**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 | Release | 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 |

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

Table of Contents

[Chapter One | Introduction 4](#_Toc394406355)

[1.1 Purpose 4](#_Toc394406356)

[1.2 Project Scope 4](#_Toc394406357)

[Chapter Two | Traceability Matrix Table 6](#_Toc394406358)

[2.1 User Requirement Specification and System Requirement Specification 6](#_Toc394406359)

[2.2 User Requirement Specification and Use Case 7](#_Toc394406361)

[2.3 User Requirement Specification and Sequence Diagram 7](#_Toc394406362)

[2.4 User Requirement Specification and User Interface Design 8](#_Toc394406364)

[2.5 User Requirement Specification and Method Description Server 8](#_Toc394406365)

[2.6 User Requirement Specification and Method Description Mobile 9](#_Toc394406366)

[2.7 Class Diagram Server and Method Description Server 9](#_Toc394406367)

[2.8 Class Diagram Mobile and Method Description Mobile 10](#_Toc394406368)

[2.9 Unit Test Case and Method Description Server 11](#_Toc394406369)

[2.10 Unit Test Case and Method Description Mobile 11](#_Toc394406371)

[Chapter Three| Traceability Record Table 12](#_Toc394406372)

[Chapter Four | Appendix 13](#_Toc394406373)

[4.1 User Requirement Specification 13](#_Toc394406374)

[4.2 System Requirement Specification 13](#_Toc394406375)

[4.3 Use Case 14](#_Toc394406376)

[4.4 Sequence Diagram 14](#_Toc394406377)

[4.5 User Interface Design 14](#_Toc394406378)

[4.6 Method Description 14](#_Toc394406379)

[4.7 Class Diagram 17](#_Toc394406380)

[4.8 Unit Test Case 18](#_Toc394406381)

# 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 |
| #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 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



## 2.2 User Requirement Specification and Use Case



## 2.3 User Requirement Specification and Sequence Diagram

## 

## 2.4 User Requirement Specification and User Interface Design



## 2.5 User Requirement Specification and Method Description Server



## 2.6 User Requirement Specification and Method Description Mobile



## 

## 2.7 Class Diagram Server and Method Description Server



## 2.8 Class Diagram Mobile and Method Description Mobile



## 2.9 Unit Test Case and Method Description Server

## 

## 2.10 Unit Test Case and Method Description Mobile



# Chapter Three| Traceability Record Table



# Chapter Four | Appendix

## 4.1 User Requirement Specification

**URS-16:** The mobile application can get list of all help places in the database.

**URS-17:** The mobile application can get the nearest help place by the selected category.

**URS-18:** The mobile application can get list of help places where locate in the setting scope.

**URS-19:** The user can search help place’s name by keyword in online map.

**URS-20:** The user can find the nearest help place by selection the category in online map.

**URS-21:** The user can set the scope for downloading data.

**URS-22:** The mobile application can collect help place information automatically.

.

## 4.2 System Requirement Specification

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

**SRS-64:** The system shall show list of all help places in form of JSON.

**SRS-65:** The system shall find the nearest help place by the selected category.

**SRS-66:** The system shall show the nearest help place in form of JSON.

**SRS-67:** The system shall retrieve list of help places where locate in the setting scope from the database.

**SRS-68:** The system shall show list of help places where locate in setting scope in form of JSON.

**SRS-69:** The system shall provide search button UI.

**SRS-70:** The system shall provide text field UI.

**SRS-71:** The system shall receive all help places from server.

**SRS-72:** The system shall search help places by keyword from user inputting.

**SRS-73:** The system shall matching keyword with help places.

**SRS-74:** The system shall display help places which matching with keyword.

**SRS-75:** The system shall provide map with help place that user selection from searching by keyword.

**SRS-76:** The system shall provide category button.

**SRS-77:** The system shall receive the current location of user and category’s id.

**SRS-78:** The system shall send the current location of user and category’s id to the server.

**SRS-79:** The system shall receive help place object from the server.

**SRS-80:** The system shall change color of marker to show the position of nearest help place by searching.

**SRS-81:** The system shall display the nearest help place of each category from user selection.

**SRS-82:** The system shall provide menu setting UI.

**SRS-83:** The system shall provide number for setting scope with radio button UI.

**SRS-84:** The system shall define a default value of scope.

**SRS-85:** The system shall check distance between original coordinates and new coordinates position automatically.

**SRS-86:** The system shall send latitude and longitude of user to the server.

**SRS-87:** The system shall receive help places from server.

**SRS-88:** The system shall delete the latest data of help places from the database.

**SRS-89:** The system shall add new data of help places into database.

## 4.3 Use Case

**UC-16:** Get list of all help places

**UC-17:** Get the nearest help place

**UC-18:** Get list of help places where locate in the setting scope

**UC-19:** Search help place’s name by keyword in online map

**UC-20:** Find the nearest help place

**UC-21:** Set the scope for downloading data

**UC-22:** collect help place information automatically

## 4.4 Sequence Diagram

**SD -16:** Get nearest help place diagram

**SD -17:** Get all help places diagram

**SD -18:** Get all help places in setting scope diagram

**SD -19:** Search help place’s name by keyword in online map diagram

**SD -20:** Find the nearest help place diagram

**SD -21:** Set the scope for downloading data diagram

**SD -22:** Collect help place information automatically diagram

## 4.5 User Interface Design

**UI-16:** Home page

**UI-17:** Remove confirm dialog

**UI-18:** Successfully remove dialog

**UI-19:** Update information page

**UI-20:** Successfully add dialog

**UI-21:** View information page

**UI-22:** Start page

## 4.6 Method Description

**4.6.1 Method Description Server**

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

**4.6.2 Method Description Mobile**

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. getHelpPlaces
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

## 

## 4.7 Class Diagram

**4.7.1 Class Diagram Server**

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

**4.7.2 Class Diagram Mobile**

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

## 

## 4.8 Unit Test Case

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

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

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

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

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

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

**UTC-07:** getHelpPlacesByCategoryAndProvince(Integer categoryId,Integer

provinceId)**:**List<HelpPlace>

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

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

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

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

**UTC-12:** getNearestHelpPlace (double userLatitude,double userLongitude,Integer

categoryId) :HelpPlace

**UTC-13:** getHelpPlacesInScope(double userLatitude, double userLongitude, double

scope)**:**List<HelpPlace>

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

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

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

**UTC-17:** insertHelpPlace(ArrayList <HelpPlace> helpPlace): int

**UTC-18:** deleteAllHelpPlaces (): boolean

**UTC-19:** getJSONObjShowHelpPlacesInOnlineMap(String url): JSONObject

**UTC-20:** getJSONArrayShowHelpPlacesInOnlineMap(JSONObjetc jsonOBJ): JSONArray

**UTC-21:** getJSONObjToSaveInDB(String url): JSONObject