FPT-GREENWICH UNIVERSITY

Vietnam, Hanoi

MOBILE APPLICATION DESIGN AND DEVELOPMENT

COMP1786

Contents

[Table of Figures 1](#_Toc88272161)

[Before start. 3](#_Toc88272162)

[Week 1. 5](#_Toc88272163)

[1. Basic Information 5](#_Toc88272165)

[2. Exercise answer 6](#_Toc88272170)

[2.1. Achievement demonstrating screen shot 6](#_Toc88272171)

[2.2. Written code 7](#_Toc88272172)

[Week 2. 9](#_Toc88272173)

[1. Basic Information 9](#_Toc88272175)

[2. Exercise answer 11](#_Toc88272180)

[2.1. Achievement demonstrating screen shot 11](#_Toc88272181)

[2.2. Written code 12](#_Toc88272182)

[Week 3. 19](#_Toc88272183)

[1. Basic Information 19](#_Toc88272185)

[2. Exercise answer 20](#_Toc88272190)

[2.1. Achievement demonstrating screen shot 20](#_Toc88272191)

[2.2. Written code 20](#_Toc88272192)

[Week 4. 25](#_Toc88272194)

[1. Basic Information 25](#_Toc88272196)

[2. Exercise answer 26](#_Toc88272201)

[2.1. Achievement demonstrating screen shot 26](#_Toc88272202)

[2.2. Written code 28](#_Toc88272203)

# Table of Figures

[Figure 1. Index.html (root page) 3](#_Toc88272375)

[Figure 2. Confirmation dialog. 6](#_Toc88272376)

[Figure 3. Window notification and buttons 6](#_Toc88272377)

[Figure 4. The form and full fields. 11](#_Toc88272378)

[Figure 5. Fields validations. 12](#_Toc88272379)

[Figure 6. Reporter field validation. 12](#_Toc88272380)

[Figure 7. Web SQL 20](#_Toc88272381)

[Figure 8. Form and all 7 fields deployed on native Android. 27](#_Toc88272382)

[Figure 9. Fields validations. 28](#_Toc88272383)

**COMP1786 Logbook.**

# Before start.

Here are general resources that apply to all Logbooks (Exercises)



#### Figure 1. Index.html (root page)

This is the main web page for user to access each function corresponding to each Logbook (Exercise).

**Index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>iExplore</title>

<meta name="format-detection" content="telephone=no" />

<meta name="msapplication-tap-highlight" content="no" />

<meta name="viewport"

content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<script type="text/javascript" src="cordova.js"></script>

<link rel="stylesheet" href="css/custom.css">

</head>

<body>

<div class="container border" style="width: 30%; box-shadow: 0px 4px 5px 0 rgba(0, 0, 0, 0.8); padding-top: 20px; padding-bottom: 20px; border-radius: 5px;">

<h3 class="title">Tran Minh Quang</h3>

<h4 class="title">RentalZ</h4>

<a href="Logbook\_1.html" class="btn btn-custom">

Logbook 1. Notification API

</a>

<a href="Logbook\_2.html" class="btn btn-custom">

Logbook 2. Form Validation

</a>

<a href="Logbook\_3.html" class="btn btn-custom">

Logbook 3. SQLite Database

</a>

</div>

</body>

</html>

Explanation: This is the originating website that the user reaches first. The user presses one of the 3 buttons to proceed to use the function corresponding to the requirements of each Logbooks (Exercises).

# Week 1.

Duration: 17/10/2021 – 23/10/2021

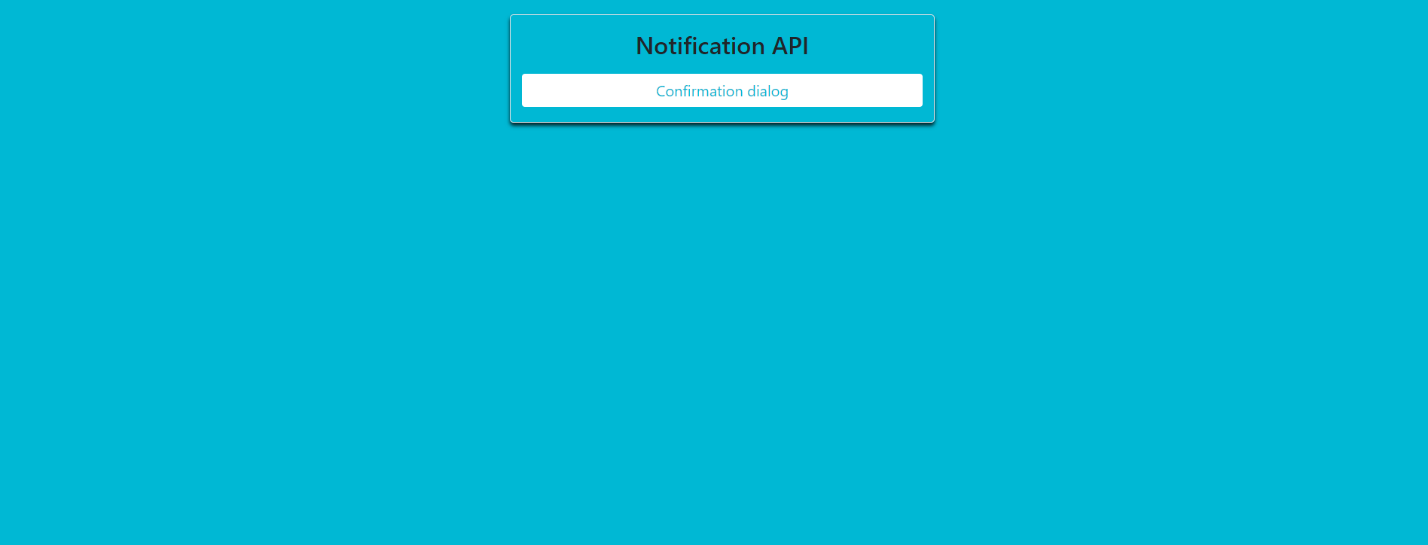
## 1. Basic Information

|  |  |
| --- | --- |
| 1.1 Student name | Tran Minh Quang |
| 1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. | I worked alone. |
| 1.3 Which Exercise is this? | Generate a PhoneGap app with a confirmation dialog box which require user to press one button to ring a bell and another to vibrate. |
| 1.4 How well did you complete the exercise? Tick as appropriate. | * I tried but couldn't complete it 🗹 * I did it but I feel I should have done better ☐ * I did everything that was asked ☐ * I did more than was asked for ☐ |
| 1.5 Briefly explain your answer to question 1.4 | I could not completed the Vibration function. I have consulted many ways from sources like Stackoverflow, Codepen, Github, etc but every test doesn't work no matter what. No effects occur. Currently, I still have not grasped the cause as well as found a way to solve the problem. |

## 2. Exercise answer

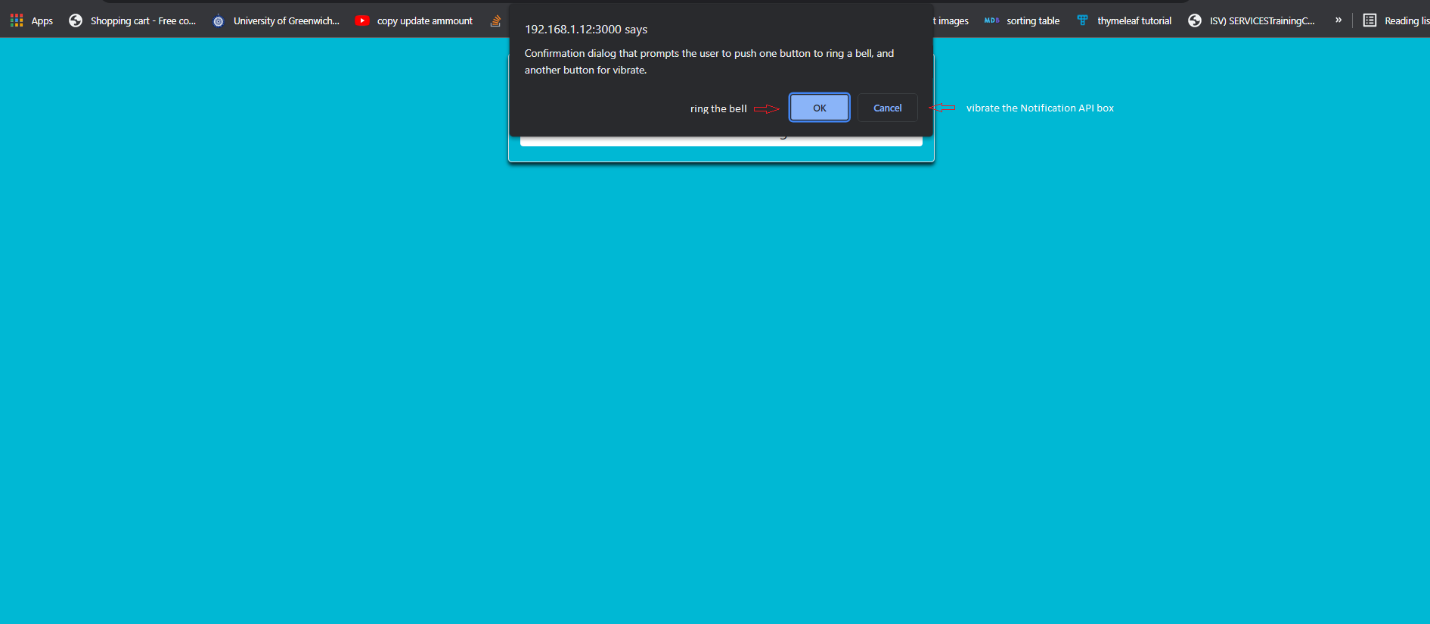
### 2.1. Achievement demonstrating screen shot

Paste screen shots in here. Add explanation of what each screen shot demonstrates



#### Figure 2. Confirmation dialog.

The user clicks on the Confirmation dialog, the window will display a message including a message and 2 buttons, the first one to ring the bell, the other to vibrate.



#### Figure 3. Window notification and buttons

After the user clicks on the Notification dialog, Window will give a notification for the user to choose 1 of 2 buttons (ring the bell or vibrate the Notification API box).

### 2.2. Written code

**Logbook\_1.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Notification API</title>

<meta name="format-detection" content="telephone=no" />

<meta name="msapplication-tap-highlight" content="no" />

<meta name="viewport"

content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<script type="text/javascript" src="cordova.js"></script>

<script type="text/javascript" src="js/notification\_api.js"></script>

<link rel="stylesheet" href="css/custom.css">

</head>

<body>

<div class="container border" style="width: 30%; box-shadow: 0px 4px 5px 0 rgba(0, 0, 0, 0.8); padding-top: 20px; padding-bottom: 20px; border-radius: 5px;">

<h2 class="text-center">Notification API</h2>

<button id="dialog" class="btn btn-custom" style="display: block; margin-top: 20px; width: 100%">

Confirmation dialog

</button>

</div>

<script>

$(document).ready(() => {

$("#dialog").click(function (event) {

showDialog();

});

})

</script>

</body>

</html>

Explanation: This is the page showing the Notification API box and the Confirmation dialog button.

**notification\_api.js**

let ring\_title = "Ring a bell";

let vibrate\_title = "Vibrate";

let dialog\_title = "Confirmation dialog that prompts the user to push one button to ring a bell, and another button for vibrate.";

let dialog\_type = "Warning";

var options = [ring\_title, vibrate\_title];

function onConfirm(index) {

if (options[index - 1] == ring\_title) {

navigator.notification.beep(3);

} else {

navigator.vibrate([1000, 1000, 1000]);

}

}

function showDialog() {

navigator.notification.confirm(

dialog\_title,

onConfirm,

dialog\_type,

options

);

}

Explanation: This is the effect processing page of each button. If the user presses the “OK” button, the ring a bell effect will be deployed. If the user presses the “Cancel” button, the vibrate effect will be deployed.

# Week 2.

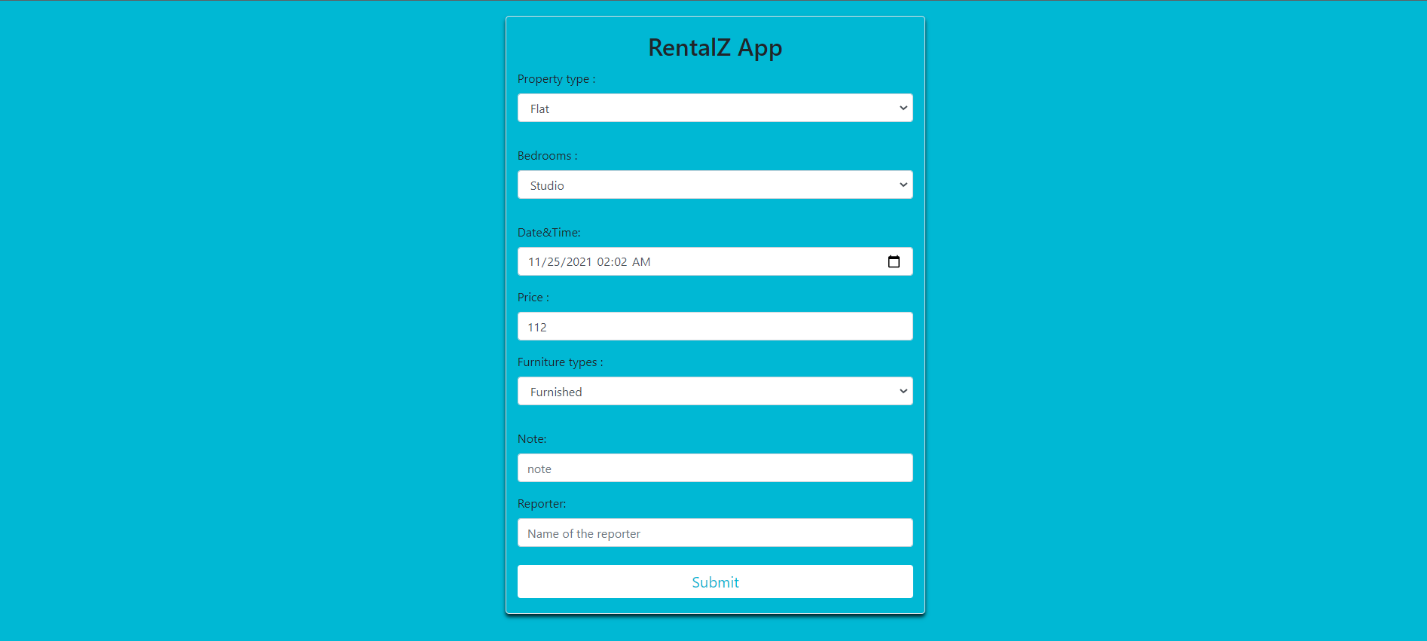
Duration: 24/10/2021 – 30/10/2021

## 1. Basic Information

|  |  |
| --- | --- |
| 1.1 Student name | Tran Minh Quang |
| 1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. | I worked alone. |
| 1.3 Which Exercise is this? | A form with fields for the user to enter the desired information is designed on PhoneGap. In it, each information field will include some validation to check the information entered. Error messages will be displayed next to the corresponding information field if the input is not allowed after validated. |
| 1.4 How well did you complete the exercise? Tick as appropriate. | * I tried but couldn't complete it ☐ * I did it but I feel I should have done better ☐ * I did everything that was asked 🗹 * I did more than was asked for ☐ |
| 1.5 Briefly explain your answer to question 1.4 | I followed the requirements in the coursework and designed a simple form on PhoneGap that fully met the requirements, from the number of fields and validations for each field respectively. |

## 2. Exercise answer

### 2.1. Achievement demonstrating screen shot

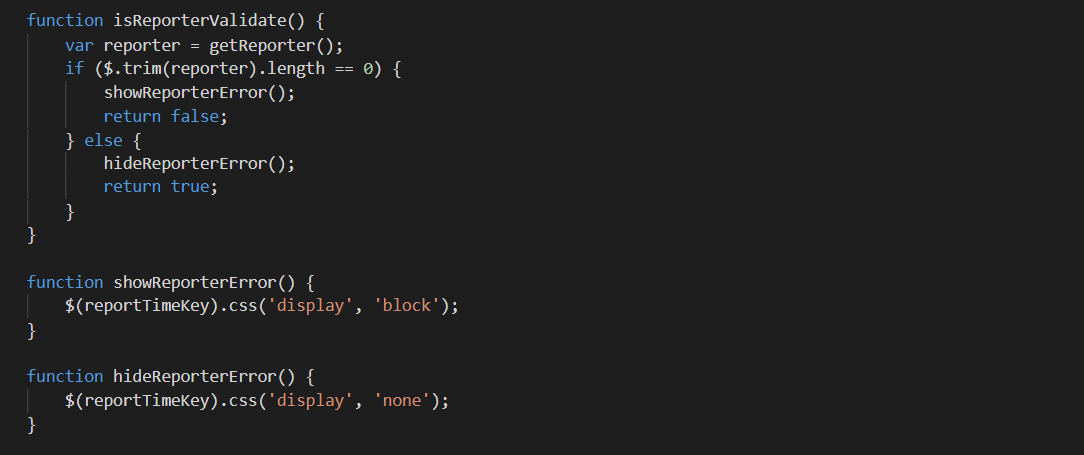


#### Figure 4. The form and full fields.

Explanation: a form contains full 7 required fields according to the coursework requirements.



#### Figure 5. Fields validations.



#### Figure 6. Reporter field validation.

Explanation: the two figures above show how to validate each field. The field “Date&Time” requires a local date and time data type and cannot be empty. The field “Price” requires an integer data type, cannot be empty and the value is limited from 0 to 999. The field “Reporter” accepts text data types and cannot be empty. The remaining fields like “Property Type”, “Bedrooms” and “Furniture Type” are displayed as a list, the default value is the first value in the list whether the user selects a value or not. The “Note” field is optional (no validation required).

### 2.2. Written code

**Logbook\_2.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Data entry screen </title>

<meta name="format-detection" content="telephone=no" />

<meta name="msapplication-tap-highlight" content="no" />

<meta name="viewport"

content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<script type="text/javascript" src="cordova.js"></script>

<script type="text/javascript" src="js/validate.js"></script>

<link rel="stylesheet" href="css/custom.css">

</head>

<body>

<div class="container border" style="width: 30%; box-shadow: 0px 4px 5px 0 rgba(0, 0, 0, 0.8); padding-top: 20px; padding-bottom: 20px; border-radius: 5px;">

<h2 class="text-center title">RentalZ App</h2>

<form id="rentalz" class="input\_form">

<fieldset class="form-group">

<div class="form-group">

<label class="item\_label">Property type : </label>

<select class="form-control" id="property\_type">

<option value="flat">Flat</option>

<option value="house">House</option>

<option value="bungalow">Bungalow</option>

</select>

</div>

</fieldset>

<fieldset class="form-group">

<div class="form-group">

<label class="item\_label">Bedrooms : </label>

<select class="form-control" id="property\_type">

<option value="studio">Studio</option>

<option value="one">One</option>

<option value="two">Two</option>

</select>

</div>

</fieldset>

<fieldset class="form-group">

<label class="item\_label">Date&Time: </label>

<input type="datetime-local" class="form-control" name="date\_time" required>

</fieldset>

<fieldset class="form-group">

<label class="item\_label">Price : </label>

<input type="number" class="form-control" placeholder="Monthly rent price " required name="price" min="0" max="999">

<span id="price\_error" class="item\_error">Price is required</span>

</fieldset>

<fieldset class="form-group">

<div class="form-group">

<label class="item\_label">Furniture types : </label>

<select class="form-control" id="property\_type">

<option value="Furnished">Furnished</option>

<option value="Unfurnished">Unfurnished</option>

<option value="Part Furnished">Part Furnished</option>

</select>

</div>

</fieldset>

<fieldset id="notes" class="form-group">

<label class="item\_label">Note: </label>

<textarea name="note" style="width:100%; padding: 4px; display: block;"></textarea>

</fieldset>

<fieldset id="reporter" class="form-group">

<label class="item\_label">Reporter: </label>

<input type="text" class="form-control" placeholder="Name of the reporter " name="reporter">

<span id="reporter\_error" class="item\_error">Reporter is required</span>

</fieldset>

<button type="submit" class="btn btn-success" style="width: 100%">Submit</button>

</form>

</div>

<script>

$(document).ready(() => {

initDateTime();

$("#rentalz").submit(function (e) {

e.preventDefault();

checkValid();

return false;

});

})

</script>

</body>

</html>

Explanation: interface code to display the form with the required fields and include some validation on the interface.

**validate.js**

let alertError = "Any data invalid. Please re-try!";

let alertSuccess = "All data valid. Save success!";

let reporterKey = "input[name=reporter]"

let noteKey = "input[name=note]";

let timeKey = "input[name=date\_time]";

let dateKey = "input[name=date]";

let priceKey = "input[name=price]";

let reportTimeKey = "#reporter\_error";

function getPropertyType() {

return $('#property\_type').find(":selected").text();

}

function getBedrooms() {

return $('#bedrooms').find(":selected").text();

}

function getFurnitureType() {

return $('#furniture\_type').find(":selected").text();

}

function getPrice() {

return $(priceKey).val();

}

function getNote() {

return $(noteKey).val();

}

function getReporter() {

return $(reporterKey).val();

}

function checkValid() {

var isValid = isReporterValidate();

if (isValid) {

showMessageSuccess();

} else {

showMessageFailed();

}

}

function showMessage(message) {

window.alert(message);

}

function showMessageFailed() {

showMessage(alertError);

}

function showMessageSuccess() {

showMessage(alertSuccess);

}

function getDateTime() {

return $(timeKey).val();

}

function isReporterValidate() {

var reporter = getReporter();

if ($.trim(reporter).length == 0) {

showReporterError();

return false;

} else {

hideReporterError();

return true;

}

}

function showReporterError() {

$(reportTimeKey).css('display', 'block');

}

function hideReporterError() {

$(reportTimeKey).css('display', 'none');

}

Explanation: code validation in js file. Since the fields like “Property Type”, “Furniture Type”, “Bedrooms”, “Date&Time”, “Price”, “Notes” have been validated on the interface, so in the js file it is ready to receive the user input value. Enter always to prepare for the step of storing data in Logbook 3. Particularly, the field “Reporter” will be validated in this file and also receive the value after validated to prepare for the data storage step.

# Week 3.

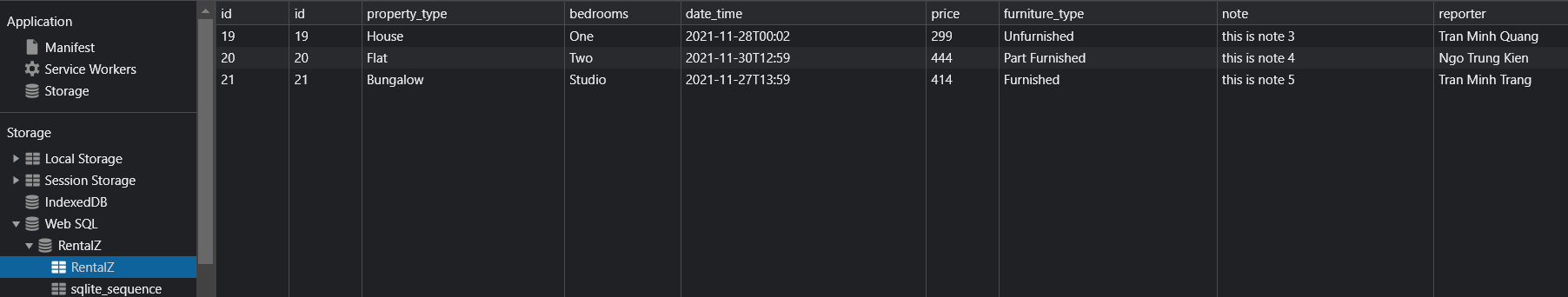
Duration: 31/10/2021 – 06/11/2021

## 1. Basic Information

|  |  |
| --- | --- |
| 1.1 Student name | Tran Minh Quang |
| 1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. | I worked alone. |
| 1.3 Which Exercise is this? | Basically, this Exercise requires preparing a space to store the data entered from the form in the above request into the SQL database. |
| 1.4 How well did you complete the exercise? Tick as appropriate. | * I tried but couldn't complete it ☐ * I did it but I feel I should have done better ☐ * I did everything that was asked 🗹 * I did more than was asked for ☐ |
| 1.5 Briefly explain your answer to question 1.4 | I followed the requirements in the coursework and designed a data storage space as desired. It's Web SQL - a hybrid solution that combines the browser with SQLite to provide database creation and administration on the client side. |

## 2. Exercise answer

### 2.1. Achievement demonstrating screen shot



#### Figure 7. Web SQL

Explanation. Web SQL on a web browser and some testing data.

### 2.2. Written code

**database.js**

search = document.getElementById('search');

let onSearch = function (event) {

keyword = search.value.toLowerCase();

searchData(keyword);

}

search.addEventListener('keyup', onSearch);

$(document).ready(() => {

initDatabase();

// initDateTime();

getAllData();

$("#rentalz").submit(function (e) {

e.preventDefault();

saveData();

getAllData();

return false;

});

})

let db;

let database = {

name: "RentalZ",

version: "1.0",

description: "Create an SQLite database to store the event details entered into the RentalZ App",

space: 5 \* 1024 \* 1024,

create: "CREATE TABLE IF NOT EXISTS RentalZ(id INTEGER PRIMARY KEY AUTOINCREMENT, property\_type character(255),bedrooms character(255), date\_time character(30), price character(255), furniture\_type character(255),note character(255), reporter character(30))",

select: "SELECT \* FROM RentalZ",

search: "SELECT \* FROM RentalZ where reporter like ?",

delete: "DELETE FROM RentalZ where id = ?",

add: "INSERT INTO RentalZ (property\_type, bedrooms, date\_time,price,furniture\_type, note, reporter) VALUES (?,?,?,?,?,?,?)"

}

let deleteSuccess = "Delete item success";

let serverError = "Server error, please re-try!";

let addSuccess = "All data has been saved in database;";

function initDatabase() {

db = openDatabase(

database.name,

database.version,

database.description,

database.space

);

db.transaction(function (tx) {

tx.executeSql(database.create);

})

}

function saveData() {

let propertyType = getPropertyType();

let bedrooms = getBedrooms();

let dateTime = getDateTime();

let furnitureType = getFurnitureType();

let price = getPrice();

let note = getNote();

let reporter = getReporter();

db.transaction(function (tx) {

tx.executeSql(database.add, [propertyType, bedrooms, dateTime, price, furnitureType, note, reporter], (transaction, result) => {

showAddSuccess();

}, (transaction, err) => {

errorServer();

console.log(err);

});

})

}

function getAllData() {

db.transaction((tx) => {

tx.executeSql(database.select, [], (transaction, result) => {

displayData(result);

})

})

}

function searchData(text) {

db.transaction((tx) => {

tx.executeSql(database.search, ["%" + text + "%"], (transaction, result) => {

displayData(result);

}, (transaction, error) => {

console.log(error);

})

})

}

function displayData(result) {

let html = "";

let rows = result.rows;

for (var i = 0; i < rows.length; i++) {

var row = rows.item(i)

html += '<div class="col-md-4 div\_item border" style="box-shadow: 0 4px 5px 0 rgba(0, 0, 0, 0.8);">';

html += ' <label style="width: 100%;"><span style="font-size: 20px;">Reporter</span><span style="float: right; font-weight: bold; font-size: 20px;">' + row['reporter'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Property Type</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['property\_type'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Bedrooms</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['bedrooms'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Furniture Type</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['furniture\_type'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Time</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['date\_time'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Note</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['note'] + '</span></label>';

html += ' <label style="width: 100%;"><span style="font-size: 16px;">Price</span><span style="float: right; font-weight: bold; font-size: 16px;">' + row['price'] + '$</span></label>';

html += ' <button style="margin-bottom: 5px; margin-top: 20px; float: right; width: 25%;" class="btn btn-warning" onclick="deleteItem(' + row['id'] + ')">Delete</button>';

html += '</div>';

}

$("#content").html(html);

}

function deleteItem(id) {

db.transaction((tx) => {

tx.executeSql(database.delete, [id], (transaction, result) => {

displayData(result);

showMessage(deleteSuccess);

getAllData();

}, (transaction, error) => {

showMessage(errorServer);

console.log(error);

})

})

}

function showAddSuccess() {

showMessage(addSuccess);

}

function errorServer() {

showMessage(serverError);

}

Explanation: the database.js file contains a function to receive the value entered from the user's side after validation; contains functions to initialize the database and declare query statements for some basic functions such as adding, deleting, searching for data; Contains functions to display an error message if there is a problem with storing the data, as well as a success message if the data is successfully stored.

# Week 4.

Duration: 07/11/2021 – 13/11/2021

## 1. Basic Information

|  |  |
| --- | --- |
| 1.1 Student name | Tran Minh Quang |
| 1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. | Name: Phung Thai Linh  Login id: none |
| 1.3 Which Exercise is this? | Deploy the second Exercise on a native Android platform. |
| 1.4 How well did you complete the exercise? Tick as appropriate. | * I tried but couldn't complete it ☐ * I did it but I feel I should have done better ☐ * I did everything that was asked 🗹 * I did more than was asked for ☐ |
| 1.5 Briefly explain your answer to question 1.4 | With the support of Phung Thai Linh - an experienced person in Android programming, we followed the requirements in the coursework and designed an interface with full fields of information and validations corresponding to each field. |

## 2. Exercise answer

### 2.1. Achievement demonstrating screen shot

|  |  |  |  |
| --- | --- | --- | --- |
| bed room | furniture type | note | price |
| Bedrooms | Furniture Type | Notes | Price |

|  |  |  |
| --- | --- | --- |
| property type | specific time picker | time picker |
| Property Type | Time Picker | Date Picker |

#### Figure 8. Form and all 7 fields deployed on native Android.

Explanation: All 7 fields which are deployed on a native Android platform.

|  |  |  |  |
| --- | --- | --- | --- |
| general submit valid | valid bedrooms | valid property type | valid price |
| General Submit Validation | Bedrooms Validation | Property Type Validation | Price Validation |

|  |
| --- |
| valid reporter |
| Reporter Validation |

#### Figure 9. Fields validations.

Explanation: validations corresponding to each field.

### 2.2. Written code

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

android:padding="@dimen/\_24sdp">

<TextView

android:id="@+id/add\_rental\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/circle\_background"

android:gravity="center"

android:includeFontPadding="false"

android:padding="@dimen/\_12sdp"

android:text="@string/title"

android:textColor="@color/main\_color"

android:textSize="@dimen/\_18sdp"

android:textStyle="italic"

app:layout\_constraintTop\_toTopOf="parent" />

<ScrollView

android:id="@+id/add\_screen"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_marginTop="@dimen/\_12sdp"

android:background="@drawable/circle\_background"

android:overScrollMode="never"

android:scrollbars="none"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/add\_rental\_title">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingTop="@dimen/\_24sdp">

<vn.quang.rentalz.BaseSpinner

android:id="@+id/property\_type\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bs\_error="@string/property\_type\_error"

app:bs\_note="@string/property\_type\_note"

app:bs\_title="@string/property\_type\_title" />

<vn.quang.rentalz.BaseSpinner

android:id="@+id/bedrooms\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bs\_error="@string/bedrooms\_error"

app:bs\_note="@string/bedrooms\_note"

app:bs\_title="@string/bedrooms\_title" />

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_24sdp"

android:layout\_marginTop="@dimen/\_12sdp"

android:layout\_marginEnd="@dimen/\_24sdp">

<TextView

android:id="@+id/date\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@color/white"

android:includeFontPadding="false"

android:text="@string/date\_title"

android:textColor="@color/main\_color"

android:textSize="@dimen/\_14sdp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/date\_note"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

android:background="@color/white"

android:includeFontPadding="false"

android:text="@string/date\_note"

android:textColor="@color/main\_color"

android:textSize="@dimen/\_10sdp"

android:textStyle="italic"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/date\_title" />

<DatePicker

android:id="@+id/date\_edit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/date\_note" />

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

android:paddingStart="@dimen/\_24sdp"

android:paddingEnd="@dimen/\_24sdp">

<TextView

android:id="@+id/time\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@color/white"

android:includeFontPadding="false"

android:text="@string/time\_title"

android:textColor="@color/main\_color"

android:textSize="@dimen/\_14sdp"

android:textStyle="italic"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/time\_note"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

android:background="@color/white"

android:includeFontPadding="false"

android:text="@string/time\_note"

android:textColor="@color/main\_color"

android:textSize="@dimen/\_10sdp"

android:textStyle="italic"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/time\_title" />

<TimePicker

android:id="@+id/time\_edit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/time\_note" />

</androidx.constraintlayout.widget.ConstraintLayout>

<vn.quang.rentalz.BaseEditText

android:id="@+id/price\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bet\_error="@string/price\_error"

app:bet\_note="@string/price\_note"

app:bet\_title="@string/price\_title" />

<vn.quang.rentalz.BaseSpinner

android:id="@+id/furniture\_types\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bs\_error="@string/furniture\_types\_error"

app:bs\_note="@string/furniture\_types\_note"

app:bs\_title="@string/furniture\_types\_title" />

<vn.quang.rentalz.BaseEditText

android:id="@+id/note\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bet\_note="@string/note\_note"

app:bet\_title="@string/note\_title" />

<vn.quang.rentalz.BaseEditText

android:id="@+id/reporter\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_12sdp"

app:bet\_error="@string/reporter\_error"

app:bet\_note="@string/reporter\_note"

app:bet\_title="@string/reporter\_title" />

<Button

android:id="@+id/btn\_submit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:layout\_marginBottom="36dp"

android:background="@drawable/button\_custom"

android:padding="@dimen/\_8sdp"

android:text="@string/button\_title"

android:textColor="@color/white" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

Explanation: Because I don't have much experience or time to learn about Android Studio, most of the code is done by Phung Thai Linh. My task is to write lines of code that declare the values ​​of the layout so that my partner can call them and display them on the layout.