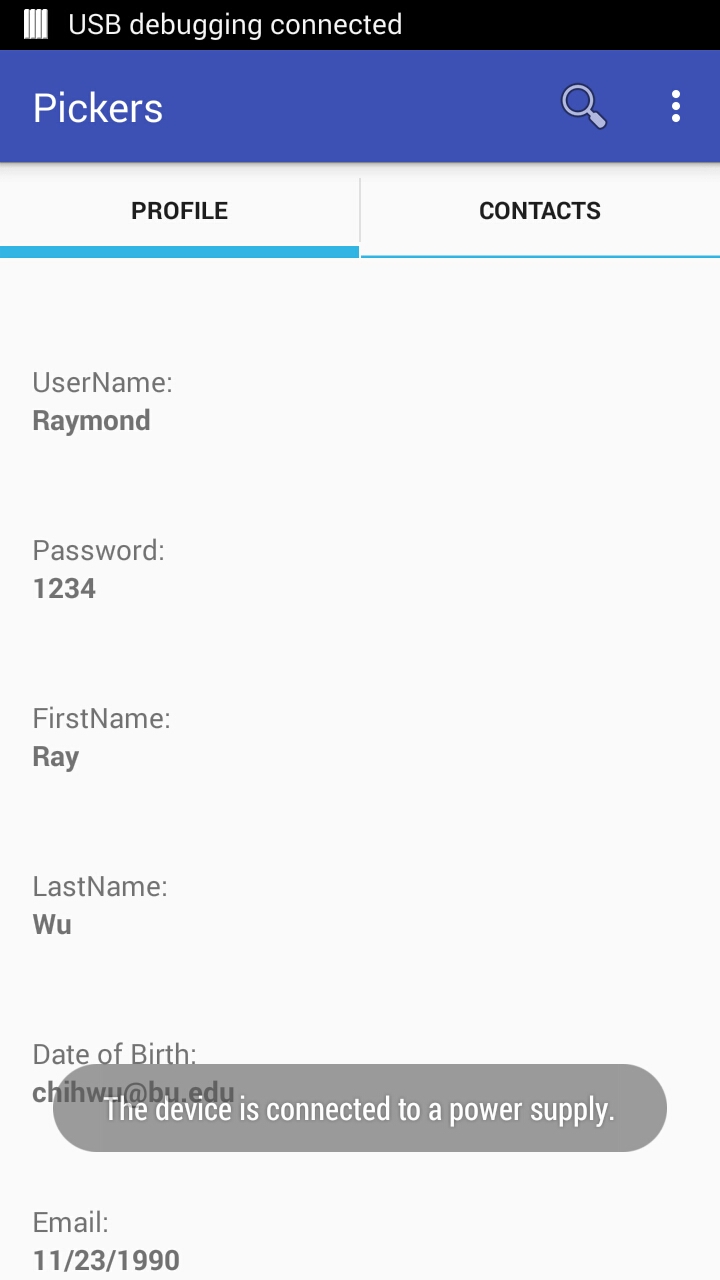
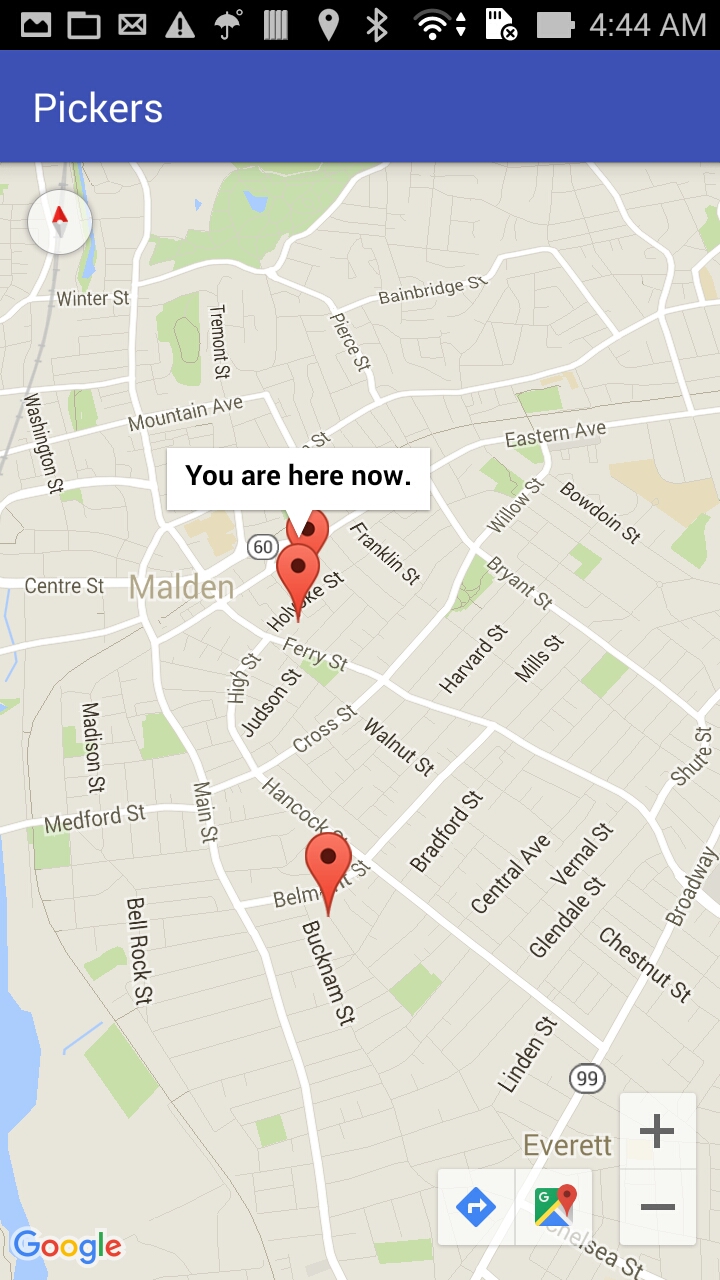
 



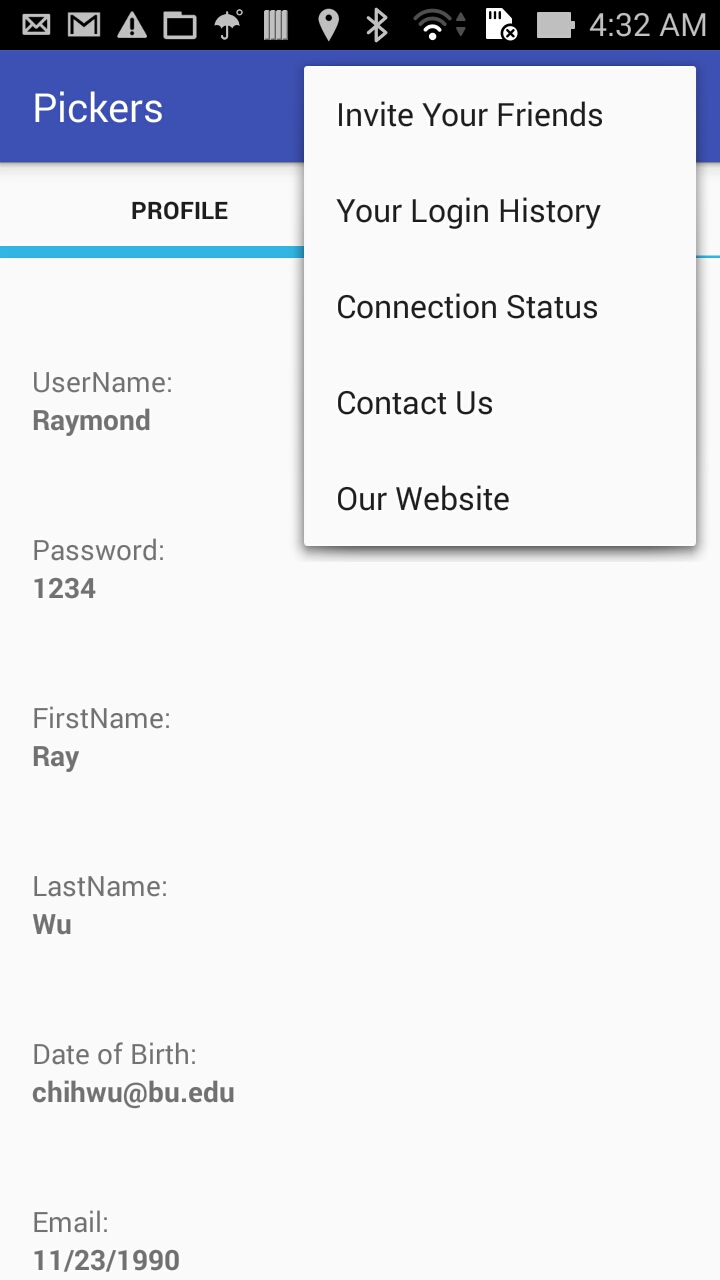
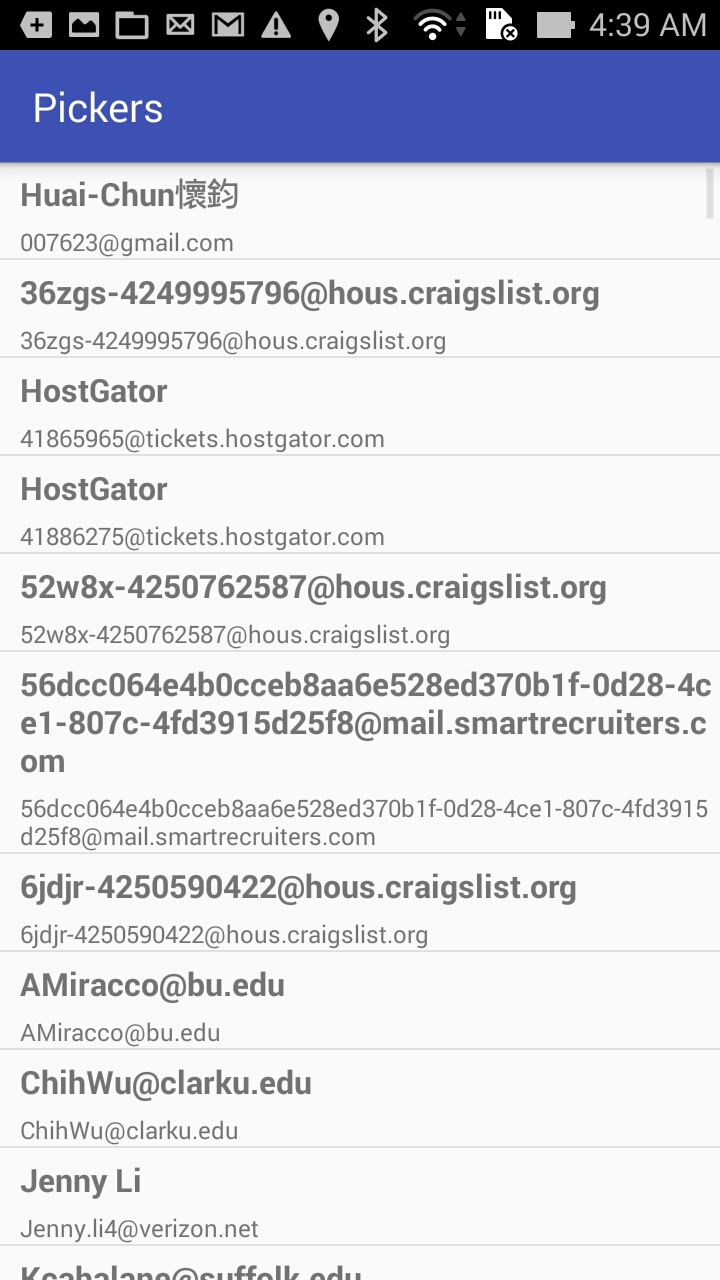
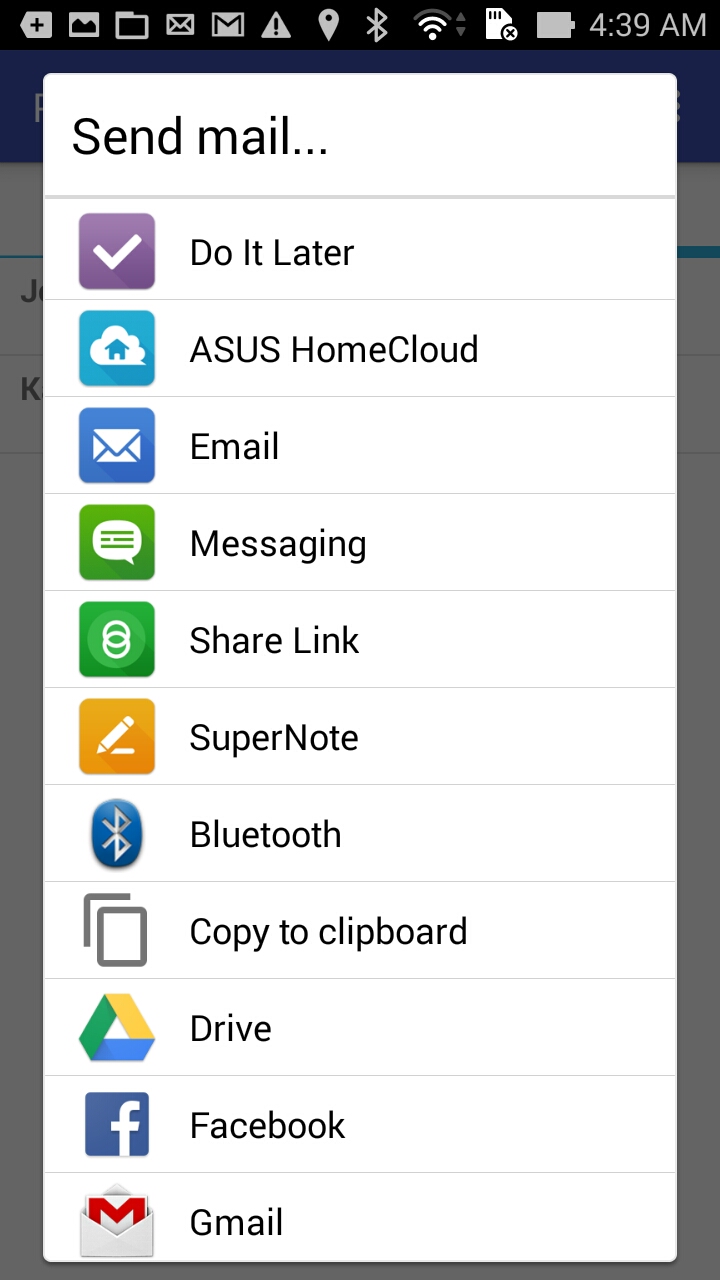
Application Feature 3: (Users can use BroadcastReceiver now to receive any notification whenever their device is charged.)

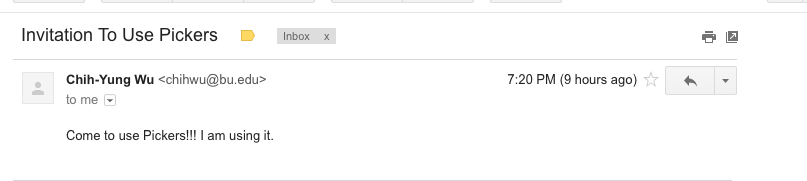


Application Feature 4: (Users can now view his/her location and other nearby users’ locations on GoogleMap using Google Map Android API when they click the Magnifier icon in the actionbar.)

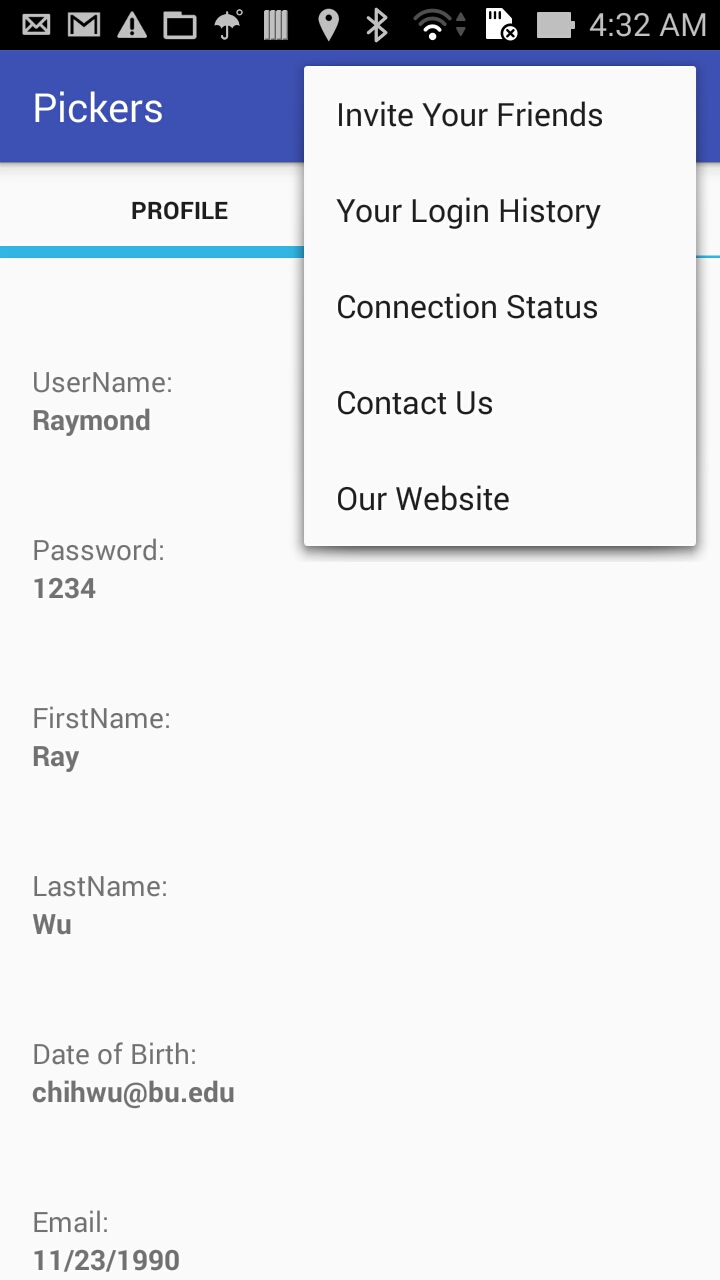
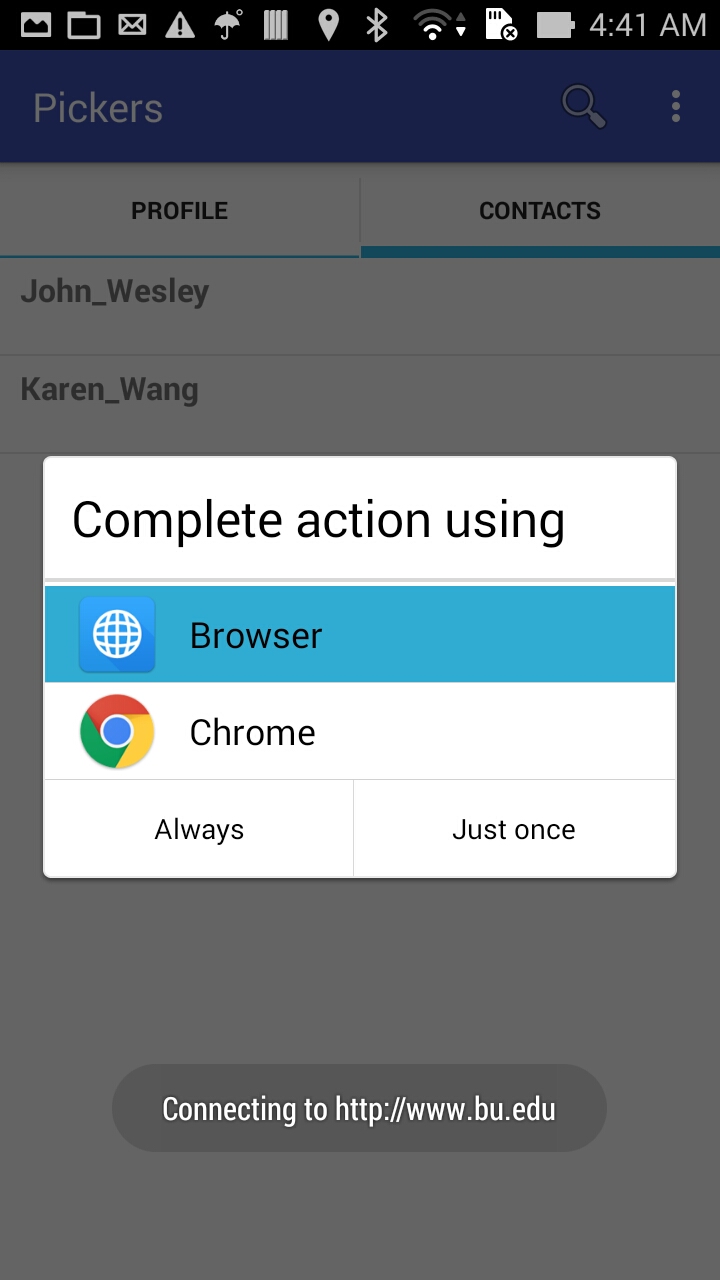
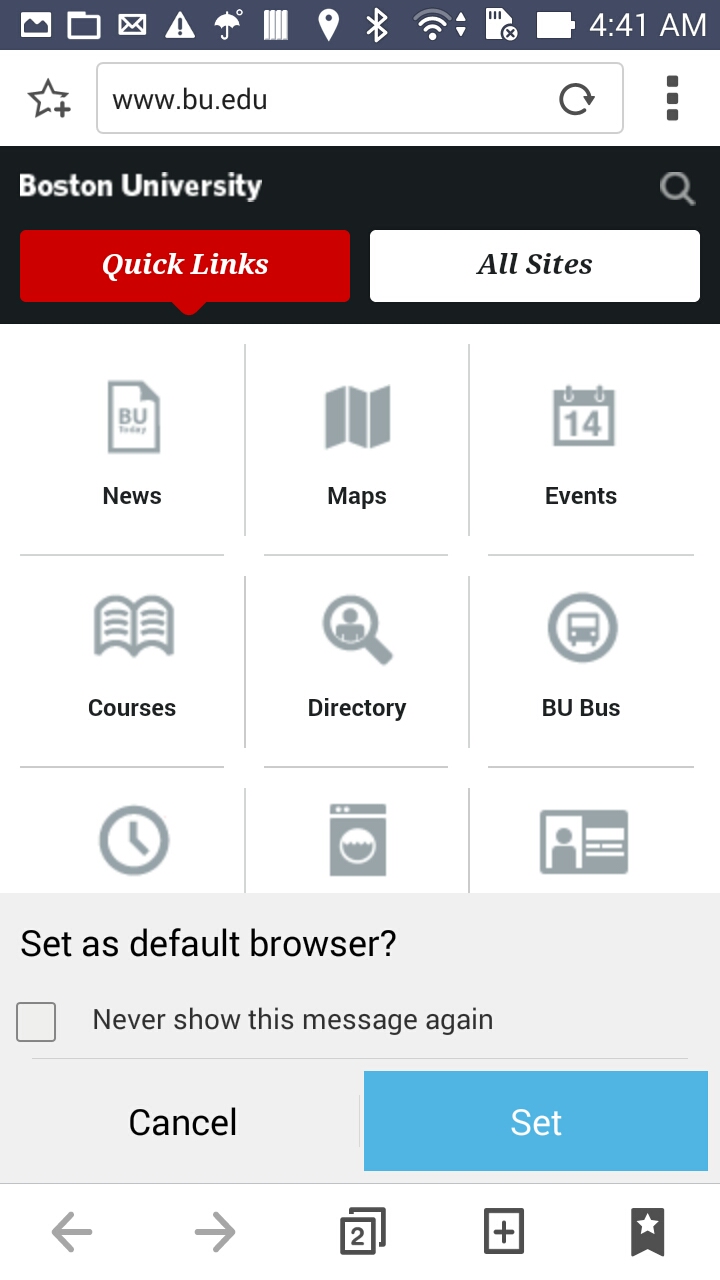
 

Application Feature 5: (Users can now send an email message to invite their friends in the Google contacts to use this application.)



Application Feature 6: (Users can now view our company’s webpage, which is our school’s site in this case.)

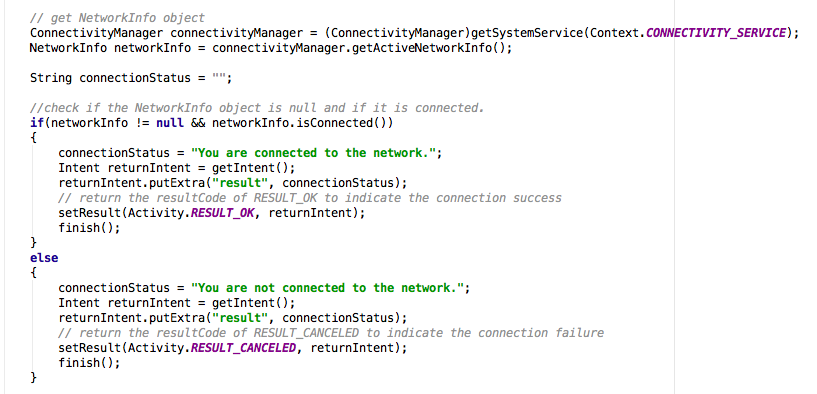
  

# Part 4—List of Android Elements: *(maximum 1 page of 12-point text)*

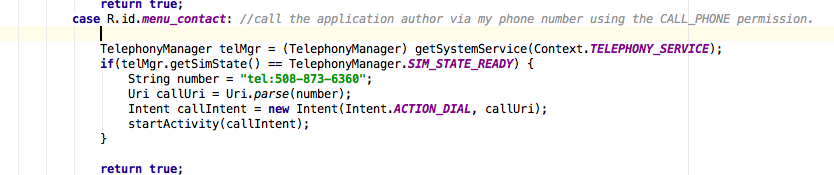
Use as many of the Android elements (e.g., built-in keywords, data types or functions) as possible covered in module 1. Provide lines from your code and where they were used once.

(New elements covered in this week’s module)

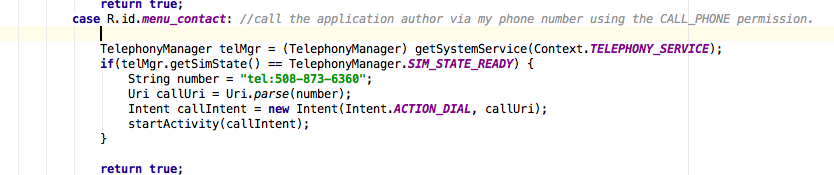
**New Element #1 (Network connectivity status checking.)**



**New Element #2 (Telephony condition checking. )**

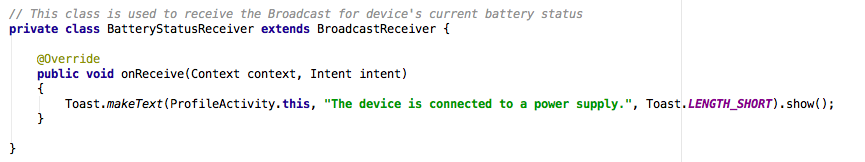


**New Element #3 (Making a phone call. )**

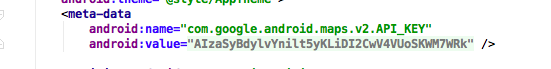


**New Element #4 (Use of Broadcast Receiver )**



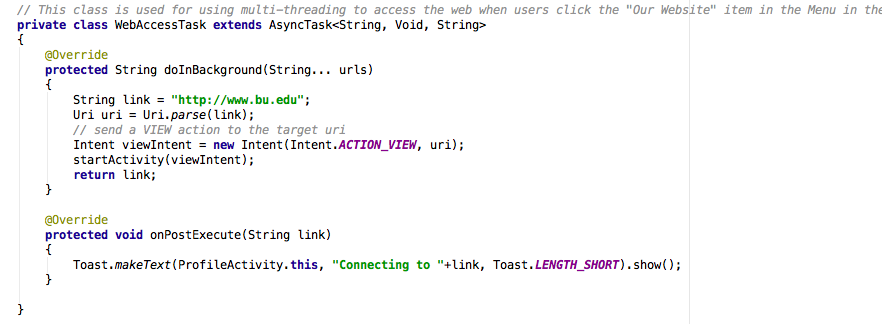


**New Element #5 (Use of Google Map Android API )**

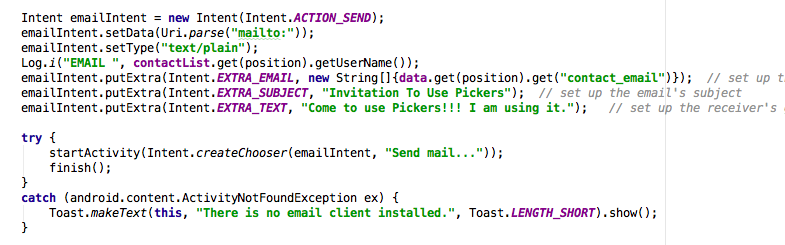




**New Element #6 (Use of multithreading )**



**New Element #7 (Use of Intent for email client on Android for sending emails)**

****

# Part 5—Example Code

Provide a page—in 10-point Ariel Narrow font—of your best (clearly documented)[[1]](#footnote-1) code. You may precede it with a paragraph (not counted in the page limit) explaining where and how it fits with the rest of your project.

The code I selected is the one in the ProfileActivity.java file as it now contains the logic of the routing in my application. All the major functionalities of this application, such as email sending, Display Google Map, etc, all are passed through here.

**ProfileActivity.java**

**package** com.example.chihwu.picker;  
  
**import** android.annotation.TargetApi;  
**import** android.app.Activity;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.IntentFilter;  
**import** android.location.LocationManager;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.Html;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.AdapterView;  
**import** android.widget.TextView;  
**import** android.content.Intent;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.widget.TabHost;  
**import** android.widget.TabHost.TabSpec;  
**import** com.google.tabmanager.TabManager;  
**import** android.support.v7.app.ActionBarActivity;  
**import** android.widget.Toast;  
**import** android.widget.AdapterView.OnItemClickListener;  
**import** android.view.View;  
**import** android.location.Location;  
**import** android.location.LocationManager;  
**import** android.location.LocationListener;  
**import** android.telephony.TelephonyManager;

*//Known issues: Profile Activity contains the FunctionListFragment and now displays an action bar for menu*  
**public class** ProfileActivity **extends** AppCompatActivity **implements** OnItemClickListener{  
  
 **private** TextView **username\_textview**;  
 **private** TextView **profile\_greeting\_textview**;  
 **private** TabHost **tabHost**;  
 **private** TabManager **tabManager**;  
  
 **private** BatteryStatusReceiver **batteryStatusReceiver**; *//A Broadcast instance used to detect the broadcast for current battery status* **private** IntentFilter **intentFilter**;  
  
  
  
 @TargetApi(11)  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_profile***);  
  
 *//get the tab set up* **tabHost** = (TabHost)findViewById(android.R.id.***tabhost***);  
 **tabHost**.setup();  
 **tabManager** = **new** TabManager(**this**, **tabHost**, R.id.***realtabcontent***);  
  
 String[] tabTitles = {**"Profile"**, **"Contacts"**};  
  
 **for**(String tabTitle : tabTitles)  
 {  
 TabSpec tabSpec = **tabHost**.newTabSpec(tabTitle);  
 tabSpec.setIndicator(tabTitle);  
 **tabManager**.addTab(tabSpec, FunctionListFragment.**class**, **null**);  
 }  
  
  
  
 **if**(savedInstanceState != **null**)  
 {  
 **tabHost**.setCurrentTabByTag(savedInstanceState.getString(**"tab"**));  
 }  
  
  
  
 *// I created an actionbar for displaying menu in this activity as the tabs function used in this activity prevents a default actionbar from displaying  
 // change the color of the default action bar to our dark blue theme* getSupportActionBar().setTitle(Html.*fromHtml*(**"<font color='#FFFFFF'>Pickers </font>"**));  
  
  
 **batteryStatusReceiver** = **new** BatteryStatusReceiver(); *// instantiate our inner class BatteryStatusReceiver for receiving the broadcast for the device's battery status update* **intentFilter** = **new** IntentFilter(**"android.intent.action.ACTION\_POWER\_CONNECTED"**);  
  
 registerReceiver(**batteryStatusReceiver**, **intentFilter**); *// register our Broadcase Receiver for specific "ACTION\_POWER\_CONNECTED" Intent action* }  
  
  
 @Override  
 **protected void** onSaveInstanceState(Bundle outState)  
 {  
 **super**.onSaveInstanceState(outState);  
 outState.putString(**"tab"**, **tabHost**.getCurrentTabTag());  
 }  
  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***profile\_settings***, menu);  
 **super**.onCreateOptionsMenu(menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item)  
 {  
 **switch** (item.getItemId())  
 {  
 **case** R.id.***search***: *// accessing the SearchOnMapActivity for displaying the map* Intent searchOnMapIntent = **new** Intent(**this**, SearchOnMapActivity.**class**);  
 startActivity(searchOnMapIntent);  
 **return true**;  
 **case** R.id.***login\_history***: *// accessing the LoginHistoryActivity for showing login history* Intent checkLoginHistoryIntent = **new** Intent(**this**, LoginHistoryActivity.**class**);  
 startActivity(checkLoginHistoryIntent);  
 **return true**;  
 **case** R.id.***invite\_friends***: *// accessing the AddFriendsActivity for contacts list* Intent invitationIntent = **new** Intent(**this**, AddFriendsActivity.**class**);  
 startActivity(invitationIntent);  
 **return true**;  
 **case** R.id.***menu\_contact***: *//call the application author via my phone number using the CALL\_PHONE permission.* TelephonyManager telMgr = (TelephonyManager) getSystemService(Context.***TELEPHONY\_SERVICE***);  
 **if**(telMgr.getSimState() == TelephonyManager.***SIM\_STATE\_READY***) {  
 String number = **"tel:508-873-6360"**;  
 Uri callUri = Uri.*parse*(number);  
 Intent callIntent = **new** Intent(Intent.***ACTION\_DIAL***, callUri);  
 startActivity(callIntent);  
 }  
  
 **return true**;  
 **case** R.id.***menu\_connection***: *// show the network connection status when selected* Intent connectionStatusCheckIntent = **new** Intent(**this**, ConnectionStatusCheckActivity.**class**);  
 startActivityForResult(connectionStatusCheckIntent, 1);  
 **return true**;  
 **case** R.id.***our\_website***: *// access the website of our school* **new** WebAccessTask().execute();  
 **return true**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 }  
  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
  
 **if** (requestCode == 1) *// the requestCode of 1 is for network connection status checking* {  
 **if**(resultCode == Activity.***RESULT\_OK***) *// if the connection is successful* {  
 String result=data.getStringExtra(**"result"**);  
 Toast.*makeText*(**this**, result, Toast.***LENGTH\_LONG***).show();  
 }  
 **if** (resultCode == Activity.***RESULT\_CANCELED***) *// if the connection is not successful* {  
 String result=data.getStringExtra(**"result"**);  
 Toast.*makeText*(**this**, result, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 }  
  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View v, **int** position, **long** id)  
 {  
  
 }  
  
 *// This class is used to receive the Broadcast for device's current battery status* **private class** BatteryStatusReceiver **extends** BroadcastReceiver {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent)  
 {  
 Toast.*makeText*(ProfileActivity.**this**, **"The device is connected to a power supply."**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
  
 *// This class is used for using multi-threading to access the web when users click the "Our Website" item in the Menu in the Actionbar* **private class** WebAccessTask **extends** AsyncTask<String, Void, String>  
 {  
 @Override  
 **protected** String doInBackground(String... urls)  
 {  
 String link = **"http://www.bu.edu"**;  
 Uri uri = Uri.*parse*(link);  
 *// send a VIEW action to the target uri* Intent viewIntent = **new** Intent(Intent.***ACTION\_VIEW***, uri);  
 startActivity(viewIntent);  
 **return** link;  
 }  
  
 @Override  
 **protected void** onPostExecute(String link)  
 {  
 Toast.*makeText*(ProfileActivity.**this**, **"Connecting to "**+link, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
}

# Below is the matrix that will be used to evaluate your response:



A regular A translates as 95, A-=93, B+=87, B=85, B-=83, C+=77, C=75, C-=73, D+=67, F=0 etc. To get an A grade for the course, your weighted average should be >93. A-:>=90. B+:>=87. B:>83. B-:>=80. C+:>=77. C:>73. C-:>=70 etc.

1. Document intentions—don’t paraphrase code. Nontrivial functions should have (an informal) *Intent* statement*,* (precise) *Preconditions* (if any), *Returns* (if any),and *Postconditions* (always). Each block of code should be preceded by its intended objectives. [↑](#footnote-ref-1)