APPLICATION DEVELOPMENT FOR MOBILEDEVICES

Bui Huy Hoang

Contents

Introduction

This report is about **Application Development for Mobile Devices**, it will reflect the knowledge necessary of writer's personal skills and professional development by discussing key issues for Application Development for Mobile Devices. The writer will also be able to design and implement effective solutions for given scenarios.

The task is to develop a mobile app for use by people who participate in volunteering activities. The app will allow user to enter, store and upload reports about the activities they are involved with. The app will be called **iVolunteer**. Description of the application and deliverables are noted in the coursework. The objectives of this report keep up with the learning outcomes and deliverables of the coursework.

This assignment is written in the essay style, although a formal essay structure will not be required. Tables, pictures, diagram, etc. which give sources of information will be considered to use. A list of references to all cited sources is included if provided. The writer uses font Calibri size 12 with 1.5 spacing.

Section 1. Checklist of features

This is a concise table containing a checklist of the features that have been able to be implemented.

Feature	Description	Implementation
a	Design app screens - Basic details input screen	Fully implemented
b	Implement forms validation and check for duplicate events	Fully implemented
c	Store, view and delete the basic details	Fully implemented
d	Search	Fully implemented
e	Add a report input screen	Fully implemented
f	Features a and b are to be implemented as a native Android app coded in Java	Fully implemented
g	Add additional features to either or both the Android or PhoneGap version of the app	One additional feature implemented

Section 2. Bugs and Weaknesses

A concise list of any bugs and/or weaknesses in the apps is given below:

In PhoneGap version

Database exceptions (on personal evaluation) are handled with code but not so efficient. As we can see, when it comes to happen error in database transaction, it will only show the error information in console log. This weakness should have a solution that to alert user and be able to automatically fix the transferring data process. But in my opinion, the app has been developed carefully, so the probability of database error is almost nonexistent.

```
loadVolunteerActivities: function(displayVolunteerActivities){
    dbHandler.db.readTransaction(
        function(tx){
            tx.executeSql(
                "select * from volunteerActivities",
                [],
                function(tx, results){
                    displayVolunteerActivities(results);
                },
                function(tx, error){
                    console.log("add volunteerActivities error: " + error.message);
                }
            );
        }
    );
},
```

Figure 1 - Code of catching exception

In Android version

The Android version follows the requirement that has one adding volunteer activity page. The page's screen contains two specialize date and time textfields that user has to insert numeric type with default format (dd/mm/yyyy, hh:mm). But it appears a problem that, if user accidentally inserts wrong format of date and time, the database still accepts the data and allow to store them in wrong format as string.

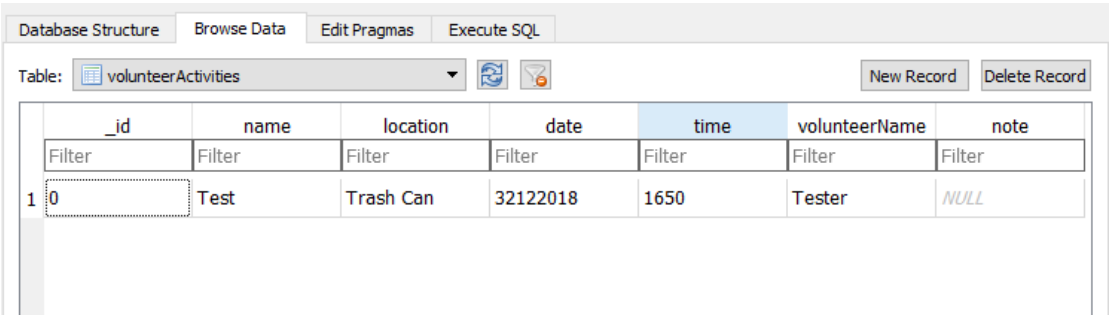


Figure 2 – Wrong format of input in Date and Time

Section 3. Strength

A brief description of any special strengths of the app(s).

Overall, every requirement of the coursework has been completed. Both versions have their interface built with simple look in order to increase system performance and easy to use or understand. Besides full implementation required in the coursework, the PhoneGap version has been included additional feature (Update volunteer activity) that allows user to modify the data of existed volunteer activities. The **Update function** contains validation that user must not leave activity name, date and volunteer name fields empty. Read the section 4.7 for further detail about this feature.

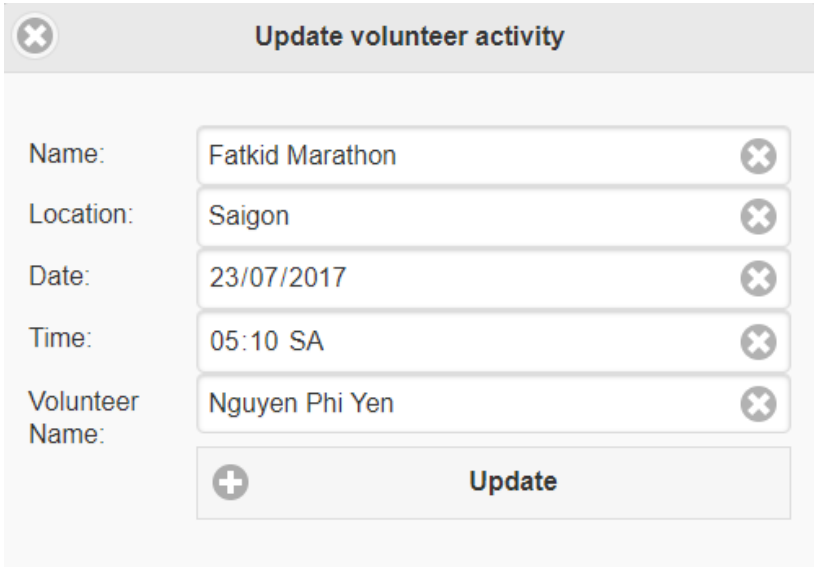


Figure 3 – Additional feature – Update a volunteer activity

Section 4. Screenshots

Screen shots demonstrating each of the features that have been implemented. Captions or annotations will be given to explain which features are being demonstrated.

4.1 PhoneGap: Basic details input screen

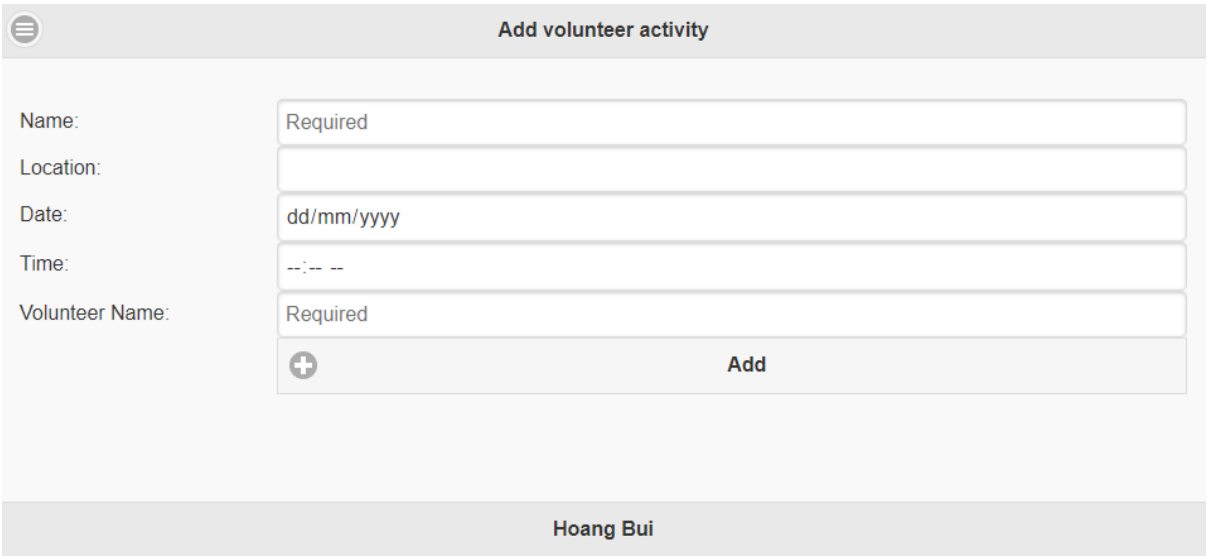


Figure 4 - Add volunteer activity page

This is the **Add volunteer activity page**, it contains 5 textfields (2 specialized inputs to insert date and time) and an add button.

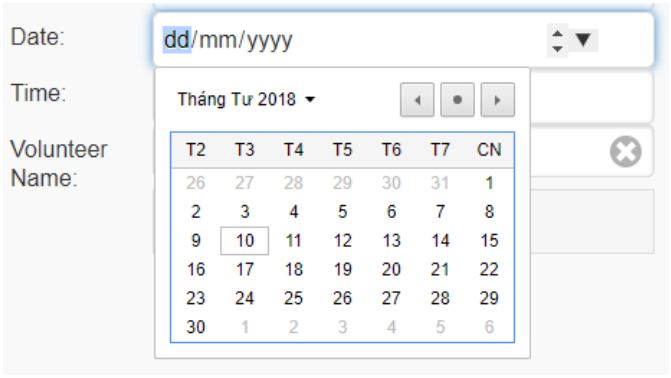


Figure 5 - Date picker

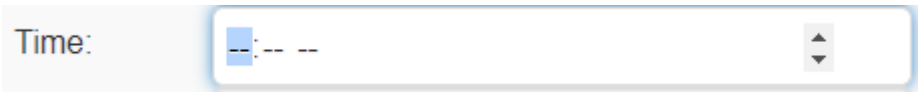


Figure 6 - Time insert

The add function includes several validations that are mentioned below.

4.2 PhoneGap: Implement forms validation

The app will check validation for the input, if user does not insert anything in one of the required fields (Name, Date and Volunteer Name), the app will display an error message to the user.

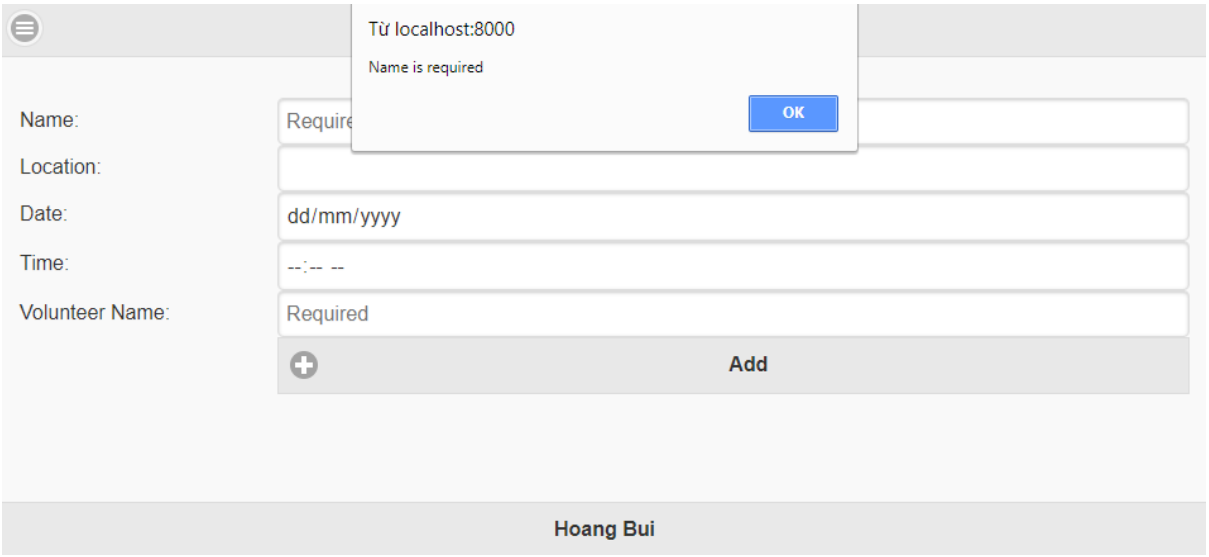


Figure 7 - The volunteer activity name textfield must not be left empty. The textfield has been set placeholder to notice user.

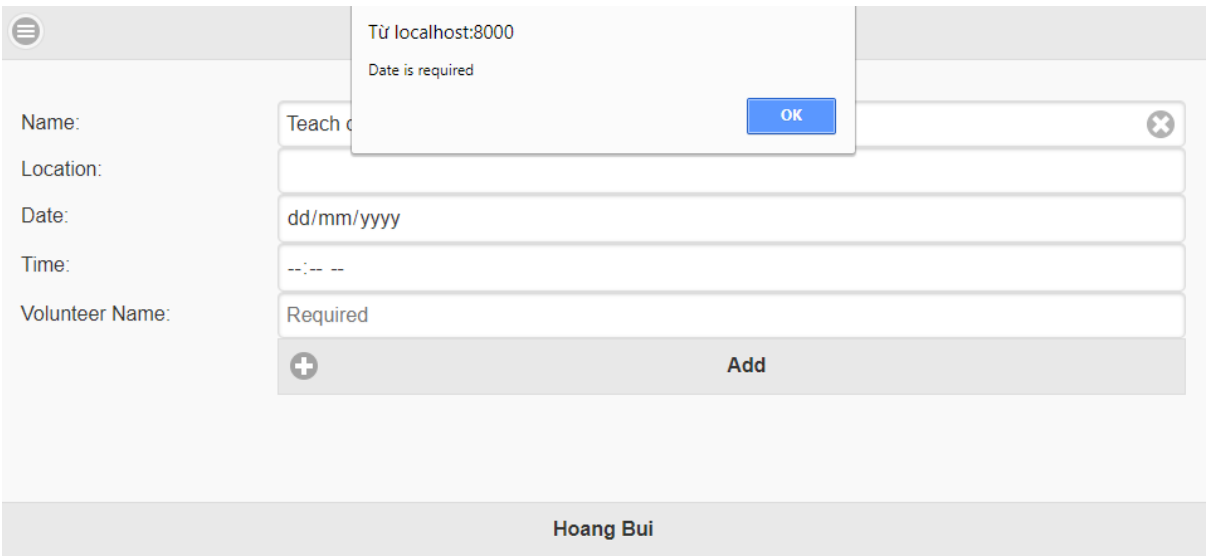


Figure 8 - The date textfield/picker must not be left empty.

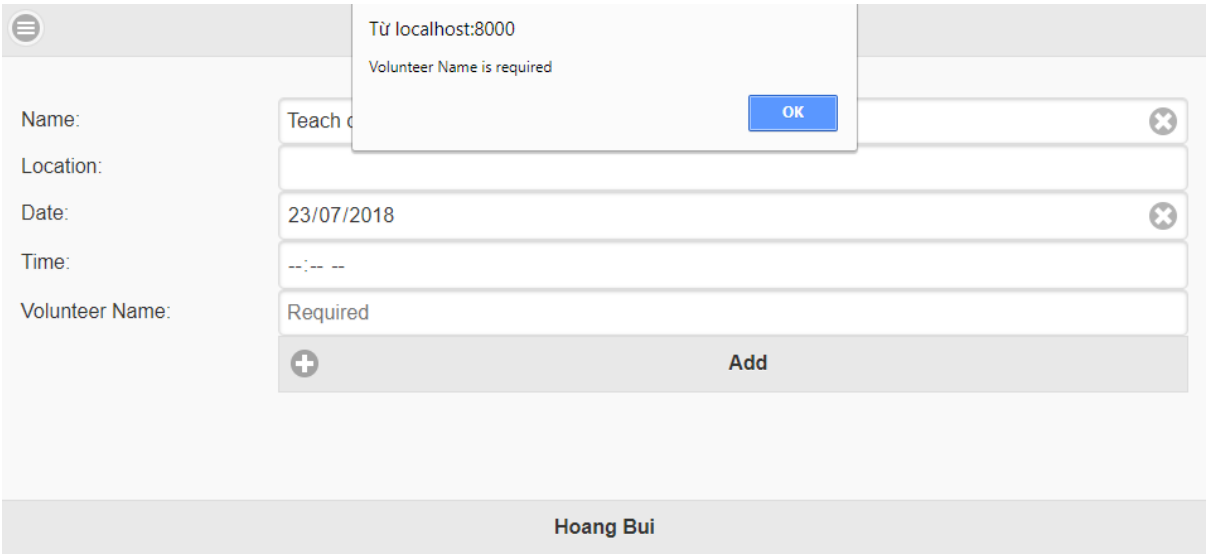


Figure 9 - The volunteer name textfield must not be left empty. The textfield has been set placeholder to notice user

Once the details have been accepted by the app (e.g. no required fields left empty) it will show a confirmation popup and allow them to go back and change what needed.

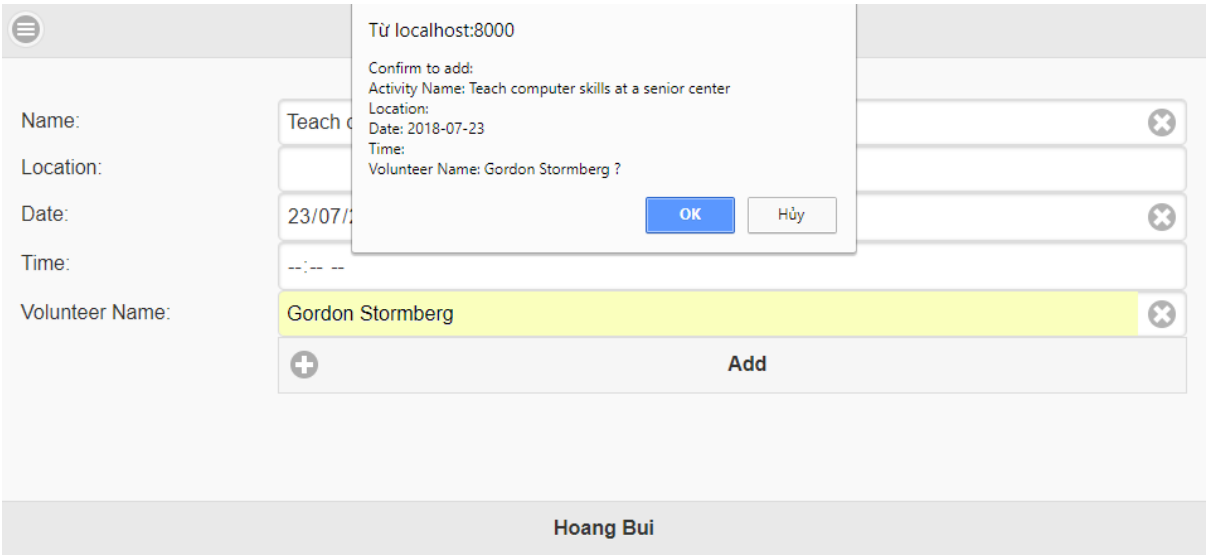


Figure 10 - User must enter all required input. It will show a confirmation popup before officially store in the database.

A notification dialog will appear with message “Added!” after user confirms the adding process.

Figure 11 - Successfully added data.

4.3 PhoneGap: Store, view and delete the basic details

To enter the **load volunteer activity page**, user will have to click the icon in the left top corner, a panel which navigates between pages will appear showing the link to the load page.

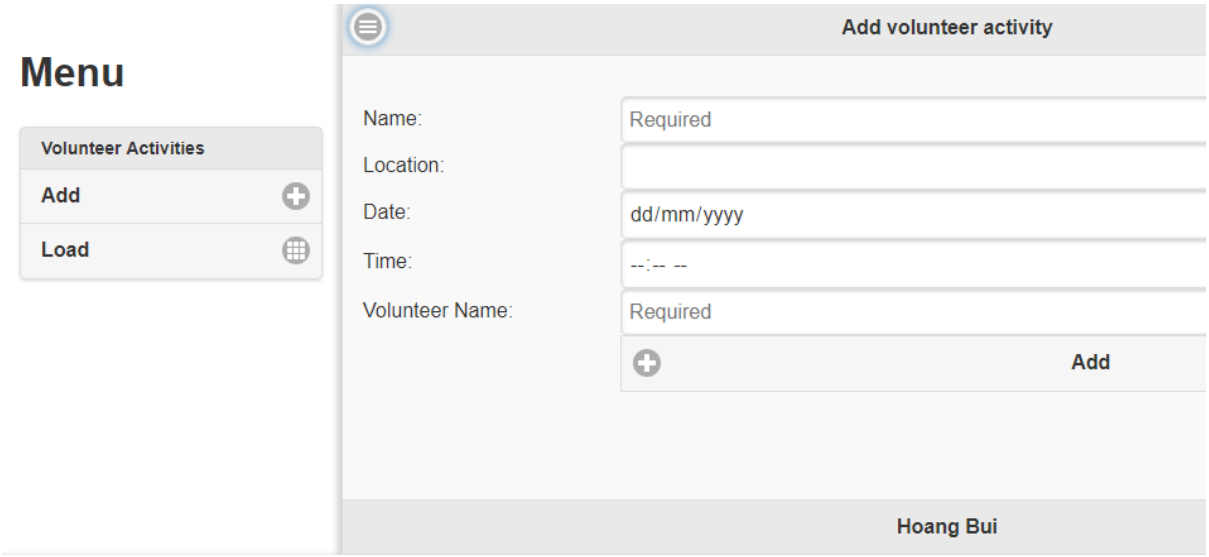


Figure 12 - Navigating panel

The load page contains a list of volunteer activities currently stored in the database with their detail information:

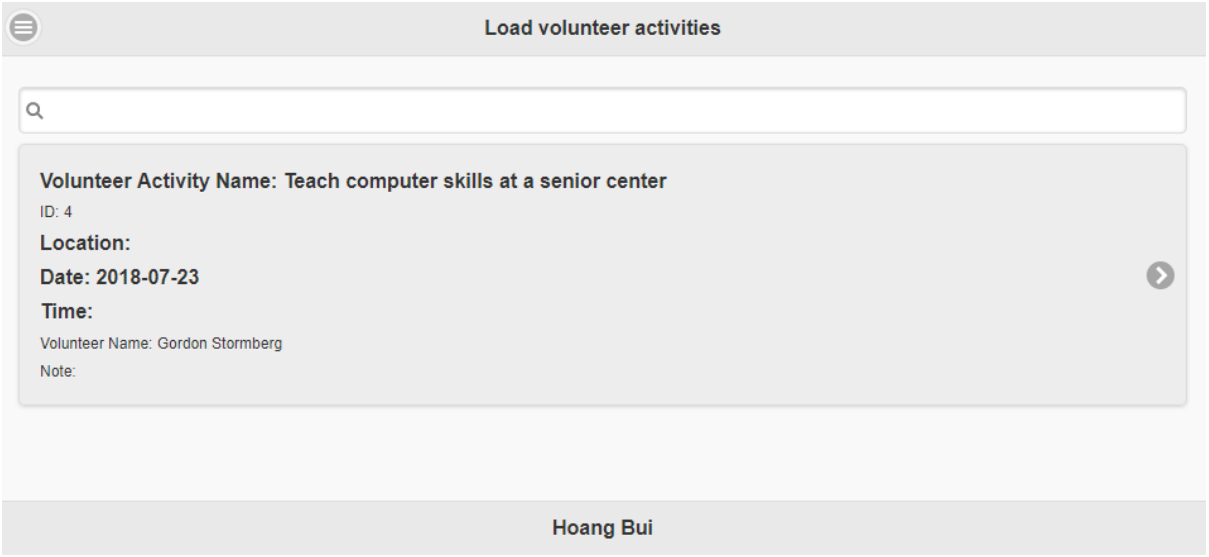


Figure 13 - Load volunteer activities page.

After choosing one activity item on the list, a popup will show three options, in this case we will choose **delete button.**

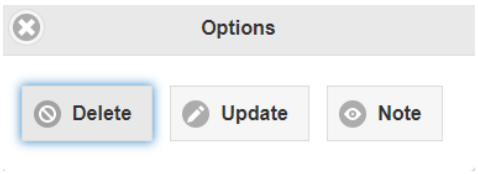


Figure 14 - Options when clicking one activity.

It will show a confirmation dialog before officially delete the volunteer activity:

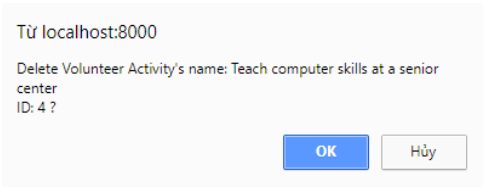


Figure 15 - A confirmation popup.

When the delete process is executed, the load page will reload.

.

4.4 PhoneGap: Check for duplicate events

In the **add volunteer activity page**, when we add a new activity, the process will check for duplicated data that currently existed in the database. For example, as we can see in the list table mentioned in section 4.3, there is an existed activity that if we insert the exact information of Location, Date and Time in the add page, the process will be terminated with an alert dialog appears:

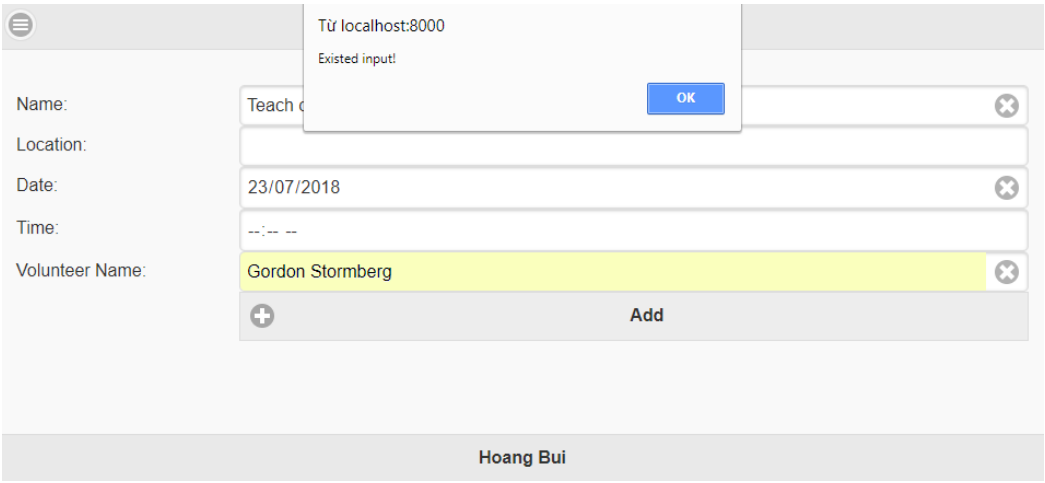


Figure 16 - Input the duplicated data.

4.5 PhoneGap: Search

The search function has been implemented in the load volunteer page. User will find activities by inserting its part of name.

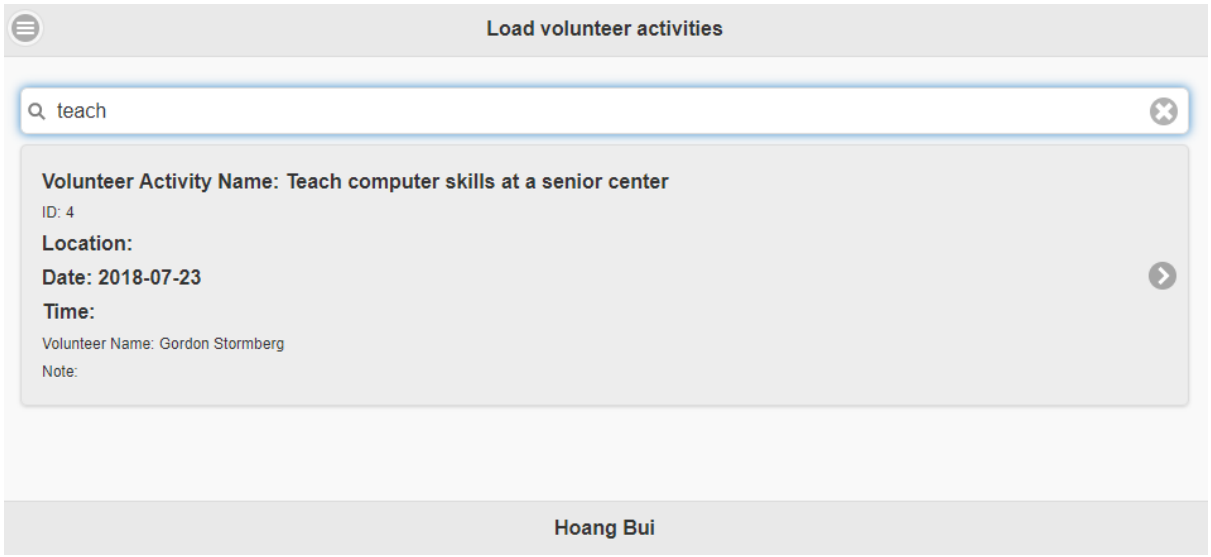


Figure 17 - Search volunteer activities.

4.6 PhoneGap: Add a report (note) input screen

Each volunteer activity has Note section:

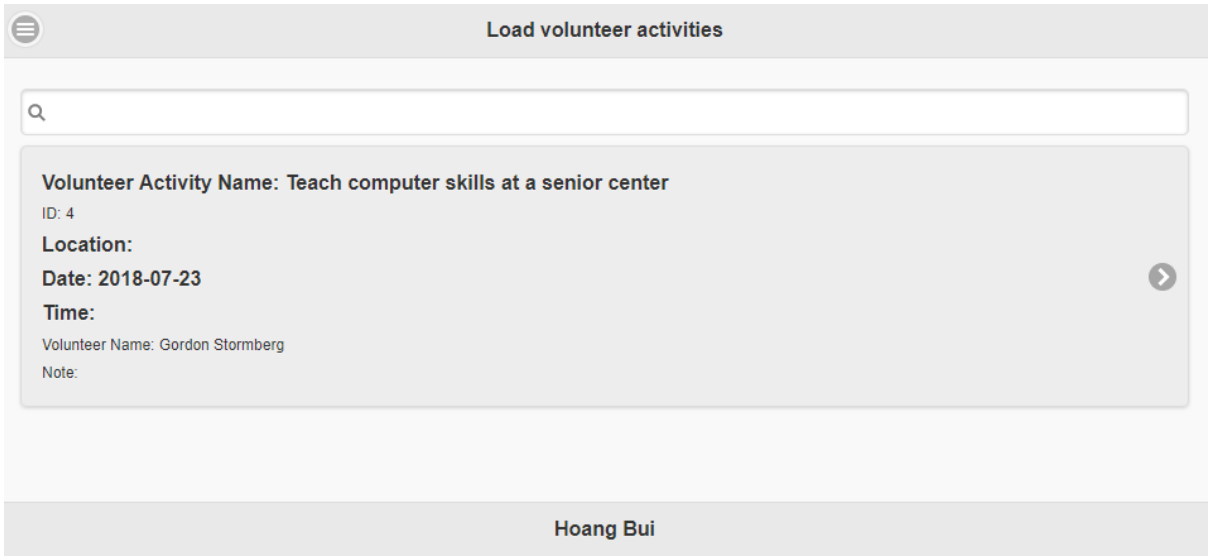


Figure 18 - An existed activity having empty note.

User can select one of the activities on the table in **load volunteer page** and select **note button** to enter the update dialog containing information about something that happens during that activity.

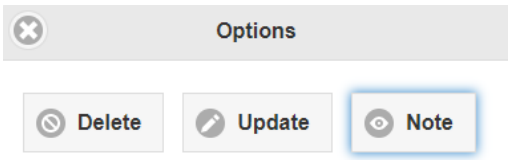


Figure 19 - Options when clicking one activity.

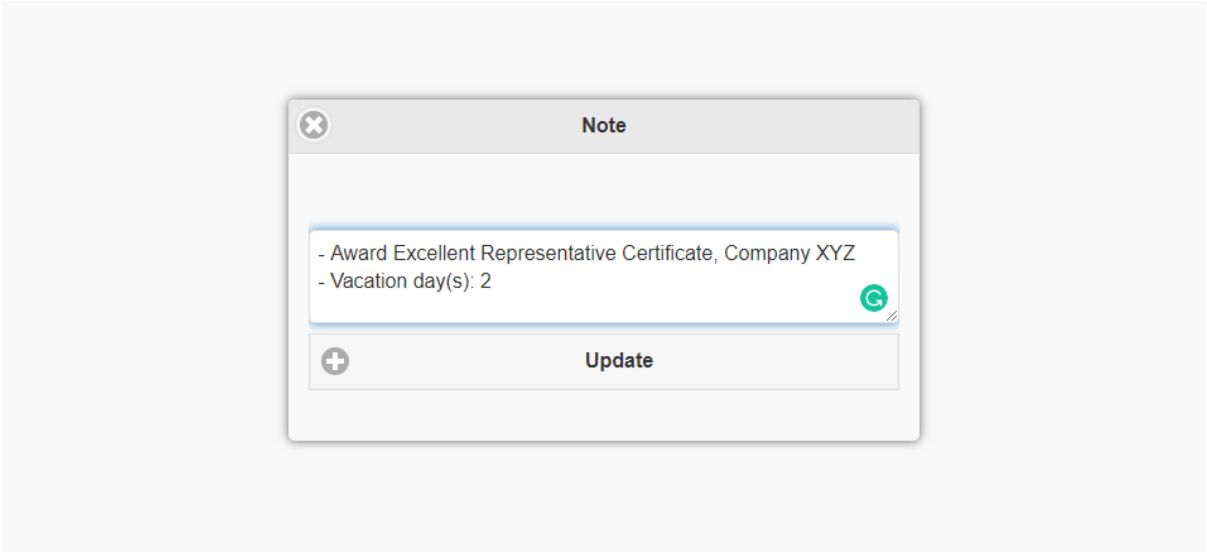


Figure 20 - Update note dialog.

The result will be illustrated after changing note's information:

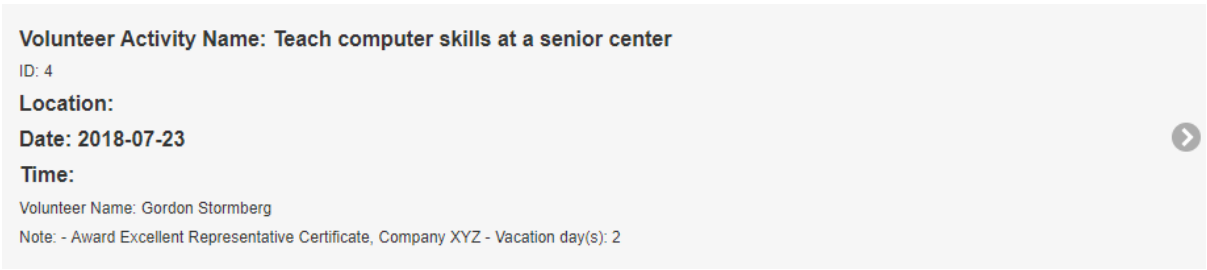


Figure 21 - An existed activity having updated note.

4.7 PhoneGap: Additional feature (Update volunteer activity)

An existed volunteer activity can be modified in PhoneGap version. User can select one of the activities on the table in **load volunteer page** and select Update:

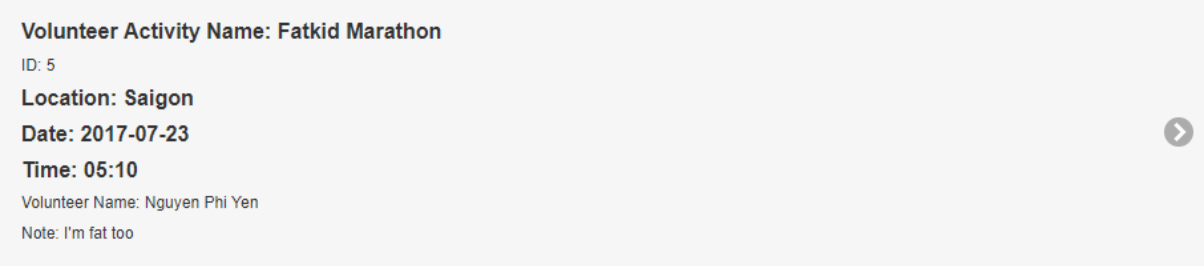


Figure 22 - Before updating activity.

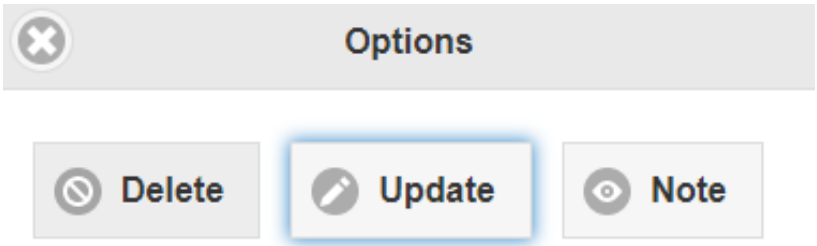


Figure 23 - Options when clicking one activity.

A box containing their current information will appear for user to modify. Validations (insert all required fields, check duplicate activity) are also implemented. After inserting new information, the app will ask user to confirm their decision.

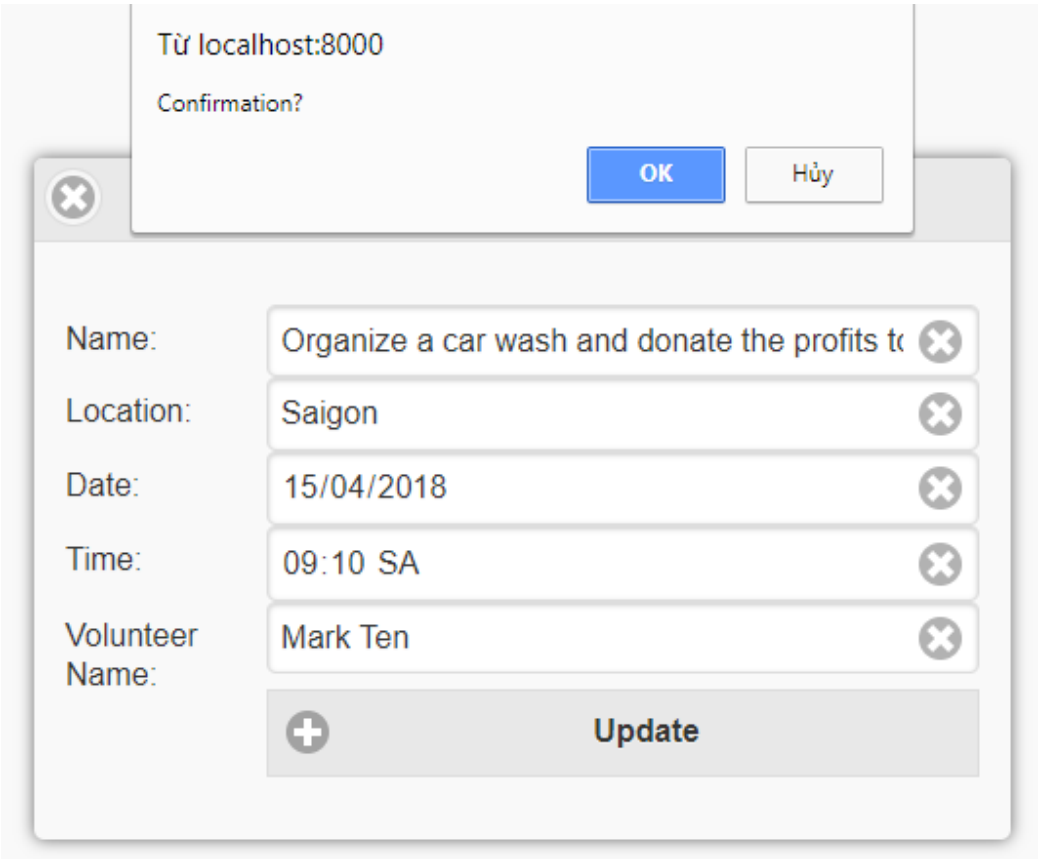


Figure 24 - A confirmation popup before updating an activity.

The result will be shown below:

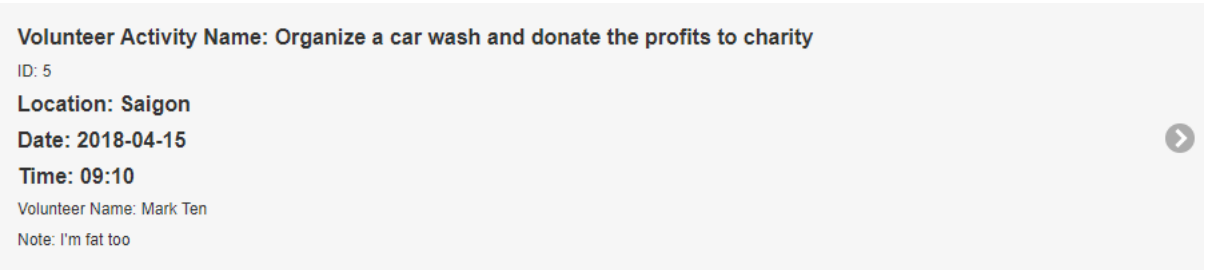
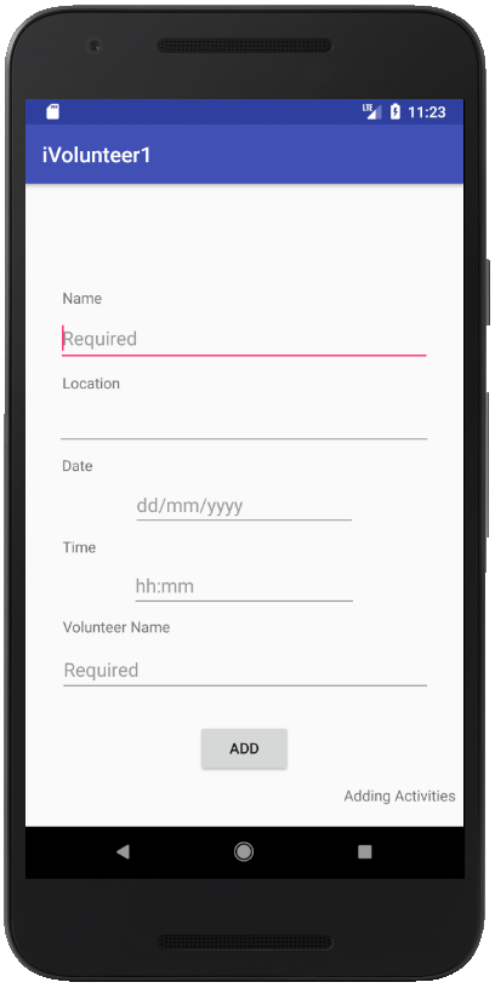


Figure 25 - Updated activity.

4.8 Android: Input screen



This is the **Add volunteer activity page**, it contains 5 textfields (2 specialized format textfields to insert date and time) and an Add button. The add function includes several validations that will be mentioned below.

Figure 26 - Add volunteer activity page.

4.9 Android: Implement forms validation

The app will check the input and if the user does not insert anything in one of the required fields (Name, Date and Volunteer Name) the app will display an error message to the user.

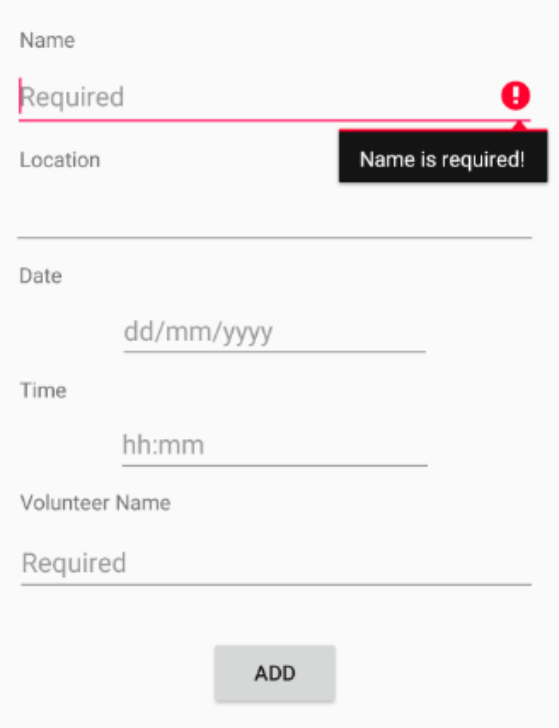


Figure 28 - The volunteer activity name textfield must not be left empty.

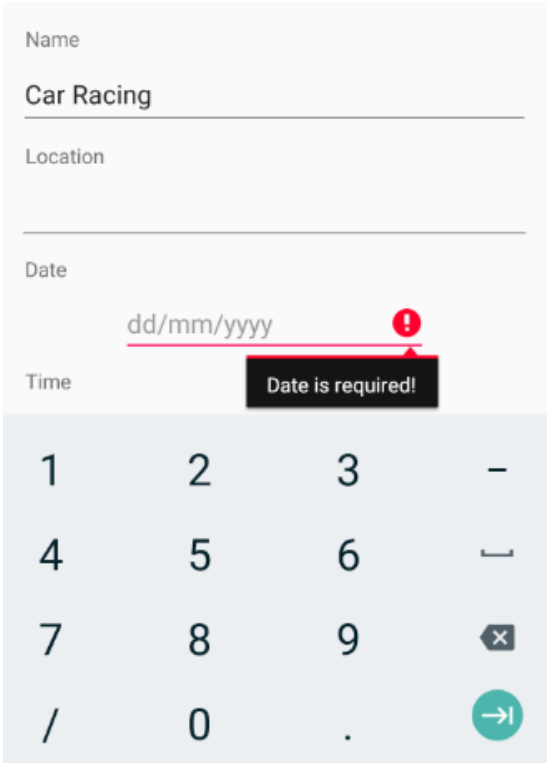


Figure 27 - The date textfield must not be left empty. As we can see, the input format is illustrated by numeric keyboard and specialized symbol "/". It can be understood that we have to insert with the format dd/mm/yyyy.

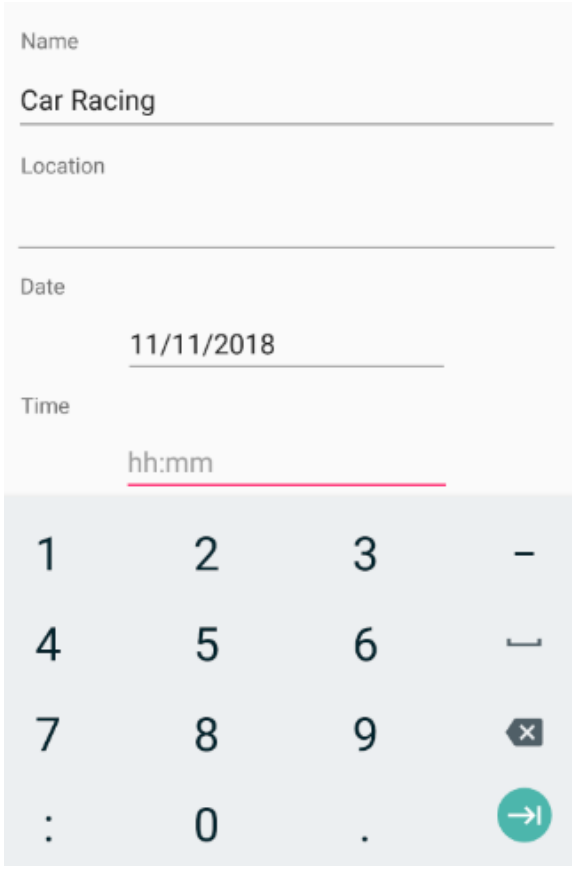


Figure 30 - The time textfield has the same feature that the input format is illustrated by numeric keyboard and specialized symbol ":". It can be understood that we have to insert with the format "hh:mm".

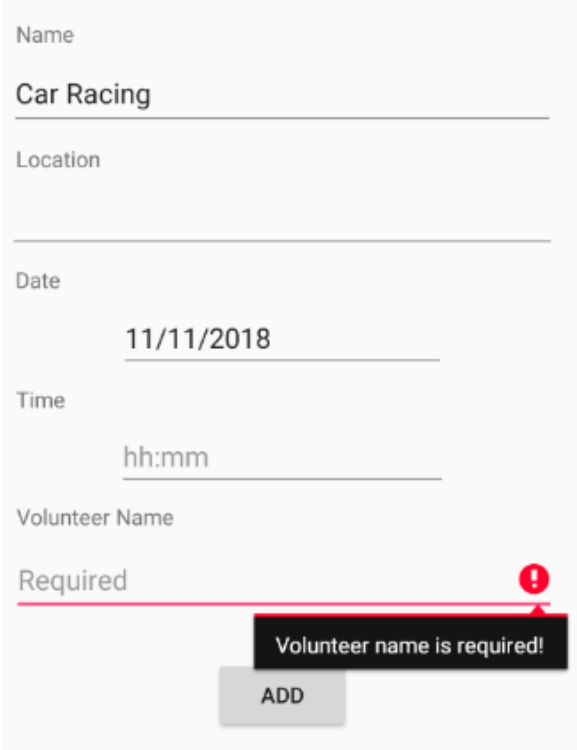


Figure 29 - The volunteer name textfield must not be left empty.

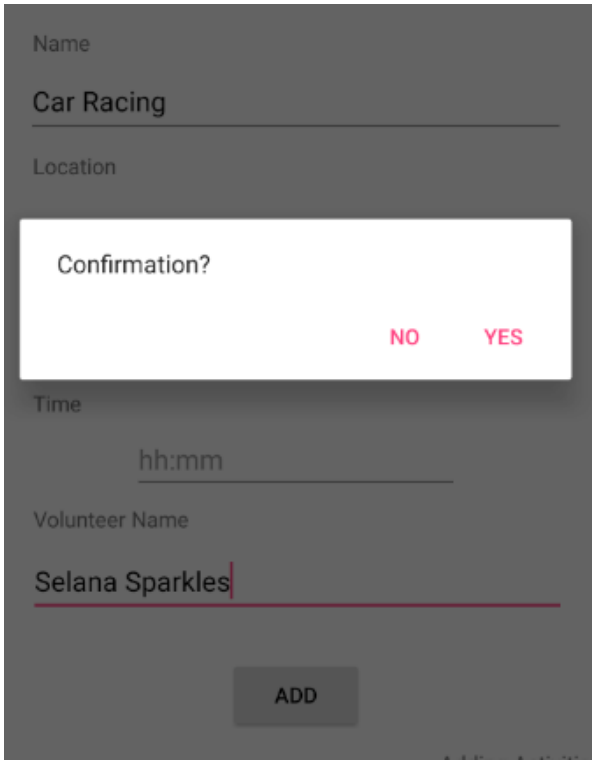


Figure 32 - It will show a confirmation dialog before officially store in the database.

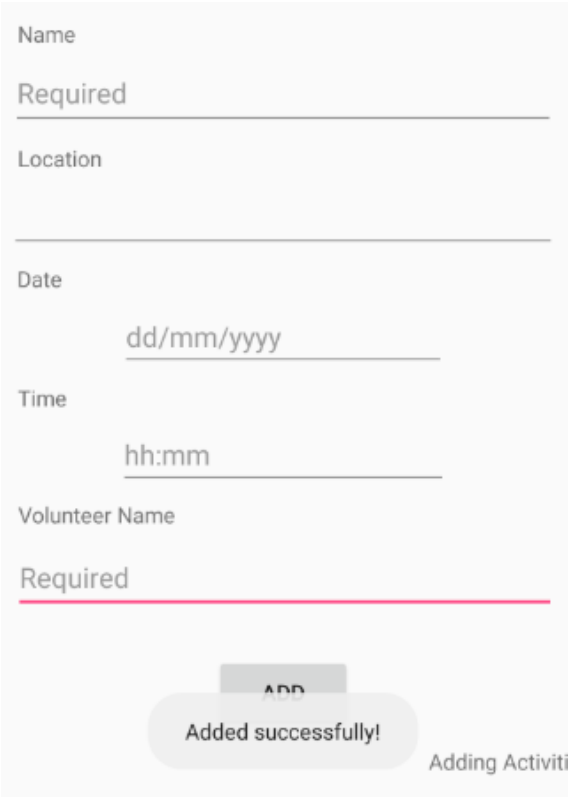


Figure 31 - A notification popup will appear with message "Added!" after user confirms the adding process, finally all the textfields will be reset to be empty.

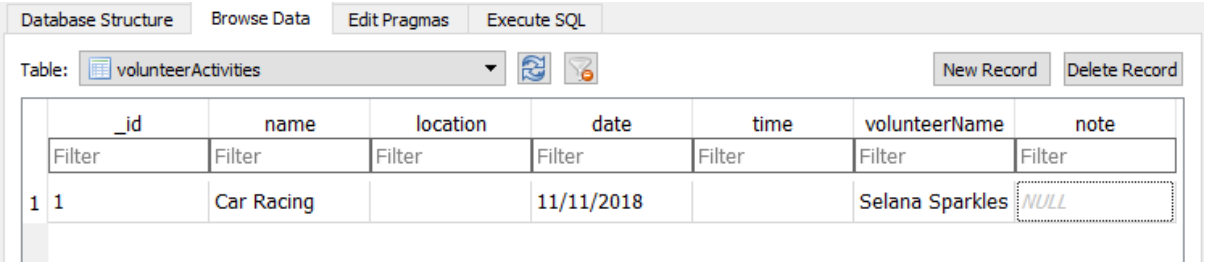


Figure 33 - Result

4.10 Android: Check for duplicate events

In the add volunteer activity page, when we add a new activity, the process will check for duplicated data that currently existed in the database. For example, there is an existed activity that if we insert the exact information of Location, Date and Time, a popup “Data exists!” will appear:

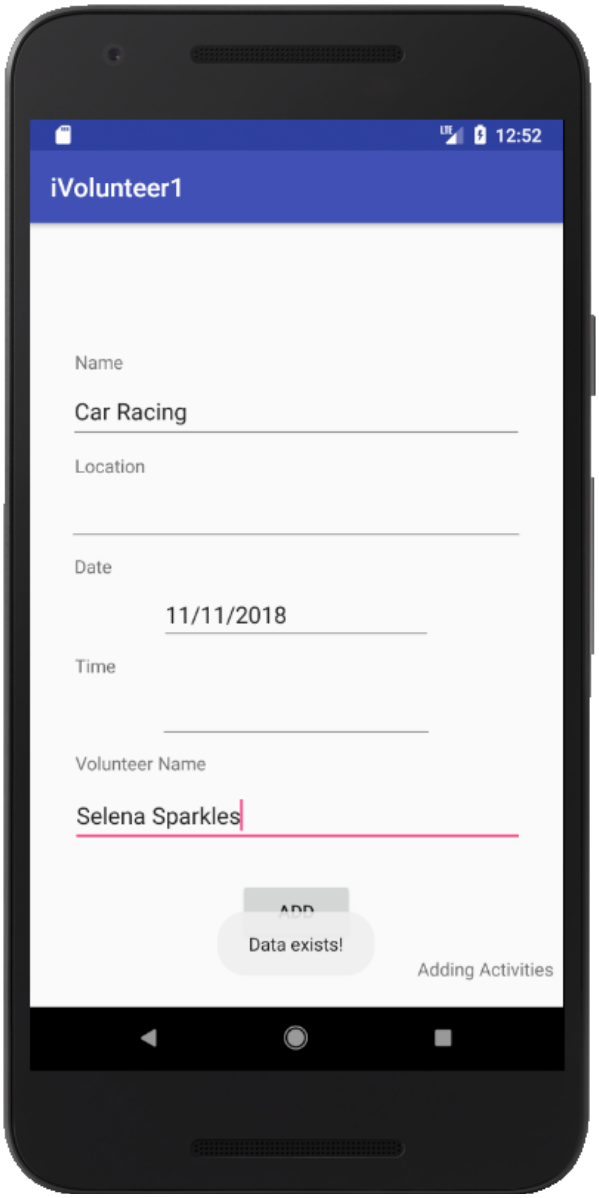


Figure 34 – Duplicate event in Android

Section 5. Evaluation

5.1 Human computer interaction

5.1.1 General UI design principles

The applications were designed based on some general UI design principles selected in Nielsen's 10 heuristics [1], Norman's 7 principles [2] and Shneiderman's 8 golden rules [3] such as:

Consistency

Designing with consistency and standards will reduce the time of user's thinking process, make the contents are quickly caught by human cognition. In iVolunteer, familiar background elements (icons, font, button...) are used; layout/content/navigation structure follows common web structure...

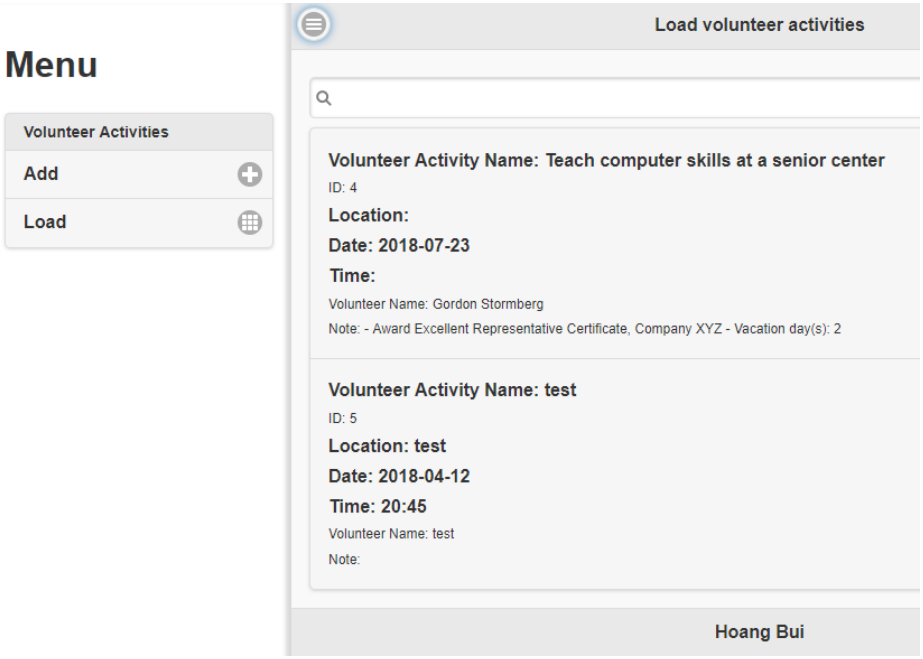


Figure 35 - Consistency

Provide feedback

User always needs feedback about what they are dealing with to raise their confidence when using the app. This method is applied from little actions (click a button/textfield/icon will change its colour to show it has been clicked) to large dialogs/popups to aware user (before/after adding new data will show confirm/alert messages)

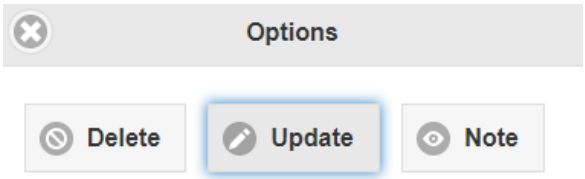


Figure 36 – Glowing button

Prevention is better than cure

In the developing process, we always have to make sure the chance of raising bugs and errors is decreased to the lowest as possible. For instance, table volunteerActivities has declared Name, Date and Volunteer Name to be not null types in the database, and to prevent catching exception happening when entering null data, validations have been implemented.

Figure 37 – Prevent exeception

Easy reversal of actions

It is necessary to implement functions allowing user to reverse their previous action because people usually make mistake. This matter is not focused on in iVolunteer because the app was built to reach the requirement only. However, to be caution to avoid making mistake: Before interacting with data, the system will always show confirmation alert so that user can be able to return back.

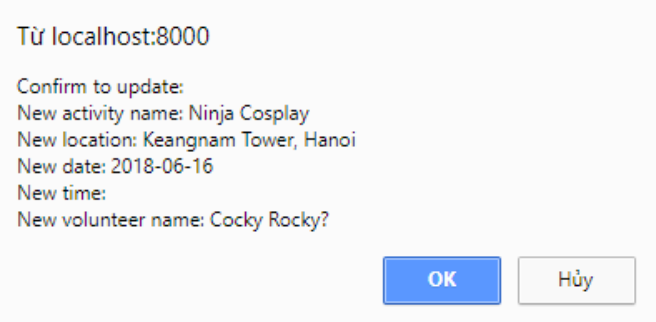


Figure 38 – Confirmation dialog

Recognize rather than recall

It is frustrated for user to remember information from previous part relating to current action. In the PhoneGap version, this method is applied when user clicks an activity to delete, a

confirmation dialog will appear and show all information of user's chosen activity to remind them what they had picked.

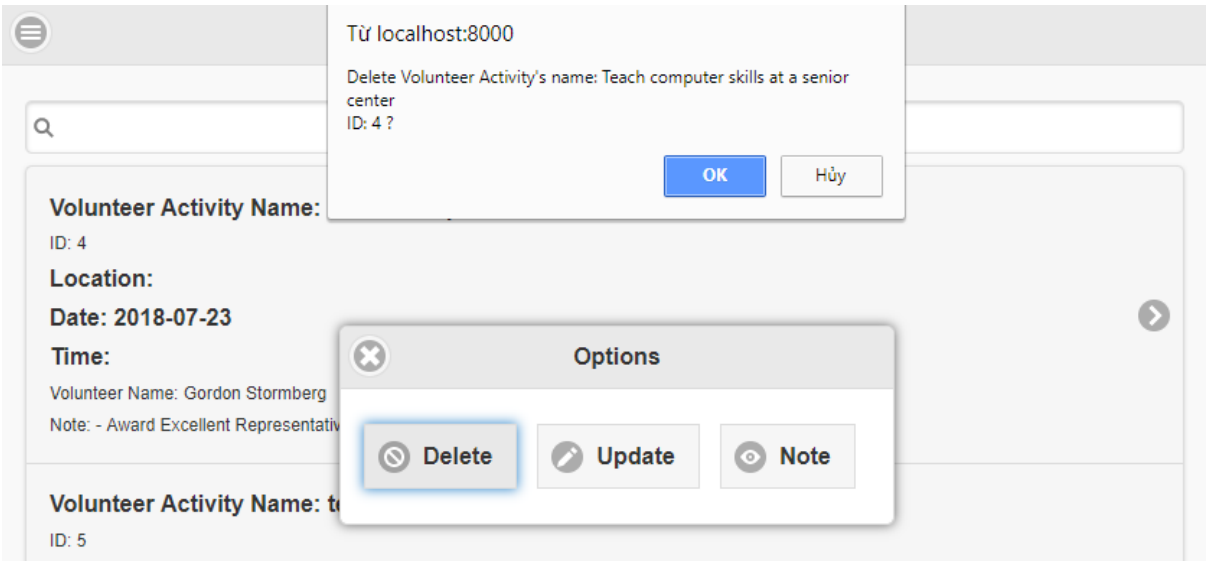


Figure 39 – Recall the information

5.1.2 Mobile HCI

A lot of researches has been spent in mobile HCI recently, some of guidelines[4] in this area are noted below:

Keep it simple

The apps have been implemented based on a simple design in order to keep it easy to use. For example, iVolunteer only contains necessary standard features (navigation bar, add page, load page, header, footer...).

Avoid Scrolling

Non-scrolled screens are user's preference, because user tends to see all content in one solid screen. Both versions of iVolunteer are built based on this method.

Often controls are better at the bottom of the screen than the top

The reason is that user holds the phone at the bottom and their fingers mostly cannot reach the top of the screen. This guideline did not apply in both versions.

On touch screens design for thumbs

Our app's touchscreen is developed for interaction by thumbs because user is usually lazy to move all fingers when using phone.

Textual input - be kind to the user

The input screen in add volunteer page has appropriate keyboard for each input field: text, number, date/time picker. It also enables to erase text by clicking clear button on the right.

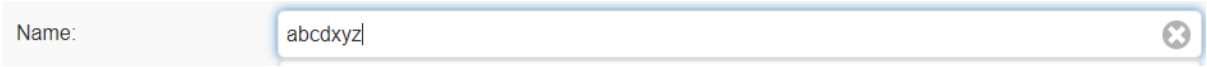


Figure 40 – Clear button

Expect interruptions

Interruption when using app is inevitable (network error, incoming phone call, device loses power...), we should care about this method to allow user continue their work. iVolunteer did not apply this method to concentrate on requirements of coursework only.

Design for different screen sizes/layouts

Responsive design has been applied in PhoneGap version (with the help of Cordova), which means its content can fit many different devices.

5.2 Security

Security is very essential that protects the app from unauthorized access. In this coursework, the applications developed does not provide many security methods because it is built for studying purpose focusing on the requirement of the coursework.

5.3 Maintainability

The role of maintainability is very important in development life cycle, so that we have to make sure the implementation quality is well coded in order to easily manage and fix error in the future. This area has been put efforts in iVolunteer, for example: The app is built with readable code; variable, file, classes and function/method are named properly (e.g. function addVolunteerActivity() executes adding code...); logic indentation; necessary comments provided; files are organized well with distinctive roles (e.g. file volunteerActivitiesHandler.js manages database transaction code, file index.js manages interaction between screen and system...).

5.4 Change(s) needed for live use

If there is a chance to deployed the app for live use, user-friendly will be focused on. For one example, in Android version, user will have to insert two textfields of date and time in designated date and time format. It is exhausted to insert specialize data type by text. Instead, a more valuable option is to implement date and time pickers because they seem to be universally accepted pattern for date and time input and easy to use or understand.

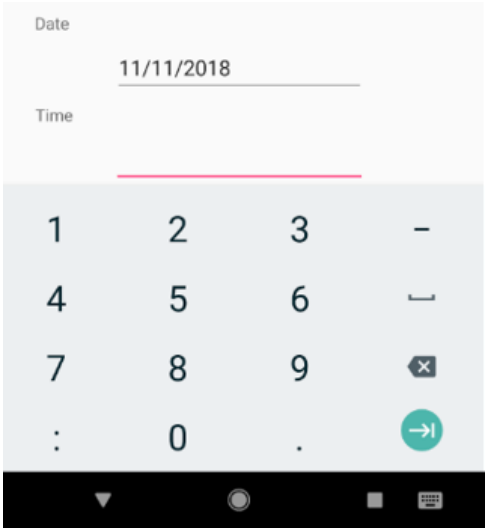


Figure 41 – Uncomfortable input

References

[1] Jakob Nielsen - 10 Usability Heuristics for User Interface Design. [Online]

Available at: https://www.nngroup.com/articles/ten-usability-heuristics/

[Accessed 2018].

[2] Don Norman - Norman's 7 Principles. [Online]

Available at: https://sites.google.com/a/nu.edu.pk/hci-060129/lectures-1/norman-s-7-principles

[Accessed 2018].

[3] Shneiderman - Shneiderman's "Eight Golden Rules of Interface Design". [Online]

Available at:

http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html

[Accessed 2018].

[4] Enrico Rukzio - Physical Mobile Interactions: Mobile Devices as Pervasive Mediators for Interactions with the Real World. PhD Dissertation. Faculty for Mathematics, Computer Science and Statistics. University of Munich. 2007