MAD Project Evaluation 2 – Regular Intake 2020													
Project Name:	**Mobile application for phone shop**		**Batch: Weekday**						**Group ID: MAD_2020_G5_23**				
Student ID	Student Name	Integrated system using a repository (Out of 5)	Marketing the app using a video clip published in a social media account (eg: LinkedIn) (Out of 10)	Function completeness (Out of 25)	Database connectivity (Out of 5)	Usability (Out of 5)	Validations (Out of 5)	Good coding practice (Out of 5)	Out of box features (Out of 5)	Test cases (Out of 10)	Report (Out of 10)	Q & A (Out of 15)	Total (Out of 100)
IT19057248	G.L.I.R. Liyanage												
IT19094328	Vidarshana K.L.G.T												
IT19114422	Dissanayaka G.E.M												
IT19171302	T.V. Thimira Isiwara Vithanage												

Git Repository Link: https://github.com/ishinir/madproject
https://github.com/GayathriVidarshana/MAD_Project.git
https://github.com/Thimira98/Thimira_Delivery

Link for the video in social media: https://m.facebook.com/story.php?story_fbid=2757190294553300&id=100007870684719&sfnsn=mo
(All group members must be tagged)

----------For Evaluators----------

Evaluator's name:

Comments:

Project Introduction

Mobile application for phone shop.

Electro is a mobile application which we are going to introduce to the mobile shops. As the current situation in the world and also everyone is almost getting used to this online buying and selling system we have decided to make an online mobile device selling shop so that people can use the app to order mobile accessories to their door step. And also the shop owners can make their customers satisfied with their service.
This application can be used by every person who has an android device. When designing this application we decided to have 4 main components. Those are,

- Admin management
- User management
- Product and Order management
- Payment management

Admin management
In admin management component employee login can be done. And, admin can add/ delete/ update all the items, employees and item information to the application, search for the item with item code, and search for the user with user name can be done through this.

User registration, Login and Deliver
User management is to manage user login and manage user registration. User can edit their information and delete their details. Inside the deliver management system Admin can assign a driver and system send a message to the driver about the deliver.

Product and Order management
Order management component includes managing the customer view of order list, and separate information on each item, managing customer cart information, calculating the total price in cart items.

Payment management
Payment management includes calculating the total amount including delivery charges, getting customer delivery information. And at last all the payment details and order details will be shown here.

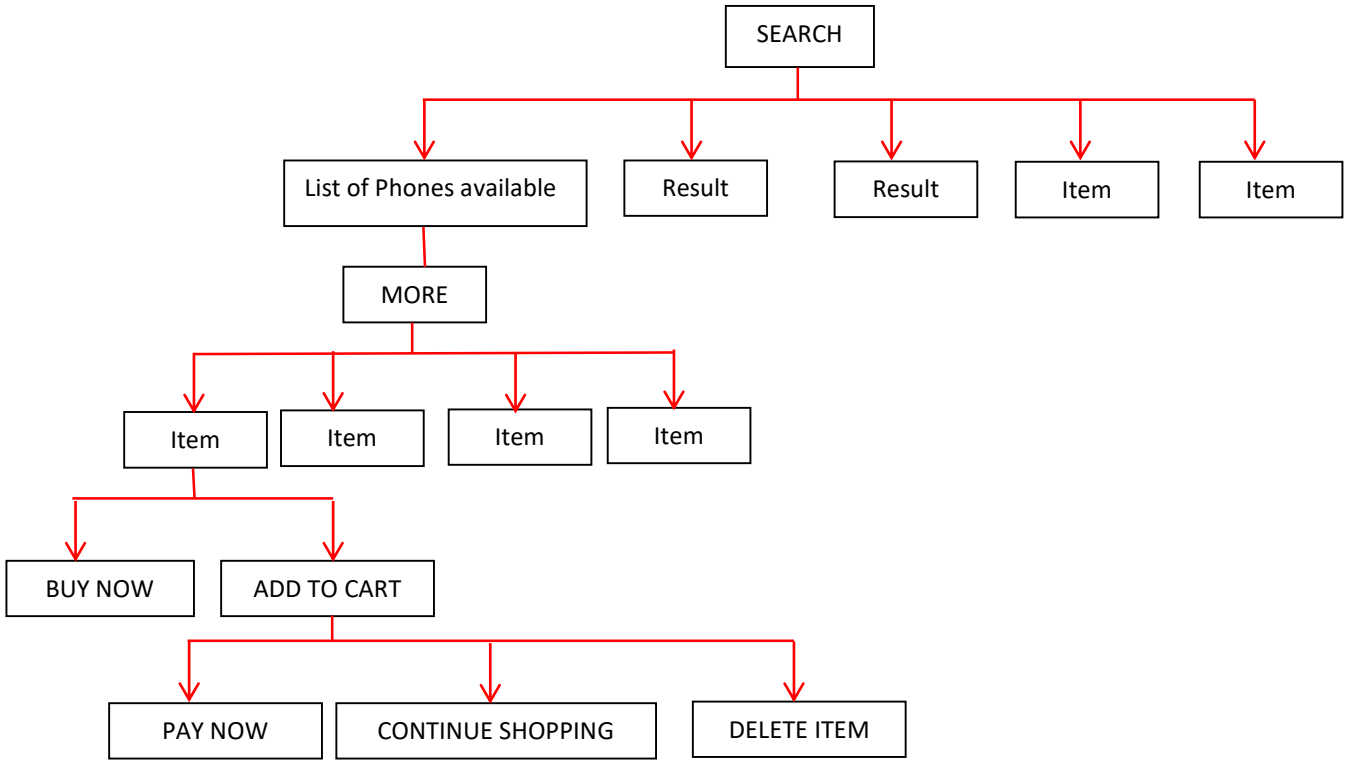
We have designed this application clearer and simple for customer to use. Customers should get registered to make order. But anyone can view item available in the shop by installing our application. We have a cart which customer can store items which they wish to buy later this will be an advantage for the customer so that they can store the items without searching them again and again. Customers and the shop owners will be benefited by using this application.

IT19057248 – Product and Order Management Component

Function Description

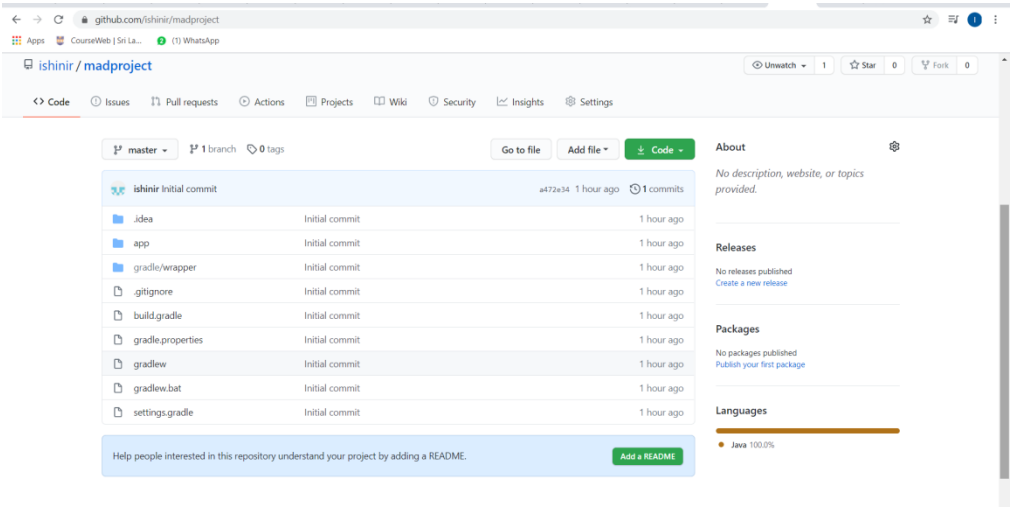
- In my component there are some important buttons,
  - → Search (customer can search for the items they are looking for.)
  - → More (when customer search for an item and wants to buy it they have to look for more details to get a clear idea about that item.)
  - → Add to cart (there will be item list which customer wish to by, that will be created in the cart.)
  - → Continue shopping (while customer is in the cart they can add more items to the cart by clicking "continue shopping button.)
  - → Delete sign (by clicking this image button customer can delete the items they order while they are in the shopping cart.)
  - → Login (this will be an image button which allows customer to log in to the system.)
  - → Cart (an image button which shows customer their cart. The list of items they wish to buy)

- When a customer open our app it will be showing the home page which has the images of the shop, image of the available brands in the shop, a slight description about the shop, Logo of the shop and there will be two image buttons which allows customer to log in to the system and the other one will be showing the customer cart.
- Customer need to sear for the items they are looking for in the search bar. By clicking the "SEARCH" button system will be showing the item list which customer searched. In Item list page customer can search for more details which they wish to buy by clicking the "MORE" buttons. From there customer will be moved to a new page which shows a brief description about the product. In that page customer has two options. Customer can directly by the product by clicking "BUY NOW" button or else customer can add the items to the cart so they can buy it later when they wants by clicking on "ADD TO CART" button.
- Whenever customer wants to go to the cart they can click on the cart image button, but they have to log in to the system before going to the cart. Customer can add items to the cart and there will be a maximum number of products that can be added to the cart. Inside the cart customer can delete the items which they do not need. If there is some more items to buy customer can click on "CONTINUE SHOPPING" button and go back to the home page. Inside this cart we will be showing the customer item name, item price and the total amount of the items in the cart.

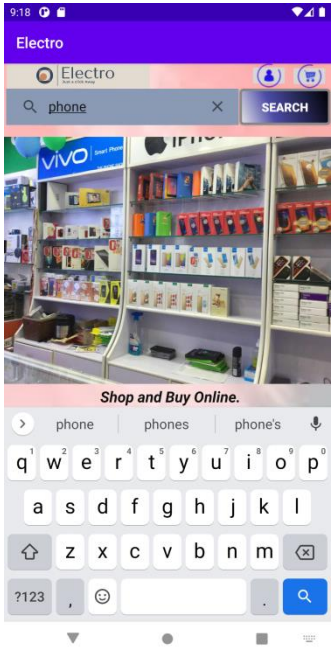


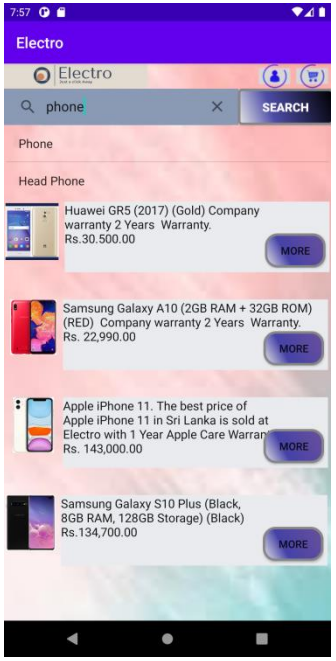
GitHub individual Contribution

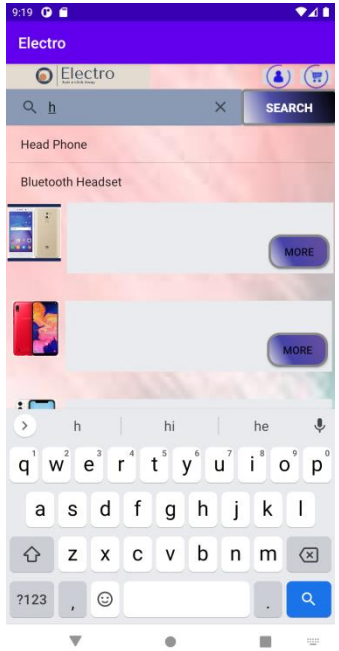
https://github.com/ishinir/madproject

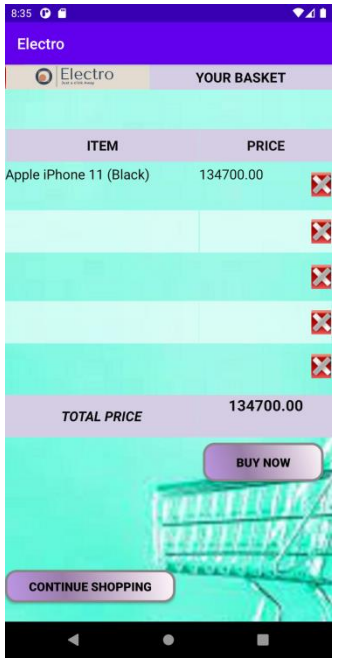


Snapshots of running Application









Calculations and Operations

```java
item3 = (TextView) findViewById(R.id.item3);
item4 = (TextView) findViewById(R.id.item4);
item5 = (TextView) findViewById(R.id.item5);

price1 = (TextView) findViewById(R.id.price1);
price2 = (TextView) findViewById(R.id.price2);
price3 = (TextView) findViewById(R.id.price3);
price4 = (TextView) findViewById(R.id.price4);
price5 = (TextView) findViewById(R.id.price5);

insertCart = new cart_DB();
datab = FirebaseDatabase.getInstance().getReference().child("insertCart");

pay = (Button) findViewById(R.id.pay);
pay.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        double Price1 = Double.parseDouble(price1.getText().toString().trim());
        double Price2 = Double.parseDouble(price2.getText().toString().trim());
        double Price3 = Double.parseDouble(price3.getText().toString().trim());
        double Price4 = Double.parseDouble(price4.getText().toString().trim());
        double Price5 = Double.parseDouble(price5.getText().toString().trim());

        insertCart.setItem1(item1.getText().toString().trim());
        insertCart.setPrice1(Price1);
        insertCart.setItem2(item2.getText().toString().trim());
        insertCart.setPrice2(Price2);
        insertCart.setItem3(item3.getText().toString().trim());
        insertCart.setPrice3(Price3);
        insertCart.setItem4(item4.getText().toString().trim());
        insertCart.setPrice4(Price4);
        insertCart.setItem5(item5.getText().toString().trim());
        insertCart.setPrice5(Price5);
```

```java
Button searchbtn,more1, more2, more3, more4, more5;

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

    searchItem = (SearchView) findViewById(R.id.searchItem);
    listItem = (ListView) findViewById(R.id.listItem);

    list=new ArrayList<String>();

    list.add("Phone");
    list.add("Head Phone");
    list.add("Chargers");
    list.add("Phone Covers");
    list.add("Screen Protectors");
    list.add("Data Cables");
    list.add("Bluetooth Headset");
    list.add("Memory Cards");

    adapter = new ArrayAdapter<>( context: this,android.R.layout.simple_list_item_1,list);
    listItem.setAdapter(adapter);

    searchItem.setOnQueryTextListener(new SearchView.OnQueryTextListener()
    {
        @Override
        public boolean onQueryTextSubmit(String s) { return false; }

        @Override
        public boolean onQueryTextChange(String s)
        {
            adapter.getFilter().filter(s);
```

```java
package com.example.madproject.DATABASE;

public class items_DB
{
    public items_DB()
    {
    }

    private String descrip1;
    private String descrip2;
    private String descrip3;
    private String descrip4;

    private byte item1;
    private byte item2;
    private byte item3;
    private byte item4;

    public String getDescrip1() {
        return descrip1;
    }

    public void setDescrip1(String descrip1) {
        this.descrip1 = descrip1;
    }

    public String getDescrip2() {
        return descrip2;
    }

    public void setDescrip2(String descrip2) {
        this.descrip2 = descrip2;
    }
```

```java
package com.example.madproject;

import ...

public class MainActivity extends AppCompatActivity {

    private Button searchbtn;
    ImageButton  cart1;
    ImageButton login, cart;

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


        searchbtn = (Button) findViewById(R.id.searchbtn);
        searchbtn.setOnClickListener((V) -> { openActivity_items(); });

        cart1 = (ImageButton) findViewById(R.id.cart1);
        cart1.setOnClickListener((v) -> { openActivity_cart(); });

        login = (ImageButton) findViewById(R.id.login);
        login.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                openActivity_login();
            }
        });
```

```java
}

private String Item1;
private String Item2;
private String Item3;
private String Item4;
private String Item5;

private double price1;
private double price2;
private double price3;
private double price4;
private double price5;


public String getItem1() { return Item1; }

public void setItem1(String item1) { Item1 = item1; }

public String getItem2() { return Item2; }

public void setItem2(String item2) { Item2 = item2; }

public String getItem3() { return Item3; }

public void setItem3(String item3) { Item3 = item3; }

public String getItem4() { return Item4; }

public void setItem4(String item4) { Item4 = item4; }

public String getItem5() { return Item5; }
```

IT19094328 – Admin management system

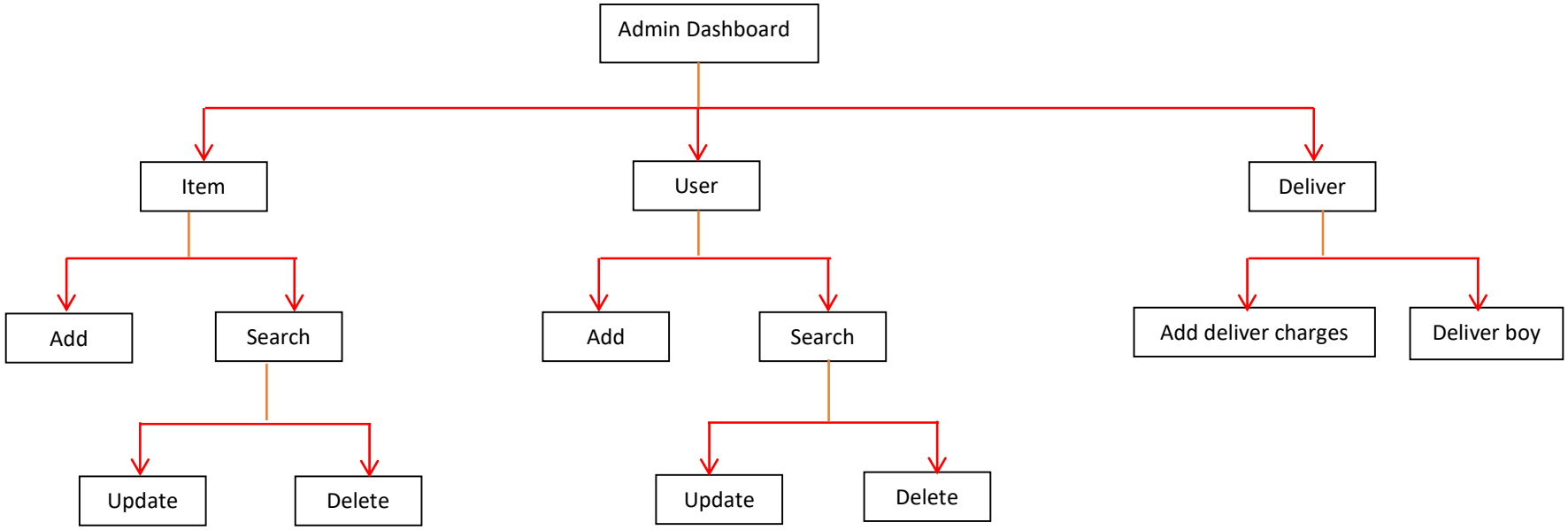
Function Description

In my component, the admin is the main user and the dashboard is the main interface. On the dashboard, it displays three buttons. There are the Item button, User Button, and Deliver Button. In my function, the admin is the only one who can manage the item details, manage the user of the admin panels, and manage the delivery charges and the drivers.

When clicking the Item button on the dashboard, then it displays the Add item button and the search bar to search the item. When clicking the Add item button the admin can add new items to the system. When adding a new item the admin should insert item code, item model, item name, price, discount, description, and images of that item. The admin wants to search item details or update item details or delete some items then enter the item code to search bar and search. Then it displays details of that item. That interface has two buttons to delete or update.

When clicking the user button on the dashboard, then it displays a button for add user and search bar for search user. When clicking the Add user button then can add a new user to the system. The users are shop owners and members of the administration panel. And the admin can search the user by entering the user name then the admin can update or delete the users.

When click the deliver button on the dashboard, then it displays two buttons. The delivery charges for manage the delivery charges. And deliver boy button to add the drivers for the system.



GitHub individual Contribution graph

https://github.com/GayathriVidarshana/MAD_Project.git

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 0 tags | Go to file | Add file | Code

GayathriVidarshana MAD project/Admin 0b21257 2 hours ago 1 commits

.idea	MAD project/Admin	2 hours ago
app	MAD project/Admin	2 hours ago
gradle/wrapper	MAD project/Admin	2 hours ago
.gitignore	MAD project/Admin	2 hours ago
build.gradle	MAD project/Admin	2 hours ago
gradle.properties	MAD project/Admin	2 hours ago
gradlew	MAD project/Admin	2 hours ago
gradlew.bat	MAD project/Admin	2 hours ago
settings.gradle	MAD project/Admin	2 hours ago

About

No description, website, or topics provided.

Releases

No releases published

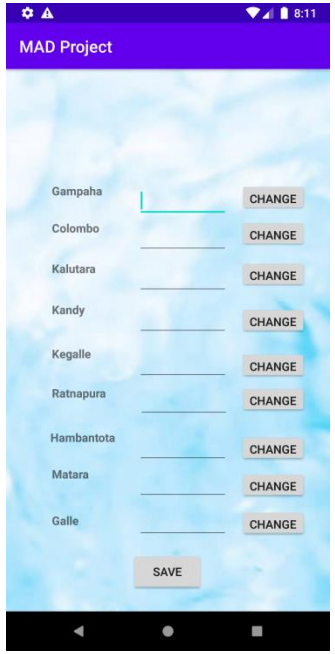
Packages

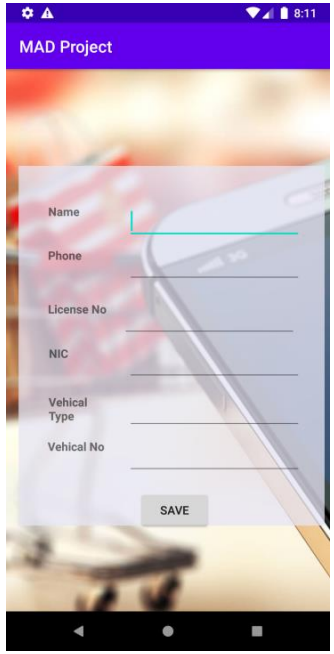
No packages published

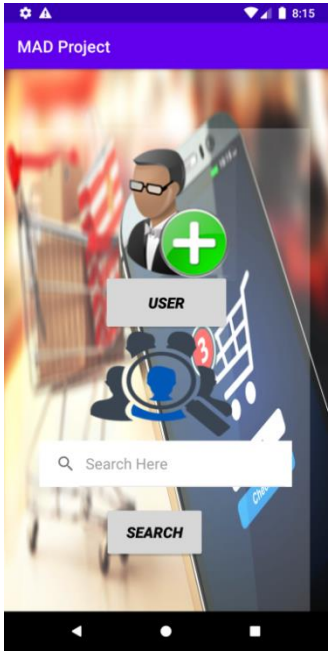
Languages

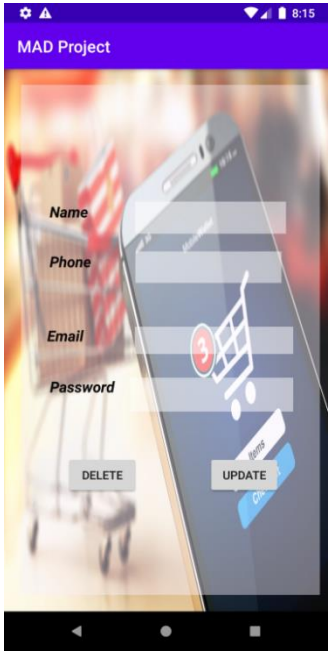
Java 100.0%

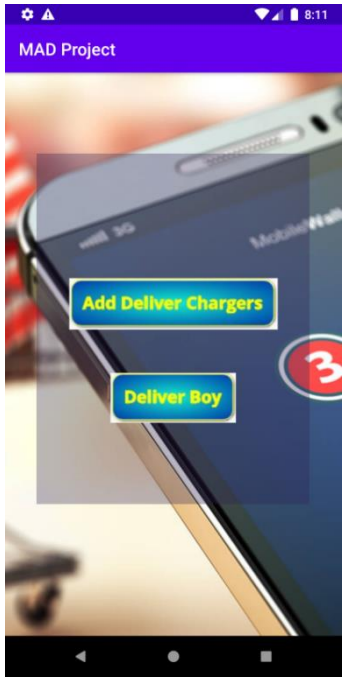
Snapshots of running Application

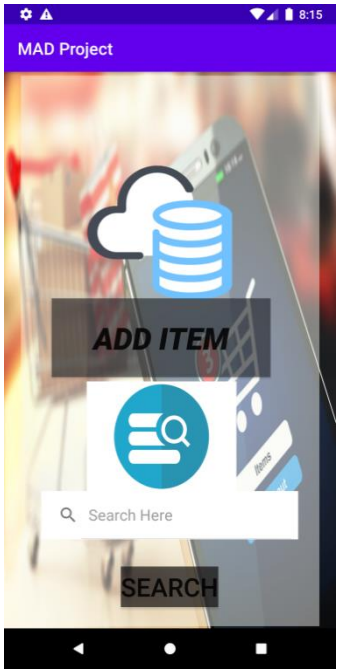












Calculations and Operations

```java
//choose image fro gallery
@Override
protected void onActivityResult ( int requestCode, int resultCode, @Nullable Intent data){
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 1 && requestCode == RESULT_OK && data != null && data.getData() != null) {
        imageView = findViewById(R.id.imageView);
        final List<Bitmap> bitmaps = new ArrayList<>();
        ClipData clipData = data.getClipData();

        if (clipData != null) {
            for (int i = 0; i < clipData.getItemCount(); i++) {

                imageUri = clipData.getItemAt(i).getUri();
                try {
                    InputStream is = getContentResolver().openInputStream(imageUri);
                    Bitmap bitmap = BitmapFactory.decodeStream(is);
                    bitmaps.add(bitmap);
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                }
            }
        } else {

            imageUri = data.getData();
            try {
                InputStream is = getContentResolver().openInputStream(imageUri);
                Bitmap bitmap = BitmapFactory.decodeStream(is);
                bitmaps.add(bitmap);
```

```java
private void ClearFields(){

    Inputcode.setText("");
    InputName.setText("");
    InputModel.setText("");
    InputDiscount.setText("");
    InputPrice.setText("");
    InputDescription.setText("");
    InputStock.setText("");

}

}
```

```java
//declare database reference
DatabaseReference reff;
//declare storage Reference
StorageReference myStorage;
Add_Item_DB Add_Item;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_item);

    //CODE FOR INPUT IMAGES FROM GALLERY
    TextView ImgButtonAdd = findViewById(R.id.btnimage);
    ImgButtonAdd.setText("Add Image");
    ImgButtonAdd.setOnClickListener((v) -> {
            //Filechooser();
            if (ActivityCompat.checkSelfPermission( context: AddItem.this,
                    Manifest.permission.READ_EXTERNAL_STORAGE)
                    != PackageManager.PERMISSION_GRANTED) {

                ActivityCompat.requestPermissions( activity: AddItem.this,
                        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},  requestCode: 1000);

            }

            Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
            intent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE,  value: true);
            intent.setType("image/*");
            startActivityForResult(intent,  requestCode: 1);

    });
```

```java
Add_User = new Add_User_DB();

//location to save
reff = FirebaseDatabase.getInstance().getReference().child("User_Details");

btnsave.setOnClickListener((v) -> {
        int phone_u = Integer.parseInt(InputPhone.getText().toString().trim());

        Add_User.setUName(inputusrename.getText().toString().trim());
        Add_User.setUphone(phone_u);
        Add_User.setUEmail(InputEmail.getText().toString().trim());
        Add_User.setUPw(inputPswrd.getText().toString().trim());


        //set Primary key
        reff.child(inputusrename.getText().toString().trim()).setValue(Add_User);

        reff.push().setValue(Add_User);
        Toast.makeText( context: AddNewUser.this,  text: "Done", Toast.LENGTH_LONG).show();

});
```

```java
//firebase table
Add_Item = new Add_Item_DB();
//location to save Item data in firebase
reff = FirebaseDatabase.getInstance().getReference().child("Item_Details");
//location to save images of Item in firebase
myStorage = FirebaseStorage.getInstance().getReference( location: "Item_Image");

SaveNewItemBtn.setOnClickListener((v) -> {

        double InputPriceC = Double.parseDouble(InputPrice.getText().toString().trim());
        double InputDiscountI = Double.parseDouble(InputDiscount.getText().toString().trim());
        int StockI = Integer.parseInt(InputStock.getText().toString().trim());

        Add_Item.setIcode(Inputcode.getText().toString().trim());
        Add_Item.setBrName0(InputName.getText().toString().trim());
        Add_Item.setImodel(InputModel.getText().toString().trim());
        Add_Item.setIprice(InputPriceC);
        Add_Item.setIDiscount(InputDiscountI);
        Add_Item.setIdescription(InputDescription.getText().toString().trim());
        Add_Item.setStock(StockI);

        //set primary key(Icode)
        reff.child(Inputcode.getText().toString().trim()).setValue(Add_Item);

        reff.push().setValue(Add_Item);
        Toast.makeText( context: AddItem.this,  text: "Successfully saved", Toast.LENGTH_LONG).show();
        ClearFields();

});
```

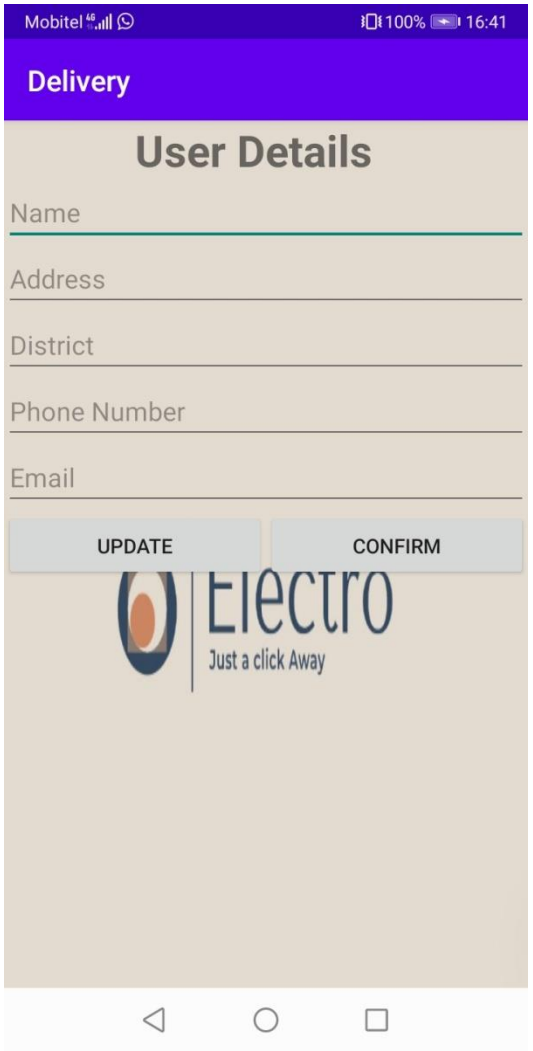
```java
//save image as a url in firebase
private String getExtention(Uri uri) {

    ContentResolver cr = getContentResolver();
    MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton();
    return mimeTypeMap.getExtensionFromMimeType(cr.getType(uri));

}
private void FileUpload() {

    StorageReference Ref = myStorage.child(System.currentTimeMillis() + "." + getExtention(imageUri));

    Ref.putFile(imageUri)
            .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

                    // Get a URL to the uploaded content
                    // Uri downloadUri = taskSnapshot.getUploadSessionUri();
                    //Uri downloadUri = taskSnapshot.getDownloadUrl();
                    Toast.makeText( context: AddItem.this,  text: "Image Uploaded successfully", Toast.LENGTH_LONG).show();

                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception exception) {
                    // Handle unsuccessful uploads
                    // ...
                }
            });
}
```

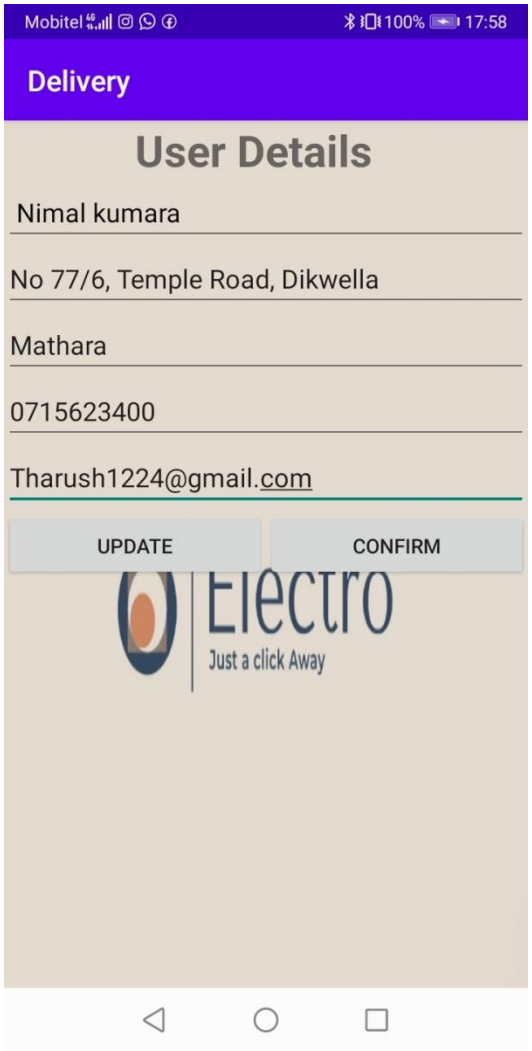
IT19171302 - Payment management

GitHub: - https://github.com/Thimira98/Thimira_Delivery

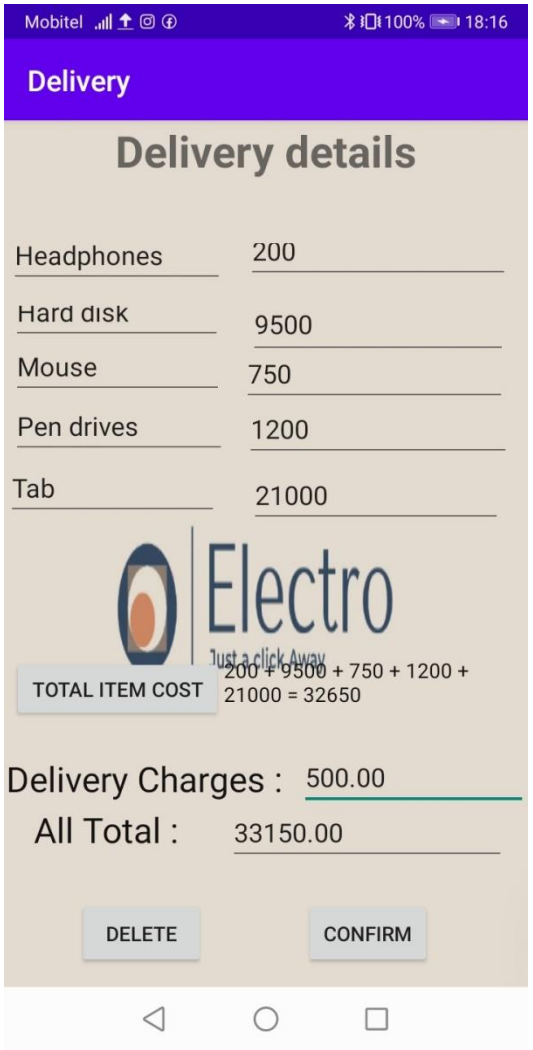
1. A customer can see the details of the order he bought from Figure 1 above in the mobile shop. Click on the "Total Item Cost" button to see the full price of some of the items selected above. Customers can update some of the required items by clicking the "Update Button" above to fix an item.

2. The customer then clicks the "Confirm Button" above and the user details are displayed. This username should include his name, address, district, telephone number, and email. If you have incorrectly entered the relevant details, you can correct it with the "Update button" above.





3. Then click on the second "Confirm button" and the distribution details will appear. Customer pre-selected order details are displayed directly in the delivery details. For remote customers he will be charged additional district delivery fees and the sum of the two district distribution fees for selected items will be shown. In the "Delete button" above, the customer can remove as many selected items as he wants from the above order details at any time. Then by clicking on the "Confirm button" a short message "Thank you" will appear as the form is completed.



Calculations and Operations

```
butUpdate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        dbRef = FirebaseDatabase.getInstance().getReference();
        dbRef.child("Order").child("Ord1").child("item1").setValue(etItem1.getText().toString().trim());
        dbRef.child("Order/Ord1/item2").setValue(etItem2.getText().toString().trim());
        dbRef.child("Order/Ord1/item3").setValue(etItem3.getText().toString().trim());
        dbRef.child("Order/Ord1/item4").setValue(etItem4.getText().toString().trim());
        dbRef.child("Order/Ord1/item5").setValue(etItem5.getText().toString().trim());

        dbRef.child("Order/Ord1/item1Price1").setValue(etItem2.getText().toString().trim());
        dbRef.child("Order/Ord1/item2Price2").setValue(etItem3.getText().toString().trim());
        dbRef.child("Order/Ord1/item3Price3").setValue(etItem4.getText().toString().trim());
        dbRef.child("Order/Ord1/item4Price4").setValue(etItem5.getText().toString().trim());
        dbRef.child("Order/Ord1/item5Price5").setValue(etItem5.getText().toString().trim());

        Toast.makeText(getApplicationContext(), text: "Sucessfully updated", Toast.LENGTH_SHORT).show();
        clearControls();
    }
});
```

Update

```
butUpdate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        dbRef = FirebaseDatabase.getInstance().getReference();
        dbRef.child("Order").child("Ord1").child("name").setValue(etName.getText().toString().trim());
        dbRef.child("Order/Ord1/address").setValue(etAddress.getText().toString().trim());
        Toast.makeText(getApplicationContext(), text: "Sucessfully updated", Toast.LENGTH_SHORT).show();
        clearControls();
    }
});
```

Save (Confirm)

```
butSaveCon.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        int userPhone = Integer.parseInt(etPhone.getText().toString().trim());

        order.setName(etName.getText().toString().trim());
        order.setAddress(etAddress.getText().toString().trim());
        order.setDisrict(etDistrict.getText().toString().trim());
        order.setPhoneNumber(userPhone);
        order.setEmail(etEmailAddress.getText().toString().trim());

        dbRef.push().setValue(order);
        Toast.makeText( context: activity_main_3.this,  text: "Sucessfully Inserted", Toast.LENGTH_LONG).show();
    }
});
```

```
butDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        dbRef = FirebaseDatabase.getInstance().getReference().child("Order").child("Ord1");
        dbRef.child("Order").child("Ord1").child("item1").setValue(etItem1.getText().toString().trim());
        dbRef.child("Order/Ord1/item2").setValue(etItem2.getText().toString().trim());
        dbRef.child("Order/Ord1/item3").setValue(etItem3.getText().toString().trim());
        dbRef.child("Order/Ord1/item4").setValue(etItem4.getText().toString().trim());
        dbRef.child("Order/Ord1/item5").setValue(etItem5.getText().toString().trim());
        dbRef.removeValue();
        Toast.makeText(getApplicationContext(),  text: "Successfully deleted", Toast.LENGTH_SHORT).show();
    }
});
```

IT19114422 - User registration, Login and Deliver

Function Description

Inside my component I have three main parts.

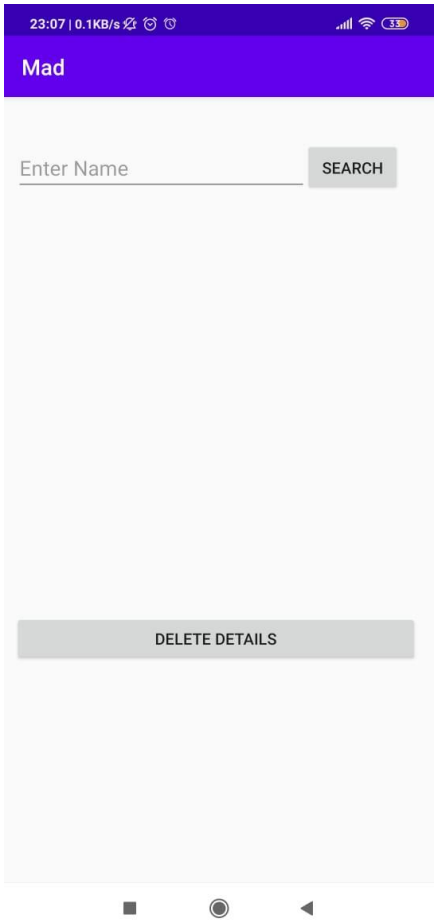
- → User registration
- → User login
- → Deliver

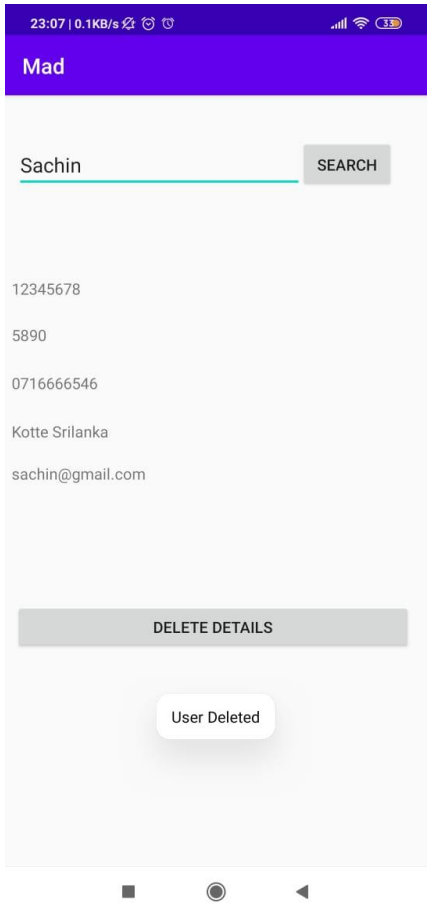
Admin dashboard it the main interface which shows the path to all the above parts. User registration is designed to all the customers who use this system. They can register to the system using their personal details. Once they get registered they can edit detail inside “EDIT” button. If a customer thinks that this is not usable for them they can delete their account.

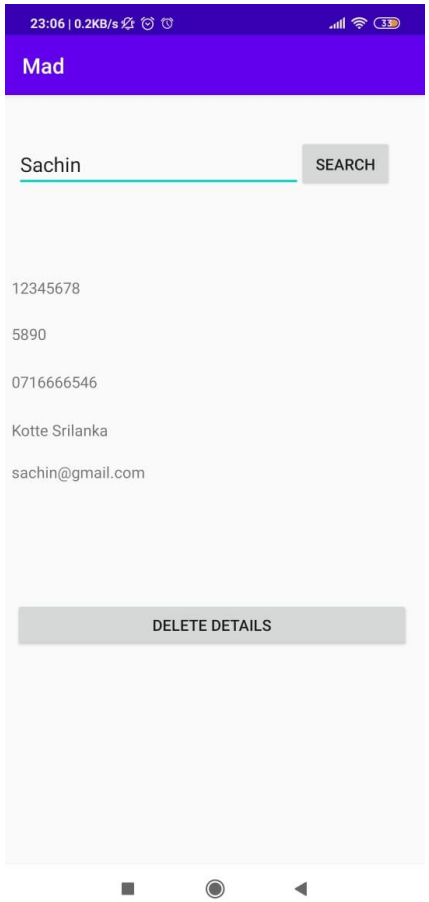
User log in is very important in this system. Customer should log in to our system whenever they want to buy something. Login can be done by using customer username and password.

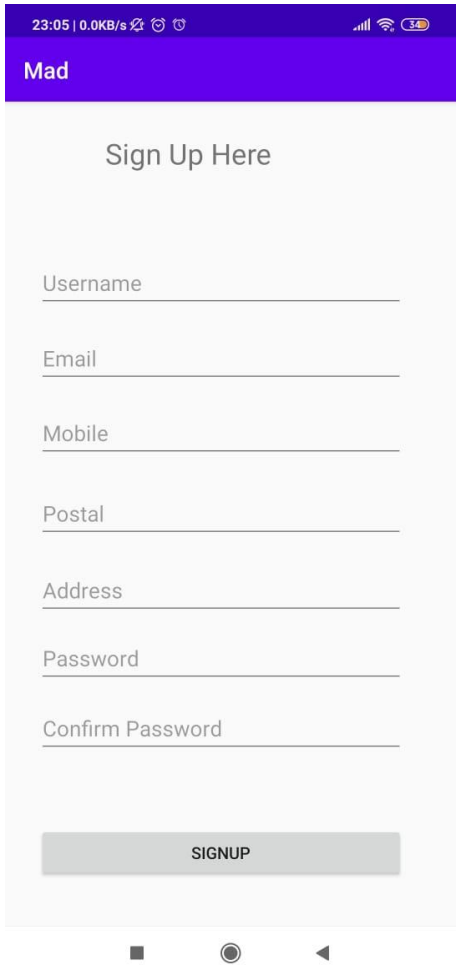
Deliver part is handled by the admin. When customer entered the deliver details and confirmed it will be send to admin. Then the admin can search for available drivers through the system. Once the admin assign a driver all the deliver details will be send to the driver through the system.

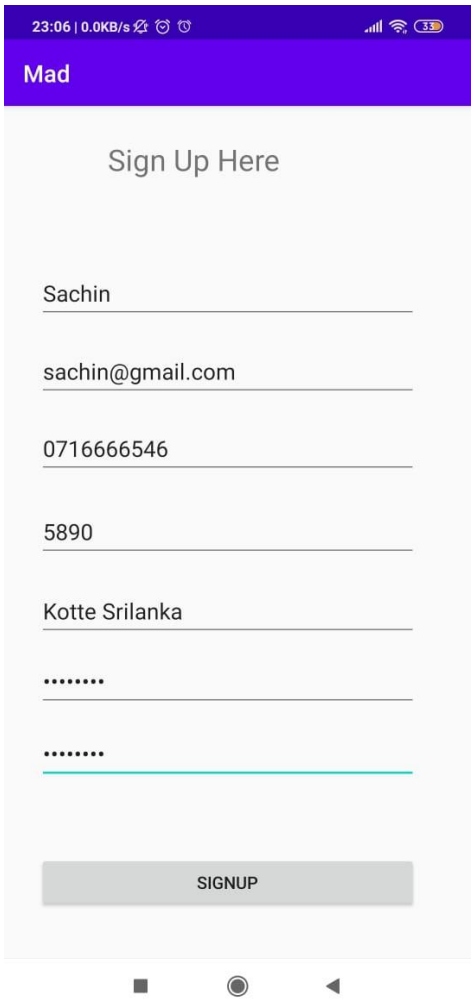
Snapshots of running Application



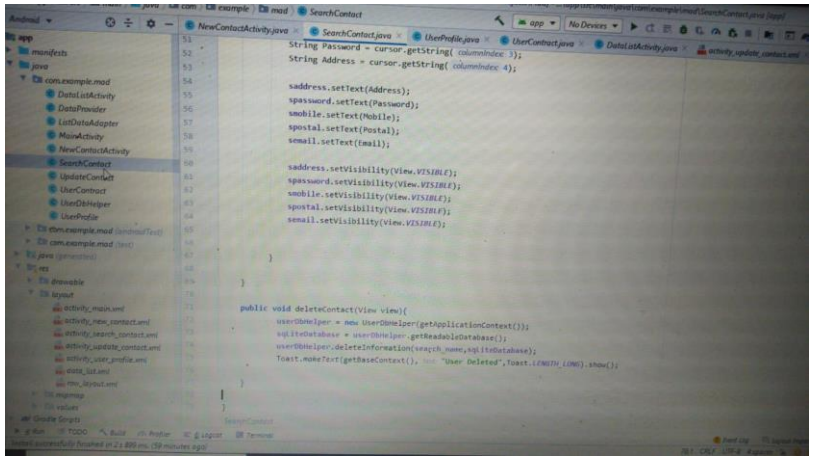


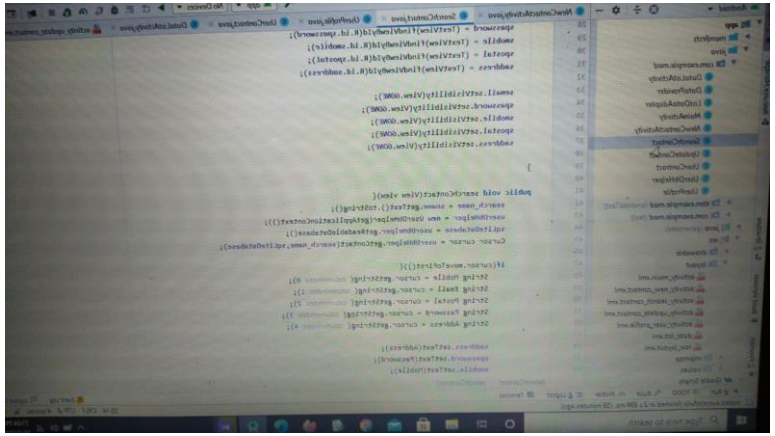


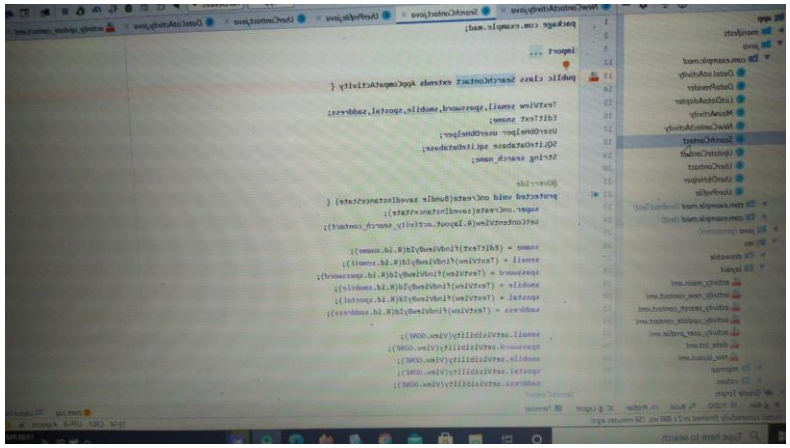

23:05 | 0.0KB/s
Mad
Sign Up Here
Username
Email
Mobile
Postal
Address
Password
Confirm Password
SIGNUP

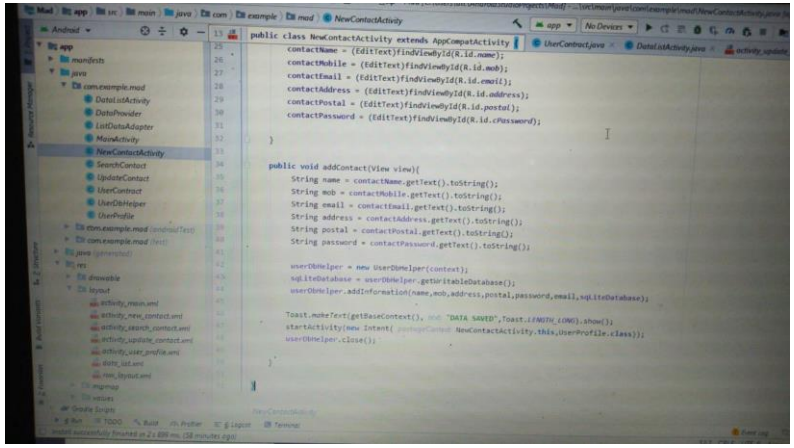

23:06 | 0.0KB/s
Mad
Sign Up Here
Sachin
sachin@gmail.com
0716666546
5890
Kotte Srilanka
SIGNUP

Calculations and Operations









```java
import ...

public class NewContactActivity extends AppCompatActivity {

    EditText contactName,contactMobile,contactEmail,contactAddress,contactPostal,contactPassword;
    Context context = this;
    UserDbHelper userDbHelper;
    SQLiteDatabase sqLiteDatabase;

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

        contactName = (EditText)findViewById(R.id.name);
        contactMobile = (EditText)findViewById(R.id.mob);
        contactEmail = (EditText)findViewById(R.id.email);
        contactAddress = (EditText)findViewById(R.id.address);
        contactPostal = (EditText)findViewById(R.id.postal);
        contactPassword = (EditText)findViewById(R.id.cPassword);

    }

    public void addContact(View view){
        String name = contactName.getText().toString();
        String mob = contactMobile.getText().toString();
        String email = contactEmail.getText().toString();
```

```java
public class DataListActivity extends AppCompatActivity {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.data_list);

    listView = (ListView)findViewById(R.id.list_view);

    listDataAdapter = new ListDataAdapter(getApplicationContext(),R.layout.row_layout);
    listView.setAdapter(listDataAdapter);

    userDbHelper = new UserDbHelper(getApplicationContext());
    sqLiteDatabase = userDbHelper.getReadableDatabase();

    cursor = userDbHelper.getInformation(sqLiteDatabase);
    if(cursor.moveToFirst()){
        do{
            String name,mob,email,postal,address,password;
            name = cursor.getString( columnIndex 0);
            mob = cursor.getString( columnIndex 1);
            address = cursor.getString( columnIndex 2);
            postal = cursor.getString( columnIndex 3);
            password = cursor.getString( columnIndex 4);
            email = cursor.getString( columnIndex 5);

            DataProvider dataProvider = new DataProvider(name,mob,address,postal,password,email);
            listDataAdapter.add(dataProvider);
        }while(cursor.moveToNext());
    }
}
```

```java
            layoutHandler = new LayoutHandler();
            layoutHandler.name = (TextView)row.findViewById(R.id.text_user_name);
            layoutHandler.address = (TextView)row.findViewById(R.id.text_user_address);
            layoutHandler.email = (TextView)row.findViewById(R.id.text_user_email);
            layoutHandler.password = (TextView)row.findViewById(R.id.text_user_pass);
            layoutHandler.postal = (TextView)row.findViewById(R.id.text_user_postal);
            layoutHandler.mob = (TextView)row.findViewById(R.id.text_user_mobile);

            row.setTag(layoutHandler);

        }
        else{
            layoutHandler = (LayoutHandler) row.getTag();
        }

        DataProvider dataProvider = (DataProvider)this.getItem(position);
        layoutHandler.name.setText(dataProvider.getUname());
        layoutHandler.mob.setText(dataProvider.getMobile());
        layoutHandler.postal.setText(dataProvider.getPostal());
        layoutHandler.password.setText(dataProvider.getPassword());
        layoutHandler.email.setText(dataProvider.getEmail());
        layoutHandler.address.setText(dataProvider.getAddress());

        return row;
    }
}
```

```java
package com.example.mad;

import ...

public class DataListActivity extends AppCompatActivity {

    ListView listView;
    SQLiteDatabase sqLiteDatabase;
    UserDbHelper userDbHelper;
    Cursor cursor;
    ListDataAdapter listDataAdapter;

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

        listView = (ListView)findViewById(R.id.list_view);

        listDataAdapter = new ListDataAdapter(getApplicationContext(),R.layout.row_layout);
        listView.setAdapter(listDataAdapter);

        userDbHelper = new UserDbHelper(getApplicationContext());
        sqLiteDatabase = userDbHelper.getReadableDatabase();

        cursor = userDbHelper.getInformation(sqLiteDatabase);
        if(cursor.moveToFirst()){
            do{
                String name,mob,email,postal,address,password;
```

```java
        super.add(object);
        list.add(object);
    }

    public int getCount() { return list.size(); }

    public Object getItem(int position) { return list.get(position); }

    public View getView(int position, View convertView, ViewGroup parent){
        View row = convertView;
        LayoutHandler layoutHandler;
        if(row == null){
            LayoutInflater layoutInflater = (LayoutInflater)this.getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            row = layoutInflater.inflate(R.layout.row_layout,parent, attachToRoot: false);

            layoutHandler = new LayoutHandler();
            layoutHandler.name = (TextView)row.findViewById(R.id.text_user_name);
            layoutHandler.address = (TextView)row.findViewById(R.id.text_user_address);
            layoutHandler.email = (TextView)row.findViewById(R.id.text_user_email);
            layoutHandler.password = (TextView)row.findViewById(R.id.text_user_pass);
            layoutHandler.postal = (TextView)row.findViewById(R.id.text_user_postal);
            layoutHandler.mob = (TextView)row.findViewById(R.id.text_user_mobile);

            row.setTag(layoutHandler);

        }
        else{
```