KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

T120B169 App Development for Smart Mobile Systems

Car Deal

Group, Name and Surname:
IFU-8 Pedro García-Cañal Sánchez
IFU-8 José Antonio Pérez Miguel
IFE-8 Edvinas Kulboka
IFU-8 Mohanad Kher

Date: *18/10/2021*

Kaunas, 2021

Tables of Contents

Description of Your app

1. **What type is your application/game?**
   Car sales app

2. **Description.**

   This car sales application will help users upload pictures, a description, defects, the specification of the car the user wants to sell. The goal of this application is to make it simpler and faster to create adverts for a car sale and to keep track of these adverts. It will use a database to store sales and users that are using the application. Notifications will be used to inform users if any new sales get created in the application, that the user might be interested in. The app will also show sales from other sources, by parsing ad sites. Additionally, each advert for a car sale can be shared to friends or followers using social media.

Functionality of your app

List of functions (adapt to your own app)

1- Login
2- Register
3- Recycler view(view the list of users from database)

Solution

Task #1. Login

Description of the implementation .

The application's first screen is the login window where you can sign in if you already have an account, if you don't you can click on the "I don't have an account…" button and the application will send you to the registration window.

To login, email and password will be required.

UI components:

- Application Logo (Image View).
- Email (EditText).
- Password (EditText).
- Login (Button).
- I don't have an account (Button).

All these components are inside of a linear layout.

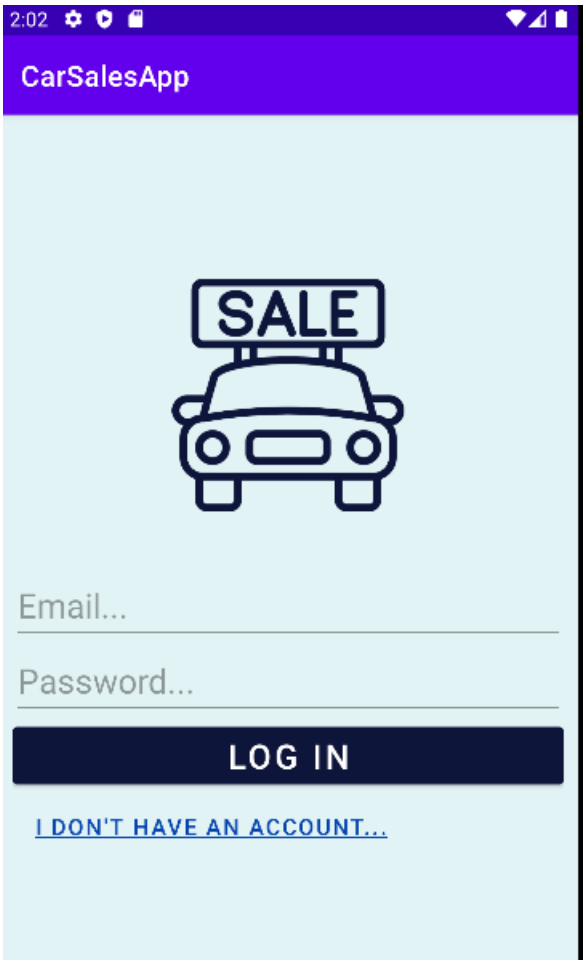


Figure 1. Login

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    emailInput = findViewById(R.id.emailInput);
    passwordInput = findViewById(R.id.passwordInput);
    toRegister = findViewById(R.id.toRegisterBtn);
    loginButton = findViewById(R.id.loginBtn);
    toRegister.setOnClickListener(v -> {
        Intent toRegister = new Intent(LoginActivity.this,
RegisterActivity.class);
        startActivity(toRegister);
    });

    userInformationViewModel = new ViewModelProvider
.AndroidViewModelFactory(LoginActivity.this
            .getApplication()).create(UserInformationViewModel.class);

    loginButton.setOnClickListener(view -> {String email =
emailInput.getText().toString().trim();
        String password = passwordInput.getText().toString().trim();
        UserInformation userInformation = new
UserInformation(email,password);
        if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password))
        {
            if (userInformationViewModel.getUser(email)!=null){
                Intent logSuccess = new Intent(LoginActivity.this,
MainActivity.class);
                startActivity(logSuccess);
            }else {

Toast.makeText(this,R.string.failLogin,Toast.LENGTH_SHORT).show();
            }
        }else {
            Toast.makeText(this,R.string.empty,Toast.LENGTH_SHORT).show();
        }
    });
}
```

Task #2. Register

Description of the implementation .

In this screen you can see a form that you have to fill with your information to complete the registration. If you already have an account you can press the “Sign in instead” button and you will be sent to the login. Also when you press the register button to finish with this process, the app will send you to the login too, so you can log in in your newly created account.

UI Components:

- Application Logo (Image View).
- Email (EditText).
- Password (EditText).
- ConfirmPassword (EditText).
- Register (Button).
- Sign in instead (Button).

  All these components are inside of a linear layout.

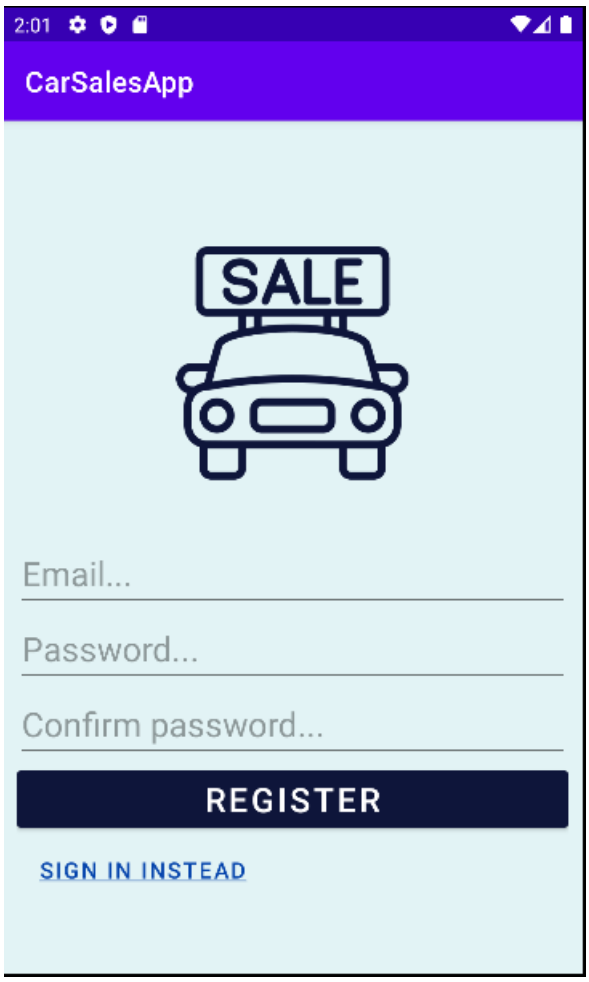


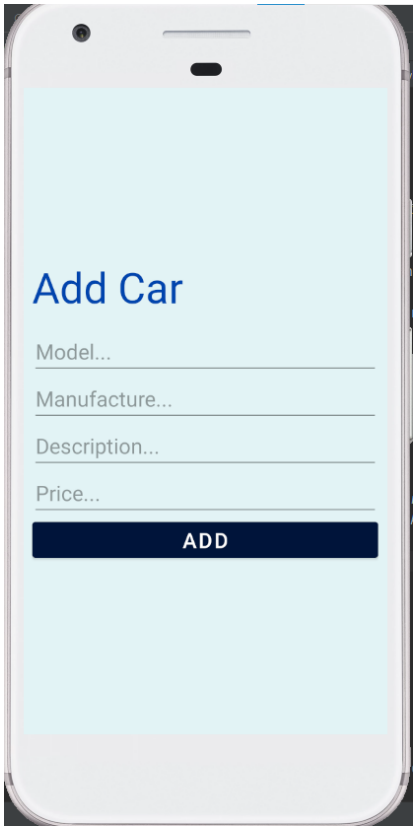
Figure 2. Registration

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    registerButton = findViewById(R.id.registerBtn);
    emailInput = findViewById(R.id.emailInput);
    passwordInput = findViewById(R.id.passwordInput);
    confirmPasswordInput = findViewById(R.id.confPasswordInput);
    toLogin = findViewById(R.id.toLoginBtn);

    userInformationViewModel = new
ViewModelProvider.AndroidViewModelFactory(RegisterActivity.this
            .getApplication()).create(UserInformationViewModel.class);

    toLogin.setOnClickListener(view -> {Intent toLogin = new
Intent(RegisterActivity.this,
        LoginActivity.class);
        startActivity(toLogin);
    });

    registerButton.setOnClickListener(v -> {
        String email = emailInput.getText().toString().trim();
        String password = passwordInput.getText().toString().trim();
        String confPassword =
confirmPasswordInput.getText().toString().trim();

        if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password) &&
!TextUtils.isEmpty(confPassword))
        {
```

```
            if (password.equals(confPassword))
            {
                UserInformation userInformation = new
UserInformation(email,password);
                UserInformationViewModel.insert(userInformation);
                Intent toLogin = new Intent(RegisterActivity.this,
LoginActivity.class);
                startActivity(toLogin);
            }else {

Toast.makeText(this,R.string.passNotMatch,Toast.LENGTH_SHORT).show();
            }
        }else
        {
            Toast.makeText(this,R.string.empty,Toast.LENGTH_SHORT).show();
        }
    });
```

Task #3. Add Car

We created a window where you can add a new car ad. The user can see some inputs to write the model, a description, the manufacturer and the price.

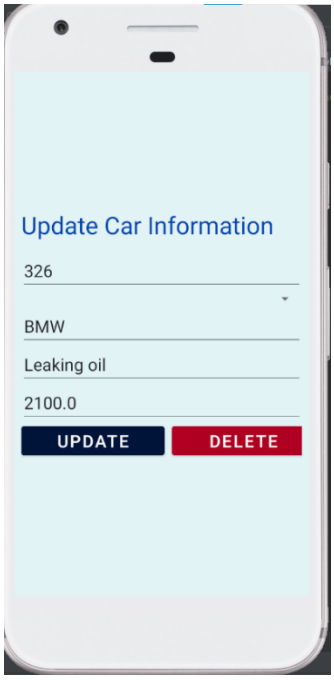



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_car);
    modelInput = findViewById(R.id.modelInput);
    descInput = findViewById(R.id.descInput);
    manufactureInput = findViewById(R.id.manufactureInput);
    priceTx = findViewById(R.id.priceInput);
    carImage = findViewById(R.id.carImage);
    addCarBtn = findViewById(R.id.updateCarBtn);
    selectImageBtn = findViewById(R.id.selectImage);
```

```
    selectImageBtn.setOnClickListener(this);
    carViewModel = new ViewModelProvider
            .AndroidViewModelFactory(AddCarActivity.this.getApplication())
            .create(CarViewModel.class);
    addCarBtn.setOnClickListener(v -> {
        String model = modelInput.getText().toString().trim();
        String description = descInput.getText().toString().trim();
        String price = priceTx.getText().toString().trim();
        Double priceVal = Double.parseDouble(priceTx.getText().toString());
        String manufacturer = manufactureInput.getText().toString().trim();
        Bundle data = getIntent().getExtras();
        if(data!=null){FK = data.getString(MainActivity.USER_EMAIL);}
        Log.d("FK", "onCreate:FK " + FK);
        if (!TextUtils.isEmpty(model) && !TextUtils.isEmpty(description) &&
!TextUtils.isEmpty(price)
                && !TextUtils.isEmpty(manufacturer))
        {
            CarEntity carEntity = new
CarEntity(model,manufacturer,description,priceVal,FK);
            CarViewModel.insert(carEntity);
            Intent carAdded = new Intent(AddCarActivity.this,
MainActivity.class);
            startActivity(carAdded);
        }else
        {
            Toast.makeText(this,R.string.empty,Toast.LENGTH_SHORT).show();
        }
    });

    Bundle data = getIntent().getExtras();
    if(data!=null){
        int id = data.getInt(MainActivity.CAR_ID);
        carViewModel.get(id).observe(this, carEntity -> {
            modelInput.setText(carEntity.getModel());
            descInput.setText(carEntity.getDescription());
            manufactureInput.setText(carEntity.getManufacturer());
            priceTx.setText(carEntity.getPrice().toString());
        });
    }
}
```

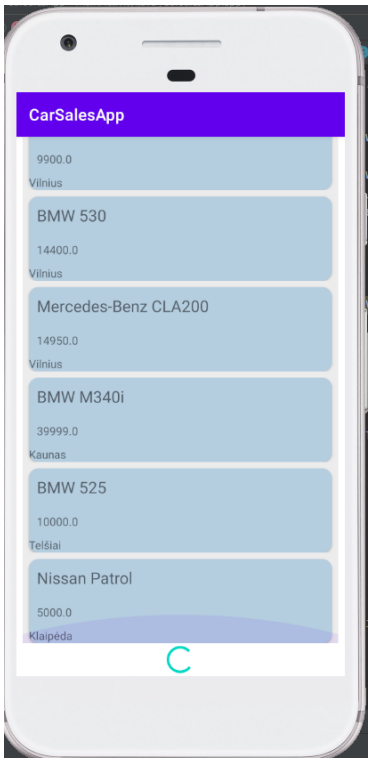
Task #4. Update Car Information

In this window you can change an ad information or delete it.




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

    modelInput = findViewById(R.id.modelInput);
    descInput = findViewById(R.id.descInput);
    manufactureInput = findViewById(R.id.manufactureInput);
    priceTx = findViewById(R.id.priceInput);
    updateCarBtn = findViewById(R.id.updateCarBtn);
    deleteCarBtn = findViewById(R.id.deleteCarBtn);

    carViewModel = new ViewModelProvider

.AndroidViewModelFactory(UpdateCarInfoActivity.this.getApplication())
            .create(CarViewModel.class);

    if(getIntent().hasExtra(MainActivity.CAR_ID)){
        carId = getIntent().getIntExtra(MainActivity.CAR_ID,0);
        carViewModel.get(carId).observe(this, carEntity -> {
            if (carEntity!=null){
                modelInput.setText(carEntity.getModel());
                descInput.setText(carEntity.getDescription());
                manufactureInput.setText(carEntity.getManufacturer());
                priceTx.setText(carEntity.getPrice().toString());
            }
        });
        isEdit = true;
    }
    deleteCarBtn.setOnClickListener(view -> {
        edit(true);
    });
    updateCarBtn.setOnClickListener(view -> {
        edit(false);
    });

}
```

Task #5. HTTP Scraping




```
public class MainActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private RecyclerViewAdapter recyclerViewAdapter;
    private CarInformationViewModel carInformationViewModel;
    private RequestQueue requestQueue;
    private int PageNumber = 1;
    private LinearLayoutManager manager;
    private ProgressBar progressBar;

    private boolean isButtonAvailable = false;
    private boolean isScrolling = false;
    int currentItems, totalItems, scrollOutItems;


    private static final int InternetRequestCode = 1;

    Button loadMoreVehiclesButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestQueue = Volley.newRequestQueue(MainActivity.this);
        setContentView(R.layout.activity_main);
        manager = new LinearLayoutManager(this);

        progressBar = findViewById(R.id.progressBar);
        recyclerView = findViewById(R.id.recyclerViewId);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(manager);
        recyclerView.addOnScrollListener(new RecyclerView.OnScrollListener()
{
```

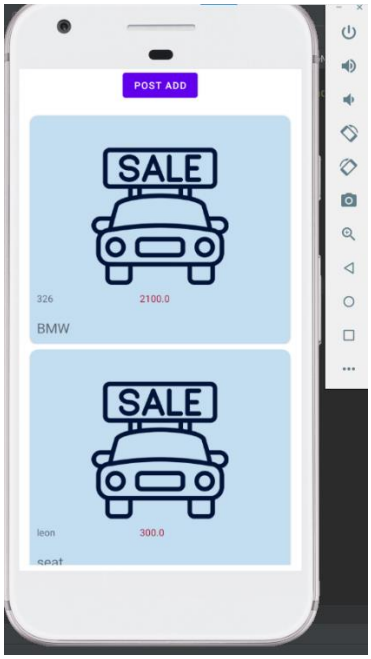
```
            @Override
            public void onScrollStateChanged(RecyclerView recyclerView, int
newState) {
                super.onScrollStateChanged(recyclerView, newState);
                if (newState ==
AbsListView.OnScrollListener.SCROLL_STATE_TOUCH_SCROLL)
                    isScrolling = true;
            }

            @Override
            public void onScrolled(RecyclerView recyclerView, int dx, int dy)
{
                super.onScrolled(recyclerView, dx, dy);

                currentItems = manager.getChildCount();
                totalItems = manager.getItemCount();
                scrollOutItems = manager.findFirstVisibleItemPosition();

                if (isScrolling && (currentItems + scrollOutItems ==
totalItems) && !isButtonAvailable)
                {
                    isScrolling = false;
                    QueueRequestToOutsources();
                }
            }
        });


        carInformationViewModel = new ViewModelProvider
                .AndroidViewModelFactory(MainActivity.this.getApplication())
                .create(CarInformationViewModel.class);

        carInformationViewModel
                .getCars()
                .observe(this, carInformation -> {
                    // Set the adapter
                    recyclerViewAdapter = new
RecyclerViewAdapter(carInformation,MainActivity.this);
                    recyclerView.setAdapter(recyclerViewAdapter);
                });

        loadMoreVehiclesButton = findViewById(R.id.loadMoreVehiclesButton);
        loadMoreVehiclesButton.setOnClickListener(v -> {
            loadMoreVehiclesButtonPressed();
        });
    }

    private void loadMoreVehiclesButtonPressed()
    {
        if (ContextCompat.checkSelfPermission( this,
Manifest.permission.INTERNET) == PackageManager.PERMISSION_GRANTED)
        {
            RemoveMoreVehiclesButton();
            QueueRequestToOutsources();
        }
        else
            ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.INTERNET}, InternetRequestCode);
    }

    @SuppressLint("MissingSuperCall")
    @Override
    public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults)
```

```
    {
        switch (requestCode)
        {
            case InternetRequestCode:
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)
                    QueueRequestToOutsources();
                else
                    RemoveMoreVehiclesButton();

                return;
        }
    }

    private void RemoveMoreVehiclesButton()
    {
        ViewGroup parentView = (ViewGroup)
loadMoreVehiclesButton.getParent();
        parentView.removeView(loadMoreVehiclesButton);
        isButtonAvailable = false;
    }

    private void QueueRequestToOutsources()
    {
        progressBar.setVisibility(View.VISIBLE);
        String request = "https://en.autoplius.lt/ads/used-
cars?order_by=3&order_direction=DESC&page_nr=" + PageNumber;// for further
requests add &page_nr=2
        PageNumber = PageNumber + 1;
        StringRequest stringRequest = new StringRequest(Request.Method.GET,
request,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        ParseVehicles(response);
                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        });

        requestQueue.add(stringRequest);
    }

    private void ParseVehicles(String response)
    {
        Scanner scanner = new Scanner(response);
        boolean started = false;
        List list = new ArrayList();
        while(scanner.hasNextLine())
        {
            String line = scanner.nextLine();
            if (started)
            {
                if(line.contains("</a>"))
                {
                    try{
                        AddCarFromStrings(list);
                    }
                    catch (Exception ex) {
                        Toast.makeText(this, "Failed to load vehicle",
Toast.LENGTH_SHORT);
```

```
                    }
                    list.clear();
                }
                else if(!line.trim().isEmpty())
                    list.add(line.trim());
            }
            else if (line.contains("<div class=\"list-items\">"))
                started = true;
        }
        progressBar.setVisibility(View.GONE);
        recyclerViewAdapter.notifyDataSetChanged();
    }

    private void AddCarFromStrings(List<String> list)
    {
        String description = "";
        Double price = 0.0;
        String city = "";
        String link = "";
        String imageLink = "";

        for(int i = 0; i<list.size(); i++)
        {
            String line = list.get(i);
            if (line.contains("<div class=\"line1\">"))
            {
                String descriptionLine = list.get(i+1).trim();
                if (descriptionLine.contains("</div>"))
                    descriptionLine = descriptionLine.replace("</div>", "");
                description = descriptionLine.trim();
            }
            else if (line.contains("<div class=\"pricing-container\">"))
            {
                String priceString = list.get(i+2).trim();
                price = Double.parseDouble(priceString.substring(0,
priceString.indexOf("&")-1).replace(" ", ""));
            }
            else if (line.contains("<div class=\"item-parameters\">"))
            {
                String gasLine = list.get(i+2);
                String gasType = gasLine.substring(0, gasLine.length() -
7).trim();

                String volumeAndPowerLine = list.get(i+4);
                String volumeAndPower = volumeAndPowerLine.substring(0,
volumeAndPowerLine.length() - 7).trim();

                String gearboxLine = list.get(i+5);
                String gearbox = gearboxLine.substring(6,
gearboxLine.length() - 7);

                String kilometrageLine = list.get(i+6);
                String kilometrage = kilometrageLine.substring(6,
kilometrageLine.length() - 7);

                String bodyTypeLine = list.get(i+7);
                String bodyType = bodyTypeLine.substring(6,
bodyTypeLine.length() - 7);

                String cityLine = list.get(i+8).trim();
                city = cityLine.substring(6,cityLine.length()-7);
            }
            else if(line.contains("href"))
            {
```

```
                String trimmed = line.trim();
                link = trimmed.substring(trimmed.indexOf("=")+2,
trimmed.length() - 1);
            }
            else if(line.contains("img class"))
            {
                String identifier = "src=\"";
                imageLink =
line.substring(line.indexOf(identifier)+identifier.length()-1, line.length()
-  8);
            }
        }
        if (price != 0.0 && !description.isEmpty())
        {
            CarInformation carInfo = new CarInformation(description, city,
price, link, imageLink);

            recyclerViewAdapter.addItem(carInfo);
        }
    }

    @Override
    protected void onStop()
    {
        super.onStop();
        if(requestQueue != null)
            requestQueue.cancelAll(InternetRequestCode);
    }
}
```

Task #6. Design for the ads cards




```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addPost = findViewById(R.id.addCadrId);
        recyclerView = findViewById(R.id.recyclerViewId);
        recyclerView.setHasFixedSize(true);
```

```
        carViewModel = new ViewModelProvider
                .AndroidViewModelFactory(MainActivity.this.getApplication())
                .create(CarViewModel.class);

        carViewModel.getCars().observe(this, carEntities -> {
            // Set the adapter
            recyclerViewAdapter = new
RecyclerViewAdapter(carEntities,MainActivity.this,this);
            recyclerView.setAdapter(recyclerViewAdapter);
        });

        Bundle userData = getIntent().getExtras();
        if(userData!=null){
            CarFK = userData.getString(LoginActivity.USER_EMAIL);
        }
        addPost.setOnClickListener(view -> {
            Intent addCar = new
Intent(MainActivity.this,AddCarActivity.class);
            startActivity(addCar);
        });
    }
```

Task #e. Recycler View

For this to work we created to classes as shown below in the code.

Class RecyclerViewAdapter., and ViewHolder within it.

```
public class RecyclerViewAdapter extends RecyclerView.Adapter
<RecyclerViewAdapter.ViewHolder>{
    private List<UserInformation> userList;
    private Context context;

    public RecyclerViewAdapter(List<UserInformation> userList, Context
context) {
        this.userList = userList;
        this.context = context;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {+
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.recycler_row,parent,false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        UserInformation userInformation =
Objects.requireNonNull(userList.get(position));
        holder.email.setText(userInformation.getEmail());
    }

    @Override
    public int getItemCount() {
        return Objects.requireNonNull(userList.size());
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
```

```
        public TextView email;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            email = itemView.findViewById(R.id.recyclerEmailId);
        }
    }
}
```

And in the MainActivity we get the data from the database and pass them to the recycler view adapter and then to the recycler view as shown in the code.

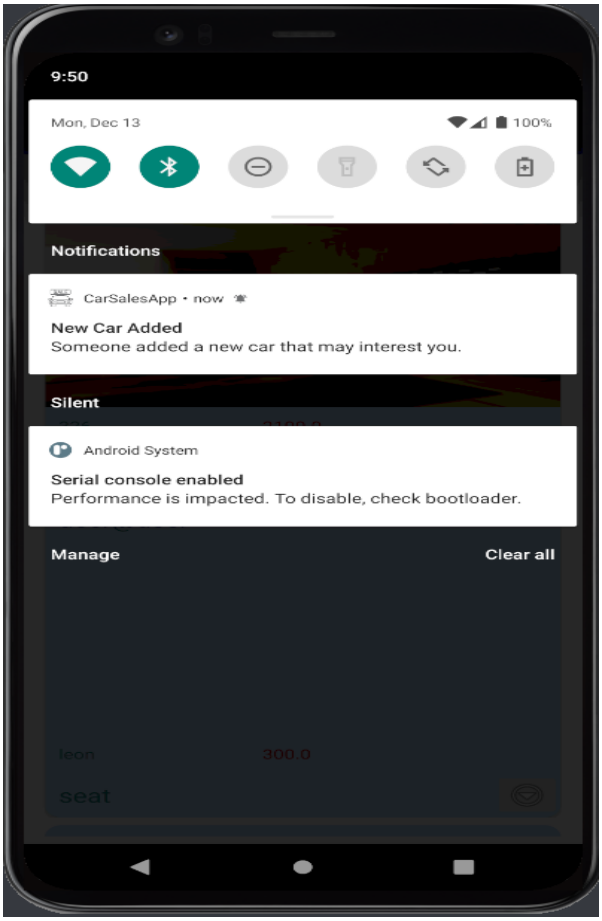
```
     private RecyclerView recyclerView;
    private RecyclerViewAdapter recyclerViewAdapter;
    private UserInformationViewModel userInformationViewModel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        recyclerView = findViewById(R.id.recyclerViewId);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));


         userInformationViewModel = new
ViewModelProvider.AndroidViewModelFactory(MainActivity.this
         .getApplication()).create(UserInformationViewModel.class);

        userInformationViewModel.getAllUsers().observe(this, userInformation
-> {
        // Set the adapter
        recyclerViewAdapter = new
RecyclerViewAdapter(userInformation,MainActivity.this);
        recyclerView.setAdapter(recyclerViewAdapter);
        });
    }
}
```

Task #7. Upload or take photo


```
@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.selectImageBtnId:
            Intent galleryIntent = new Intent(Intent.ACTION_GET_CONTENT);
            galleryIntent.setType("image/*");
            startActivityForResult(galleryIntent, GALLERY_CODE);
            break;
        case R.id.TakePhotoBtnId:
            // take photo from camera
            if(ContextCompat.checkSelfPermission(AddCarActivity .this,
Manifest.permission.CAMERA)!= PackageManager.PERMISSION_GRANTED){
                ActivityCompat.requestPermissions(AddCarActivity.this,new
String[]{
                        Manifest.permission.CAMERA
                },REQUEST_IMAGE_CAPTURE);
            }
            Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(takePicture, REQUEST_IMAGE_CAPTURE);
            break;
    }
}
```

We create 2 buttons that can take a photo or upload a photo when the user is adding a car.

Task #8. Notification



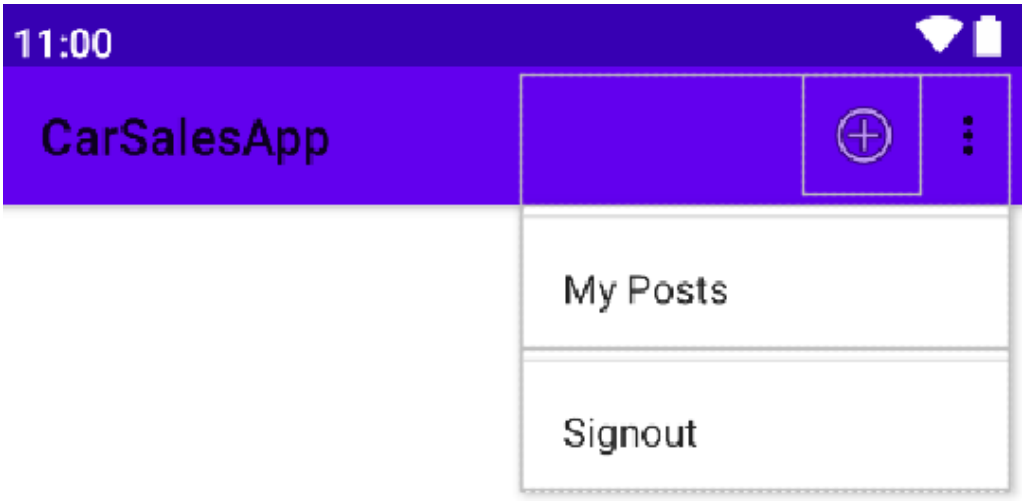

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
    NotificationChannel channel = new NotificationChannel("My Notification",
"My Notification", NotificationManager.IMPORTANCE_DEFAULT);
    NotificationManager manager =
getSystemService(NotificationManager.class);
    manager.createNotificationChannel(channel);
}
```

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "My
Notification");
builder.setContentTitle("New Car Added");
builder.setContentText("Someone added a new car that may interest you.");
builder.setSmallIcon(R.drawable.logo);
builder.setAutoCancel(true);

Intent intent = new Intent(AddCarActivity.this, MainActivity.class);
PendingIntent pendingIntent = PendingIntent.getActivity(AddCarActivity.this
        ,0,intent,PendingIntent.FLAG_UPDATE_CURRENT);
builder.setContentIntent(pendingIntent);

NotificationManagerCompat managerCompat =
NotificationManagerCompat.from(this);
managerCompat.notify(1,builder.build());
```

This notification show to an user when another user add a new add post.

This is created in the addCarActivity.java

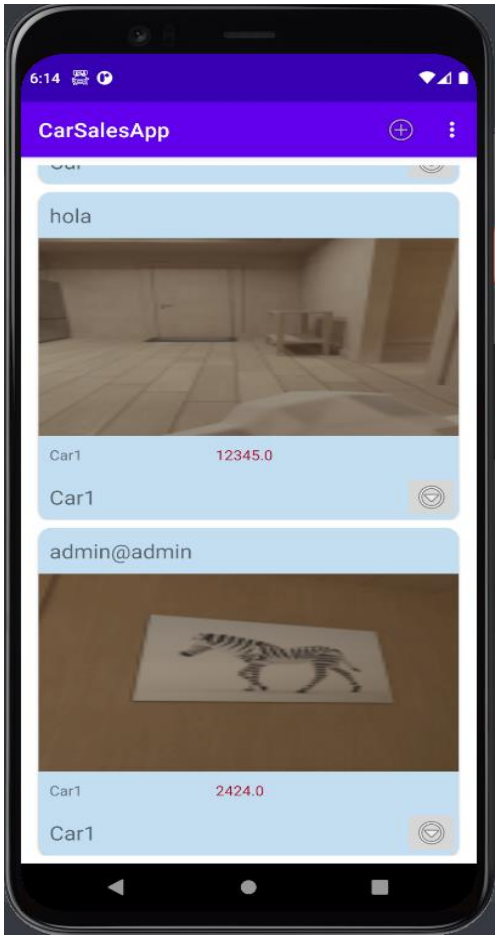
Task #9. Menu




```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
    return super.onCreateOptionsMenu(menu);
}
```

We created a button to add a new car ad, a label to see your own posts and a button to signout.

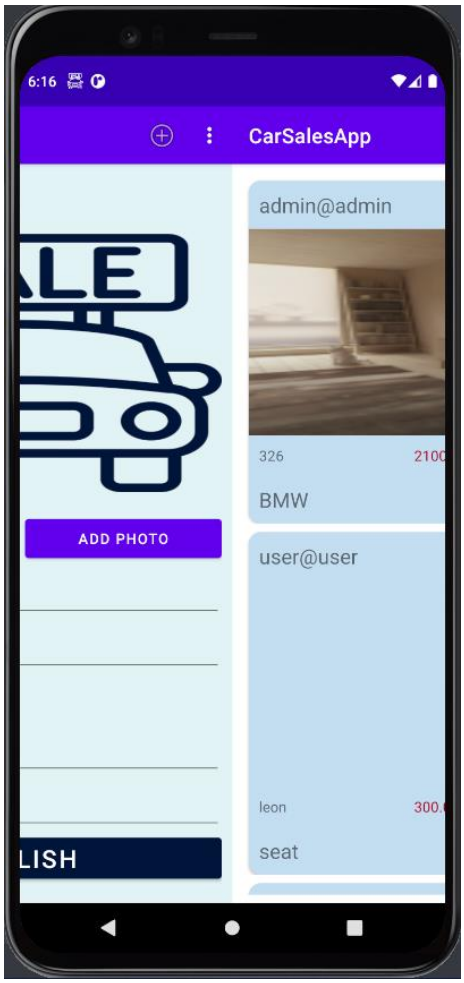
Task #10. Main page photos and upgrading car ad cards




```
@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    CarEntity carEntity = Objects.requireNonNull(carList.get(position));
    holder.Model.setText(carEntity.getModel());
    holder.Manufacturer.setText(carEntity.getManufacturer());
    holder.Price.setText(carEntity.getPrice().toString());
    holder.Owner.setText(carEntity.getFK());

holder.carImage.setImageBitmap(Converters.ByteToBitMap(carEntity.getImage()))
;
}

@Override
public int getItemCount() {
    return Objects.requireNonNull(carList.size());
}

public class ViewHolder extends RecyclerView.ViewHolder{
    public TextView Model;
    public TextView Manufacturer;
    public TextView Price;
    public TextView Owner;
    public ImageView carImage;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        Model = itemView.findViewById(R.id.model);
        Manufacturer = itemView.findViewById(R.id.manufacturer);
        Price = itemView.findViewById(R.id.price);
        Owner = itemView.findViewById(R.id.userName);
```

```
        carImage = itemView.findViewById(R.id.recyclerCarImage);
    }
```

Task #11. Transition animation between activities

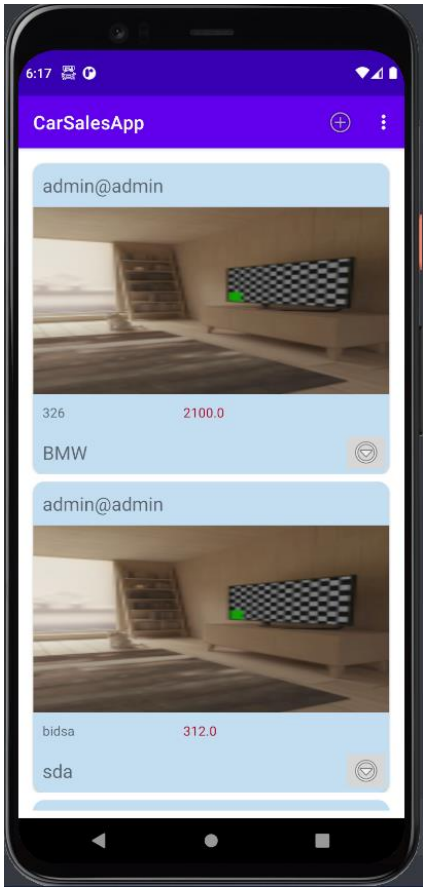



```
<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="@android:integer/config_mediumAnimTime"
        android:fromXDelta="-100%p"
        android:toXDelta="0" />
</set>
```

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

<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="@android:integer/config_mediumAnimTime"
        android:fromXDelta="0"
        android:toXDelta="100%p" />
</set>
```

This transitions are in anim folder

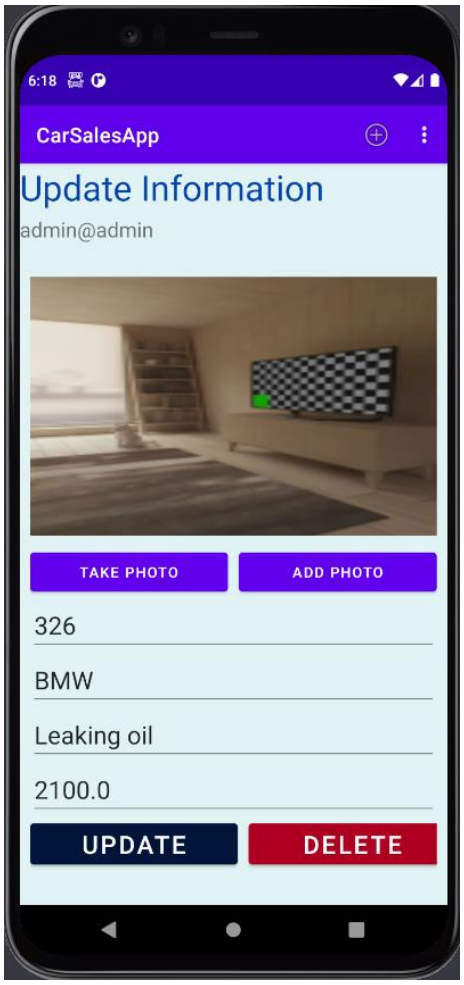
Task #12. MyPost window




```
package com.example.carsalesapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.ViewModelProvider;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;

import com.example.carsalesapp.adapter.CurrentUserRecyclerView;
import com.example.carsalesapp.adapter.RecyclerViewAdapter;
import com.example.carsalesapp.model.CarEntity;
import com.example.carsalesapp.model.UserInformation;
import com.example.carsalesapp.viewmodel.CarViewModel;

import java.util.List;
import java.util.Objects;

import util.CarApi;

public class CurrentUserActivity extends AppCompatActivity  implements
CurrentUserRecyclerView.OnCardClickListener{

    public static final String CAR_ID = "car_id";
    private LiveData<List<UserInformation>> userList;
    private RecyclerView currentUserRecyclerViewId;
```

```
    private CurrentUserRecyclerView currentUserRecyclerView;
    private CarViewModel carViewModel;
    private String currentUser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_current_user);

        currentUserRecyclerViewId =
findViewById(R.id.currentUserRecyclerViewId);
        currentUserRecyclerViewId.setHasFixedSize(true);
        currentUserRecyclerViewId.setLayoutManager(new
LinearLayoutManager(this));

        carViewModel = new ViewModelProvider

.AndroidViewModelFactory(CurrentUserActivity.this.getApplication())
                .create(CarViewModel.class);
        currentUser = CarApi.getInstance().getUserName();
        carViewModel.getAllCurrentUsrCars(currentUser).observe(this,
carEntities -> {
            // Set the adapter
            currentUserRecyclerView = new
CurrentUserRecyclerView(carEntities,CurrentUserActivity.this,this);
            currentUserRecyclerViewId.setAdapter(currentUserRecyclerView);
        });
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        switch (item.getItemId()){
            case R.id.action_add:
                Intent addCar = new Intent(this, AddCarActivity.class);
                startActivity(addCar);
                return true;
            case R.id.action_signout:
                Intent logout = new Intent(this, LoginActivity.class);
                startActivity(logout);
                return true;
            case R.id.myPostsId:
                Intent currentUserActivity = new Intent(this,
CurrentUserActivity.class);
                startActivity(currentUserActivity);
                return true;
        }
        return super.onOptionsItemSelected(item);
    }
    @Override
    public void ocCardClick(int position) {
        CarEntity carEntity =
Objects.requireNonNull(carViewModel.allCurrentUserCars.getValue().get(positio
n));
        Log.d("Tag","onCardClick" + carEntity.getId());
        Intent intent = new
Intent(CurrentUserActivity.this,UpdateCarInfoActivity.class);
        intent.putExtra(CAR_ID,carEntity.getId());
        startActivity(intent);
    }
}
```

Task #13. Fixing delete and update



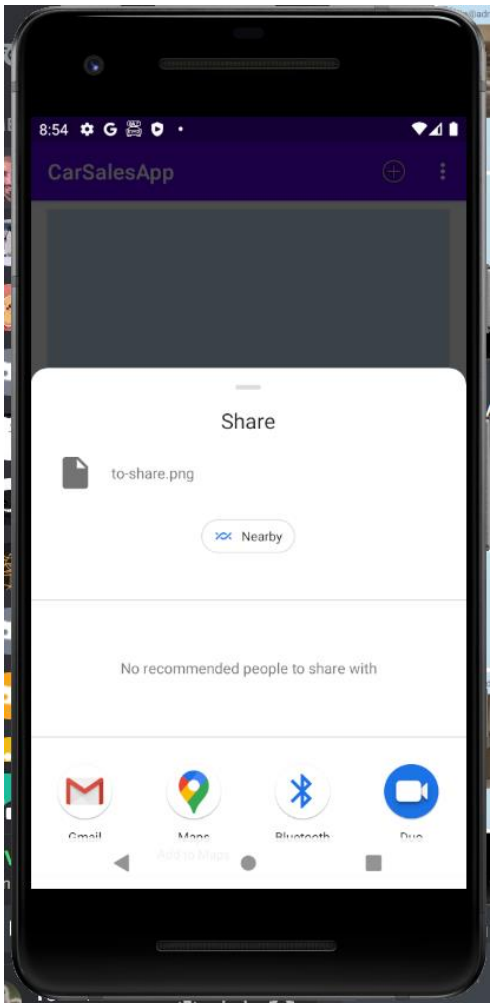

```
package com.example.carsalesapp;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.lifecycle.ViewModelProvider;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.TextUtils;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
```

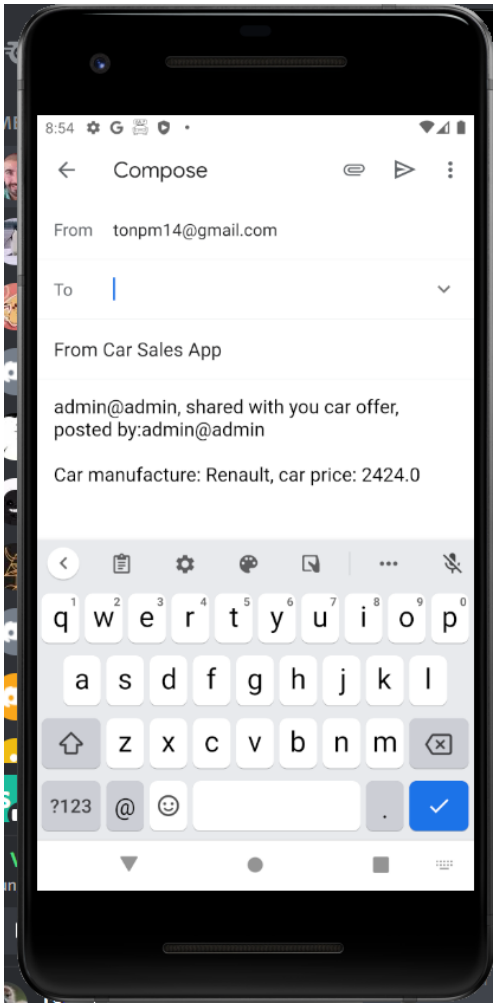
```
import android.widget.TextView;
import android.widget.Toast;

import com.example.carsalesapp.converters.Converters;
import com.example.carsalesapp.model.CarEntity;
import com.example.carsalesapp.viewmodel.CarViewModel;

import util.CarApi;
import util.ImageApi;

public class UpdateCarInfoActivity extends AppCompatActivity {
    public static final int GALLERY_CODE = 0;
    public static final int REQUEST_IMAGE_CAPTURE = 1;
    private EditText modelInput;
    private EditText descInput;
    private EditText manufactureInput;
    private TextView priceTx;
    private Button updateCarBtn;
    private Button deleteCarBtn;
    private TextView userNameIdInUpdate;
    private int carId = 0;
    private Boolean isEdit = false;
    private CarViewModel carViewModel;
    private ImageView updateCarImageId;
    private Button UpdateImageBtnId;
    private Button update_take_image_btn_id;
    Bitmap imageBitmap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_car_info);

        userNameIdInUpdate = findViewById(R.id.userNameIdInUpdate);
        modelInput = findViewById(R.id.modelInput);
        descInput = findViewById(R.id.descInput);
        manufactureInput = findViewById(R.id.manufactureInput);
        priceTx = findViewById(R.id.priceInput);
        updateCarBtn = findViewById(R.id.updateCarBtn);
        deleteCarBtn = findViewById(R.id.deleteCarBtn);
        updateCarImageId = findViewById(R.id.updateCarImageId);
        UpdateImageBtnId = findViewById(R.id.UpdateImageBtnId);
        update_take_image_btn_id =
findViewById(R.id.update_take_image_btn_id);

        carViewModel = new ViewModelProvider

.AndroidViewModelFactory(UpdateCarInfoActivity.this.getApplication())
                .create(CarViewModel.class);

        if(getIntent().hasExtra(MainActivity.CAR_ID)){
            carId = getIntent().getIntExtra(MainActivity.CAR_ID,0);
            carViewModel.get(carId).observe(this, carEntity -> {
                if (carEntity!=null){

userNameIdInUpdate.setText(CarApi.getInstance().getUserName());
                    modelInput.setText(carEntity.getModel());
                    descInput.setText(carEntity.getDescription());
                    manufactureInput.setText(carEntity.getManufacturer());
                    priceTx.setText(carEntity.getPrice().toString());

updateCarImageId.setImageBitmap(Converters.ByteToBitMap(carEntity.getImage())
);
                }
            });
```

```
            isEdit = true;
        }
        deleteCarBtn.setOnClickListener(view -> {
            edit(true);
        });
        updateCarBtn.setOnClickListener(view -> {
            edit(false);
        });

        UpdateImageBtnId.setOnClickListener(view -> {
            Intent galleryIntent = new Intent(Intent.ACTION_GET_CONTENT);
            galleryIntent.setType("image/*");
            startActivityForResult(galleryIntent, GALLERY_CODE);
        });

        update_take_image_btn_id.setOnClickListener(view -> {
            if(ContextCompat.checkSelfPermission(UpdateCarInfoActivity .this,
Manifest.permission.CAMERA)!= PackageManager.PERMISSION_GRANTED){

ActivityCompat.requestPermissions(UpdateCarInfoActivity.this,new String[]{
                        Manifest.permission.CAMERA
                },REQUEST_IMAGE_CAPTURE);
            }
            Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(takePicture, REQUEST_IMAGE_CAPTURE);
        });

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
//        if (requestCode == GALLERY_CODE && resultCode == RESULT_OK){
//            if (data != null){
//                imageUri = data.getData();
//                carImage.setImageURI(imageUri);
//            }
//        }

        switch(requestCode) {
            case 0:
                if(resultCode == RESULT_OK){
                    Uri selectedImage = data.getData();
                    updateCarImageId.setImageURI(selectedImage);
                }

                break;
            case 1:
                if(resultCode == RESULT_OK){
                    Bundle extras = data.getExtras();
                    imageBitmap = (Bitmap) extras.get("data");
                    updateCarImageId.setImageBitmap(imageBitmap);
                }
                break;
        }
    }

    private void edit(Boolean isDelete) {
        int id = carId;
        String model = modelInput.getText().toString().trim();
        String description = descInput.getText().toString().trim();
        String price = priceTx.getText().toString().trim();
        Double priceVal = Double.parseDouble(priceTx.getText().toString());
```

```
        String manufacturer = manufactureInput.getText().toString().trim();
        if (!TextUtils.isEmpty(model) && !TextUtils.isEmpty(manufacturer) &&
!TextUtils.isEmpty(price))
        {
            CarEntity carEntity = new CarEntity();
            carEntity.setId(id);
            carEntity.setModel(model);
            carEntity.setDescription(description);
            carEntity.setPrice(priceVal);
            carEntity.setManufacturer(manufacturer);
            carEntity.setFK(CarApi.getInstance().getUserName());
            updateCarImageId.invalidate();
            BitmapDrawable drawable = (BitmapDrawable)
updateCarImageId.getDrawable();
            Bitmap bitMap = drawable.getBitmap();
            carEntity.setImage(Converters.BitMapToByte(bitMap));
            if (isDelete)
                CarViewModel.delete(carEntity);
            else
                CarViewModel.update(carEntity);
            finish();
        }else
        {
            Toast.makeText(this,R.string.empty,Toast.LENGTH_SHORT).show();
        }
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        switch (item.getItemId()){
            case R.id.action_add:
                Intent addCar = new Intent(this, AddCarActivity.class);
                startActivity(addCar);
                return true;
            case R.id.action_signout:
                Intent logout = new Intent(this, LoginActivity.class);
                startActivity(logout);
                return true;
            case R.id.myPostsId:
                Intent currentUserActivity = new Intent(this,
CurrentUserActivity.class);
                startActivity(currentUserActivity);
                return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Task #14. Share information with other apps






```
                @Override
                public void onClick(View view) {
                  int position = getAdapterPosition();
                    carViewModel = new ViewModelProvider
                            .AndroidViewModelFactory((Application)
context.getApplicationContext())
                            .create(CarViewModel.class);

                    //String imageUrl =
carList.get(position).getImage().toString();
                    //shareImage(imageUrl, ctx, position, carList);
                    StrictMode.VmPolicy.Builder builder = new
StrictMode.VmPolicy.Builder();
                    StrictMode.setVmPolicy(builder.build());
                    if (isExternalStorageWritable()){
                        Intent shareIntent = new Intent(Intent.ACTION_SEND);
                        shareIntent.putExtra(Intent.EXTRA_SUBJECT, "From Car
Sales App");
                        shareIntent.putExtra(Intent.EXTRA_TEXT,
CarApi.getInstance().getUserName()
                                +", shared with you car offer," +" posted
by:"+carList.get(position).getFK()
                                +"\n\n"+"Car manufacture: " +
                                 carList.get(position).getManufacturer() + ",
car price: " +
                                 carList.get(position).getPrice());
```

```
                        shareIntent.setType("image/*");
                        shareIntent.putExtra(Intent.EXTRA_STREAM,
saveImageExternal(Converters.ByteToBitMap(carList.get(position).getImage())))
;

context.startActivity(Intent.createChooser(shareIntent, "Send Image"));
                    }

                }
            });

            seeMoreButton.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    int position = getAdapterPosition();
                    Intent informationIntent = new Intent(context,
PostDetailsActivity.class);

informationIntent.putExtra(CAR_ID,carList.get(position).getId());
                    context.startActivity(informationIntent);
                }
            });
        }
    }
    public boolean isExternalStorageWritable() {
        String state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state)) {
            return true;
        }
        return false;
    }
    private Uri saveImageExternal(Bitmap image) {
        //TODO - Should be processed in another thread
        Uri uri = null;
        try {
            File file = new
File(context.getExternalFilesDir(Environment.DIRECTORY_PICTURES), "to-
share.png");
            FileOutputStream stream = new FileOutputStream(file);
            image.compress(Bitmap.CompressFormat.PNG, 90, stream);
            stream.close();
            uri = Uri.fromFile(file);
        } catch (IOException e) {
            Log.d("TAG", "IOException while trying to write file for sharing:
" + e.getMessage());
        }
        return uri;
    }
```

Defense 1

Tasks:

- Add another table "cars"
- Display it
- It should have manufacturer and model fields.

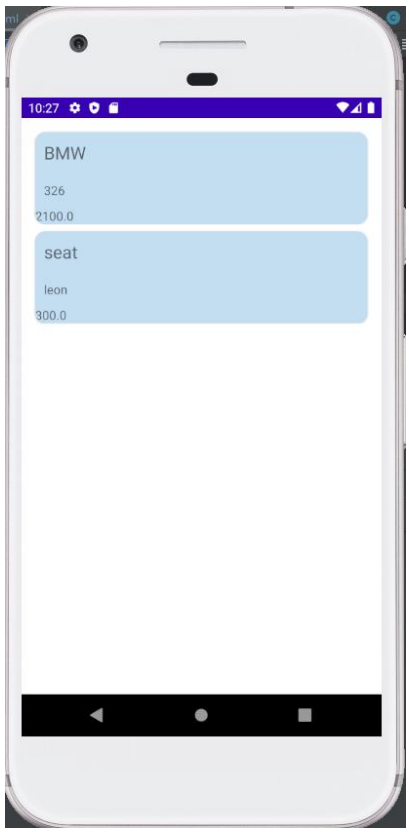


Figure 3 Application showing current cars for sale.

Code:

```
public class MainActivity extends AppCompatActivity {

      private LiveData<List<UserInformation>> userList;
    private RecyclerView recyclerView;
    private RecyclerViewAdapter recyclerViewAdapter;
    private UserInformationViewModel userInformationViewModel;
    private CarInformationViewModel carInformationViewModel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().hide();
        recyclerView = findViewById(R.id.recyclerViewId);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        carInformationViewModel = new ViewModelProvider
                .AndroidViewModelFactory(MainActivity.this.getApplication())
                .create(CarInformationViewModel.class);

        carInformationViewModel.getCars().observe(this, carInformation -> {
            // Set the adapter
            recyclerViewAdapter = new
RecyclerViewAdapter(carInformation,MainActivity.this);
            recyclerView.setAdapter(recyclerViewAdapter);
        });
    }
}
```

Defense 2

Task: Create an animation.

We decided to create a zoom out animation for the Add Post button.

XML animation code.

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">

    <scale
        android:duration="800"
        android:fromXScale="1"
        android:fromYScale="1"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="0"
        android:toYScale="0"
        />
</set>
```

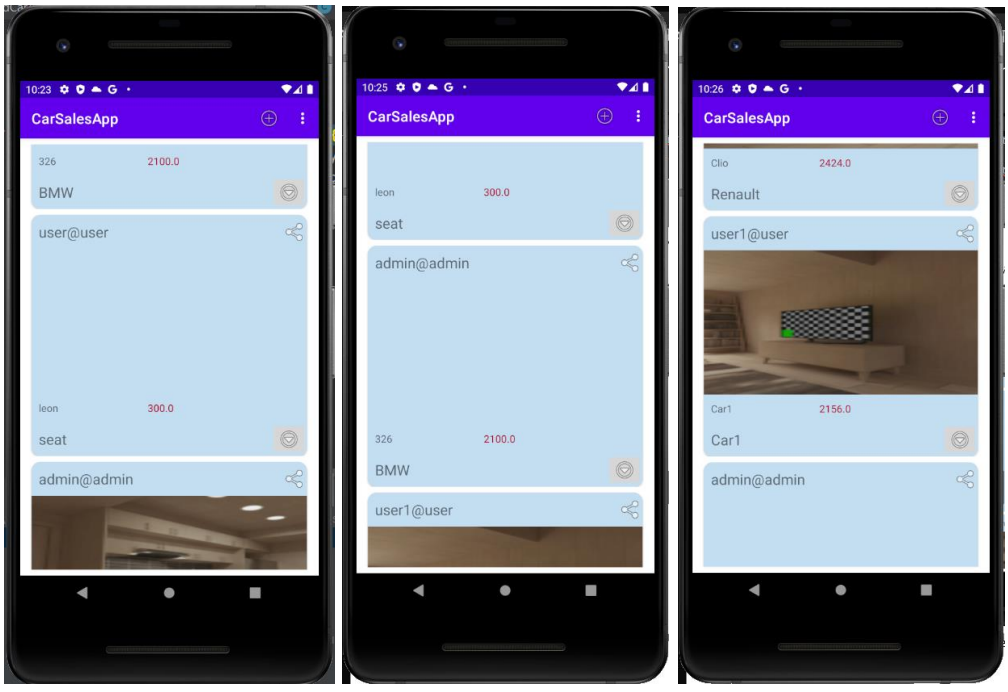
Java animation code.

```
addPost.setOnClickListener(view -> {

addPost.startAnimation((AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.close));
    Intent addCar = new Intent(MainActivity.this,AddCarActivity.class);
    startActivity(addCar);
});
```

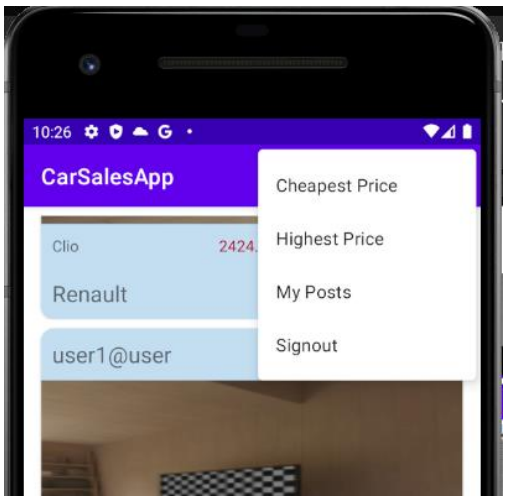
Defense 3

Task: Sort the RecyclerView by price.

We have created 2 menu items that let you sort by price, one from cheapest to highest and another one from highest to cheapest.



Default sorted **Sorted by cheapest** **Sorted by highest price**



Created compareTo method in the CarEntity class.

```
@Override
public int compareTo(Object car) {
    Double compareTo = ((CarEntity)car).getPrice();
    if(this.price - ((CarEntity) car).getPrice() > 0){
        return 1;
    }else if(this.price - ((CarEntity) car).getPrice() < 0){
        return -1;
    }
    return 0;
}
```

Sorting methods in RecyclerViewAdapter,

```
public void sortByPriceLH(){
    Collections.sort(carList);
}

public void sortByPriceHL(){
    Collections.sort(carList);
    Collections.reverse(carList);
}
```

In the main activity on click listener of the menu item.

```
case R.id.sortByPriceHL:
    recyclerViewAdapter.sortByPriceHL();
    carViewModel.getCars().observe(this, carEntities -> {
                recyclerViewAdapter = new
RecyclerViewAdapter(carEntities,MainActivity.this);
                recyclerView.setAdapter(recyclerViewAdapter);
            });
    return true;
case R.id.sortByPriceLH:
    recyclerViewAdapter.sortByPriceLH();
    carViewModel.getCars().observe(this, carEntities -> {
        recyclerViewAdapter = new
RecyclerViewAdapter(carEntities,MainActivity.this);
        recyclerView.setAdapter(recyclerViewAdapter);
    });
    return true;
```

Reference list