A FOOD SEARCHING APPLICATION



Testing Document   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BIT302 Software Engineering:  
Assignment 2  
  
**Author: B1402427 CHIN WEI QING  
Lecturer: MS. SEETHA**13 April 2016

**Contact Detail:** [**WEIQING.CHIN@GMAIL.COM**](mailto:WEIQING.CHIN@GMAIL.COM)

## **Abstract**

## The process of developing is recorded and to be shown in this document for future referencing.

Contents

[**Abstract** 2](#_Toc448291913)

[**Test objectives** 4](#_Toc448291914)

[**Test Strategies** 5](#_Toc448291915)

[**Test Case** 6](#_Toc448291916)

[Integration Testing 56](#_Toc448291917)

[System testing 61](#_Toc448291918)

[**Test Summary** 74](#_Toc448291919)

**Test objectives**  
Tobj-1.0  
To identify all potential cause of problem by having Mofo Application going through stress test

Tobj-2.0  
To solve potential problem cause by external factor

Tobj-3.0  
To solve errors found by alpha tester.

Tobj-4.0  
To verify functions requirement and design are align with the specification and design description

Tobj-5.0   
To ensure functional requirement and design are met during Mofo Application development

Tobj-6.0  
To ensure all component work according to the function description.

## **Test Strategies**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Technique | Date | Schedule | Aim |
| Unit Testing | 20/2/16 – 24/3/16 | After each section of code are completed individually | * To ease error detection as code are all in small parts * To ease code change * To ensure all individual parts of the code are correct before merging. |
| Integration Testing | 31/2/16 -  29/3/16 | After component with same function is merged and completed | * To make sure different module can work together as one without problem * To ensure the transition between page work correctly |
| System Testing | 29/4/16 -  10/4/16 | When the whole application is completed | * To check if Mofo application met the functional and non-functional requirement |

**Test Case**Unit Testing

|  |
| --- |
| **Unit Test Plan** |
| **TCase-1.0 Unit Testing #1** |
| Valid email at login page |
| **TCase-1.1 Description** |
| To validate email, display test if it’s not a valid email when clicked. |
| **TCase-1.2 Source Code** |
| **login\_button**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {   **login\_button**.setEnabled(**false**);  **signup\_text**.setEnabled(**false**);   **final** String email = **Email**.getText().toString();  **if** (!validEmail(email)) {  **Email**.setError(**"Invalid Email"**);  **Email**.requestFocus();  }    **if** (validEmailLength(email)) {  **Email**.setError(**"email max length is 70 digit"**);  **Email**.requestFocus();  }  }  });  **private boolean** validEmail(String validEmail) {  String emailPattern = **"^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"** + **"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$"**;   Pattern pattern = Pattern.*compile*(emailPattern);  Matcher matcher = pattern.matcher(validEmail);  **return** matcher.matches(); }  *// validating email length* **private boolean** validEmailLength(String email){  **if** (email != **null** && email.length() <71) {  **return true**;  }  **return false**; } |
| **TCase-1.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **email** | | 1 | applepie | | 2 | applepie01applepie02applepie03applepi04applepie05applepie0611@gmail.com | | 3 | applepie@gmail.com | |
| **TCase-1.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Invalid email” text displayed | | 2 | “email max length is 70 digit” text displayed | | 3 | - | |
| **TCase-1.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Invalid email” text displayed | | 2 | “email max length is 70 digit” text displayed | | 3 | - | |
| **TCase-1.6 Remark** |
| The Login page validation for email work as expected. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-2.0 Unit Testing #2** |
| Validate password at login page |
| **TCase-2.1 Description** |
| To validate password, display text if it’s not a valid password when clicked. |
| **TCase-2.2 Source Code** |
| **login\_button**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {   **final** String pass = **Pass**.getText().toString();  **if** (!validPassword(pass)) {  **Pass**.setError(**"password length require 6-8 digit"**);  **Pass**.requestFocus(); }  }  });  **private boolean** validPassword(String pass) {  **if** (pass != **null** && pass.length() >5 && pass.length() <9) {  **return true**;  }  **return false**; } |
| **TCase-2.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **password** | | 1 | apple | | 2 | 123456789 | | 3 | password | |
| **TCase-2.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “password length require 6-8digit”text displayed | | 2 | “password length require 6-8digit”text displayed | | 3 | - | |
| **TCase-2.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “password length require 6-8digit”text displayed | | 2 | “password length require 6-8digit”text displayed | | 3 | - | |
| **TCase-2.6 Remark** |
| The Login page validation for password work as expected. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-3.0 Unit Testing #3** |
| Validate filled field at login page |
| **TCase-3.1 Description** |
| To validate password, display text if it’s not a valid password when clicked. |
| **TCase-3.2 Source Code** |
| **login\_button**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {  **if** (**Email**.getText().toString().equals(**""**) || **Pass**.getText().toString().equals(**""**)) {  **builder** = **new** AlertDialog.Builder(LoginActivity.**this**);  **builder**.setTitle(**"Error"**);  **builder**.setMessage(**"Please fill up all fields"**);  **builder**.setPositiveButton(**"OK"**, **new** DialogInterface.OnClickListener() {   @Override  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.dismiss();  }  });  AlertDialog alertDialog = **builder**.create();  alertDialog.show(); }  }  }); |
| **TCase-3.3 Test Data** |
| |  |  |  | | --- | --- | --- | | **Test number** | **email** | **password** | | 1 | - | apple | | 2 | apple | - | | 3 | email | password | |
| **TCase-3.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Please fill up all fields” alert dialog display. | | 2 | “Please fill up all fields” alert dialog display. | | 3 | - | |
| **TCase-3.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Please fill up all fields” alert dialog display. | | 2 | “Please fill up all fields” alert dialog display. | | 3 | - | |
| **TCase-3.6 Remark** |
| The Login page validation for empty field work as expected. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-4.0 Unit Testing #4** |
| Login using valid email and password |
| **TCase-4.1 Description** |
| To validate email and password on clicked. Toast displayed based on login status. |
| **TCase-4.2 Source Code** |
| **private static final String login\_url = "http://mofo.16mb.com/login.php";**  @Override  **public void** onClick(View v) {   **request** = **new** StringRequest(Request.Method.***POST***, ***login\_url***, **new** Response.Listener<String>() {   *//info returned* @Override  **public void** onResponse(String response) {  **try** {   JSONObject jsonObject = **new** JSONObject(response);  JSONArray srArray = jsonObject.getJSONArray(**"server\_response"**);  JSONObject srObject = srArray.getJSONObject(0);   String code = srObject.getString(**"code"**);  String message = srObject.getString(**"message"**);   *//check if login is true* **if** (code.equals(**"login\_true"**)) {   Toast.*makeText*(LoginActivity.**this**, **"Welcome back "**, Toast.***LENGTH\_LONG***).show();   }  **else if** (code.equals(**"login\_false"**))  {  Toast.*makeText*(LoginActivity.**this**, message,Toast.***LENGTH\_LONG***).show();    } **else** {  Toast.*makeText*(LoginActivity.**this**, **"Something is wrong"**, Toast.***LENGTH\_LONG***).show();  }   }   }  }){  *// details push to login authentication* @Override  **protected** Map<String,String> getParams() **throws** AuthFailureError {  HashMap<String,String> hashMap = **new** HashMap<String, String>();  hashMap.put(**"email"**,**Email**.getText().toString());  hashMap.put(**"password"**,**Pass**.getText().toString());  **return** hashMap;  }  };  **requestQueue**.add(**request**);   } }); |
| **TCase-4.3 Test Data** |
| |  |  |  | | --- | --- | --- | | **Test number** | **email** | **password** | | 1 | admin@mofo.com | mofoadmin | | 2 | fake@email.com | fakeemail | | 3 | Yoyo22@email.com | Yoyo21 | |
| **TCase-4.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Welcome back” Toast display. | | 2 | “Login Fail, email don’t exist” Toast display. | | 3 | “Login Fail, Email & password mismatch” Toast display. | |
| **TCase-4.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | “Welcome back” Toast display. | | 2 | “Login Fail, email don’t exist” Toast display. | | 3 | “Login Fail, Email & password mismatch” Toast display. | |
| **TCase-4.6 Remark** |
| The valid user logged in successfully. Login fail error toast displayed correctly. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-5.0 Unit Testing #5** |
| Directing to Setting page on login |
| **TCase-5.1 Description** |
| To direct the Login page to Setting page upon login success, else remain at Login page |
| **TCase-5.2 Source Code** |
| **LoginActivity**  **if** (code.equals(**"login\_true"**)) {  User.*setLogin*(**true**);  System.***out***.println(User.*getLogin*());  startActivity(**new** Intent(getApplicationContext(), MainActivity.**class**));  } **else** {  }  **MainActivity**  @Override **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);    *//Set the fragment initially* SettingFragment fragment = **new** SettingFragment();  android.support.v4.app.FragmentTransaction fragmentTransaction =  getSupportFragmentManager().beginTransaction();  fragmentTransaction.replace(R.id.***fragment\_container***, fragment);  fragmentTransaction.commit(); |
| **TCase-5.3 Test Data** |
| |  |  |  | | --- | --- | --- | | **Test number** | **email** | **password** | | 1 | admin@mofo.com | mofoadmin | | 2 | fake@email.com | fakeemail | | 3 | id@email.com | password | |
| **TCase-5.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | Login page directed to Setting page | | 2 | Remain at Login page | | 3 | Remain at Login page | |
| **TCase-5.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | Login page directed to Setting page | | 2 | Remain at Login page | | 3 | Remain at Login page | |
| **TCase-5.6 Remark** |
| The page is directed to setting page upon login successful. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-6.0 Unit Testing #6** |
| Valid email, password and username in Registration page. |
| **TCase-6.1 Description** |
| To validate email, password and user name enter in Registration page. Display text if the length is invalid, alert box if any field is empty. |
| **TCase-6.2 Source Code** |
| *//check email* **if** (!validEmail(email)) {  **Email**.setError(**"Invalid Email"**);  **Email**.requestFocus();   }  *//check email length* **if** (!validEmailLength(email)) {  **Email**.setError(**"email max length is 70 digit"**);  **Email**.requestFocus();  }  *//check name length*  **if** (!validName(name)) {  **Name**.setError(**"name length require 6-15 digit "**);  **Name**.requestFocus();  }   *//check password* **if** (!validPassword(pass)) {  **Pass**.setError(**"password length require 6-8 digit"**);  **Pass**.requestFocus();  }   *//check 2 password identical* **if**(!conPass.equals(pass)) {  **Pass**.setError(**"password mismatch"**);  **Pass**.requestFocus();  **ConPass**.setError(**"password mismatch"**);  **ConPass**.requestFocus();  }   *//check if there is any empty field* **if** (email.equals(**""**) ||pass.equals(**""**)|| conPass.equals(**""**)|| name.equals(**""**)) {  **builder** = **new** AlertDialog.Builder(RegisterActivity.**this**);  **builder**.setTitle(**"Error"**);  **builder**.setMessage(**"Please fill up all fields"**);  **builder**.setPositiveButton(**"OK"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.dismiss();  Log.*i*(**"infohelp: "**, **"13"**);  }   });  AlertDialog alertDialog = **builder**.create();  alertDialog.show();  }  **private boolean** validEmail(String email) {  String EMAIL\_PATTERN = **"^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"** + **"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$"**;   Pattern pattern = Pattern.*compile*(EMAIL\_PATTERN);  Matcher matcher = pattern.matcher(email);   **return** matcher.matches(); }  *// validate email length* **private boolean** validEmailLength(String email){  **if** (email != **null** && email.length() <71) {  **return true**;  }  **return false**; }  *//validate name length* **private boolean** validName(String name){  **if** (name != **null** && name.length() >5 && name.length() <16) {  **return true**;  }  **return false**; }  *// validating password with retype password* **private boolean** validPassword(String pass) {  **if** (pass != **null** && pass.length() > 5 && pass.length() <9) {  **return true**;  }  **return false**; } |
| **TCase-6.3 Test Data** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **name** | **email** | **password** | **conpassword** | | 1 | user | user@mofo.com | mofouser | mofouser | | 2 | thisIsEqualto016 | id@email.com | password | password | | 3 | user | user | password | password | | 4 | user | aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaddddddddddddaaaaaaaa@gggggggggg.com | password123 | password123 | | 5 | user | user@mofo.com | password | password123 | | 6 | thisIsEqualto016 | user | - | password | | 7 | - | - | password123 | password | |
| **TCase-6.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | - | | 2 | Alert dialog displayed.  “name max length is 15 digit” Text display. | | 3 | Alert dialog displayed.  “invalid email” Text display. | | 4 | Alert dialog displayed.  “email max length is 70 digit” Text display. “password length require 6-8 digit” Text display. | | 5 | Alert dialog displayed.  “password mismatch” Text display. | | 6 | Alert dialog displayed.  “Please fill up all field” alert dialog display.  “password length require 6-8 digit” Text display. “password mismatch” Text display. | | 7 | Alert dialog displayed.  “name max length is 15 digit” Text display. “invalid email” Text display. “password length require 6-8 digit” Text display. “password mismatch” Text display. | |
| **TCase-6.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | - | | 2 | Alert dialog displayed.  “name max length is 15 digit” Text display. | | 3 | Alert dialog displayed.  “invalid email” Text display. | | 4 | Alert dialog displayed.  “email max length is 70 digit” Text display. “password length require 6-8 digit” Text display. | | 5 | Alert dialog displayed.  “password mismatch” Text display. | | 6 | Alert dialog displayed.  “Please fill up all field” alert dialog display.  “password length require 6-8 digit” Text display. “password mismatch” Text display. | | 7 | Alert dialog displayed.  “name max length is 15 digit” Text display. “invalid email” Text display. “password length require 6-8 digit” Text display. “password mismatch” Text display. | |
| **TCase-6.6 Remark** |
| The validation of register input field is working as expected. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-7.0 Unit Testing #7** |
| Registration validation |
| **TCase-7.1 Description** |
| To register user base on input field and check the status of registration. |
| **TCase-7.2 Source Code** |
| **reg\_Button**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {  **request** = **new** StringRequest(Request.Method.***POST***, ***register\_url***, **new** Response.Listener<String>() {  @Override  **public void** onResponse(String response) {  **try** {  JSONObject jsonObject = **new** JSONObject(response);  JSONArray jsonArray = jsonObject.getJSONArray(**"server\_response"**);  JSONObject JO = jsonArray.getJSONObject(0);   String code = JO.getString(**"code"**);  System.***out***.println(code);  String message = JO.getString(**"message"**);  **if** (code.equals(**"reg\_true"**)) {  Toast.*makeText*(RegisterActivity.**this**, **"Registration Success"**, Toast.***LENGTH\_LONG***).show();    } **else if** (code.equals(**"reg\_false"**)) {  Toast.*makeText*(RegisterActivity.**this**, message, Toast.***LENGTH\_LONG***).show();    } **else** {  Toast.*makeText*(RegisterActivity.**this**, **"Something is wrong"**, Toast.***LENGTH\_LONG***).show();  }  }  }   })  {  @Override  **protected** Map<String,String> getParams() **throws** AuthFailureError {  HashMap<String,String> hashMap = **new** HashMap<String,String>();  hashMap.put(**"username"**,name);  hashMap.put(**"email"**,email);  hashMap.put(**"password"**,pass);  **return** hashMap;  }  };  **requestQueue**.add(**request**);  }   } }); |
| **TCase-7.3 Test Data** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **name** | **email** | **password** | **conpassword** | | 1 | admin | user123@email.com | 123456 | 123456 | | 2 | newguy | admin@mofo.com | 123456 | 123456 | | 3 | newguy | newguy@email.com | 123456 | 123456 | |
| **TCase-7.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | "Username Already Exist. Please try another username." Toast display. | | 2 | “Email Already Exist. Please try another email.” Toast display. | | 3 | “Registration Success” Toast display. | |
| **TCase-7.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | "Username Already Exist. Please try another username." Toast display. | | 2 | “Email Already Exist. Please try another email.” Toast display. | | 3 | “Registration Success” Toast display. | |
| **TCase-7.6 Remark** |
| The user is created and stored in the database “username” under the user name “newguy” and, email “newguy@email.com” and password “123456”. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-8.0 Unit Testing #8** |
| Validate Google Place API URL |
| **TCase-8.1 Description** |
| To validate Google Place API URL by entering the address containing, location, radius and type of food. If the URL is valid, “status: ok” will be print on console, else “status: request deny” will be printed instead. |
| **TCase-8.2 Source Code** |
| **HomeFragment.java**  APIDownloadTask task = **new** APIDownloadTask();  String register\_url = *“”;* task.execute(register\_url);  **APIDownloadTask.java**  **public class** APIDownloadTask **extends** AsyncTask<String, Void, String> {  **public static** List<String> *place\_idArray* = **new** ArrayList<String>();    @Override  **protected** String doInBackground(String... urls) {  URL url;  HttpURLConnection urlConnection;  **try** {  url = **new** URL(urls[0]);  urlConnection = (HttpURLConnection) url.openConnection();  **try** {  BufferedReader bufferedReader = **new** BufferedReader(**new** InputStreamReader(urlConnection.getInputStream()));  StringBuilder stringBuilder = **new** StringBuilder();  String line;  **while** ((line = bufferedReader.readLine()) != **null**) {  stringBuilder.append(line).append(**"\n"**);  }  bufferedReader.close();  **return** stringBuilder.toString();  }  **finally**{  urlConnection.disconnect();  }  }  **catch**(Exception e) {  **return null**;  }  }   @Override  **protected void** onPostExecute(String result) {  **super**.onPostExecute(result);  **try** {   JSONObject jsonObject = **new** JSONObject(result);   String status = jsonObject.getString(**"status"**);  System.***out***.println(**"STATUS : "** + status);    } **catch** (JSONException e) {  e.printStackTrace();  }   } |
| **TCase-8.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Input String register\_url** | | 1 | *https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=500&keyword=japanese&type=food&key=AIzaSyDQG21ZWu1Sojx6PtZjy\_boIJQobaERVHo* | | 2 | https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=500&minprice=1&maxprice=4&type=food&key=AIzaSyA7MsBwPh1xoX7I2pgaoZdNyKBx2Df7R7c | | 3 | https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=500&type=*japanese*food&key=AIzaSyA7MsBwPh1xoX7I2pgaoZdNyKBx2Df7R7c | |
| **TCase-8.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | "STATUS : OK" print on the console | | 2 | "STATUS : OK" print on the console | | 3 | "STATUS : REQUEST\_DENIED" print on the console | |
| **TCase-8.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | "STATUS : OK" print on the console | | 2 | "STATUS : ZERO\_RESULT " print on the console | | 3 | "STATUS : REQUEST\_DENIED" print on the console | |
| **TCase-8.6 Remark** |
| The status printed “OK” on the console to show that there’s data returned when the input URL is valid. The status printed “ZERO\_RESULT” on the console to show that there is no data returned when the input URL is valid. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-9.0 Unit Testing #9** |
| Retrieve the places API from Google |
| **TCase-9.1 Description** |
| To retrieve a list of place IDs from Google by entering the valid address, and save it into an array. |
| **TCase-9.2 Source Code** |
| **HomeFragment.java**  APIDownloadTask task = **new** APIDownloadTask();  String register\_url = *“”;* task.execute(register\_url);  **APIDownloadTask.java**  @Override **protected void** onPostExecute(String result) {  **super**.onPostExecute(result);   **try** {    JSONObject jsonObject = **new** JSONObject(result);  String results = jsonObject.getString(**"results"**);  JSONArray arr = **new** JSONArray(results);    *//check aar length* Log.*i*(**"arr length"**, String.*valueOf*(arr.length()));     **for** (**int** i = 0; i < arr.length(); i++) {   String place\_id = arr.getJSONObject(i).getString(**"place\_id"**);  *place\_idArray*.add(place\_id);    } |
| **TCase-9.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Input String register\_url** | | 1 | *https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=5000&keyword=japanese&type=food&key=AIzaSyDQG21ZWu1Sojx6PtZjy\_boIJQobaERVHo* | | 2 | *https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=0&keyword=japanese&type=food&key=AIzaSyDQG21ZWu1Sojx6PtZjy\_boIJQobaERVHo* | | 3 | *https://maps.googleapis.com/maps/api/place/radarsearch/json?location=3.152209,101.671317&radius=2000&keyword=japanese&type=food&key=AIzaSyDQG21ZWu1Sojx6PtZjy\_boIJQobaERVHo* | |
| **TCase-9.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | System display  “I/arr length: 200” | | 2 | System display  “I/arr length: 0” | | 3 | System display  I/arr length: >10 | |
| **TCase-9.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | System display  “I/arr length: 200” | | 2 | System display  “I/arr length: 0” | | 3 | System display  “I/arr length: 16” | |
| **TCase-9.6 Remark** |
| The system logged the total number of places returned on the console when the valid URL is entered. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-10.0 Unit Testing #10** |
| Retrieve place APIs details |
| **TCase-10.1 Description** |
| To retrieve the place API details of the place by inputting the place ID in the address. |
| **TCase-10.2 Source Code** |
| **HomeFragment.java**  retrieveData task2 = **new** retrieveData();  String link = (**"https://maps.googleapis.com/maps/api/place/details/json?placeid="** + place\_id + **"&key=AIzaSyDQG21ZWu1Sojx6PtZjy\_boIJQobaERVHo"**);  task2.execute(link);  **retrieveData.java**  **public class** retrieveData **extends** AsyncTask<String, Void, String> {  **protected** String doInBackground(String... urls) {   URL url;  HttpURLConnection urlConnection;  **try** {  url = **new** URL(urls[0]);  urlConnection = (HttpURLConnection) url.openConnection();  **try** {  BufferedReader bufferedReader = **new** BufferedReader(**new** InputStreamReader(urlConnection.getInputStream()));  StringBuilder stringBuilder = **new** StringBuilder();  String line;  **while** ((line = bufferedReader.readLine()) != **null**) {  stringBuilder.append(line).append(**"\n"**);  }  bufferedReader.close();  **return** stringBuilder.toString();  }  **finally**{  urlConnection.disconnect();  }  }  **catch**(Exception e) {  Log.*e*(**"ERROR"**, e.getMessage(), e);  **return null**;  }  }   @Override  **protected void** onPostExecute(String result) {  **super**.onPostExecute(result); **try** {  JSONObject jObj = **new** JSONObject(result);  JSONObject response = jObj.getJSONObject(**"result"**);  *place\_id* = response.getString(**"place\_id"**);  *title* = response.getString(**"name"**);    **if** (response.has(**"open\_now"**)){  *open\_now* = response.getString(**"open\_now"**);  }**else**{  *open\_now* =**"not available"**;  }   **if** (response.has(**"formatted\_address"**)){  *address* = response.getString(**"formatted\_address"**);  }**else**{  *address* =**"not available"**;  }   **if** (response.has(**"formatted\_phone\_number"**)){  *phoneNo* = response.getString(**"formatted\_phone\_number"**);  }**else**{  *phoneNo* =**"not available"**;  }   }  **if** (response.has(**"website"**) ){  *website* = response.getString(**"website"**);  }**else**{  *website* =**"not available"**;  }    **if** (response.has(**"permanent\_closed"**)){  *permanently\_closed* = response.getString(**"permanently\_closed"**);  **if** (*permanently\_closed*.equals(**true**))  *permanently\_closed* =**"not available"**;  }**else**{  *permanently\_closed* =**"not available"**;  } |
| **TCase-10.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Place id** | | 1 | ChIJAQMEOyo2zDERbyw\_1WBjCIU | | 2 | ChIJUXdnEphJzDERs6K5gkZKs\_Q | | 3 | ChIJSWxFSolJzDERE1aLg5t-s\_8 | |
| **TCase-10.4 Expected Result** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Test number** | **Open\_now** | **address** | **Phone\_ number** | **website** | **Permanently\_ closed** | | 1 | false | 46, Changkat Bukit Bintang, Bukit Bintang, 50200 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia | 03-2143 2268 | http://thewhiskybarkl.com/ | Not available | | 2 | false | 10, Jalan Telawi 4, Bangsar Baru, 59100 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia | 03-2202 1312 | http://www.tedboy.com/ | Not available | | 3 | false | GF , Mid Valley Megamall , Lingkaran Syed Putra , Mid Valley City , 59200 Kuala Lumpur , Wilayah Persekutuan, Malaysia | 03-2141 6986 | http://www.lavender.com.my | Not available | |
| **TCase-10.5 Test Outcome** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Test number** | **Open\_now** | **address** | **Phone\_ number** | **website** | **Permanently\_ closed** | | 1 | Not available | 46, Changkat Bukit Bintang, Bukit Bintang, 50200 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia | 03-2143 2268 | http://thewhiskybarkl.com/ | Not available | | 2 | Not available | 10, Jalan Telawi 4, Bangsar Baru, 59100 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia | 03-2202 1312 | http://www.tedboy.com/ | Not available | | 3 | Not available | GF , Mid Valley Megamall , Lingkaran Syed Putra , Mid Valley City , 59200 Kuala Lumpur , Wilayah Persekutuan, Malaysia | 03-2141 6986 | http://www.lavender.com.my | Not available | |
| **TCase-10.6 Remark** |
| Open\_now code is wrong and not retrieving from the APIs The rest of the field is retrieving properly |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-11.0 Unit Testing** |
| Updating userSetting |
| **TCase-11.1 Description** |
| To validate the switch button update correctly based on user setting |
| **TCase-11.2 Source Code** |
| String userLastSetting = User.*userSetting*; String userLastSettings = userLastSetting.replace(**'|'**, **' '**); *// for(int i =0; i<userLastSettings.length();i++) {* **if** (userLastSettings.indexOf(**"cafe"**) >= 0) {  cafe.setChecked(**true**); } **if** (userLastSettings.indexOf(**"western"**) >= 0) {  western.setChecked(**true**); } **if** (userLastSettings.indexOf(**"chinese"**) >= 0) {  chinese.setChecked(**true**); } **if** (userLastSettings.indexOf(**"bar"** ) >= 0) {  bar.setChecked(**true**); } **if** (userLastSettings.indexOf(**"japanese"**) >= 0) {  japanese.setChecked(**true**); } **if** (userLastSettings.indexOf(**"korean"**) >= 0) {  korean.setChecked(**true**); } **if** (userLastSettings.indexOf(**"mamak"**) >= 0) {  mamak.setChecked(**true**); } **if** (userLastSettings.indexOf(**"bakery"**) >= 0) {  bakery.setChecked(**true**); } **if** (userLastSettings.indexOf(**"halal"**) >= 0) {  halal.setChecked(**true**); } User.*initiateSetting* =**true**; |
| **TCase-11.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | User.*userSetting* | | 1 | Café|western|chinese | | 2 | Bar|Japanese| | | 3 | - | |
| **TCase-11.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | Café button, western button, Chinese button checked , the rest of the button remain uncheck | | 2 | Bar button and Japanese button checked, the rest of the button remain uncheck | | 3 | All button remain unchecked | |
| **TCase-11.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Result** | | 1 | Café button, western button, Chinese button checked , the rest of the button remain uncheck | | 2 | Bar button and Japanese button checked, the rest of the button remain uncheck | | 3 | All button remain unchecked | |
| **TCase-11.6 Remark** |
| Button updated based on user setting. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-12.0 Unit Testing #12** |
| Update keyword on check |
| **TCase-12.1 Description** |
| To validate the keyword update correctly based on user input |
| **TCase-12.2 Source Code** |
| cafe.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *// The toggle is enabled* cafe.setTextOn(**"on"**);   *//add into the list and indicate variable to run  keyword* += **"|cafe"**;  User.*setUserSetting*(*keyword*); } **else** {  *// The toggle is disabled* cafe.setTextOff(**"off"**);  *keyword* = *keyword*.replace(**"|cafe"**, **""**);  User.*setUserSetting*(*keyword*); }  } });  western.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {   *//add into the list and indicate variable to run  keyword*+= **"|western"**;  User.*setUserSetting*(*keyword*);  } **else** {  *keyword* = *keyword*.replace(**"|western"**,**""**);  User.*setUserSetting*(*keyword*);  }  } }); chinese.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {   *//add into the list and indicate variable to run  keyword* += **"|chinese"**;  User.*setUserSetting*(*keyword*);   *// chinese.setChecked(true);* } **else** { String a = *keyword*.replace(**"|chinese"**,**""**);  *keyword* = a;  User.*setUserSetting*(*keyword*); }  } });  bar.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *//add into the list and indicate variable to run  keyword*+=**"|bar"**;  User.*setUserSetting*(*keyword*);   } **else** {  *keyword* = *keyword*.replace(**"|bar"**,**""**);  User.*setUserSetting*(*keyword*); }  } }); japanese.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {   *//add into the list and indicate variable to run  keyword*+= **"|japanese"**;  User.*setUserSetting*(*keyword*); } **else** {  *keyword* = *keyword*.replace(**"|japanese"**,**""**);  User.*setUserSetting*(*keyword*); }  } }); korean.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *//add into the list and indicate variable to run  keyword*+=**"|korean"**;  User.*setUserSetting*(*keyword*);  } **else** {  *keyword* = *keyword*.replace(**"|korean"**,**""**);  User.*setUserSetting*(*keyword*);  }  } }); mamak.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {   *//add into the list and indicate variable to run  keyword* += **"|mamak"**;  User.*setUserSetting*(*keyword*);} **else** {  *keyword* = *keyword*.replace(**"|mamak"**, **""**);  User.*setUserSetting*(*keyword*);  }  } });  halal.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *//add into the list and indicate variable to run  keyword* += **"|halal"**;  User.*setUserSetting*(*keyword*); } **else** {  *keyword* = *keyword*.replace(**"|halal"**, **""**);  User.*setUserSetting*(*keyword*);}  } });    bakery.setOnCheckedChangeListener(**new** Switch.OnCheckedChangeListener() {  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *//add into the list and indicate variable to run  keyword* += **"|bakery"**;  User.*setUserSetting*(*keyword*);} **else** { *keyword* = *keyword*.replace(**"|bakery"**, **""**);  User.*setUserSetting*(*keyword*); }  } });  Log.*i*(**"keyword"**, User.*getUserSetting*()); |
| **TCase-12.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | Switch checked | | 1 | café, western, chinese | | 2 | japanese, korean, bakery | | 3 | Halal (check twice), mamak | |
| **TCase-12.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **Keyword** | | 1 | café|western|chinese | | 2 | japanese|korean|bakery | | 3 | mamak | |
| **TCase-12.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **Keyword** | | 1 | café|western|chinese | | 2 | japanese|korean|bakery | | 3 | mamak | |
| **TCase-12.6 Remark** |
| Button keyword updated correctly |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-13.0 Unit Testing #13** |
| Updating distance |
| **TCase-13.1 Description** |
| To update the distance seek bar after retrieving it from database |
| **TCase-13.2 Source Code** |
| final SeekBar distance = (SeekBar) rootView.findViewById(R.id.sbdistance);  distance.setProgress(User.getDistance());  distance.incrementProgressBy(500);  distance.setMax(5000);  distance.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  @Override  public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  progress = progress / 1000;  progress = progress \* 1000;  float km = progress / 1000;  distancetv.setText("Distance (" + km + "KM)");  radius = Integer.toString(progress);  if ( progress<999) {  distancetv.setText("Distance (0.5KM)");  radius = Integer.toString(500);  }  }  @Override  public void onStartTrackingTouch(SeekBar seekBar) {  }  @Override  public void onStopTrackingTouch(SeekBar seekBar) {  }  }); |
| **TCase-13.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | User.*getDistance* | | 1 | 5000 | | 2 | 2000 | | 3 | 500 | |
| **TCase-13.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | progress | | 1 | 5000 | | 2 | 2000 | | 3 | 500 | |
| **TCase-13.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | progress | | 1 | 500 | | 2 | 500 | | 3 | 500 | |
| **TCase-13.6 Remark** |
| Progress is bugged. Input doesn’t matter in the progress bar as it will be stuck at 500 Distance seeks display at 500. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-14.0 Unit Testing #14** |
| Update user’s user setting database |
| **TCase-14.1 Description** |
| To update user’s user setting database after they have update their choice of input |
| **TCase-14.2 Source Code** |
| settingAsyncTask task1 = new settingAsyncTask();  task1.execute();  String setBookmark = "http://mofo.16mb.com/setUserSetting.php";  @Override  protected String doInBackground(String... params) {  try {  URL url = new URL(setBookmark);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  httpURLConnection.setRequestMethod("POST");  httpURLConnection.setDoOutput(true);  httpURLConnection.setDoInput(true);  OutputStream outputStream = httpURLConnection.getOutputStream();  BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));  String email = User.getEmail();  String distance = Integer.toString(User.getDistance());  String userSetting = User.getUserSetting();    String data = URLEncoder.encode("email", "UTF-8") + "=" + URLEncoder.encode(email, "UTF-8") + "&" +  URLEncoder.encode("distance", "UTF-8") + "=" + URLEncoder.encode(distance, "UTF-8")+"&"+  URLEncoder.encode("userSetting", "UTF-8") + "=" + URLEncoder.encode(userSetting, "UTF-8");  bufferedWriter.write(data);  bufferedWriter.flush();  bufferedWriter.close();  outputStream.close();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));  StringBuilder stringBuilder = new StringBuilder();  String line = "";  while ((line = bufferedReader.readLine()) != null) {  stringBuilder.append(line + "\n");  Log.i("infohelp: ", "9");  }  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (MalformedURLException e) {  e.printStackTrace();  } catch (IOException e) {  e.printStackTrace();  }  return null;  } |
| **TCase-14.3 Test Data** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **keyword** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | | 3 | User001@gmail.com | 5000 | Halal|mamak | |
| **TCase-14.4 Expected Result** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **database email** | **database distance** | **Database keyword** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | | 3 | User001@gmail.com | 5000 | Halal|mamak | |
| **TCase-14.5 Test Outcome** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **database email** | **database distance** | **Database keyword** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | | 3 | User001@gmail.com | 5000 | Halal|mamak | |
| **TCase-14.6 Remark** |
| User setting database updated correctly. |
| **Unit Test Plan** |
| **TCase-15.0 Unit Testing #15** |
| Retrieve user data and bookmark data |
| **TCase-15.1 Description** |
| To retrieve User data and Bookmark data from database to save into each individual class, upon login. |
| **TCase-15.2 Source Code** |
| **if** (code.equals(**"login\_true"**)) {   JSONArray uArray = jsonObject.getJSONArray(**"user"**);  *//make it into Json object* JSONObject uObject = uArray.getJSONObject(0);  String username= uObject.getString(**"username"**);  String email= uObject.getString(**"email"**); User.*username* = username;  User.*email* = email;   *//get the Json array user\_setting* JSONArray usArray = jsonObject.getJSONArray(**"user\_setting"**);  *//make it into Json object* JSONObject usObject = usArray.getJSONObject(0);   *//user setting part* **int** distance = usObject.getInt(**"distance"**);  String userSetting = usObject.getString(**"userSetting"**);  User.*setDistance*(distance);  User.*setUserSetting*(userSetting);   *//get the Json array user\_setting* JSONArray bArray = jsonObject.getJSONArray(**"bookmark"**);   *//check if its empty* **if** (bArray != **null** && bArray.length() > 0) {  *//make it into Json object* JSONObject bObject = bArray.getJSONObject(0);   **for** (**int** i = 0; i < bArray.length(); i++) {  User.*bookmark*.add(bArray.get(i).toString());  }   **for** (**int** i = 0; i < bArray.length(); i++) {  JSONObject jsonobj = bArray.getJSONObject(i);  String id = jsonobj.getString(**"id"**);  String name = jsonobj.getString(**"name"**);   Bookmark.*bookmarkArrayList*.add(**new** Bookmark(id, name));    }  } } |
| **TCase-15.3 Test Data** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **Database email** | **Database distance** | **Database userSetting** | **Database bookmark** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | ChIJXYntSVRIzDERVljchNG3jYs | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | ChIJXYntSVRIzDERVljchNG3jYs | | 3 | User001@gmail.com | 5000 | Halal|mamak | - | |
| **TCase-15.4 Expected Result** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **userSetting** | **bookmark** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | ChIJXYntSVRIzDERVljchNG3jYs | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | ChIJXYntSVRIzDERVljchNG3jYs | | 3 | User001@gmail.com | 5000 | Halal|mamak | - | |
| **TCase-15.5 Test Outcome** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **userSetting** | **bookmark** | | 1 | Yoyo22@hotmail.com | 3000 | Café|western|chinese | ChIJXYntSVRIzDERVljchNG3jYs | | 2 | [admin@mofo.com](mailto:admin@mofo.com) | 2000 | Japanese|Korean|bakery | ChIJXYntSVRIzDERVljchNG3jYs | | 3 | User001@gmail.com | 5000 | Halal|mamak | - | |
| **TCase-15.6 Remark** |
| The User data and Bookmark data is retrieve correctly. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-16.0 Unit Testing #16** |
| Updating bookmark |
| **TCase-16.1 Description** |
| To show the bookmark list and bookmark database is updated at the same time, based on the input place ID. |
| **TCase-16.2 Source Code** |
| retrieveData.*place\_id = “”*  **HomeFragment.java**  *// The check if bookmarked ID is in the array*  **if** (Bookmark.*findID*(retrieveData.*place\_id*) == **null**) {  *// if it is not*  bookmarkAsyncTask task = **new** bookmarkAsyncTask();  task.execute(**"bookmark"**, retrieveData.*place\_id*, retrieveData.*title*);  } **else** {  Bookmark.*deleteID*(retrieveData.*place\_id*);  }  **Bookmark.java**  *// find the ID position, delete and update the list*  **public static void** deleteID (String id){  **for**(**int** i=0; i<*bookmarkArrayList*.size(); i++) {  **if** (*bookmarkArrayList*.get(i).getId().equals(id)) {   bookmarkAsyncTask task = **new** bookmarkAsyncTask();  task.execute(*bookmarkArrayList*.get(i).getId(),*bookmarkArrayList*.get(i).getName() );  *bookmarkArrayList*.remove(i);  }  }}  **bookmarkAsyncTask.java**  String **setBookmark** = **"http://mofo.16mb.com/setBookmark.php"**;  *// Update online*  @Override  **protected** String doInBackground(String... params) {   String method = params[0];   **try** {  URL url = **new** URL(**setBookmark**);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  httpURLConnection.setRequestMethod(**"POST"**);  httpURLConnection.setDoOutput(**true**);  httpURLConnection.setDoInput(**true**);  OutputStream outputStream = httpURLConnection.getOutputStream();  BufferedWriter bufferedWriter = **new** BufferedWriter(**new** OutputStreamWriter(outputStream, **"UTF-8"**));   String email = User.*email*;  String id = params[1];  String name = params[2];   String data = URLEncoder.*encode*(**"email"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(email, **"UTF-8"**) + **"&"** +URLEncoder.*encode*(**"id"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(id, **"UTF-8"**)+**"&"**+URLEncoder.*encode*(**"name"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(name, **"UTF-8"**);   bufferedWriter.write(data);  bufferedWriter.flush();  bufferedWriter.close();  outputStream.close();   InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = **new** BufferedReader(**new** InputStreamReader(inputStream));  StringBuilder stringBuilder = **new** StringBuilder();  String line = **""**;   **while** ((line = bufferedReader.readLine()) != **null**) {  stringBuilder.append(line + **"\n"**);  }httpURLConnection.disconnect();  **return** stringBuilder.toString().trim();   } **catch** (MalformedURLException e) {  e.printStackTrace();  } **catch** (IOException e) {  e.printStackTrace();  }   **return null**;  }   *// Get the updated status*  @Override  **protected void** onPostExecute(String json) {  **try** {   String s = **""**;  s += json;  JSONObject jsonObject = **new** JSONObject(s);  JSONArray srArray = jsonObject.getJSONArray(**"server\_response"**);  JSONObject srObject = srArray.getJSONObject(0);  *code* = srObject.getString(**"code"**);  *message* = srObject.getString(**"message"**);  *//check its successfully added*  **if** (*code*.equals(**"true"**)) {  Bookmark.*bookmarkArrayList*.add(**new** Bookmark(retrieveData.*place\_id*, retrieveData.*title*));  }  } **catch** (JSONException e1) {    e1.printStackTrace();  }  } } |
| **TCase-16.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Bookmark ID** | | 1 | ChIJXYntSVRIzDERVljchNG3jYs | | 2 | ChIJXYntSVRIzDERVljchNG3jYs | | 3 | ChIJa5Vpq1RIzDER8-Wwe3eFNUI | |
| **TCase-16.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **message responded , user bookmarklist** | | 1 | "added" , [ChIJXYntSVRIzDERVljchNG3jYs] | | 2 | "deleted" ,[] | | 3 | "added" [ChIJa5Vpq1RIzDER8-Wwe3eFNUI] | |
| **TCase-16.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **message responded , user bookmarklist** | | 1 | "added" , [ChIJXYntSVRIzDERVljchNG3jYs] | | 2 | "deleted" ,[] | | 3 | "added", [ChIJa5Vpq1RIzDER8-Wwe3eFNUI] | |
| **TCase-16.6 Remark** |
| The bookmark list and bookmark database updated correctly |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-17.0 Unit Testing #17** |
| On location Change |
| **TCase-17.1 Description** |
| To check if the device listen to the location changes |
| **TCase-17.2 Source Code** |
| **locationListener** = **new** LocationListener() {   *//on changes, display location* @Override  **public void** onLocationChanged(Location location) {  **if** (ActivityCompat.*checkSelfPermission*(getActivity(), Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(getActivity(), Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  **return**;  }  **locations** = **locationManager**.getLastKnownLocation(**provider**);  displayLocation();   Log.*i*(**"info "**, **"onlocationchange"**);  } |
| **TCase-17.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Input Location** | | 1 | geo fix 101.671317 3.152209 | | 2 | geo fix 102.671317 3.252209 | | 3 | geo fix 104.111111 4.111111 | |
| **TCase-17.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | I/info : onlocation change | | 2 | I/info : onlocation change | | 3 | I/info : onlocation change | |
| **TCase-17.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | I/info : onlocation change | | 2 | I/info : onlocation change | | 3 | I/info : onlocation change | |
| **TCase-17.6 Remark** |
| The location listen work correctly. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-18.0 Unit Testing #18** |
| GPS provider enabling |
| **TCase-18.1 Description** |
| To check if the application listen to the devices GPS enabling. |
| **TCase-18.2 Source Code** |
| @Override **public void** onProviderEnabled(String provider) {  **locationManager** = (LocationManager) getActivity().getSystemService(Context.***LOCATION\_SERVICE***);   *//check if its enable* **if** (**locationManager**.isProviderEnabled(LocationManager.***GPS\_PROVIDER***)) {   *//check permission, required by system* **if** (ActivityCompat.*checkSelfPermission*(getContext(), Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(getContext(), Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  **return**;  }   *//update the location base on GPS* **locationManager**.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, **locationListener**);  *//display msg if no location available,else check if findFood method is ran* **if** (!checkGPS()){  Toast.*makeText*(getContext(), **"GPS location not available"**, Toast.***LENGTH\_LONG***).show();  }**else**{  Toast.*makeText*(getContext(), **"GPS location available."**, Toast.***LENGTH\_LONG***).show();  }   }  } } |
| **TCase-18.3 Test Data** |
| Start running with device’s GPS off, and on it again after application start.   |  |  | | --- | --- | | **Test number** | **Input Location** | | 1 | - | | 2 | geo fix 102.671317 3.252209 | |
| **TCase-18.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | Toast message “GPS location not available” appear | | 2 | Toast message “GPS location available” appear | |
| **TCase-18.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | Toast message “GPS location not available” appear | | 2 | Toast message “GPS location available” appear | |
| **TCase-18.6 Remark** |
| The device GPS enabling is listened by the application. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-19.0 Unit Testing #19** |
| GPS provider disabling |
| **TCase-19.1 Description** |
| To check if the application listen to the devices GPS disabling, and display an alert dialog which direct to system location page when ok is clicked. |
| **TCase-19.2 Source Code** |
| @Override  **public void** onProviderDisabled(String provider) {  *//bring user to GPS setting page*  toGPS();   }  **public void** toGPS(){  **GPSalreadyExceuted**=**true**;  *//display a alert box that bring user to turn on GPS* **builder** = **new** AlertDialog.Builder(getContext());  **builder**.setTitle(**"GPS Disabled"**);  **builder**.setMessage(**"Please turn on the GPS to allow us to locate you"**);  **builder**.setPositiveButton(**"OK"**, **new** DialogInterface.OnClickListener() {   *//click ok bring user to turn GPS on* @Override  **public void** onClick(DialogInterface dialog, **int** which) {  Intent intent = **new** Intent(Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  startActivity(intent);  dialog.dismiss();  }  });  AlertDialog alertDialog = **builder**.create();  alertDialog.show(); } |
| **TCase-19.3 Test Data** |
| |  |  | | --- | --- | | **Test number** | **Input Location** | | 1 | Start running with device’s GPS on, and off it after application start. | | 2 | Start running with device’s GPS off, and on it after application start, then off it again. | |
| **TCase-19.4 Expected Result** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | Alert dialog appear after GPS is turned off. Directed to system location page when ok is clicked. | | 2 | Alert dialog appear after GPS is turned off. Directed to system location page when ok is clicked. | |
| **TCase-19.5 Test Outcome** |
| |  |  | | --- | --- | | **Test number** | **responded** | | 1 | Alert dialog appear after GPS is turned off. Upon clicking ok on alert dialog, page is directed to system location page. | | 2 | Alert dialog appear after GPS is turned off. Upon clicking ok on alert dialog, page is directed to system location page. | |
| **TCase-19.6 Remark** |
| The device GPS disabling is listened by the application. Alert dialog working properly by directing to system login page. |

|  |
| --- |
| **Unit Test Plan** |
| **TCase-20.0 Unit Testing #20** |
| Updating post like count |
| **TCase-20.1 Description** |
| To retrieving and setting the number of current post like |
| **TCase-20.2 Source Code** |
| String register\_url2 = "http://mofo.16mb.com/setPostLike.php";  **protected** String doInBackground(String... params) {   String method = params[0];   **try** {   **if** (method.equals(**"setPostLike"**)) {  **url** = **new** URL(**register\_url2**);  }   HttpURLConnection httpURLConnection = (HttpURLConnection) **url**.openConnection();  httpURLConnection.setRequestMethod(**"POST"**);   httpURLConnection.setDoOutput(**true**);  httpURLConnection.setDoInput(**true**);  OutputStream outputStream = httpURLConnection.getOutputStream();  BufferedWriter bufferedWriter = **new** BufferedWriter(**new** OutputStreamWriter(outputStream, **"UTF-8"**));   String id = params[1];  String username = User.*getUsername*();  String data = URLEncoder.*encode*(**"id"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(id, **"UTF-8"**) + **"&"** + URLEncoder.*encode*(**"username"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(username, **"UTF-8"**);   bufferedWriter.write(data);  bufferedWriter.flush();  bufferedWriter.close();  outputStream.close();   InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = **new** BufferedReader(**new** InputStreamReader(inputStream));  StringBuilder stringBuilder = **new** StringBuilder();  String line = **""**;    **while** ((line = bufferedReader.readLine()) != **null**) {  stringBuilder.append(line + **"\n"**);  }  httpURLConnection.disconnect();  **return** stringBuilder.toString().trim();  } **catch** (MalformedURLException e) {    e.printStackTrace();   } **catch** (IOException e) {  e.printStackTrace();   }  **return null**;  }   @Override  **protected void** onPostExecute(String json) {  **try** {   String s = **""**;  s += json;  JSONObject jsonObject = **new** JSONObject(s);  JSONArray srArray = jsonObject.getJSONArray(**"server\_response"**);  JSONObject srObject = srArray.getJSONObject(0);   *code* = srObject.getString(**"code"**);  *total\_like* = srObject.getString(**"total\_like"**);   } **catch** (JSONException e1) {  e1.printStackTrace();  } } |
| **TCase-20.3 Test Data** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **username** | **method** | **id** | | 1 | Yoyo22 | setPostLike | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | | 2 | Yoyo22 | setPostLike | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | | 3 | Yoyo22 | setPostLike | ChIJAQMEOyo2zDERbyw\_1WBjCIU | |
| **TCase-20.4 Expected Result** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **id** | **code** | **Total like** | | 1 | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | true | 2 | | 2 | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | false | 1 | | 3 | ChIJAQMEOyo2zDERbyw\_1WBjCIU | true | 3 | |
| **TCase-20.5 Test Outcome** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **id** | **code** | **Total like** | | 1 | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | true | 2 | | 2 | ChIJBZw7Sm1IzDEREX\_rq97zIU4 | false | 1 | | 3 | ChIJAQMEOyo2zDERbyw\_1WBjCIU | true | 3 | |
| **TCase-20.6 Remark** |
| Like updated correctly. |

### Integration Testing

|  |
| --- |
| **Integration Test Plan** |
| **TCase-21.0 Integration Testing #1** |
| Register button for login page |
| **TCase-21.1 Description** |
| To test whether the login page will be directed to register page when register button was click |
| **TCase-21.2 Diagram** |
| **login page long arrow register** |
| **TCase-20.3 Test Data** |
| Signup button is clicked |
| **TCase-21.4 Expected Result** |
| Directed to register page |
| **TCase-21.5 Test Outcome** |
| Directed to register page |
| **TCase-21.6 Remark** |
| Page directed without any problem |

|  |
| --- |
| **Integration Test Plan** |
| **TCase-22.0 Integration Testing #2** |
| Cancel button for register page |
| **TCase-22.1 Description** |
| To test whether the register page will be directed to login page when cancel button was click |
| **TCase-22.2 Diagram** |
| **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\register.jpg C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\login page.jpg** |
| **TCase-22.3 Test Data** |
| Cancel button is clicked |
| **TCase-22.4 Expected Result** |
| Directed to login page |
| **TCase-22.5 Test Outcome** |
| Directed to login page |
| **TCase-22.6 Remark** |
| Page directed without any problem |

|  |
| --- |
| **Integration Test Plan** |
| **TCase-23.0 Integration Testing #3** |
| Login button for login page |
| **TCase-23.1 Description** |
| To test whether the login page will be directed to setting page when login button was click |
| **TCase-23.2 Diagram** |
| **login page long arrow Setting 1** |
| **TCase-23.3 Test Data** |
| Login button is clicked |
| **TCase-23.4 Expected Result** |
| Directed to setting page |
| **TCase-23.5 Test Outcome** |
| Directed to setting page |
| **TCase-23.6 Remark** |
| Page directed without any problem |

|  |
| --- |
| **Integration Test Plan** |
| **TCase-24.0 Integration Testing #4** |
| Main Activity transition |
| **TCase-24.1 Description** |
| To test whether the Home page, Setting page, Bookmark page is directed when its name is click in the side menu. |
| **TCase-24.2 Diagram** |
| **Setting 1 C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg** **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg Home (GPS location not available) 2**  **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg Bookmark clear C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg** |
| **TCase-24.3 Test Data** |
| Side menu button is click  Either home button, setting button, or bookmark button is click. |
| **TCase-24.4 Expected Result** |
| Directed to the page of its name when click on side menu. |
| **TCase-24.5 Test Outcome** |
| Directed to the page of its name when click on side menu. |
| **TCase-24.6 Remark** |
| Page directed without any problem. |

### System testing

|  |
| --- |
| **System Test Plan** |
| **TCase-25.0 System Testing #1** |
| Register |
| **TCase-25.1 Description** |
| To test whether the user will be able to register an account when the input is valid and is never use to register before |
| **TCase-25.2 Diagram** |
| **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\register (sucess1).jpgC:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpgC:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\register (sucess).jpgC:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpgC:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\login page.jpg** |
| **TCase-25.3 Test Data** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **name** | **email** | **password** | **conpassword** | | 1 | user005 | user005@gmail.com | user005 | user005 | | 2 | user006 | user005@gmail.com | user006 | user005 | | 3 | yoyo25 | yoyo25@hotmail.com | yoyo25 | yoyo25 | |
| **TCase-25.4 Expected Result** |
| “Registration success” toast message appears, followed by the register page being directed to login page |
| **TCase-25.5 Test Outcome** |
| Toast message appear, register page being directed to login page. |
| **TCase-25.6 Remark** |
| Toast message displayed correctly. |

|  |
| --- |
| **System Test Plan** |
| **TCase-26.0 System Testing #2** |
| Login |
| **TCase-26.1 Description** |
| To test if registered member will be able to login to their account |
| **TCase-26.2 Diagram** |
| **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg E:\Files\Q\AndroidStudioProjects\WORKING BACKUP FOLDER\Screenshot\Setting (welcome page).JPG** |
| **TCase-26.3 Test Data** |
| Login pressed with correct email and password input |
| **TCase-26.4 Expected Result** |
| Toast message appear to great user |
| **TCase-26.5 Test Outcome** |
| Toast message appear “welcome back,” together with user name |
| **TCase-26.6 Remark** |
| Toast message displayed correctly. |

|  |
| --- |
| **System Test Plan** |
| **TCase-27.0 System Testing #3** |
| Adjust Setting |
| **TCase-27.1 Description** |
| To test if user’s setting is save and retrievable upon login, by setting the setting and re-log into the application |
| **TCase-27.2 Diagram** |
| **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Setting 1.jpg** |
| **TCase-27.3 Test Data** |
| |  |  |  |  | | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **cateogry** | | 1 | user001@gmail.com | 3km | Café, western | | 2 | user002@gmail.com | 2km | Bar, Japanese | | 3 | yoyo22@hotmail.com | 5km | Mamak, halal, western | |
| **TCase-27.4 Expected Result** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **database distance** | **cateogry** | | 1 | user001@gmail.com | 3km | 3km | Café, western | | 2 | user002@gmail.com | 2km | 2km | Bar, Japanese | | 3 | yoyo22@hotmail.com | 5km | 5km | Mamak, halal, western | |
| **TCase-27.5 Test Outcome** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Test number** | **email** | **distance** | **database distance** | **cateogry** | | 1 | user001@gmail.com | 0.5km | 3km | Café, western | | 2 | user002@gmail.com | 0.5km | 2km | Bar, Japanese | | 3 | yoyo22@hotmail.com | 0.5km | 5km | Mamak, halal, western | |
| **TCase-27.6 Remark** |
| The distance is bugged. Distance saved in database but irretrievable. |

|  |
| --- |
| **System Test Plan** |
| **TCase-28.0 System Testing #4** |
| View Post |
| **TCase-28.1 Description** |
| To view the Post retrieve base on user’s location and setting |
| **TCase-28.2 Diagram** |
| **SideMenu C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Home (GPS location available).jpg**  C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Home (GPS location available) 2.jpg |
| **TCase-28.3 Test Data** |
| Home button pressed |
| **TCase-28.4 Expected Result** |
| Post is retrieved. Name of the food place, picture, likes, and phone number is displayed |
| **TCase-28.5 Test Outcome** |
| Post is retrieved. Name of the food place, picture, likes, and phone number is displayed |
| **TCase-28.6 Remark** |
| Post displayed correctly |

|  |
| --- |
| **System Test Plan** |
| **TCase-29.0 System Testing #5** |
| View Post Information |
| **TCase-29.1 Description** |
| To test if the post information is displayed on the directed page when more info button is clicked |
| **TCase-29.2 Diagram** |
| C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Home (GPS location available) 2.jpg **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg more info** |
| **TCase-29.3 Test Data** |
| More info button pressed |
| **TCase-29.4 Expected Result** |
| Home page is directed to a more info page with additional information of the previous screen food page |
| **TCase-29.5 Test Outcome** |
| Home page is directed to more info page Additional detail displayed |
| **TCase-29.6 Remark** |
| Home page is directed to more info page correctly Information is retrieved and display correctly on the page. |
| **System Test Plan** |
| **TCase-30.0 System Testing #6** |
| Review post |
| **TCase-30.1 Description** |
| To test if the post like is updated and synchronize across all pages |
| **TCase-30.2 Diagram** |
| like3 **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg like2** |
| **TCase-30.3 Test Data** |
| Like button is press |
| **TCase-30.4 Expected Result** |
| The number of like is updated and synchronize across all pages from home page, more info, and bookmark page. |
| **TCase-30.5 Test Outcome** |
| Number of like on the post is updated on all pages including home, more info and bookmark. |
| **TCase-30.6 Remark** |
| Like updated correctly across all pages and network. |

|  |
| --- |
| **System Test Plan** |
| **TCase-31.0 System Testing #7** |
| Bookmark post |
| **TCase-31.1 Description** |
| To test if the bookmarked post appear in bookmark list |
| **TCase-31.2 Diagram** |
| C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Home (GPS location available) 2.jpg **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Home (bookmarked).jpg**  **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg** C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Bookmark.jpg |
| **TCase-31.3 Test Data** |
| Bookmark button is press on home page Click on side menu and tap bookmark |
| **TCase-31.4 Expected Result** |
| The recent bookmarked restaurant title is updated and appeared in bookmark page |
| **TCase-31.5 Test Outcome** |
| Bookmarked restaurant title appear in bookmark page |
| **TCase-31.6 Remark** |
| The bookmark updated correctly. |

|  |
| --- |
| **System Test Plan** |
| **TCase-32.0 System Testing #8** |
| View bookmarked post |
| **TCase-32.1 Description** |
| To test if the bookmarked post information is retrievable |
| **TCase-32.2 Diagram** |
| C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Bookmark.jpg **C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\long arrow.jpg C:\Users\Home\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Info (bookmark).jpg** |
| **TCase-32.3 Test Data** |
| Not applicable |
| **TCase-32.4 Expected Result** |
| Clicking on any bookmarked post on bookmark page will retrieve the food place information and display it on information page |
| **TCase-32.5 Test Outcome** |
| Bookmark page is directed to information page  The post details including image, contact number and website is retrieve and displayed |
| **TCase-32.6 Remark** |
| The bookmark page display correctly information retrieve and displayed correctly |

**Test Summary**  
Mofo Application is a food searching application. It consists of different module including login, register, home, more information, setting, and bookmark. There were a total number of 32 testing planned, and all of there were executed. All test cases have a positive result, except 3. One where the setting page’s distance was not being able to retrieve and the other is the empty result return when price is entered in the Google PlaceAPI. The last one is the opening now of the entire place APIs is not able to retrieve due to the code error.

White box testing, black box testing and the included system testing, is carried out to test the functional and non-function of the application. Acceptance testing was carried out as there was user testing in alpha stage. Feedback were collected to further improve the application and to fix a certain issues. At the completion of every module, unit test was carried out, followed by the integration test after parts was put together. The application is then assess to see if the functional and non-functional requirement is met.

There was a problem during non-functional testing, where tester do not know what to do after the setting pages, as the side menu bar is not clickable on certain smartphone. Other than that, the result given by the tester was all good.

**Testing tools**

Android studio has built in emulator, which work with the build in debug mode, logging all the errors occur when navigating the application. In addition, by putting Log.i(“Info”, ” increasing number”) between every part of the code, error can be easily identify as it shows where the code stop running. This speed up the debug process as sometime the error logged by the debug mode is more of a general statement instead of highlighting the cause of problem. From there we can focus on the breaking point and make changes base on the debug error statement.

The application is also export and test on 2 physical smartphones, which is Xiaomi’s Note and Samsung’s Galaxy Note2. Application stress test is done by tester on these 2 physical devices, where they perform a series of random event clicking and entering input into the application itself. Before the stress test is carried out, the phone is linked to the computer with android studio opened, so it can log any error or crashes occur during stress test. Out of 80 random events executed from both phone, providing 40 tests each, there is 19crashes occur in total. 2crashes happen in the absent of internet, while the other 8 crashes were cause by tester’s multiple inputs on single button, thinking the button is not responding when it’s actually processing and connecting to the internet. There were other 9crashes happen due to the compatible of the smartphone as I figured out by searching error logged, as it works fine when the same step and procedure was carried out on the same device.

A FOOD SEARCHING APPLICATION



Training/ Education Strategy / Installation

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BIT302 Software Engineering:  
Assignment 2  
  
**Author: B1402427 CHIN WEI QING  
Lecturer: MS. SEETHA**13 April 2016

**Contact Detail:** [**WEIQING.CHIN@GMAIL.COM**](mailto:WEIQING.CHIN@GMAIL.COM)

**Abstract**

As a brand new application, the interface and control might not be clear enough for user. Guidelines are provided to teach user the basic control of the application as well as introduce them the interface of the application.

**Installation Guide**

1. Visit Play Store application in Android smartphone.
2. Go to the search bar and key in “Mofo food”.
3. Click on the search.
4. Click the result showing “Mofo Food searching app”.
5. Click on install, and click accept on the permission alert box.
6. Wait for the application to be installed.
7. Click open, or find and select the Mofo app through the smart phone menu.

**Troubleshooting**

Unable to find the application

1. Android smartphone with version lower than 4.4 will not be able to find the application in the Play Store.
2. If android version is 4.4 and above, please recheck the entered keyword.
   1. Make sure the keyword entered is “Mofo food”.
   2. Scroll through the list and check if it’s there.
   3. If it’s still not there, enter keyword “Mofo Food searching app” in the search bar and search again.

Unable to download

1. Check the Android smartphone’s version and make sure if it meet the application minimal size requirement.
   1. Android smartphone version can be located in:
      1. Setting > About phone > Android version   
         \*(may vary from different smartphone model)
   2. Application download size can be found by clicking into the “read more” above the slide picture and rating.

Tutorial

1. To use Mofo Application, the application installed android smartphone must be connected to internet through data or Wifi, and GPS enabled.
2. Upon starting the app, user are required to login to Mofo Application, or register one.
3. Once login, user can choose the range to search, as well as the choice of their liking.
4. Tap on the top right menu icon, or slide your finger right on the screen to access the menu.
5. Tap home, and the food place matches the user input will be display.
6. You can like the post, bookmark it, or click more info to view its information.
7. To go back from more info, tap on the back button on the smartphone.
8. Bookmarked post can be access thru the menu.

Frequency Asked Question (FAQ)

**Why must I create an account in order to use the Application?**A Mofo account is needed so we can save the bookmarked post of your choice and get it back to you when u needs it.

**Why is the Home page not showing anything even though I have the data and GPS turned on?**There can be a few reasons to it. Either there is no result that matches your choice and range, or the GPS is not getting your location even though it’s turned on. This can happen when you are in a covered area, or building. Try moving around under open area for the GPS to locate your position.

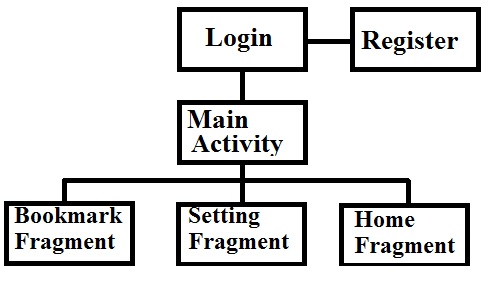
**Why isn’t the application showing any restaurant picture?**There is currently no available picture for the particular restaurant.   
  
**Why isn’t the app showing the opening time for the restaurant?**   
Least assures we are trying our best to get that cover, but sadly our team is outnumbered by the amount of food places available. We’ll have that updated as soon as possible, stay tune!

**Restaurant around my area are not showing in the application?**We are trying our best expanding the database as the moment. Please hold on while we upgrade to serve you better.

**Why is there no logout button? How can I re-log into the app without it?**To go back to login, simple press back either home, setting, or bookmark page.

**I have a question or suggestion. How do I contact the developer?**For any feedback or enquiry, please write in to [qdevtestingemail@gmail.com](mailto:qdevtestingemail@gmail.com). We’ll get back to you as soon as possible.

**Top down Integration**



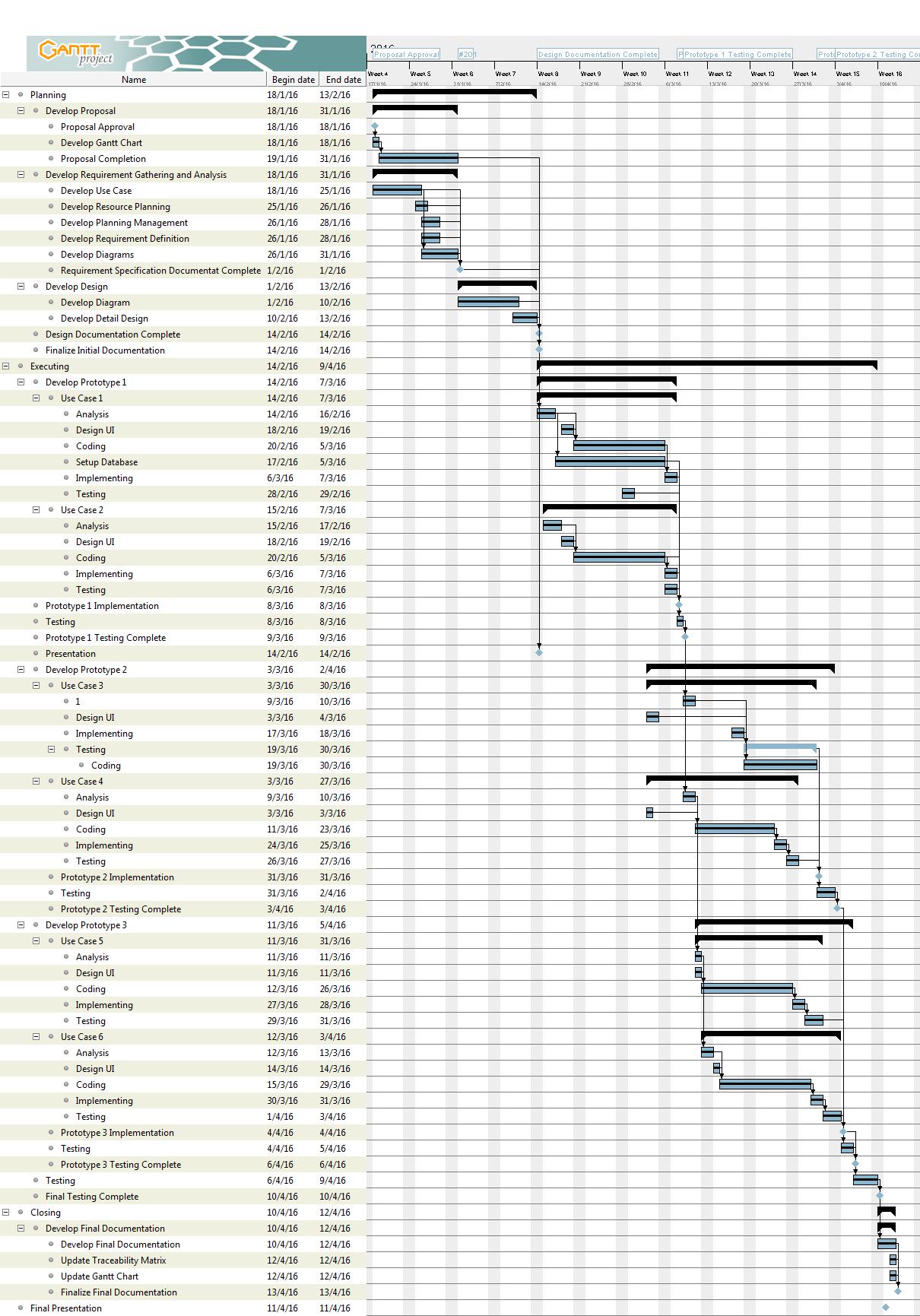
Top down integration is carried out because it’s an incremental approach. Top down is also chosen for the benefit of early testing of major function. For such, Login and Home module is done before moving on to Bookmark module. This is because Bookmark can only be done when a proper place id and user is given. With the user id provided by Login, and place id provided by Home, bookmark data can successfully be save and retrieve after user login. As a result, no stubs are required and it’s easier to test the module and cases.

Even though top down integration is chosen for the application integration testing, it is not followed to the dot. Module with similar functional is created and tested before moving on to other part. For example, login and register is tested as one while setting and home fragment is tested as another. On the other hand, all other modules are tested each individually before merging.

In this integration, white box testing and black box testing is carried out. In unit testing, white box testing is used to find any miscode or logical error, to ensure no error will occur in later progression. Black box testing is carried out at the end of every phase, where testers are asked to try out without developer guidance.

**Requirements Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement** | **In Design Specification** | **In Code** | **In Test Plan** | **In Implementation Plan** | **Notes** |
| 1.1 Registration | Figure 2.1 | LoginActivity.java | 1, 2, 3, 4, 5, 22, 25 | - |  |
| 1.2 Login | Figure 2.2 | RegisterActivity.java  User.java | 6, 7, 15, 21, 26 | - |  |
| 1.3 Adjust Setting | Figure 2.3 | SettingFragment.java  setting.Asynctask.java | 11, 12, 13, 14, 23, 24, 27 | - |  |
| 1.4 View Post | Figure 2.4 | HomeFragment.java  APIDownloadTask.java  retrieveData.java  retrieveImage.java  retrievePostLike.java | 8, 9, 17, 18, 19, 24, 28 | - |  |
| 1.5 View Post Information | Figure 2.5 | InformationActivity.java  retrieveData.java  retrieveImage.java  retrievePostLike.java | 10,29 | - |  |
| 1.6 Review Post | Figure 2.6 | retrievePostLike.java | 20, 24, 30 | - |  |
| 1.7 Bookmark Post | Figure 2.7 | Bookmark.java  BookmarkAdapter.java  bookmarkAsynctask.java | 16, 31 | - |  |
| 1.8 View Bookmark | Figure 2.8 | BookmarkFragment.java  Bookmark.java  BookmarkAdapter.java  bookmarkAsynctask.java | 12, 32 | - |  |

**Gann Chart**  


**Appendix**

**Individual Contribution Report**

Solo on an assignment mean hundred percent of the contribution came from a single person. From login to bookmark, every single interface design, function implementation, or documentation is done by me. Each function is first build separately, than implement it together as one. Other than bookmark, other unit test cases is made and tested each individually. This is because bookmark requires user id and the place data to work. Integration testing is done in part to ease the bug finding process, and to ensure the navigation flow work correctly. System testing was carried out upon completion of the application. Lastly, the System testing was done all of the above, to conclude the coding phase. For the documentation, guidance was provided by peers to ensure format written is correct.

Developing alone is especially hard when everything from platform to coding is very new to you. New language such as PHP, and the platform Android studio, cause the developments curve to be as steep as possible. Without prior experience with android studio and PHP, it’s a learning phase till the completion of the coding. But with internet, tutorial video are widely available online and samples code can also be found though Stack Overflow or GitHub. That being said, the recent depreciation of code in Android Studio makes implantation very different from how other java platform use to be, and widely cut down the resources available online. This also happen in PHP, which gives unexpected problem when connecting it from Android Studio. By reading the APIs, understanding the basic flows of Android Studio coding through online videos, and with the help of similar solution from Stack Overflow, the implementation phase is completed through try and error.

Doing alone often make one getting blinded by the obvious. To get a different point of view, feedback and evaluation is given by peer throughout the development phase. While facing unknown error when retrieving data from PHP, insight is given by peers to solve the problem. Discussing with peers who are facing similar problem in Android Studio also gives us better idea on how to solve the problem.

There are times where problem could not be solved. Such as the lack of information in Google maps API, causing the search functions to be unfavourable if search details such as time or price are entered along. This can only be overcome by reducing the application search function, hence not meeting a certain requirement. There is this one unsolved problem in the setting’s page, whereby the distance is not saveable.