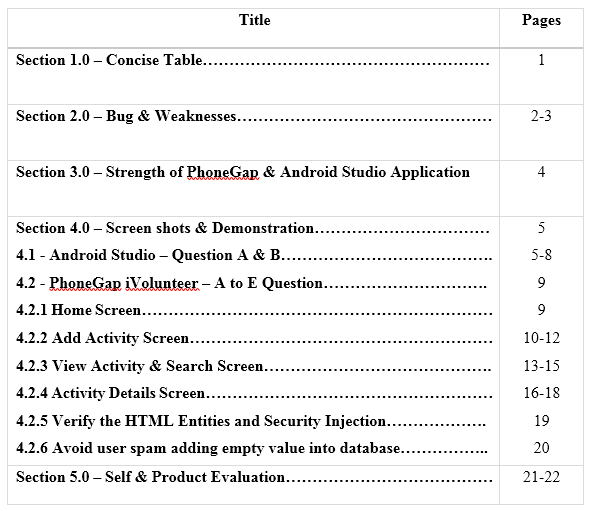


**Table of Content**



**Section 1.0 – Concise Table**

|  |  |
| --- | --- |
| **Feature** | **Implementation** |
| a) | * Fully implemented. |
| b) | * Fully implemented with set date range for validation. |
| c) | * Fully implemented Store, view and delete basic details. * For duplicate it works with different alphabets. * It can bypass the html entities by encoding it, in order to avoid user spamming these kind of value as query injection attack. |
| d) | * Fully implemented search with backend-code using query to retrieve data from database and display it in list-view. * This search function is implemented with advanced search option, such as Search by activity name, location, date, time and volunteer name. |
| e) | * I have created user interface design for adding report in the view activity details screen. * User can add multiple reports with timestamp given. * User can delete for particular report when tap it. |
| f) | * I have fully implemented the android studio which the features and functions is according to question a) and b). |
| g) | * No additional features implemented. |

**Section 2.0 – Bug & Weaknesses**

**Part A – Create an input field in the app screens**

* No any bug and weaknesses in this Part A. All are fully implemented.

**Part B – Implement function for input & duplicate validation**

* In this Part B, all the validation is fully implemented but the validation is not including the limited number of words.

**Part C – Implement store, view and delete the activity details**

* In this part C, the system is able to detect the duplicate activity name; no matter the input is lower case or upper case. Therefore, currently there are no any bugs & weaknesses in this part.

**Part D – Implement search function with advanced**

* Currently no bugs found in this Part D. The search function is fully implemented.

**Part E – Add a report input in specific activity**

* In this Part E, the add report function is fully implemented but the validation is not including the limited number of words.

**Part F – Android Studio (Part A & B)**

* It is similar to Part A. the only issue is that date picker must set a date range within 3 months. I have implemented it in this android studio version also.
* For part B, once all the validation is pass, it will start the intent function to go second screen (Activity Details) then pass all the input values and display it.

**Section 3.0 – Strength of the Application**

**Add Activity Validation**

For date input, I have set a date range only for today and within 3 months for date dialog picker in order to avoid user pick an illogical date as input.

**Security Application**

Another benefit for this application is that it allows the html entities to be insert in SQLite, for instance, html entity like this ‘<!--’. It was using html encoding function to bypass it. So that, user can insert any html entities without any errors. Other than that, this application still can avoid from user doing unethical behavior like using back button from real device (android phone) to confirmation box and spam adding empty value into database.

**Avoid Activity Name Duplicate**

Moreover, for the activity name duplicate validation, this application will detect any duplicate activity name, for instance, the input with upper case or lower case. However, I use my own way to solve this issues by making the input and query select to lower case, so that the system can identify whether it is duplicate or not.

**Advanced Search Function**

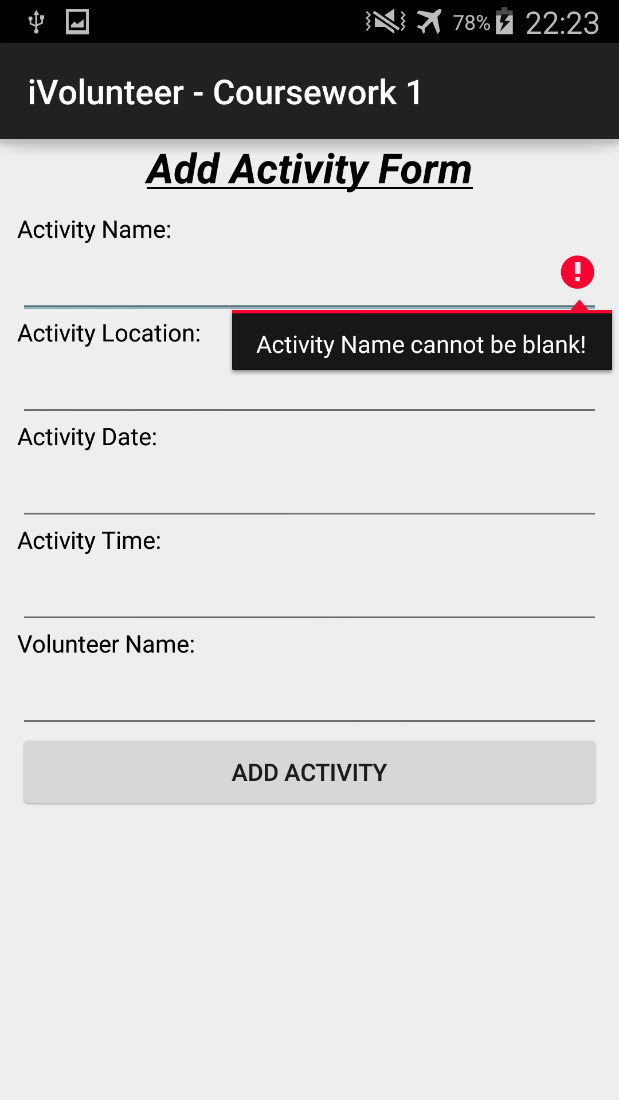
Other than that, this application provided an advanced search function for user. The user can search by activity name, activity location, date, time and volunteer in order to make user more convenient to search in more details. After all, this is for search in more details purpose.

**Delete All Activities at Once**

Furthermore, the application provided a function which delete all activities at once. Therefore, when user mistake in inserting wrong activities in multiple times, user can just simply click the delete all activities button.

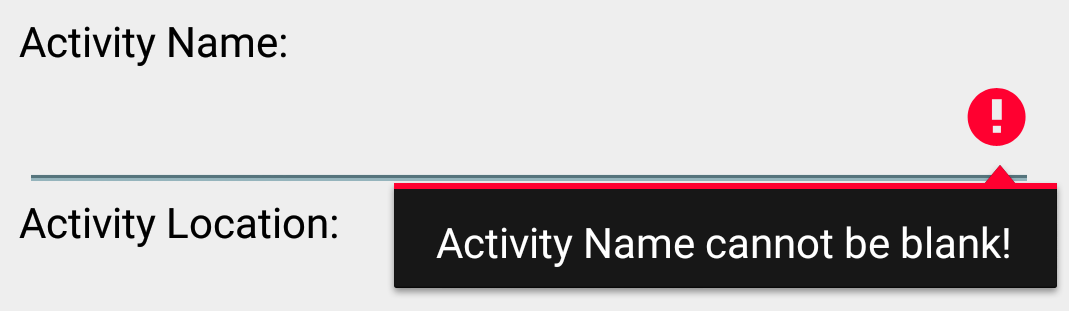
**Section 4.0 – Screen shots & Demonstration**

4.1 Android Studio – Question A & B:



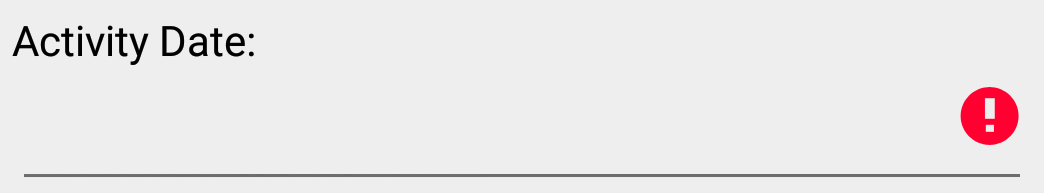
**Figure 4.1.1 – Add Activity Form Screen**

In this figure 4.1.1, this is Add Activity Form Screen. In this screen it contains edit texts and button. For edit texts it including activity name, location, date, time and volunteer name. There is only 1 Add Activity button there. This add activity button is use to submit the edit text value to another activity.



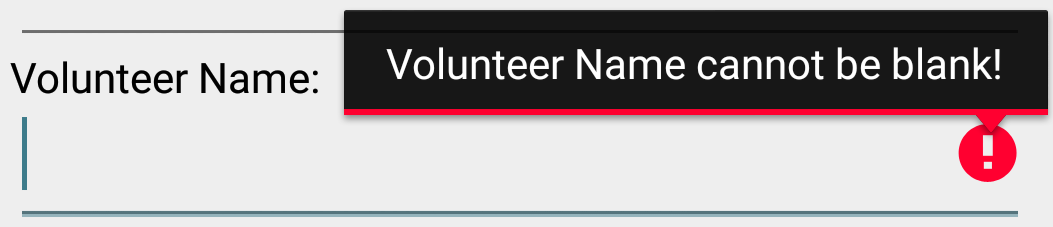
**Figure 4.1.2 – Activity Name Validation**

In this figure 4.1.2, when user clicked the Add Activity button but edit text of activity name is blank, this error message will appear.



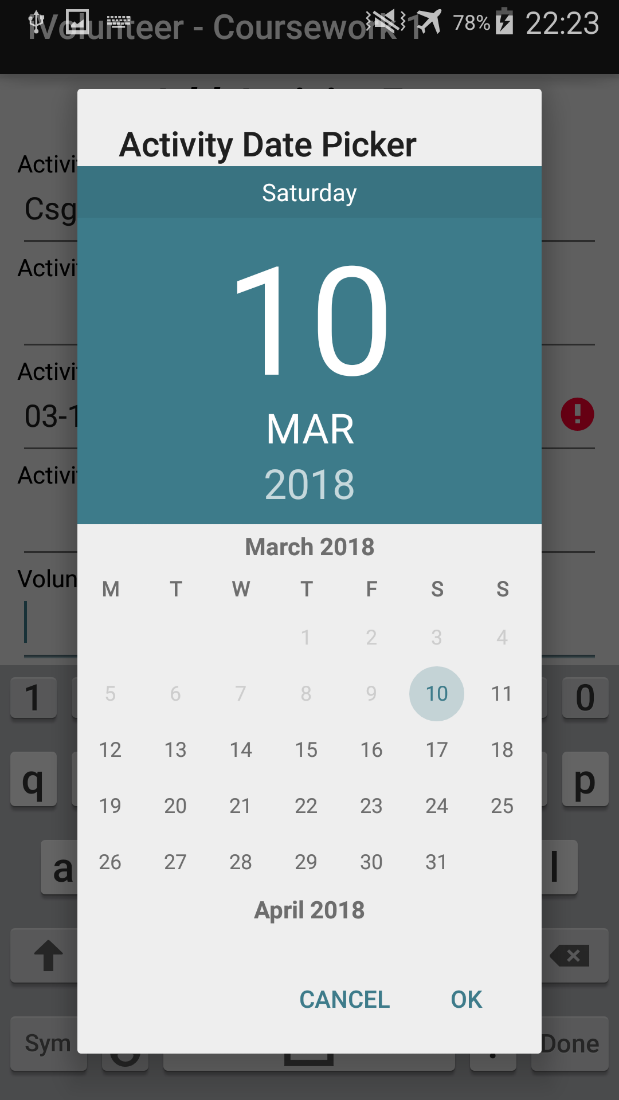
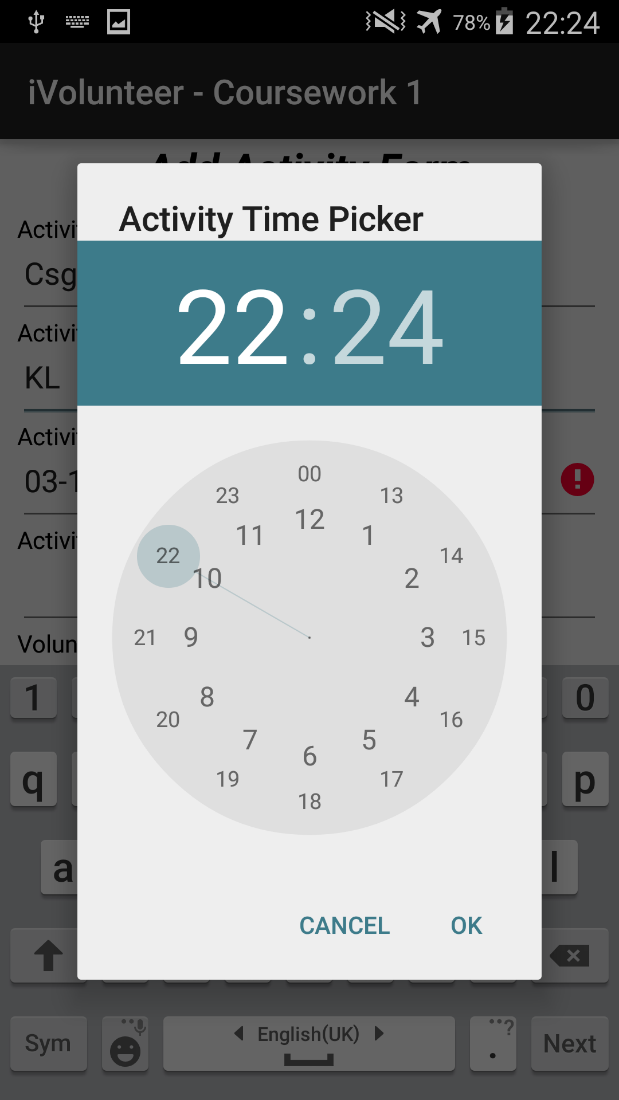
**Figure 4.1.3 – Date Validation**

In this figure 4.1.3, when user clicked the Add Activity button but edit text of activity date is blank, this error message will appear.



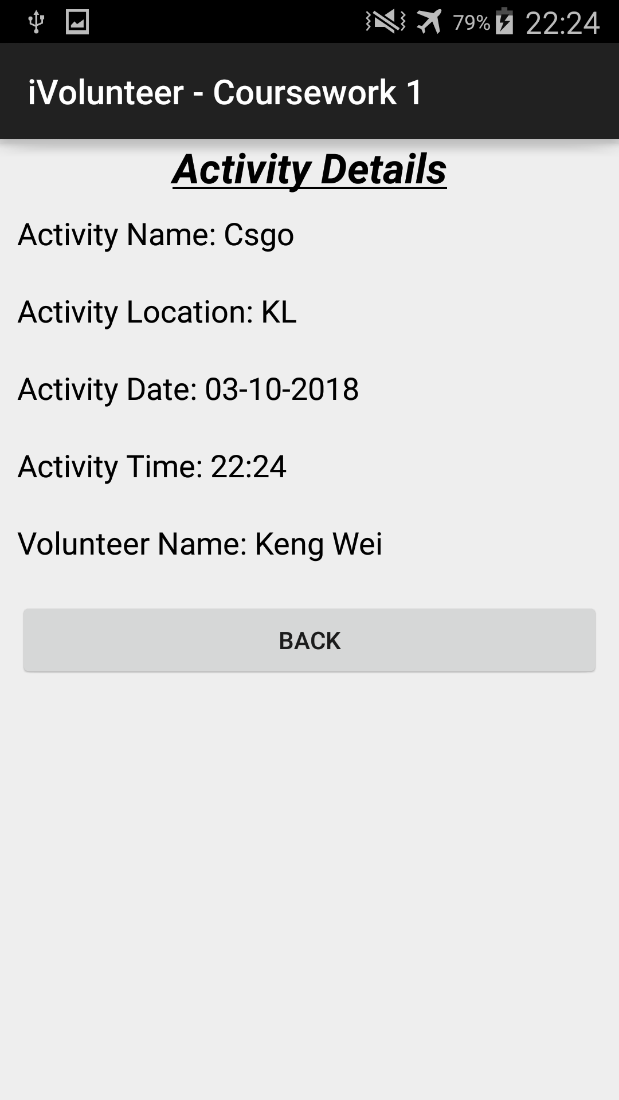
**Figure 4.1.4 – Volunteer Name Validation**

In this figure 4.1.4, when user clicked the Add Activity button but edit text of volunteer name is blank, this error message will appear.

**Figure 4.1.5 – Activity Date Picker Figure 4.1.6 – Activity Time Picker**

1. In this figure 4.1.5, when user clicked the edit text of activity date, it will pop up this Activity Date Picker. Furthermore, the date dialog picker only displays for past 3 months and next 3 months.
2. In this figure 4.1.6, when user clicked the edit text of activity time, it will pop up this Activity Time Picker.

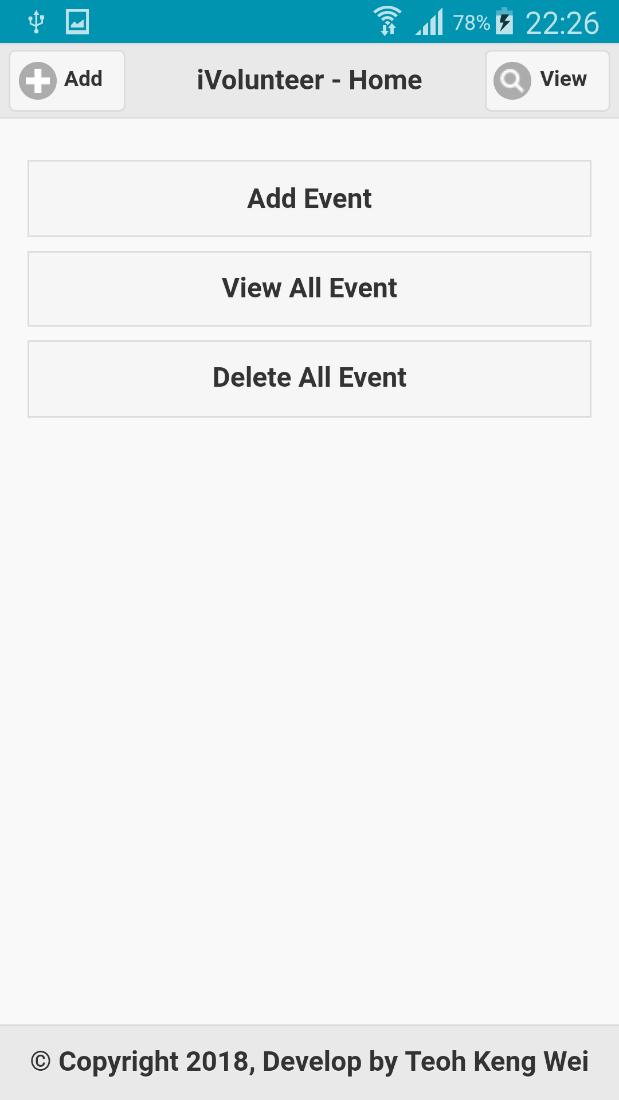


**Figure 4.1.7 - Activity Details Screen**

Once add activity is submitted, the function is triggered and it will start the intent activity which mean it called the app to go second activity called ‘Activity Details’ as well as the input value in order to display it in second activity. For the back button, when user clicked it. It will start the intent activity back to main activity as well as the value.

**4.2 PhoneGap iVolunteer – A to E Question**

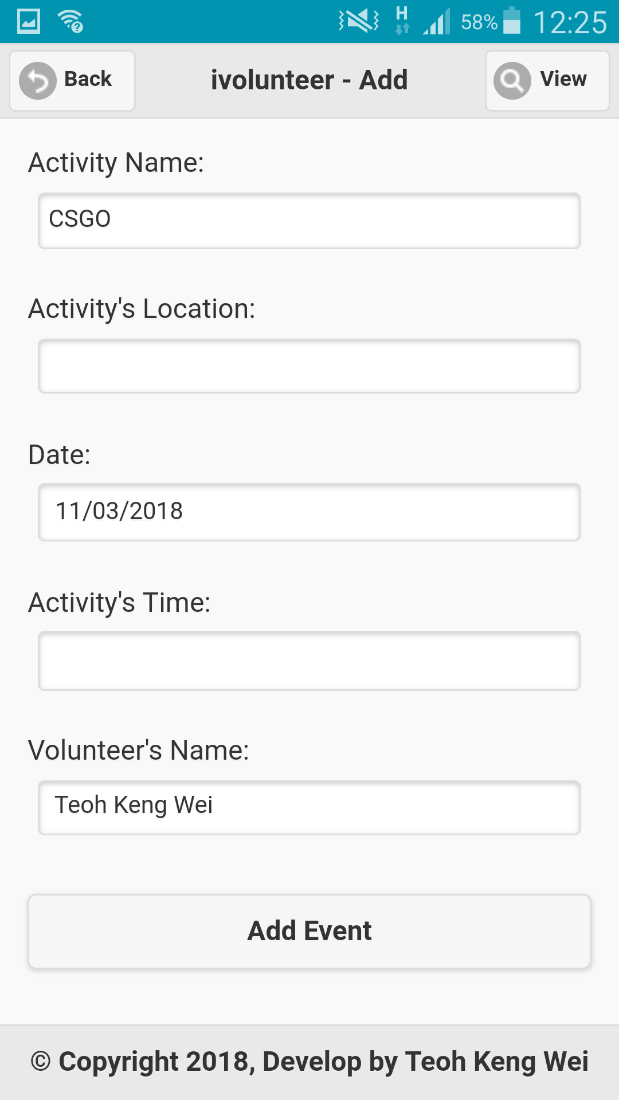
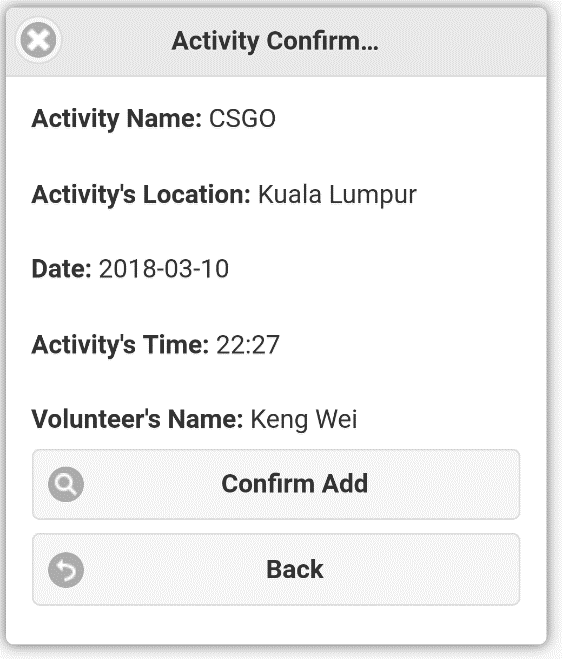
4.2.1 Home Screen



**Figure 4.2.1 – Home Screen**

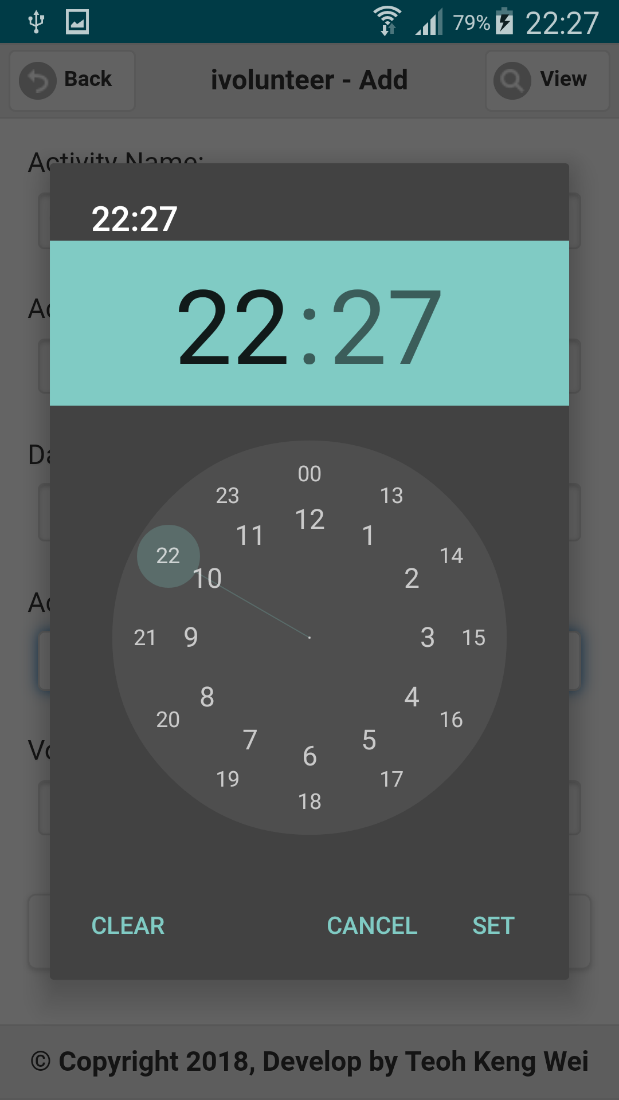
First of all, when user open this iVolunteer application, will see this home screen first. In the first screen it contains 3 buttons which are ‘Add Event’, ‘View All Event’ and ‘Delete All Event’. Other than that, there are also 2 buttons on the header which are ‘Add’ and ‘View’. User is able to click for each button to trigger for each specific function.

4.2.2 Add Activity Screen

**Figure 4.2.2 – Add Screen Figure 4.2.3 Activity Confirmation**

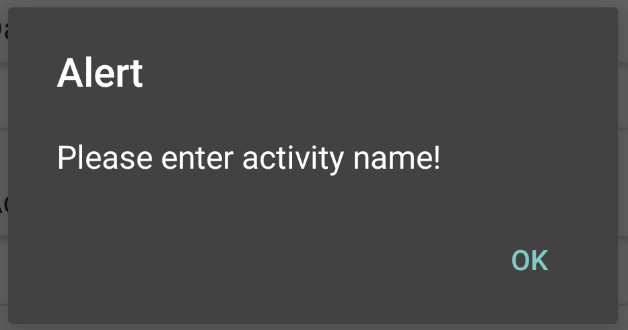
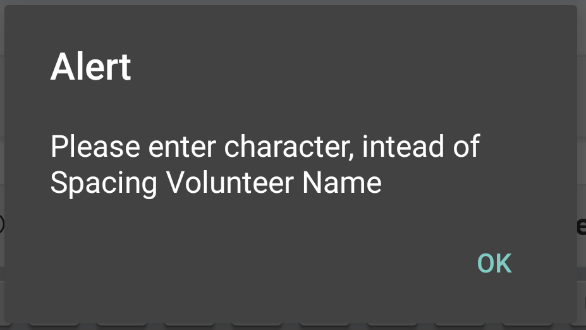
1. In this Figure 4.2.2, when user tapped button of Add Event and it will trigger the validation function first.
2. In figure 4.2.3, if it passed all the validation, it will pop up activity confirmation box. The confirmation box contains the description of input and 2 buttons.
3. If tap Confirm Add button, it will store in SQLite Database Table.
4. If tap Back button, it will return back to Add Screen.

**Figure 4.2.4 – Date Picker Dialog Figure 4.2.5 – Time Picker Dialog**

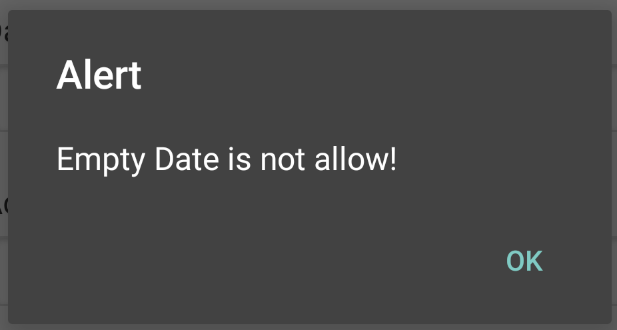
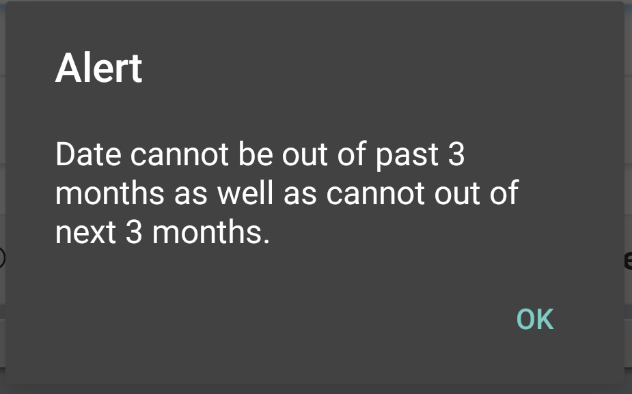
In figure 4.2.4, when user tap textbox of activity date, it will pop up the Date Picker Dialog. User can select their preferred date in this dialog. Furthermore, the date dialog picker only displays for past 3 months and next 3 months.

In figure 4.2.5, when user tap textbox of activity time, it will pop up the Time Picker Dialog. User can select their preferred time in this dialog.

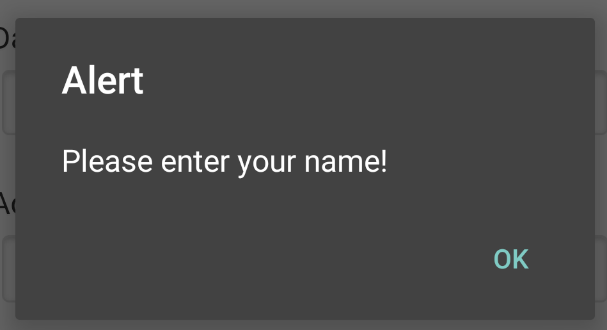
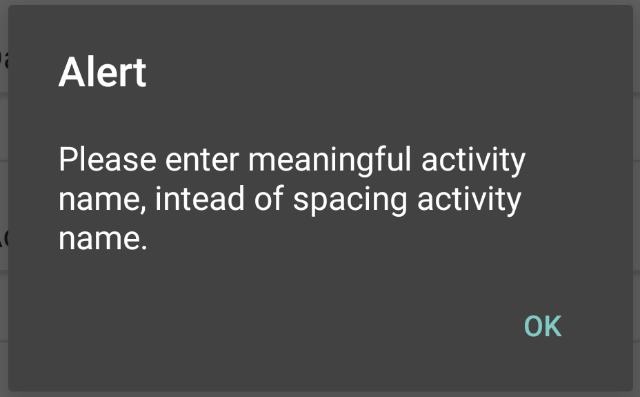
**Figure 4.2.6 – Activity Name Validation Error Message**

1. When user tap button of Add Event, but the activity name is blank it will appear the error message.
2. When user tap button of Add Event, but input spacing considered as input then will appear the error message.

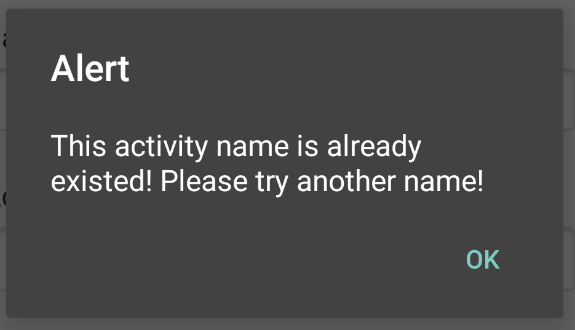
**Figure 4.2.7 – Date Validation Error Message**

1. When user tap button of Add Event, but the activity date is blank it will pop up the error message.
2. When user tap button of Add Event, but selected a date that is out of range, which are more than past 3 months and more than next 3 months. Then the error message box will pop up.

**Figure 4.2.8 – Time Validation Error Message**

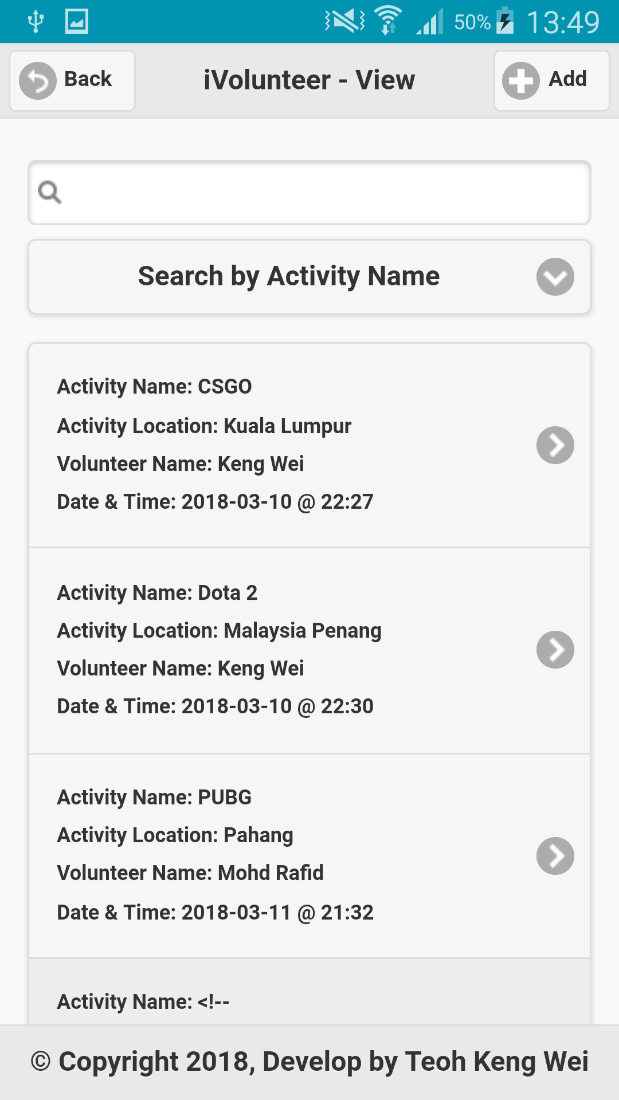
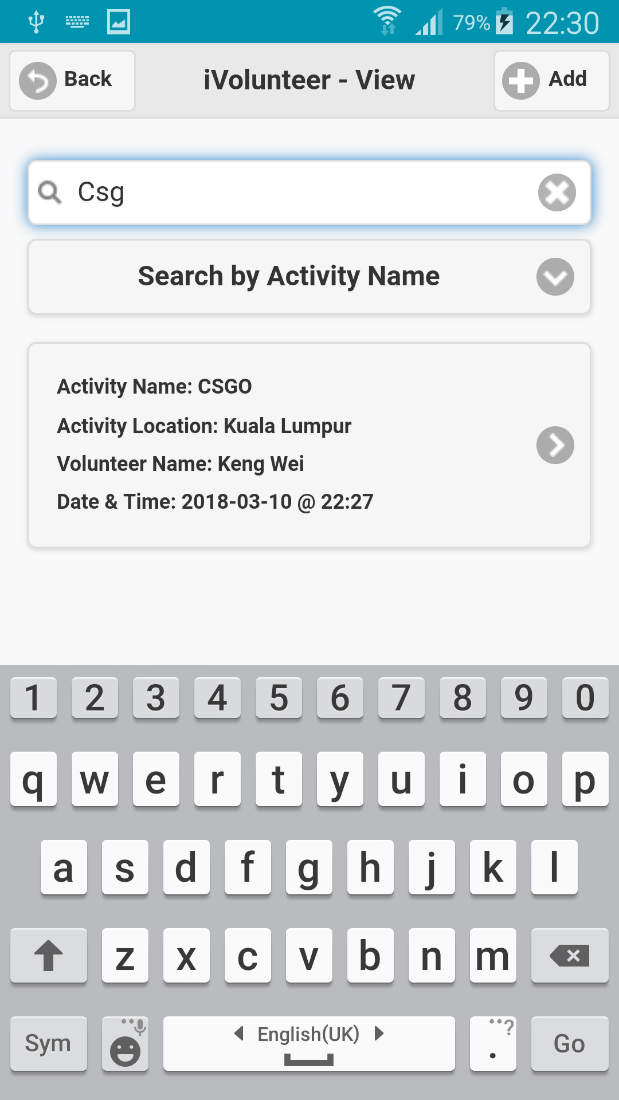
1. When user tap button of Add Event, but volunteer’s name input is blank then appear the first error message
2. When user tap button of Add Event, but input spacing considered as input then appear the second error message.



**Figure 4.2.9 – Activity Name Duplicate Validation Error Message**

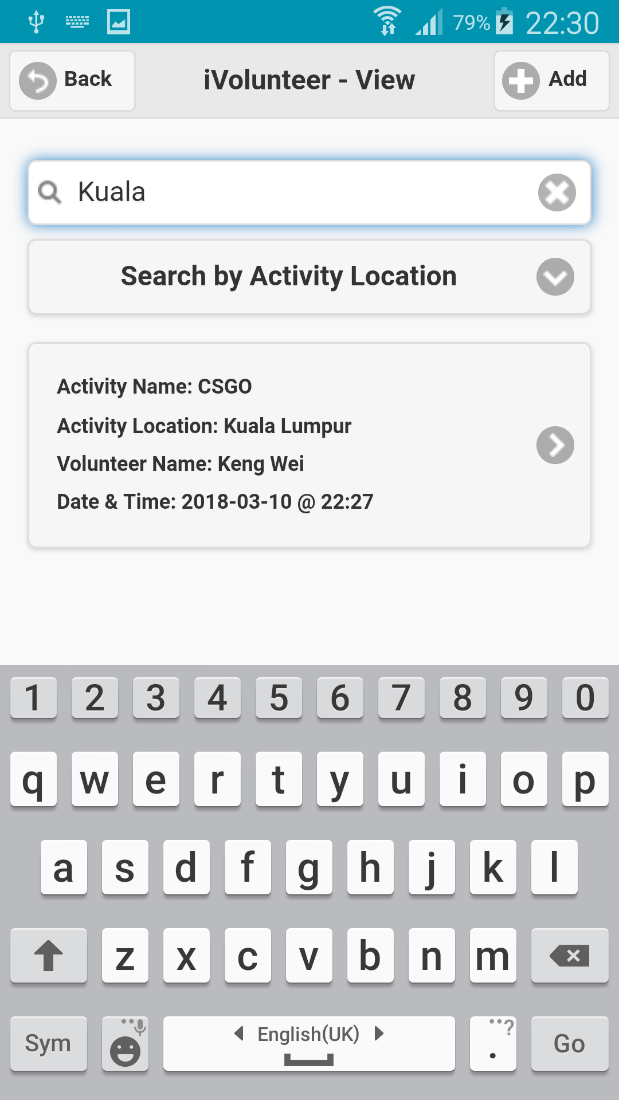
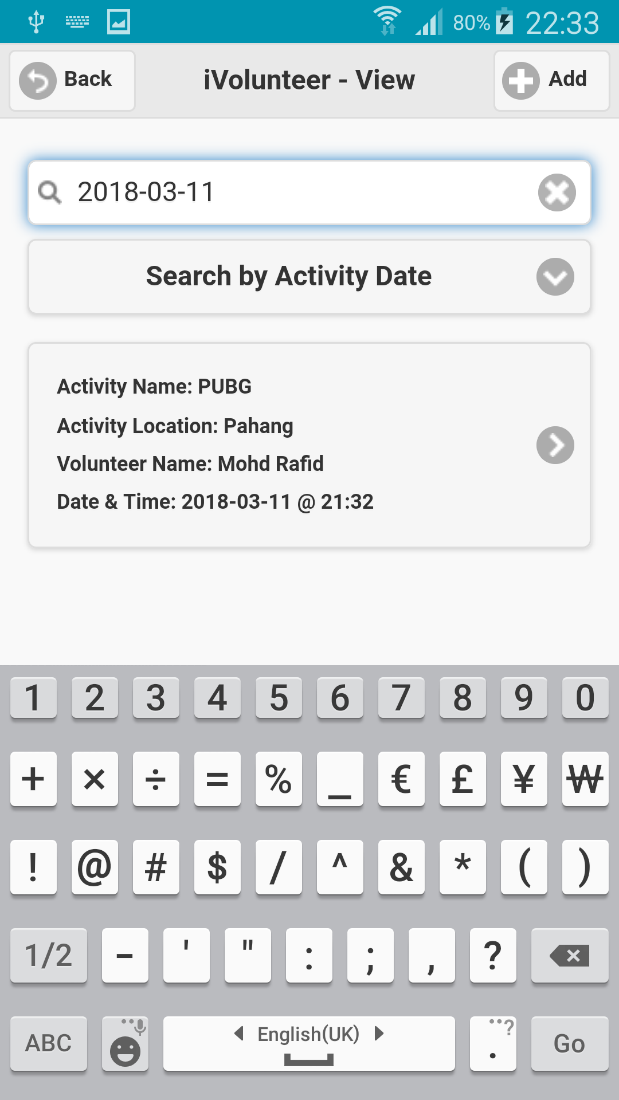
When user tap button of Add Event, but input the existed activity name, then this error message will pop up.

**4.2.3 View Activity & Search Screen**

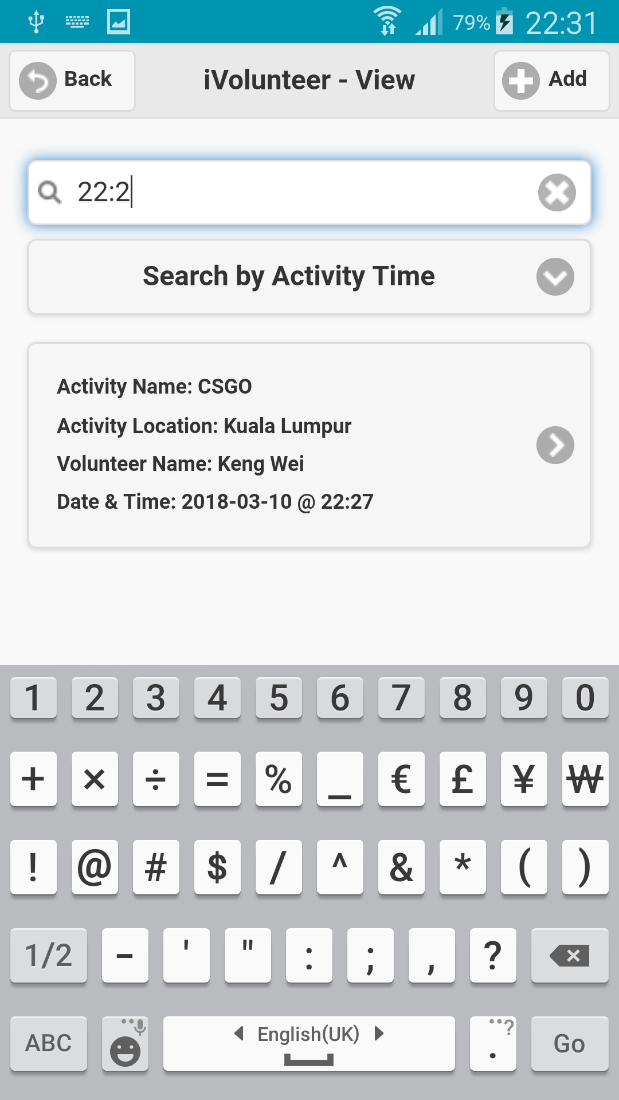
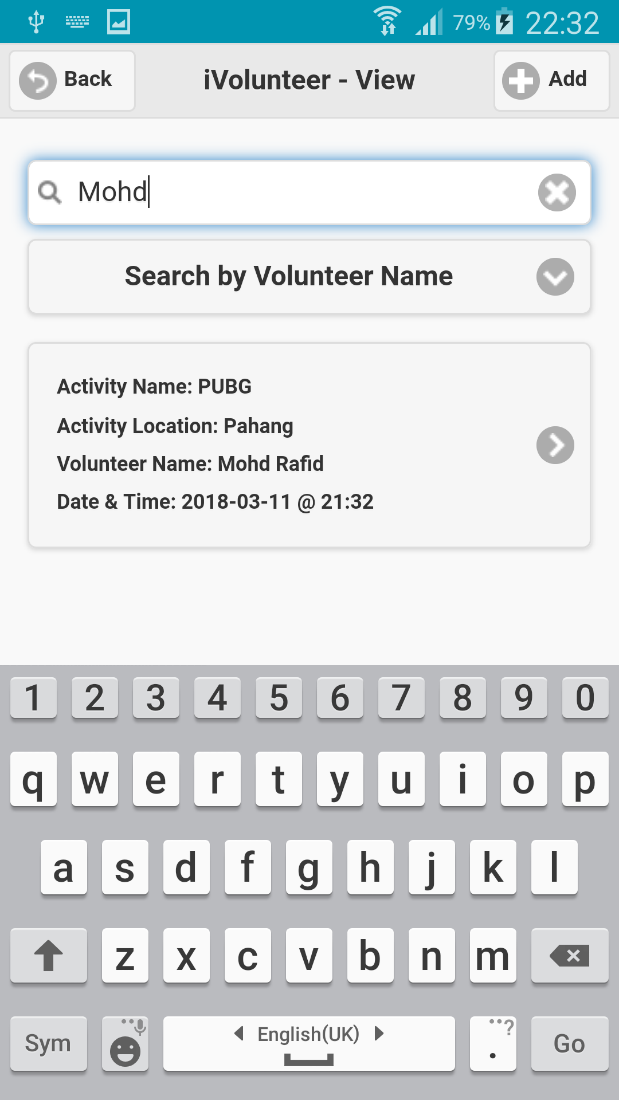
**Figure 4.3.0 – Advanced Search Function Screen & Search by Activity Name**

1. This is the screen which provide for user to view and search for list of activity. The author has created a function which let user to search in advanced, which mean user can search by activity name, activity location, activity date, activity time and volunteer’s name.
2. Select the ‘Search by Activity Name’ from the drop down list option.
3. In the textbox of search, input the activity name as ‘Csg’.
4. If there is any activity name that similar to ‘Csg’ then it will appear in the list view.

**Figure 4.3.1 – Search by Activity Location & Date**

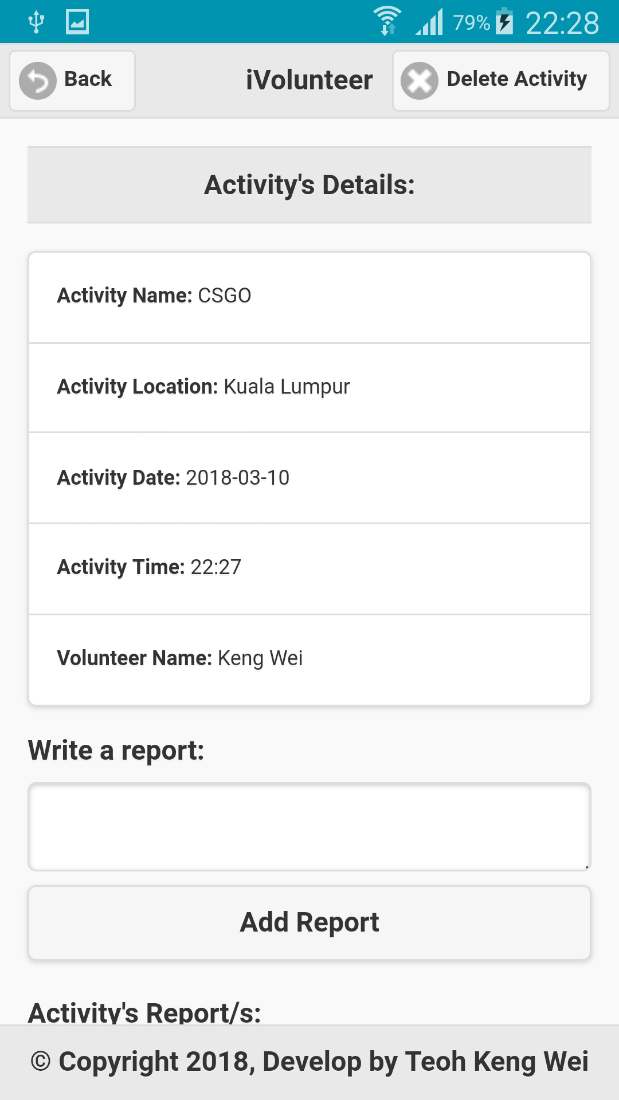
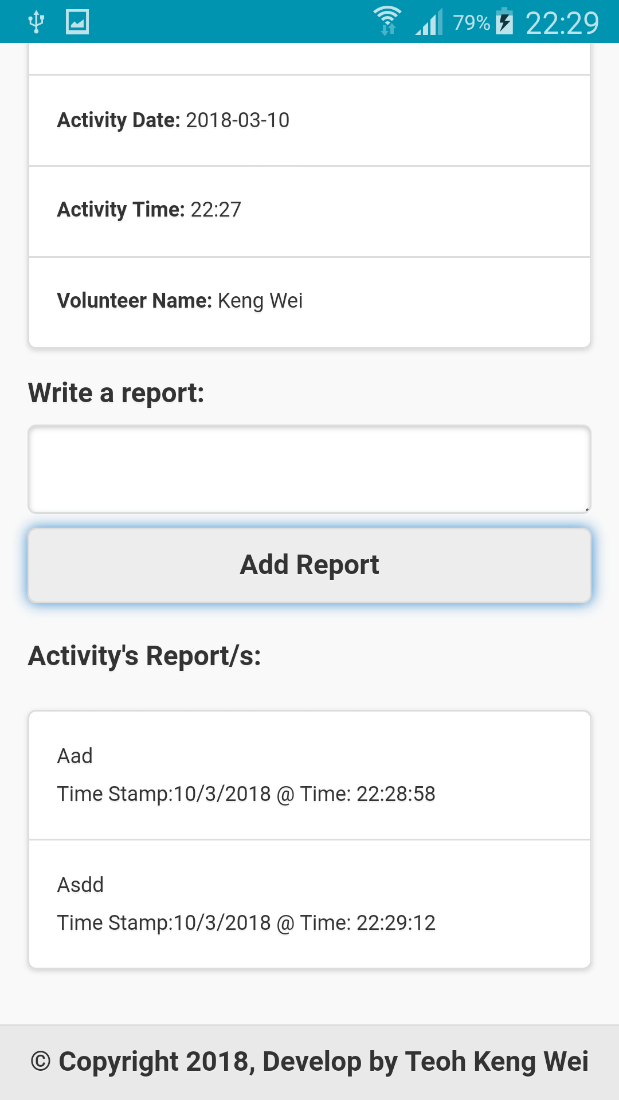
1. Select the ‘Search by Activity Location’ from the drop down list option.
2. In the textbox of search, input the activity location as ‘Kuala’.
3. If there is any activity location that similar to ‘Kuala’ then it will appear in the list view.
4. Select the ‘Search by Activity Date’ from the drop down list option.
5. In the textbox of search, input the activity date as ‘2018-03-11’.
6. If there is any activity date that similar to ‘2018-03-11’ then it will appear in the list view.

**Figure 4.3.2 – Search by Activity Time & Volunteer Name**

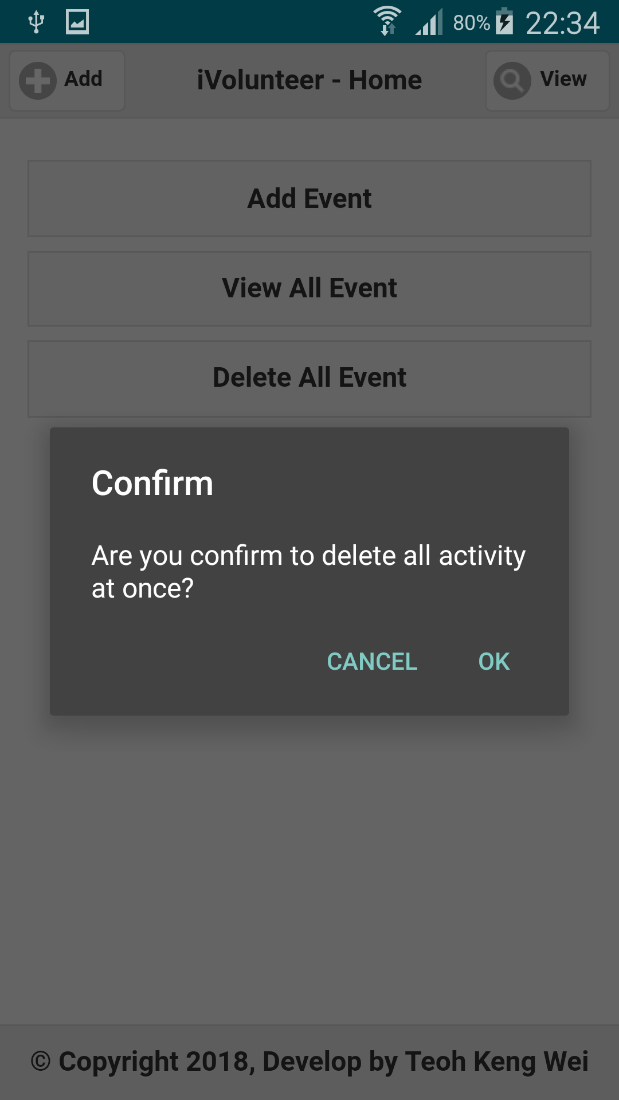
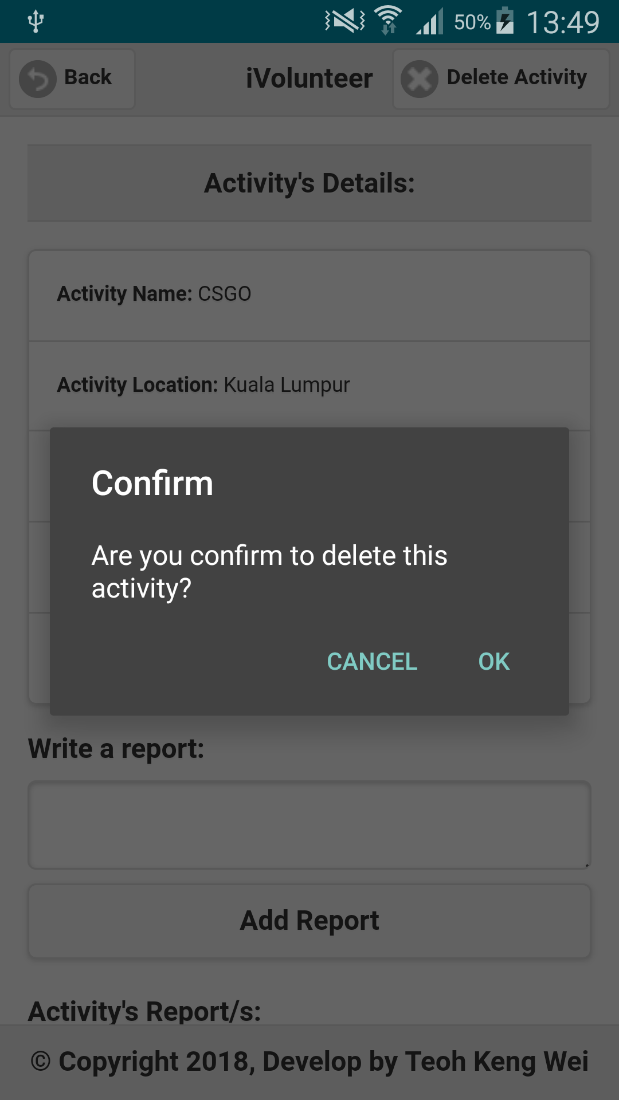
1. Select the ‘Search by Activity Time’ from the drop down list option.
2. In the textbox of search, input the activity time as ‘22:2’.
3. If there is any activity time that similar to ‘22:2’ then it will appear in the list view.
4. Select the ‘Search by Volunteer Name’ from the drop down list option.
5. In the textbox of search, input the volunteer name as ‘Mohd’.
6. If there is any volunteer name that similar to ‘Mohd’ then it will appear in the list view.

**4.2.4 Activity Details Screen**

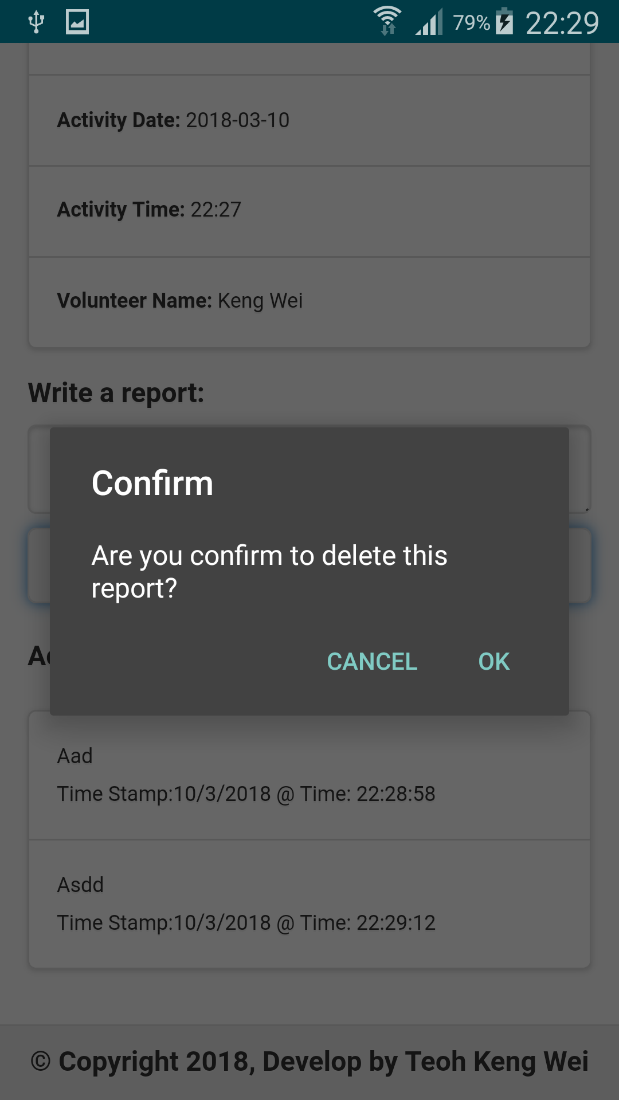
**Figure 4.3.3 – Activity Details Screen & Add Report Section**

1. This is Activity’s Details screen, which mean user is available to select a preferred activity and view in details. User is able to delete for that specific activity or add a report for that activity.
2. User can input the report description in this text area.
3. Once the input the description input in the text area, user can tap this Add Report button in order to store the report description in SQLite Database table.
4. After successfully inserted the report, it will refresh the screen and show it in list view of report.

**Figure 4.3.4 – Delete all event function & Delete specific activity function**

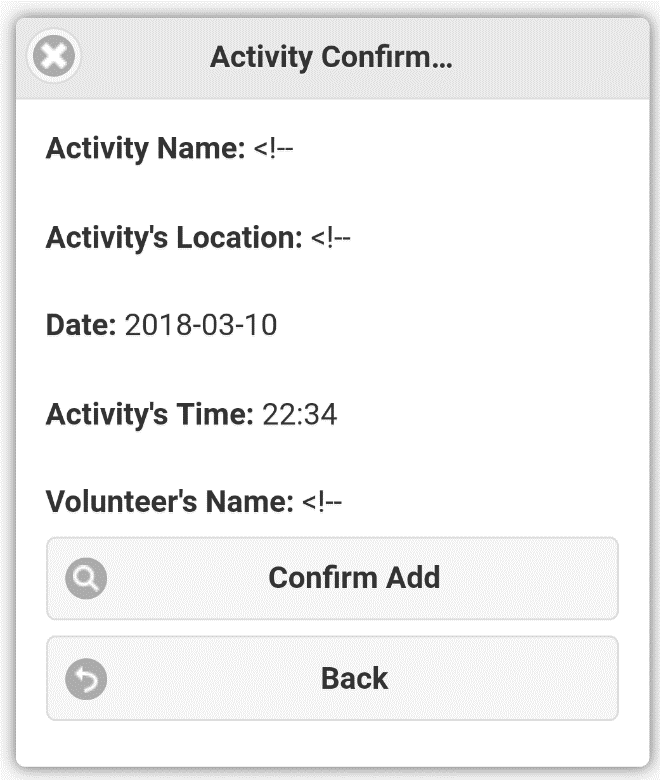
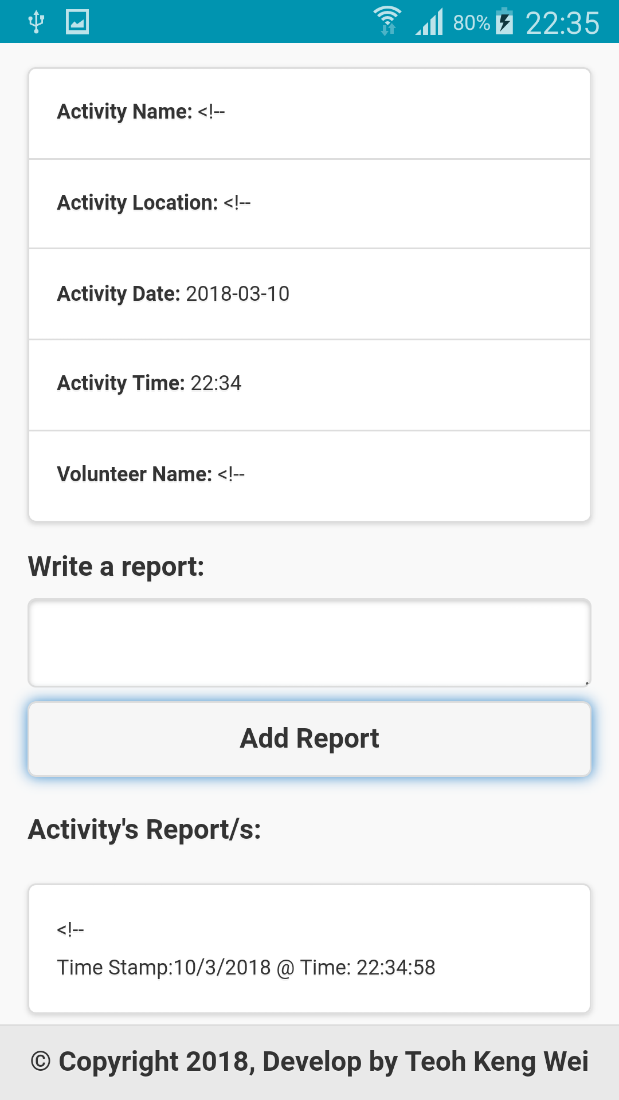
1. To delete all the event at once. Tap the Delete All Event button.
2. Once tapped Delete All Event button, it will pop up confirmation box with cancel and OK button.
3. If user clicked OK button, it will trigger the delete function to delete all (\*) from the activity table.
4. To delete specific activity, user can tap Delete Activity button.
5. Once tapped Delete Activity button, it will pop up confirmation box with cancel and OK button
6. If user clicked OK button, it will trigger the delete specific function to delete for that particular activity recorded.



**Figure 4.3.5 – Delete for specific report**

1. To delete for particular report, just tap for a report that are preferred.
2. Once tapped the report, it will pop up confirmation box with cancel and OK button.
3. If user clicked OK button, it will trigger the delete report function to delete for that particular report recorded.

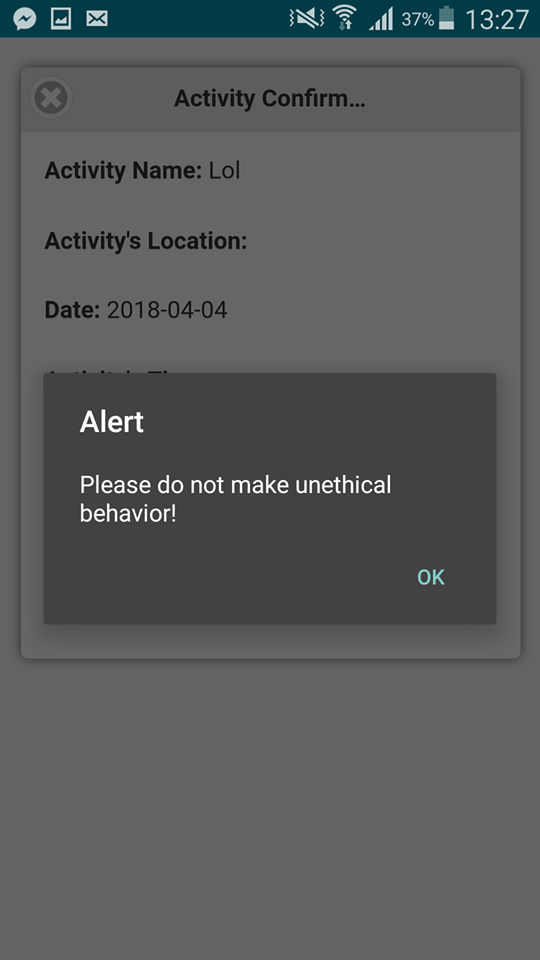
**4.2.5 Verify the HTML Entities and Security Injection**



**Figure 4.3.6 – Bypass HTML Entities**

In this section, the author will demonstrate about how user can bypass the html entities. Normally, when user try to insert into database with input html entities like this ‘<!--’, it is unable to pass the value to database. Therefore, the author will implement a function which encode the html entities. So that, the system can read it and insert into database.

**4.2.6 Avoid user spam adding empty value into database**



**Figure 4.3.7 – Spam Empty Value into Database table**

This error message will be pop up, while user keep spamming add empty value into database by using back button from their real device, redirect the screen into confirmation box for trigger the add activity function.

**Section 5.0 – Self & Product Evaluation**

**Human Computer Interaction**

First of all, I will start to design all the screens by using html with JQuery mobile CSS. The JQuery mobile CSS is a front-end library for designing html and that is why I’m using this front-end library to design the html to make it looks more simple and tidy. The user-interface design of this application is all meet the requirement of mobile app design. For instance, the proper design of button, list-view, textbox and others are all fit to the size of the mobile application layout. Therefore, no matter what kind of mobile phone with different size that users use, they still can use iVolunteer application comfortably and conveniently. Furthermore, I had tested to use this application with different type of mobile view, for instance, Samsung Note 3, Galaxy S4 and iPhone 8. As the result, this iVolunteer application is suitable for all mobile responsive designs.

**Security**

Other than that, this application still can bypass all the html entities by using the encode function to insert the data into the database and display it to list-view. For example, when user trying to input html entity like this ‘<!--’, it will causes some issues for insert and select query because of html unable to pass the html entities value to back-end. Therefore, I’ve implemented encode function to bypass it, so that it can also prevent from user to inserting html entities value as query injection attack for purposely. Furthermore, when user using back button from the real device (android phone) to add confirmation box, it will avoid from user doing unethical behavior, for instance, spam adding empty value into database.

**Maintainability**

For the back-end coding, all the function is reusable and each function structure will come with explanation in comments. For example, declareDate and htmlEntities function, both of these function can be reuse in future enhancement. Therefore, it can be very easy for my team project member to understand how is the code looks like and also easy for enhancement of this application in future.

**Changes that is required in order for the app(s) to be deployed for live use**

Other than that, there are still many best function will be update to be deployed for live use. Therefore, the author has implemented search with advanced function. It can be search by different way, such as search by activity name, search by location, search by date, search by time, and search by volunteer’s name. All these search function are designed and implemented with query to retrieve the data from database and display in list-view in order to accurately fetch the data to the user. The purpose of advanced search is to give user a convenient way in searching an activity with more details and more specific. Moreover, the author will be update for new function, for instance, upload picture by camera for particular activity and share the activity details to any social medias.