

朝陽美食 APP

林坤緯	劉彥麟	許致嘉	吳智華
朝陽科技大學資 訊工程系副教授	朝陽科技大學資訊工 程系大學生	朝陽科技大學資訊工 程系大學生	朝陽科技大學資訊工 程系大學生
kwlin@cyut.edu.tw	s9927026@cyut.edu.tw	s9927060@cyut.edu.tw	s9927062@cyut.edu.tw

摘要

每到吃飯的時刻或者討論聚餐時，總會聽到“要到那兒吃?”這類的問句。顯然吃的問題困擾人們已久。

近年來智慧型手機普及，利用此一優勢製作有關美食 APP 將可解決此項問題。本開發設計之美食 APP 以朝陽科技大學附近商店為例，收尋學校附近餐廳。本開發 APP 提供美食搜尋方式有，一、依美食種類，二、依美食價格，三、依目前位置，四、依關鍵字搜尋，當使用者選擇要消費的店家時，可以從店家資訊獲得店家的地址、消費均價……等訊息。當使用者不知道自己身在何處時，可以藉由此功能來了解自己所在位置。此外，我們還開發搜尋餐廳附近的停車場，進而讓使用者方便停車。

關鍵字:手機 APP、美食種類、美食價格、目前位置

Keywords: Position, Mobile Phone APP, Prices, Delicacies

Abstract

”Where are we going to eat?” When we discussed where to eat, such kinds of question have been the most idioms. Obviously, eating problem has been tormenting people for long time.

In recent years, the smart phone has been more popular. To design a delicacies APP can solve the mentioned troublesome. The developed APP was in Chaoyang University of Technology (CYUT) as a case. The search methods of delicacies include 1. type of delicacies, 2. prices, 3. current position, 4. keyword searching. When the restaurant has been chosen, address, prices and location...etc., will show on the user smart phone screen. The studied APP has a positioning function, so the user can find the location of chosen restaurant and how to go to the chosen restaurant. Besides, the above functions, the APP also combine the concept of parking lot, which the users can find the location of parking lot and the parking information.

1. Introduction

由於 CPU 的進步、硬體體積縮小化以及 3G 網路的普及化，智慧型手機逐漸在手機市場中嶄露頭角[1]，在路上，不管是學生、上班族、家庭主婦、甚至是老年人，都會低頭看著手上的那塊大螢幕滑來滑去，螢幕的內容大多是社群網站、通訊軟體、電子郵件、遊戲、影片……等。隨著智慧型手機的功能日益強大，原本只是拿來通話、傳簡訊的「手機」，因為科技的日新月異，已經成為眾人不可或缺的隨身品。現在，只要一機在手，幾乎可以完成電腦所可以完成的工作，甚至可以獲得更好的操作體驗。外出吃飯時經常因停車的問題而不知道該怎麼解決，可能找停車位就花了不少時間而影響了吃飯的情緒，我們希望能夠改善這樣的問題，能夠讓使用者，順利的知道停車場的資訊。

我們製作了手機 APP，提供台中市朝陽科技大學附近的餐點店家資訊，並結合了停車場的設置以及資訊，方便使用者從查詢到進入餐廳點餐。此手機 APP 以 android 平台開發，另外停車場的部分結合了資料庫以及 Arduino 達到實作之目的。

2. 架構分析

圖 1 為本研究之 APP 與資料庫架構，在使用者 APP 方面包含了美食關鍵字搜尋、停車場關鍵字搜尋以及地圖資訊以及定位、店家位置所在。另外，如圖 1 左下方圖所示，我們使用了 SERVER 伺服器與 APP 作資料溝通，SERVER 伺服器主要是由 PHP 以及資料庫作結合，透過 PHP 與資料庫的連結，當 PHP 收到特定的值時會因為收到的值不同而對資料庫有不同的動作，舉個例子:我們將動作的不同以數值的方式代表，假設 APP 需要知道目前停車場的空位數，那它將會透過網路發送相對應的值給 PHP，假設當 PHP 收到值為 0，它將會執行抓取資料庫的程式碼，當資料庫收到抓取程式碼時，將會執行程式碼並且加抓取後的值回傳給 PHP，而 APP 將會抓取 PHP 的值回傳至使用者手機，因此使用者能隨時去更新停車場空位數。停車場架構如圖二所示。

第二是 APP 的部分，在使用者開啟 APP 後會進入預設首頁，預設首頁為第一頁面所顯示的是店家的清單欄，在店家清單以及第二頁面停車場清單最上方都有關鍵字的收尋可使

用，在點選店家以及停車場清單後，會進入預設首頁，預設首頁為第一頁面所顯示的是店家的清單欄，在店家清單以及第二頁面停車場清單最上方都有關鍵字的收尋可使用，在點選店家以及停車場清單後，會進入更詳細的資訊頁面，與資料庫作資料溝通也將是在詳細資訊頁面執行，另外第三頁面也就是地圖的頁面可供使用者以地圖的形式觀看目前的所在位置，使用者位置會以紅點顯示在正中央，並且也會將店家以美食的圖案標示在地圖上。

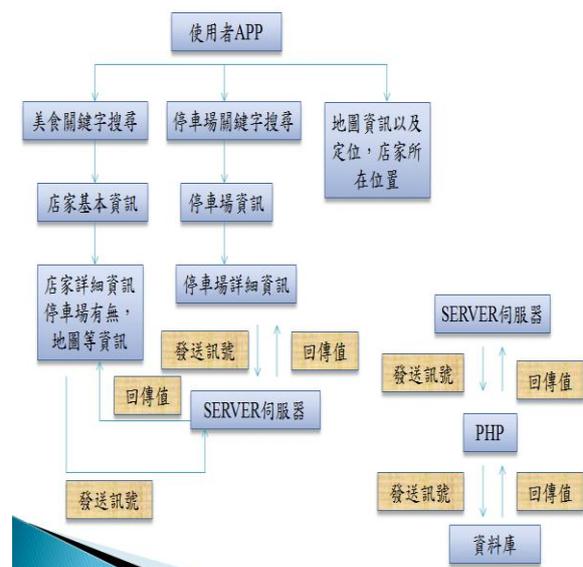


圖 1. App 與資料庫系統圖

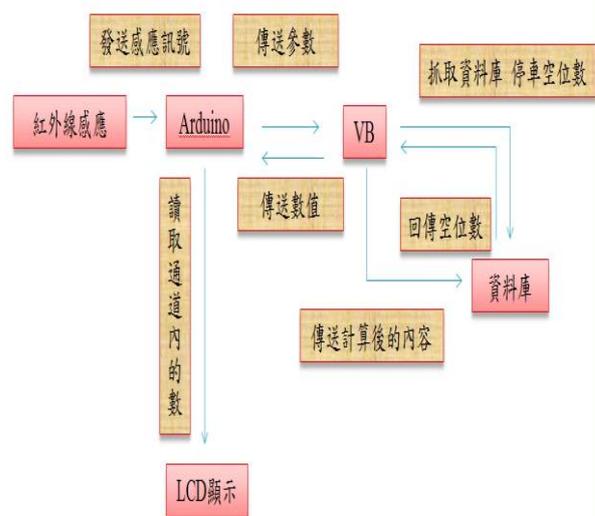


圖 2. 停車場架系統圖

停車場的部分，主要是利用 VB 來更新資料庫裡面的數值，VB 會先讀取資料庫的數值，將數值傳送給 Arduino 顯示在 LCD 上，再透過紅外線感應，當紅外線感應到有車子經過會傳送訊號給 Arduino，當 Arduino 收到訊號時將會傳送一個訊號通知 VB 作運算，不同的值有可能加 1 或減 1，VB 會將運算後的數值在更新資料庫，完成了一套更新的方式。

3. 技術分析

此 APP 開發為多項技術整合之實例，我們將為新穎的技術進行簡單分析。

3.1 顯示清單

首先在顯示店家清單的部分如圖 3 所示，主要是利用 ListView 來作顯示。我們先將資料存儲在系統中，再經由系統的需求設定把資料顯示在我們規劃的面板位置，如圖 4 店名放置在紅圈處，以及參考均價放置在藍圈處。圖 5 為片段程式碼，我們宣告了用來儲存的陣 mylist1，再從用來存儲店家資料的類別中，抓取所需要的值，再放入 mylist1 陣列中，並利用 SimpleAdapter mSchedule 來處理 Listview 形式，並且根據放入資料的位置由左至右對應如:05 變數資料將放入 R.id.text05，其中 ListView 是可以依設計者想像作不同的設計，先新增 XML 檔，在設計完後再引入至 ListView 清單即可，圖 5 紅圈處為自行設計清單檔名。

3.2 清單排列顯示

清單排列顯示的順序是按照使用者目前的位置與店家位置之間的距離計算其結果由近到遠作排列的，如圖 6 為片段程式碼，其中 Location getLocation 用來定位使用者目前位置，主要是先由 GPS 來做位置定位，假如 GPS 未開啟其位置為空，因此將會轉由網路來做位置的抓取並透過 public double Distance 此函式來計算，放入的 4 個的數值分別為店家經度與緯度，以及使用者位置的經度及緯度，再利用距離演算法計算出彼此之間的距離，透過 DistanceText 函式若是距離大於 1000 公尺則使用公里表示，最後再利用 DistanceSort 將資料清單按照距離大小由近到遠作重新排列，如圖 7 紅圈處為公里數排列情況。



圖 3. 店家清單



圖 4. 資訊放置處

```

}); //

final ArrayList<HashMap<String, String>> mylist1 = new ArrayList<HashMap<String, String>>();

for(int i=0;i<Pois.size();i++)
{
    HashMap<String, String> map = new HashMap<String, String>();
    map.put("05", "地址:"+Pois.get(i).getAddress());
    map.put("01",Pois.get(i).getName());
    map.put("02",Pois.get(i).getInformation());
    map.put("03", DistanceText(Pois.get(i).getDistance()));
    map.put("04", "S"+Pois.get(i).getPrice());
    map.put("06", Pois.get(i).getKind());
    map.put("07", Pois.get(i).getValue());

    mylist1.add(map);
}

mSchedule = new SimpleAdapter(getActivity(),
    mylist1,
    R.layout.Fragment1_list_0, new String[] {"05","01", "02","03","04","06","07"},
    new int[] {R.id.text05,R.id.text01,R.id.text02,R.id.text03,R.id.text04,R.id.text011,R.id.text012,

    public long getItemId(int position) {
        return mylist1.indexOf(getItem(position));
    }
};

listView.setAdapter(mSchedule);
listView.setTextFilterEnabled(true);

```

圖 5.ListView 片段程式碼

```

public double Distance(double longitude1, double latitude1, double longitude2, double latitude2)
{
    double radLatitude1 = latitude1 * Math.PI / 180;
    double radLatitude2 = latitude2 * Math.PI / 180;
    double l = radLatitude1 - radLatitude2;
    double p = longitude1 * Math.PI / 180 - longitude2 * Math.PI / 180;
    double distance = 2 * Math.asin(Math.sqrt(Math.pow(Math.sin(l / 2), 2)
        + Math.cos(radLatitude1) * Math.cos(radLatitude2)
        * Math.pow(Math.sin(p / 2), 2)));
    distance = distance * 6378137.0;
    distance = Math.round(distance * 10000) / 10000;

    return distance ;
}

private String DistanceText(double distance)
{
    if(distance < 1000 ) return String.valueOf((int)distance) + "m" ;
    else return new DecimalFormat("#.00").format(distance/1000) + "km" ;
}

public Location getLocation(Context context) {
    // TODO Auto-generated method stub
    LocationManager locMan = (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
    Location location = locMan.getLastKnownLocation(LocationManager.GPS_PROVIDER);

    if (location == null)
    {
        location = locMan.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
    }
    return location;
}

private void DistanceSort(ArrayList<Poi> poi)
{
    Collections.sort(poi, new Comparator<Poi>()
    {
        @Override
        public int compare(Poi poi1, Poi poi2)
        {
            return poi1.getDistance() < poi2.getDistance() ? -1 : 1 ;
        }
    });
}

```

圖 6.定位與距離計算片段程式碼

3.3 停車場資訊

停車場資訊主要是顯示本研究中模擬出的停車場，在研究中我們建立了資料庫並且將資料庫設置在 SERVER 遠端主要透過網路來做溝通，如圖 8 為片段程式碼在研究中我們先

將動作數值化，根據不同的動作有不同的數值，再透過此程式碼發送動作數值至伺服器。此模擬方法為抓取資料庫數值，因此伺服器抓取數值再作回傳，而程式碼會將此數值顯示在



圖 7. 公里數排列



圖 9.顯示更新數值

APP 頁面上,圖 9 紅框處為 APP 與資料庫作溝通時能立即更新的數值。圖 10 為資料庫顯示圖。

```

public void dostep1()
{
    String uriAPI = stringuriAPI ;
    /*建立HTTP Post連線*/
    HttpPost httpRequest = new HttpPost(uriAPI);
    /*
    * Post運作傳送變數必須用NameValuePair[]陣列儲存
    */
    List <NameValuePair> params = new ArrayList <NameValuePair>();
    params.add(new BasicNameValuePair("id", stringid));
    try
    {
        /*發出HTTP request*/
        httpRequest.setEntity(new UrlEncodedFormEntity(params, HTTP.UTF_8));
        /*取得HTTP response*/
        HttpResponse httpResponse = new DefaultHttpClient().execute(httpRequest);
        /*若狀態碼為200 ok*/
        if (httpResponse.getStatusLine().getStatusCode() == 200)
        {
            /*取出回應字串*/
            String strResult = EntityUtils.toString(httpResponse.getEntity(),"utf_8");

            Message msg = new Message();
            Bundle data=new Bundle();
            data.putString("value", strResult);
            msg.setData(data);
            msg.what =0;
            mHandler.sendMessage(msg);
        }
        else
        {
            String strResult =String.valueOf("Error Response: "+httpResponse.getStatusLine().toString());
            Message msg = new Message();
            Bundle data=new Bundle();
            data.putString("value", strResult);
            msg.setData(data);
            msg.what =0;
            mHandler.sendMessage(msg);
        }
    }
    catch (ClientProtocolException e)
    {
    }
}

```

圖 8.與 SERVER 程式碼片段

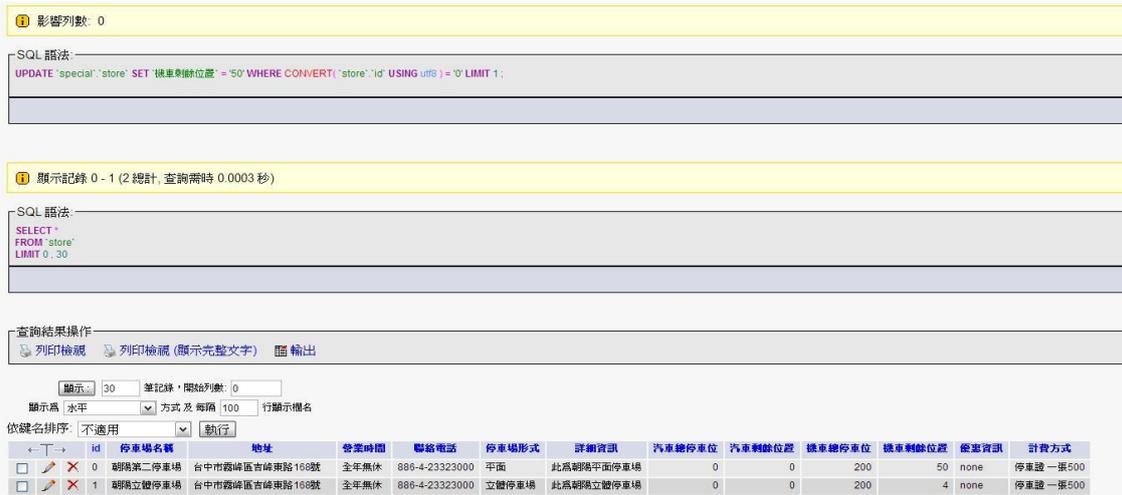


圖 10.資料庫

3.4 路徑規劃

使用者點選路徑規劃時，系統將會抓取使用者目前的位置坐標，以及讀取系統中儲存的店家位置坐標，在將兩個參數值透過 intent 的方式傳送至 Google 提供的路徑功能，利用此方法可讓使用者選擇多種路徑，像是開車、走路等，圖 11 為片段程式碼，圖 12 為結果圖。

3.4 詳細菜單

詳細菜單為本研究提供讓使用者能夠更了解店家的菜色，圖 13 顯示片段程式碼，圖 14 顯示為菜單結果圖，此處我們利用的方法與停車場資訊中與 SERVER 資料庫作溝通的方法相同，唯一不同的地方在於 APP 接收圖片的方式，首先我們必須先輸入網址才能讓 APP

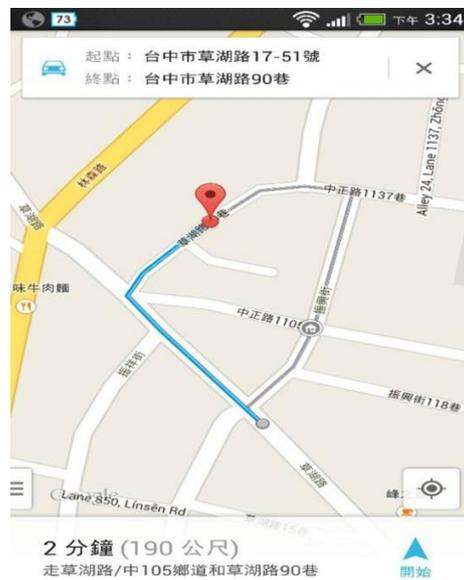


圖 12.路徑規劃結果

```

public void onClick(View v) {
    // TODO Auto-generated method stub

    Uri uri = Uri.parse("http://maps.google.com/maps?f=d&saddr="+ GeoPointToString1(fromGeoPoint) + "&daddr="+G);
    Intent it = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(it);
}

private String GeoPointToString1(GeoPoint gp)
{
    String strReturn="";
    try
    {
        /*當Location存在*/
        if (gp != null)
        {
            double geoLatitude = (int)gp.getLatitudeE6()/1E6;
            double geoLongitude = (int)gp.getLongitudeE6()/1E6;
            strReturn = String.valueOf(geoLatitude)+"."+
                String.valueOf(geoLongitude);
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    return strReturn;
}

```

圖 11.intent 片段程式碼

```

1
//建立一個AsyncTask執行緒進行圖片讀取動作，並帶入圖片連結網址路徑
new AsyncTask<String, Void, Bitmap>()
{
    @Override
    protected Bitmap doInBackground(String... params)
    {
        String url = params[0];
        return getBitmapFromURL(url);
    }
    |
    @Override
    protected void onPostExecute(Bitmap result)
    {
        img.setImageBitmap(result);
        super.onPostExecute(result);
    }
}.execute("圖片連結網址路徑");
}
};
}
//讀取網路圖片，型態為Bitmap
private static Bitmap getBitmapFromURL(String imageUrl)
{
    try
    {
        URL url = new URL(imageUrl);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setDoInput(true);
        connection.connect();
        InputStream input = connection.getInputStream();
        Bitmap bitmap = BitmapFactory.decodeStream(input);
        return bitmap;
    }
    catch (IOException e)
    {
        e.printStackTrace();
        return null;
    }
}
}
}

```

圖 13. 與 SERVER 溝通

能夠知道圖片存放在哪?我們將網址存入 String url 中，再透過 Bitmap getBitmapFromURL 函式來與資料庫作圖片的接收，其中圖片型態為 Bitmap。

3.6 Google Map

此方法是為了讓使用者能明確的知道目前自己正處於哪個位置上，並且用圖示的方式顯示出來，如圖 15 所示中央紅點為使用者目前位置，除了目前位置外，周圍也會有我們儲存的店家名稱並且以圖案顯示，讓使用者更能夠知道目前的位置附近有哪些的店家。如圖 16 為片段程式碼，在程式碼中，我們必須先使用 Google 所提供的地圖，因此必須先去 Google 官網申請 API key，在申請完後必須將 API key 放置 AndroidManifest.xml 檔中，如圖 17 紅框所示 android:name 為名稱，android:value 則是申請的 API key，在寫入 API key 之後我們必須創建地圖，因此先宣告了 MapView mapView，MapView 為形式，mMapView 為名稱，再透過 mMap = mapView.getMap()即可創建出地圖，在 mMap.addMarker 中則是放置手用者目前定位坐標 NKUT，而在後面的 title、snippet 都是點選圖 15 中央紅點後會出現的文字，NKUT 為使用者定位後回傳的座標值，我們再利用存儲的店家利用我們設計的 for 迴圈，將所有的店家建在 Google map 上。



圖 14.詳細菜單



圖 15.使用者位置

```

view = inflater.inflate(R.layout.fragment5, container, false);
mMapView = (MapView) view.findViewById(R.id.map);
mMapView.onCreate(savedInstanceState);
mMapView.onResume();

try {
    MapsInitializer.initialize(getActivity());
} catch (GooglePlayServicesNotAvailableException e) {
    e.printStackTrace();
}

mMap = mMapView.getMap();

mMap.addMarker(new MarkerOptions().position(NKUT).title("hello").snippet("您的位置"));

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(NKUT, 16));

for(int i=0;i<Pois.size();i++)
{
    double dLat1 = Pois.get(i).getLatitude();
    double dLng1 = Pois.get(i).getLongitude();

    LatLng NKUT1 = new LatLng(dLat1, dLng1);
    String string=Pois.get(i).getName();
    String string1=Pois.get(i).getInformation();
    // double stringLat=bundle.getDouble("returnLat");
    // double stringLong=bundle.getDouble("returnLong");

    mMap.addMarker(new MarkerOptions().position(NKUT1)
        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE))
        .title(string)
        .snippet(string1)
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.unnamed)));;
}

```

圖 16.地圖片段程式碼

```

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.example.specialx4.MainActivity"
        android:label="@string/app_name"
        android:theme="@style/TitleBarTheme"
        >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <activity android:name="Fragment1"></activity>
        <activity android:name="Fragment2"></activity>
        <activity android:name="Fragment3"></activity>
        <activity android:name="Fragment4"></activity>
        <activity android:name="Otheractivity"
            android:theme="@style/TitleTheme"
        >
        </activity>
        <activity android:name="TitleFragment" ></activity>
        <activity android:name="Fragment2_parking"
            android:theme="@style/TitleTheme"
        ></activity>
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyBt3C7VxCdc9nDx8N46GGE952Wk7fDxWig" />
    </application>
</manifest>

```

圖 17.API KEY 放置處

4. Conclusions

本開發已經完成一個實用之美食 APP，將為朝陽科大的師生以及到大理、霧峰區的人們提供方便之美食收尋。本開發使用 Arduino、LCD、紅外線之橋、Visual Basic2010、APP Server、Android 等，手機 APP 裡面的功能是為了讓使用者感到方便、好用而精心設計的，此開發提供了以下其他 APP 所沒有提供的功能：

(1) 路徑規劃的問題:能夠讓使用者順利的到達目的地

- (2) 詳細菜單:讓使用者明確的知道店家的菜色。
- (3) 停車場的資訊:使用者可以知道店家是否有無附設停車場。
- (4) 即時的空位數:使用者可以立即的知道目前停車場的空位數。

未來我們希望能夠加入評論的功能，讓使用者更能知道店家所帶來的服務是否舒適，將以星星的數量來代表服務品質，另外我們也將製作一套軟體，能夠讓店家去更改自己的優惠

資訊，或者是營業時間等，能夠將店家更改的資訊立刻的更新到 APP，讓使用者也能與店家的更新來達到同步。

參考文獻

- [1] Chihsiu Wang, DA-DA Chen, Kuofeng Tsai, 以策略管理與六力分析之方式去研究智慧型手機未來發展趨勢。
- [2] Daniel Gallegoa, Wolfgang Woernd, Gabriel Huecas, “Evaluating the impact of proactivity in the user experience of a context-aware restaurant recommender for Android smartphones” *Journal of Systems Architecture*, Vol.59 ,pp.748–758, 2013
- [3] Andrew Hoog,” *Android software development kit and android debug bridge*”, pp. 65 – 103, 2011.
- [4] Esther Irawati Setiawanb, Gunawanb, Indra Maryatia, Joan Santosoa, and Rossy Prabowo Chandrab, “Shortest Path Problem for Public Transportation Using GPS and Map Service”, *Procedia - Social and Behavioral Sciences*”, Vol. 57, pp. 426 – 431, 2012
- [5] Christopher Dong, Xing Liu, “ Development of Android Application for Language Studies”, Vol. 4, pp. 8–16, 2013.