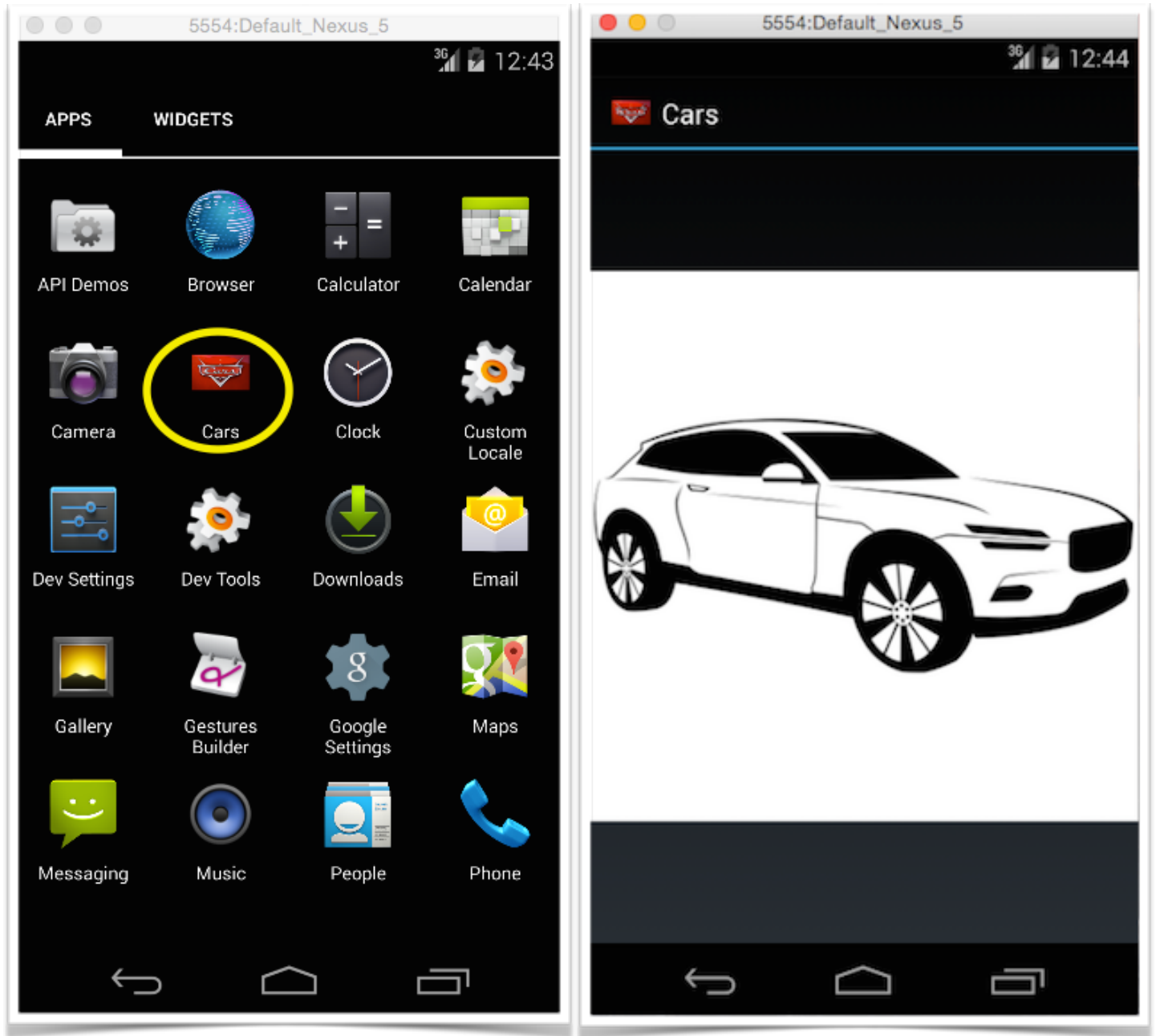


# CARS

-Abhinav Jayanthi, mim289



---

## ABSTRACT

Cars is a mobile application which runs on any smart phone with Android platform. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code. It uses XML for the User Interface. The Java and XML can be bound together to build the applications

Cars is an informative application which allows the user to select a car based on his choice of Make, Class and Price-Range. The information includes various key components like the Fuel consumption, dimension, power and so on which help the user to select the car of his choice. Cars is targeted for the latest version of the Android operating system Android 5.0 "Lollipop" (API level 21). It also has the capability to support the lower versions until Android 3.2 "Honeycomb"(API level 13).

---

## CONTENTS

1.0	INTRODUCTION.....	4
1.1	DESCRIPTION.....	4
1.1	MOTIVATION.....	4
2.0	APPLICATION.....	5
2.1	DESIGN AND IMPLEMENTATION.....	5
2.1.1	SPLASH SCREEN.....	5
2.1.2	MAIN ACTIVITY.....	6
2.1.2	SELECTION.....	7
2.1.3	VIEW ALL.....	8
2.2	FUNCTIONALITIES.....	8
2.2.1	SPINNERS.....	8
2.2.2	BUTTONS.....	9
2.2.3	DATA BASE.....	10
2.2.4	IMAGES.....	10
3.0	CONCLUSION.....	10
3.1	LIMITATIONS.....	11
3.2	FUTURE SCOPE.....	11
3.3	REFERENCES.....	11

---

# **1.INTRODUCTION**

## **1.1 DESCRIPTION**

The goal of this application is to provided the user a detailed information about the car he chooses to buy and the other variants available. The application must be able to provide a detailed content of each car he wishes to consider for a purchase.The user can navigate through the application to look for other cars of his interest irrespective of any other criteria. Since the application is in a premature state, the data based at present consist of eighteen cars for the user to browse about.

## **1.2 MOTIVATION**

The main motivation for this application is the lack of user friendly and descriptive applications for the user to help them find the car of his choice. Out in the market there are different application which can be non-informative or having excessive information which is either not useful or cannot be comprehended by the user. More particularly, the applications are not user-friendly.

Also, with a lot of different models of cars and different types of cars available, the user might be interested to keep himself updated regarding the cars in the market and know more about the specifications of each and every car.

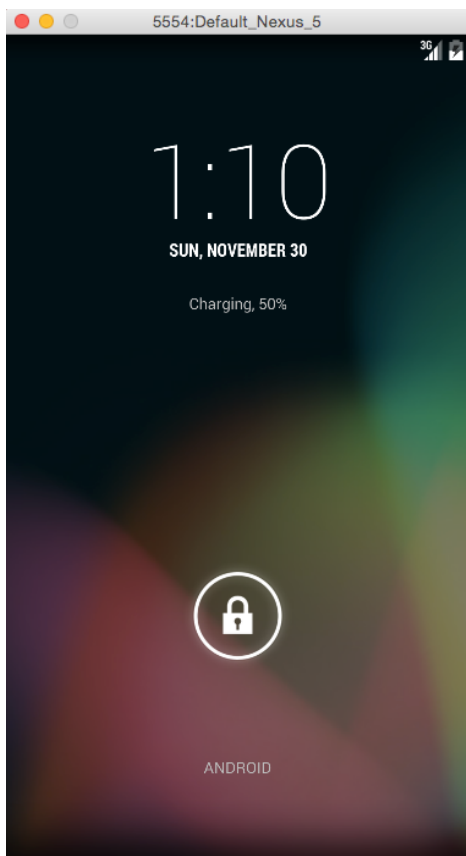
---

## 2.APPLICATION

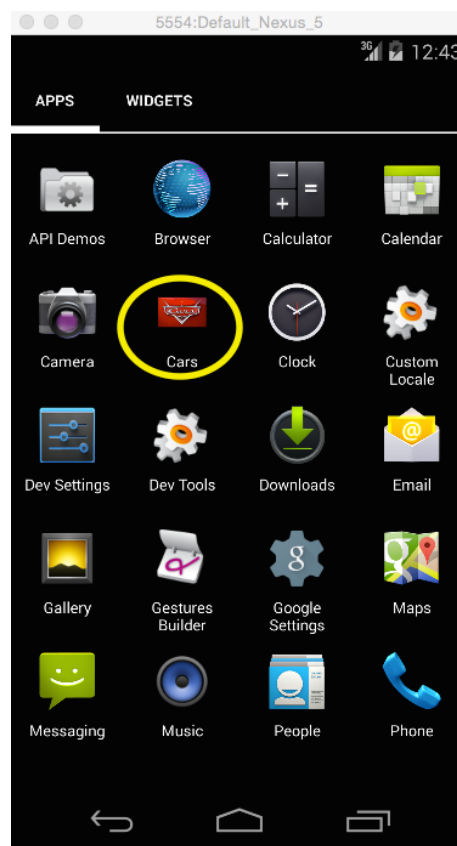
### 2.1 DESIGN AND IMPLEMENTATION

#### 2.1.1 SPLASH SCREEN

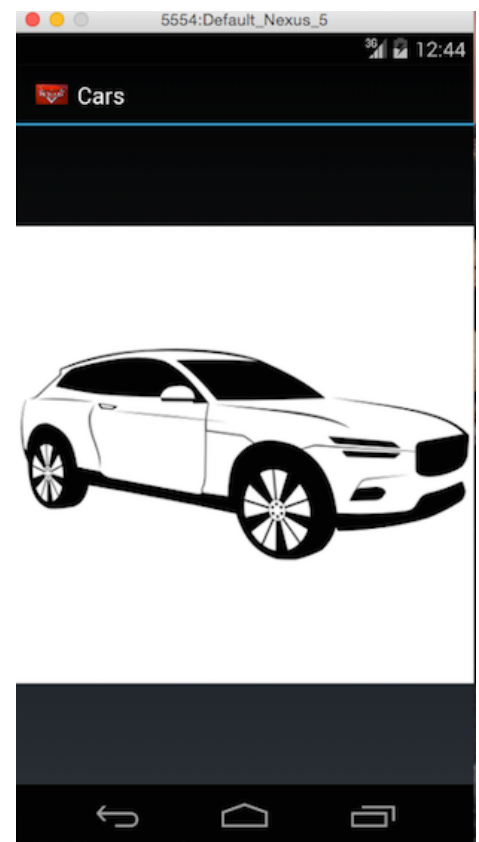
The Splash screen is the opening screen of the application, that is , it is the first screen which a user encounters once the application is launched from the home screen. The splash screen consist of an image which pops up every time the application is launched or the application is resumed. It has a timer set for two seconds from start of the application.



2.1 Lock Screen



2.2 App Screen

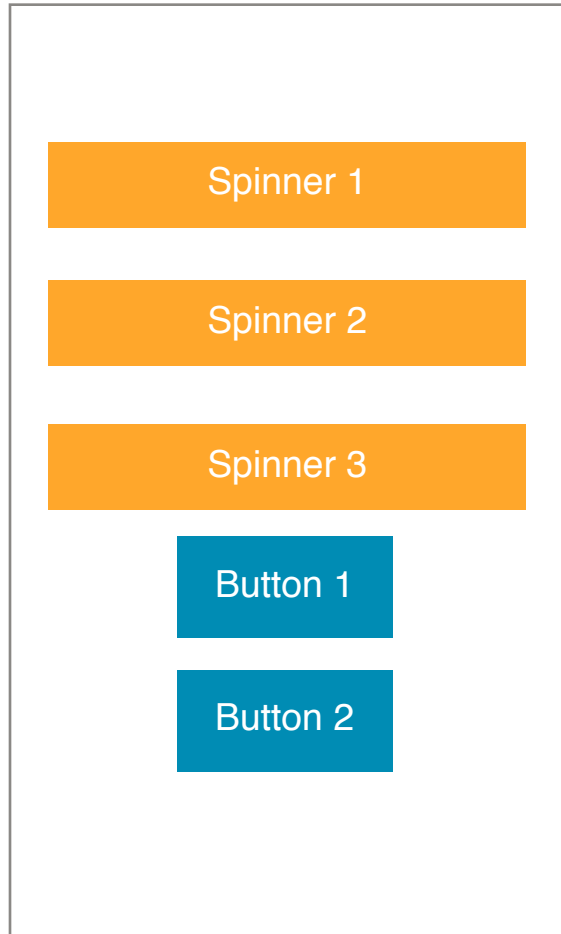


2.3 Splash Screen

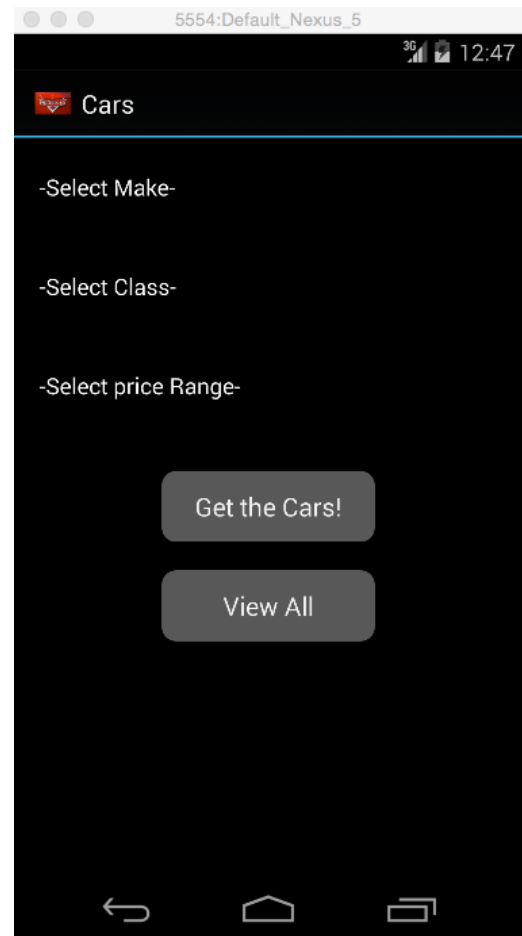
---

## 2.1.2 MAIN ACTIVITY

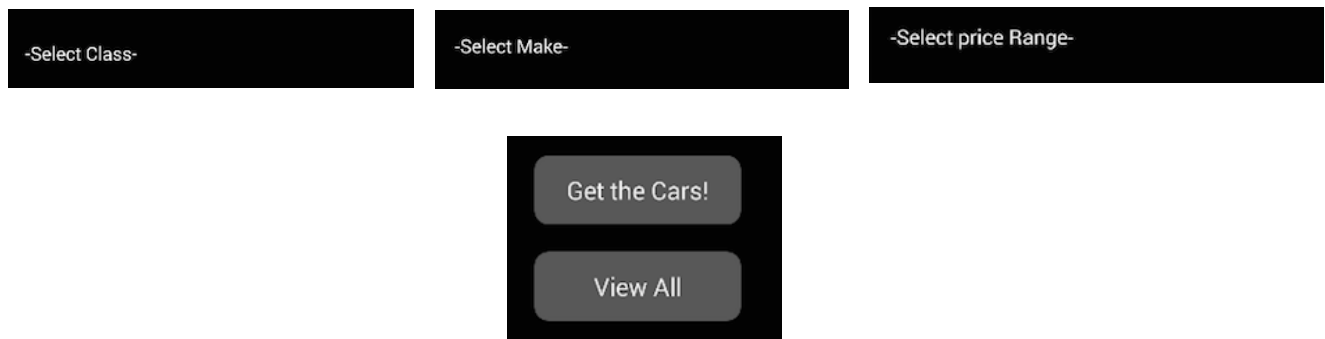
The Main activity is the first interact-able screen for the user. It consists of three Spinners ( Drop down ) and two Buttons.



2.4 Design

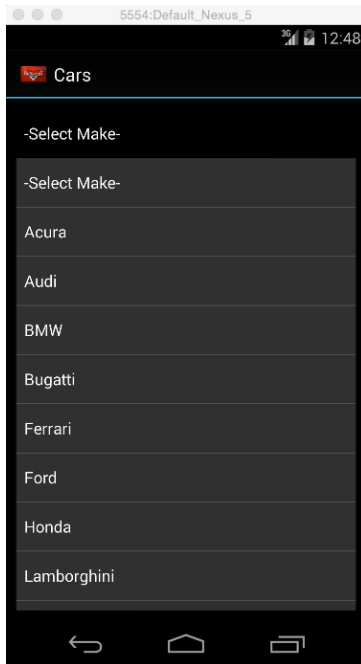


2.5 Main Activity

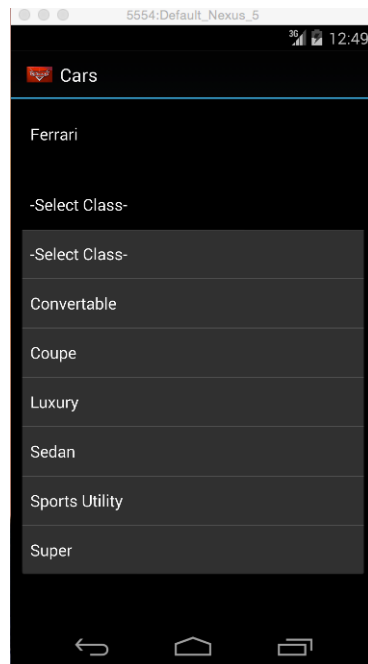


## 2.1.2 SELECTION

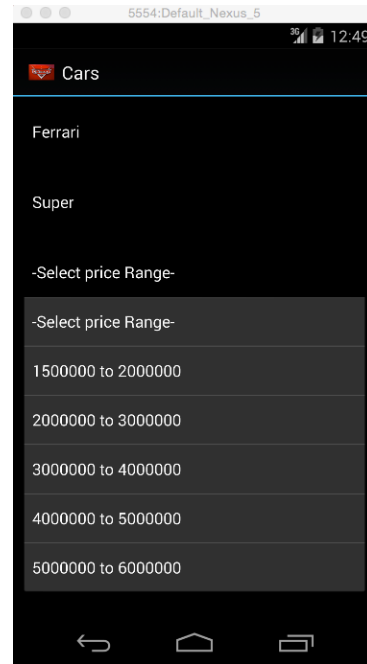
The user can select the Make, Class and the price Range and then click "Get the Cars!" .



2.6 Make Selection

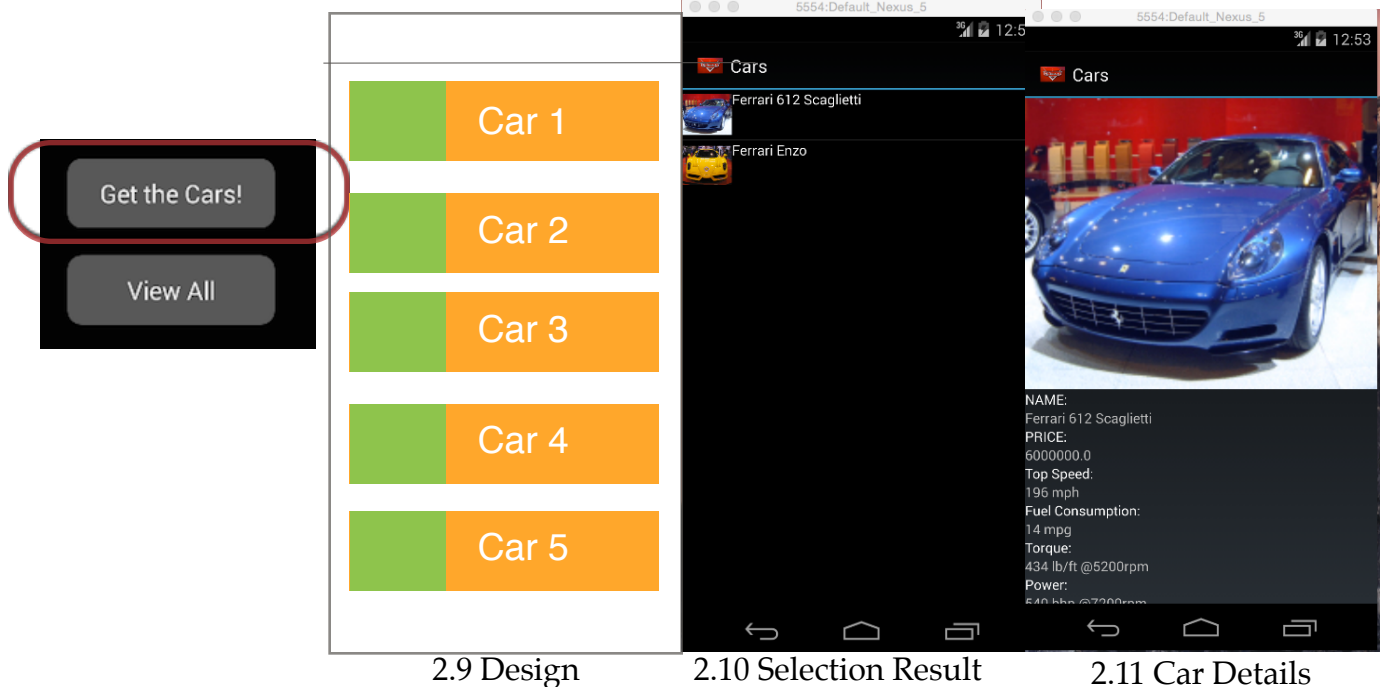


2.7 Class Selection



2.8 Price Selection

The result is a list of cars for the combination given by the user, and when clicked on a particular car, we get the details of the same



2.9 Design

2.10 Selection Result

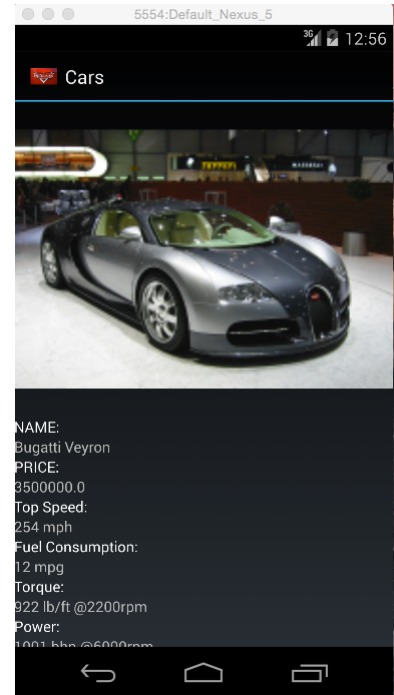
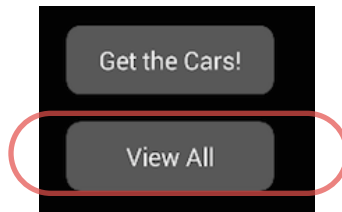
2.11 Car Details

## 2.1.3 VIEW ALL

In the Main Activity we have a button called "View ALL. By selecting this button we get the list of all the cars available in the SQLITE Data Base and selecting the car would result in giving the description of the car



2.12 View All



2.13 Car Details

## 2.2 FUNCTIONALITIES

### 2.2.1 SPINNERS

Spinners can be implemented in the layout by using the spinner object which has to be done in the XML file and can be populated in the String.xml file

#### 2.2.2 Populate Spinner

```
<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/make_spinner"
    android:gravity="center"
    android:layout_margin="10dp"
    android:background="@drawable/spinnerbackground"
/>
```

#### 2.2.1 To create Spinner

```
<string-array name="make_of_cars">
    <item>-Select Make-</item>
    <item>Acura</item>
    <item>Audi</item>
    <item>BMW</item>
    <item>Bugatti</item>
    <item>Ferrari</item>
    <item>Ford</item>
    <item>Honda</item>
    <item>Lamborghini</item>
    <item>Porsche</item>
</string-array>
```



---

and the response can be implemented by

```
...
mMakeSpinner.setOnItemSelectedListener(new OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position,
        long id) {
        if(position!=0){
            mSelectedMAKE=parent.getSelectedItem().toString();
        }
        else{
            Toast.makeText(getApplicationContext(), "Please Select a Make", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        // TODO Auto-generated method stub
    }
});
```

## 2.2.2 BUTTONS

The buttons can be implemented in the same way as the Spinners but the one thing that changes is in the action of an event ( Clicking of the button ) we use an interface called `onClickListener` which responds to the user's action

```
public OnClickListener mListener = new OnClickListener() {

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.get_button:
                Bundle bundle = new Bundle();
                bundle.putString("selectedMake", mSelectedMAKE);
                bundle.putString("selectedClass", mSelectedClass);
                bundle.putFloat("startRange", mStartRange);
                bundle.putFloat("endRange", mEndRange);

                Intent i= new Intent(getApplicationContext(), ActivityCarsList.class);
                i.putExtra("values", bundle);
                startActivity(i);

                break;
            case R.id.ViewAll:
                Intent newIntent = new Intent(getApplicationContext(),ActivityAllCarsList.class);
                startActivity(newIntent);
                break;
            default:
                break;
        }
    }
};
```

```
<Button
    android:layout_marginTop="25dp"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:text="Get the Cars!"
    android:layout_gravity="center"
    android:id="@+id/get_button"
    android:background="@drawable/buttons_background"
    android:padding="2dp"/>
```

### 2.2.3 To create Button

## 2.2.3 DATA BASE

The data base used here is SQLITE, we have a `MySqliteHepler.class` which implement all the data base functionalities

```
public class MySqliteHelper extends SQLiteOpenHelper
{
    private static final String DATABASE_NAME = "cars.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "myCarsList";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_MAKE = "make";
    private static final String COLUMN_PRICE = "price";
    private static final String COLUMN_CLASS = "class";
    private static final String COLUMN_IMAGE = "imageUrl";
    private static final String COLUMN_TOP_SPEED = "topSpeed";
    private static final String COLUMN_FUEL_CONS = "fuelCons";
    private static final String COLUMN_TORQUE = "torque";
    private static final String COLUMN_POWER = "power";
    private static final String COLUMN_LENGTH = "length";
    private static final String COLUMN_WIDTH = "width";
    private static final String COLUMN_HEIGHT = "height";
    private static final String COLUMN_WEIGHT = "weight";
    private static final String TABLE_CREATE = "create table " + TABLE_NAME
        + "(" + COLUMN_NAME + " VARCHAR(80) " + COLUMN_MAKE
        + " VARCHAR(20) " + COLUMN_CLASS + " VARCHAR(20) " + COLUMN_PRICE
        + " INT(10) " + COLUMN_TOP_SPEED + " VARCHAR(20) " + COLUMN_FUEL_CONS + " VARCHAR(20) "
        + COLUMN_POWER + " VARCHAR(20) " + COLUMN_TORQUE + " VARCHAR(20) " + COLUMN_LENGTH + " VARCHA
        + COLUMN_WIDTH + " VARCHAR(20) " + COLUMN_HEIGHT + " VARCHAR(20) " + COLUMN_WEIGHT + " VARCHA
        + COLUMN_IMAGE + " VARCHAR(500));";

    public void addCar(Car car)
    {
        SQLiteDatabase database = getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COLUMN_NAME, car.getName());
        contentValues.put(COLUMN_MAKE, car.getMake());
        contentValues.put(COLUMN_CLASS, car.getClassOfCar());
        contentValues.put(COLUMN_PRICE, car.getPrice());
        contentValues.put(COLUMN_FUEL_CONS, car.getfuelCons());
        contentValues.put(COLUMN_TOP_SPEED, car.gettopSpeed());
        contentValues.put(COLUMN_TORQUE, car.gettorque());
        contentValues.put(COLUMN_POWER, car.getpower());
        contentValues.put(COLUMN_LENGTH, car.getlength());
        contentValues.put(COLUMN_WIDTH, car.getwidth());
        contentValues.put(COLUMN_HEIGHT, car.getheight());
        contentValues.put(COLUMN_WEIGHT, car.getweight());
        contentValues.put(COLUMN_IMAGE, car.getUrl());
        database.insert(TABLE_NAME, null, contentValues);

        database.close();
    }

    public List<Car> getCarsBasedonMake(String make)
    {
        List<Car> filteredCars = new ArrayList<Car>();
        String getColumnQuery = "SELECT * " + " FROM " + TABLE_NAME
            + " WHERE " + COLUMN_MAKE + " = " + make;
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(getColumnQuery, null);
        if (cursor.moveToFirst()) {
            do {
                Car filteredCar = new Car(cursor.getString(0),
                    cursor.getString(1), cursor.getString(2),
                    cursor.getFloat(3), cursor.getString(4), cursor.getString(5), cursor.getString(6)
                    , cursor.getString(7), cursor.getString(8), cursor.getString(9), cursor.getString(10), cursor.getString(11), cur
                filteredCars.add(filteredCar);

            } while (cursor.moveToNext());
        }
        cursor.close();
        return filteredCars;
    }
}
```

## 2.2.4 Data base implementation

## 2.2.4 IMAGES

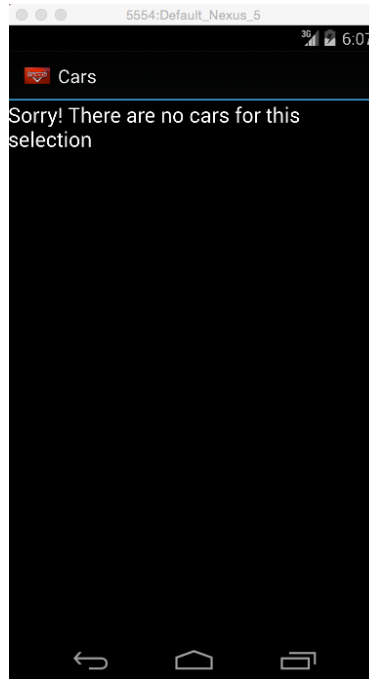
The images are borrowed from <http://www.supercarworld.com/cgi-bin/listall.cgi> and are hardcoded into the database to save memory rather than storing images into the data base

---

## 3.CONCLUSION

### 3.1 LIMITATIONS

The application has a very small data base, so for some combinations of inputs, the application will show an message saying "Sorry! There are no cars for this selection"



3.1 Limitation

### 3.2 FUTURE SCOPE

- Feedback from the user in the form of ratings of the car
- GPS navigations to the nearest showroom and the contact information of the showroom
- Increase the number of images by using a slide show

### 3.3 REFERENCES

- <http://developer.android.com/design/patterns/help.html>
- <http://www.supercarworld.com/cgi-bin/listall.cgi>
- <http://slidenerd.com/2014/07/18/1-android-activity-lifecycle/>