**Chapter 7**

**Traceability Record**

**Emergency Information on Mobile**

Traceability Record

By

Putchakarn Jaikon 542115031

Sawatdiporn Kitirot 542115065

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

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

**Aj.Chartchai Doungsa-ard**

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Details** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | |
| **EIOM-TraceabilityRecord-V.0.1.docx** | **Chapter 1**      Introduction | Draft | 1/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.0.2.docx** | **Chapter 2**      Traceability Record Table | Draft | 2/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.0.3.docx** | **Modify Chapter 2**      Traceability Record Table | Draft | 3/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.1.0.docx** |     Add Table of content and cover page | Release | 4/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.1.1.docx** |  Modify chapter 1-2 | Release | 30/7/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.1.5.docx** |  add table and appendix for progress 2 | Daft | 15/9/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.2.0docx** |  Review and modify all table and appendix | Release | 21/10/2014 | PJ, SK, CD | PJ, SK | PJ, SK |
| **EIOM- TraceabilityRecord -V.3.0docx** |  Review and combine all task table and appendix | Release | 11/12/2014 | PJ, SK, CD | PJ, SK | PJ, SK |

PJ – Putchakarn Jaikon, SK – Sawatdiporn Kitirot, CD – Chartchai Doungsa-ard

Table of Contents

[Chapter One | Introduction 4](#_Toc406062743)

[1.1 Purpose 4](#_Toc406062744)

[1.2 Project Scope 4](#_Toc406062745)

[Chapter Two | Traceability Matrix Table 7](#_Toc406062746)

[2.1 User Requirement Specification and System Requirement Specification 7](#_Toc406062747)

[2.2 User Requirement Specification and Use Case 9](#_Toc406062748)

[2.3 User Requirement Specification and Sequence Diagram 11](#_Toc406062749)

[2.4 User Requirement Specification and User Interface Design 13](#_Toc406062751)

[2.5 User Requirement Specification and Method Description 15](#_Toc406062752)

[2.6 Class Diagram Server and Method Description 17](#_Toc406062753)

[2.7 Unit Test Case and Method Description 20](#_Toc406062754)

[Chapter Three| Traceability Record Table 22](#_Toc406062756)

[Chapter Four | Appendix 24](#_Toc406062757)

[4.1 User Requirement Specification 24](#_Toc406062758)

[4.2 System Requirement Specification 25](#_Toc406062759)

[4.3 Use Case 31](#_Toc406062760)

[4.4 Sequence Diagram 33](#_Toc406062761)

[4.5 User Interface Design 34](#_Toc406062762)

[4.6 Method Description 35](#_Toc406062763)

[4.7 Class Diagram 40](#_Toc406062764)

[4.8 Unit Test Case 41](#_Toc406062765)

# Chapter One | Introduction

## Purpose

The purpose of traceability record for Emergency Information on Mobile project is to show the relation of the project. The traceability is linked between user requirements to system requirement specification, use case, sequence diagram and user interface design.

## Project Scope

Emergency Information on Mobile is an application that runs on android operating system. The application provides online map and offline map to help people about the information of the help pace. Emergency Information on Mobile will provide the offline map with information around the user when they lost Internet connection.

The main features of EIOM will be following:

**Mobile Part**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Name** | **User Requirement Specification** |
| **Name** |
| #1. | Map and help information system | The user can view the online map with their current location. |
| The user can view the offline map with their current location. |
| The user can view the help places in online map. |
| The user can view the help places in offline map. |
| The user can view help information of each help place in online map |
| The user can view help information of each help place in offline map |
| The user can make emergency call to each help place in online map |
| The user can make emergency call to each help place in online map |
| The user can view the route of distance between the current locations of user to the destination. |
| The user can view details of each point on routing the direction. |
| #2. | Search information system | The user can search help place’s name by keyword in online map. |
| The user can find the nearest help place by selection the category in online map |
| #3. | Rating location | The user can rate the help place. |
| The user can view average rating score of each help place in online map. |
| The user can view average rating score of each help place in offline map. |
| #4. | Automatic collecting data system | The user can set the scope for downloading data. |
| The mobile application can collect help place information automatically |

**Server Part**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Name** | **User Requirement Specification** |
| **Name** |
| #5. | Manage information system | The administrator can add help place’s information, which includes name, address, district, province, zip code, phone number, category, latitude and longitude. |
| The administrator can edit help place’s information, which includes name, address, district, province, zip code, phone number, category, latitude and longitude. |
| The administrator can remove help place. |
| The administrator can view help information of each help place. |
| The administrator can browse the help place by category. |
| The administrator can browse the help place by province of Thailand. |
| The administrator can browse the help place by category and province of Thailand. |
| The mobile application can get list of all help places in the database. |
| The mobile application can get the nearest help place by the selected category. |
| The mobile application can get list of help places where locate in the setting scope. |
| The administrator can login to the system. |
| The administrator can logout of the system. |
| The administrator can update their account’s password. |
| #6. | Support information for mobile | The mobile application can get list of all help places in the database. |
| The mobile application can get the nearest help place by the selected category. |
| The mobile application can get list of help places where locate in the setting scope. |
| The mobile application can retrieve the help place with new average rating score. |

The document will include Traceability Record, Requirement Specification, User Requirement Specification, Class Diagram, Sequence Diagram and User Interface Design.

**Traceability Record** is a document which shows relation of the entire project.

# Chapter Two | Traceability Matrix Table

## 2.1 User Requirement Specification and System Requirement Specification

* Mobile Part



* Server Part



## 2.2 User Requirement Specification and Use Case

* Mobile Part



* Server Part



## 2.3 User Requirement Specification and Sequence Diagram

* Mobile Part

## 

* Server Part



## 

## 2.4 User Requirement Specification and User Interface Design

* Mobile Part



* Server Part



## 2.5 User Requirement Specification and Method Description

* Mobile Part



* Server Part





## 2.6 Class Diagram Server and Method Description

* Mobile Part



* Server Part





## 2.7 Unit Test Case and Method Description

* Mobile Part



* Server Part

## 

## 

# Chapter Three| Traceability Record Table

* Mobile Part



* Server Part



# Chapter Four | Appendix

## 4.1 User Requirement Specification

* Mobile Part

**URSm-01:** The user can view the online map with their current location.

**URSm-02:** The user can view the offline map with their current location.

**URSm-03:** The user can view the help places in online map.

**URSm-04:** The user can view the help places in offline map.

**URSm-05:** The user can view help information of each help place in online map.

**URSm-06:** The user can view help information of each help place in offline map.

**URSm-07:** The user can make emergency call to each help place in online map.

**URSm-08:** The user can make emergency call to each help place in offline map.

**URSm-13:** The user can view the route of distance between the current locations of user to the destination.

**URSm-14:** The user can view details of each point on routing the direction.

**URSm-09**: The user can search help place’s name by keyword in online map.

**URSm-10:** The user can find the nearest help place by selection the category in online map.

**URSm-15:** The user can view average rating score of each help place in online map.

**URSm-16**: The user can view average rating score of each help place in offline map.

**URSm-17:** The user can rate the help place in online map.

**URSm-11:** The user can set the scope for downloading data.

**URSm-12:** The mobile application can collect help place information automatically.

* Server Part

**URSs-01:** The administrator can add help place’s information, which include name, address, district, province, zip code, phone number, category, latitude and longitude.

**URSs-02:** The administrator can edit help place’s information, which include name, address, district, province, zip code, phone number, category, latitude and longitude.

**URSs-03:** The administrator can remove help place.

**URSs-04:** The administrator can view help information of each help place.

**URSs-05:** The administrator can browse the help place by category.

**URSs-06:** The administrator can browse the help place by province of Thailand.

**URSs-07**: The administrator can browse the help place by category and province of Thailand.

**URSs-08**: The administrator can login to the system.

**URSs-09:** The administrator can logout of the system.

**URSs-10**: The administrator can update their account’s password.

**URSs-11:** The mobile application can get list of all help places in the database.

**URSs-12:** The mobile application can get the nearest help place by the selected category.

**URSs-13:** The mobile application can get list of help places where locate in the setting scope.

**URSs-14:** The mobile application can retrieve the help place with new average rating score.

## 4.2 System Requirement Specification

* Mobile Part

1. The system shall obtain the latitude and longitude of the user’s current location.
2. The system shall show the online map UI.
3. SRS-: The system shall show the user’s current location on the online map.
4. The system shall obtain the latitude and longitude of the user’s current location.
5. The system shall check MapsWithMe application is installed in the device.
6. The system shall show dialog to offer user download MapsWithMe application.
7. The system shall connect MapsWithMe application.
8. SRS-: The system shall show the offline map UI.
9. SRS-: The system shall show the user’s current location on the offline map.
10. The system shall show the online map UI
11. The system will retrieve all help places from server.
12. The system shall input marker of help places into online map UI.
13. The system shall show all markers of help places on an online map UI.
14. The system shall check MapsWithMe application is installed in device.
15. The system shall show dialog to offer user download MapsWithMe application**.**
16. The system shall connect MapsWithMe application**.**
17. The system shall show the offline map UI**.**
18. The system shall retrieve the loaded help places from the user’s device**.**
19. The system shall input marker of help places into offline map UI.
20. The system shall show all markers of help places on an offline map UI.
21. The system shall receive the help place object that user selected
22. The system shall retrieve information of help place object.
23. The system provides information UI to show the help information, which are name, address, district, province, zip code, and phone number.
24. The system shall receive the help place object that user selected.
25. The system shall retrieve information of help place object.
26. The system provides information UI to show the help information, which are name, address, district, province, zip code, and phone number.
27. The system shall receive the help place object that user selected from offline map.
28. The system shall retrieve information of help place object.
29. The system provides information UI to show the help information, which are name, address, district, province, zip code, and phone number.
30. The system shall provide the call UI.
31. The system shall call to the selected help place.
32. The system shall receive the help place object that user selected from offline map.
33. The system shall retrieve information of help place object.
34. The system provides information UI to show the help information, which are name, address, district, province, zip code, and phone number.
35. The system shall provide the call UI.
36. The system shall call to the selected help place.
37. The system shall provide search button UI.
38. The system shall provide text field UI.
39. The system shall receive all help places from server.
40. The system shall search help places by keyword from user inputting.
41. The system shall matching keyword with help places.
42. The system shall display help places which matching with keyword.
43. The system shall provide map with help place that user selection from searching by keyword.
44. The system shall provide category button.
45. The system shall receive the current location of user and category’s id.
46. The system shall send the current location of user and category’s id to the server.
47. The system shall receive help place object from the server.
48. The system shall change color of marker to show the position of nearest help place by searching.
49. The system shall display the nearest help place of each category from user selection.
50. The system shall provide menu setting UI.
51. The system shall provide number for setting scope with radio button UI.
52. The system shall define a default value of scope.
53. The system shall check distance between original coordinates and new coordinates position automatically.
54. The system shall send latitude and longitude of user to the server.
55. The system shall receive help places from server.
56. The system shall delete the latest data of help places from the database.
57. The system shall add new data of help places into database.
58. The system shall retrieve information file from XML file of Google direction API**.**
59. The system shall provide Map UI for showing route distance.
60. The system shows polyline of distance from current location to destination
61. The system shall provide Map UI for showing route distance.
62. The system shall provide UI for viewing the details.
63. The system shall retrieve information file from XML file of Google direction API**.**
64. The system provides UI to show details of distances, which are distance of each point on the map, time to use for driving to each point, average time and the average point**.**
65. The system shall receive the help place object that user selected.
66. The system shall retrieve information of help place object.
67. The system provides information UI to show the help information and rating score of help place
68. The system shall receive the help place object that user selected.
69. The system shall retrieve information of help place object.
70. The system provides information UI to show the help information and rating score of help place
71. The system provides UI with the rating bar.
72. The system shall send help place’s id and score that user rate to the server.
73. The system shall retrieve the new average score of help place’s that user rate from the server.
74. The system shall refresh the average score on the information page.

* Server Part

1. The system provides the UI, which receive name, address, district, province, zip code, phone number, category, latitude and longitude.
2. The system provides map UI.
3. The system provides searched province UI.
4. The system shall receive latitude and longitude from map.
5. The system shall check the name length. The name must be 1- 50 characters.
6. The system shall check the address length. The address must be 0- 50 characters.
7. The system shall check the district length. The district must be 0- 50 characters.
8. The system shall check the zip code length. The zip code must be 0-5 characters.
9. The system shall check the phone number format. The phone number should be 9-10 digits.
10. The system shall check the latitude format. The latitude should not be null.
11. The system shall check the longitude format. The longitude should not be null.
12. The system shall check the province format. The province should not be null.
13. The system shall check the category format. The category should not be null.
14. The system shall display the error message “Name must between 1 to 50 characters”
15. The system shall display the error message “The address length should be 0-50 characters”
16. The system shall display the error message “The district length should be 0-50 characters”
17. The system shall display the error message “The zip code length should be 0-5 characters”
18. The system shall display the error message “Please choose province”
19. The system shall display the error message “The phone number length should be 9-10 characters”
20. The system shall display the error message “Please put latitude”
21. The system shall display the error message “Please put longitude”
22. The system shall display the error message “Please choose province”
23. The system shall add a new help place into the database.
24. The system shall provide the successful adding help information UI with message “The help place has been added to database.”
25. The system shall retrieve all help places from system database.
26. The system shall provide UI to show all lists of help places.
27. The system shall provide edit UI for all lists of help places.
28. The system retrieves information of the selected help place from database.
29. The system shall show information of the selected help place which include name, address, district, province, zip code, phone number, category, latitude and longitude.
30. The system provides the UI, which receive name, address, district, province, zip code, phone number, category, latitude and longitude.
31. The system provides map UI.
32. The system provides searched province UI.
33. The system shall receive latitude and longitude from map.
34. The system shall check the name length. The name must be 1- 50 characters.
35. The system shall check the address length. The address must be 0- 50 characters.
36. The system shall check the district length. The district must be 0- 50 characters.
37. The system shall check the zip code length. The zip code must be 0-5 characters.
38. The system shall check the phone number format. The phone number should be 9-10 digits.
39. The system shall check the latitude format. The latitude should not be null.
40. The system shall check the longitude format. The longitude should not be null.
41. The system shall check the province format. The province should not be null.
42. The system shall check the category format. The category should not be null.
43. The system shall display the error message “Name must between 1 to 50 characters”
44. The system shall display the error message “The address length should be 0-50 characters”
45. The system shall display the error message “The district length should be 0-50 characters”
46. The system shall display the error message “The zip code length should be 0-5 characters”
47. The system shall display the error message “Please choose province”
48. The system shall display the error message “The phone number length should be 9-10 characters”
49. The system shall display the error message “Please put latitude”
50. The system shall display the error message “Please put longitude”
51. The system shall display the error message “Please choose province”
52. The system shall update the help information into the database.
53. The system shall provide the successful editing help information UI with message “The help place has been added to database.”
54. The system shall retrieve all help places from system database.
55. The system shall provide UI to show all lists of help places.
56. The system shall provide remove UI for all lists of help places.
57. The system shall provide UI with message “Are you sure to delete?” to ask permission before remove help place.
58. The system shall delete the help place out of the database.
59. The system shall provide the successful removing help place UI with message “The help place has been removed!”
60. The system shall retrieve all help places from system database.
61. The system shall provide UI to show all lists of help places.
62. The system retrieves information of the selected help place from database.
63. The system shall show information of the selected help place which include name, address, district, province, zip code, phone number, category, latitude, longitude, and average rate.
64. The system provides categories UI, which are police station, highway police station, hospital, and garage.
65. The system shall retrieve help places from database by the selected category.
66. The system shall show lists of help places by selected category.
67. The system provides UI to show lists of Thailand’s provinces.
68. The system shall retrieve the help place from database by the selected province.
69. The system shall show lists of help places by the selected province.
70. The system provides UI to show lists of categories and Thailand’s provinces.
71. The system shall retrieve help places from database by the selected category and province.
72. The system shall show lists of help places by the selected category and selected province
73. The system provides the login page with UI to receive username and password.
74. The system checks username and password in database.
75. The system shows home page.
76. The system provides log in page with UI to receive username and password.
77. The system provides change password UI.
78. The system updates a new password into the system database.

**SRSs-25:** The system shall retrieve all help places from system database.

**SRSs-50:** The system shall show list of all help places in form of JSON.

**SRSs-51:** The system shall find the nearest help place by the selected category.

**SRSs-52:** The system shall show the nearest help place in form of JSON.

**SRSs-53:** The system shall retrieve list of help places where locate in the setting scope from the database.

**SRSs-54:** The system shall show list of help places where locate in setting scope in form of JSON.

**SRSs-55:** The system calculates a new average rating score from the new rating score that the mobile application requested.

**SRSs-56:** The system updates a new average rating score into the system database.

**SRSs-57:** The system shows help place with new average rating score in form of JSON.

## 4.3 Use Case

* Mobile Part

**UCm-01:** View the online map

**UCm-02:** View the offline map

**UCm-03:** View help places in online map

**UCm-04:** View help places in offline map

**UCm-05:** View information of each help place in online map

**UCm-06:** View information of each help place in offline map

**UCm-07:** Make emergency call to each help place in online map

**UCm-08:** Make emergency call to each help place in offline map

**UCm-09:** Search help place’s name by keyword in online map

**UCm-10:** Find the nearest help place

**UCm-11:** Set the scope for downloading data

**UCm-12:** Collect help place information automatically

**UCm-13:** View the route of distance between the current locations of user to the destination.

**UCm-14:** View details of each point on routing the direction.

**UCm-15:** View average rating score of each help place in online map.

**UCm-16:** View average rating score of each help place in offline map.

**UCm-17:** Rate the help place in online map.

**UCm-01:** View the online map

**UCm-02:** View the offline map

**UCm-03:** View help places in online map

**UCm-04:** View help places in offline map

**UCm-05:** View information of each help place in online map

**UCm-06:** View information of each help place in offline map

**UCm-07:** Make emergency call to each help place in online map

**UCm-08:** Make emergency call to each help place in offline map

**UCm-09:** Search help place’s name by keyword in online map

**UCm-10:** Find the nearest help place

**UCm-11:** Set the scope for downloading data

**UCm-12:** Collect help place information automatically

**UCm-13:** View the route of distance between the current locations of user to the destination.

**UCm-14:** View details of each point on routing the direction.

**UCm-15:** View average rating score of each help place in online map.

**UCm-16:** View average rating score of each help place in offline map.

**UCm-17:** Rate the help place in online map.

* Server Part

**UCs-01:** Add help place

**UCs-02:** Edit help place

**UCs-03:** Remove help place

**UCs-04:** View help place’s information

**UCs-05:** Browse the help place by category

**UCs-06:** Browse the help place by province

**UCs-07:** Browse the help place by province and category

**UCs-08:** Login to the system

**UCs-09:** Logout of the system

**UCs-10:** Update account’s password

**UCs-11:** Get list of all help places

**UCs-12:** Get nearest help place

**UCs-13:** Get list of all help places in setting scope

**UCs-14:** Retrieve new average rating score

## 4.4 Sequence Diagram

* Mobile Part

1. View online map diagram
2. View offline map diagram
3. View the help places in online map diagram
4. View the help places in offline map diagram
5. View information of each help place in online map diagram
6. View information of each help place in offline map diagram
7. Make emergency call to each help place in online map diagram
8. Make emergency calls to each help place in offline map diagram
9. Search help place’s name by keyword in online map diagram
10. Find the nearest help place diagram
11. Set the scope for downloading data diagram
12. Collect help place information automatically diagram
13. View the route of distance between the current locations of user to the destination
14. View details of each point on routing the direction
15. View average rating score of each help place in online map
16. View average rating score of each help place in offline map

* Server Part

1. Add help place sequence diagram
2. Edit help place sequence diagram
3. Remove help place sequence diagram
4. View information sequence diagram
5. Search by category sequence diagram
6. Search by province sequence diagram
7. Search by category and province sequence diagram
8. Login to the system
9. Logout of the system
10. Update account’s password
11. Get all help places diagram
12. Get nearest help place diagram
13. Get all help places in setting scope diagram
14. Retrieve new average rating score

## 4.5 User Interface Design

**UI-01:** Start Page

**UI-02:** Show Online map

**UI-03:** Connect Offline map

**UI-04:** Show Offline map

**UI-05:** MapsWithMe application installed

**UI-06:** MapsWithMe application

**UI-07:** Show Information Page

**UI-08:** Search Help place Page

**UI-09:** Setting scope

**UI-10:** Routing direction

**UI-11:** Give rating score

**UI-12:** Home Page

**UI-13:** Remove Confirm Dialog

**UI-14:** SuccessfullyRemove Dialog

**UI-15:** Update Information Page

**UI-16**: Successfully Add Dialog

**UI-17**: View Information Page

**UI-18:** Login Page

**UI-19:** Change PasswordPage

## 4.6 Method Description

* Mobile Part

1. getJSONObjShowHelpPlacesInOnlineMap
2. getJSONArrayShowHelpPlacesInOnlineMap
3. getJSONObjToSaveInDB
4. getJSONArrayToSaveInDB
5. getJSONObjShowHelpPlacesInOnlineMap
6. getJSONArrayShowHelpPlacesInOnlineMap
7. getJSONObjToSaveInDB
8. getJSONArrayToSaveInDB
9. getHelpPlaceService
10. OnlineMapController
11. createMarker
12. OfflineMapController
13. getAllHelpplaceSaved
14. DatabaseConnection
15. insertHelpPlaces
16. deleteAllHelpPlaces
17. getPendingIntent
18. handleIntent
19. getJSONFromUrl
20. onCallStateChanged
21. selectDatabase
22. copyDatabase
23. getAllHelpPlaces
24. findJSONObjNearestHelpPlace
25. findJSONArrayNearestHelpPlace
26. getHelpPlacesList
27. getNearestHelpPlace
28. setHelpPlaceDAO
29. deleteAllHelpPlaces
30. getAllHelpPlacesOnDevice
31. insertHelpPlace
32. findJSONObjNearestHelpPlace
33. findJSONArrayNearestHelpPlace
34. getHelpPlacesList
35. getNearestHelpPlace
36. setHelpPlaceDAO
37. deleteAllHelpPlaces
38. getAllHelpPlacesOnDevice
39. insertHelpPlace
40. setURL
41. findNearestHelpPlace
42. setHelpPlacesForSearching
43. getShowSearchPoint
44. setScope
45. calculateDistance
46. getDistanceBetweenPoint
47. saveDistance
48. findNearestHelpplace
49. setDB
50. getDB
51. getInstance
52. showDetailsOfPosition
53. getNodeIndex

* Server Part

1. HelpPlaces
2. getHelpPlaces
3. updateHelpPlace
4. deleteHelpPlace
5. findById
6. getHelpPlacesByCategory
7. getHelpPlacesByProvince
8. getHelpPlacesByCategoryAndProvince
9. getHelpPlaces
10. updateHelpPlace
11. deleteHelpPlace
12. findById
13. getHelpPlacesByCategory
14. getHelpPlacesByProvince
15. getHelpPlacesByCategoryAndProvince
16. getHelpPlaces
17. findById
18. updateHelpPlace
19. deleteHelpPlace
20. getHelpPlacesByCategory
21. getHelpPlacesByProvince
22. getHelpPlacesByCategoryAndProvince
23. setHelpPlaceDAO
24. getHelpPlaces
25. updateHelpPlace
26. deleteHelpPlace
27. findById
28. getHelpPlacesByCategory
29. getHelpPlacesByProvince
30. getHelpPlacesByCategoryAndProvince
31. setHelpPlaceDAO
32. Category
33. getCategories
34. getCategoryById
35. getCategories
36. getCategoryById
37. getCategories
38. getCategoryById
39. setCategoryDAO
40. getCategories
41. getCategoryById
42. setCategoryDAO
43. Province
44. getProvinces
45. getProvinceById
46. getProvinces
47. getProvinceById
48. getProvinces
49. getProvinceById
50. setProvinceDAO
51. getProvinces
52. getProvinceById
53. setProvinceDAO
54. listHelpPlaces
55. getHelpPlaceByCategoryAndProvince
56. getHelpPlaceById
57. addHelpPlace
58. updateHelpPlace
59. removeHelpPlace
60. addValidHelpPlace
61. getNearestHelpPlace
62. getLatitudeDistance
63. getLongitudeDistance
64. findDistance
65. getHelpPlacesInScope
66. getNearestHelpPlace
67. getLatitudeDistance
68. getLongitudeDistance
69. findDistance
70. getHelpPlacesInScope
71. getNearestHelpPlaceJson
72. listHelpPlacesJson
73. listHelpPlacesInScopeJson
74. calculateAverageRate
75. calculateAverageRate
76. helpPlaceNewAverageRate
77. getUserSession
78. updateAdmin
79. addValidAdmin
80. getLoginPage
81. findByName
82. saveUser
83. findByName
84. saveUser
85. findByName
86. saveUser
87. findByName
88. saveUser
89. loadByUsername
90. getAuthorities

## 

## 4.7 Class Diagram

* Mobile Part

1. HelpPlace
2. HelpPlaceService
3. HelpPlaceServiceImpl
4. HelpPlaceSingleton
5. OnlineMapController
6. OffllineMapController
7. InformationView
8. JSONParser
9. PhoneCallListener
10. DatabaseConnection
11. CheckingDistance
12. SearchingHelpPlaces
13. DatabaseConnectionSingleton
14. DirectionRoute

* Server Part

1. HelpPlace
2. HelpPlaceDAO
3. HelpPlaceDAOImpl
4. HelpPlaceService
5. HelpPlaceServiceImpl
6. Category
7. CategoryDAO
8. CategoryDAOImpl
9. CategoryService
10. CategoryServiceImpl
11. Province
12. ProvinceDAO
13. ProvinceDAOImpl
14. ProvinceService
15. ProvinceServiceImpl
16. HelpPlaceController
17. AdminController
18. LoginLogoutController
19. Admin
20. AdminDAO
21. AdminDAOImpl
22. AdminService
23. AdminServiceImpl
24. UserDetailsServiceImpl

## 4.8 Unit Test Case

* Mobile Part

**UTCm-01:** testGetHelpPlaceByJsonOBJ (JSONObject jsonOBJ):JSONArray

**UTCm-02:** testGetHelpPlacesOnDevice():HelpPlace []

**UTCm-03:** insertHelpPlace(ArrayList <HelpPlace> helpPlace): int

**UTCm-04:** deleteAllHelpPlaces (): Boolean

* Server Part

**UTCs-01:** getHelpPlaces():List<HelpPlace>

**UTCs-02:** findById(Integer id):HelpPlace

**UTCs-03:** updateHelpPlace(HelpPlace helpPlace):HelpPlace

**UTCs-04:** deleteHelpPlace(HelpPlace helpPlace):boolean

**UTCs-05:** getHelpPlacesByCategory(Integer categoryId)**:**List<HelpPlace>

**UTCs-06:** getHelpPlacesByProvince(Integer provinceId)**:**List<HelpPlace>

**UTCs-07:** getHelpPlacesByCategoryAndProvince(Integer categoryId,Integer provinceId)**:**List<HelpPlace>

**UTCs-08:** getCategories():List<Category>

**UTCs-09:** getCategoryById (Integer id):Category

**UTCs-10:** getProvinces():List<Province>

**UTCs-11:** getProvinceById (Integer id): Province

**UTCs-12:** getNearestHelpPlace(double userLatitude,double userLongitude,Integer categoryId) :HelpPlace

**UTCs-13:** getHelpPlacesInScope(double userLatitude, double userLongitude, double scope)**:**List<HelpPlace>

**UTCs-14:** getLatitudeDistance(double latitude)**:**double

**UTCs-15:** getLongitudeDistance(double longitude)**:**double

**UTCs-16:** findDistance(double la1, double long1, double la2, double long2)**:**double

**UTCs-17:** calculateAverageRate(Integer id, double score):HelpPlace

**UTCs-18:** findByName(String name):Admin

**UTCs-19:** saveUser(Admin admin):Admin