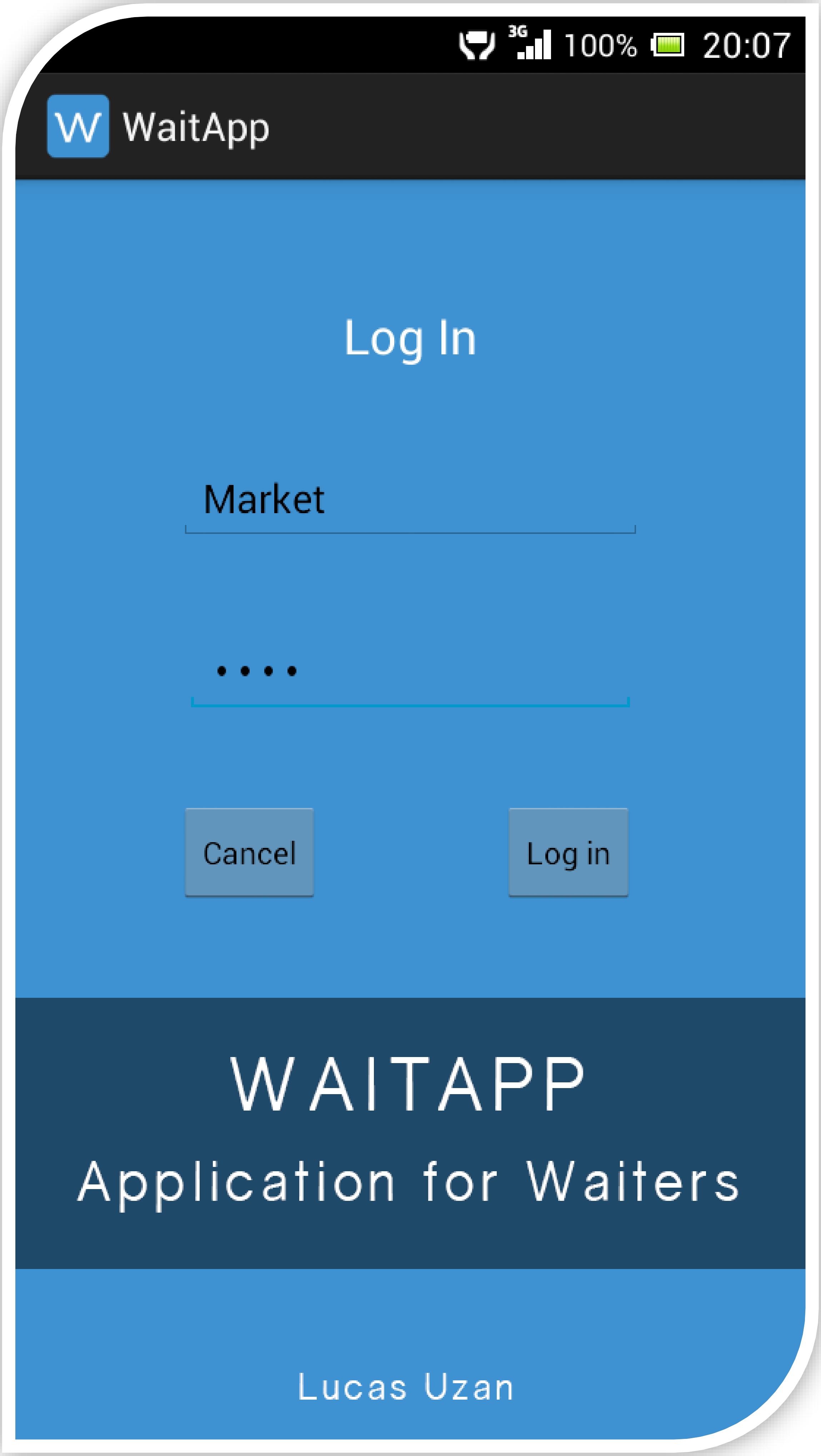


Log In

Market

••••

# WAITAPP

Application for Waiters

Lucas Uzan

# Mobile Assignment Two

UZAN|COM3|lucas.uzan@mycit.ie

1. Design and implement an Android App for a waitperson taking orders in a restaurant. The app should include the following:

(a) Basic functionality that addresses App's purpose: at least 3 Activities and navigation between them. Simple UI components e.g. TextView, EditText & Button.

(b) Use of 5 different layout managers e.g. LinearLayout H, LinearLayout V, TableLayout, RelativeLayout, FrameLayout etc.

(c) Use of the notification manager and a pending intent.

(d) Include 5 different types of UI widgets e.g. e.g. Spinner, radiobutton, imagebutton, Switch, ListView

(e) Menus with icons (no need for multiple resolutions) and 1 menu item shown/hidden depending on App state. Contextual menus & trigger.

(10 marks ea, 50 marks total)

2. Three Advanced features or innovation/creativity. Here are some examples but you can implement different ones if you want:

(f) A screen with a layout created pragmatically and four widgets. Two variables saved using javaio or sql.

(g) Gestures with own gesture library. Flings.

(h) API functionality not covered in lectures nor in assignment 1. E.G. home screen widgets, Fragments, Action bar, Navigation Drawer etc.

(10 marks ea, 30 marks total)

3. Documentation includes the following:

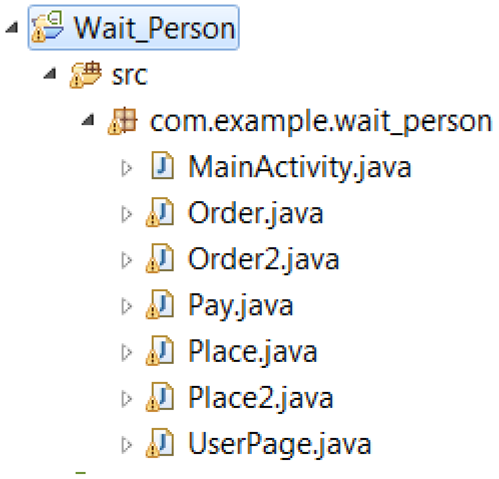
a) Commented code & the use a coding standard. Use extra classes where appropriate.

b) This document (odt | pdf | doc<x>}) stating what done including screen shots of code snippets & the running app code (on pages 3+). App Specification (~½ page). List of classes and what each is for. Evaluation (~½ page in 3 bullet points): how has it worked out, what could be done differently & future outlook.

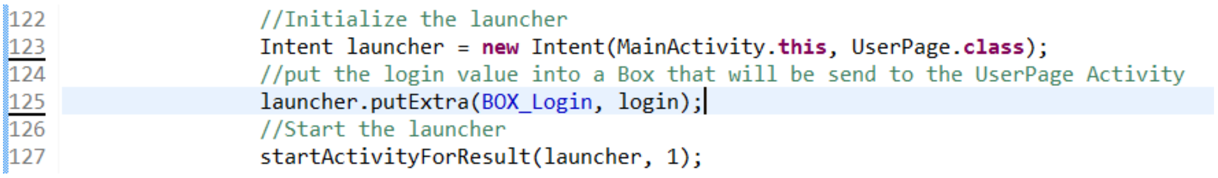
(10 marks ea, 20 marks total)

1) A) Basic functionality that addresses App’s purpose Completed? Yes

a. 3 Activities + Navigation Between Them



```
Wait_Person
  src
    com.example.wait_person
      MainActivity.java
      Order.java
      Order2.java
      Pay.java
      Place.java
      Place2.java
      UserPage.java
```

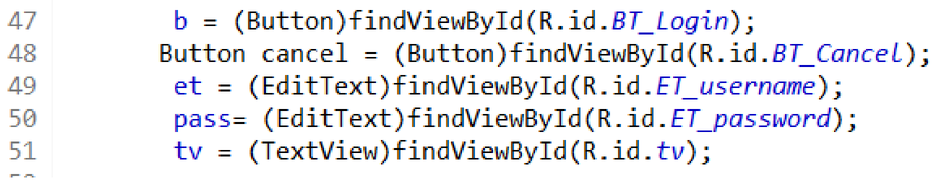


```
        //Initialize the launcher
        Intent launcher = new Intent(MainActivity.this, UserPage.class);
        //put the login value into a Box that will be send to the UserPage Activity
        launcher.putExtra(BOX_Login, login);
        //Start the launcher
        startActivityForResult(launcher, 1);
```

*GOAL : LAUNCH USERPAGE FROM MAINACTIVITY AND PASS THE LOGIN*

*SOURCE : MAINACTIVITY.JAVA*

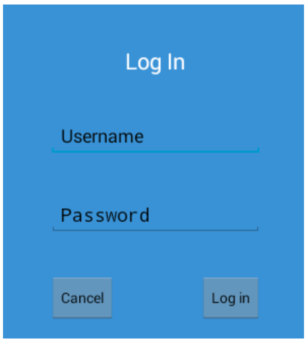
b. TextView, EditText, Button

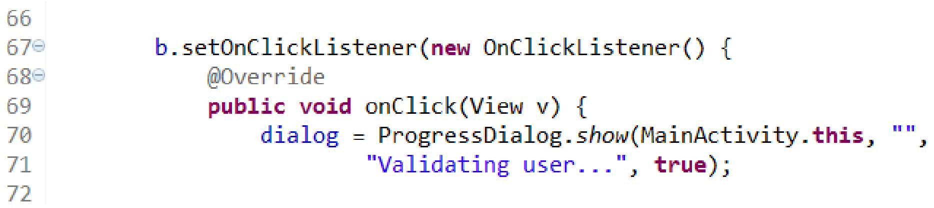


```
         b = (Button)findViewById(R.id.BT_Login);
        Button cancel = (Button)findViewById(R.id.BT_Cancel);
         et = (EditText)findViewById(R.id.ET_username);
         pass= (EditText)findViewById(R.id.ET_password);
         tv = (TextView)findViewById(R.id.tv);
```

*GOAL : DECLARE VARIABLE IDENTIFIED BY COMPONENTS TO SET EVENT*

*SOURCE : MAINACTIVITY.JAVA*





```

        b.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                dialog = ProgressDialog.show(MainActivity.this, "",
                        "Validating user...", true);

```

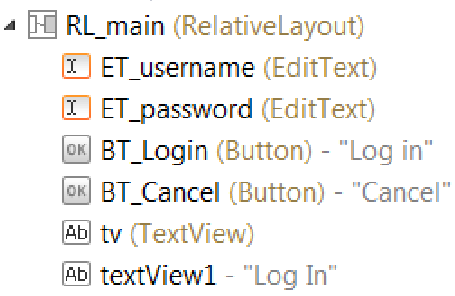
*GOAL : SET EVENT ONCLICK BUTTON*

*APPARENCE DE LA MAINACTIVITY*

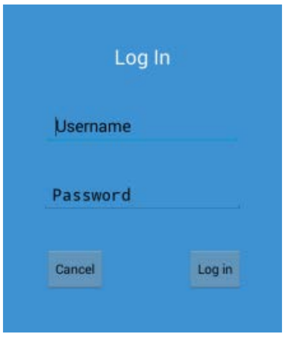
## B) Use of 5 different layout managers

Completed? Yes

### a) RelativeLayout

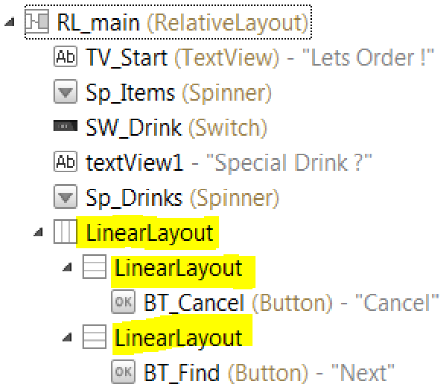


- RL_main (RelativeLayout)
  - ET_username (EditText)
  - ET_password (EditText)
  - BT_Login (Button) - "Log in"
  - BT_Cancel (Button) - "Cancel"
  - tv (TextView)
  - textView1 - "Log In"

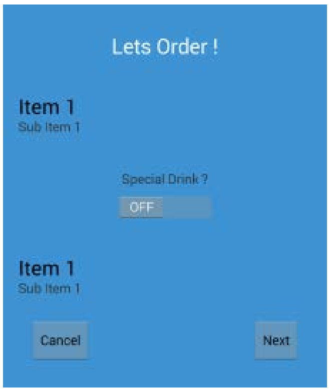


*SOURCE : ACTIVITY_MAIN.XML*

### b & c) LinearLayout Horizontal <u>AND</u> Vertical

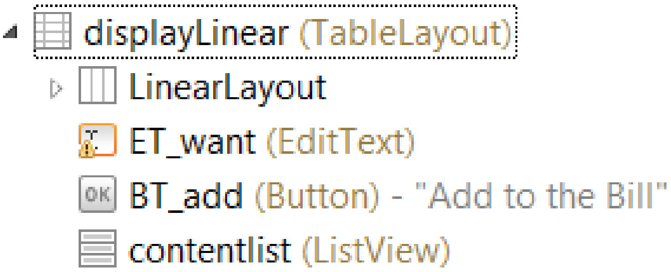


- RL_main (RelativeLayout)
  - TV_Start (TextView) - "Lets Order !"
  - Sp_Items (Spinner)
  - SW_Drink (Switch)
  - textView1 - "Special Drink ?"
  - Sp_Drinks (Spinner)
  - LinearLayout
    - LinearLayout
      - BT_Cancel (Button) - "Cancel"
    - LinearLayout
      - BT_Find (Button) - "Next"

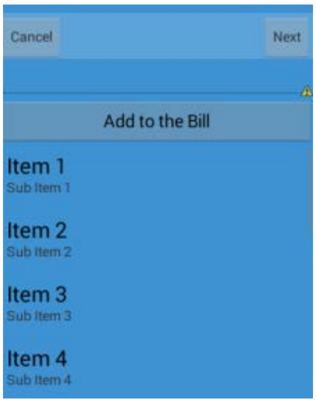


*SOURCE : ORDER.XML*

### d) TableLayout

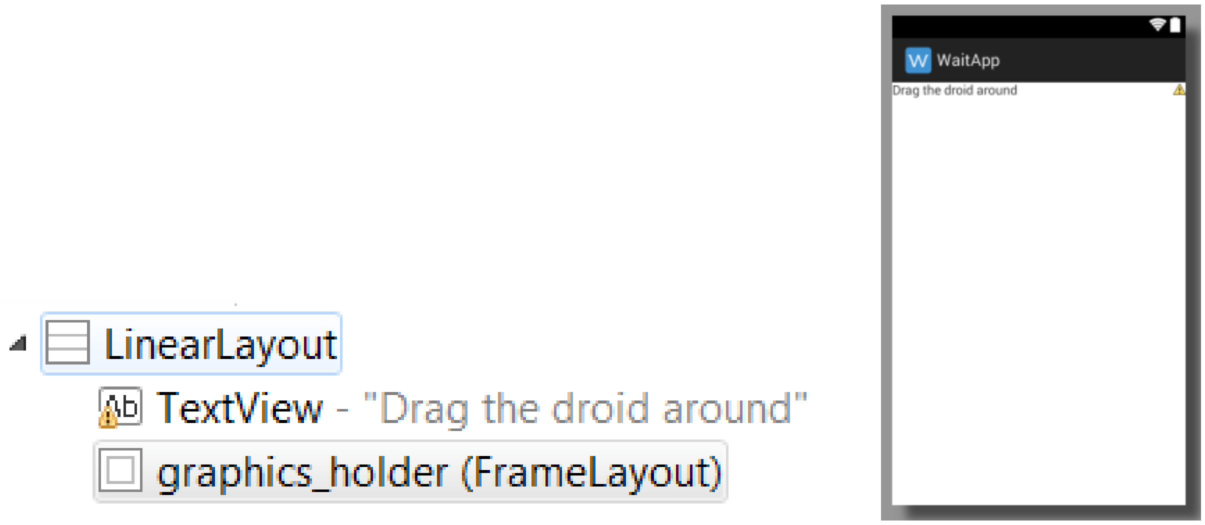


- displayLinear (TableLayout)
  - LinearLayout
  - ET_want (EditText)
  - BT_add (Button) - "Add to the Bill"
  - contentlist (ListView)



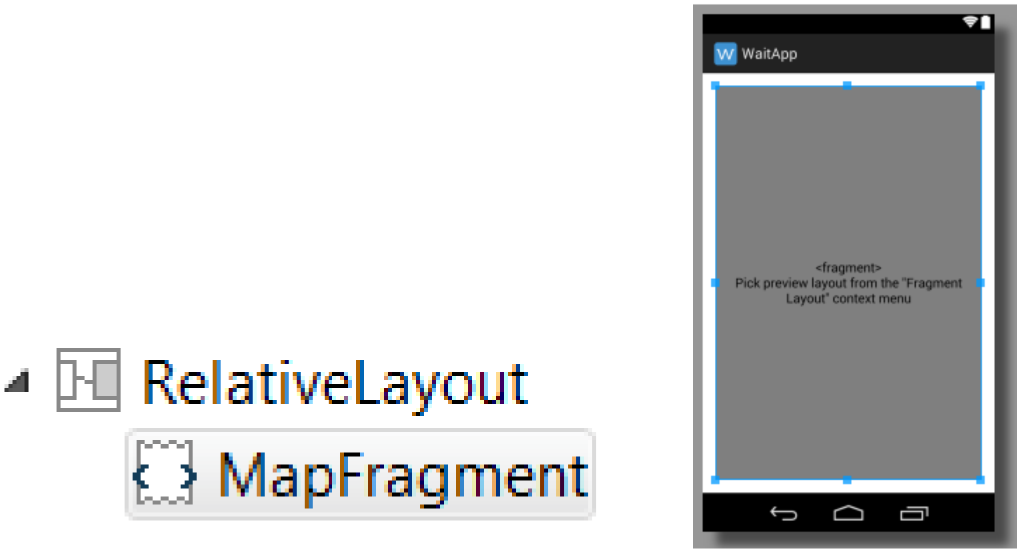
*SOURCE : ORDER2.XML*

e) FrameLayout


LinearLayout
TextView - "Drag the droid around"
graphics_holder (FrameLayout)
WaitApp
Drag the droid around

*SOURCE : PLACE.XML*

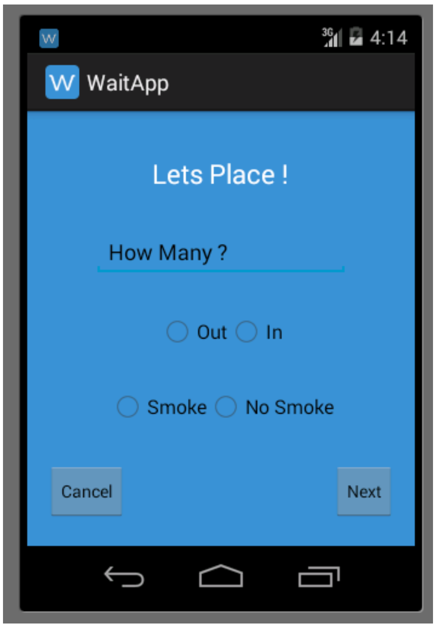
f) FragmentLayout

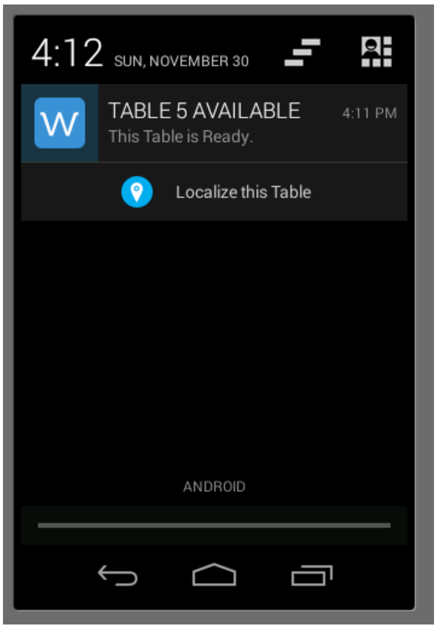


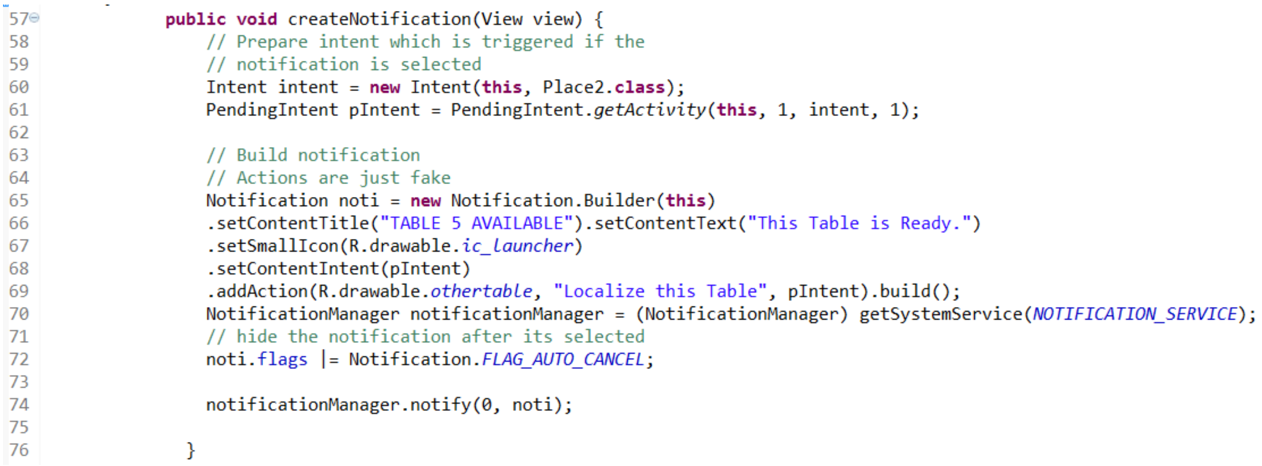
*SOURCE : PLACE.2*

## c) Use of the notification manager and a pending intent.

Completed? Yes







```java
public void createNotification(View view) {
    // Prepare intent which is triggered if the
    // notification is selected
    Intent intent = new Intent(this, Place2.class);
    PendingIntent pIntent = PendingIntent.getActivity(this, 1, intent, 1);

    // Build notification
    // Actions are just fake
    Notification noti = new Notification.Builder(this)
    .setContentTitle("TABLE 5 AVAILABLE").setContentText("This Table is Ready.")
    .setSmallIcon(R.drawable.ic_launcher)
    .setContentIntent(pIntent)
    .addAction(R.drawable.othertable, "Localize this Table", pIntent).build();
    NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    // hide the notification after its selected
    noti.flags |= Notification.FLAG_AUTO_CANCEL;

    notificationManager.notify(0, noti);

  }
```

*SOURCE : PLACE.JAVA*

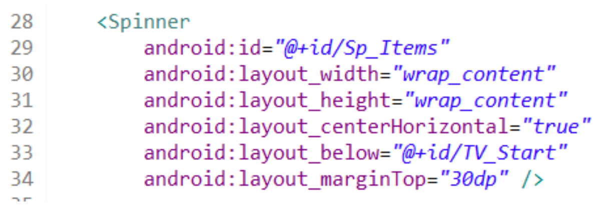
If the user click on the notification, it was supposed to open a Google Map but it doesn't work.

I explain everything in 2) h)

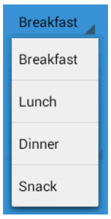
## D) Include 5 different types of UI widgets

Completed? Yes

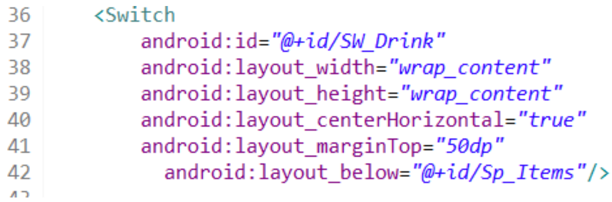
a) Spinner



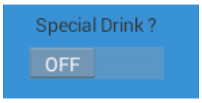
```
<Spinner
    android:id="@+id/Sp_Items"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/TV_Start"
    android:layout_marginTop="30dp" />
```



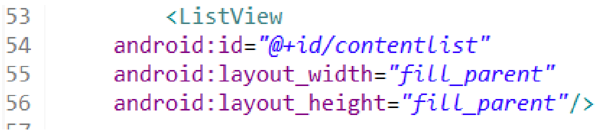
b) Switch



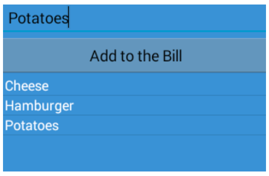
```
<Switch
    android:id="@+id/SW_Drink"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
      android:layout_below="@+id/Sp_Items"/>
```



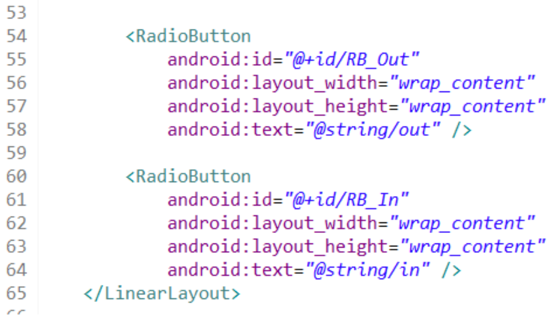
c) ListView



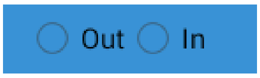
```
    <ListView
android:id="@+id/contentlist"
android:layout_width="fill_parent"
android:layout_height="fill_parent"/>
```



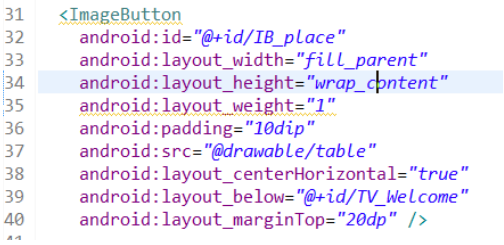
d) RadioButton



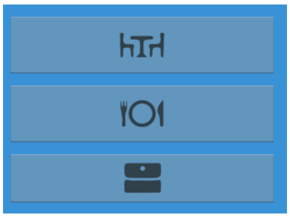
```

    <RadioButton
        android:id="@+id/RB_Out"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/out" />

    <RadioButton
        android:id="@+id/RB_In"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/in" />
</LinearLayout>
```



e) ImageButton



```
<ImageButton
  android:id="@+id/IB_place"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:padding="10dip"
  android:src="@drawable/table"
  android:layout_centerHorizontal="true"
  android:layout_below="@+id/TV_Welcome"
  android:layout_marginTop="20dp" />
```

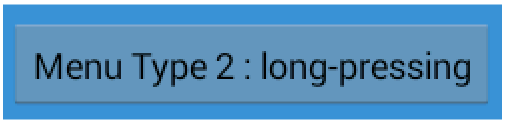


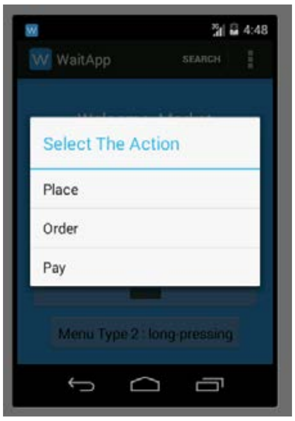
## E) Menus with icons & one menu item shown/hidden depending on App state

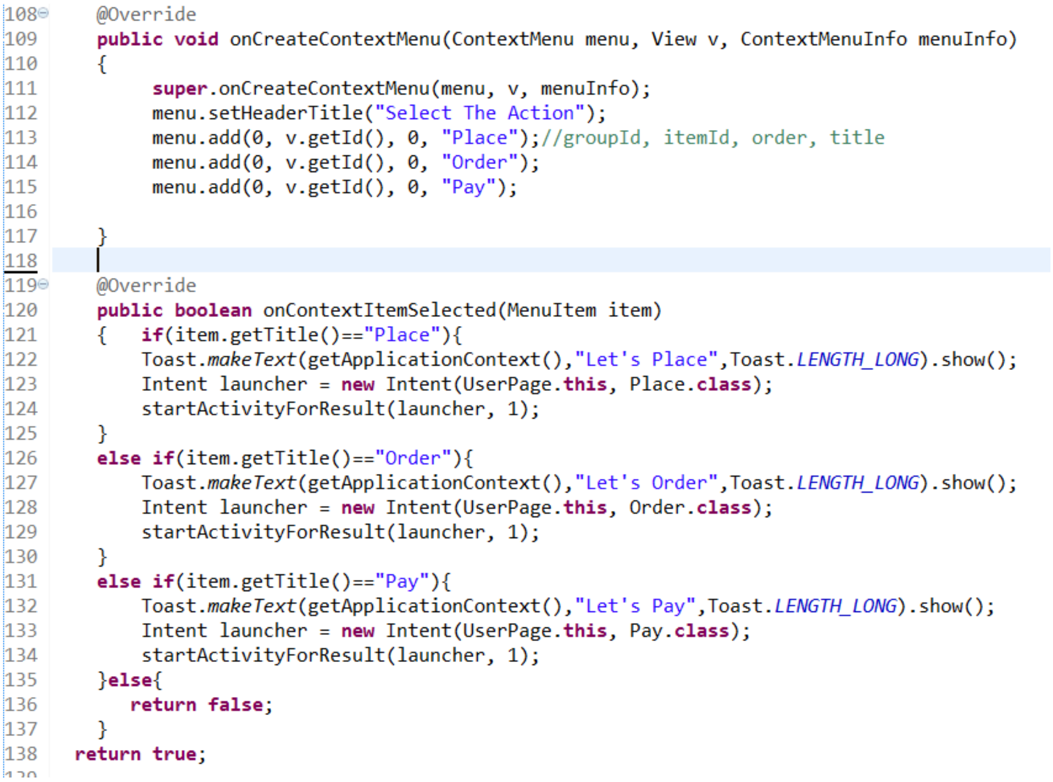
Completed? Yes

It was not really clear so I put three different menus :

- ContextMenu



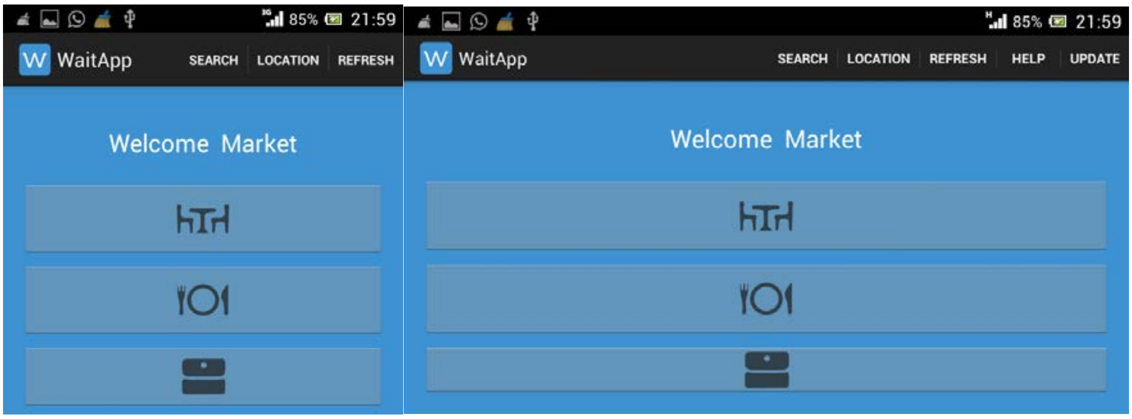


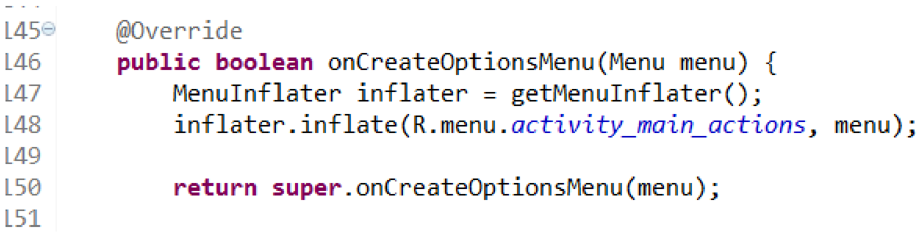


```
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo)
    {
         super.onCreateContextMenu(menu, v, menuInfo);
         menu.setHeaderTitle("Select The Action");
         menu.add(0, v.getId(), 0, "Place");//groupId, itemId, order, title
         menu.add(0, v.getId(), 0, "Order");
         menu.add(0, v.getId(), 0, "Pay");

    }

    @Override
    public boolean onContextItemSelected(MenuItem item)
    {   if(item.getTitle()=="Place"){
        Toast.makeText(getApplicationContext(),"Let's Place",Toast.LENGTH_LONG).show();
        Intent launcher = new Intent(UserPage.this, Place.class);
        startActivityForResult(launcher, 1);
    }
    else if(item.getTitle()=="Order"){
        Toast.makeText(getApplicationContext(),"Let's Order",Toast.LENGTH_LONG).show();
        Intent launcher = new Intent(UserPage.this, Order.class);
        startActivityForResult(launcher, 1);
    }
    else if(item.getTitle()=="Pay"){
        Toast.makeText(getApplicationContext(),"Let's Pay",Toast.LENGTH_LONG).show();
        Intent launcher = new Intent(UserPage.this, Pay.class);
        startActivityForResult(launcher, 1);
    }else{
       return false;
    }
  return true;
```

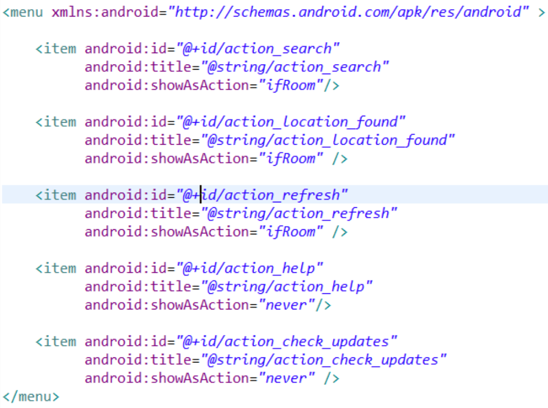
You can see the trigger to launch the next Activity depending of the item clicked.

- Menu Inflater in Action Bar (2 items hidden in vertical layout and shown in horizontal)

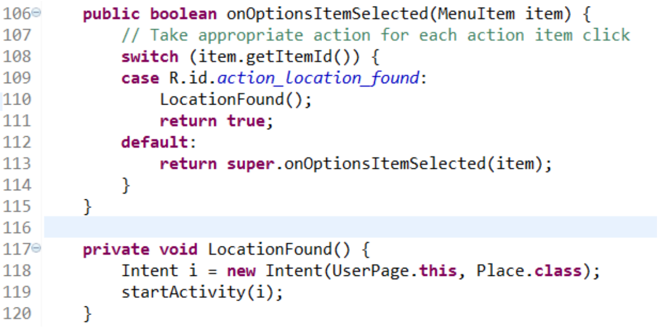




```
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.activity_main_actions, menu);

        return super.onCreateOptionsMenu(menu);
```

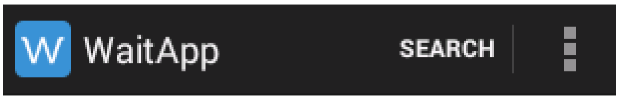


```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item android:id="@+id/action_search"
          android:title="@string/action_search"
          android:showAsAction="ifRoom"/>

    <item android:id="@+id/action_location_found"
          android:title="@string/action_location_found"
          android:showAsAction="ifRoom" />

    <item android:id="@+id/action_refresh"
          android:title="@string/action_refresh"
          android:showAsAction="ifRoom" />

    <item android:id="@+id/action_help"
          android:title="@string/action_help"
          android:showAsAction="never"/>

    <item android:id="@+id/action_check_updates"
          android:title="@string/action_check_updates"
          android:showAsAction="never" />
</menu>
```

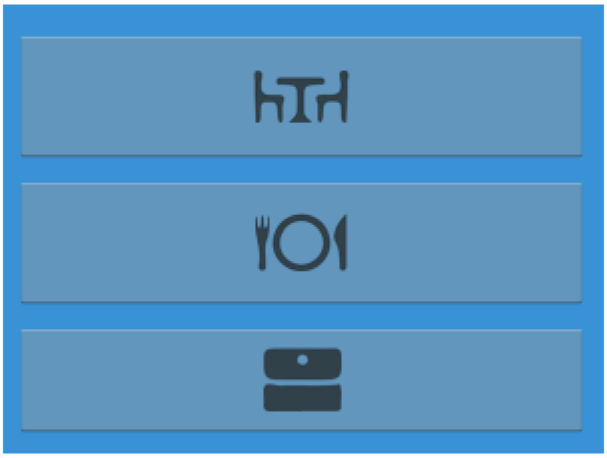


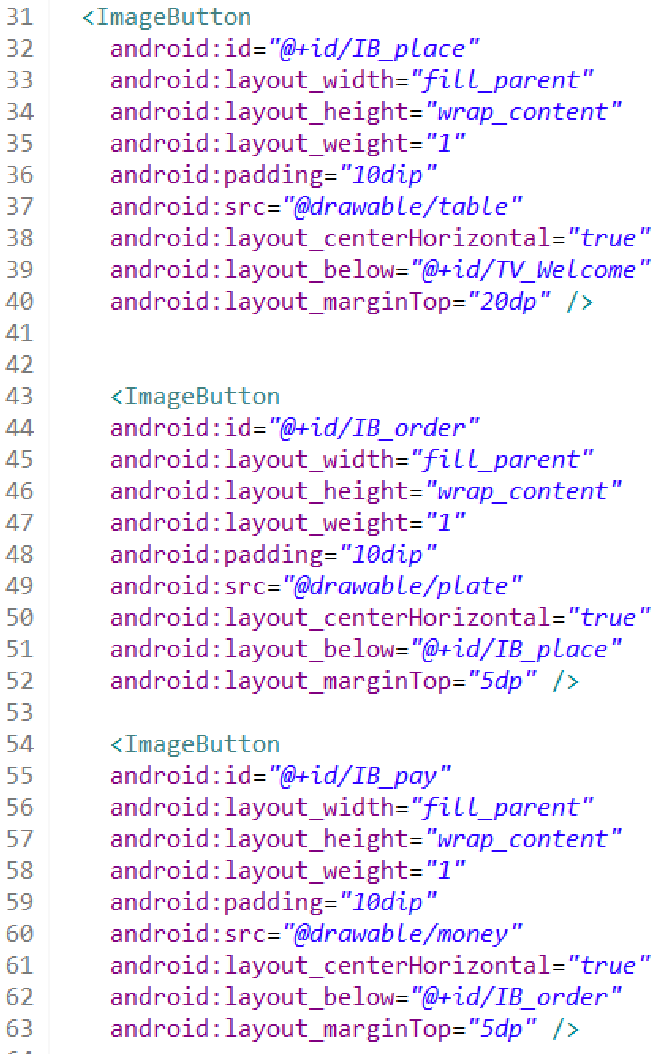
```
    public boolean onOptionsItemSelected(MenuItem item) {
        // Take appropriate action for each action item click
        switch (item.getItemId()) {
        case R.id.action_location_found:
            LocationFound();
            return true;
        default:
            return super.onOptionsItemSelected(item);
        }
    }

    private void LocationFound() {
        Intent i = new Intent(UserPage.this, Place.class);
        startActivity(i);
    }
```

NB : For the two last item of the XML, I changed the *ShowAsAction* as ifRoom otherwise we can't see them. The previous screenshots come from my mobile phone but on the emulator it's look like this :



- And finnaly, one menu simple with ImageButton



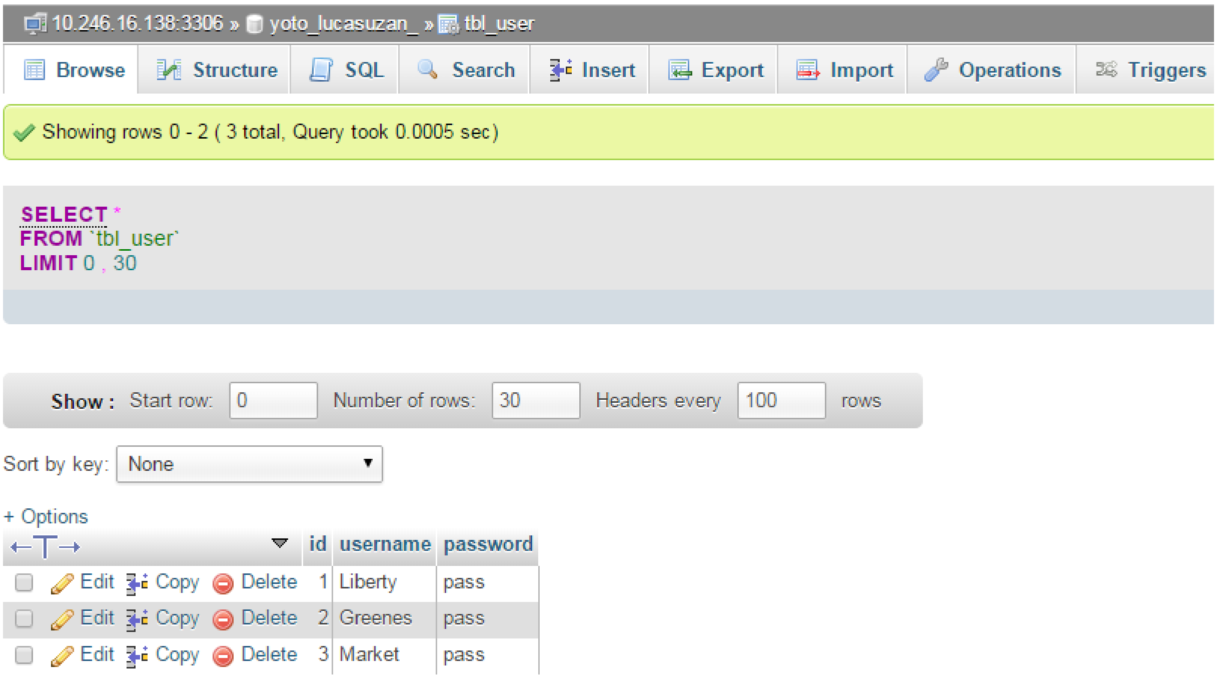


```
<ImageButton
  android:id="@+id/IB_place"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:padding="10dip"
  android:src="@drawable/table"
  android:layout_centerHorizontal="true"
  android:layout_below="@+id/TV_Welcome"
  android:layout_marginTop="20dp" />


  <ImageButton
  android:id="@+id/IB_order"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:padding="10dip"
  android:src="@drawable/plate"
  android:layout_centerHorizontal="true"
  android:layout_below="@+id/IB_place"
  android:layout_marginTop="5dp" />

  <ImageButton
  android:id="@+id/IB_pay"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:padding="10dip"
  android:src="@drawable/money"
  android:layout_centerHorizontal="true"
  android:layout_below="@+id/IB_order"
  android:layout_marginTop="5dp" />
```

## 2) Three Advanced features or innovation/creativity

### F) Login System (setup's time ≃ 8h) Completed? Yes

I was very interested by the login system and I wanted to try it. Given that I have already some databases, I created a table with 3 users & passwords:



✓ Showing rows 0 - 2 ( 3 total, Query took 0.0005 sec)

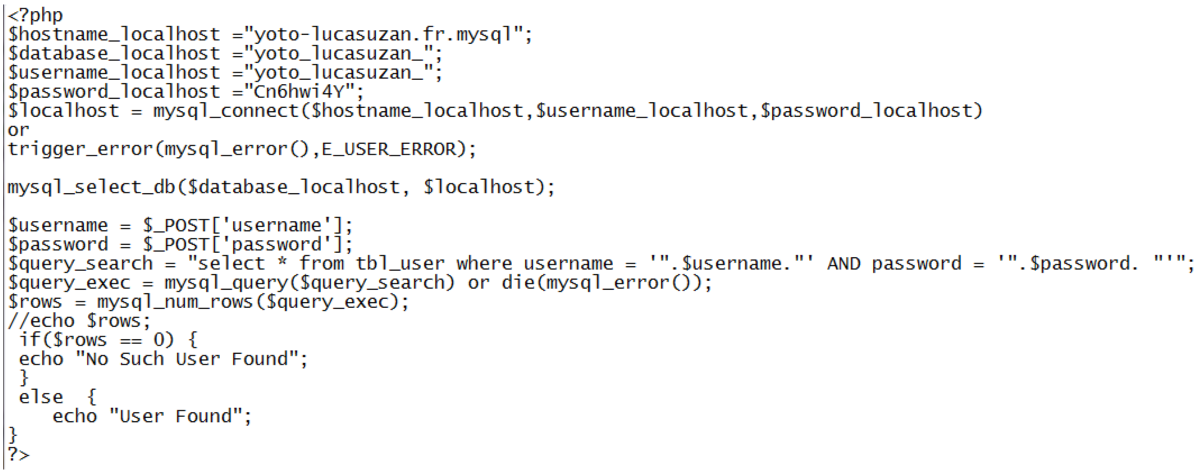
```
SELECT *
FROM `tbl_user`
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

| | | | | id | username | password |
|---|---|---|---|---|---|---|
| □ | Edit | Copy | Delete | 1 | Liberty | pass |
| □ | Edit | Copy | Delete | 2 | Greenes | pass |
| □ | Edit | Copy | Delete | 3 | Market | pass |

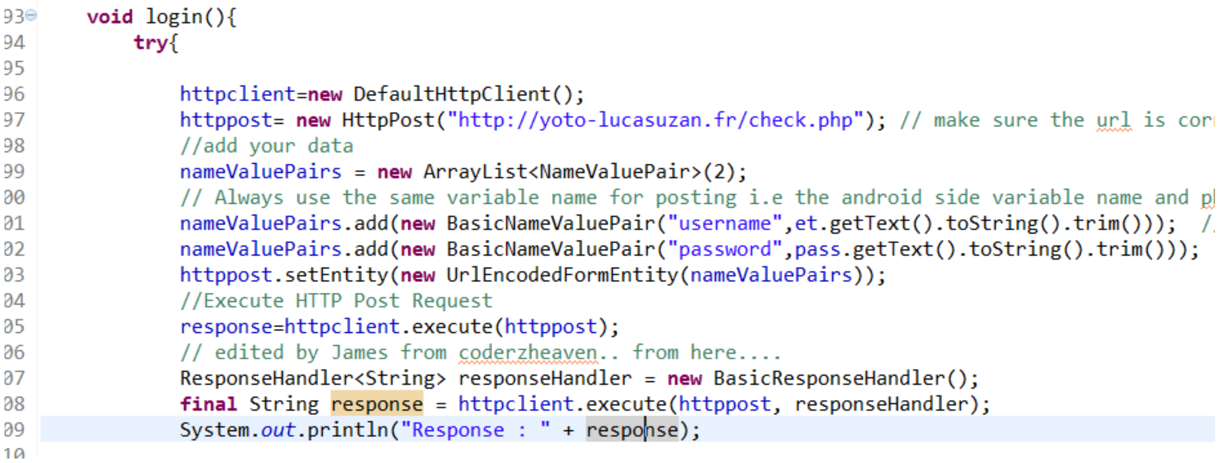
Then I create a PHP script to put on my server and that will make the connection between the database and the application:



```php
<?php
$hostname_localhost ="yoto-lucasuzan.fr.mysql";
$database_localhost ="yoto_lucasuzan_";
$username_localhost ="yoto_lucasuzan_";
$password_localhost ="Cn6hwi4Y";
$localhost = mysql_connect($hostname_localhost,$username_localhost,$password_localhost)
or
trigger_error(mysql_error(),E_USER_ERROR);

mysql_select_db($database_localhost, $localhost);

$username = $_POST['username'];
$password = $_POST['password'];
$query_search = "select * from tbl_user where username = '".$username."' AND password = '".$password. "'";
$query_exec = mysql_query($query_search) or die(mysql_error());
$rows = mysql_num_rows($query_exec);
//echo $rows;
 if($rows == 0) {
 echo "No Such User Found";
 }
 else  {
    echo "User Found";
}
?>
```

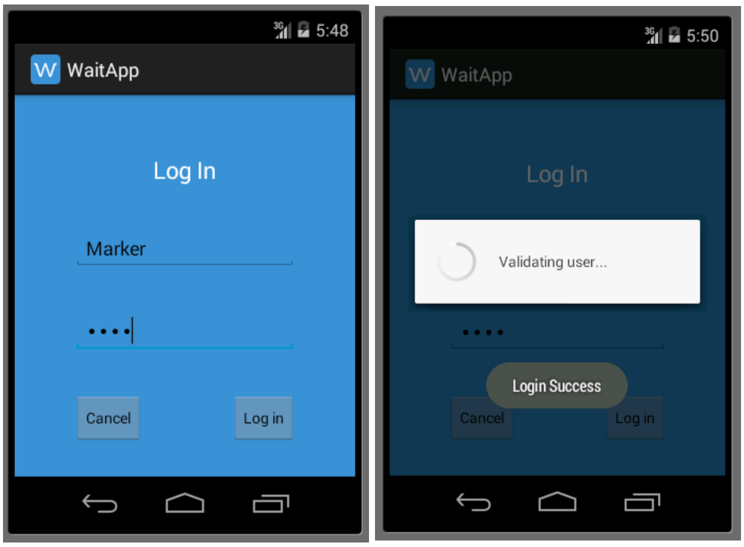
If needed, I'm able to explain the PHP script.

To finish, I put the Java code that allow the connection with the script and then recovered the response of the server if it’s the good details or not (need to correspond with the details entered in the table of the database):



```java
    void login(){
        try{

            httpclient=new DefaultHttpClient();
            httppost= new HttpPost("http://yoto-lucasuzan.fr/check.php"); // make sure the url is cor
            //add your data
            nameValuePairs = new ArrayList<NameValuePair>(2);
            // Always use the same variable name for posting i.e the android side variable name and pl
            nameValuePairs.add(new BasicNameValuePair("username",et.getText().toString().trim()));  /
            nameValuePairs.add(new BasicNameValuePair("password",pass.getText().toString().trim()));
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
            //Execute HTTP Post Request
            response=httpclient.execute(httppost);
            // edited by James from coderzheaven.. from here....
            ResponseHandler<String> responseHandler = new BasicResponseHandler();
            final String response = httpclient.execute(httppost, responseHandler);
            System.out.println("Response : " + response);
```

Correct login details (with a Toast and a ProgressDialog) :

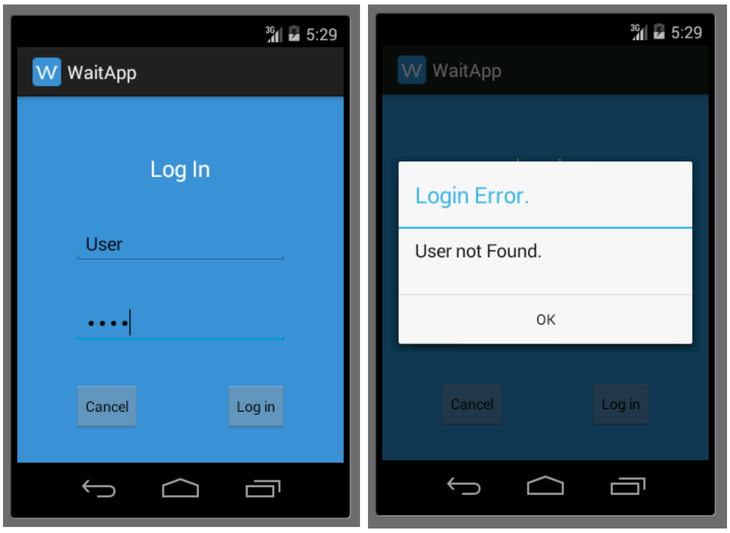


Marker

Validating user...

Login Success

Wrong Login or Password (with AlertDialog) :



User

Login Error.

User not Found.

OK

Sources of the tutorials I used are available here :

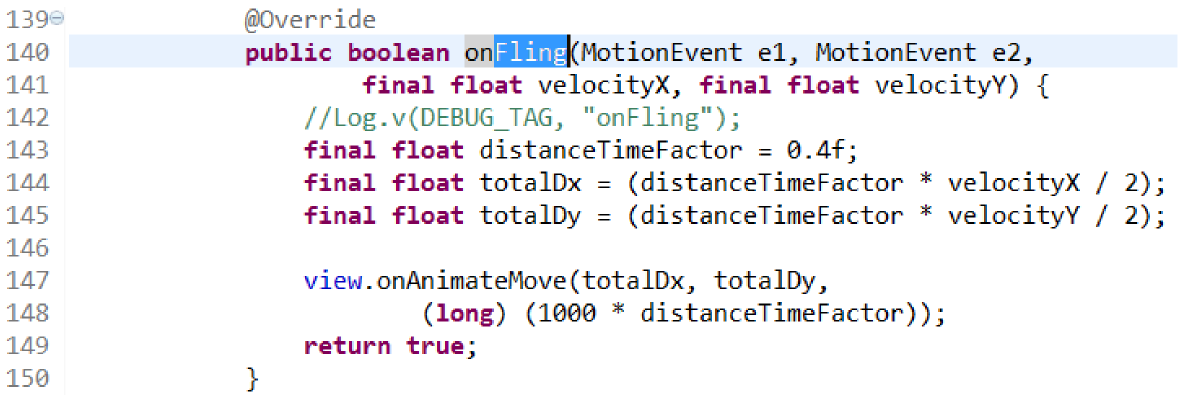
http://www.tutorialspoint.com/android/android_login_screen.htm

http://www.coderzheaven.com/2012/04/22/create-simple-login-form-php-android-connect-php-android/

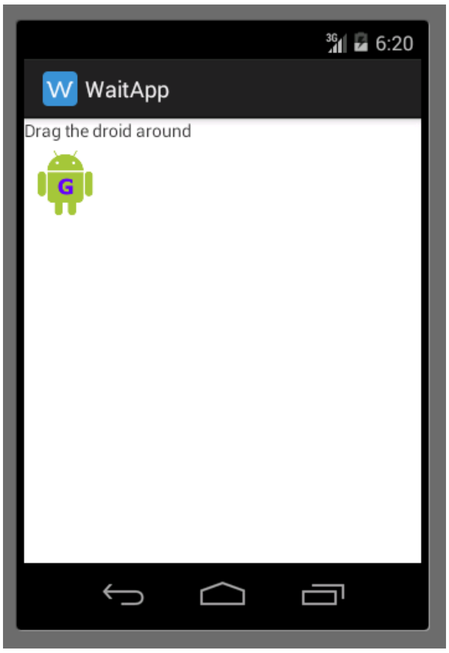
http://www.learn2crack.com/2013/08/develop-android-login-registration-with-php-mysql.html

## G) Gestures with own gesture library with Flint. (setup's time ≃ 2h)

Completed? Yes



```
        @Override
        public boolean onFling(MotionEvent e1, MotionEvent e2,
                final float velocityX, final float velocityY) {
            //Log.v(DEBUG_TAG, "onFling");
            final float distanceTimeFactor = 0.4f;
            final float totalDx = (distanceTimeFactor * velocityX / 2);
            final float totalDy = (distanceTimeFactor * velocityY / 2);

            view.onAnimateMove(totalDx, totalDy,
                    (long) (1000 * distanceTimeFactor));
            return true;
        }
```



You can move the image with your finger.

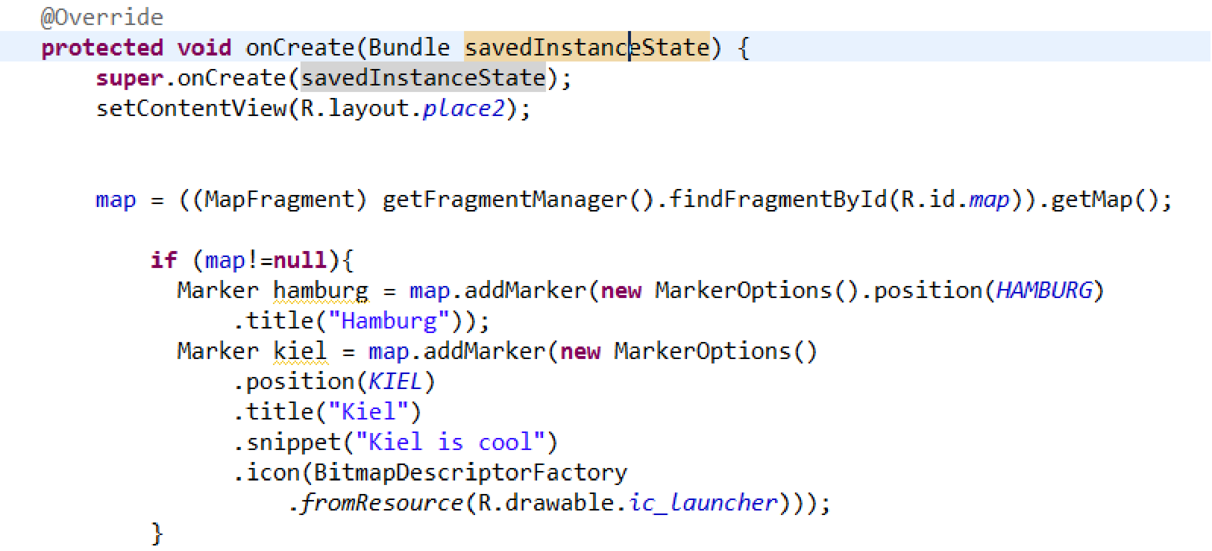
Sources of the tutorial I used are available here:

http://code.tutsplus.com/tutorials/android-sdk-introduction-to-gestures--mobile-2239

## H) Goolgle Map (setup's time ≃ 21h) Completed? Yes

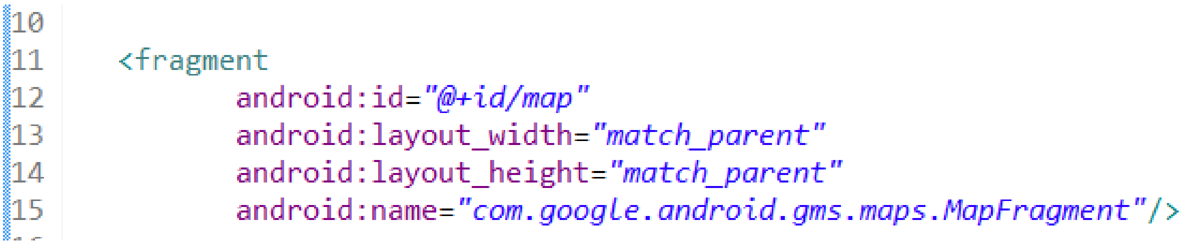
I installed the Google Map Library and generate an API key for my application (to recover the Google Map data).

I created a Java class with some parameters to initiate the map:



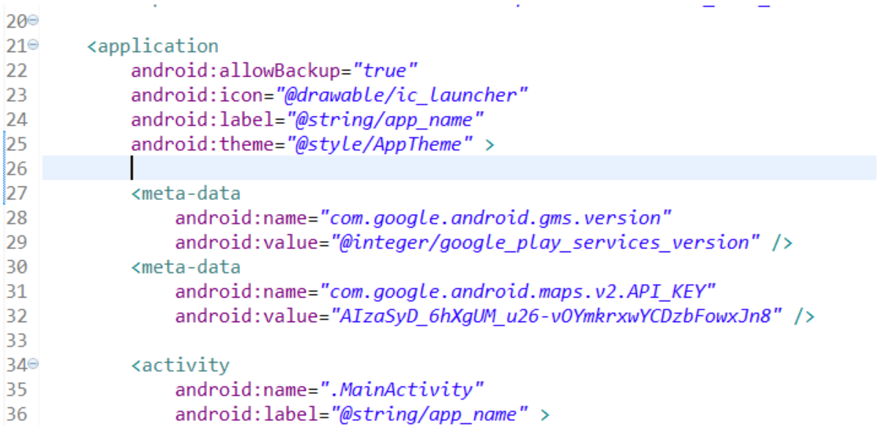
```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.place2);


    map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();

        if (map!=null){
          Marker hamburg = map.addMarker(new MarkerOptions().position(HAMBURG)
              .title("Hamburg"));
          Marker kiel = map.addMarker(new MarkerOptions()
              .position(KIEL)
              .title("Kiel")
              .snippet("Kiel is cool")
              .icon(BitmapDescriptorFactory
                  .fromResource(R.drawable.ic_launcher)));
        }
```

Then I put a FragmentLayout in the XML file where the map will be display :



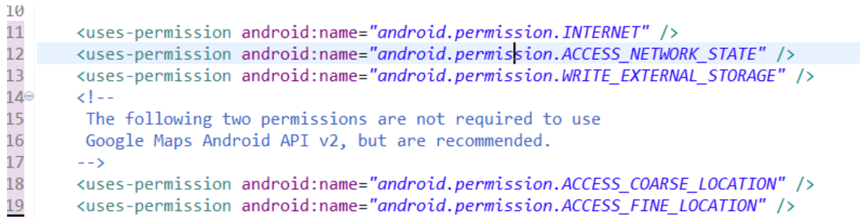
```xml

    <fragment
            android:id="@+id/map"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:name="com.google.android.gms.maps.MapFragment"/>
```

Modify the Manifest, with my API :



```xml

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyD_6hXgUM_u26-vOYmkrxwYCDzbFowxJn8" />

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
```

And give ask permissions of the user when he install it :



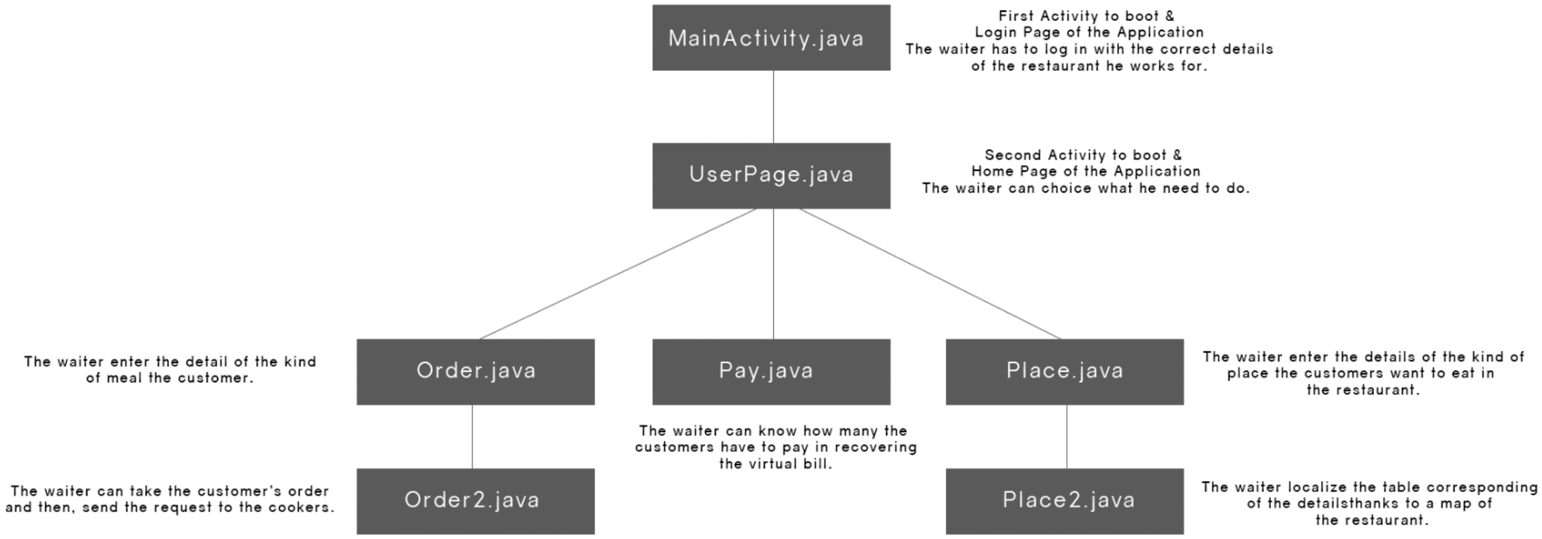
```
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!--
     The following two permissions are not required to use
     Google Maps Android API v2, but are recommended.
    -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Sources of the tutorial I used are available here :

https://developers.google.com/maps/documentation/android/start

# CONCLUSION

## 1) APP SPECIFICATION


MainActivity.java
First Activity to boot &
Login Page of the Application
The waiter has to log in with the correct details
of the restaurant he works for.
UserPage.java
Second Activity to boot &
Home Page of the Application
The waiter can choice what he need to do.
The waiter enter the detail of the kind
of meal the customer.
Order.java
Pay.java
Place.java
The waiter enter the details of the kind of
place the customers want to eat in
the restaurant.
The waiter can know how many the
customers have to pay in recovering
the virtual bill.
The waiter can take the customer's order
and then, send the request to the cookers.
Order2.java
Place2.java
The waiter localize the table corresponding
of the detailsthanks to a map of
the restaurant.

## 2) EVALUATION

### a. HOW HAS IT WORKED OUT

This application was really cool to develop, I had just one big difficulty to display the Google Map.

A little problem for the login System because like I said before, the String sending back by the database didn't check with the true string. I suppose that the solution will be to put the string into a Variable or another data which can't be modified when sending.

For the rest, I didn't find any problems.

### b. WHAT COULD BE DONE DIFFERENTLY

For this application, the Google Map was not adapted because it's not related with the restaurant. It will be better to put a dynamic map of the restaurant with all the table (and to see if it's available or not).

### c. FUTURE OUTLOOK

This application could integrate a payment system allowing the customers to pay directly via the smartphone of the waiter with a Credit Card.