**Software Design Document**

**Team Collaboration System for Mobility Water Monitoring**

**By**

**Mr. Peerapong Chompootepa 542115044**

**Mr. Worrasete Tansurat 542115056**

**Department of Software Engineering**

**College of Arts, Media and Technology**

**Chiang Mai University**

**Project Advisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Ms.Siraprapa Wattanakul**

**Document History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **12** | **Detail** | **Status** | **Date** | **View able** | **Reviewer& Responsible** |
| **TCS-SDD-V0.1.docx** | -Introduction  System Architecture | Draft |  | SW | PC, WT |
| **TCS-SDD-V0.2.docx** | Detailed Design | Draft |  | SW | PC, WT |
| **TCS-SDD-V0.3.docx** | -Update Detailed Design  -Data Architecture | Draft |  | SW | PC, WT |
| **TCS-SDD-V0.4.docx** | -Update Detailed Design  -User Interface design  - Data Architecture | Draft |  | SW | PC, WT |
| **TCS-SDD-V1.0.docx** | -User Interface design  -Update Data Architecture | Release |  | SW | PC, WT |
| **TCS-SDD-V1.1.docx** | -Update all document | Draft |  | SW | PC, WT |
| **TCS-SDD-V2.0.docx** | -Update all document | Release |  | SW | PC, WT |
| **TCS-SDD-V3.0.docx** | -Update all document | Release |  | SW | PC, WT |

**\*SW = Ms.Siraprapa Wattanakul**

**\* PC = Mr.Peerapong Chompootepa**

**\* WT = Mr.Worrasete Tansurat**

Table of Contents

[**Chapter 4-1 | Introduction** 4](#_Toc422429068)

[1.1 Objective 4](#_Toc422429069)

[1.2 User Characteristics 4](#_Toc422429070)

[1.3 Acronyms and Definitions 5](#_Toc422429071)

[**Chapter 4-2 | Overall Description** 6](#_Toc422429072)

[2.1 Overall Description 6](#_Toc422429073)

[2.1.1 Product Perspective 6](#_Toc422429074)

[2.1.2 Product Features 6](#_Toc422429075)

[**Chapter 4-3 | System Architecture** 9](#_Toc422429076)

[**Chapter 4-4 | Detailed Design** 14](#_Toc422429078)

[4.1 Class Diagram 14](#_Toc422429079)

[4.2 Class Description 16](#_Toc422429080)

[4.3 Sequence Diagram 47](#_Toc422429081)

[**Chapter 4-5 | Data Architecture** 68](#_Toc422429082)

[5.1 Database Design 68](#_Toc422429083)

[**Chapter 4-6 | User Interface design** 69](#_Toc422429084)

# Chapter 4-1 | Introduction

## 1.1 Objective

The purpose of the Software Design Document (SDD) is to describe the functional and design diagram of the Team collaboration system for mobility water monitoring. Software Design Document (SDD) of Team collaboration system for mobility water monitoring is the document that describes each function and progress of Team collaboration system for mobility water monitoring by diagram to more explain. The diagram are consisting of class diagram, sequence diagram, database design.

## 1.2 User Characteristics

The system divided users into 3 groups. Information and characteristics of each group are listed below.

**Administrator**

Administrator can access to the system on web application for managing the system (profile, project and team member). When the collector sends register request on mobile application, administrator can see a list of new collector requests and also can select to approve or decline the account. Administrator is able to create or modify the new project with selects team and defines team leader into the project. Before to create the project, administrator can create a team by select members existing in the system. And also the administrator can view relevant information such as team information and project information.

**Team leader**

Team leader can access to the system both of web application and mobile application. Team leader can see the information of each project and able to manage team that he is the team leader includes editing name, adding team member and removing team member. On the part of the project, team leader can see the information of each project on web application and able to assigns work by marking location on Google Maps to the collector. Each project will show the status of the work for the members to know the progress of the work.

**Collector**

Collector needs to be registered an account on mobile application and wait for administrator approves an account request before login to the system. Collector can view work assignment assigned by team leader. Collector can collect samples of the water sources that team leader assign location by Google Maps. Collector must update the status of the work and send test result to the web service.

## **1.3 Acronyms and** Definitions

Acronyms

TCS = Team collaboration system for mobility water monitoring

PMP = Project Management Plan

SDD = Software Design Document

SRS = Software Requirment Specification

PC = Peerapong Chompootepa

WT = Worrasete Tansurat

URS = User Requirement Specification

SRS = Software Requirement Specification

UC = Use Case

AD = Activity Diagram

Definitions

**Feature :** Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a product in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90]

**IEEE:**  Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90]

**Requirement:** A condition or capability needed by a user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. A documented representation of a condition or capability as in definition. [IEEE90]

**Specification:** Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g., what is to be done, how it will be done)must not be mixed. [IEEE90]

**UML:** Unified Modeling Languages. Standardized notation for modeling design descriptions, architectures or scenarios. Not depending on a specific method. Issued and maintained by the object Management Group(OMG). [IEEE90]

# Chapter 4-2 | Overall Description

## 2.1 Overall Description

### 2.1.1 Product Perspective

Team collaboration system for mobility water monitoring application is developed for the chemical analysis of the water quality which makes it easy for collectors working as a team. The system separates in two parts. The first part is web application, all user can use web application to view the test result. Administrator can create the project and manage the member in the system and also select members to create the new team. The team leader has a role to manage the involved project which is assigned by administrator. The application can see the team members and their responsibility in each of projects. For example, team leader can assign the work location to each collector by marking the pin on the Google Maps and also can review the test result sent from the collector. The second part is mobile application, a part of team leader and collector. Collector can use the mobile application to collect the data, then the application will send the data to calculation component containing the calculating algorithm (produced by I-ANALY-S-T). The system returns the test result from chemical analysis. The status will update automatically when user send the test result. After that, team leader can review the test result by marking the status to finish or recollect on both mobile application and web application.

### 2.1.2 Product Features

The TCS is developed in both mobile application and web application and also extended functions of the existing mobile software to have further completion for team collaboration. There are 6 main features of TCS are shown below:

Feature 1: Authentication and member management system

Feature 2: Project management

Feature 3: Map location management

Feature 4: Water parameter calculation

Feature 5: Parameter result tracing

Feature 6: Messaging system

All of features will create a step-by-step suit by priority in 3 progress:

**Progress Report I consist of**

**Feature 1: Authentication and member management system**

* Administrator, Team leader, and Collector can log in to the system on web application.
* Administrator, Team leader, and Collector can logout from the system on web application.
* Team leader and Collector can log in to the system on mobile application.
* Team leader and Collector can logout from the system on mobile application.
* New collector can send register request to the administrator on mobile application requesting to be a member.
* Administrator can view list of new collector requests sorted by date on web application.
* Administrator can search a new collector request by using new collector name on web application.
* Administrator can select to approve or decline new member on web application.
* Administrator can create a team which contains a team name, team leader and team members on web application.
* Administrator and team leader can view list of team sorted by team name on web application.
* Administrator and Team leader can search the team by using team name on a web application.
* Administrator and Team leader can modify the selected team information on web application.
* Administrator and Team leader can view the selected team information includes a team name, team leader name, and list of members on web application.
* Administrator can remove a team out of the system on web application.
* Administrator, Team leader, and Collector can edit profile information which includes a name, password and telephone number on web application.

**Feature 2: Project management**

* Administrator can create the project includes a project name, project description, and team on web application.
* Administrator can remove the project out of the system on web application.
* Administrator can view list of all project sorted by created date of the project on a web application
* Team leader and Collector can view list of the involved projects sorted by created date of the project on a web application and mobile application.
* Administrator, Team leader and Collector can search the project by using project name on web application.
* Administrator can modify the project information includes editing project name, editing project description, and changing a team on web application.
* Administrator can view detail of selected projects information which consisting of map with assigned pins and location information on a web application.
* Team leader and Collector can view detail of selected projects information which consisting of map with assigned pins and location information of the selected project on a web application and mobile application.
* Administrator can view list of water parameter of the selected member sorted by date on a web application.
* Team leader and Collector can view list of water parameter of the selected member sorted by date on a web application and mobile application.

**Progress Report II consist of**

**Feature 3: Map location management**

* Team leader can assign work location on Google map to each collectors on web application.
* Team leader and Collector can view the direction to the selected location on Google Maps on mobile application.
* Team leader and Collector can view location information of each assigned pin on selected project by pressing on the pin on mobile application.

**Feature 4: Water parameter calculation**

* Collector can add water parameter to collect the test data on mobile application.
* Collector can delete water parameter on mobile application.
* Collector can choose the image from camera roll or take a new photo to collect RGB value with the test result on mobile application.
* Collector can input data to let the system calculate the test result based on standard color scale on mobile application.

**Progress Report III consist of**

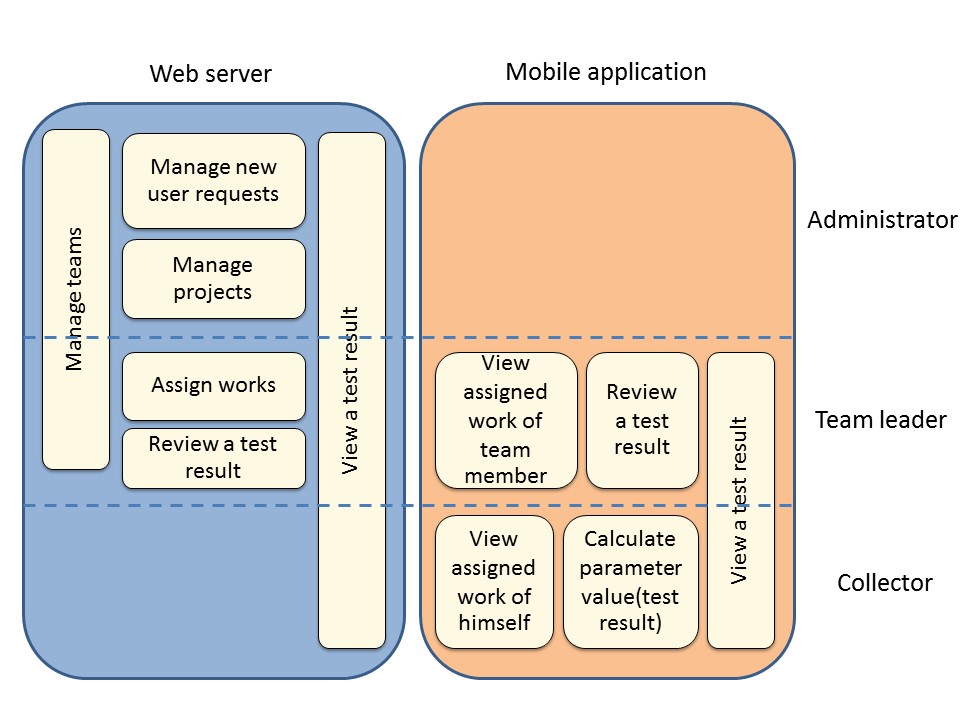
**Feature 5: Parameter result tracing**

* Administrator can view detail of test result of selected water parameter on a web application.
* Team leader and Collector can view detail of test result of selected water parameter on a web application and mobile application.
* Team leader can mark status of the selected water parameter to “Recollect” on a web application and mobile application.
* Team leader can mark status of the selected water parameter to “Finish” on a web application and mobile application.

**Feature 6: Messaging system**

* Team leader and Collector can send the message to each other via group message within the team on a mobile application.

# Chapter 4-3 | System Architecture

****

**Figure1: Functional architecture**

Figure 1 show the overview of Team collaboration system for mobility water monitoring system. There are 2 parts in the system consist of Mobile Application part and Web application part. Mobile Application for Collector is a software for water parameters monitoring. Then, the test result will be recorded to database. Mobile Application for Team Leader provides the result presentation retrieving the data from database in and trace a working position all of the collectors in his team. Finally, Web Application provides to all of users consisting of Administrator, Team Leader and Collector. Administrator can create and manage teams and projects. They also can approve a new registered member. Team leader can assign works on Google Maps to collectors and review the details of the test result of the involved project. Collector can know the assigned work from team leader to do the water quality monitoring, then send the test result to the database via JSON. In every testing, the information will be recorded in the Google Sheet. In addition, team leader and collectors can have a connection by messaging in the mobile application within each team.

**There are 6 features in the system:**

|  |  |  |
| --- | --- | --- |
| Feature 1: Authentication and member management system | | |
| On a Web application and Mobile application | | |
| 🗹Administrator | 🗹Team leader | 🗹Collector |
| This feature is an authentication of all user involving in this system. The application can manage a team or member in the system. | | |
| Feature 2: Project management | | |
| On a Web application | | |
| 🗹Administrator | 🗹Team leader | 🗹Collector |
| This feature is a part of operation involve to the project. Administrator can manage the project, for example create and modify the project on web application. | | |
| Feature 3: Project browsing | | |
| On a Web application and Mobile application | | |

|  |  |  |
| --- | --- | --- |
| 🗹Administrator | 🗹Team leader | 🗹Collector |

|  |
| --- |
| The holistic view of this feature is to view the project list and the others information. For example, the list of parameter test, the responsibility of each collector (place) and the date and time of the project created. |
| Feature 4: Map location management |
| On a Web application and Mobile application |

|  |  |  |
| --- | --- | --- |
| 🗹Administrator | 🗹Team leader | 🗹Collector |

|  |
| --- |
| Google map was used in this project for work location of collector. Google maps can help the team’s work easier, as shows the information related to the project and also shows the route to the collector to the destination. |
| Feature 5: Result presentation |
| On a web application and Mobile application |

|  |  |  |
| --- | --- | --- |
| 🗹Administrator | 🗹Team leader | 🗹Collector |

|  |
| --- |
| The test result obtained from the collector is to collect and calculate on a mobile application. In this feature, team leader and collector can view the detail of test result and also team leader can review test result by mark status of test result on a web application and mobile application. |
| Feature 6: Messaging system |
| On a mobile application |

|  |  |  |
| --- | --- | --- |
| 🗷Administrator | 🗹Team leader | 🗹Collector |

|  |
| --- |
| About the various alerts such as notification of new projects, notification of recollect. Furthermore, user can make a conversation by sending message via a group message within the team on a mobile application. |
| Feature 7: Extension of existing application |
| On a mobile application |

|  |  |  |
| --- | --- | --- |
| 🗷Administrator | 🗷Team leader | 🗹Collector |

|  |
| --- |
| The last feature is a part of collector to collect the data (parameter test) by managing a parameter and using the algorithm to calculate test result on a mobile application. |

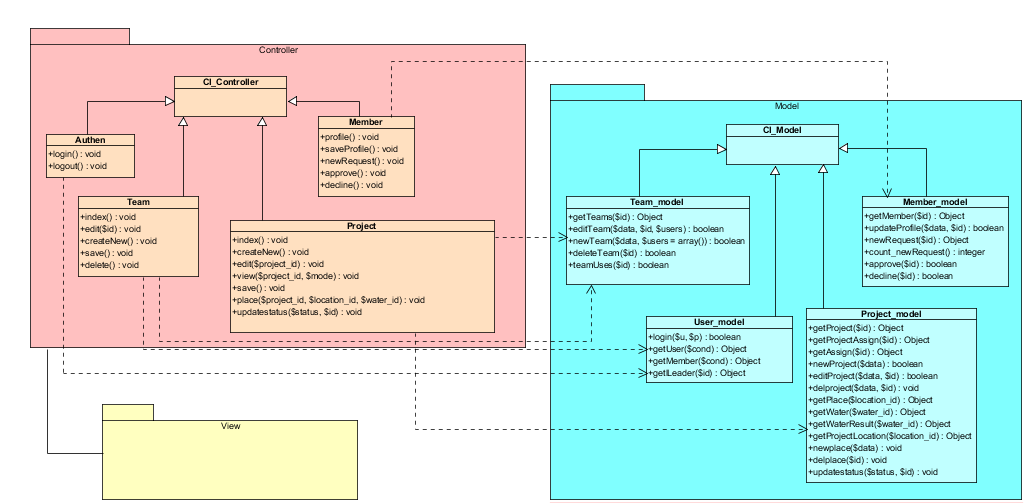
**Each of features is separated as follows:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| **Feature 1: Authentication and member management system** | Authentication on web application | √ | √ | √ |
| Edit profile on web application | √ | √ | √ |
| View list of new collector request on web application | √ |  |  |
| Select to approve or decline new collector request on web application | √ |  |  |
| Create team on web application | √ |  |  |
| Modify team on web application | √ | √ |  |
| View the selected team information on web application | √ | √ |  |
| Send register request on mobile application |  |  | √ |
| Authentication on mobile application |  | √ | √ |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| **Feature 2: Project management** | Create the project on web application | √ |  |  |
| Search the project on web application | √ | √ | √ |
| Modify the project on web application | √ |  |  |
| View the project list on web application | √ |  |  |
| View the involved project list on web application and mobile application |  | √ | √ |
| View a project information on web application | √ |  |  |
| View a project information on web application and mobile application |  | √ | √ |
| View list of parameter test on web application | √ | √ | √ |
| View list of parameter test on web application and mobile application |  | √ | √ |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| **Feature 3: Map location management** | Assign work location on Google Maps on web application |  | √ |  |
| View direction to assigned location on Google Maps on mobile application |  | √ | √ |
| View other place information by press on pin on mobile application |  | √ | √ |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| **Feature 4: Water parameter calculation** | Add test parameter to collect the test data on mobile application |  |  | √ |
| Delete test parameter on mobile application |  |  | √ |
| Choose the image from camera roll or take a new photo to collect RGB value with the test result on mobile application |  |  | √ |
| Input value and RGB to let the system calculate the test result based on standard color scale on mobile application |  |  | √ |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| Feature 5: Parameter result tracing | View detail of test result of selected parameter on a web application. | √ |  |  |
| View detail of test result of selected parameter  on a web application and mobile application |  | √ | √ |
| Mark status to recollect on parameter test on a web application and mobile application. |  | √ |  |
| Mark status to finish on parameter test on a web application and mobile application. |  | √ |  |
| **Feature** | **Function name** | **Administrator** | **Team leader** | **Collector** |
| **Feature 6: Messaging system** | Send the message to each other via group message within the team on a mobile application. |  | √ | √ |

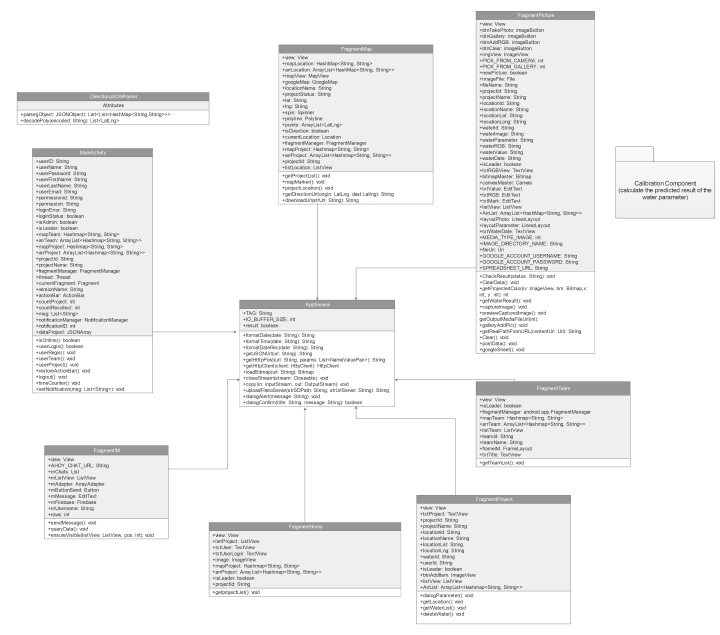
# Chapter 4-4 | Detailed Design

## 4.1 Class Diagram

* **Web application Part**



* **Mobile application Part**

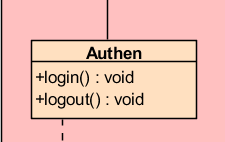


## 4.2 Class Description

* **Web application Part**

**W-CD-01: Authen**

**Description:** Overview of this class is a controller to control the authentication of the users.

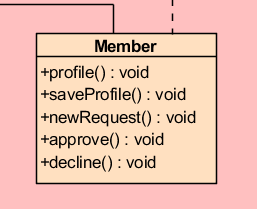
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| **1** | Login() | Method uses to login. | void |
| **2** | Lohout() | Method uses to log out. | void |

**W-CD-02: Member**

**Description:** Overview of this class is controller to control the member in the system.

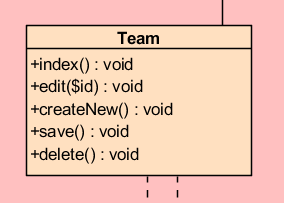
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | profile() | Method uses to get member data from the database. | void |
| 2 | saveProfile() | Method uses to update the profile of administrator or team leader to the database. | void |
| 3 | newRequest) | Method uses to get the list of new collector requests from the database. | void |
| 4 | approve() | Method uses to approve the new collector request to be a member in the database. | void |
| 5 | decline() | Method uses to decline the new collectors request and delete account from the database. | void |

**W-CD-03: Team**

**Description:** Overview of this class is controller to control the team in the system.

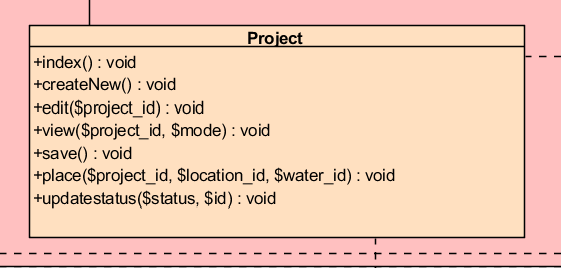
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | index() | Method uses to show the team information page. | void |
| 2 | edit($id) | Method uses to modify team information in the database. | void |
| 3 | createNew() | Method uses to create a new team in the database. | void |
| 4 | Save() | Method uses to send the operation to newTeam() for creating a new team and to editTeam() for editing a team. | void |
| 5 | Delete() | Method uses to remove the team from the database. | void |

**W-CD-04: Project**

**Description:** Overview of this class is controller to control the project in the system.

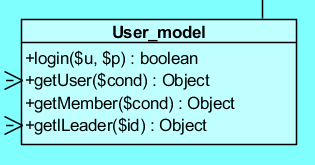
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | index() | Method uses to show the project information page. | void |
| 2 | createNew() | Method uses to create a new project in the database. | void |
| 3 | edit($project\_id) | Method uses to modify project information in the database. | void |
| 4 | view($project\_id,$mode) | Method uses to view the project information in the database. | void |
| 5 | save() | Method uses to send the operation to newProject() for creating a new project and to editProject() for editing a project. | void |
| 6 | place($project\_id, $location\_id, $water\_id) | Method uses to view the water information in that place. | void |
| 7 | updatestatus($status, $id) | Method uses to update the current status of each parameter | void |

**W-CD-05: User\_model**

**Description:** Overview of this class is a model to receive the user information from the database.

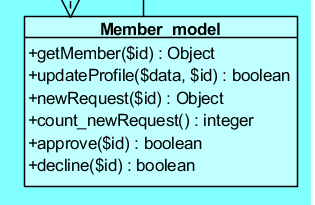
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | login($u, $p) | Method uses to login to system. | boolean |
| 2 | getUser ($cond) | Method uses to get user data from the database. | Object |
| 3 | getMember($cond) | Method uses to get member data from the database. | Object |
| 4 | getLeader($id) | Method uses to get team leader data from the database. | Object |

**W-CD-06: Member\_model**

**Description:** Overview of this class is a model to receive the member information from the database.

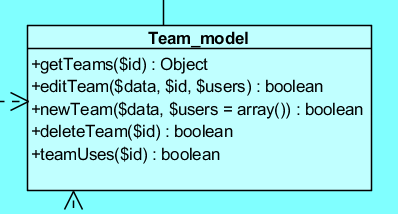
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getMember($id) | Method uses to get member data from the database. | Object |
| 2 | updateProfile ($data, $id) | Method uses to update the profile of administrator or team leader to the database. | boolean |
| 3 | newRequest($id) | Method uses to get the list of new collector requests from the database. | Object |
| 4 | count\_newRequest() | Method uses execute a number of new collector requests in the database. | integer |
| 5 | approve($id) | Method uses to approve the new collector request to be a member in the database. | boolean |
| 6 | decline($id) | Method uses to decline the new collectors request and delete account from the database. | boolean |

**W-CD-07: Team\_model**

**Description:** Overview of this class is a model to receive the team information from the database.

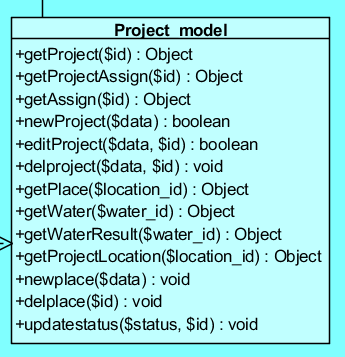
****

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getTeams($id) | Method uses to get team data from the database. | Object |
| 2 | editTeam($data, $id, $users) | Method uses to edit team data in the database. | boolean |
| 3 | newTeam($data, $users) | Method uses to add a new team to the database. | boolean |
| 4 | deleteTeam($id) | Method uses to remove a team from the database. | boolean |
| 5 | teamUsers($id) | Method uses to check team usage before delete a team. | boolean |

**W-CD-08: Project\_model**

**Description:** Overview of this class is a model to receive the project information from the database.

****

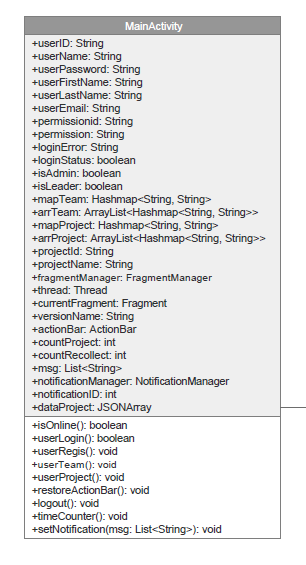
**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getProject($id) | Method uses to get project data from the database. | Object |
| 2 | getProjectAssign ($id) | Method uses to get project assigned data from the database to view the project information. | Object |
| 3 | getAssign($id) | Method uses to get assign location to collector from the database. | Object |
| 4 | newProject($data) | Method uses to add a new project to the database. | boolean |
| 5 | editProject($data, $id) | Method uses to edit project data in the database. | boolean |
| 6 | delProject($data, $id) | Method uses to delete the project in the database. | void |
| 7 | newplace($data) | Method uses to add a new location in the database. | void |
| 8 | delplace($id) | Method uses to get the location in the database. | void |
| 9 | getProjectLocation($location\_id) | Method uses to get the location in the database. | Object |
| 10 | getWater($water\_id) | Method uses to get the water parameter information in the database. | Object |
| 11 | getPlace($location\_id) | Method uses to get the location information and water information in the database. | Object |
| 12 | getWaterResult($water\_id) | Method uses to get the test result of water parameter in the database. | Object |
| 13 | updatestatus($status, $id) | Method uses to get the location in the database. | void |

* **Mobile application Part**

**M-CD-01: MainActivity**

**Description:** Overview of this class is to check the authentication of user.



**Property**

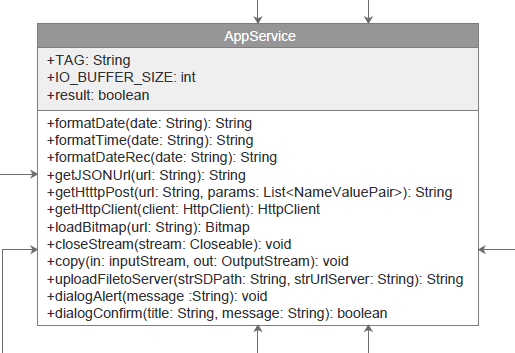
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | userID | Declare a variable  to identify a user id | Type: String |
| 2 | userName | Declare a variable  to identify a username | Type: String |
| 3 | userPassword | Declare a variable  to identify a password | Type: String |
| 4 | userFirstName | Declare a variable  to identify the first name | Type: String |
| 5 | userLastName | Declare a variable  to identify the last name | Type: String |
| 6 | userEmail | Declare a variable  to identify an email of user | Type: String |
| 7 | permissionid | Declare a variable  to identify a permission id | Type: String |
| 8 | permission | Declare a variable  to identify a permission name | Type: String |
| 9 | loginError | Declare a variable  to identify an error message | Type: String |
| 10 | loginStatus | Declare a variable  to identify login status | Type: Boolean |
| 11 | isAdmin | Declare a variable to identify the user is an administrator | Type: Boolean |
| 12 | isLeader | Declare a variable to identify the user is a team leader | Type: Boolean |
| 13 | mapTeam | Declare a variable to identify set of data of the team before insert to the array list, adding column by column till finish the roll | Type: Hashmap<String, String> |
| 14 | arrTeam | Declare a variable to identify set of data of the team before insert to the table list by receiving from Hashmap | Type: ArrayList<Hashmap<String, String>> |
| 15 | mapProject | Declare a variable to identify set of data of the project before insert to the array list, adding column by column till finish the roll | Type: Hashmap<String, String> |
| 16 | arrProject | Declare a variable to identify set of data of the project before insert to the table list by receiving from Hashmap | Type: ArrayList<Hashmap<String, String>> |
| 17 | projectId | Declare a variable to identify a project id | Type: String |
| 18 | projectName | Declare a variable to identify a project name | Type: String |
| 19 | fragmentManager | Declare a variable to identify a service to control every fragment operation | Type: FragmentManager |
| 20 | thread | Declare a variable to identify project for running on background for the service that requests the information from server | Type: Thread |
| 21 | currentFragment | Declare a variable to identify what fragment you are | Type: Fragment |
| 22 | versionName | Declare a variable to identify a version when upload to the store | Type: String |
| 23 | actionBar | Declare a variable to identify the title of the that page | Type: ActionBar |
| 24 | countProject | Declare a variable to identify the amount of project | Type: int |
| 25 | countRecollect | Declare a variable to identify the amount of requested project to test again | Type: int |
| 26 | msg | Declare a variable to identify the text message | Type: List<String> |
| 27 | notificationManager | Declare a variable to identify a controller of notification operation | Type: NotificationManager |
| 28 | NOTIFICATIONID | Declare a variable to identify notification id | Type: int |
| 29 | numMessages | Declare a variable to identify the amount of messages. | Type: int |
| 30 | dataProject | Declare a variable to identify the received data from JSON sent by server. | Type: JSONArray |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | isOnline() | Method uses to check the internet connection | boolean |
| 2 | userLogin() | Method uses to check the user authentication to access the system requested from the user. | boolean |
| 3 | userRegis() | Method uses to register the new member to access the system. | void |
| 4 | userTeam() | Method uses to verify the authentication of user in the team referring arr.Team variable. | void |
| 5 | userProject() | Method uses to verify the authentication of user in the project referring arr.Project variable. | void |
| 6 | restoreActionBar() | Method uses to refresh the name changing on action bar. | void |
| 7 | logout() | Method uses to logout the system. | void |
| 8 | timeCounter() | Method uses to manage a posting delay. | void |
| 9 | setNotification(msg: List<String>) | Method uses to manage a notification message. | void |

**M-CD-02: AppService**

**Description:** The overview of this class is to control the data transfer with a web service via JSON.



**Property**

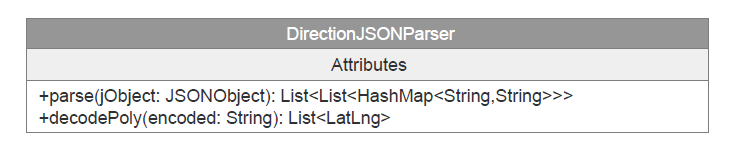
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | TAG | Declare a variable to identify the error message when system cannot load an image | Type: String |
| 2 | IO\_BUFFER\_SIZE | Declare a variable to identify a file reading | Type: int |
| 3 | result | Declare a variable to identify the  data sent from the server | Type: boolean |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | formatDate(date: String) | Method uses to adjust the date format to be Thai format containing date and time. | String |
| 2 | formatTime(date: String) | Method uses to request time format only | String |
| 3 | formatDateRec(date: String) | Method uses to request date format only | String |
| 4 | getJSONUrl(url: String) | Method uses to identify path on a server that you want to request the data. | String |
| 5 | getHtttpPost(url: String, params: List<NameValuePair>) | Method uses to request the data from a web service. | String |
| 6 | getHttpClient(client: HttpClient) | Method uses to connect or access the server. | HttpClient |
| 7 | loadBitmap(url: String) | Method uses to load the image. | Bitmap |
| 8 | closeStream(stream: Closeable) | Method uses to close the the operation after loading Bitmap. | void |
| 9 | copy(in: inputStream, out: OutputStream) | Method uses to create Temp in the system after loading the image. | void |
| 10 | uploadFiletoServer(strSDPath: String, strUrlServer: String) | Method uses to upload images to server through the path. | String |
| 11 | dialogAlert(message :String) | Method uses to display alert messages . | void |
| 12 | dialogConfirm(title: String, message: String) | Method uses to display the confirmation alert. | Boolean |

**M-CD-03: DirectionJSONParser**

**Description:** The overview of this class is to receives a JSONObject and returns a list of lists containing latitude and longitude.

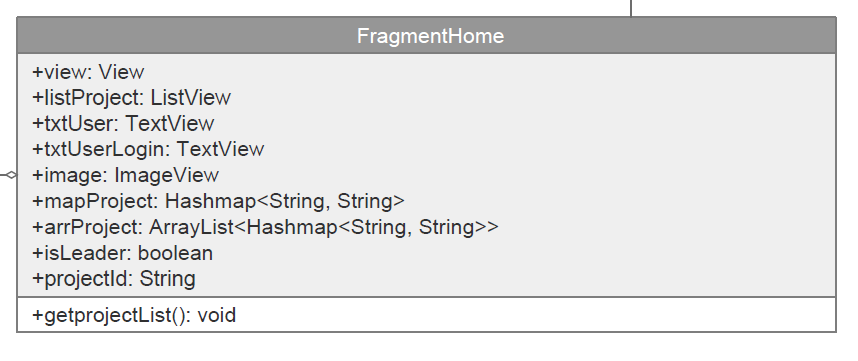


**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | parse(jObject: JSONObject): | Method uses to get a map direction receiving a JSONObject and returns a list of lists containing latitude and longitude. | List<List<HashMap<String,String>>> |
| 2 | decodePoly(encoded: String): | Method uses to decode polyline points from Google maps direction API. | List<LatLng> |

**M-CD-04: FragmentHome**

**Description:** This class is a fragment about the project list that user involved.



**Property**

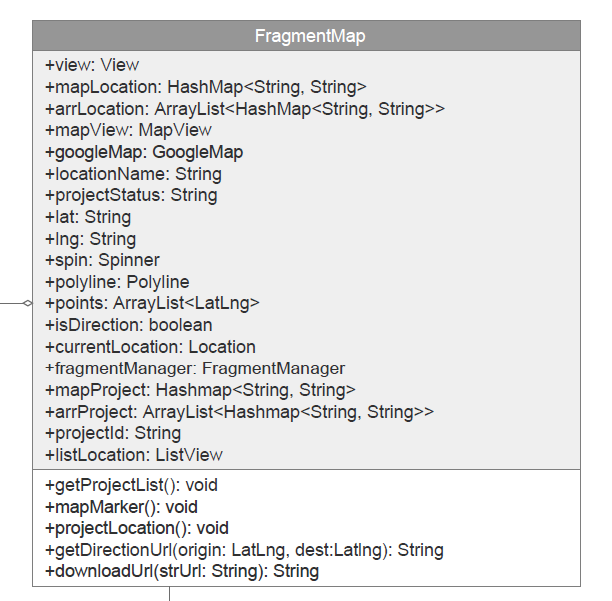
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | listProject | Declare a variable to identify a table to show the list of project | Type: ListView |
| 2 | txtUser | Declare a variable to identify a text field to show the text of user | Type: TextView |
| 3 | txtUserLogin | Declare a variable to identify a text field to show the text of user’s login | Type: TextView |
| 4 | image | Declare a variable to identify an image to view | Type: ImageView |
| 5 | mapProject | Declare a variable to identify set of data of the project before insert to the array list, adding column by column till finish the roll | Type: HashMap<String, String> |
| 6 | arrProject | Declare a variable to identify set of data of the project before insert to the table list by receiving from Hashmap | Type: ArrayList<HashMap<String, String>> |
| 7 | isLeader | Declare a variable to identify the user is a team leader | Type: boolean |
| 8 | projectId | Declare a variable to identify the project id | Type: String |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getProjectList() | Method uses to retrieve the list of project such as ProjectId, ProjectName, ProjectDesc, LeaderId, TeamLeader, etc from database. | void |

**M-CD-05: FragmentMap**

**Description:** The overview of this class is a fragment about the location on Google Map and parsing the Google Places in JSON format.



**Property**

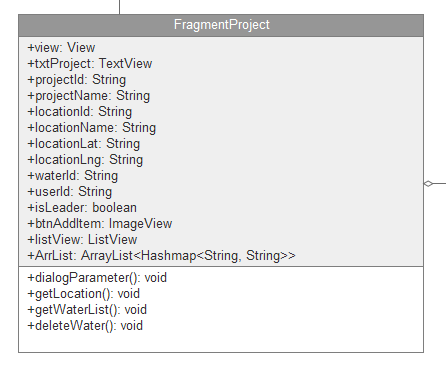
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | view | Declare a variable to identify the  view for every function | Type: View |
| 2 | mapLocation | Declare a variable to identify set of data of the location before insert to the array list, adding column by column till finish the roll | Type: HashMap<String, String> |
| 3 | arrLocation | Declare a variable to identify set of data of the location before insert to the table list by receiving from Hashmap | Type: ArrayList<HashMap<String, String>> |
| 4 | mapView | Declare a variable to identify a map to view | Type: MapView |
| 5 | googleMap | Declare a variable to identify Google Maps | Type: GoogleMap |
| 6 | locationName | Declare a variable to identify a name of that location | Type: String |
| 7 | projectStatus | Declare a variable to identify a status of project consisting of finish and recollect | Type: String |
| 8 | lat | Declare a variable to identify latitude | Type: Double |
| 9 | lng | Declare a variable to identify longitude | Type: Double |
| 10 | spin | Declare a variable to identify a dropdown menu | Type: Spinner |
| 11 | BASECENTER | Declare a variable to identify a destination from latitude and longitude | Type: LatLng |
| 12 | polyline | Declare a variable to identify a direction to create the route | Type: Polyline |
| 13 | points | Declare a variable to identify a set of location in the array | Type: ArrayList<LatLng> |
| 14 | isDirection | Declare a variable to identify the route between the start and final destination | Type: Boolean |
| 15 | currentLocation | Declare a variable to identify the current location on the map | Type: Location |
| 16 | fragmentManager | Declare a variable to identify a service to control every fragment operation | Type: android.app.FragmentManager |
| 17 | mapProject | Declare a variable to identify set of data of the project before insert to the array list, adding column by column till finish the roll | Type: HashMap<String, String> |
| 18 | arrProject | Declare a variable to identify set of data of the project before insert to the table list by receiving from Hashmap | Type: ArrayList<HashMap<String, String>> |
| 19 | projectId | Declare a variable to identify an id of project | Type: String |
| 20 | listLocation | Declare a variable to identify a table to show the list of location | Type: ListView |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getProjectList() | Method uses to retrieve the list of project such as ProjectId, ProjectName, ProjectDesc, LeaderId, TeamLeader, etc from database. | void |
| 2 | mapMarker() | Method uses to create a marker on GoogleMap | void |
| 3 | projectLocation() | Method uses to retrieve the information of user’s project location such as LocationName, Latitude, Longitude, UserId, UserName, etc from database. | void |
| 4 | getDirectionsUrl(origin: LatLng ,dest: LatLng ) | Method uses to receive the direction service from DirectionJSONParser.java | String |
| 5 | downloadUrl(strUrl: String ) | Method uses to download JSON data from url | String |

**M-CD-06: FragmentProject**

**Description:** The overview of this class is about the water parameter list containing in each place.



**Property**

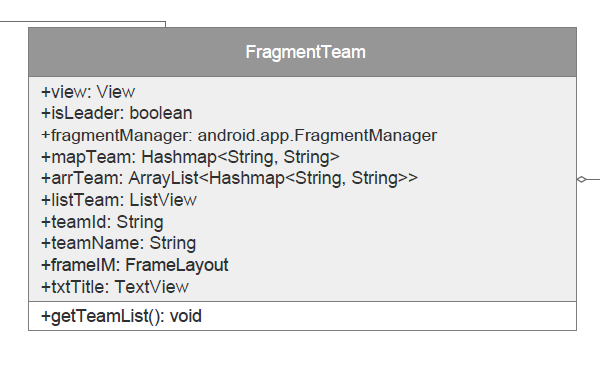
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | view | Declare a variable to identify the view for every function | Type: View |
| 2 | txtProject | Declare a variable to identify a text field to show the text of project | Type: TextView |
| 3 | projectId | Declare a variable  to identify an id of project | Type: String |
| 4 | projectName | Declare a variable  to identify a name of project | Type: String |
| 5 | locationId | Declare a variable  to identify an id of location | Type: String |
| 6 | locationName | Declare a variable  to identify a name of location | Type: String |
| 7 | locationLat | Declare a variable  to identify a latitude of location | Type: String |
| 8 | locationLng | Declare a variable  to identify a longitude of location | Type: String |
| 9 | waterId | Declare a variable  to identify an id of water | Type: String |
| 10 | userId | Declare a variable  to identify an id of user | Type: String |
| 11 | isLeader | Declare a variable to identify the user is a team leader | Type: Boolean |
| 12 | btnAddItem | Declare a variable to identify the item adding button | Type: ImageView |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | dialogParameter() | Method uses to create new parameter of water sampling. | void |
| 2 | getLocation() | Method uses to retrieve the information of location including LocationId, LocationName, LocationLat and LocationLng from database. | void |
| 3 | getWaterList() | Method uses to retrieve the list of water including LocationId, WaterId, ProjectId, UserId, Image, RGB, etc from database. | void |
| 4 | deleteWater(id: String) | Method uses to delete the parameter of water sampling. | void |

**M-CD-07: FragmentTeam**

**Description:** The overview of this class is about the team list containing in each place.



**Property**

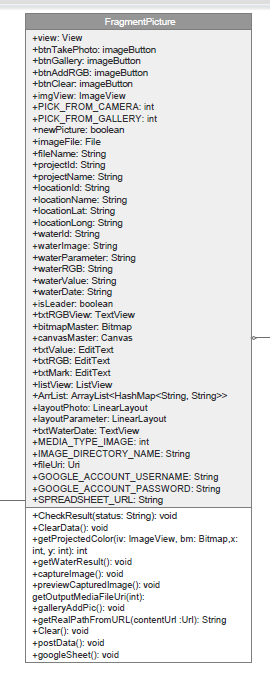
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | view | Declare a variable to identify the view for every function | Type: View |
| 2 | isLeader | Declare a variable to identify the user is a team leader | Type: Boolean |
| 3 | fragmentManager | Declare a variable to identify a service to control every fragment operation | Type: android.app.FragmentManager |
| 4 | mapTeam | Declare a variable to identify set of data of the team before insert to the array list, adding column by column till finish the roll | Type: Hashmap<String, String> |
| 5 | arrTeam | Declare a variable to identify set of data of the team before insert to the table list by receiving from Hashmap | Type: ArrayList<Hashmap<String, String>> |
| 6 | listTeam | Declare a variable to identify a table to show the list of team | Type: ListView |
| 7 | teamId | Declare a variable to identify an id of team | Type: String |
| 8 | teamName | Declare a variable to identify a name of team | Type: String |
| 9 | txtTitle | Declare a variable to identify a text field to show the text of title | Type: TextView |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | getTeamList() | Method uses to retrieve the list of team such as TeamId, TeamName, LeaderId, LeaderName, TeamLeader, etc from database. | void |

**M-CD-08: FragmentPicture**

**Description:** The overview of this class is about the picture selection and preparing for water result calculation with the developed component.



**Property**

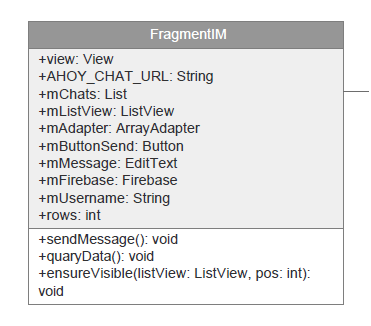
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | view | Declare a variable to identify the  view for every function | Type: View |
| 2 | btnTakePhoto | Declare a variable to identify the  button to take a new photo by mobile camera | Type: ImageButton |
| 3 | btnGallery | Declare a variable to identify the  button to choose a photo from gallery | Type: ImageButton |
| 4 | btnAddRGB | Declare a variable to identify the  button to add RGB band from image | Type: ImageButton |
| 5 | btnClear | Declare a variable to identify the  button to delete the data collection | Type: ImageButton |
| 6 | imgview | Declare a variable to identify an image to view | Type: ImageView |
| 7 | PICK\_FROM\_CAMERA | Declare a variable to identify image selection from mobile camera | Type: Integer |
| 8 | PICK\_FROM\_GALLERY | Declare a variable to identify image selection from mobile gallery | Type: Integer |
| 9 | newPicture | Declare a variable to identify a new picture | Type: Boolean |
| 10 | imageFile | Declare a variable to identify a file of image | Type: File |
| 11 | fileName | Declare a variable to identify a name of file | Type: String |
| 12 | projectId | Declare a variable to identify a name of file | Type: String |
| 13 | projectName | Declare a variable to identify a name of project | Type: String |
| 14 | locationId | Declare a variable to identify an id of location | Type: String |
| 15 | locationName | Declare a variable to identify a name of location | Type: String |
| 16 | locationLat | Declare a variable to identify latitude of location | Type: String |
| 17 | locationLng | Declare a variable to identify longitude of location | Type: String |
| 18 | waterId | Declare a variable to identify an id of water | Type: String |
| 19 | waterImage | Declare a variable to identify an image of water | Type: String |
| 20 | waterParameter | Declare a variable to identify a parameter of water | Type: String |
| 21 | waterRGB | Declare a variable to identify RGB of water | Type: String |
| 22 | waterValue | Declare a variable to identify a value of water | Type: String |
| 23 | isLeader | Declare a variable to identify the user is a team leader | Type: Boolean |
| 24 | txtRGBView | Declare a variable to identify text field to show the text of RGB | Type: TextView |
| 25 | bitmapMaster | Declare a variable to collect the image sent from the server | Type: Bitmap |
| 26 | canvasMaster | Declare a variable to send the image to the server | Type: Canvas |
| 27 | txtValue | Declare a variable to identify edit text field to keep the result value of water parameter | Type: EditText |
| 28 | txtRGB | Declare a variable to identify edit text field to keep the RGB of water parameter | Type: EditText |
| 29 | listView | Declare a variable to identify a table to show the list view | Type: ListView |
| 30 | ArrList | Declare a variable to add the data consisting of ResultId, WaterId, Parameter, RGB and Date in getWaterResult() method in to array | Type: ArrayList<HashMap<String, String>> |
| 31 | layoutPhoto | Declare a variable to show the image in in the View | Type: LinearLayout |
| 32 | layoutParameter | Declare a variable to show the parameter in the View | Type: LinearLayout |
| 33 | txtWaterDate | Declare a variable to identify text field to show a test date | Type: TextView |
| 34 | MEDIA\_TYPE\_IMAGE | Declare a variable to identify the type of media is image | Type: int |
| 35 | IMAGE\_DIRECTORY\_  NAME | Declare a variable to identify a directory of image | Type: String |
| 36 | fileUri | Declare a variable to identify the Uri of image file | Type: Uri |
| 37 | GOOGLE\_ACCOUNT\_  USERNAME | Declare a variable to identify a username of Google account | Type: String |
| 38 | GOOGLE\_ACCOUNT\_  PASSWORD | Declare a variable to identify a password of Google account | Type: String |
| 39 | SPREADSHEET\_URL | Declare a variable to identify a path of Google SpreadSheet | Type: String |

**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | checkResult(status: String) | Method uses to check the status of water parameter. | void |
| 2 | clearData() | Method uses to clear the data for refreshing after previous saving. | void |
| 3 | getProjectedColor(iv: ImageView, bm:  Bitmap , x: int , y: int ) | Method uses to refresh the data after previous saving. | Integer |
| 4 | getWaterResult() | Method uses to receive the result value of water parameter. | void |
| 5 | captureImage() | Method uses to get the image from taking a photo by camera. | void |
| 6 | previewCapturedImage() | Method uses to preview captured image. | void |
| 7 | getOutputMediaFileUri(type: int) | Method uses to retrieve file from Uri. | Uri |
| 8 | galleryAddPic() | Method uses to add a new photo to the gallery in mobile phone. | void |
| 9 | getRealPathFromURI(contentURI: Uri ) | Method uses to refer the photo’s address in the mobile phone when user selects from gallery. | String |
| 10 | clear() | Method uses to clear the image for refreshing after previous saving. | void |
| 11 | postData() | Methods uses to post the data to Google Sheet through Url | void |
| 12 | googleSheet() | Method uses to call thread to start data uploading and send to Google Sheet | void |

**M-CD-09: FragmentIM**

**Description:** The overview of this class is about the messenger system using Firebase API (Firebase is a real time chatting application platform for Android).



**Property**

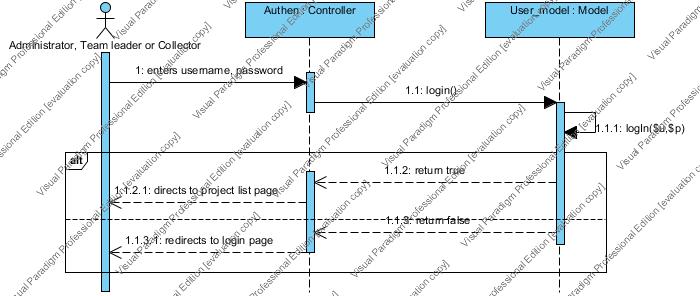
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | view | Declare a variable to identify the  view for every function | Type: View |
| 2 | AHOY\_CHAT\_URL | Declare a variable to identify the  path url from firebase | Type: String |
| 3 | mChats | Declare a variable to identify the array list of the chat containing teams. | Type: List |
| 4 | mListView | Declare a variable to identify a table to show the list view | Type: ListView |
| 5 | mAdapter | Declare a variable to notify the data set changed | Type: ArrayAdapter |
| 6 | mButtonSend | Declare a variable to identify the  data sending button | Type: Button |
| 7 | mMessage | Declare a variable to identify the  message text | Type: EditText |
| 8 | mFirebase | Declare a variable to identify Firebase object | Type: Firebase |
| 9 | mUsername | Declare a variable to identify the name of user | Type: String |
| 10 | rows | Declare a variable to identify the row in the chat | Type: Integer |

**Method**

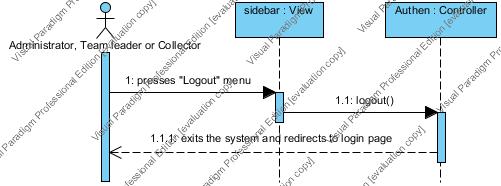
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Return** |
| 1 | sendMessage() | Method uses to send the messages. | void |
| 2 | queryData() | Method uses to retrieve the message text in the team set in row. | void |
| 3 | ensureVisible(listView: ListView , pos: int ) | Method uses to check the visible message in the chatting system. | void |

## 4.3 Sequence Diagram

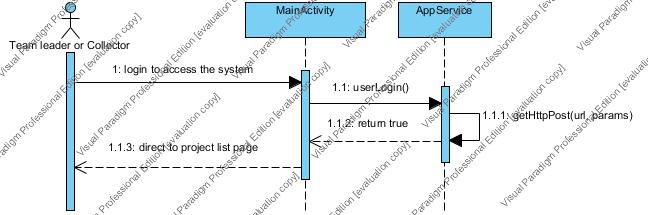
**[SD-01] Log in to the system on web application**

****

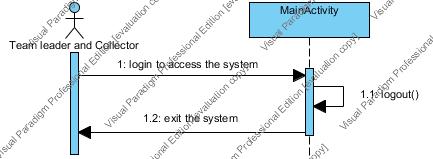
**[SD-02] Logout from the system on web application**

****

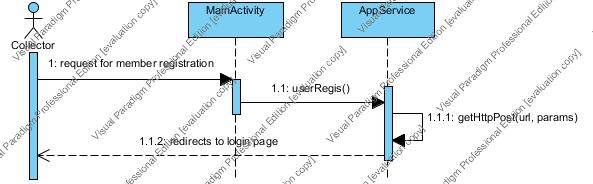
**[SD-03] Log in to the system on mobile application**



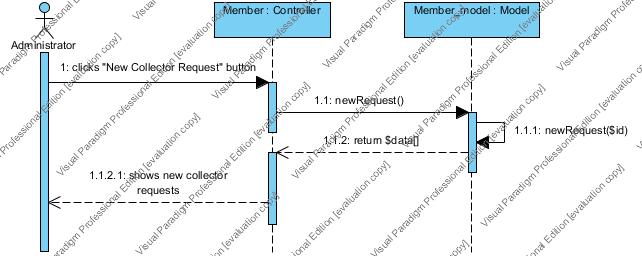
**[SD-04] Logout from the system on mobile application**

****

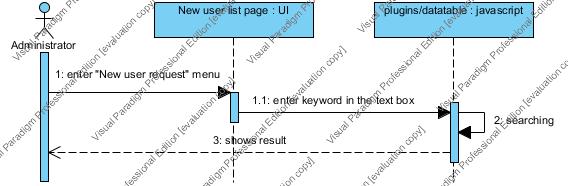
**[SD-05] Send register request to the administrator on mobile application requesting to be a member**

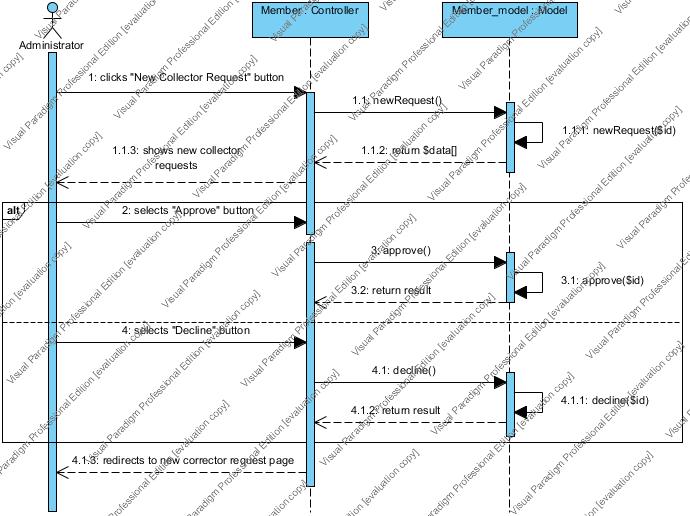
****

**[SD-06] View list of new collector requests sorted by date on web application**

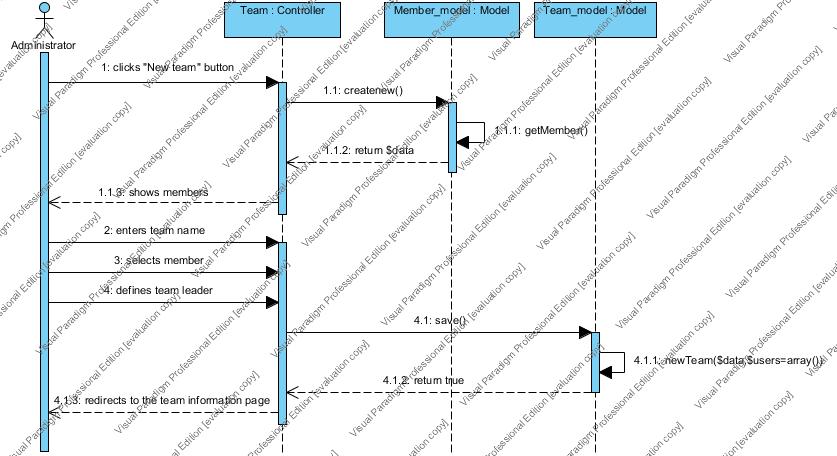
****

**[SD-07] Search a new collector request by using new collector name on web application**

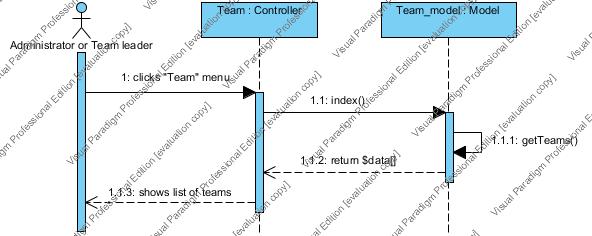
****

**[SD-08] Select to approve or decline new member on web application**

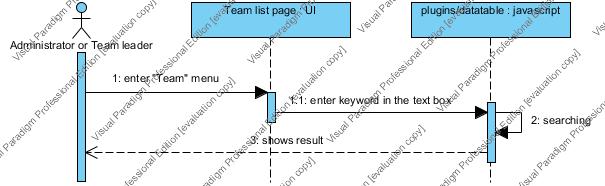
**[SD-09] Create a team which contains a team name, team leader and team members on web application**

****

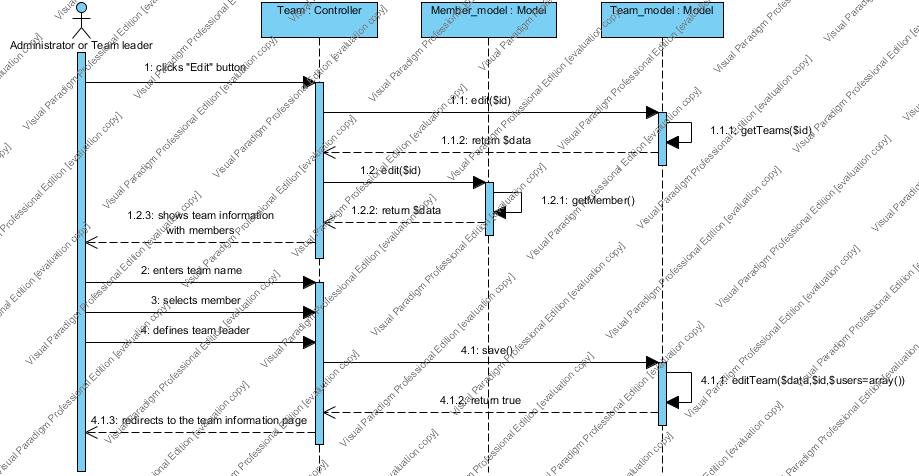
**[SD-10] View list of team sorted by team name on web application**

****

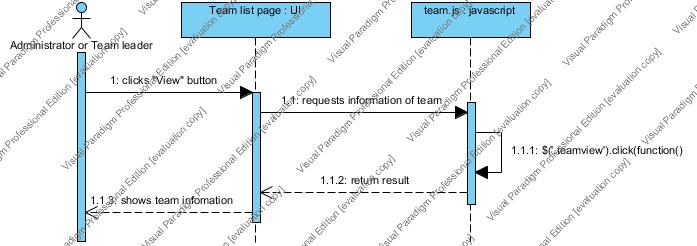
**[SD-11] Search the team by using team name on a web application**

****

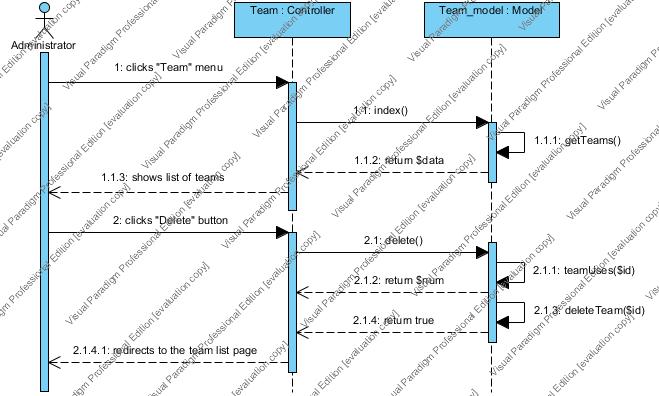
**[SD-12] Modify the selected team information on web application**

****

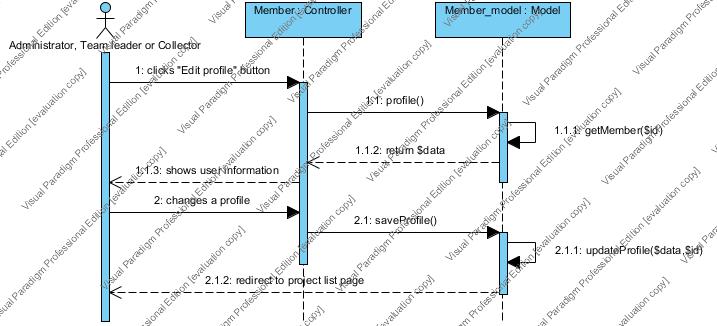
**[SD-13] View the selected team information includes a team name, team leader name, and list of members on web application.**

****

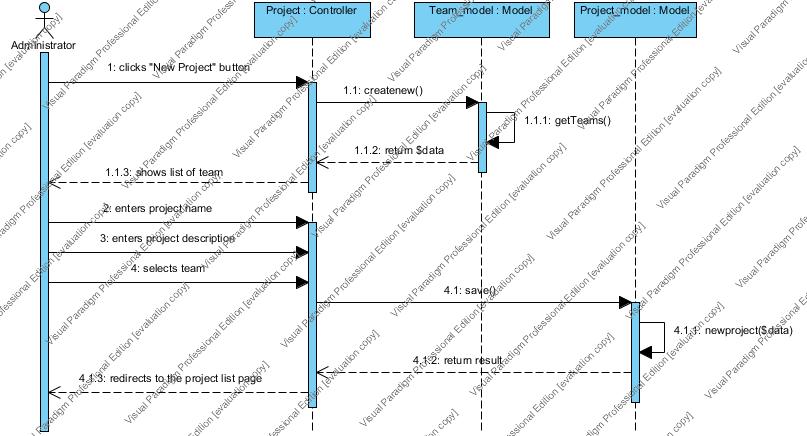
**[SD-14] Remove a team out of the system on web application**

****

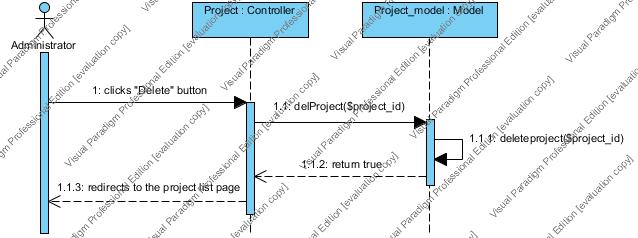
**[SD-15] Edit profile information which includes a name, password and telephone number on web application**

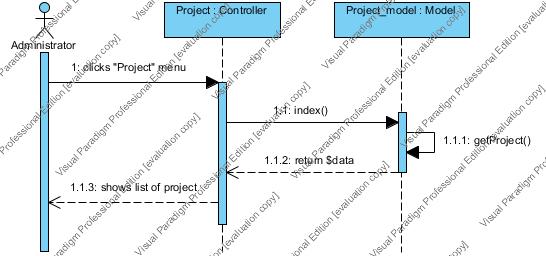
****

**[SD-16] Create the project includes a project name, project description, and team on web application**

****

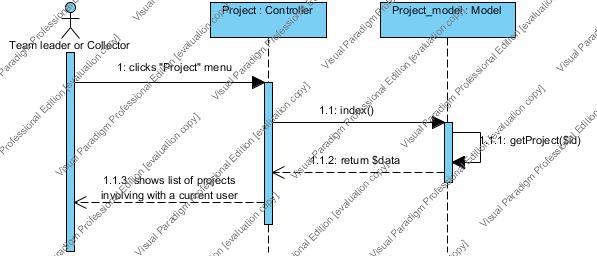
**[SD-17] Remove the project out of the system on web application**

****

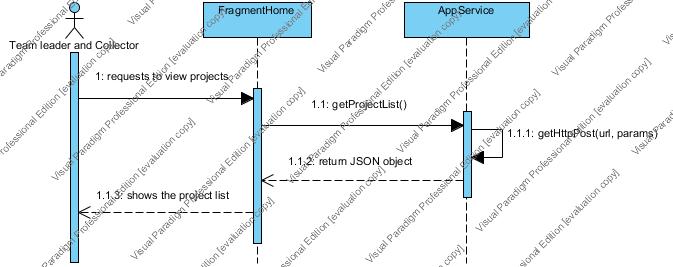
**[SD-18] View list of all project sorted by created date of the project on a web application**

**[SD-19] View list of the involved projects sorted by created date of the project on a web application and mobile application**

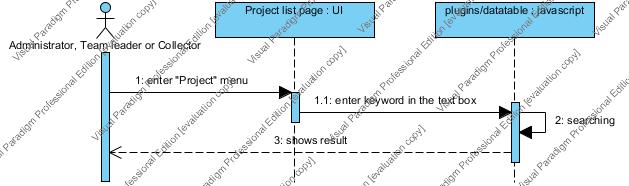
* Web application



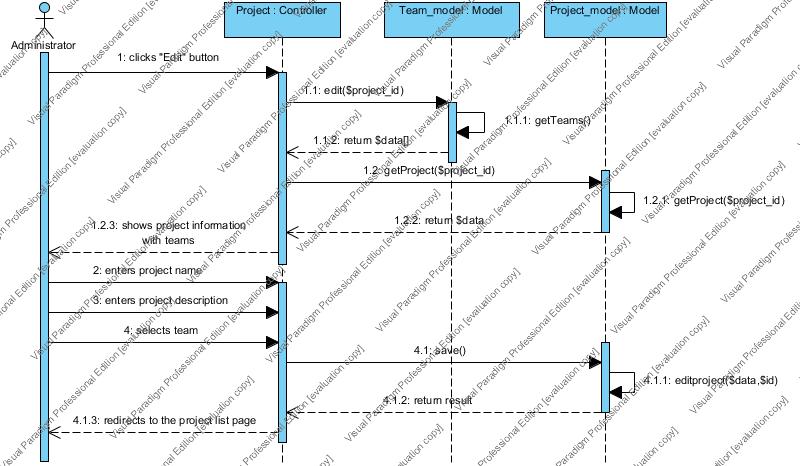
* Mobile application



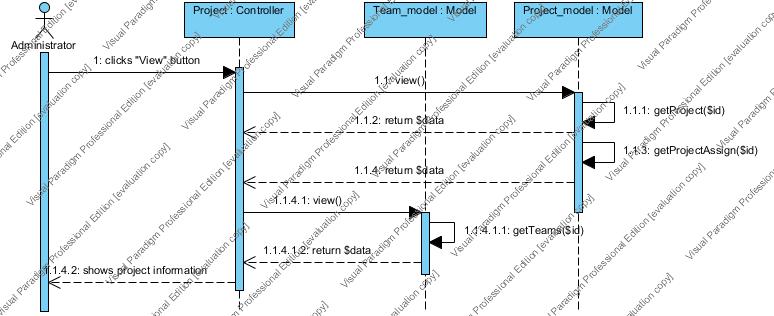
**[SD-20] Search the project by using project name on web application**

****

**[SD-21] Modify the project information includes editing project name, editing project description, and changing a team on web application**

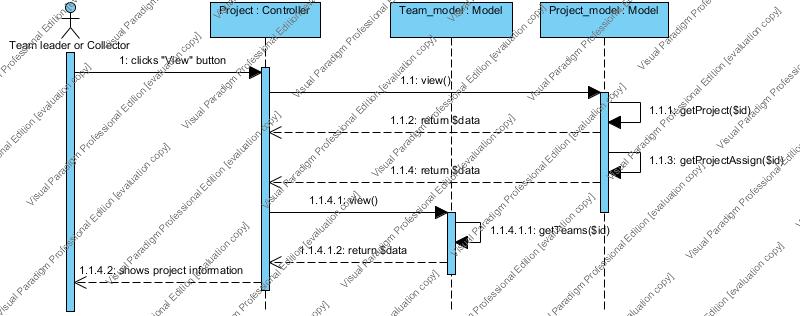
****

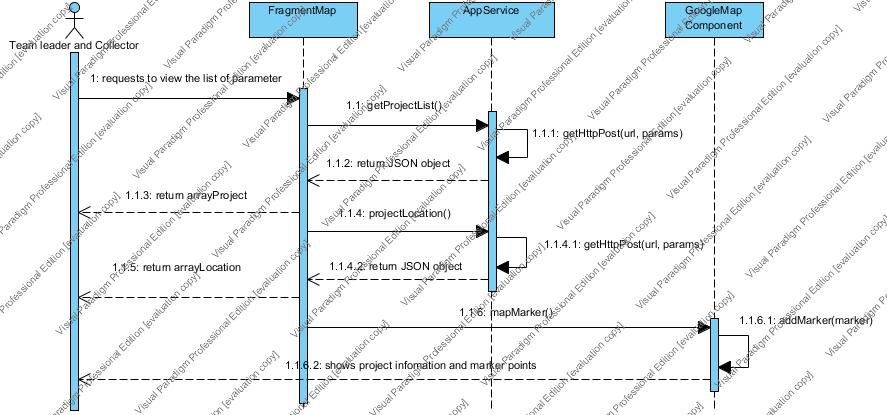
**[SD-22] View detail of selected projects information which consisting of map with assigned pins and location information on a web application**

****

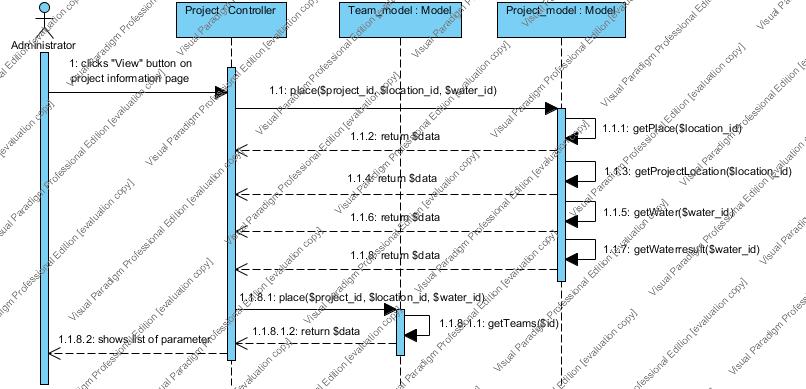
**[SD-23] View detail of selected projects information which consisting of map with assigned pins and location information of the selected project on a web application and mobile application**

* Web application

****

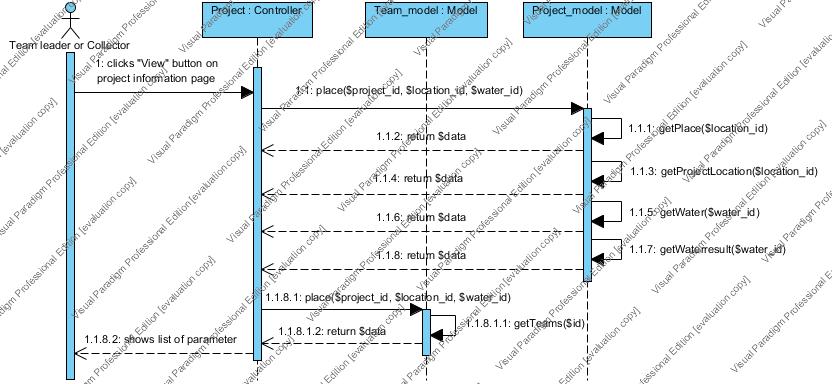
* Mobile application

**[SD-24] View list of water parameter of the selected member sorted by date on a web application**

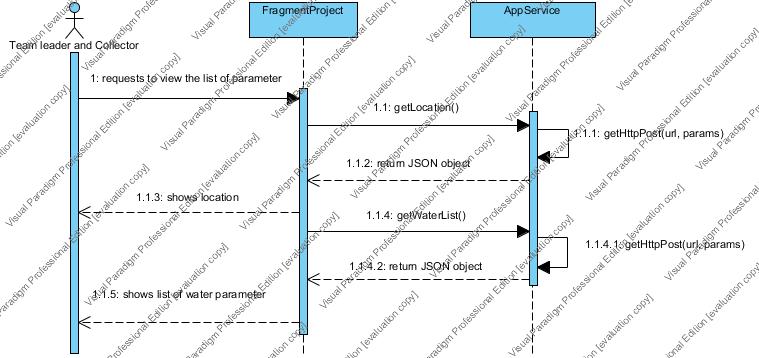
****

**[SD-25] View list of water parameter of the selected member sorted by date on a web application and mobile application**

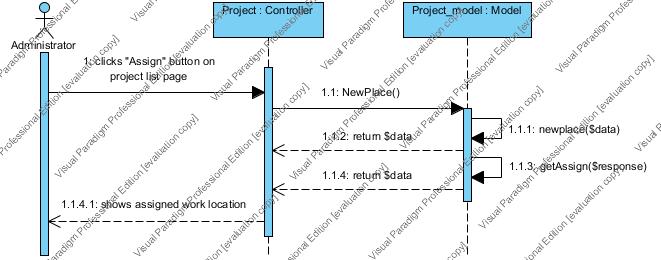
* Web application

****

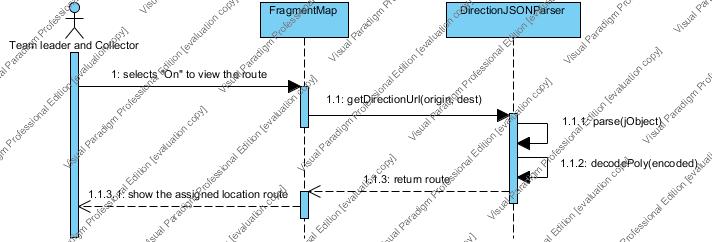
* Mobile application

****

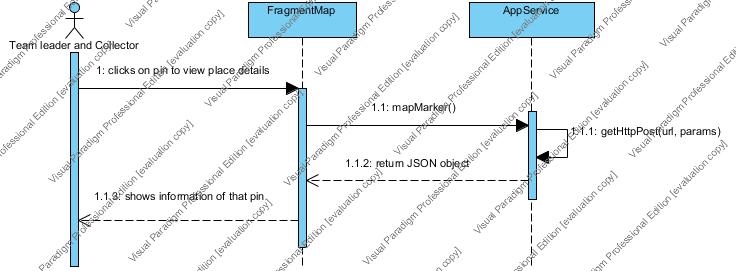
**[SD-26] Assign work location on Google map to each collectors on web application**

****

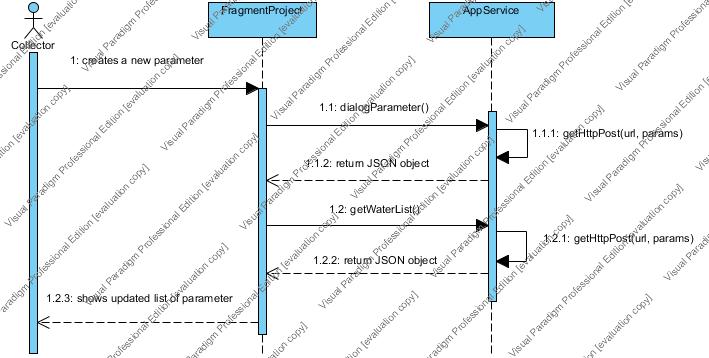
**[SD-27] View the direction to the selected location on Google Maps on mobile application**

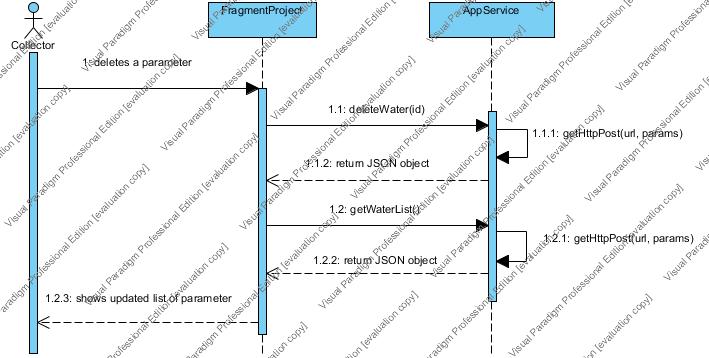
****

**[SD-28] View location information of each assigned pin on selected project by pressing on the pin on mobile application**

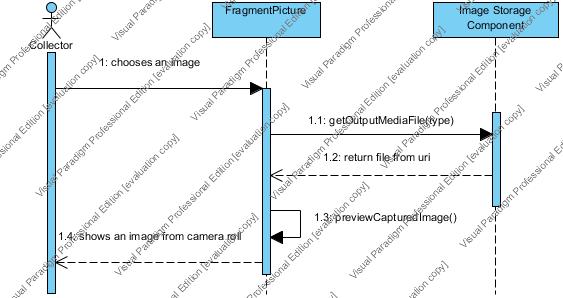
****

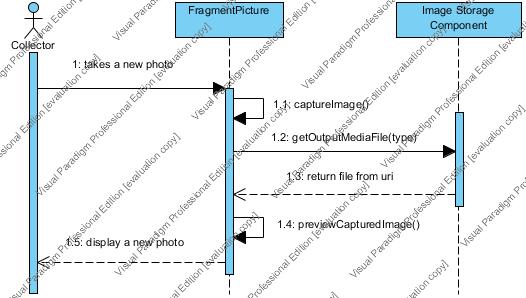
**[SD-29] Add water parameter to collect the test data on mobile application**

****

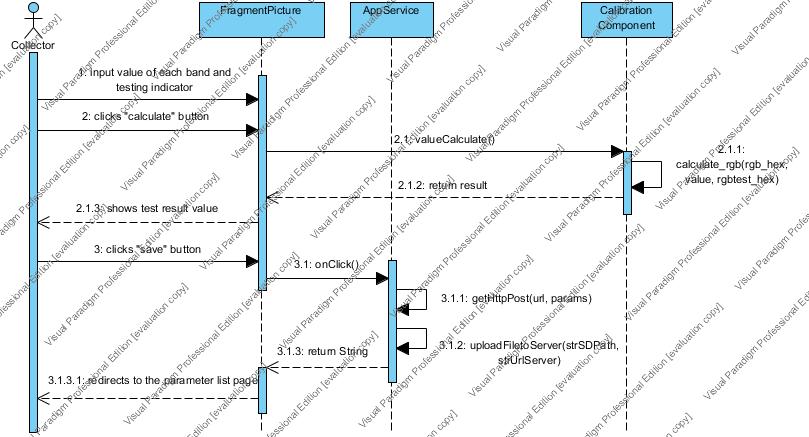
**[SD-30] Delete water parameter on mobile application**

**[SD-31] Choose the image from camera roll or take a new photo to collect RGB value with the test result on mobile application**

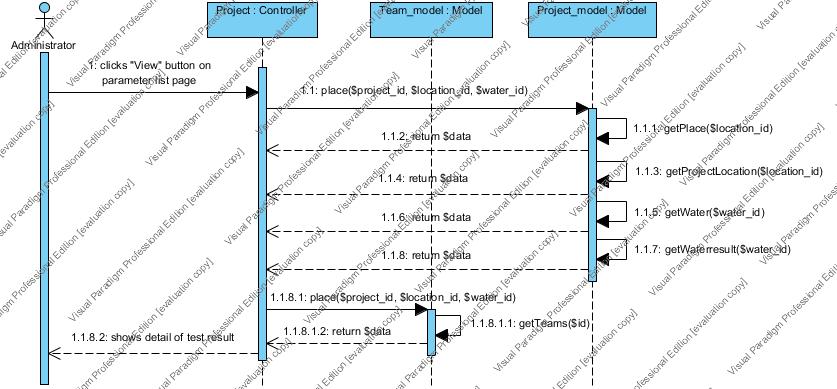
* Camera roll
* New photo



**[SD-32] Input data to let the system calculate the test result based on standard color scale on mobile application**

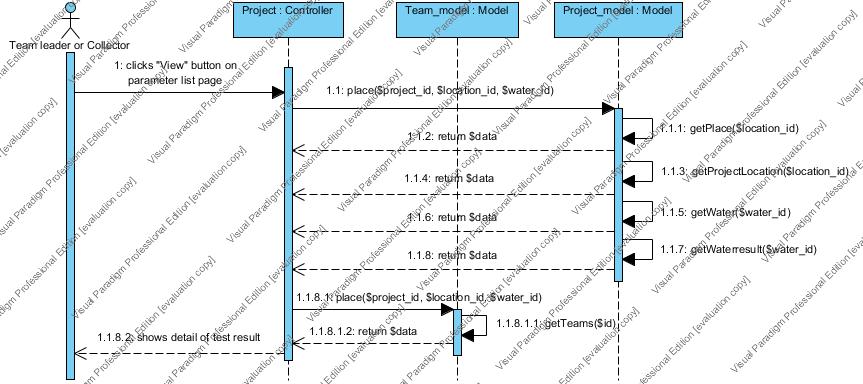
****

**[SD-33] View detail of test result of selected water parameter on a web application**

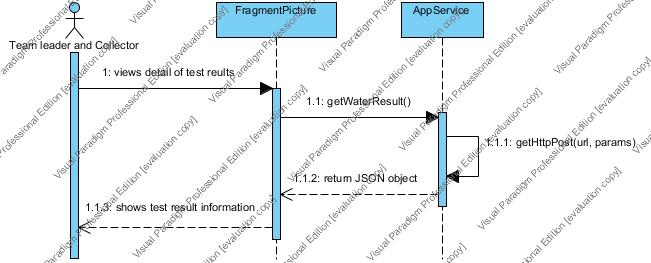
****

**[SD-34] View detail of test result of selected water parameter on a web application and mobile application**

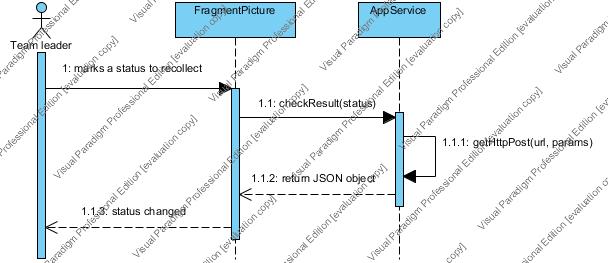
* Web application

****

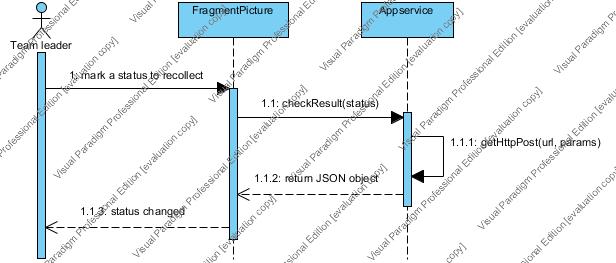
* Mobile application

****

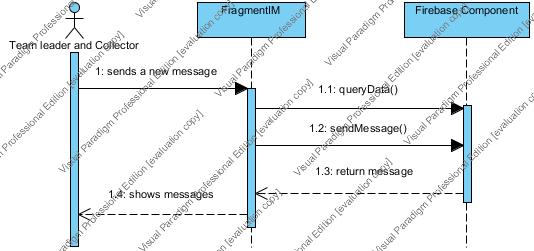
**[SD-35]** Mark status of the selected water parameter to “Recollect” on a web application and mobile application

****

**[SD-36] Mark status of the selected water parameter to “Finish” on a web application and mobile application**

****

**[SD-37] Send the message to each other via group message within the team on a mobile application**

****

# Chapter 4-5 | Data Architecture

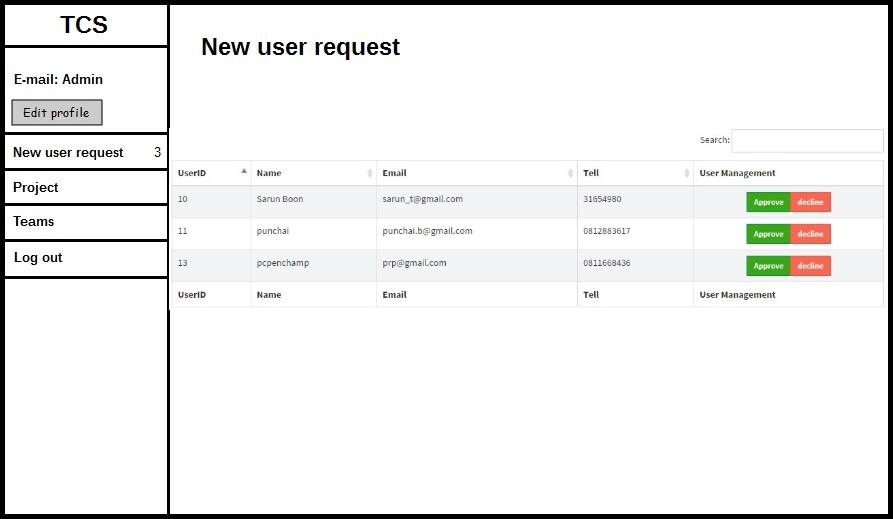
## 5.1 Database Design

# Chapter 4-6 | User Interface design

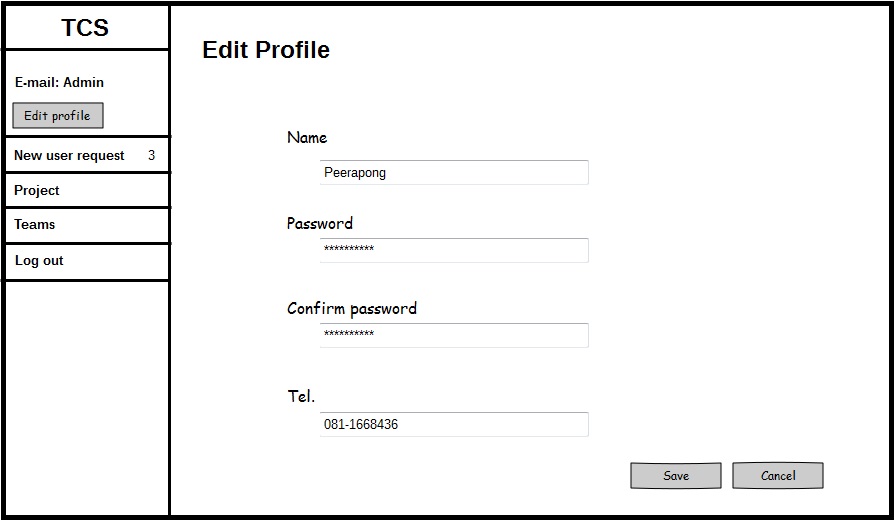
* **Web application Part**



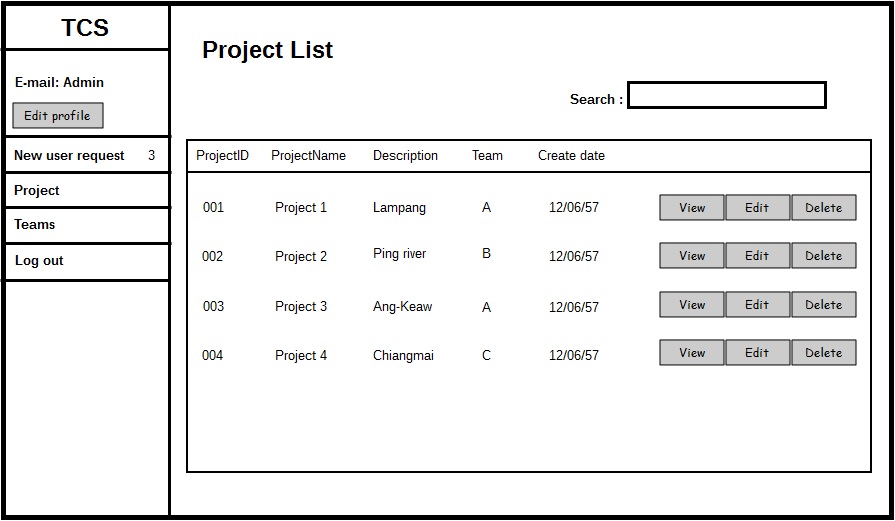
|  |  |
| --- | --- |
| **User Interface ID :** | UI-01 |
| **User Interface Name :** | W-Login page |
| **Short Description :** | User can login to the system using his username and password. |
| **Actor :** | Administrator, Team leader, Collector |



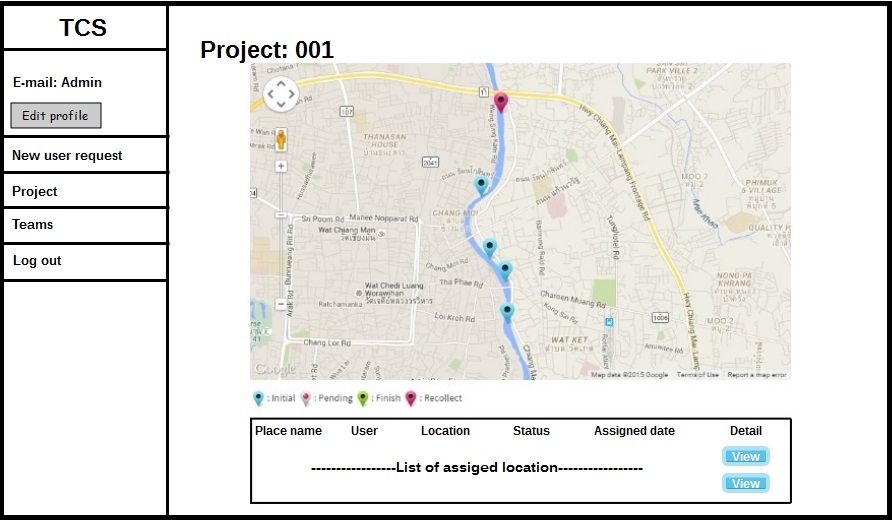
|  |  |
| --- | --- |
| **User Interface ID :** | UI-02 |
| **User Interface Name :** | WA-New User Request Page |
| **Short Description :** | User can view list of new collector requests and can select to approve or decline new collector requests on web application. |
| **Actor :** | Administrator |



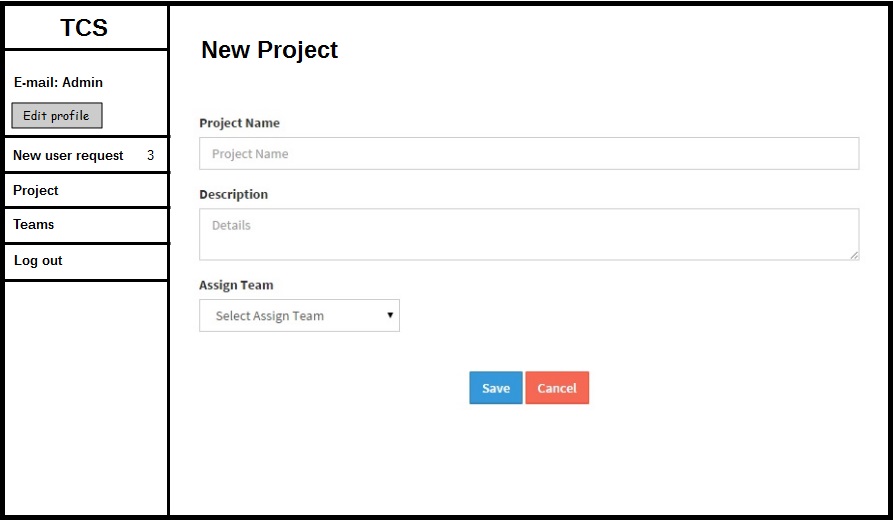
|  |  |
| --- | --- |
| **User Interface ID :** | UI-03 |
| **User Interface Name :** | WA-Edit profile page |
| **Short Description :** | -User can edit his profile which is name, password and telephone number on web application. |
| **Actor :** | Administrator, Team leader, and Collector |



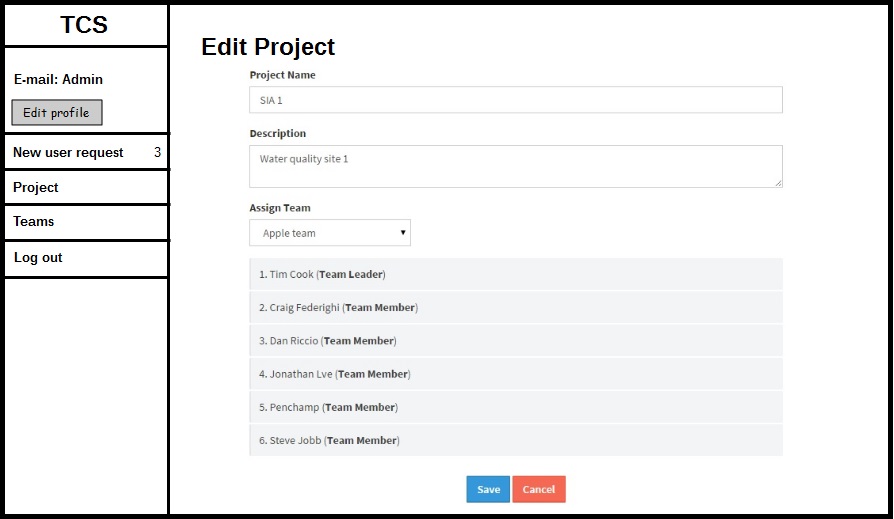
|  |  |
| --- | --- |
| **User Interface ID :** | UI-04 |
| **User Interface Name :** | WA-Project list page |
| **Short Description :** | * User can view list of all project sorted by date created and also can search project by name on web application. * User can search project by name. * User can click “New Project” button to go Create new project. * User can click “View” button to view the project information. * User can click “Edit” button to edit the project. * User can Click “Delete” button to delete the project. |
| **Actor :** | Administrator |



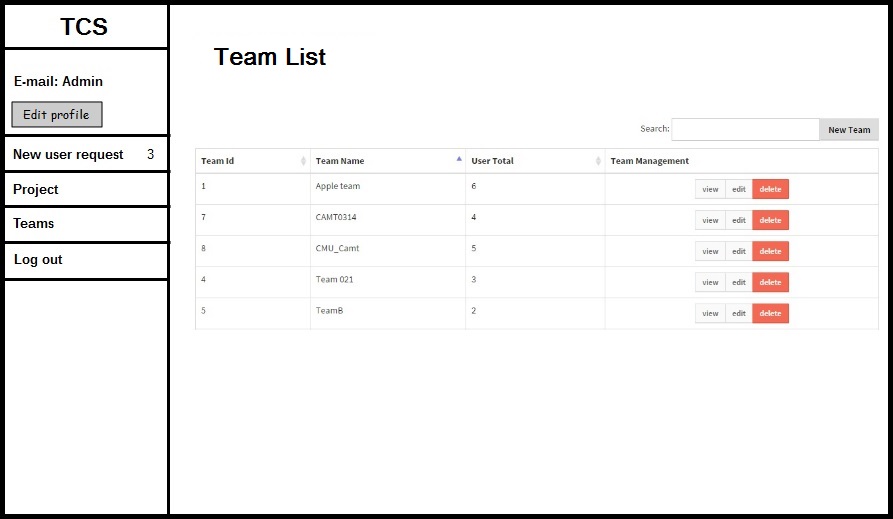
|  |  |
| --- | --- |
| **User Interface ID :** | UI-05 |
| **User Interface Name :** | WA-Project information page |
| **Short Description :** | * User can view the selected project information on a web application including Place name, name of user, latitude, longitude, status and assigned date. * User can click “View” button to view detail of test result. |
| **Actor :** | Administrator |



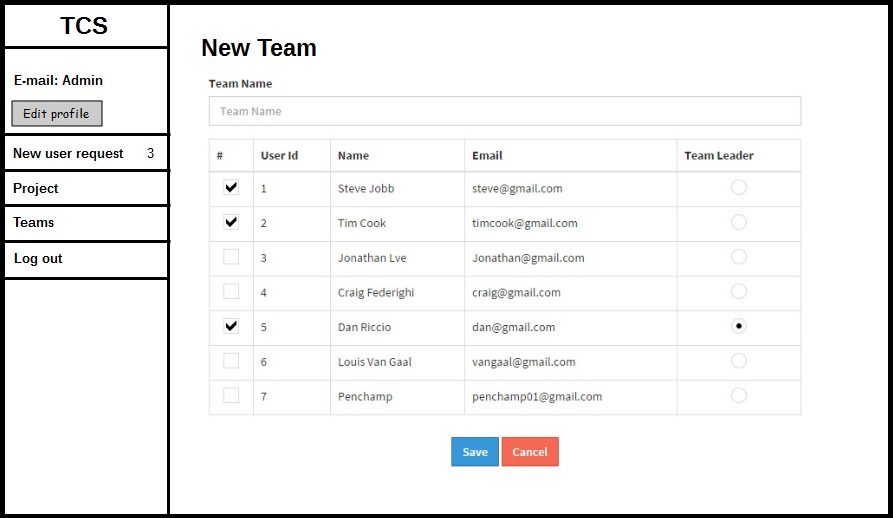
|  |  |
| --- | --- |
| **User Interface ID :** | UI-06 |
| **User Interface Name :** | WA-New project page |
| **Short Description :** | User can create the project including a project name, project description, and add a team to the project on web application. |
| **Actor :** | Administrator |



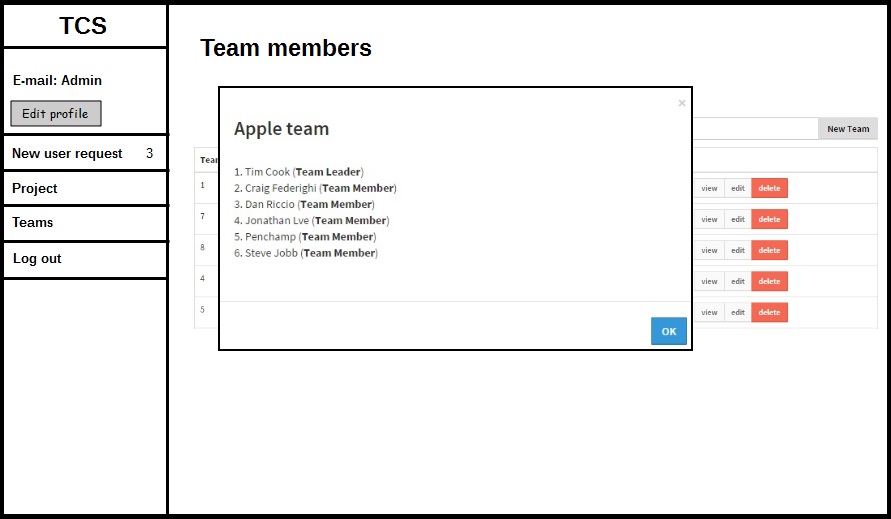
|  |  |
| --- | --- |
| **User Interface ID :** | UI-07 |
| **User Interface Name :** | WA-Edit project page |
| **Short Description :** | User can modify the project includes editing project name, project description, and changing the team on web application. |
| **Actor :** | Administrator |



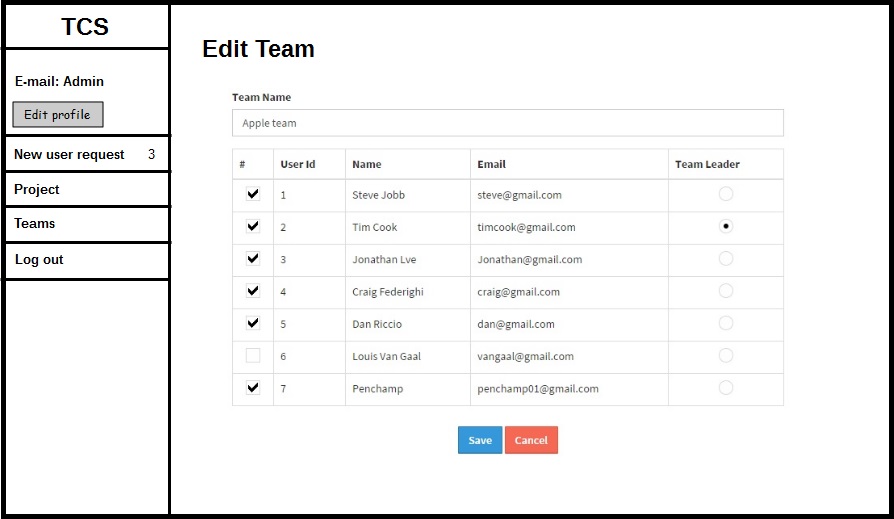
|  |  |
| --- | --- |
| **User Interface ID :** | UI-08 |
| **User Interface Name :** | WA-Team list page |
| **Short Description :** | * User can view list of team sorted by name on web application including Team Id, Team name, and amount of member. * User can click “New team” to create a new team. * User can click “View” button to view information of team. * User can click “Edit” button to edit information of team. * User can click “Delete” button to delete the team. |
| **Actor :** | Administrator |



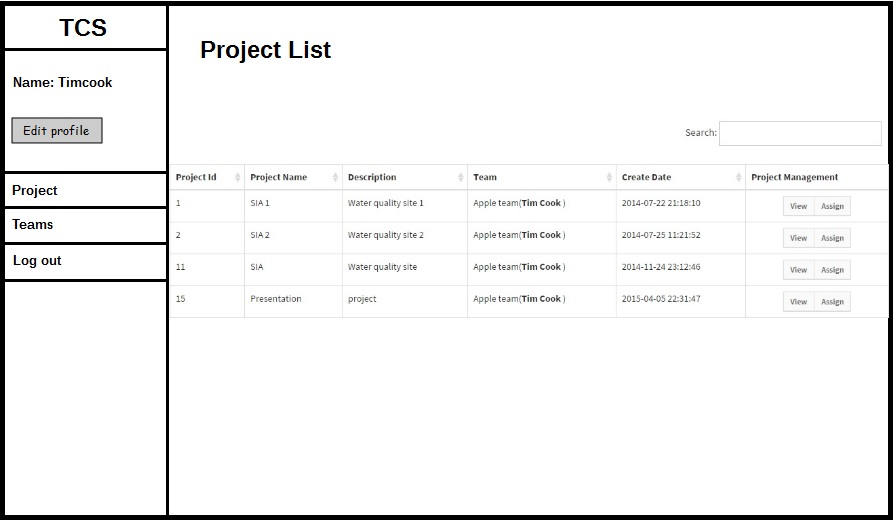
|  |  |
| --- | --- |
| **User Interface ID :** | UI-09 |
| **User Interface Name :** | WA-New team page |
| **Short Description :** | User can create a team which contains a team name, list of all member and user can select a member into a team and also define team leader on web application. |
| **Actor :** | Administrator |



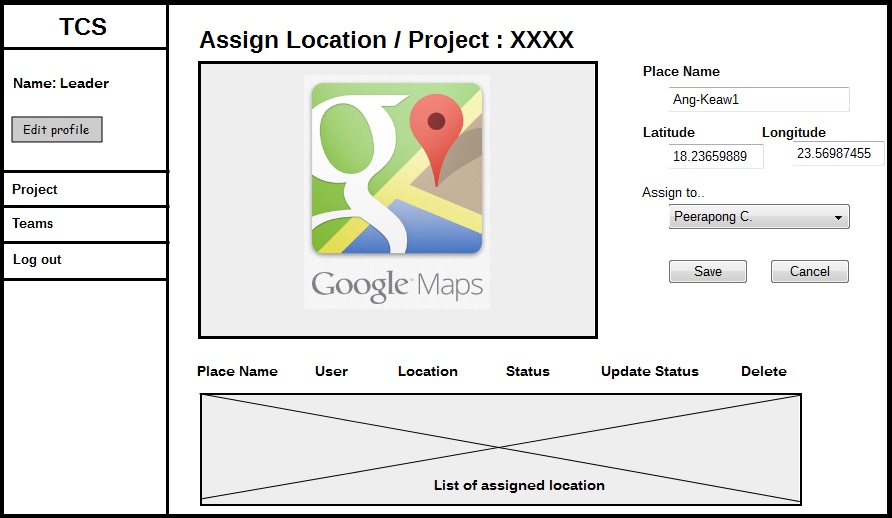
|  |  |
| --- | --- |
| **User Interface ID :** | UI-10 |
| **User Interface Name :** | WA-Team information page |
| **Short Description :** | User can view the selected team information includes a team name, team leader name, and list of members on web application. |
| **Actor :** | Administrator |



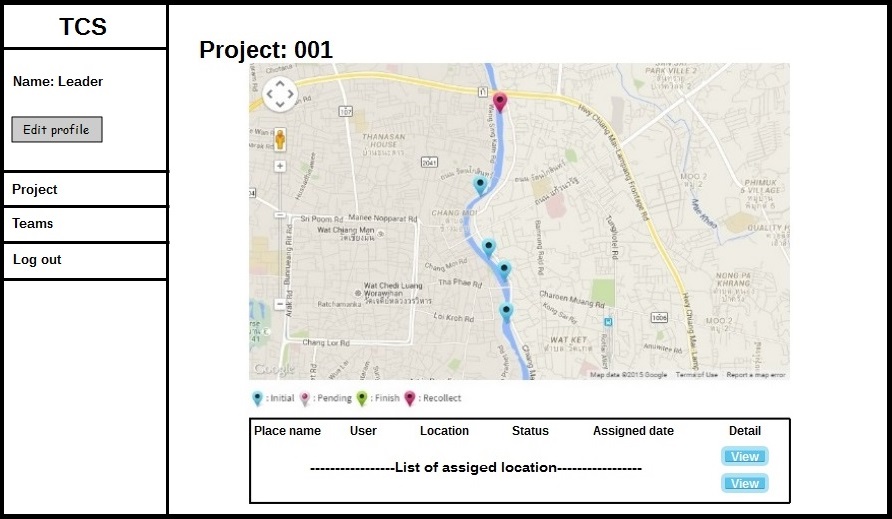
|  |  |
| --- | --- |
| **User Interface ID :** | UI-11 |
| **User Interface Name :** | WA-Edit team page |
| **Short Description :** | User can modify the team on web application including team name, team leader, adding team member, and removing team member. |
| **Actor :** | Administrator |



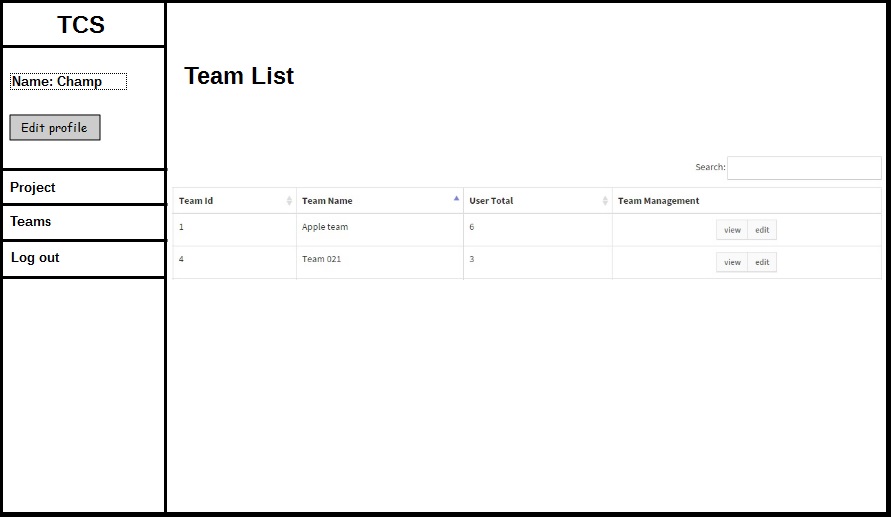
|  |  |
| --- | --- |
| **User Interface ID :** | UI-12 |
| **User Interface Name :** | WT-Project list page |
| **Short Description :** | * User can view list of project sorted by date created on web application. * User can search project by name. * User can click “Assign” to assign location to each collector |
| **Actor :** | Team leader |



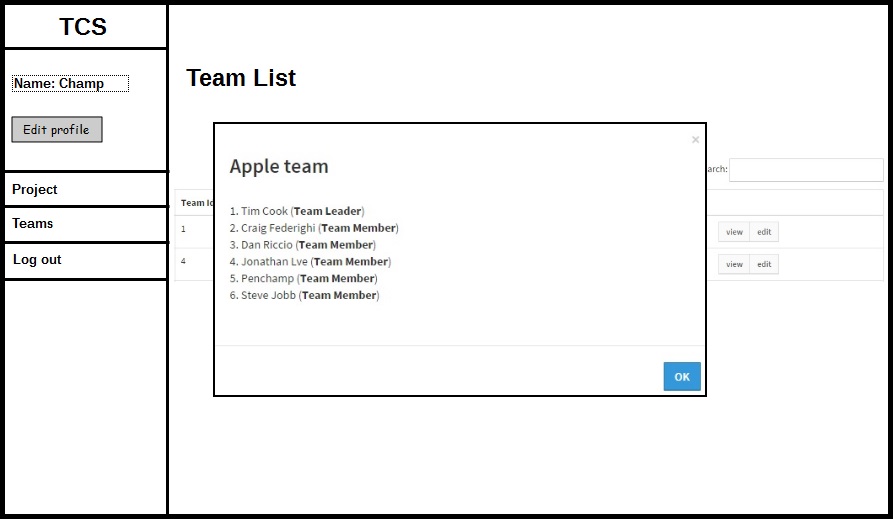
|  |  |
| --- | --- |
| **User Interface ID :** | UI-13 |
| **User Interface Name :** | WT-Assignment page |
| **Short Description :** | User assign the work location to each collector by input the place name and mark the location on Google map and also select the member to do the assigned location. |
| **Actor :** | Team leader |



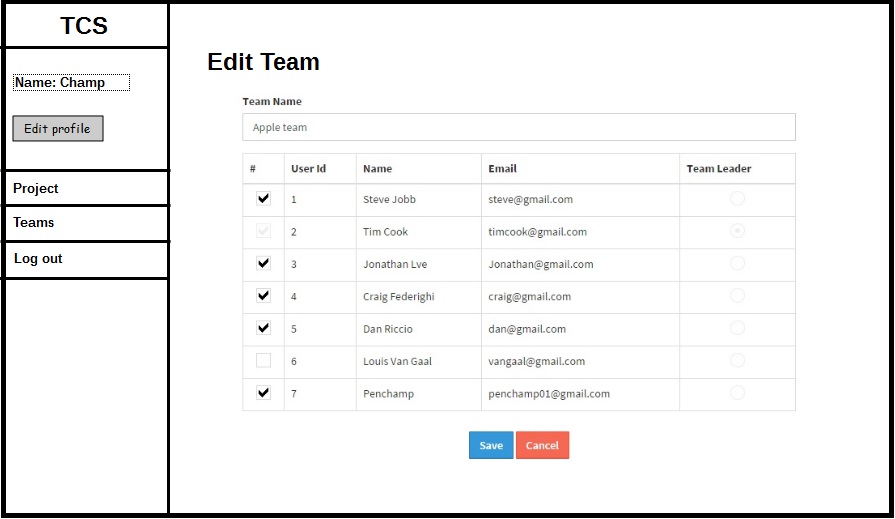
|  |  |
| --- | --- |
| **User Interface ID :** | UI-14 |
| **User Interface Name :** | WT-Project information page |
| **Short Description :** | User can view the selected project information on a web application including Place name, name of user, latitude, longitude, status and assigned date.  User can click “View” button to view detail of the project. |
| **Actor :** | Team leader |



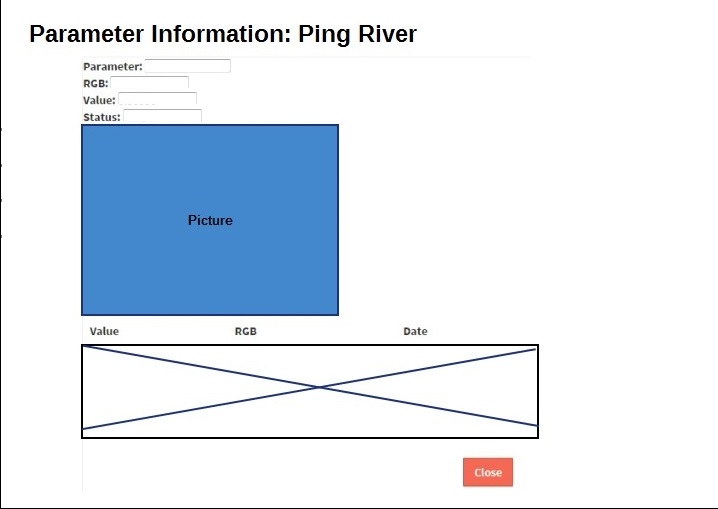
|  |  |
| --- | --- |
| **User Interface ID :** | UI-15 |
| **User Interface Name :** | WT-Team list page |
| **Short Description :** | * User can view list of team sorted by name on web application. * User can click “View” button to view information of a team. * User can click “Edit” button to edit information of a team. |
| **Actor :** | Team leader |



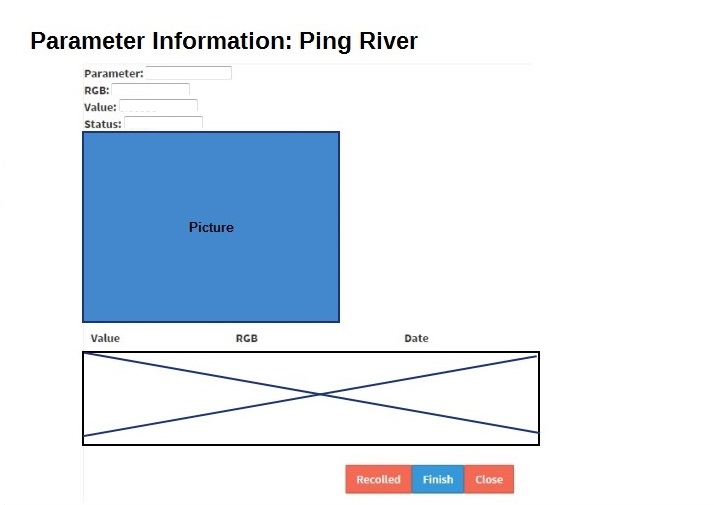
|  |  |
| --- | --- |
| **User Interface ID :** | UI-16 |
| **User Interface Name :** | WT-Team information page |
| **Short Description :** | User can view the selected team information including a team name, team leader name, and list of members on web application. |
| **Actor :** | Team leader |



|  |  |
| --- | --- |
| **User Interface ID :** | UI-17 |
| **User Interface Name :** | WT-Edit team page |
| **Short Description :** | User can modify the team on web application including team name, team leader, adding team member, and removing team member. |
| **Actor :** | Team leader |

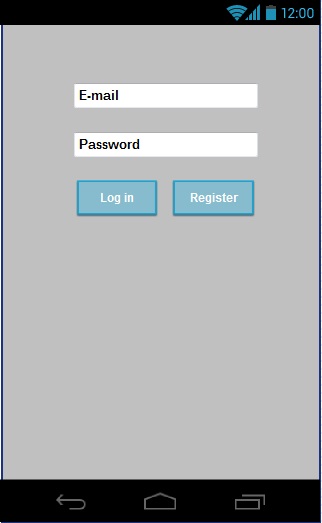


|  |  |
| --- | --- |
| **User Interface ID :** | UI-18 |
| **User Interface Name :** | W-Parameter information |
| **Short Description :** | User view detail of test result in this page. The information consist of parameter name, predicted value, RGB value and the image with the detail of the data that use to calculate the test result |
| **Actor :** | Administrator and Collector |

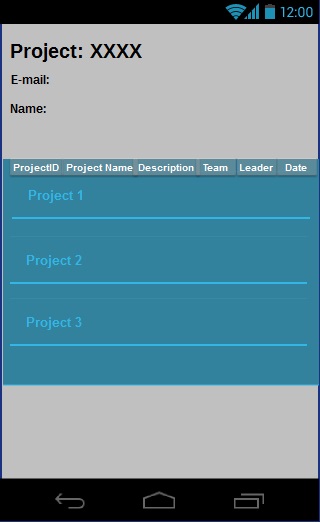


|  |  |
| --- | --- |
| **User Interface ID :** | UI-19 |
| **User Interface Name :** | WT-Parameter information |
| **Short Description :** | User view detail of test result in this page. The information consist of parameter name, predicted value, RGB value and the image with the detail of the data that use to calculate the test result  User can review the test result by click “Recollect” or “Finish” button. |
| **Actor :** | Team leader |

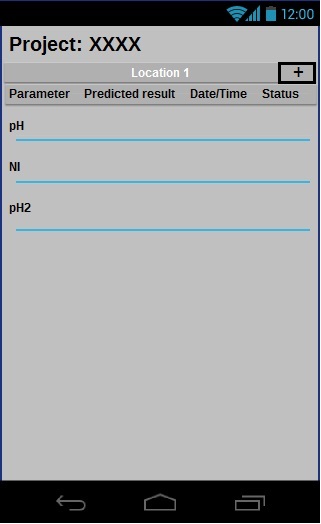
* **Mobile application Part**



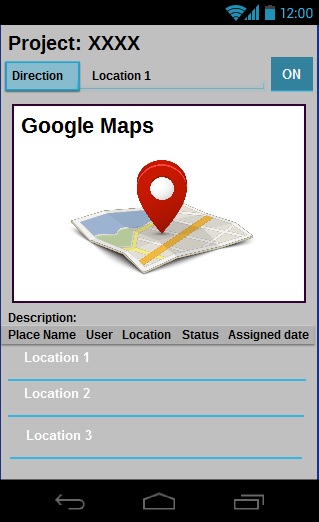
|  |  |
| --- | --- |
| **User Interface ID :** | UI-20 |
| **User Interface Name :** | M-Login page |
| **Short Description :** | * User can login to the system. * New user can register to use the system by click “Register” button |
| **Actor :** | Team leader and Collector |



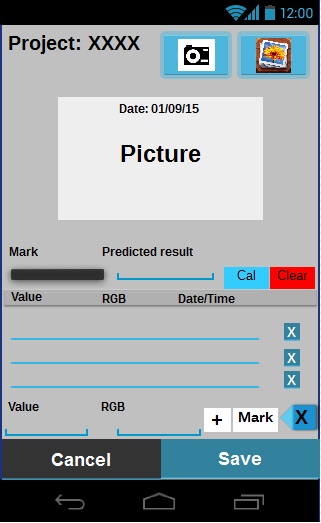
|  |  |
| --- | --- |
| **User Interface ID :** | UI-21 |
| **User Interface Name :** | M-Project list page |
| **Short Description :** | User can view list of involve project sorted by date created on web application. |
| **Actor :** | Team leader and Collector |



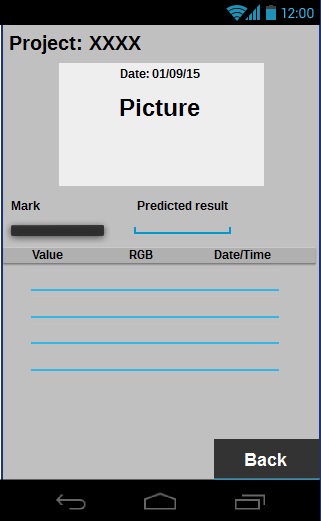
|  |  |
| --- | --- |
| **User Interface ID :** | UI-22 |
| **User Interface Name :** | M-Parameter list page |
| **Short Description :** | * User can view list of parameter with a predicted result, date, time and status * User can create new parameter by click “+” button |
| **Actor :** | Team leader and Collector |



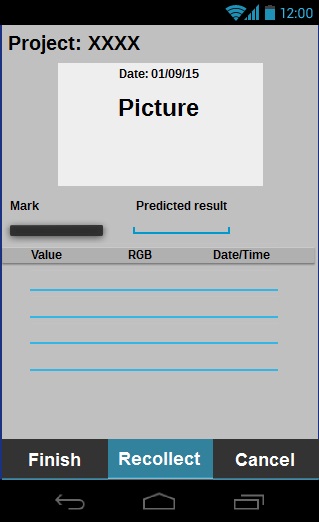
|  |  |
| --- | --- |
| **User Interface ID :** | UI-23 |
| **User Interface Name :** | M-Project information page |
| **Short Description :** | User can view the selected project information on a web application including Place name, name of user, latitude, longitude, status and assigned date. |
| **Actor :** | Team leader, Collector |



|  |  |
| --- | --- |
| **User Interface ID :** | UI-24 |
| **User Interface Name :** | M-Calculation page |
| **Short Description :** | This page is a part of calculation for collector by input the data to calculate the test result. |
| **Actor :** | Collector |



|  |  |
| --- | --- |
| **User Interface ID :** | UI-25 |
| **User Interface Name :** | M-Test result |
| **Short Description :** | User view detail of test result in this page. The information consist of parameter name, predicted value, RGB value and the image with the detail of the data that use to calculate the test result |
| **Actor :** | Collector |



|  |  |
| --- | --- |
| **User Interface ID :** | UI-26 |
| **User Interface Name :** | MT-Test result |
| **Short Description :** | * User view detail of test result in this page. The information consist of parameter name, predicted value, RGB value and the image with the detail of the data that use to calculate the test result * User can review the test result by click “Recollect” or “Finish” buttons. |
| **Actor :** | Team leader |