

MIDTERM PROJECT

CS426 – Mobile Device Application Development

Group members :

- Do Le Duan , 18125127
- Bui Xuan Vinh , 18125119
- Cao Ngoc Son , 18125111

Description :

For people who have demand in finding house chores helper in a short amount of time , we make a mobile application “Find My Helper” . This platform will allow everyone to search helper with necessary information nearby through a google maps by using an account which can be easily created . Concurrently , the customers could assess the helpers by stars from 1 – 5 , those assessment will be stored and the others could rely on to pick up a suitable helper in the most convenient way

Novelty :

It 's inconvenient for people who are always busy to come to the job centers and find proper house chores helper . With this app , we simply choose right helper in anywhere at anytime based on real-time assessment compared with helpers from centers that we have no information about them at all .

Moreover , this platform will not only create customers but also aim to people who want to have a helper job . With smart device , they could easily create profile for them online without any fees

Major functions , Techniques & Implementation

Our app consists of 7 class in Java language

- CurrentHelperHired
- CustomerHire
- HelperInfor
- HelperProfile
- Login
- Maps
- Signin

1. HelperInfo

```
public class HelperInfor implements Parcelable {

    String HName;
    String Gender;
    String DOB;
    String Address;
    String Notes;
    Float Rating;
    Integer avatar=0;
    String phone;
    boolean available = false;
    LatLng latLng;

    public HelperInfor(){
        this.HName=null;
        this.Gender = null;
        this.DOB = null;
    }
}
```

In this class, just declare attributes that a helper can have (as we assume). And have several getter and setter which can help during the project.

The helperinfor in this project we use the parcelable instead of the serilizable thanks to the faster and more easier to transmis the data in each activities of its. We write each information of helper into file to read later.

2. HelperProfile

```
public class HelperProfile extends AppCompatActivity {

    RatingBar ratingBar;//only for showing
    HelperInfor helperInfor = new HelperInfor();
    int index = 0;
    Intent intentIn=null;
    Intent intentToHired = null;

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

        intentIn = getIntent();
        Bundle arg = intentIn.getBundleExtra( name: "bundle_arg");
        helperInfor = (HelperInfor) arg.getParcelable( key: "helperInf");

        Button booking = findViewById(R.id.booking);
        booking.setOnClickListener((view) -> {
            if(helperInfor.available){
                //...
            }
        });
    }
}
```

This activity is used to represent profile of specific Helper when they are clicked on map.

Main function here is `initHelperProfile()`;

Button Booking can be activated when this Helper is available for help other Customers. If not, you will get an announcement for that situation

```
private void initHelperProfile() {  
    ratingBar = findViewById(R.id.ratingCus);  
  
    TextView name = findViewById(R.id.fullNameCus);  
    TextView gender = findViewById(R.id.sexCus);  
    TextView dob = findViewById(R.id.dob);  
    TextView address = findViewById(R.id.addsCus);  
    TextView note = findViewById(R.id.note);  
    TextView phone = findViewById(R.id.phoneCus);  
    ImageView avatar = findViewById(R.id.avatarCus);  
  
    name.setText(helperInfor.getHName());  
    gender.setText(helperInfor.getGender());  
    dob.setText(helperInfor.getDOB());  
    address.setText(helperInfor.getAddress());  
    avatar.setImageDrawable(getResources().getDrawable(helperInfor.avatar));  
    note.setText(helperInfor.getNotes());  
    phone.setText(helperInfor.getPhone());  
    ratingBar.setRating(helperInfor.getRating());  
}
```

```
Button booking = findViewById(R.id.booking);  
booking.setOnClickListener((view) -> {  
    if(helperInfor.available){  
        Toast.makeText( context: HelperProfile.this, text: "You can book this helper, because they are available",  
            Toast.LENGTH_SHORT).show();  
  
        intentToHired = new Intent( packageContext: HelperProfile.this, CustomerHire.class);  
        Bundle arg2 = new Bundle();  
        arg2.putParcelable("modifyHelper", helperInfor);  
        intentToHired.putExtra( name: "bundle_modify", arg2);  
  
        startActivity(intentToHired);  
        finish();  
    }  
    else{  
        Toast.makeText( context: HelperProfile.this, text: "You can not book this helper, because they are not available",  
            //return to current page because they are not available  
        );  
    }  
});  
  
initHelperProfile();
```

In the HelperProfile, we init the profile for each helper, that most considerable is the rating bar, and the button book is put under the rating in order to the customer can book this helper. The major function in this is the booking onclicklistener. If someone has booked this helper, we will set the available of him/her to be false, so that, others customer cannot book this helper.(Toast that the customer cannot book if someone has booked by the following code)

3. CurrentHelperHired:

```
public class CurrentHelperHired extends AppCompatActivity implements View.OnClickListener {

    Intent intentIn;
    Button vote;
    ImageView avt;
    TextView name, phone;
    HelperInfor mhelper = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    private void getCurrentHelper() {...}

    private void initComponents() {...}

    private void clearToUsersFile() {...}

    private void writeToHelperFile() {...}
```

This activity to show the current helper that user hired.
Main functions is to write/read helper infor and write to users file.

4. CustomerHire

```
TextInputEditText address, description, day,time, purchase;
ActionBar actionBar;
Intent intentHelper = null;
HelperInfor helperInfor = new HelperInfor();
Button send;

@Override
protected void onCreate(Bundle savedInstanceState) {...}

private void writeToUsersFile() {...}

private void modifyActionBar() { actionBar.setTitle("Request to " + helperInfor.getName()); }

private void initEditTexts() {...}

@Override
public void onBackPressed() {...}

private void changeInFile(HelperInfor helperInfor) {...}
```

This is show the form that a customer must to fill all the necessary information which will be sent to the selected helper such as (customer's address, time to come, day to come and salary that will be paid to helper)

Main functions : writeToUsersFile, changelnFile. Because, whenever the request is sent, that helper can not be hired by another customer.

5. Maps

We use API from google to provide map for the customer

```
private void getInforHelper() {  
    HelperInfor hlpr1 = new HelperInfor();  
    HelperInfor hlpr2 = new HelperInfor();  
    HelperInfor hlpr3 = new HelperInfor();  
  
    readFromFileHelper(R.drawable.xuan_vinh, hlpr1);  
    readFromFileHelper(R.drawable.ngoc_son, hlpr2);  
    readFromFileHelper(R.drawable.le_duan, hlpr3);  
  
    _helpers.add(hlpr1);  
    _helpers.add(hlpr2);  
    _helpers.add(hlpr3);  
}
```

At first, this begins with login layout where customers can login by their account they already created, or they can create in the rest case. Continue, we go to the maps activity where we can choose helper by clicking on the marker by the functions:

```
@Override  
public boolean onMarkerClick(Marker marker) {  
    HelperInfor tmp = new HelperInfor();  
    int index = Integer.valueOf(String.valueOf(marker.getTag()));  
    tmp = _helpers.get(Integer.valueOf(String.valueOf(marker.getTag())));  
  
    intentOut = new Intent( packageContext: Maps.this, HelperProfile.class);  
    if(tmp!= null){  
        Bundle arg = new Bundle();  
  
        arg.putParcelable("helperInf", (Parcelable) tmp);  
  
        intentOut.putExtra( name: "bundle_arg", arg);  
  
        startActivity(intentOut);  
    }  
    return false;  
}
```

The helper we create by the transmission data into the file_name.txt with the file_name equal to the value of their avatarID thanks to function readFromFileHelper. Also by using the address of the helper provided, we convert it into Latlng by function ConvertAddtoLatlng to get position of marker on the maps. Furthermore, if we click on the marker (represent for the helper) the intent will be send to the HelperProfile to create the profile of helper which include the image, phone, address,gender of the helper.

6. Login

```
mLoginBtn.setOnClickListener((v) -> {
    String email = mEmail.getText().toString().trim();
    String password = mPassword.getText().toString().trim();
    if(TextUtils.isEmpty(email)){
        mEmail.setError("Email is required");
        return;
    }
    if(TextUtils.isEmpty(password)){
        mPassword.setError("Password is required");
        return;
    }
    if(password.length() < 6){
        mPassword.setError("Password must be six or more character!");
        return;
    }
    progressBar.setVisibility(View.VISIBLE);
    FirebaseAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener((task) -> {
        if(task.isSuccessful()){
            Toast.makeText( context: Login.this, text: "Log in successfully" + FirebaseAuth.getCurrentUser().getUid()
                ,Toast.LENGTH_SHORT).show();
            startActivity(new Intent( packageContext: Login.this, Maps.class));
        }else{
            Toast.makeText( context: Login.this, text: "Error !" + task.getException().getMessage(),Toast.LENGTH_SHORT).show();
        }
    });
});
```

We created Login interface to allow people to use account and password to login app

We connect Firebase to store the customer account and password

This function is to test the format of the account (we use email format) the length of password (must >6 characters) and the appearance of password

If login successfully , it will jump to Maps class else it will announce error

```
private void writeToFile() {
    String file_name = String.valueOf(R.drawable.xuan_vinh);
    try {
        FileOutputStream fos = openFileOutput( name: file_name+".txt", mode: 0);
        OutputStreamWriter osw = new OutputStreamWriter(fos);
        BufferedWriter bw = new BufferedWriter(osw);
        bw.write( str: "Bùi Xuân Vinh" + '\n');
        bw.write( str: "0989778966" + '\n');
        bw.write( str: "Male" + '\n');
        bw.write( str: "14/09/2000" + '\n');
        bw.write( str: "231/83/8A Đường Bá Trạc, Phường 1, Quận 8" + '\n');
        //bw.write("220 Đường Bùi Hữu Nghĩa, Phường 2, Quận Bình Thạnh" + '\n');
        bw.write( str: "Work early in the noon" + '\n');
        bw.write( str: "2" + '\n');
        bw.write( str: String.valueOf(Integer.valueOf(R.drawable.xuan_vinh)) + '\n');
        bw.write( str: "true" + '\n');

        bw.close();
        fos.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

This function will provide information for the helper who will appear on the maps

7. Signin (Register account)

```
mRegisterBtn.setOnClickListener((v) -> {  
    String email = mEmail.getText().toString().trim();  
    String password = mPassword.getText().toString().trim();  
    if(TextUtils.isEmpty(email)){  
        mEmail.setError("Email is required");  
        return;  
    }  
    if(TextUtils.isEmpty(password)){  
        mPassword.setError("Password is required");  
        return;  
    }  
    if(password.length() < 6){  
        mPassword.setError("Password must be six or more character!");  
        return;  
    }  
    progressBar.setVisibility(View.VISIBLE);  
    mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener((task) -> {  
        if(task.isSuccessful()){  
            Toast.makeText( context: SignIn.this, text: "User created",Toast.LENGTH_SHORT).show();  
            startActivity(new Intent(getApplicationContext(),Login.class));  
        }else{  
            Toast.makeText( context: SignIn.this, text: "Error !" + task.getException().getMessage(),Toast.LENGTH_SHORT).show();  
        }  
    });  
});
```

Like Login class , Register will check the format of Email , Password . If account is registered successfully , it will go back to Login to enter in the Maps

```
mLoginBtn.setOnClickListener((v) -> {  
    startActivity(new Intent( packageContext: SignIn.this, Login.class));  
});
```

If you have already have an account , this will immediately help you go back to login


Note : With Login and Register , we connect with FireBase via google Account

Username : duandole9199@gmail.com


Password : 0131914567


You can use this account to check


Demo app using sreenshots




Create your new account

 duan




 leduan56@gmail.com


 0825641014


REGISTER

Already have account? [Login here](#)



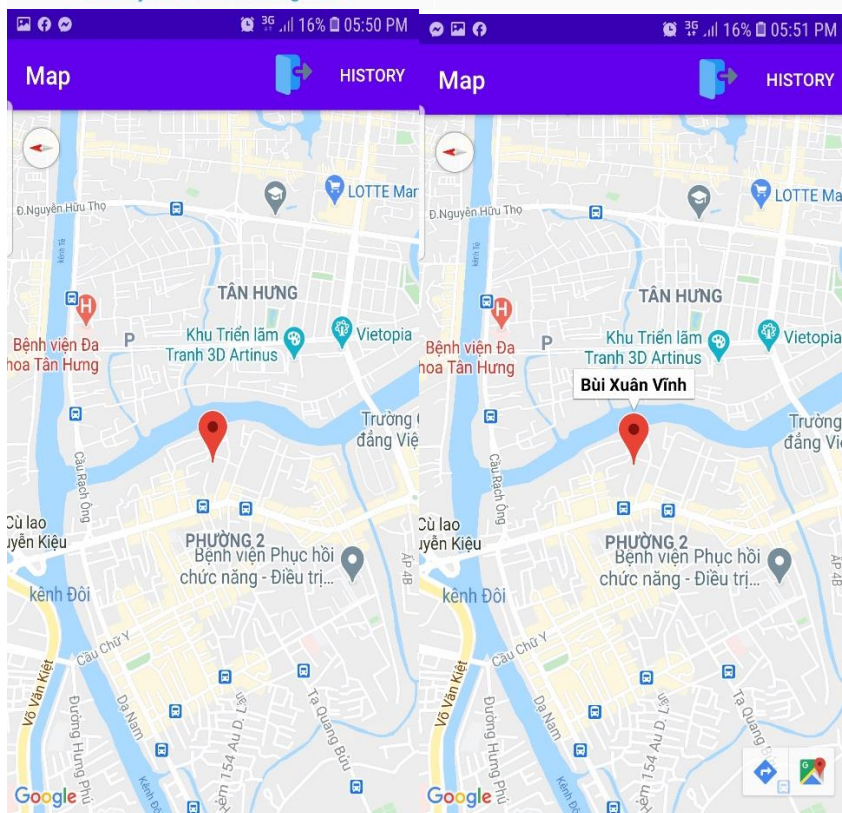
Wellcome to our app

 leduan56@gmail.com



Create your account?


LOGIN



First we create an account and login successfully , It will show the maps and we can see the red marker which means there is one helper in that place . Click on the marker we can see the name of the helper


ProjectCS426

ProjectCS426



Bùi Xuân Vĩnh

0989778966



Full Name

Phone Number


FINISH

Male

14/09/2000


231/83/8A Dương Bá Trạc, Phường 1, Quận 8

Work early in the noon




BOOK


You can not book this helper, because they are not available



Wellcome to our app




hunghang12@gmail.com



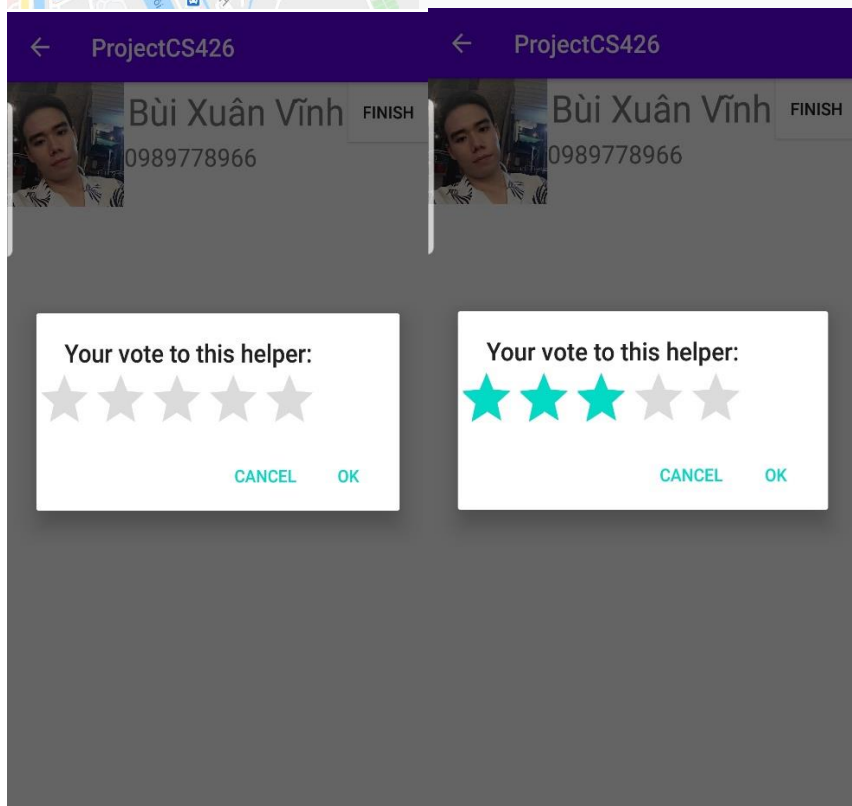
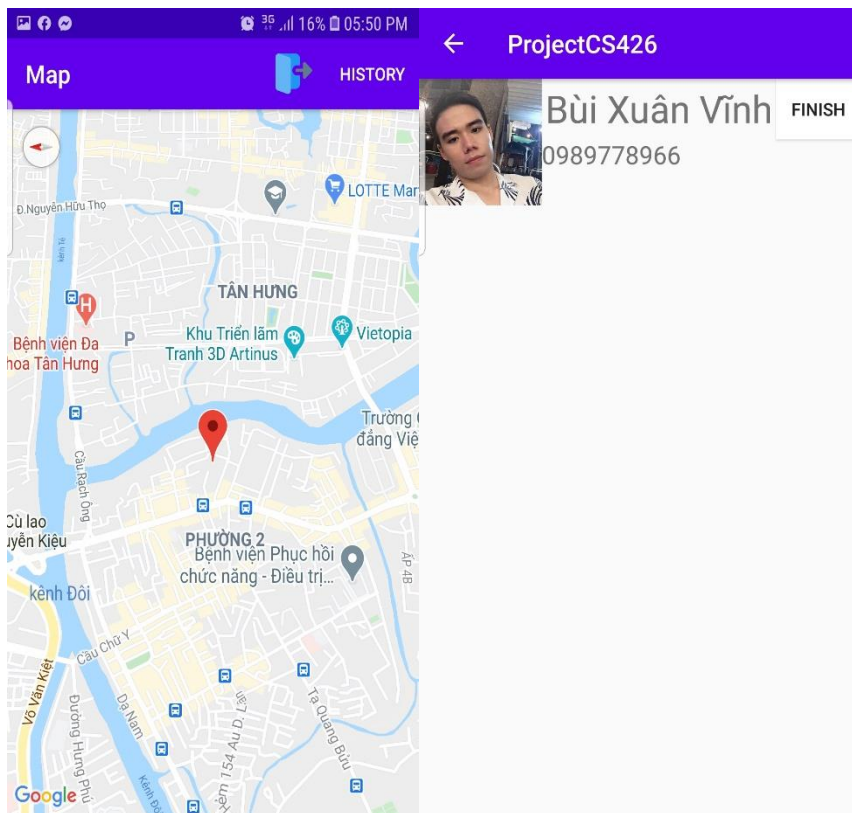
.....|

Create your account?

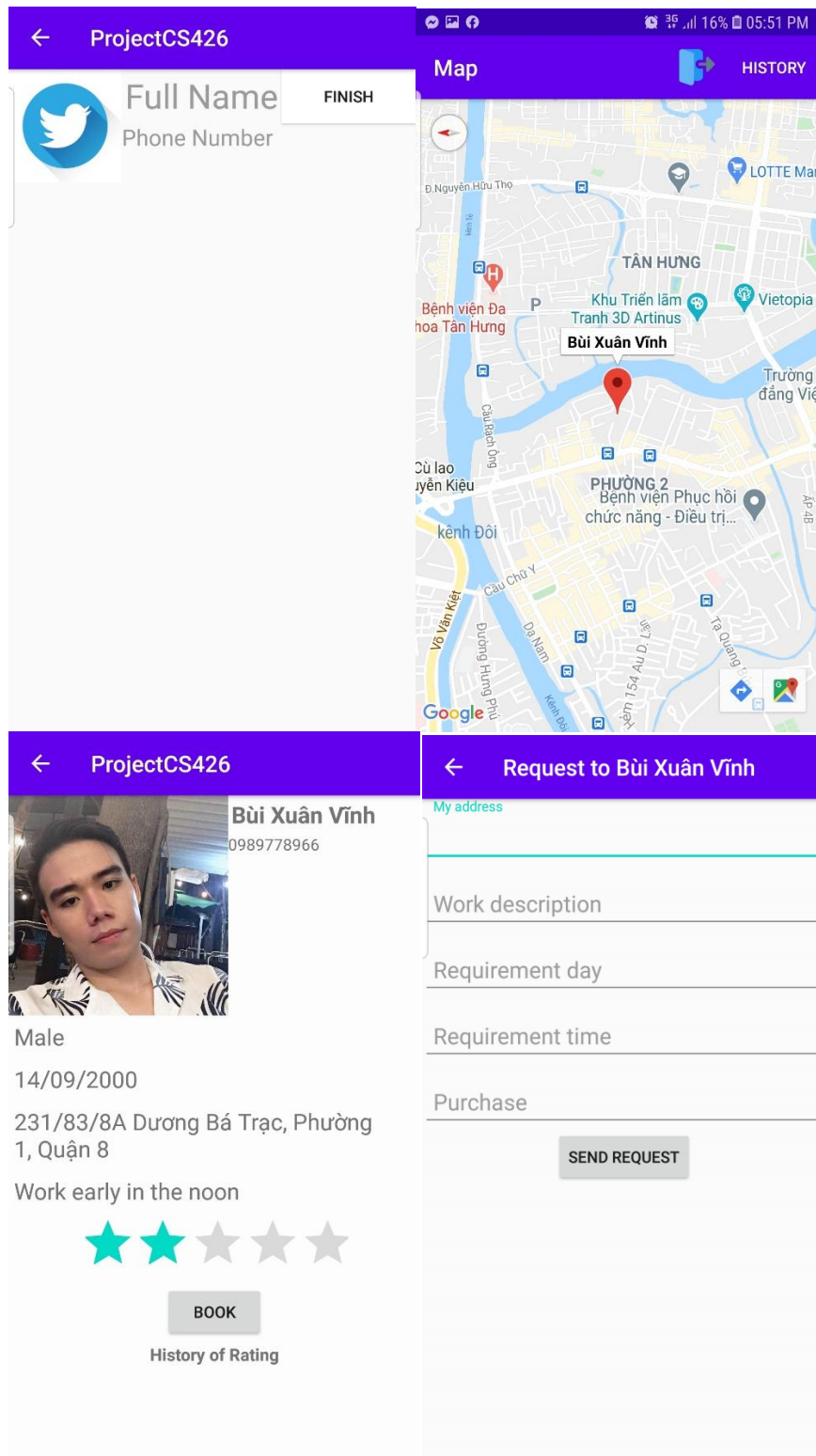
LOGIN

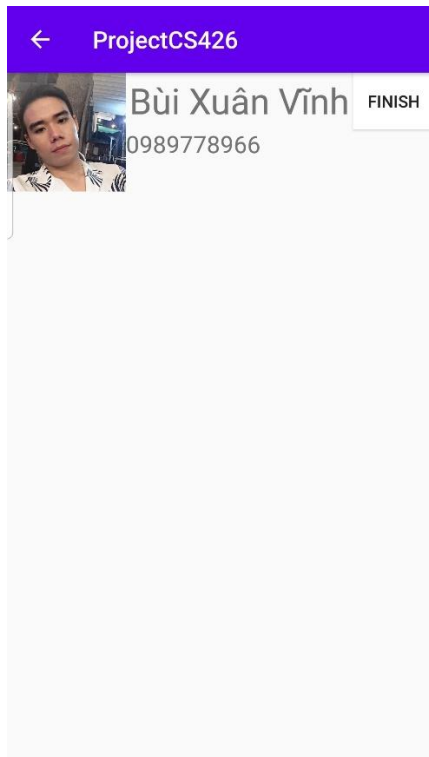


We cannot book this helper because the hunghang12@gmail.com which was created by us before , booked this helper . So that we go into hunghang12@gmail.com and the history of leduan56@gmail.com are still nothing



We click on the history and see the helper had been booked by account hanghung12@gmail.com. To end this we must rate 1 – 5 star for this helper to let them free because when finish hiring, customer often rate for the helper and click finish





Now we go to the leduan56@gmail.com account and book this helper as normal and it will go into the history

Material

- Firebase via <https://firebase.google.com/>
- Tourist App Assignment
- Stackoverflow