



APPLICATION DEVELOPMENT FOR MOBILEDEVICES

Bui Huy Hoang



Contents

Contents.....	1
Introduction	2
Section 1. Checklist of features.....	3
Section 2. Bugs and Weaknesses	4
Section 3. Strength.....	5
Section 4. Screenshots	6
4.1 PhoneGap: Basic details input screen.....	6
4.2 PhoneGap: Implement forms validation	7
4.3 PhoneGap: Store, view and delete the basic details	9
4.4 PhoneGap: Check for duplicate events.....	10
4.5 PhoneGap: Search.....	11
4.6 PhoneGap: Add a report (note) input screen	11
4.7 PhoneGap: Additional feature (Update volunteer activity)	13
4.8 Android: Input screen	15
4.9 Android: Implement forms validation	16
4.10 Android: Check for duplicate events	19
Section 5. Evaluation.....	20
5.1 Human computer interaction	20
5.2 Security	23
5.3 Maintainability.....	23
5.4 Change(s) needed for live use.....	24
References	25

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

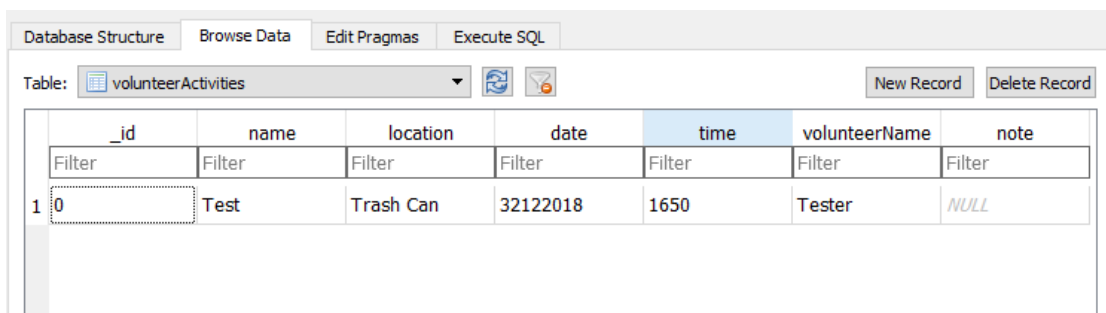
Database exceptions (on personal evaluation) are handled with code but not so efficient. As we can see, when it comes to happen error in database transaction, it will only show the error information in console log. This weakness should have a solution that to alert user and be able to automatically fix 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.



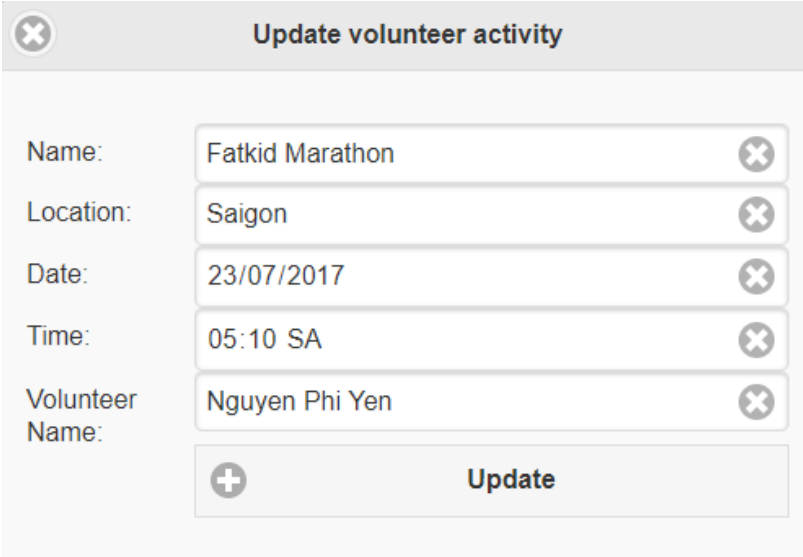
	_id	name	location	date	time	volunteerName	note
1	0	Test	Trash Can	32122018	1650	Tester	NULL

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.



Update volunteer activity	
Name:	Fatkid Marathon
Location:	Saigon
Date:	23/07/2017
Time:	05:10 SA
Volunteer Name:	Nguyen Phi Yen
<div><div>+</div>Update</div>	

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

Menu icon

Add volunteer activity

Name: Required

Location:

Date: dd/mm/yyyy

Time: --:--

Volunteer Name: Required

+ Add

Hoang Bui

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.

Date: dd/mm/yyyy

Time:

Volunteer Name:

Tháng Tư 2018

T2	T3	T4	T5	T6	T7	CN
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Figure 5 - Date picker

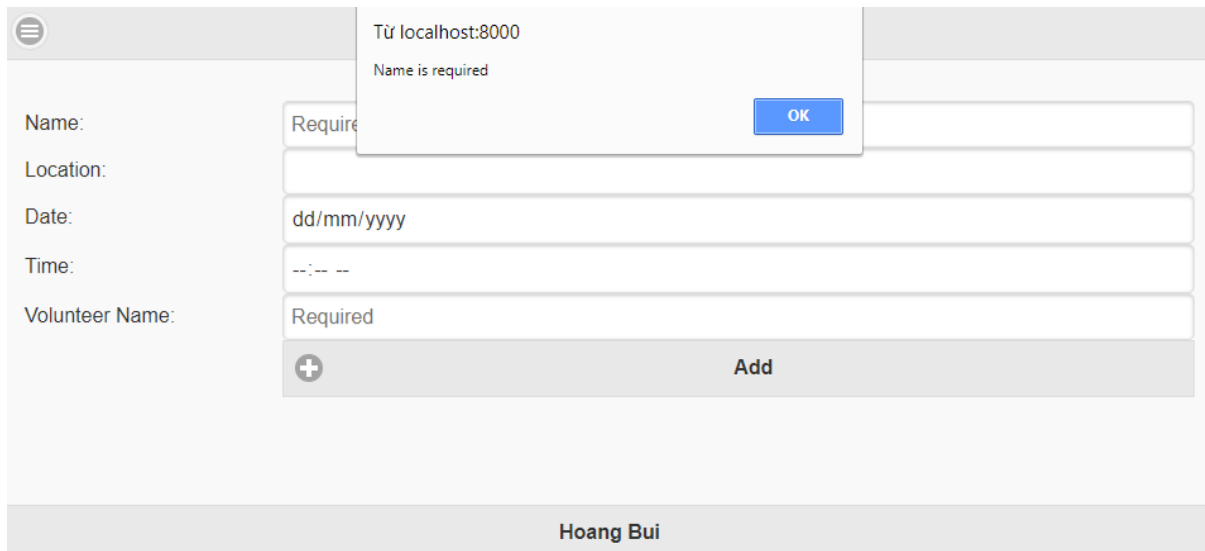
Time: --:--

Figure 6 - Time insert

The add function includes several validations that are mentioned below.

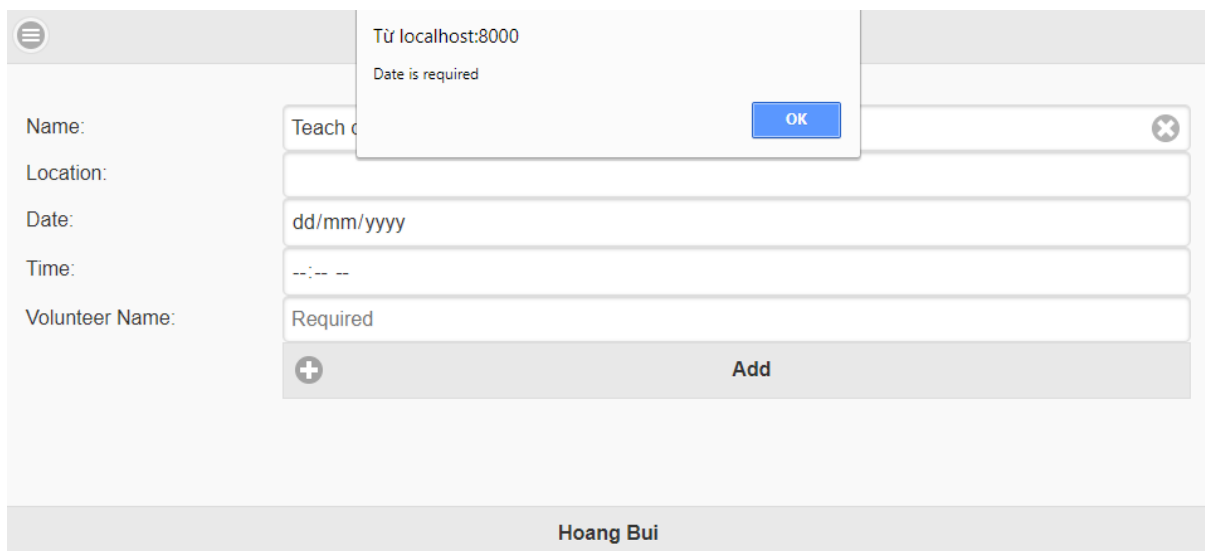
4.2 PhoneGap: Implement forms validation

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.



The screenshot shows a mobile application interface with a form for adding a volunteer activity. The form includes fields for Name, Location, Date, Time, and Volunteer Name. A validation error message is displayed over the Name field, stating "Name is required". The error message box also shows the URL "Từ localhost:8000" and an "OK" button. The form has a placeholder "Required" for the Name field. At the bottom of the form is an "Add" button. The footer of the app displays the name "Hoang Bui".

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



The screenshot shows the same mobile application interface as Figure 7, but with a validation error message displayed over the Date field, stating "Date is required". The error message box also shows the URL "Từ localhost:8000" and an "OK" button. The form has a placeholder "Required" for the Date field. At the bottom of the form is an "Add" button. The footer of the app displays the name "Hoang Bui".

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

The screenshot shows a web application interface for adding a volunteer. On the left, there is a sidebar with a hamburger menu icon and the name 'Hoang Bui'. The main area contains a form with the following fields: 'Name:' (with 'Teach' entered), 'Location:', 'Date:' (with '23/07/2018' entered), 'Time:' (with '--:--' entered), and 'Volunteer Name:' (with 'Required' as a placeholder). A modal dialog box is open over the form, displaying the text 'Từ localhost:8000' and 'Volunteer Name is required', with an 'OK' button.

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.

This screenshot shows the same form as Figure 9, but now all fields are filled: 'Name:' is 'Teach', 'Location:' is empty, 'Date:' is '23/07/2018', 'Time:' is '--:--', and 'Volunteer Name:' is 'Gordon Stormberg'. A confirmation modal dialog is open, showing the text 'Từ localhost:8000', 'Confirm to add:', 'Activity Name: Teach computer skills at a senior center', 'Location:', 'Date: 2018-07-23', 'Time:', and 'Volunteer Name: Gordon Stormberg ?'. The dialog has 'OK' and 'Hủy' buttons. The 'Volunteer Name' field in the form is highlighted in yellow.

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.

The screenshot shows a small notification dialog box with the text 'Từ localhost:8000' and 'Added!'. It has a single 'OK' button.

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.

The screenshot shows a mobile application interface. On the left, a 'Menu' panel is open, displaying 'Volunteer Activities' with 'Add' and 'Load' options. The 'Add' option is highlighted. The main area is titled 'Add volunteer activity' and contains a form with the following fields: 'Name:' (Required), 'Location:', 'Date:' (dd/mm/yyyy), 'Time:' (--:--), and 'Volunteer Name:' (Required). Below the 'Volunteer Name' field is a '+' icon and an 'Add' button. At the bottom of the screen, the name 'Hoang Bui' is displayed.

Figure 12 - Navigating panel

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

The screenshot shows the 'Load volunteer activities' page. At the top, there is a search bar with a magnifying glass icon. Below the search bar, a list item is displayed for a volunteer activity. The list item contains the following information: 'Volunteer Activity Name: Teach computer skills at a senior center', 'ID: 4', 'Location:', 'Date: 2018-07-23', 'Time:', 'Volunteer Name: Gordon Stormberg', and 'Note:'. A right arrow icon is visible on the right side of the list item. At the bottom of the screen, the name 'Hoang Bui' is displayed.

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**.

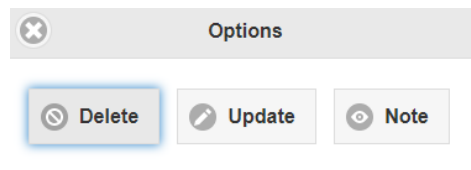


Figure 14 - Options when clicking one activity.

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

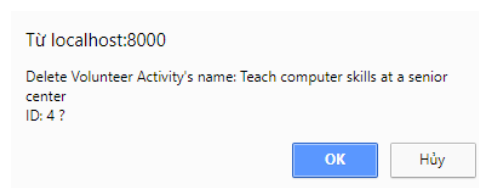


Figure 15 - A confirmation popup.

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:

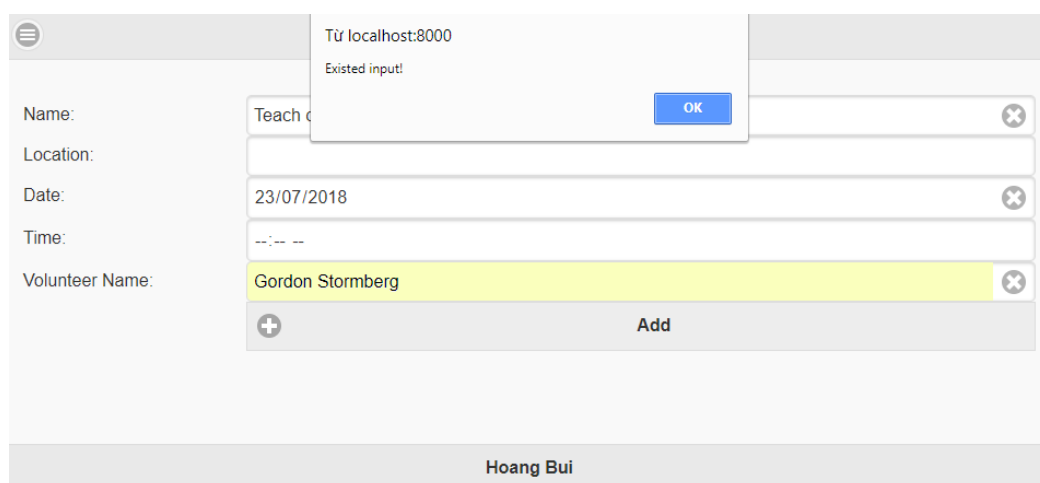


Figure 16 - Input the duplicated data.

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.

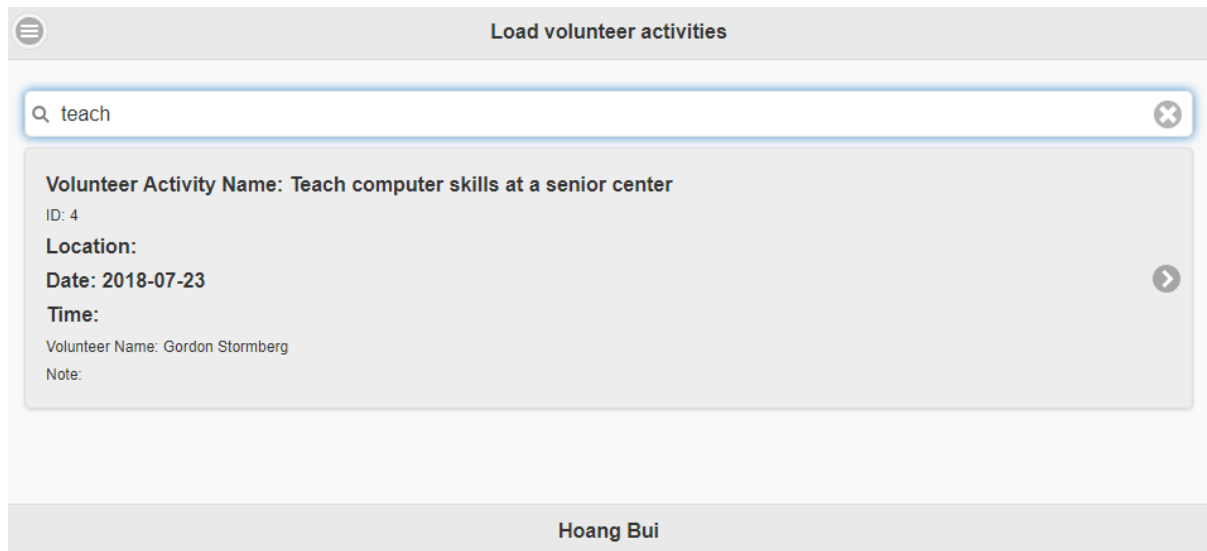


Figure 17 - Search volunteer activities.

4.6 PhoneGap: Add a report (note) input screen

Each volunteer activity has Note section:

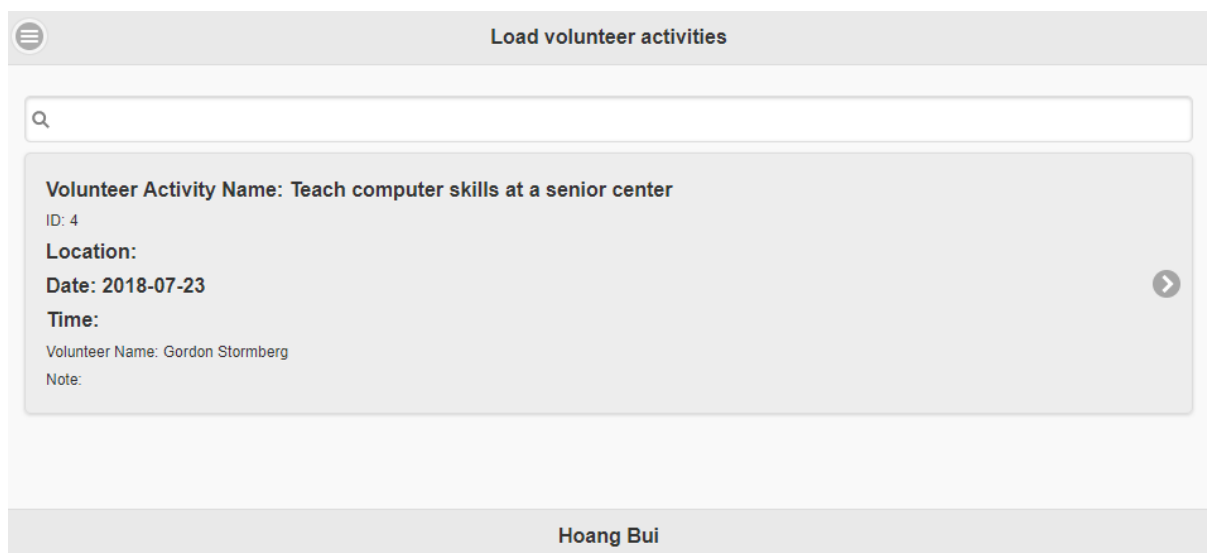


Figure 18 - An existed activity having empty note.

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.

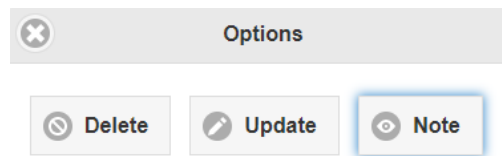


Figure 19 - Options when clicking one activity.

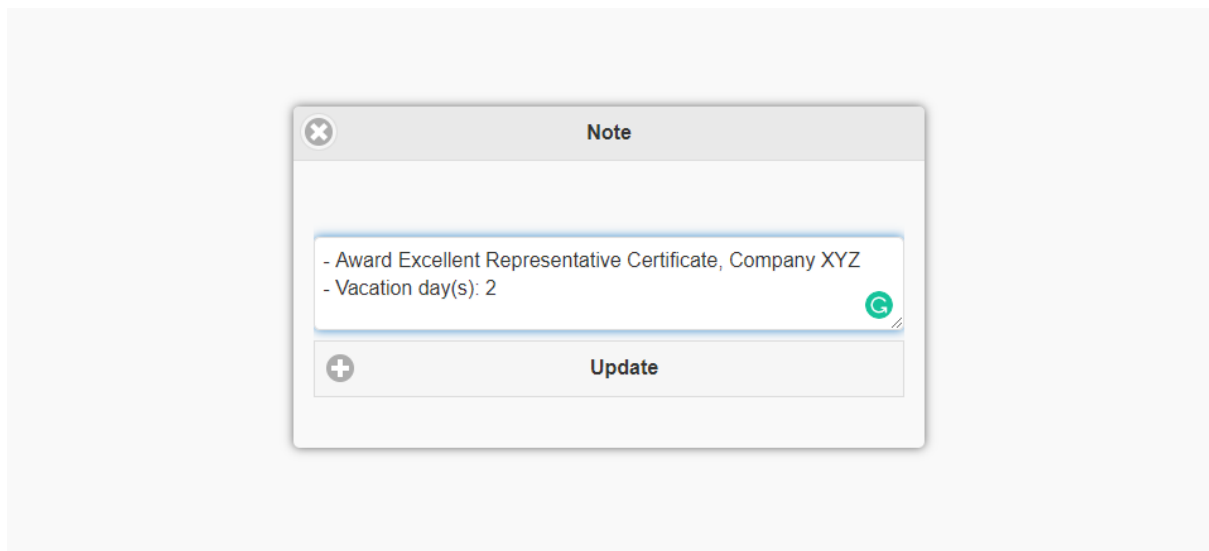


Figure 20 - Update note dialog.

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

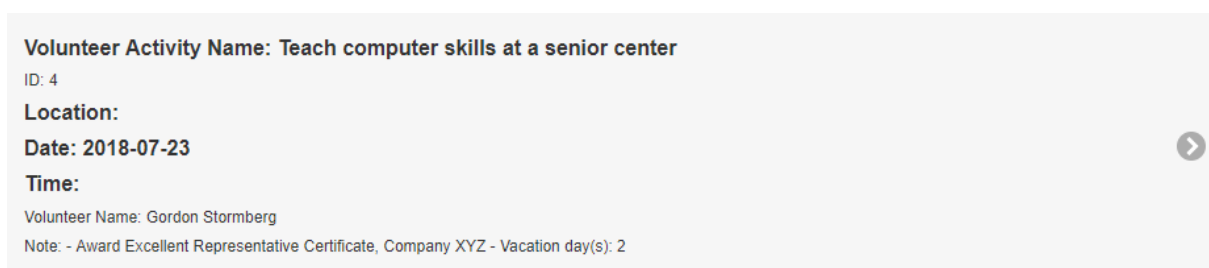


Figure 21 - An existed activity having updated note.

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:

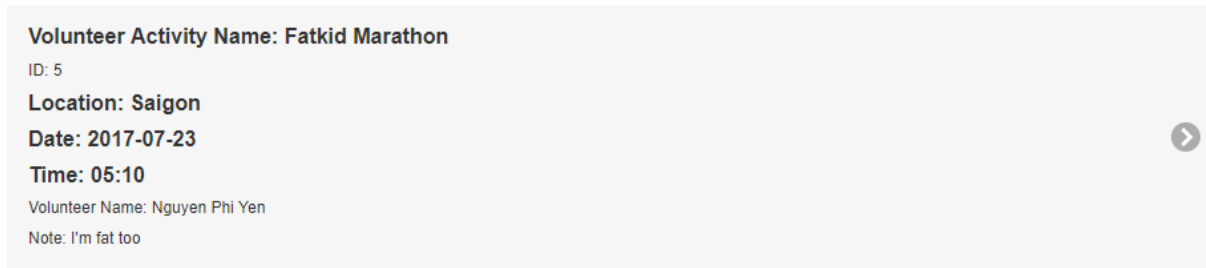


Figure 22 - Before updating activity.

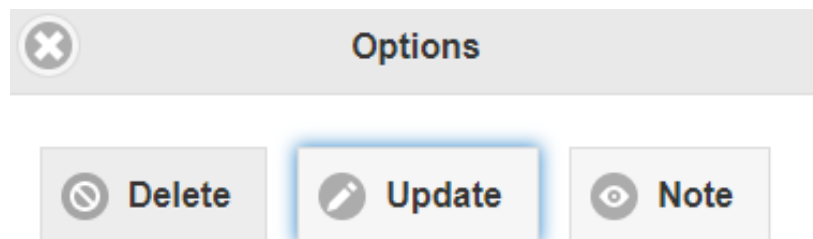


Figure 23 - Options when clicking one activity.

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.

Từ localhost:8000

Confirmation?

OK Hủy

Name: Organize a car wash and donate the profits to

Location: Saigon

Date: 15/04/2018

Time: 09:10 SA

Volunteer Name: Mark Ten

+ Update

Figure 24 - A confirmation popup before updating an activity.

The result will be shown below:

Volunteer Activity Name: Organize a car wash and donate the profits to charity

ID: 5

Location: Saigon

Date: 2018-04-15

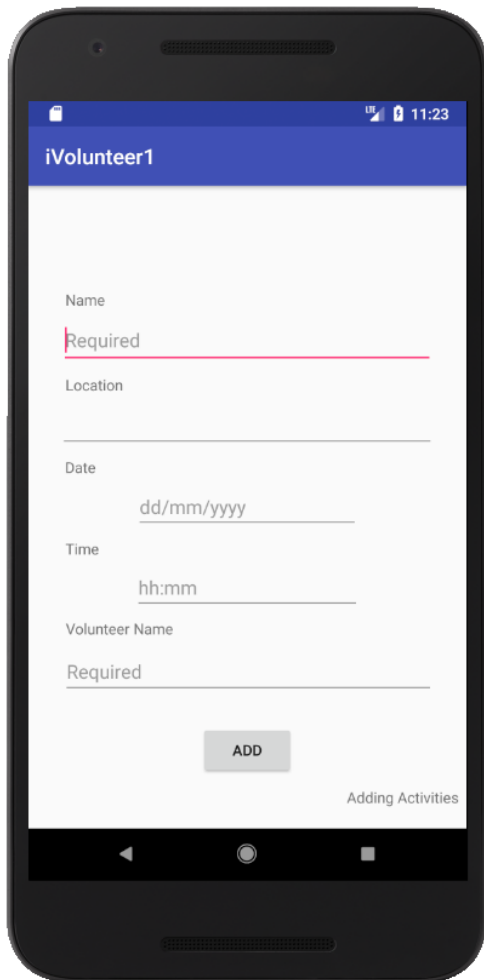
Time: 09:10

Volunteer Name: Mark Ten

Note: I'm fat too

Figure 25 - Updated activity.

4.8 Android: Input screen



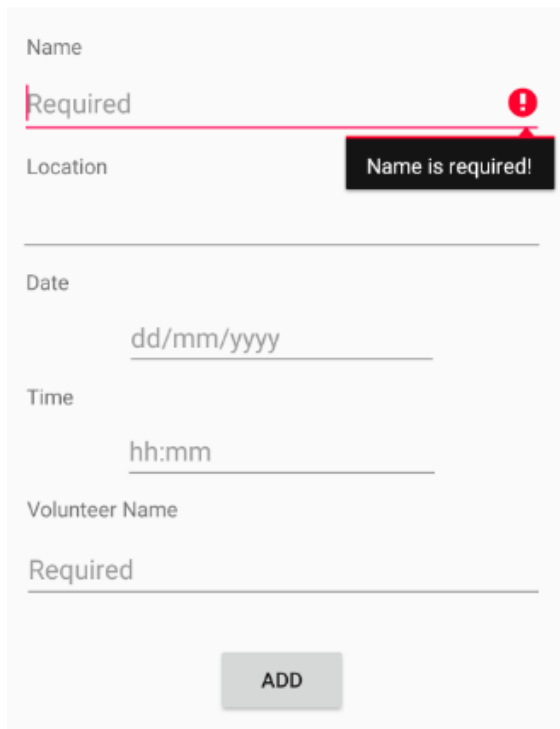
The screenshot shows a smartphone displaying the 'iVolunteer1' app. The app's interface includes a blue header bar with the title 'iVolunteer1'. Below the header, the 'Add volunteer activity' page is visible. It contains five text input fields: 'Name' (marked as 'Required' with a red underline), 'Location', 'Date' (with a 'dd/mm/yyyy' placeholder), 'Time' (with a 'hh:mm' placeholder), and 'Volunteer Name' (also marked as 'Required' with a red underline). At the bottom of the form is a grey 'ADD' button. Below the button, the text 'Adding Activities' is displayed. The smartphone's status bar at the top shows the time as 11:23 and various icons.

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.

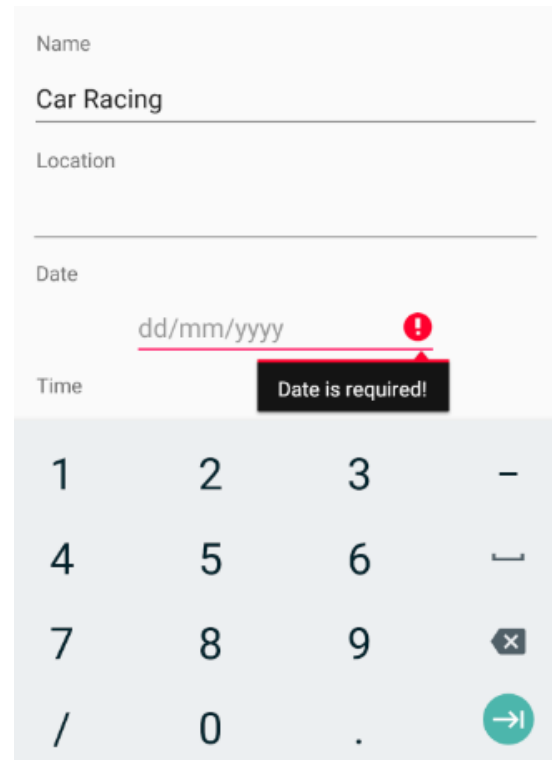
4.9 Android: Implement forms validation

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.



A screenshot of a form with the following fields: Name, Location, Date, Time, and Volunteer Name. The Name field is empty and has a red underline with a red exclamation mark icon. A black error message box with the text "Name is required!" is displayed next to it. The Date field contains the placeholder text "dd/mm/yyyy". The Time field contains the placeholder text "hh:mm". The Volunteer Name field is empty and has the text "Required" below it. An "ADD" button is at the bottom.

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



A screenshot of a form with the following fields: Name, Location, Date, and Time. The Name field contains the text "Car Racing". The Location field is empty. The Date field is empty and has a red underline with a red exclamation mark icon. A black error message box with the text "Date is required!" is displayed next to it. The Time field is empty. A numeric keypad is shown over the form, with the date format "dd/mm/yyyy" displayed above it. The keypad includes numbers 1-9, 0, a decimal point, a backspace key, and a green arrow key.

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.

Name

Car Racing

Location

Date

11/11/2018

Time

hh:mm

1	2	3	-
4	5	6	_
7	8	9	✕
:	0	.	→

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".

Name

Car Racing

Location

Date

11/11/2018

Time

hh:mm

Volunteer Name

Required

Volunteer name is required!

ADD

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

Name

Car Racing

Location

Confirmation?

NO YES

Time

hh:mm

Volunteer Name

Selana Sparkles

ADD

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

Name

Required

Location

Date

dd/mm/yyyy

Time

hh:mm

Volunteer Name

Required

ADD

Added successfully!

Adding Activiti

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.

Database Structure Browse Data Edit Pragmas Execute SQL

Table: volunteerActivities

New Record Delete Record

	_id	name	location	date	time	volunteerName	note
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Car Racing		11/11/2018		Selana Sparkles	NULL

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:

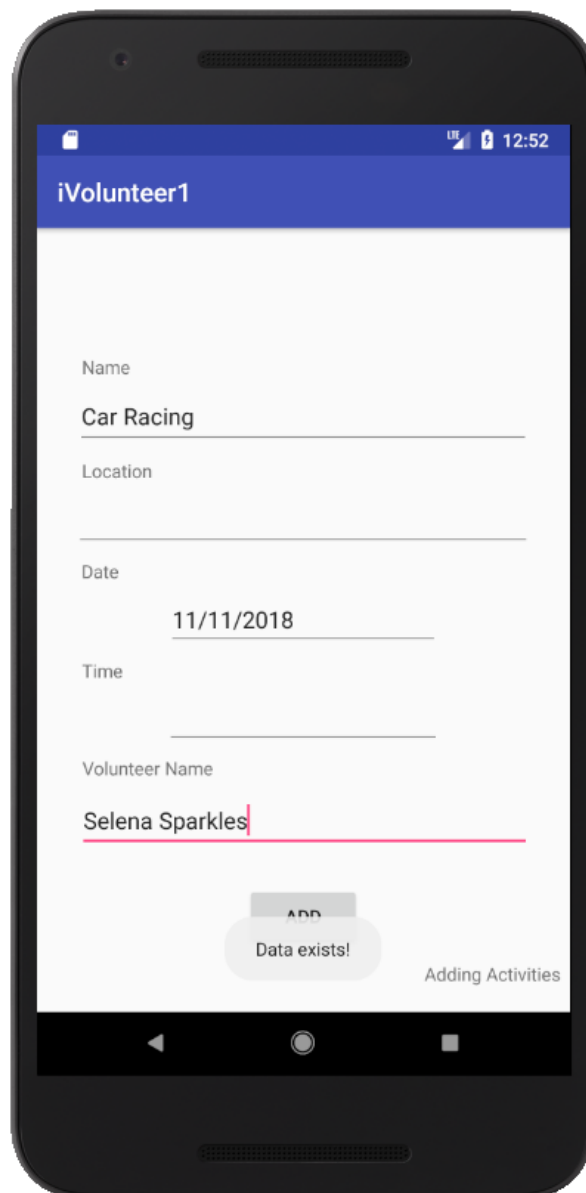


Figure 34 – Duplicate event in Android

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...

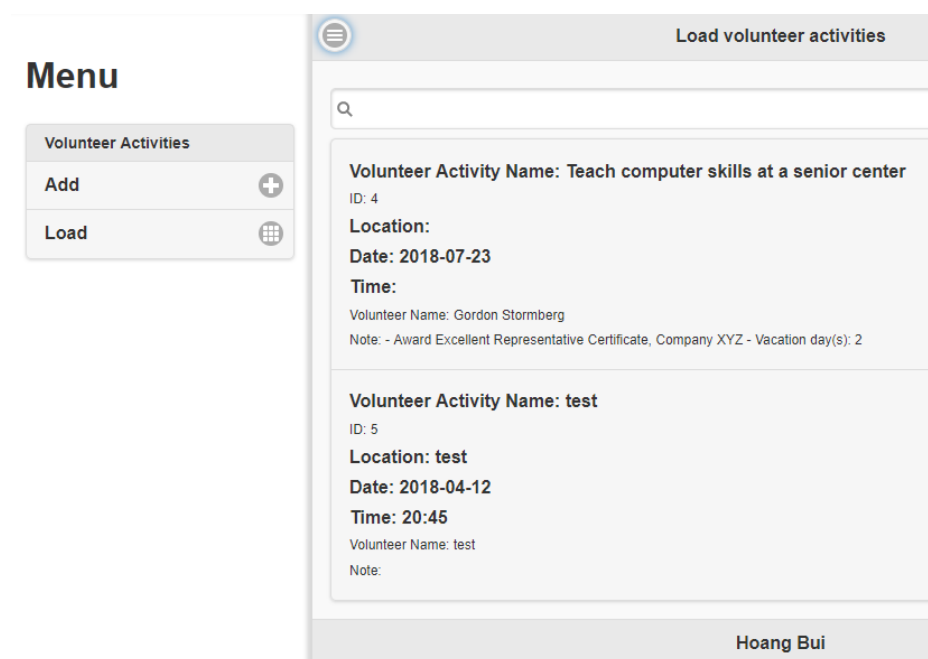


Figure 35 - Consistency

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)

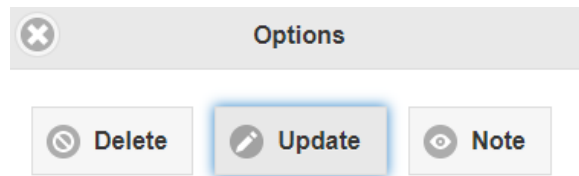


Figure 36 – Glowing button

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 exception

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.

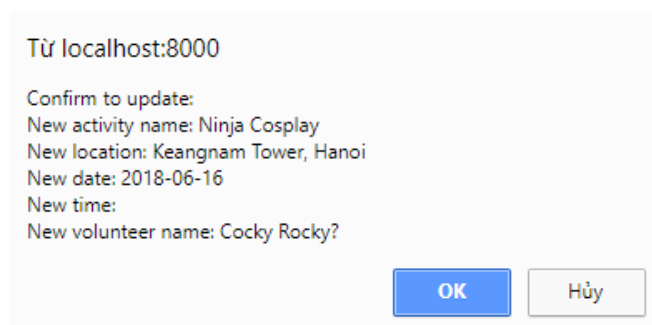


Figure 38 – Confirmation dialog

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.

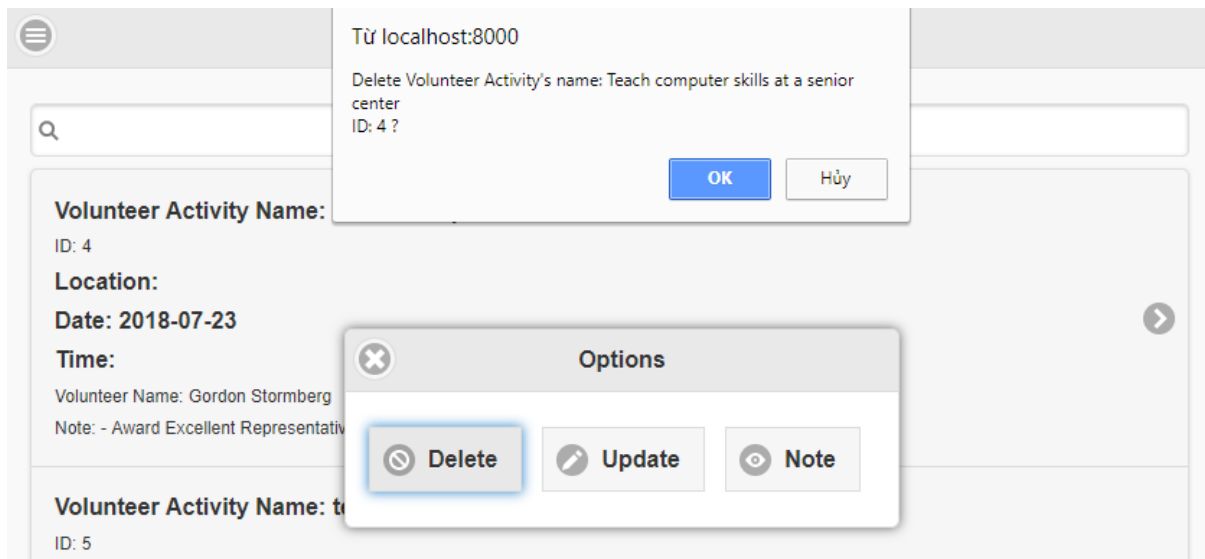


Figure 39 – Recall the information

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.

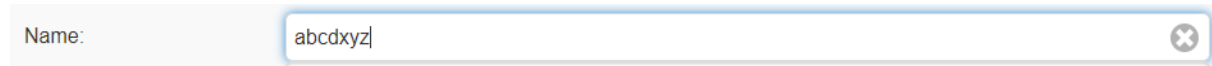


Figure 40 – Clear button

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.

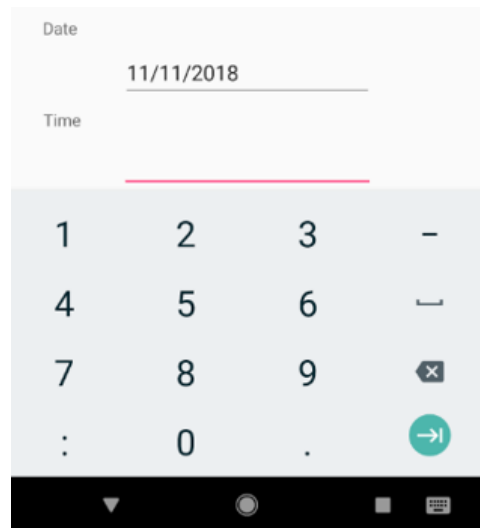


Figure 41 – Uncomfortable input

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