

Mobile Application Development – Final Evaluation **Group Project – 2019 Regular Batch**

Group ID:

Student ID	Integrated System using a Repository 05 Marks		Marketability(Presentation about the app) – 05 marks		Coding Standards- 05 Marks	Function Completeness (Insert, Update, Delete, Search) or (Level implementation, using sensors, Settings) - 25 Marks	Database Connectivity – 05 Marks	Validations - 10 Marks	Usability - 05 marks	Professionalism – 05 Marks	Out of Box features - 05 Marks	Q And A - 25 Marks	Consistency – 05 marks	Total(out of 100)

 Evaluator's Signature

Login & Register Function

➤ IT18131598 - M. A. S. P. Vimukthi

- Everyone have access to this page. When a registered student or an owner logs into the system, it will displays the “Home” screen. If the user does not registered into the system, it will displays the “Register” page. After registering to the system user can be seen the “Login” page. And then they can give their email and password and can have access to the system. User can see the dashboard at the bottom of the interface. If the user is a student, they can find places, can give compliments about the hostel as well as they can pay for hostels by using this app.
- If the user is an owner he/she can click the places icon in the dashboard and it will display the all advertisements relevant to the hostels. And owner can delete or add advertisements if they want.

○ Find

```
package org.fireengine.mvp.presenter.user;

import ...

public class UserInfoFragmentPresenter implements UserInfoFragmentContract.Presenter {
    private WeakReference<UserInfoFragmentContract.View> view;

    private Database database;

    private Callback<List<Map<String, Object>>> findCallback = new Callback<List<Map<String, Object>>>() {
        @Override
        public void onSuccess(List<Map<String, Object>> data) {
            view.get().updateTextViews(
                data.get(0).get("user_display_name").toString(),
                data.get(0).get("user_type").toString()
            );
        }

        @Override
        public void onFailure() {
        }
    };

    public UserInfoFragmentPresenter(UserInfoFragmentContract.View view, Injector injector) {
        this.view = new WeakReference<>(view);
        this.database = injector.getDatabaseInstance(new UserModel());
    }

    @Override
    public void onFragmentCreated(String userId) {
        database.where( column: "user_uid", userId, findCallback);
    }
}
```

UserInfoFragmentPresenter

```

private Database database;

private Callback<List<Map<String, Object>>> findCallback = new Callback<List<Map<String, Object>>>() {
    @Override
    public void onSuccess(List<Map<String, Object>> data) {
        view.get().updateTextViews(
            data.get(0).get("user_display_name").toString(),
            data.get(0).get("user_type").toString()
        );
    }

    @Override
    public void onFailure() {
    }
};

public UserInfoFragmentPresenter(UserInfoFragmentContract.View view, Injector injector) {
    this.view = new WeakReference<>(view);
    this.database = injector.getDatabaseInstance(new UserModel());
}

@Override
public void onFragmentCreated(String userId) {
    database.where( column: "user_uid", userId, findCallback);
}
}

```

UserInfoFragmentPresenter

Add Place Function

➤ IT18132588 - W. M. W. Lakshan Wijewardana

- Students and owners have access to this function. Mainly this function gives an idea about hostels near to a relevant university. And also hostels owners can promote their hostels by displaying advertisements. When student clicks “Places” advertisement , they can see the details of hostels like, location, boarding fees, features, availability and the contact number that they have searched.
- If owner wants to add details about hostel, they can add details clicking “ADD” button. Moreover that owners can edit non relatable details and delete advertisement that uploaded into the system previously by clicking “DELETE” button.

○ Update

```
package org.fireengine.mvp.view.place;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import org.fireengine.mvp.R;
import org.fireengine.mvp.contract.place.EditPlaceActivityContract;
import org.fireengine.mvp.dependency.Injector;
import org.fireengine.mvp.presenter.place.EditPlaceActivityPresenter;

public class EditPlaceActivity extends AppCompatActivity implements EditPlaceActivityContract.View {
    private EditPlaceActivityContract.Presenter presenter;

    private ImageView placePhoto;
    private EditText placeName;
    private TextView landlordName;
    private EditText placePrice;
    private RadioGroup placeAvailable;
    private RadioButton placeAvailableYes;
    private RadioButton placeAvailableNo;
    private EditText placeAddress;
    private EditText placePhone;
    private EditText placeFeatures;

    @Override
    EditPlaceActivity > getCheckedItem()
```

```

    }

    @Override
    public void updateFormElements(Object placeName, Object landlordName, Object placePrice, Object placeAvailable,
                                   Object placeAddress, Object placePhone, Object placeFeatures) {

        this.placeName.setText(placeName.toString());
        this.landlordName.setText(landlordName.toString());
        this.placePrice.setText(placePrice.toString());
        findCheckedItem(placeAvailable.toString());
        this.placeAddress.setText(placeAddress.toString());
        this.placePhone.setText(placePhone.toString());
        this.placeFeatures.setText(placeFeatures.toString());
    }

    @Override
    public void finishActivity() { finish(); }

    public void onConfirmEdit(View view) {
        presenter.onEditButtonClicked(
            placeName.getText().toString(),
            placePrice.getText().toString(),
            getCheckedItem(placeAvailable),
            placeAddress.getText().toString(),
            placePhone.getText().toString(),
            placeFeatures.getText().toString()
        );
    }

    private void findCheckedItem(String value) {
        switch (value) {
            case "Available":
                placeAvailableYes.setChecked(true);
                break;
            case "Not Available":
                break;
        }
    }
}

```

EditPlaceActivity > updateFormElements()

```

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_place);

        presenter = new EditPlaceActivityPresenter( view: this, new Injector());

        placePhoto = findViewById(R.id.place_photo_edit);
        placeName = findViewById(R.id.place_name_edit);
        landlordName = findViewById(R.id.landlord_name_place_edit);
        placePrice = findViewById(R.id.place_price_edit);
        placeAvailable = findViewById(R.id.place_available_edit);
        placeAvailableYes = findViewById(R.id.place_available_yes_edit);
        placeAvailableNo = findViewById(R.id.place_available_no_edit);
        placeAddress = findViewById(R.id.place_address_edit);
        placePhone = findViewById(R.id.place_phone_edit);
        placeFeatures = findViewById(R.id.place_features_edit);

        Intent intent = getIntent();
        String id = intent.getStringExtra( name: "place_id");

        presenter.onActivityCreated(id);
    }

    @Override
    public void sendMessage(String message) {
        Toast.makeText( context: this, message, Toast.LENGTH_LONG).show();
    }
}

```

```

public void finishActivity() { finish(); }

public void onConfirmEdit(View view) {
    presenter.onEditButtonClicked(
        placeName.getText().toString(),
        placePrice.getText().toString(),
        getCheckedItem(placeAvailable),
        placeAddress.getText().toString(),
        placePhone.getText().toString(),
        placeFeatures.getText().toString()
    );
}

private void findCheckedItem(String value) {
    switch (value) {
        case "Available":
            placeAvailableYes.setChecked(true);
            break;
        case "Not Available":
            placeAvailableNo.setChecked(true);
            break;
    }
}

private String getCheckedItem(RadioGroup group) {
    switch (group.getCheckedRadioButtonId()) {
        case R.id.place_available_yes_edit:
            return "Available";
        case R.id.place_available_no_edit:
            return "Not Available";
    }
    return "";
}

```

EditPlaceActivity > updateFormElements()

Payment Function

➤ IT18136784 - B. W. H. K. Senevirathna

- Both Student and Owner can access this payment function. In this function first students can see the “Add Payment Details View”. Using this interface student can enter their payment details. Such as, Student Id , Landlord Id , Place Id , Payment type , Payment method and Payment amount. Once student press the ‘CONFORM’ button data is stored in the database and student can see their payment list. (“Payment List View”). It include the payment Id and the payment amount of all the payment which is done by them. If they click one of the record in a list it will navigate to the “Payment Details view”. In this interface student can see all the details that are related to that particular payment. Within this interface student can edit their payment details by clicking “ EDIT “ button. Once student press this button it will navigate to “ Edit Payment Details view “ and student can save updated details by clicking “ SAVE “ button.
- First, Owner can see the “Payment List View”. This include Payment Id and Payment amount of all the payment done by student that are related to the particular owner. If owner click one of the record in a list it will navigate to the “Payment Details view”. In this interface owner can see all the details that are related to that particular payment.

○ Add details

```
package org.fireengine.mvp.view.payment;

import ...

public class AddPaymentActivity extends AppCompatActivity implements AddPaymentActivityContract.View {
    private AddPaymentActivityContract.Presenter presenter;

    private TextView placeName;
    private TextView studentName;
    private TextView landlordName;
    private EditText paymentAmount;
    private Spinner paymentMethod;
    private Spinner paymentDescription;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_payment);

        presenter = new AddPaymentActivityPresenter( view: this, new Injector());

        placeName = findViewById(R.id.place_name_payment_add);
        studentName = findViewById(R.id.student_name_payment_add);
        landlordName = findViewById(R.id.landlord_name_payment_add);
        paymentAmount = findViewById(R.id.payment_amount_add);
        paymentMethod = findViewById(R.id.payment_method_add);
        paymentDescription = findViewById(R.id.payment_description_add);

        Intent intent = getIntent();
        String userId = intent.getStringExtra( name: "user_id");
        String placeId = intent.getStringExtra( name: "place_id");

        presenter.onActivityCreated(userId, placeId);
    }
}
```

```

        landlordName = findViewById(R.id.landlord_name_payment_add);
        paymentAmount = findViewById(R.id.payment_amount_add);
        paymentMethod = findViewById(R.id.payment_method_add);
        paymentDescription = findViewById(R.id.payment_description_add);

        Intent intent = getIntent();
        String userId = intent.getStringExtra( name: "user_id");
        String placeId = intent.getStringExtra( name: "place_id");

        presenter.onActivityCreated(userId, placeId);
    }

    @Override
    public void sendMessage(String message) {
        Toast.makeText( context: this, message, Toast.LENGTH_LONG).show();
    }

    @Override
    public void updateTextViews(String placeName, String studentName, String landlordName) {
        this.placeName.setText(placeName);
        this.studentName.setText(studentName);
        this.landlordName.setText(landlordName);
    }

    @Override
    public void finishActivity() { finish(); }

    public void onConfirmAddPayment(View view) {
        presenter.onAddButtonClicked(
            paymentAmount.getText().toString(), paymentMethod.getSelectedItem().toString(), paymentDescription.getSelectedItem().toString()
        );
    }
}

```

AddPaymentActivity

Review Function

➤ IT18133714 – I.U. Wanaguru

- Both students and owners can access this review function. When someone login as a student, they can see the “Review” interface. In there they can give their compliments simply clicking by “Add review”. After commenting they can submit their comment about hostels by clicking “Submit” button. Furthermore students can edit their comments or delete their comments what have submitted by them if they want . When deleting a comment, the system will generate the notification to confirm that that they really want to delete that comment.
- As an owner they can get compliments of students who have visited to their hostel by clicking “Review View List” and can see comments about other hostels as well.

○ Delete

```
package org.fireengine.mvp.view.review;

import ...

public class DeleteReviewActivity extends AppCompatActivity implements DeleteReviewActivityContract.View {
    private DeleteReviewActivityContract.Presenter presenter;

    private TextView reviewTitle;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_delete_review);

        presenter = new DeleteReviewActivityPresenter( view: this, new Injector());

        reviewTitle = findViewById(R.id.review_title_delete);

        Intent intent = getIntent();
        String id = intent.getStringExtra( name: "review_id");
        presenter.onActivityCreated(id);
    }

    @Override
    public void updateTextView(String reviewTitle) { this.reviewTitle.setText(reviewTitle); }

    @Override
    public void finishActivity() { finish(); }

    public void onConfirmDeleteReview(View view) { presenter.onReviewDeleteButtonClicked(); }
}
```

DeleteReviewActivity > onCreate()