MOBILE APPLICATION DESIGN AND DEVELOPMENT

COMP1786

FPT-GREENWICH UNIVERSITY
Vietnam, Hanoi

Table of Contents

Table of Figures

Introduction:

Today, in the era of rapidly developing technology, it is undeniable that the importance of software products has significantly improved people's lives and brought huge profits in the business. For software developers, to fully exploit all the advantages of their products, developing software on multi-platforms, especially for Android and IOS - which users interact more than computers (Window) will increase access to users as well as increase revenue. Below is a report on an experimental project called RentalZ developed on PhoneGap and native Android platforms.

I. Features Testing:

Features	Implementation	
	Phonegap	Android
a)	Fully implemented	Fully implemented
b)	I used to not be able to display validating messages according to the considered condition. I have debug it.	Fully implemented
c)	I cannot pass values from two fields Date and Time. Now I can debug.	No requirement
d)	I can't make it search by keywords of the fields, I can only search by a single field (reporter). Right now I still can't debug it.	No requirement
e)	No additional features implemented	No additional features implemented
f)	Fully implemented	I'm having a bit of trouble with the Date and Time fields because I have them displayed on the template as a clock and a calendar. I am not able to make the user selectable time and date on the template. I have debug it now.
g)	No additional features implemented	No additional features implemented

II. Bugs List:

Number	Bugs
1	I have tried to separate Date and Time into two separate fields of information in the database but when checking the stage of receiving the value passed back from the user for storage, the system keeps giving an error that the column does not exist Date in the database even though I have fully declared the SQL query statements so that the system creates a data table for me. Currently I still can't debug.
2	I can't reformat the Date and Time formats to my liking. I tried many ways but the system kept underlining the code to format them. Currently I still can't debug.

III. Special Strengths:

In fact, this is a system that doesn't have many highlights. The system is not provided with an attractive user interface and the number of validations is not much. In general, the system is only suitable for amateur developers who are new to Android Development and are experiencing the

basic functions first. However, it also has certain strengths such as ease of use. Because the simplicity of the system is high, it becomes very easy to grasp the file structure, function functions as well as the processing flow of the data. In addition, the way the code is written is also followed the coding standard to support the maintenance and upgrade of the project.

Example:

- Android.

```
public BaseEditText(@NonNull Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    initView(context, attrs);
}
```

Figure 1. Coding standard example 1.

The figure “Coding standard example 1” shows how to name methods and variables according to the mixed case rule in java.

```
private void setTitle(String title) { getTitleView().setText(title); }

private void setNote(String note) { getNoteView().setText(note); }

private void setError(String note) { getErrorView().setText(note); }

private String getValue() { return getEditText().getText().toString(); }
```

Figure 2. Coding standard example 2.

In the figure “Coding standard example 2”, the variable names also contain a verb to describe what the method will do according to standard Java code.

- PhoneGap.

```
    // Update DOM on a Received Event
    receivedEvent: function(id) {
        var parentElement = document.getElementById(id);
        var listeningElement = parentElement.querySelector('.listening');
        var receivedElement = parentElement.querySelector('.received');

        listeningElement.setAttribute('style', 'display:none;');
        receivedElement.setAttribute('style', 'display:block;');

        console.log('Received Event: ' + id);
    }
};
```

Figure 3. Coding standard example 3.

The figure “Coding standard example 3” shows name of method and variable has been used mixed cased name convention, contains a verb to describe roles, also follows the Javascript coding standard.

IV. Functions Demonstration and Explanation:

Below are some demonstrating figures and explanation of each function figure meaning.

Feature a)

- Short function description.

In general, this is a project built in a simple way suitable for students with high school level. In the project, the fields of price, notes, reporter will be designed a text-field for users to be more active in filling out information; The furniture type, property type, bedrooms, date, time fields will be provided by the developer with options for users to choose from.

- Android Illustrating figures.

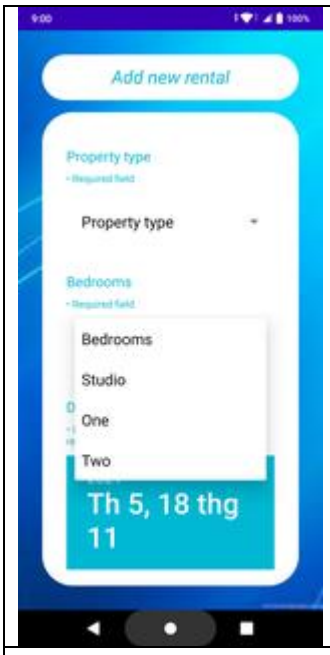


Figure 4. Bedrooms (Android)

Figure 5. Furniture Type (Android)

Figure 6. Notes (Android)

Figure 7. Price (Android)

Figure 8. Propert Type (Android)

Figure 9. Time Picker (Android)

Figure 10. Date Picker (Android)

- PhoneGap Illustrating figures.

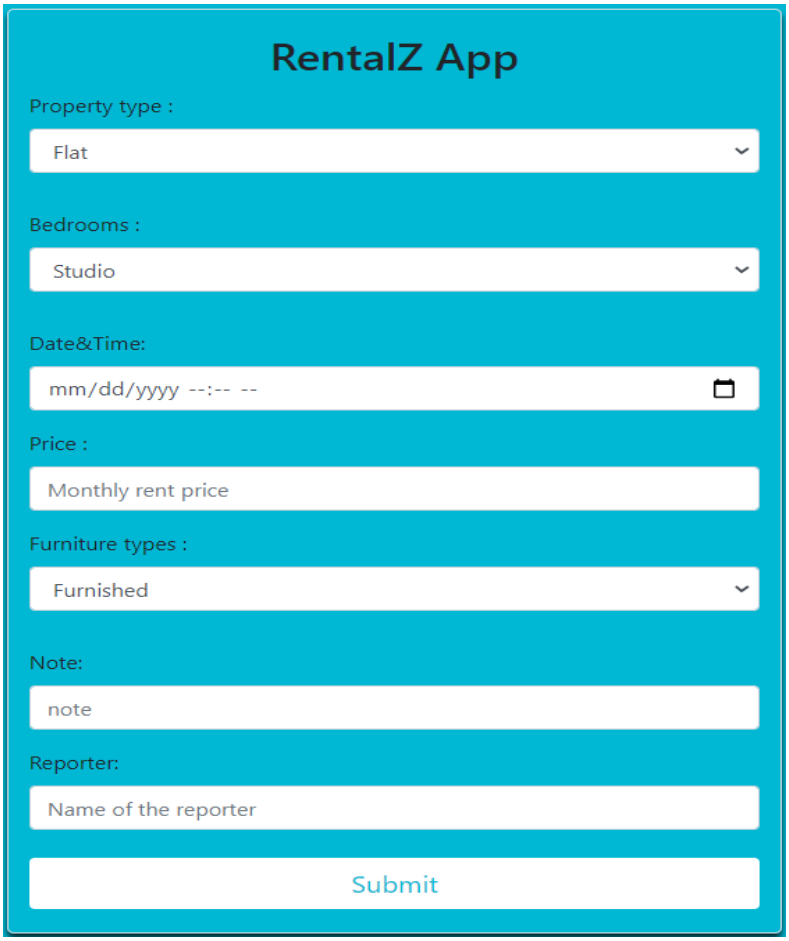


Figure 11. All fields (PhoneGap)

The figure above show 6 fields which has been developed on PhoneGap.

Feature b)

- Short functions description.

In order to preserve data and memory capacity of the system, each field will be provided by the designer with validating field variable to limit unnecessary information from the user. More specifically, the price field cannot be left blank and only allows numbers to be entered; the reporter and date time fields cannot be empty; the property type and bedrooms fields are required to select a value from the list provided

- Android Illustrating figures.

Figure 12. General Submit Validation (Android)

Figure 13. Bedrooms Validation (Android)

Figure 14. Property Type Validation (Android)

Figure 15. Price Validation (Android)

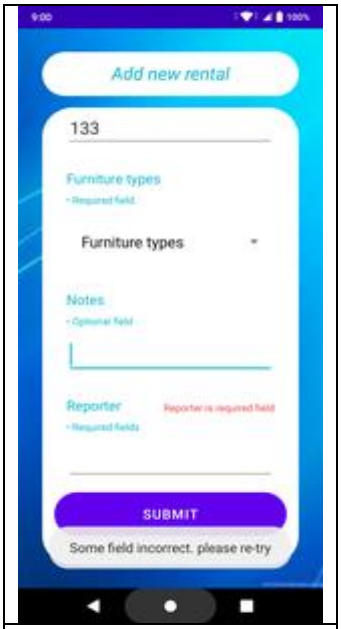


Figure 16. Reporter Validation (Android)

- PhoneGap Illustrating figures.

```
    <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="bedrooms">
                <option value="studio">Studio</option>
                <option value="one">One</option>
                <option value="two">Two</option>
            </select>
        </div>
    </fieldset>
```

Figure 17. Property Type and Bedrooms Validation (PhoneGap)

According to the coursework requirements, Property Type and Bedrooms are required fields, but I set it as select-option so it is unnecessary to validate because if user not choosing value, the default value of each select-option will be the first one.

```
<fieldset class="form-group">
    <label class="item_label">Price : </label>
    <input type="number" class="form-control" name="price" placeholder="Monthly rent price " required name="price" min="0"
        max="999">
    <span id="price_error" class="item_error">Price is required</span>
</fieldset>
```

Figure 18. Price Validation (PhoneGap)

Price field has been set type number mean only number allowed, required mean this field can not be blanked and 0 <= its value <= 999.

```
<fieldset class="form-group">
    <label class="item_label">Reporter: </label>
    <input type="text" class="form-control" placeholder="Name of the reporter " name="reporter" required>
</fieldset>
```

Figure 19. Reporter Validation (PhoneGap)

Reporter field just require to full fills. It can not be blanked.

Feature c)

- Short functions description.

  According to the logbook requirements, the functions of storing, adding, deleting and searching data will only be implemented in logbook3. More specifically, PhoneGap will show three logbook numbers first, user choose the third one and it will display a form equivalent to Android's form. When the user fills in the information and after presses the submit button, the information will be stored on Web SQL - a client-side database storage and management. After that, the user can search for data by filling in the reporter name in search-field and then can continue to add and delete data.

- PhoneGap Illustrating figures.

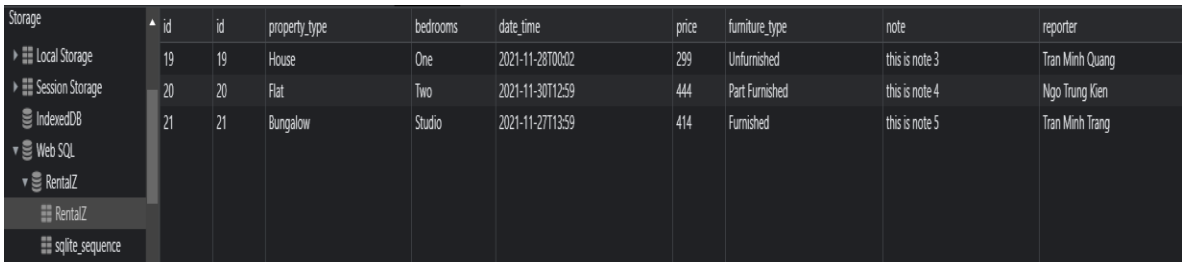


Figure 20. Web SQL data (PhoneGap)

The data which have been saved in database.

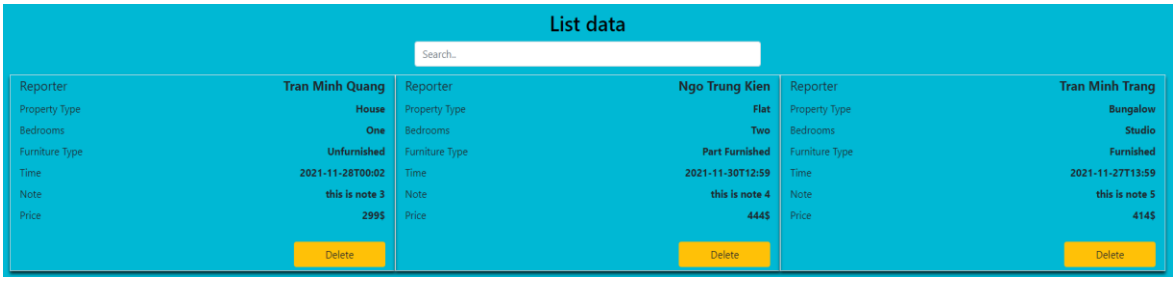


Figure 21. Display data (PhoneGap)

Figure 22. Delete all data (PhoneGap)

Empty database after delete all data.

Feature d)

- Short functions description.

  Search function will search by reporter name to get all data of its record.

- PhoneGap Illustrating figures.

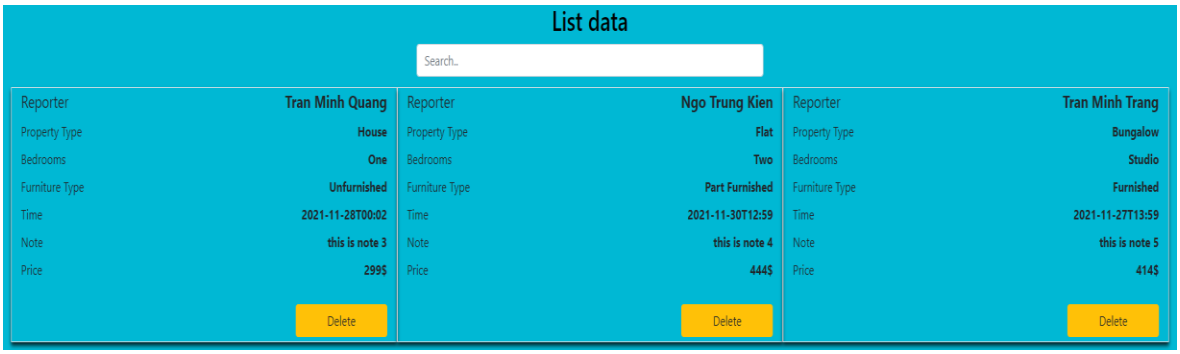


Figure 23. Before implement search function (PhoneGap)

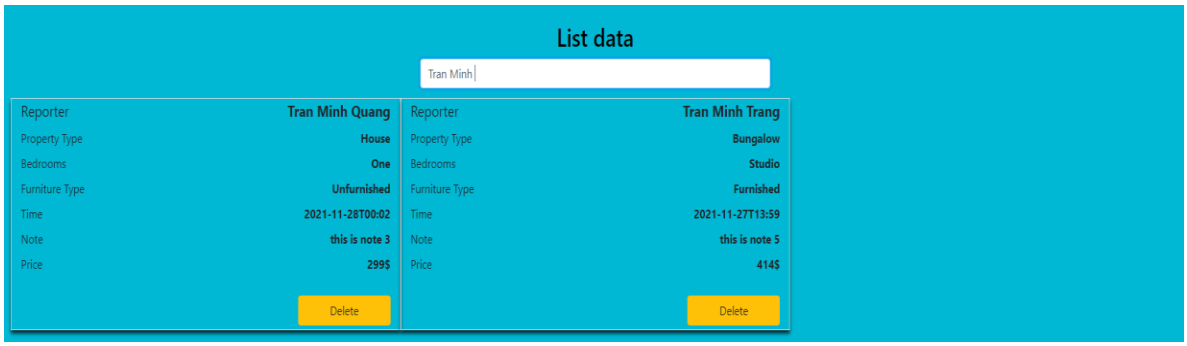


Figure 24. After implemented search function (PhoneGap)

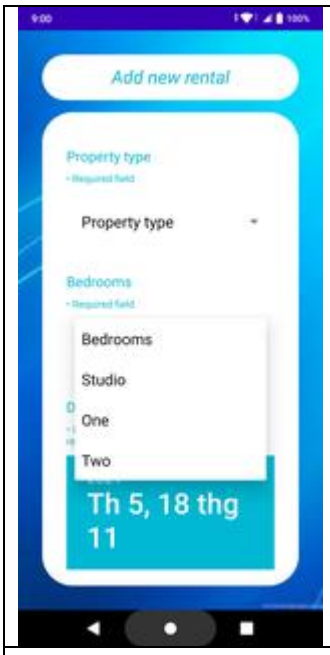
Feature e) No additional features implemented

Feature f)

- Short functions description.

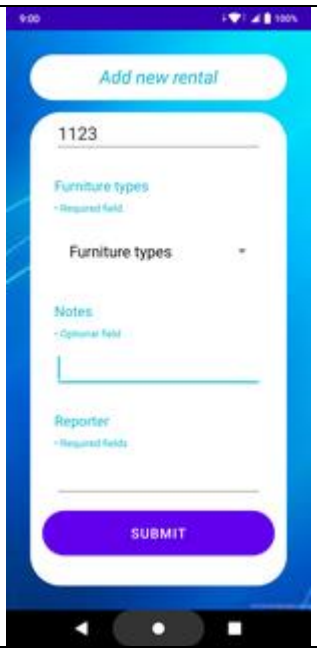
  Here are some images and proofs that feature a) and b) are developed on a native Android using the Java programming language.

- Android Illustrating figures.



Bedrooms

Furniture Type



Notes

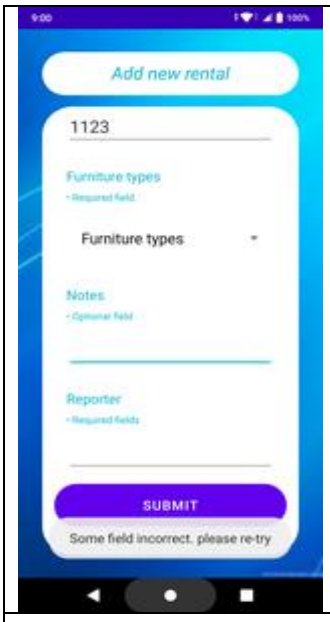
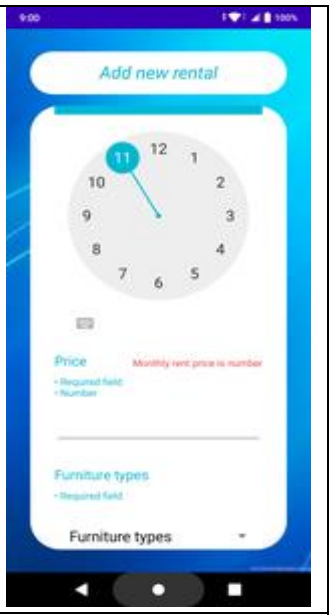
Price

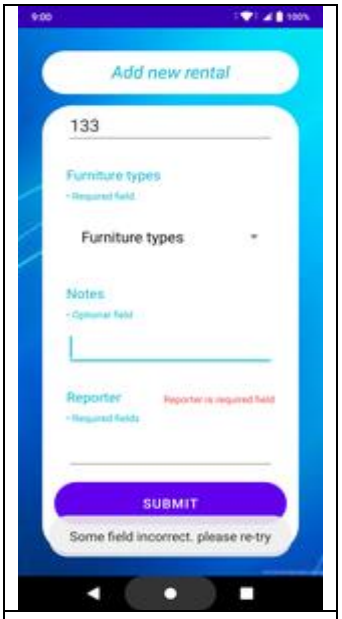
Property Type

Time Picker

Date Picker

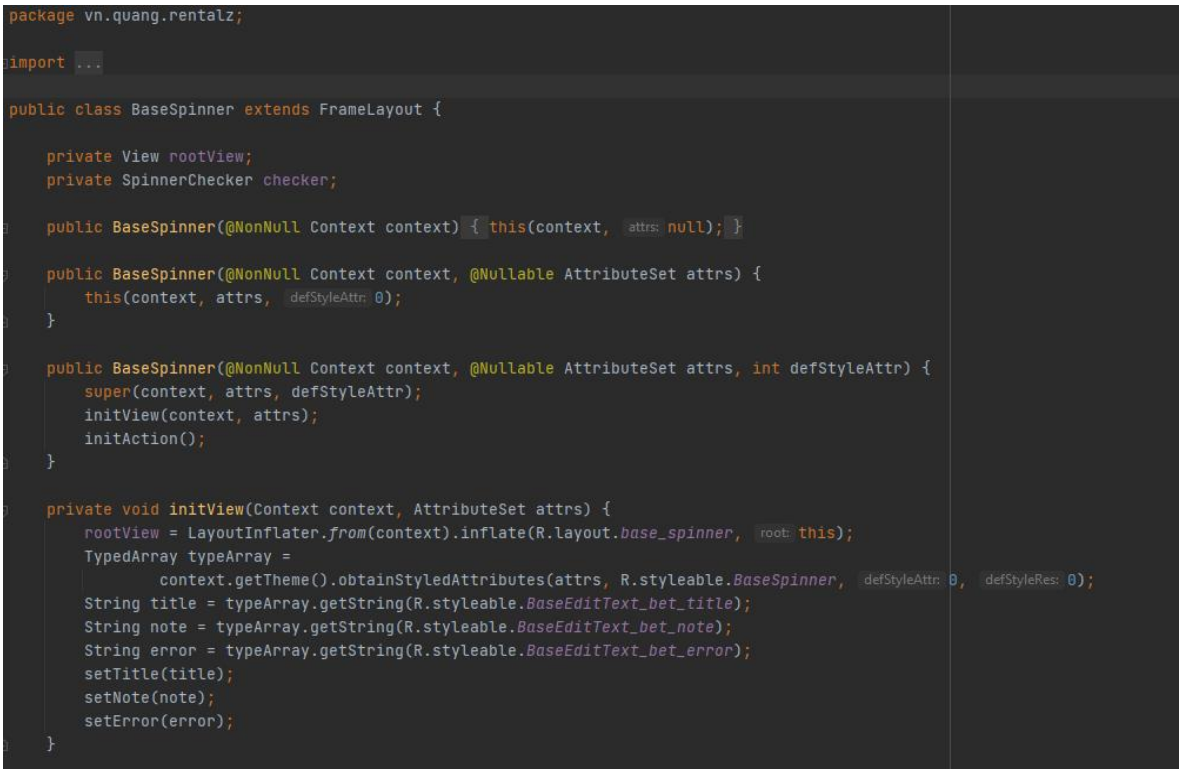
Figure 25. Features a) on native Android app

			
General Submit Validation	Bedrooms Validation	Property Type Validation	Price Validation



Reporter Validation

Figure 26. Features b) on native Android app



```
package vn.quang.rentalz;

import ...

public class BaseSpinner extends FrameLayout {

    private View rootView;
    private SpinnerChecker checker;

    public BaseSpinner(@NonNull Context context) { this(context, attrs: null); }

    public BaseSpinner(@NonNull Context context, @Nullable AttributeSet attrs) {
        this(context, attrs, defStyleAttr: 0);
    }

    public BaseSpinner(@NonNull Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        initView(context, attrs);
        initAction();
    }

    private void initView(Context context, AttributeSet attrs) {
        rootView = LayoutInflater.from(context).inflate(R.layout.base_spinner, root: this);
        TypedArray typeArray =
                context.getTheme().obtainStyledAttributes(attrs, R.styleable.BaseSpinner, defStyleAttr: 0, defStyleRes: 0);
        String title = typeArray.getString(R.styleable.BaseEditText_bet_title);
        String note = typeArray.getString(R.styleable.BaseEditText_bet_note);
        String error = typeArray.getString(R.styleable.BaseEditText_bet_error);
        setTitle(title);
        setNote(note);
        setError(error);
    }
```

Figure 27. Evidence proves code Java on native Android App

Feature g) No additional features implemented

V. Application Evaluation:

Overall, this is an incomplete system. The system was developed with the main purpose of focusing on handling user data and skipping the eye-catching interface design phase. The main functions are also developed very closely to the requirements in the coursework and logbook. Do not develop optional functions to shorten the project completion time. Although the comprehensiveness is low, the project still has a certain chance to be upgraded and developed. The most difficult part is that the business logic processing code has been well developed, in the future only need to focus on decorating the interface, adding some extra functions to increase the collection and interaction with the user, bringing provide the best user experience.