



Zagazig University
Faculty of computer and informatics
Information Technology Department

Graduation Project Report

Project name

Kidnapping Protecting System

Supervisor:

Dr.Osama Elkomy

By assistance of:

Eng.Eman Salem

Team Member:-

- 1. Mahmoud Mohamed Mahmoud ali***
- 2. Ahmed Mohamed khedira ahmed.***
- 3. Mohamed Yasser Mohammed Essa***
- 4. Sara Zaher Ibrahiem Elsayed .***
- 5. Marina Wageh zakria Gerges***
- 6. Amal Emad Elsayed fadel***
- 7. Asmaa Mohamed Saad Mohamed.***

Abstract:

In our world there is more development and we must adapt to this development so we made a mobile application that is an android based application to serve people from protecting themselves from any danger like kidnapping of humans and helping the police man for searching about the licences which finished and this help policeman for saving time.

This documentation describes our graduation project in six chapters:

- *In Chapter one you will find a brief introduction about the project.*
- *Chapter two discuss the application methodology, the main work done in developing application, the software and API's used in the project.*
- *In Chapter three you will find in this chapter the analysis process of this project.*
- *In Chapter four you will find the design process of the application.*
- *In Chapter five implementation process and also in this chapter we found the user interface design for humans for protecting themselves from kidnapping from risks from taxi..*
- *In Chapter six implementation process and also in this chapter we found future for using the application for police man for searching about the licences which finished .*

Content list:

Acknowledgement **8**

Chapter 1:

Section 1.1: Introduction. **10**

Section 1.2: Business value- **11**

Chapter 2:

Main work. **13**

2.1.1 Divide application main work **3**

2.1.1.1 learning about android developing

2.1.1.2 basic android devices

2.1.1.3 Advantages of android **12**

2.1.1.4 Disadvantages of android **12**

2.1.1.5 Android Activity Lifecycle **13**

2.1.2 Learning about php **14**

2.1.3 Database about the project.....**14**

<i>Section 2.2</i>	
<i>project requirements</i>	<i>15</i>
2.2.1 Applications languages and API's used.	. 15

Chapter3

Analysis of the project:-

3.3.1 Use case diagram. 17
3.3.2 Elements of a Use Case.....	
3.3.2.1 Elements of Data Flow Diagrams.....	
3.3.3 Entity relationship diagram.	

Section 3.1: Our work. 25
3.1.1. Android creation. 25
3.1.2. Work on wamp server 31

Chapter 4

System design

4.1.1: *System design technique*.....23

4.1.2: *What is logical data model*.....24

Section 4.2.:ER DIAGRAM MODEL. 25

Chapter 5

Android interface design

5.1.2.1 *log in*30

5.1.2.2 *log in in xml*.....31

5.1.2.3 *register*32.

5.1.2.4 *searching for car*33.

5.1.2.5 *choosing the car*.....36

5.1.2.6 *maps*.....37.

5.1.2.7 *bleutooth*.....38

Chapter 6

Introduction

section6.1

6.1.1 log in 40

6.1.2 searching for the car..... 42

6.1.3 after finding the car 43

Section 6.2

6.2.1 search for user 44

6.2.2 map in the application 45

List of Figures

CHAPTER1

Figure 1:. life cycle of android. 25

CHAPTER2

Figure 2:. Applications languages and API's used 27

Figure 3:. Use case diagram. 31

Figure 4:. Entity Relationship Diagram 37

CHAPTER4

Figure 5:. Data base for tawseila 41

Figure 6:. Data base for user. 42

Figure 7:. Data base for car and driver 45

CHAPTER5

Figure 8:. log in page 22

Figure 9:.log in page in xml 24

Figure 10:. Sign up page 28

Figure 11:. Sign up page in xml 29

Figure 12:. Car search page 31

Figure 13:. Choosing car 32

Figure 14:. Start activity 33

Figure 15:. Maps. 38

Figure 16:.Beluttoth 39

Chapter 6

<u>Figure 17:..</u> log in page	40
<u>Figure 18:..</u> Searching for the car	
<u>Figure 19 :..</u> After finding car	42
<u>Figure 20:..</u> Trip pass	52

◆ Acknowledgement:-

- *In every case Good worse..... We must thank the great merciful Allah for all gifts and blessing that we have been given starting with the blessings of Islam.*

We would like also to thank our supervisor

Dr. Osama Elkomy

- *Who directed us to new color of science, drawing and planning most of all our steps providing us good resources and references for finishing that work?*

We would like also o thank:

Eng. Eman Salem

- Who helped in solving the problems and difficulties that we face and was step by step with us to finish that work?
 - We would like also to thank everyone helped us through our gradual education through the four years especially the staff members of information technology department.
 - We hope we learned well as much as we can to be useful for Islam and Muslims hoping for Allah satisfaction every time and everywhere.
-
- To every doctor and every assistant who offer us knowledge and help through the past few years.
 - We dedicate this humble work.
 - To our fathers ,our mothers and our brothers.
 - to our friends who helped us and continue .
-
- .to ones who have provided advice to us.
 - To every person who made us better .
 - All praise to Allah most gracious, most merciful and his peace and blessing is upon his family, companions and whoever follows him.

Chapter 1

Section1.1:-

Introduction

Was and still the fastest services very important topic that preoccupies every one. Because of its importance in the development of our environment, All countries seek to give a large space of interest to look for a way to end their demands firstly and save their times for people. And from here we got the idea of our graduation project which is "CheckCar".

Problem:

As a result of the high rates of kidnapping in the taxi and the different means of transportation, we seek to solve this problem through application on the phone to track the passengers and to reduce kidnappings and In addition, this application is used to assist traffic men to know the duration of the driving license.

Section1.2:-

Business value:

Business value describes the benefits that the organization should expect from the system.

Special issues are included on the document as a catchall category for other information that should be considered in assessing the project.

For example:

The project may need to be completed by a specific deadline

Chapter2

Section 2.1:-

Main work

2.1 Divide application main work.

- We can divide our application's main work into main three parts:-*

2.1.1.1. Learning android developing basics:

Before developing our application we needed to read about android because knowledge about android developing was not sufficient enough to start developing the project.

2.1.1.2. Android devices:-

Android is a mobile operating system based on the Linux kernel and now developed by Google , It is designed for the touch screen mobile devices like the smartphones , the table computers , specialized user

interface for Android TV and Android enabled vehicles .

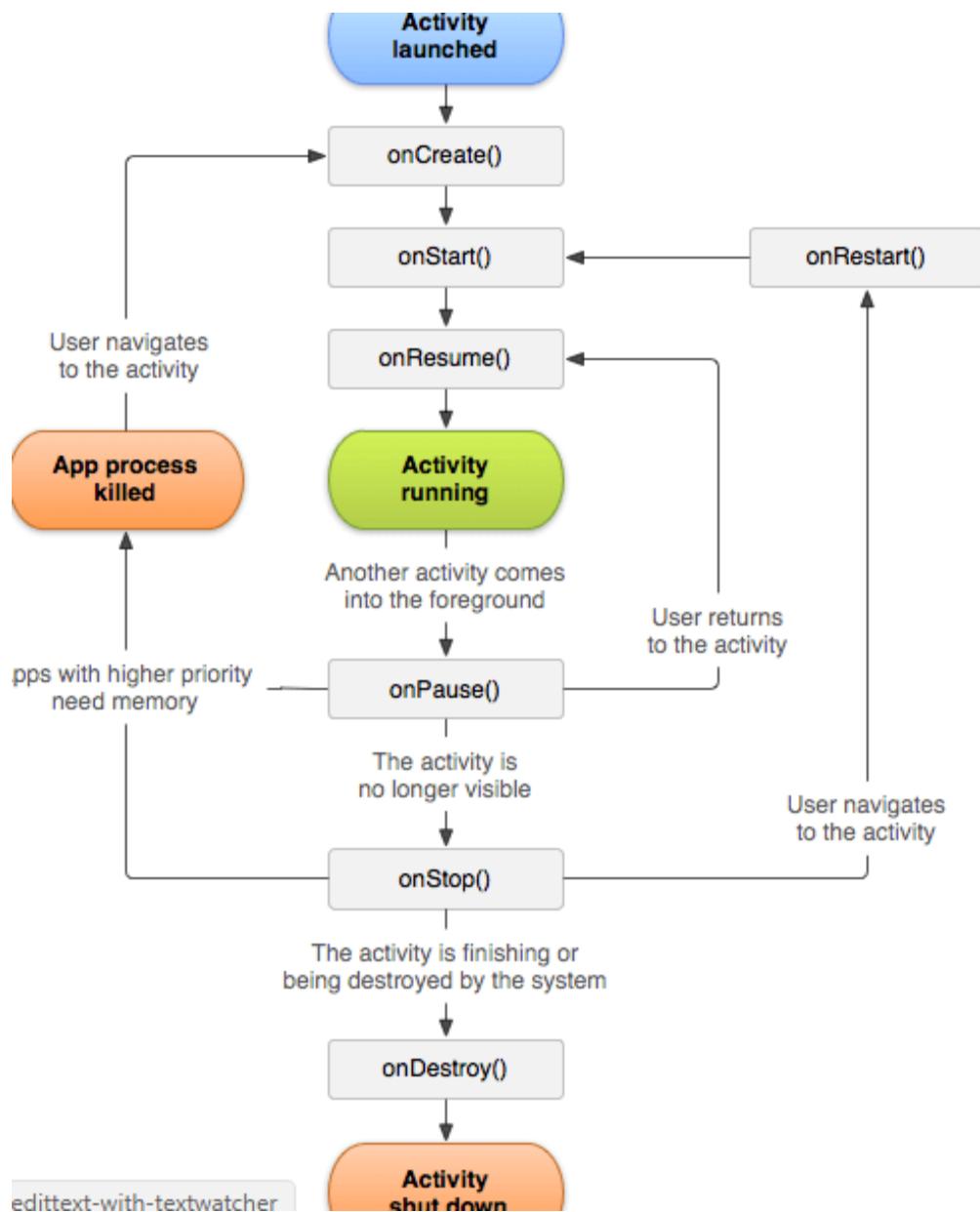
2.1.1.2. Advantage of Android

- *it has massive user base , It has increasing adoption especially in the developing countries , Android's review process for apps is fairly simple and it takes less time than iOS*
- ***for an app to get approved for publishing on the play store .***
- *Android phones can run many applications , You can browse , You can use Facebook while listened to the song , It has different phone options , It has thousands of Apps to download online and they are free .*
- *It has excellent software and application support , there are wide variety of phones to suit your budget , It has frequent OS updates for improving performance ,Android phones can also function as a router to share Internet .*

2.1.1.3 Disadvantages of android:-

- *There are multitude of devices with different screen sizes and different resolutions too , This makes the app design as well as UI development much harder and the quality of apps on the Play Store is much worse than those on the App Store .*
- *Many applications contain virus , The virus inserted android applications including Counter Strike Ground Force , Android applications contain virus also present in the Android market .*
- *Android is more wasteful than any other operating system as this operating system has a lot of process in the background that lead to the battery quickly drains .*
- *Android requires an active Internet connection , There should be a GPRS internet connection in your area , so that the device is ready to go online according to our needs .*

2.1.1.4. Android Activity Lifecycle methods.



(Figure 1).life cycle of android.

2.1.2 learning about php:-

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used

open source general-purpose scripting language that is especially suited for

web development and can be embedded into HTML.

2.1.3 Database about the project :-

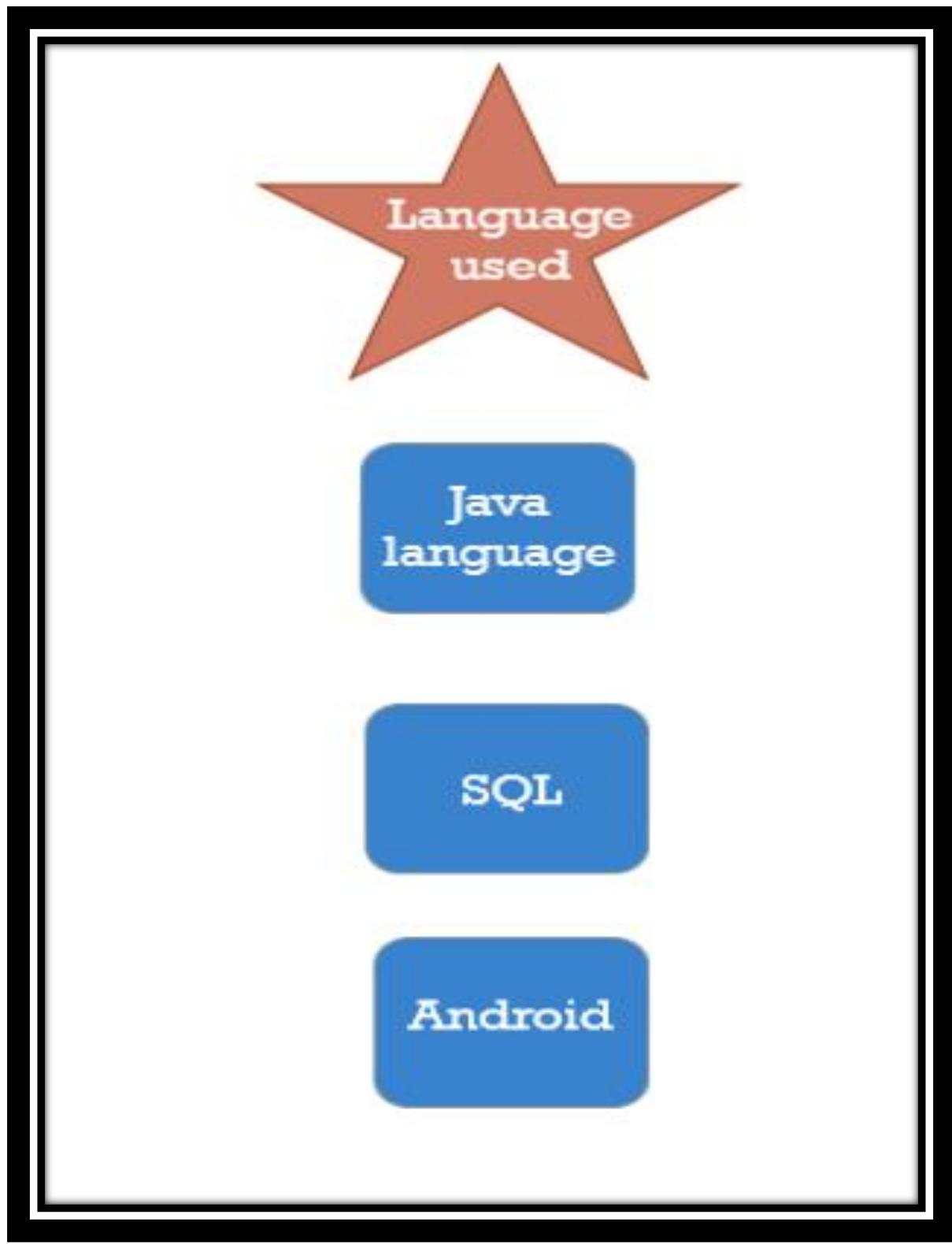
Data base which contain information about driver ,car and tawselia.

Data base written with php which any person can read it.

Section 2.2:-

Project requirements

2.2.1. Applications languages and API's used



(Figure 2) Applications languages and API's used (1).

Chapter3

Section3.k:-

Analysis of the project:-

The analysis of the role of a proposed system and the identification of the requirements that it should meet.

◆ Is the analysis of the project in two phases:-

1. *Use case diagram.*
2. *data flow diagram.*
3. *Entity relationship diagram.*

3.3.1 Use case:

are used to explain and document the interaction that is required between the user and the system to accomplish the user's task. Use cases are created to help the development team understand more fully the steps that are involved in accomplishing the user's goals. Once created, use cases often can be used to derive more detailed functional requirements for the new system.

3.3.2 Elements of a Use Case:-

➤ Use Case.

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

➤ *Actors.*

An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.

➤ *Association.*

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.

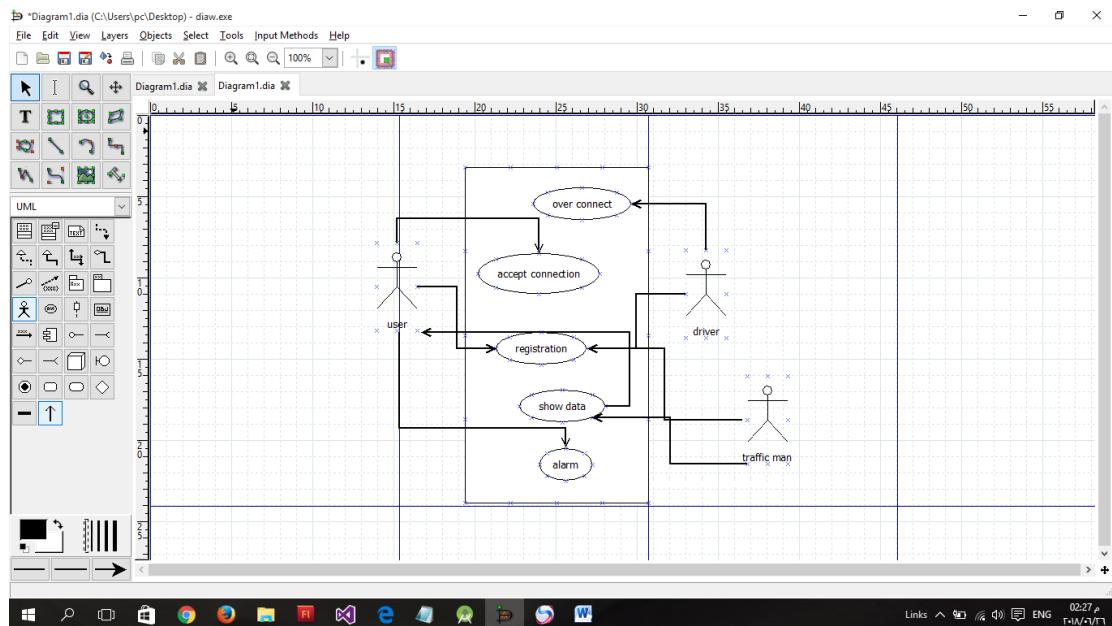
➤ *System boundary boxes (optional).*

You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.

➤ *Packages (optional).*

Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class

diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single page, to organize a large diagram into smaller ones.



(figure3):- use case diagram

3.3.3 Data flow diagram :-

data flow diagram (DFD) implies a focus on data, this is not the case. The focus is mainly on the processes or activities that are performed. Data modeling, discussed in the next chapter, presents how the data created and used by processes are organized. Process modeling—and creating DFDs in particular—is one of the most important skills needed by systems analysts.

3.3.3.1 Elements of Data Flow Diagrams:-

- *Process:-*

A process is an activity or a function that is performed for some specific business reason. Processes can be manual or computerized.

- *Data Flow:-*

A data flow is a single piece of data (sometimes called a data element), or a logical collection of several pieces of information.

- *Data Store:-*

A data store is a collection of data that is stored in some way (which is determined later when creating the physical model)

3.3.4 ENTITY RELATIONSHIP DIAGRAM:-

entity relationship diagram (ERD) is a picture which shows the information that is created, stored, and used by a business system. An analyst can read an ERD to discover the individual pieces of information in a system and how they are organized and related to each other.

3.3.4.1 Elements of an Entity Relationship

Diagram:-

- *Entity:-*

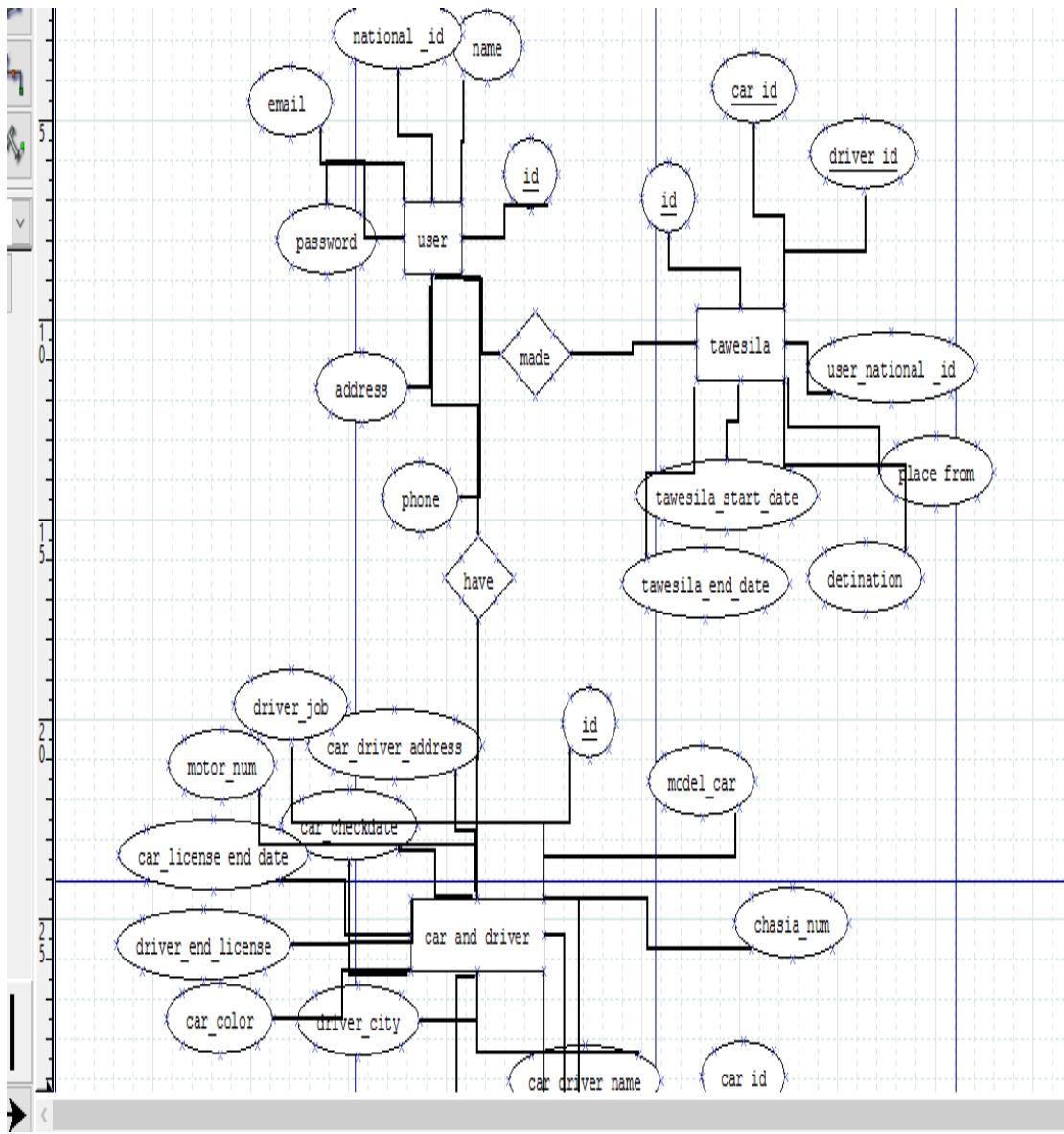
The entity is the basic building block for a data model. It is a person, place, event, or thing about which data is collected.

- *Attribute :-*

An attribute is some type of information that is captured about an entity.

- *Relationship :-*

Relationships are associations between entities, and they are shown by lines that connect the entities together.



(Figure) Entity Relationship Diagram.

Chapter4

Section 4.1:-

Systems design:-

Is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements?

4.1.1 Systems Design Techniques:-

- **Logical Data Modeling**
- **Entity/Event Modeling**
- **Logical Data Modeling**

The objectives of this section are to provide definitions for the terms Logical Data Model (LDM) and Logical Data Structure (LDS), and to explain the constituent parts of a Logical Data Structure, and finally to introduce a step by step process for constructing Logical Data Models.

4.1.2 what is a Logical Data Model?

A Logical Data Model (LDM) is a representation of the data used by a system. It shows how the data is logically grouped and the relationships between these groupings as defined by the business requirements of the system.

The LDM comprises:-

A diagram called a Logical Data Structure (LDS). NB LDS is simply the SSADM terminology for a Data Model or Entity-Relationship Model). Associated documentation of entities and relationships

4.2 ER Diagram

Here snapshots to relations used in database

And brief explanation to each of them .

4.2.1 The database construction from MySql.

The screenshot shows the phpMyAdmin interface for the database 'id5115588_checktaxi'. The left sidebar shows the database structure with tables: car_and_driver, tawsila, and user. The main area displays the table structure for these three tables.

Table	Action	Rows	Type	Collation	Size	Overhead
car_and_driver	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
tawsila	Browse Structure Search Insert Empty Drop	95	InnoDB	latin1_swedish_ci	16 KiB	-
user	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 KiB	-
3 tables	Sum	109	InnoDB	utf8_unicode_ci	48 KiB	0 B

Below the table list, there is a 'Create table' form with fields for Name (empty) and Number of columns (4). A 'Go' button is located to the right of the column count input field.

(figure 4):- tables for database.

4.2.1.1 Table for tawsila:-

(figure 5):- tables for tawselia

The screenshot shows the phpMyAdmin interface for the database 'id5115588_checktaxi'. The left sidebar lists databases: 'New', 'id5115588_checktaxi' (selected), 'New', 'car_and_driver', 'tawsila' (selected), 'user', 'information_schema', and 'mysql'. The top navigation bar includes links for 'Secure', 'Manage Databases', 'ITI Intak 35 EAD', 'C# Tutorial - Kno', and 'C# Sharp program'. The main content area displays the 'Table structure' for the 'tawsila' table. The table has 8 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop
2	car_id	varchar(500)	latin1_swedish_ci		No	None			Change Drop
3	driver_id	varchar(225)	latin1_swedish_ci		No	None			Change Drop
4	user_national_id	varchar(14)	latin1_swedish_ci		No	None			Change Drop
5	place_from	varchar(500)	latin1_swedish_ci		Yes	NULL			Change Drop
6	distenation	varchar(500)	latin1_swedish_ci		Yes	NULL			Change Drop
7	tawsial_start_date	varchar(45)	latin1_swedish_ci		No	None			Change Drop
8	tawsial_end_date	varchar(45)	latin1_swedish_ci		Yes	NULL			Change Drop

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'. A 'Print' button is also present. At the bottom, there is a search bar for 'Add column(s)' and a 'Go' button. The 'Indexes' tab is shown at the bottom, displaying a single index entry:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	73	A	No	

4.2.1.2 Tables for user:-

The screenshot shows the phpMyAdmin interface for the 'user' table in the 'id5115588_checktaxi' database. The table has 7 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)	latin1_swedish_ci		No	None		AUTO_INCREMENT	Change Drop Primary More
2	name	varchar(225)	latin1_swedish_ci		No	None			Change Drop Primary More
3	national_id	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary More
4	email	varchar(225)	latin1_swedish_ci		No	None			Change Drop Primary More
5	password	varchar(225)	latin1_swedish_ci		No	None			Change Drop Primary More
6	phone	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary More
7	address	varchar(1000)	latin1_swedish_ci		No	None			Change Drop Primary More

Below the table structure, there is an 'Indexes' section showing one primary index:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	6	A	No	

At the bottom, there is a button to 'Create an index on 1 columns'.

(Figure 6):- table for user

4.2.1.3 Table for car and driver:-

databases-auth.000webhost.com / localhost / id5115588_checktaxi / car_and_driver | phpMyAdmin 4.7.7 - Chromium

The screenshot shows the phpMyAdmin interface for a MySQL database named 'id5115588_checktaxi'. The current table is 'car_and_driver'. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	model_car	varchar(225)	latin1_swedish_ci		No	None			Change Drop More
3	chassis_no	varchar(500)	latin1_swedish_ci		No	None			Change Drop More
4	car_id	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
5	car_owner_name	varchar(225)	latin1_swedish_ci		No	None			Change Drop More
6	car_driver_name	varchar(225)	latin1_swedish_ci		No	None			Change Drop More
7	car_driver_national_id	varchar(14)	latin1_swedish_ci		No	None			Change Drop More
8	car_driver_address	varchar(500)	latin1_swedish_ci		No	None			Change Drop More
9	driver_job	varchar(500)	latin1_swedish_ci		Yes	NULL			Change Drop More
10	motor_no	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
11	car_color	varchar(115)	latin1_swedish_ci		No	None			Change Drop More
12	car_check_date	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
13	car_licence_end_date	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
14	driver_city	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
15	driver_photo	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
16	driver_end_licence	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Below the table structure, there are several buttons: Check all, With selected:, Browse, Change, Drop, Primary, Unique, Index, Print, Propose table structure, Move columns, Improve table structure, Console, and tabs for columns and alter driver and licence.

(Figure7):- table for car and driver

Chapter 5

Section 5.1:-

Android interface design

5.1.1 Commands of User Interface Design:

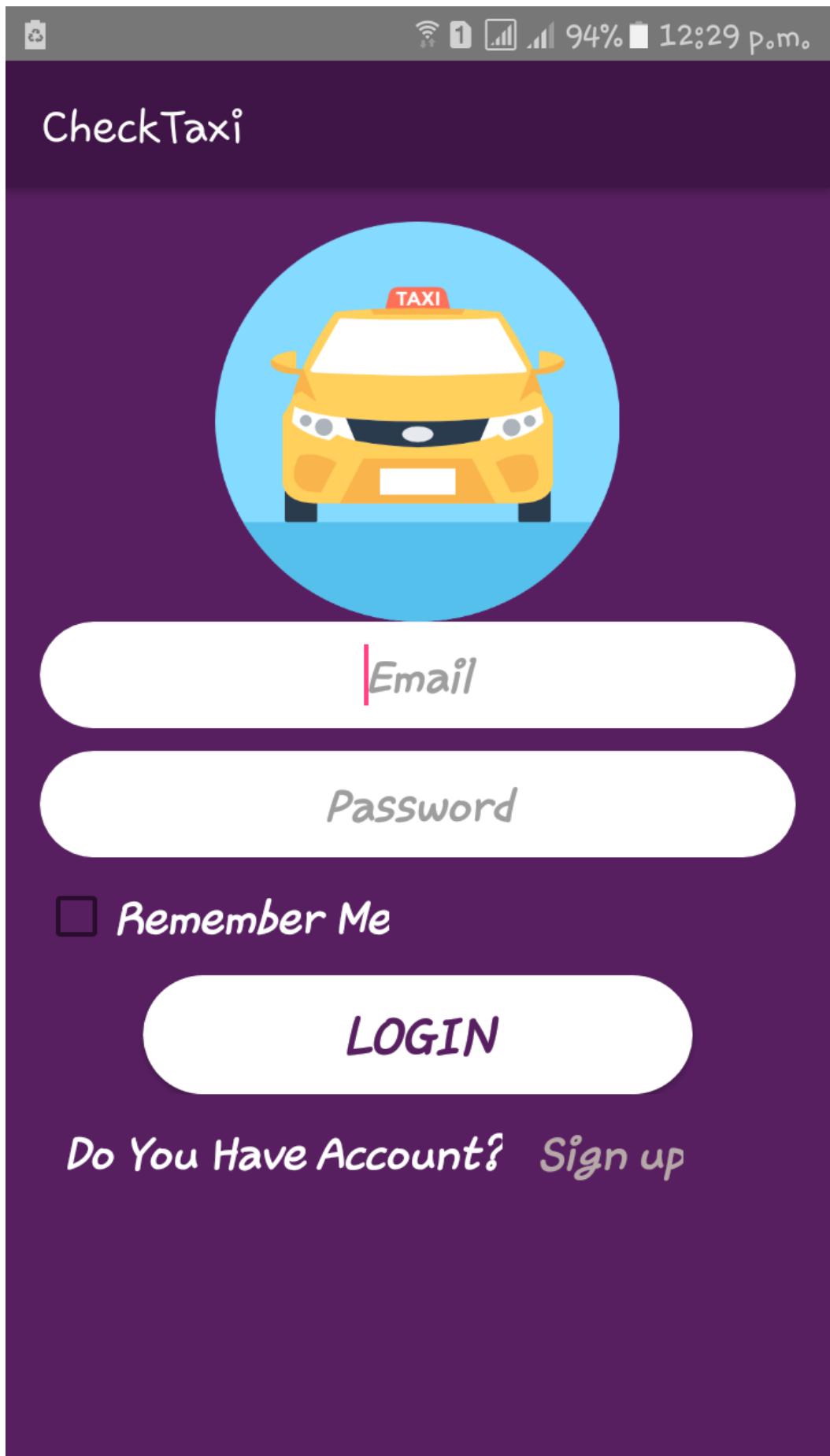
- *Understand your users and their tasks.*
- *Involve the user in interface design.*
- *Test the system on actual users.*
- *Practice iterative design*

5.1.2 android creation :-

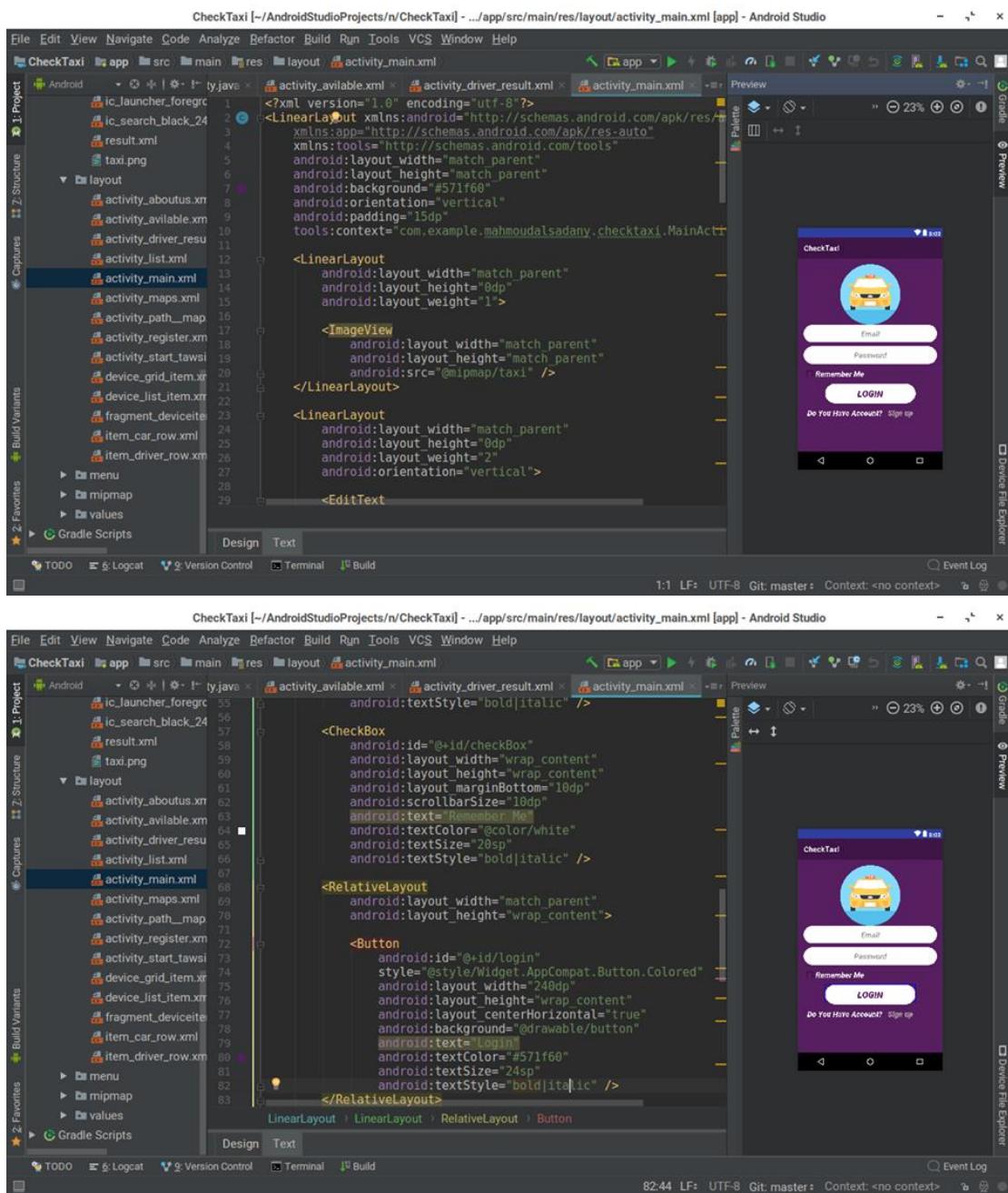
5.1.2.1 log in:

Which the user register on the application by registering email and password and press the button log in.

(Figure 8) log in



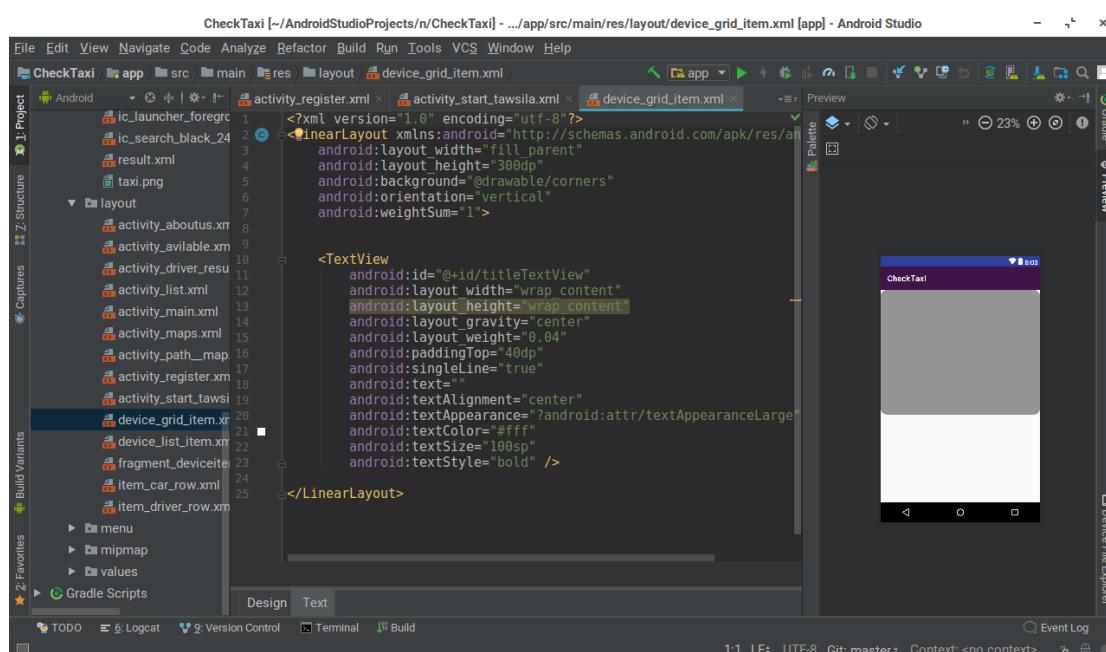
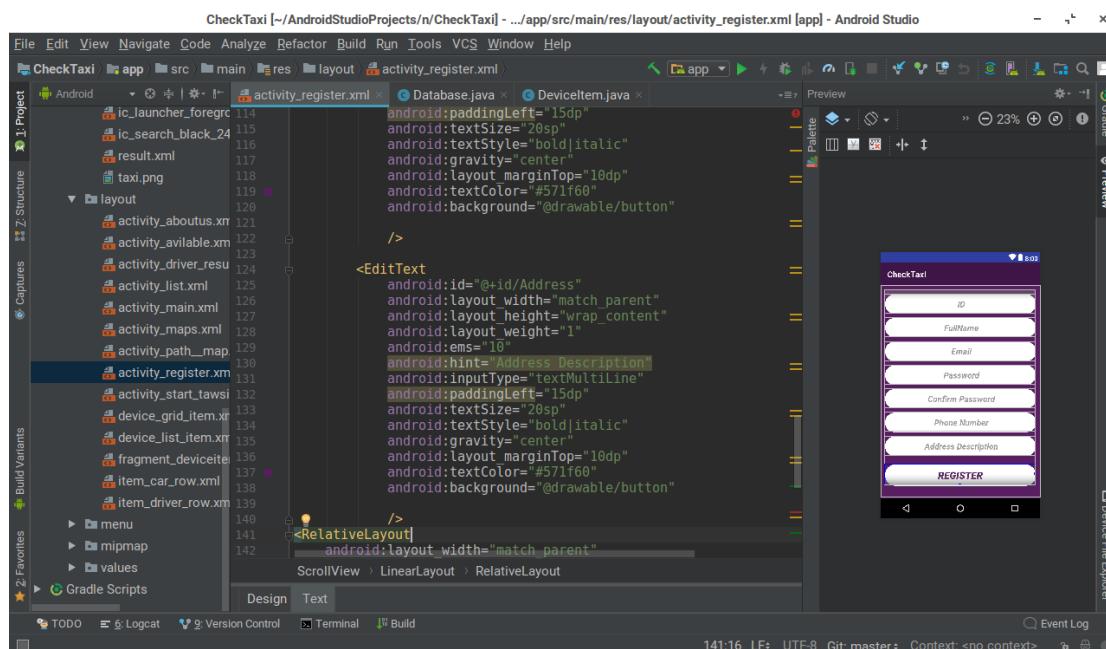
5.1.2.2 Log in page in xml:-

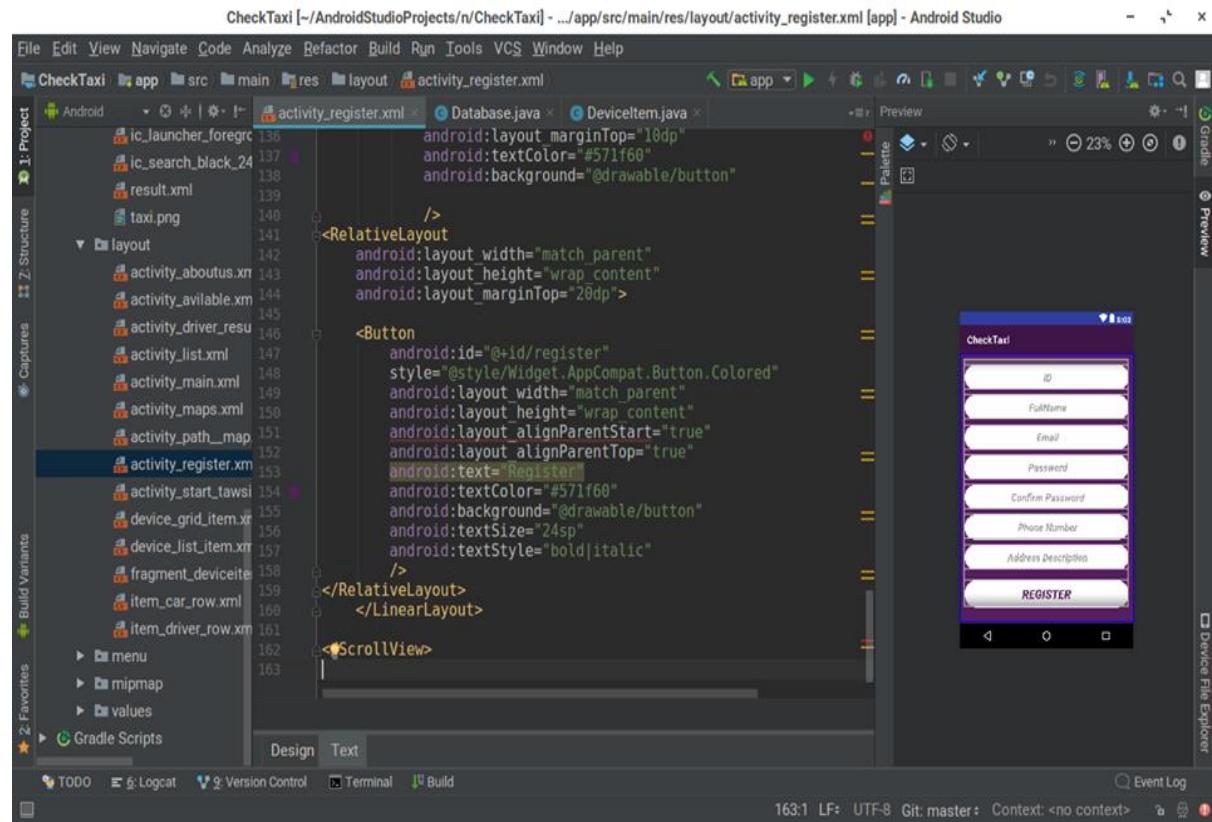


(Figure9):-login in xml.

❖ If the person doesn't have email it can sign up by pressing button of sign up

5.1.2.3 Register:-





(Figure 10):- sign up in xml.



94% 12:29 p.m.

CheckTaxi

ID

FullName

Email

Password

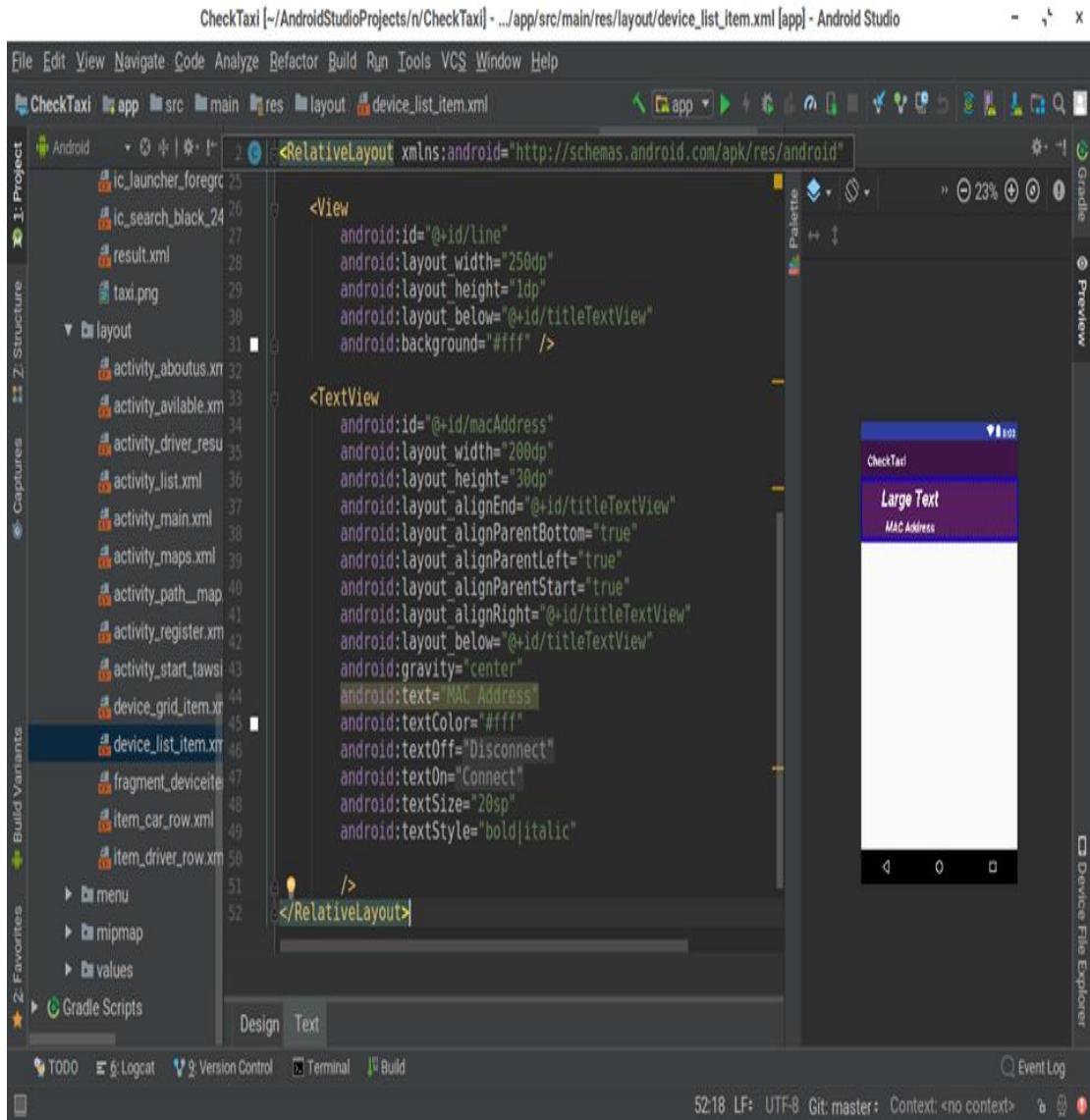
Confirm Password

Phone Number

Address Description

REGISTER

(Figure 11):- sign up.



5.1.2.4 Searching for the car

- ❖ After sign up in the application we search for the car which want to take it through switch on the beluttoth of the phone and connect with the car through beluttoth module.

- ❖ search for the car through the car number.

➤ **BELUTTOH MODULE:-**

*LeaningTech HC-05 Module Bluetooth Serial Pass-Through
Module Wireless Serial Communication with Button for
Arduino.*

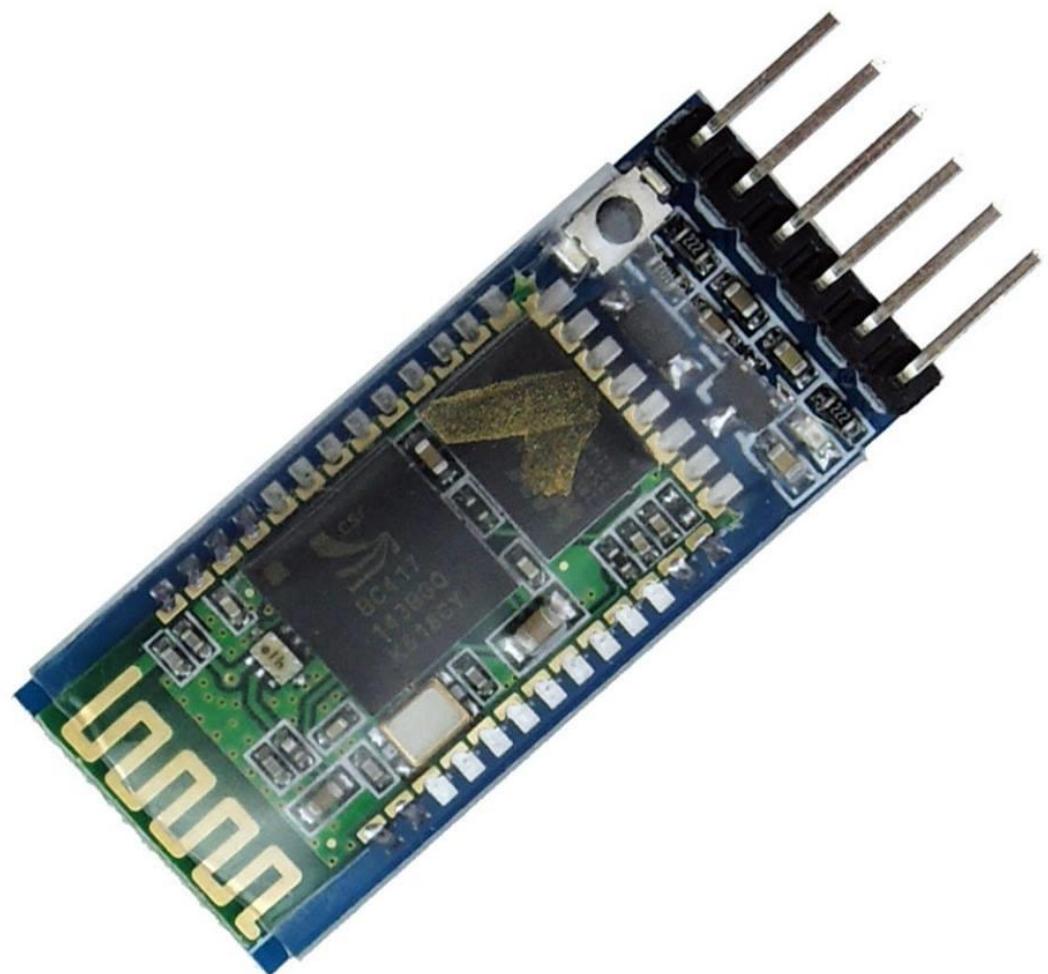
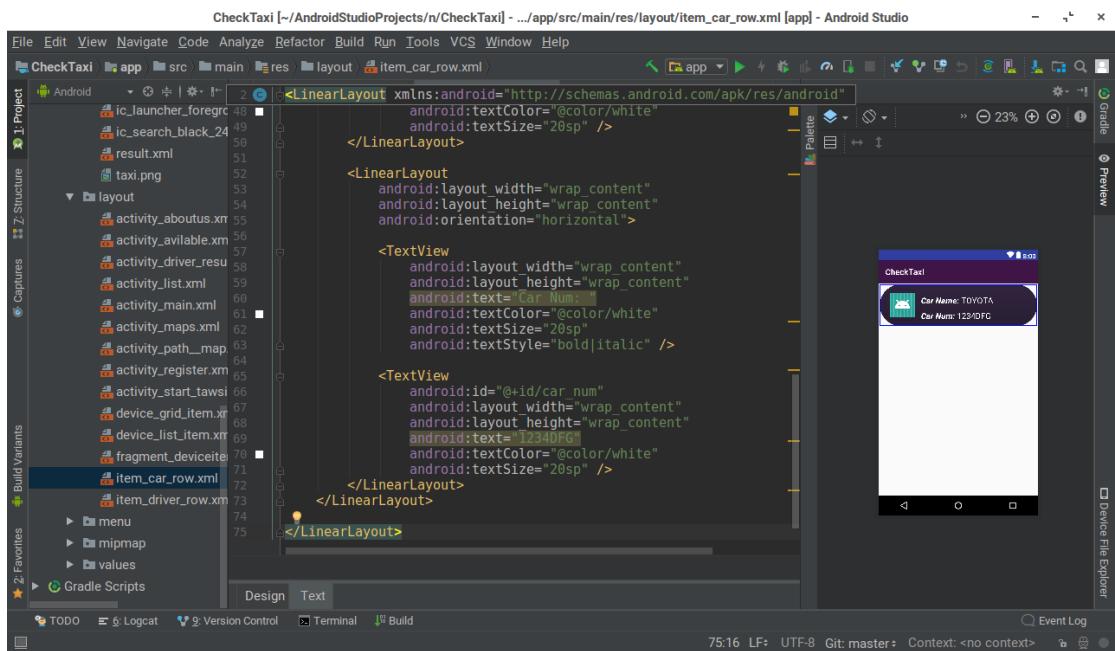
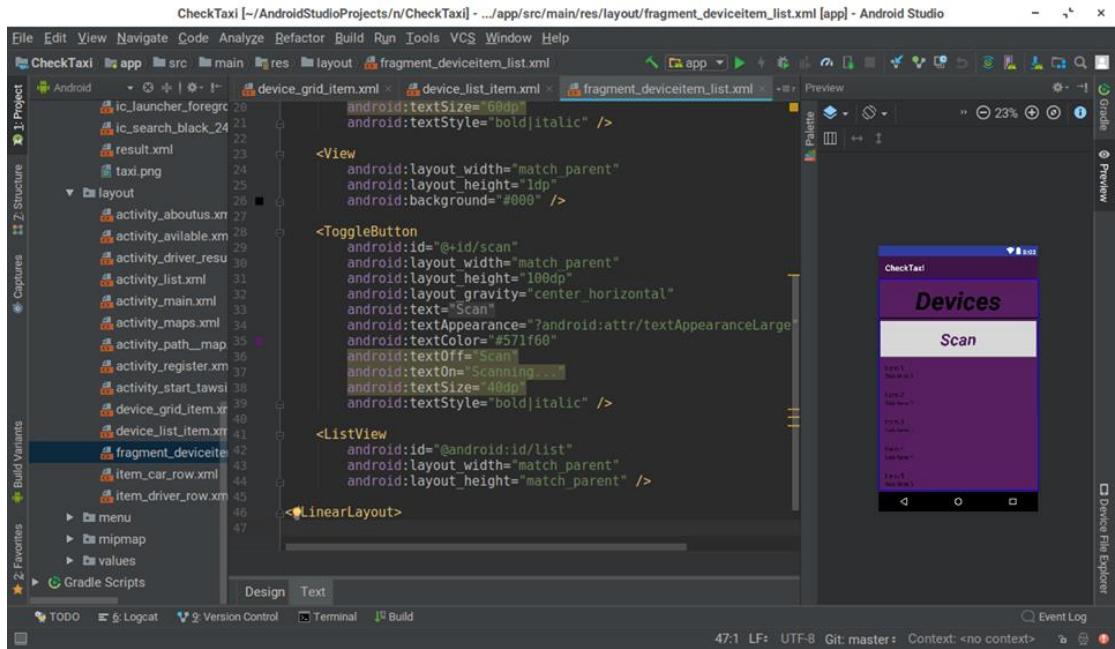
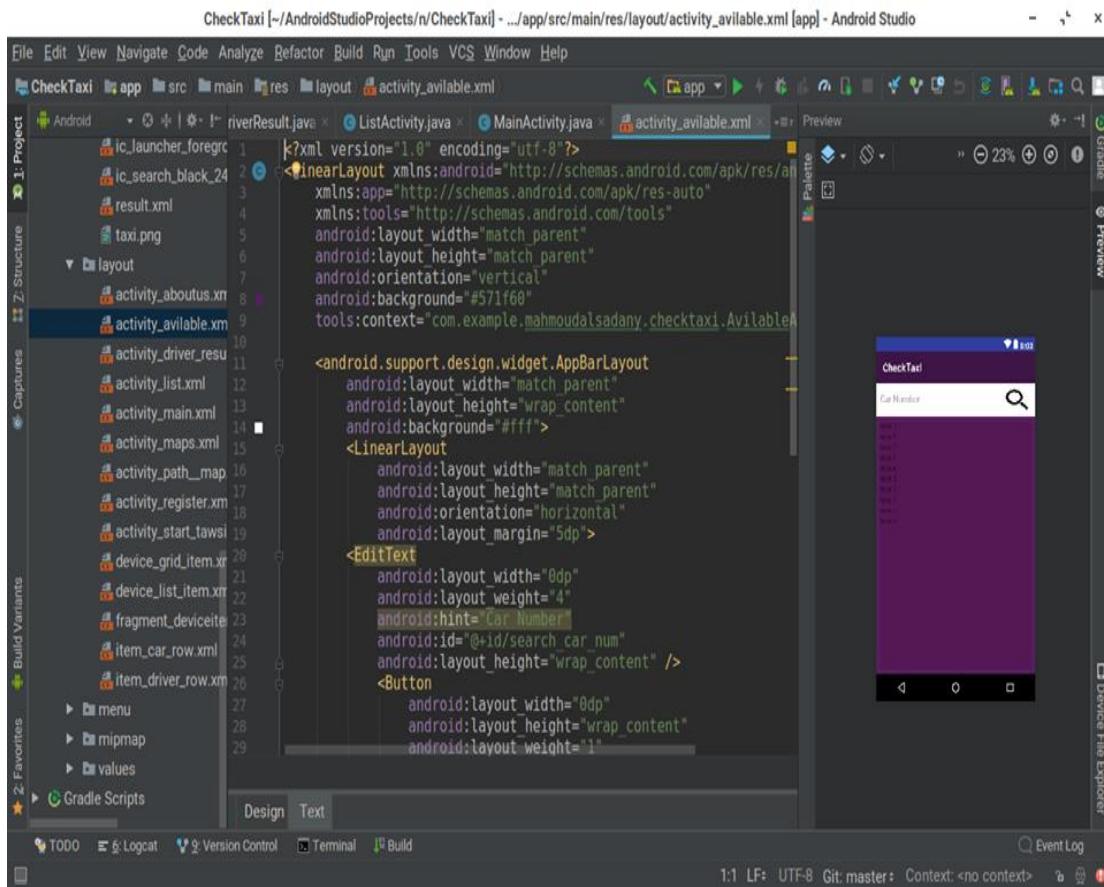


figure12: beluttoh module.





(Figure 12):- searching for the car.

5.1.2.5 choosing the car .

After searching for the car we choose the car and appear the driver_photo,num of car,national-id of the driver if information as the real we press the button of take ride.

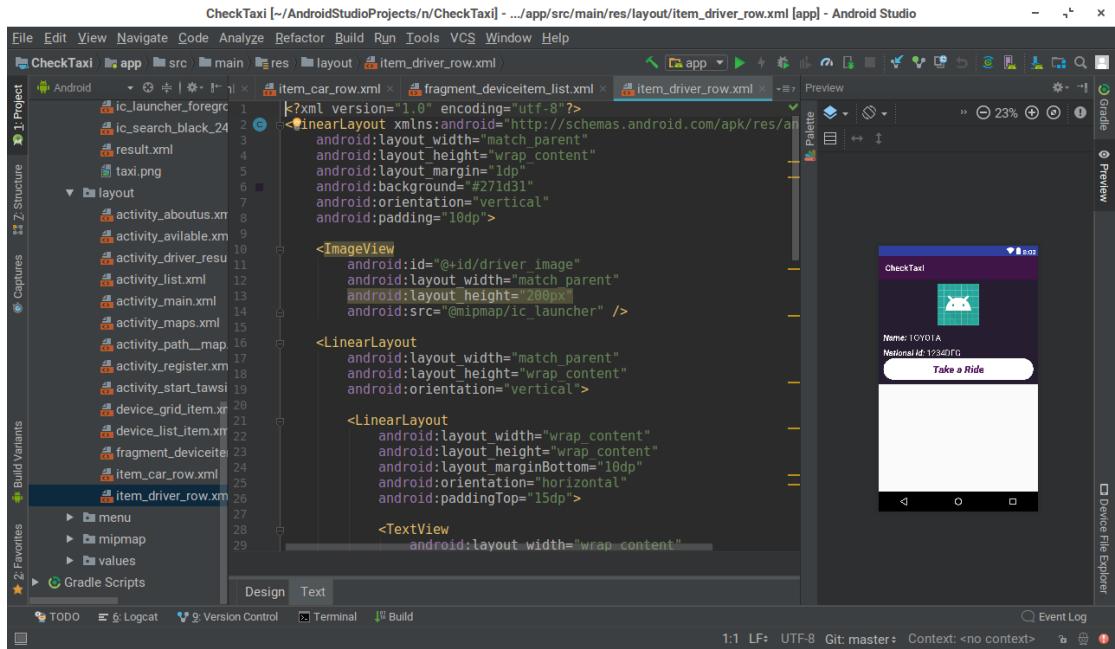
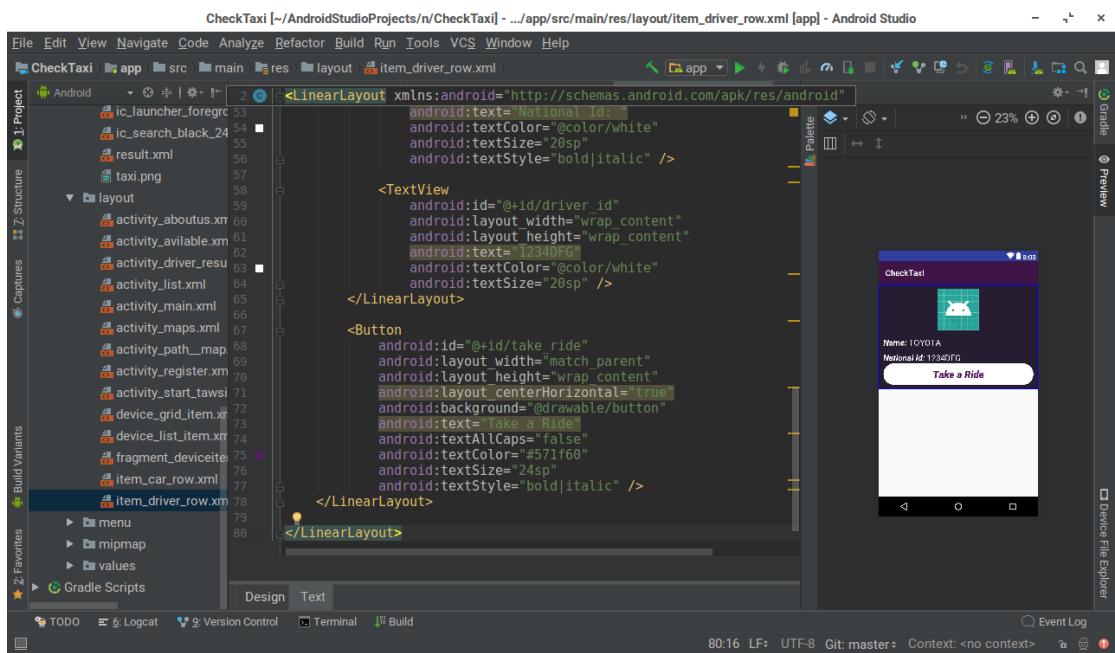


Figure 13:- choosing the car.



5.1.2.6 start

We start the way through determine place which we go and place which ride the car and pressing the button start, when reach the destination we press the button stop and if there is adanger we press

alarm.

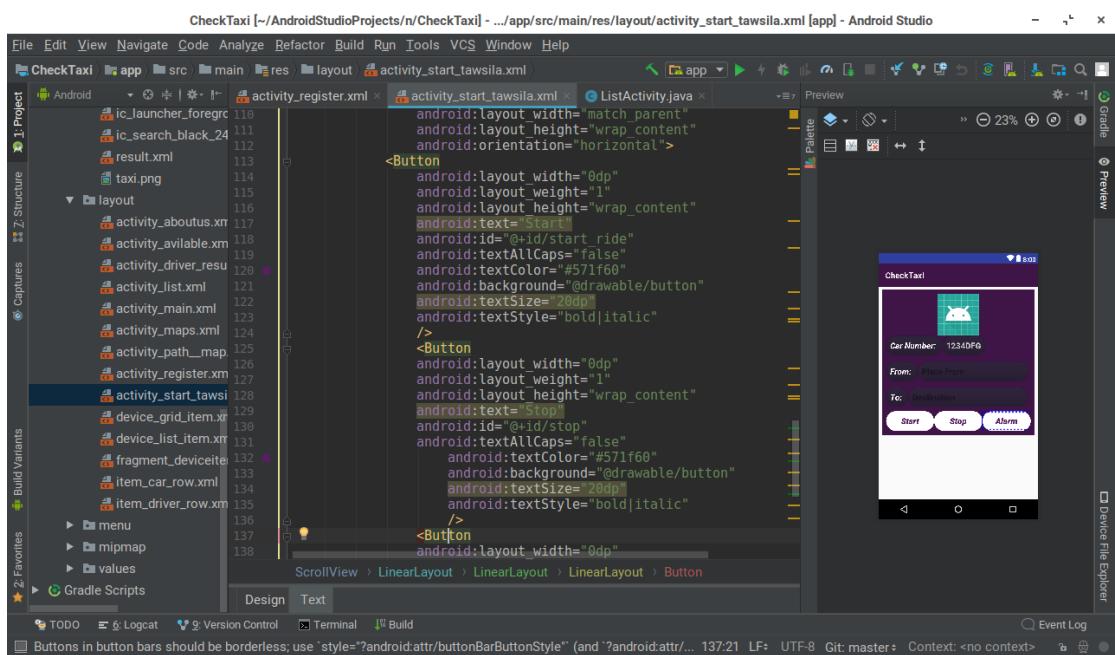
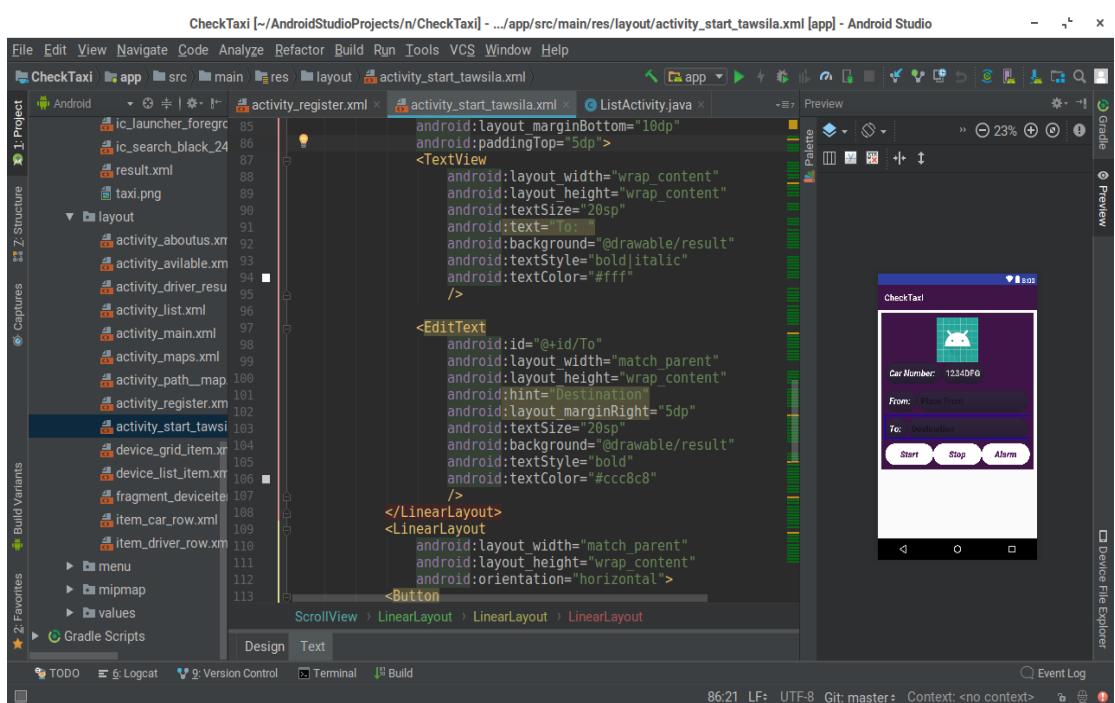
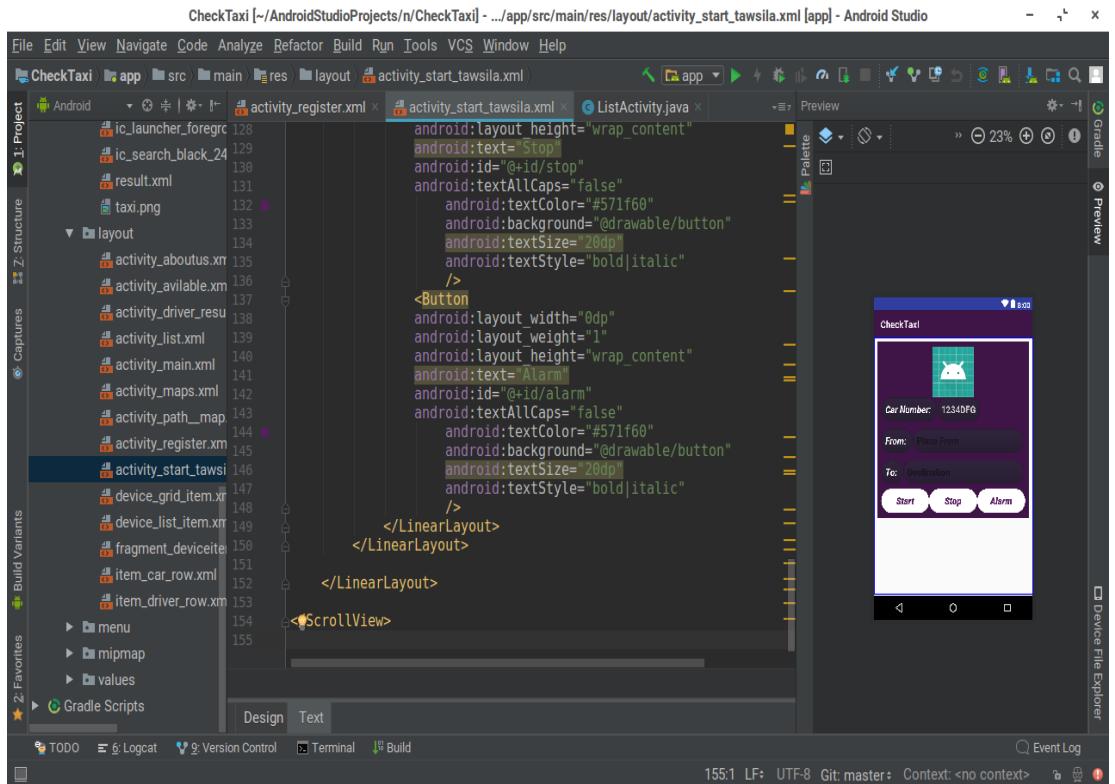
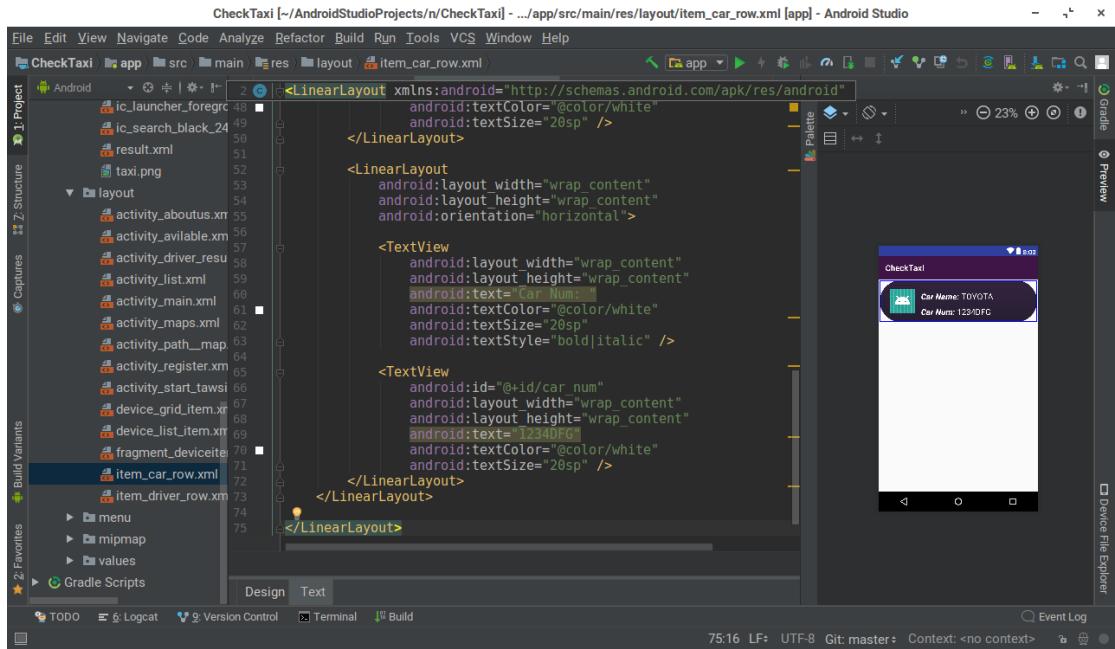


Figure 14:- start

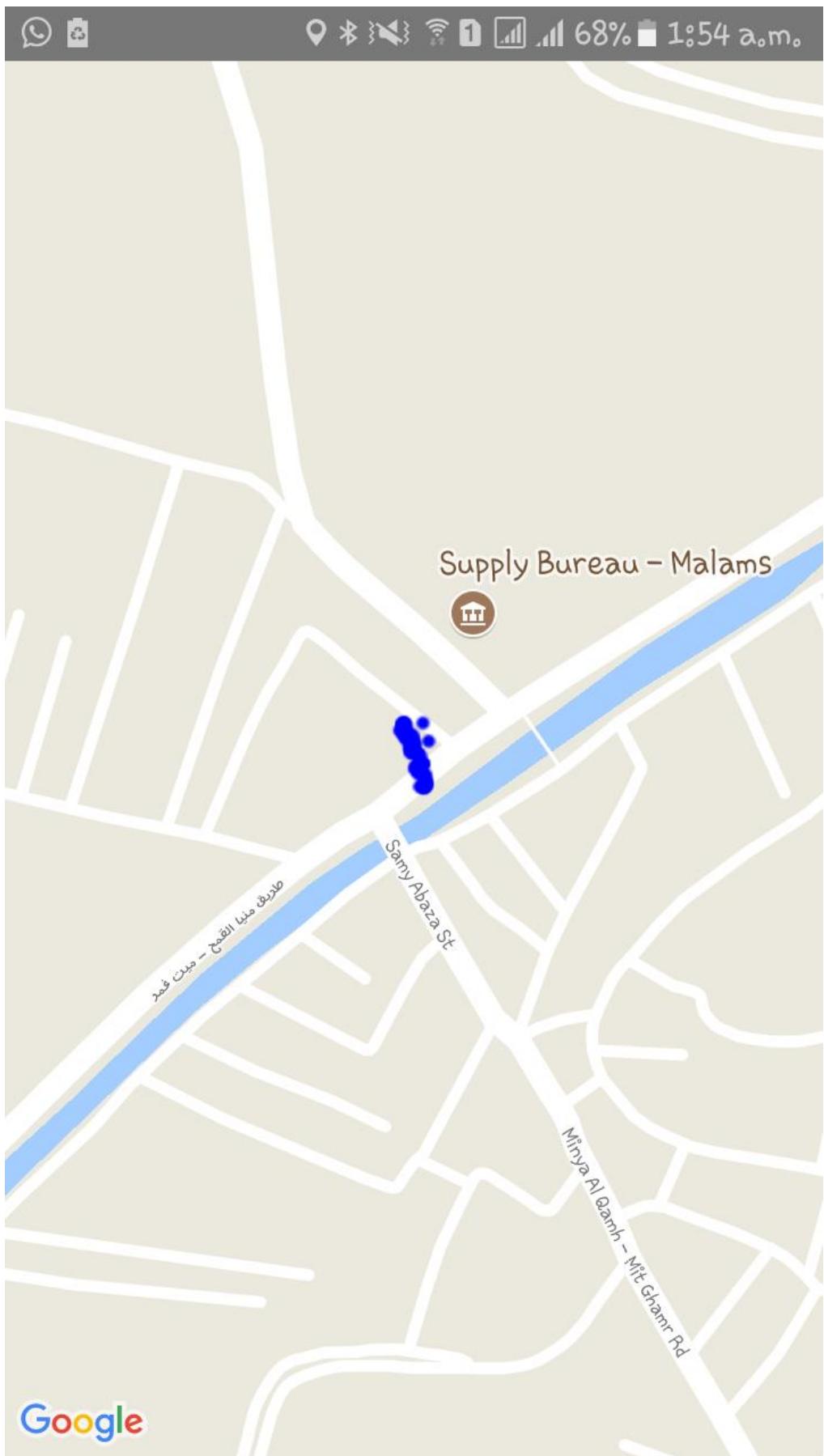


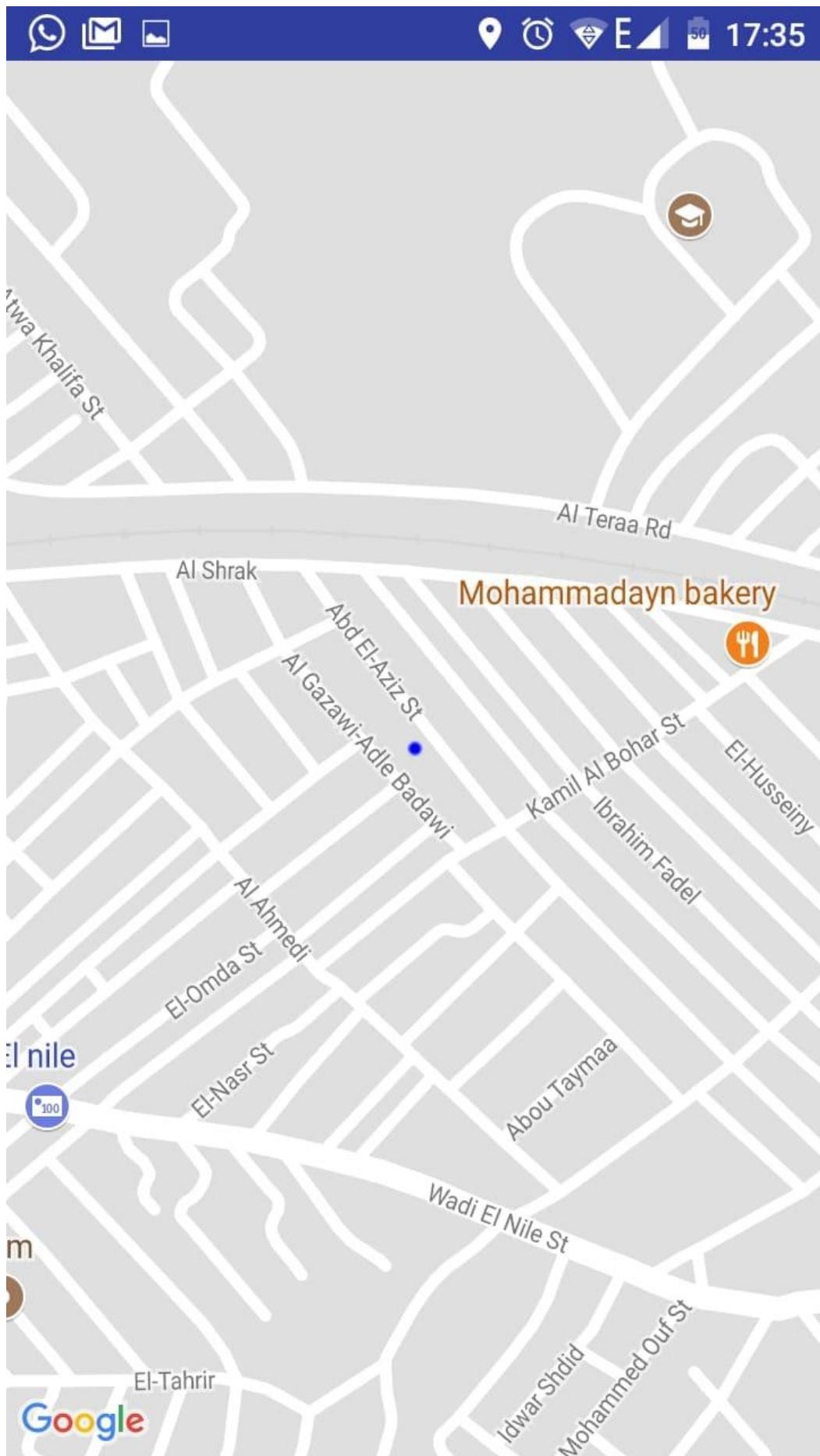


5.1.2.7 Maps:-

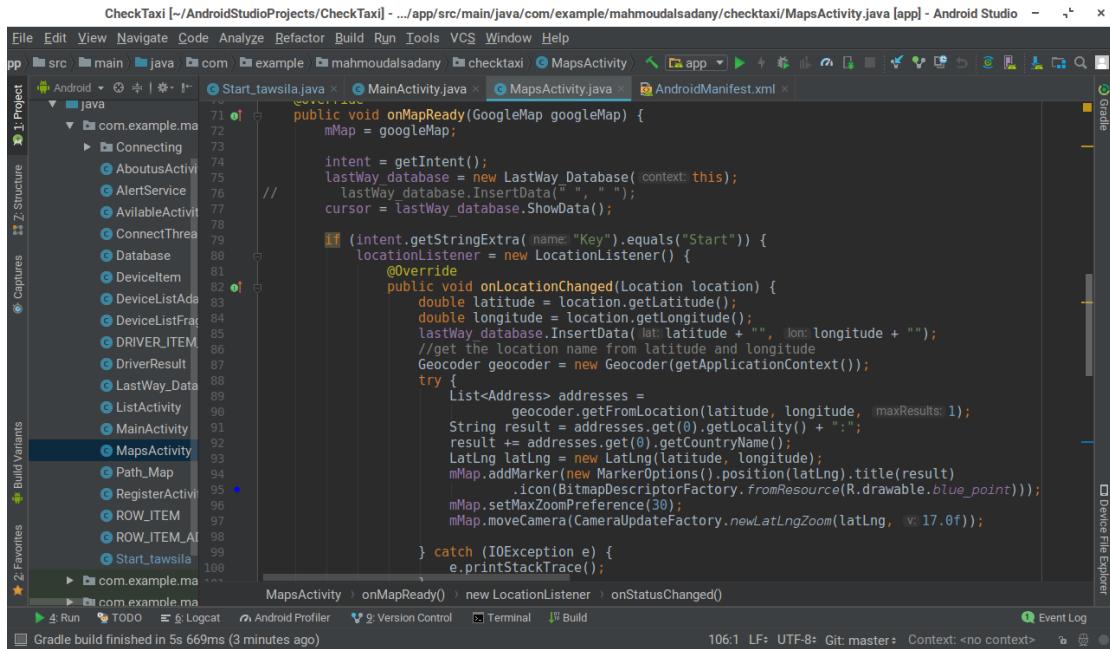
Which determine the places which pass it and register it through longitude and latitude in database and determine the shortest way.

Figure 15:maps.





5.1.2.8 Bluetooth:-



The screenshot shows the Android Studio interface with the project 'CheckTaxi' open. The code editor displays the 'MapsActivity.java' file, which contains Java code for handling Google Maps and location data. The code includes imports for Google Map, LocationListener, and Geocoder. It handles intent filters for 'Start' and 'Key' actions, retrieves location data, and performs geocoding to get addresses from latitude and longitude. The AndroidManifest.xml file is also visible at the top of the editor.

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    intent = getIntent();
    lastWay_database = new LastWay_Database(context: this);
    lastWay_database.InsertData(" ", " ");
    cursor = lastWay_database.ShowData();

    if (intent.getStringExtra(name: "Key").equals("Start")) {
        locationListener = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                double latitude = location.getLatitude();
                double longitude = location.getLongitude();
                lastWay_database.InsertData(lat: latitude + "", lon: longitude + "");
                //get the location name from latitude and longitude
                Geocoder geocoder = new Geocoder(getApplicationContext());
                try {
                    List<Address> addresses =
                        geocoder.getFromLocation(latitude, longitude, maxResults: 1);
                    String result = addresses.get(0).getLocality() + ":";
                    result += addresses.get(0).getCountryName();
                    LatLng latLng = new LatLng(latitude, longitude);
                    mMap.addMarker(new MarkerOptions().position(latLng).title(result)
                        .icon(BitmapDescriptorFactory.fromResource(R.drawable.blue_point)));
                    mMap.setMaxZoomPreference(30);
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, v: 17.0f));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        };
    }
}
```

The screenshot shows the Android Studio interface with the project 'CheckTaxi' open. The left sidebar displays the project structure, showing packages like 'com.example.mahmoudalsadany.check' containing classes such as 'ConnectThread', 'ManageConnectThread', and 'ServerConnectThread'. The right pane shows the code for 'ConnectThread.java'. The code implements a thread for connecting to a Bluetooth device, handling socket creation and connection attempts.

```
package com.example.mahmoudalsadany.checktaxi;
import ...;
/*
 * Created by User on 6/3/2015.
 */
public class ConnectThread extends Thread{
    private final BluetoothDevice btDevice;
    private final BluetoothSocket btSocket;
    try {
        tmp = this.btDevice.createRfcommSocketToServiceRecord(UUID);
    } catch (IOException e) {
        Log.d("CONNECTTHREAD", msg: "Could not start listening for RECOMM");
    }
    btSocket = tmp;
}
public boolean connect() {
    try {
        btSocket.connect();
    } catch(IOException e) {
        Log.d("CONNECTTHREAD", msg: "Could not connect: " + e.toString());
    }
}
```

Figure 16:beluttoh

Chapter 6:

Section 6.1:-

Policeman interface

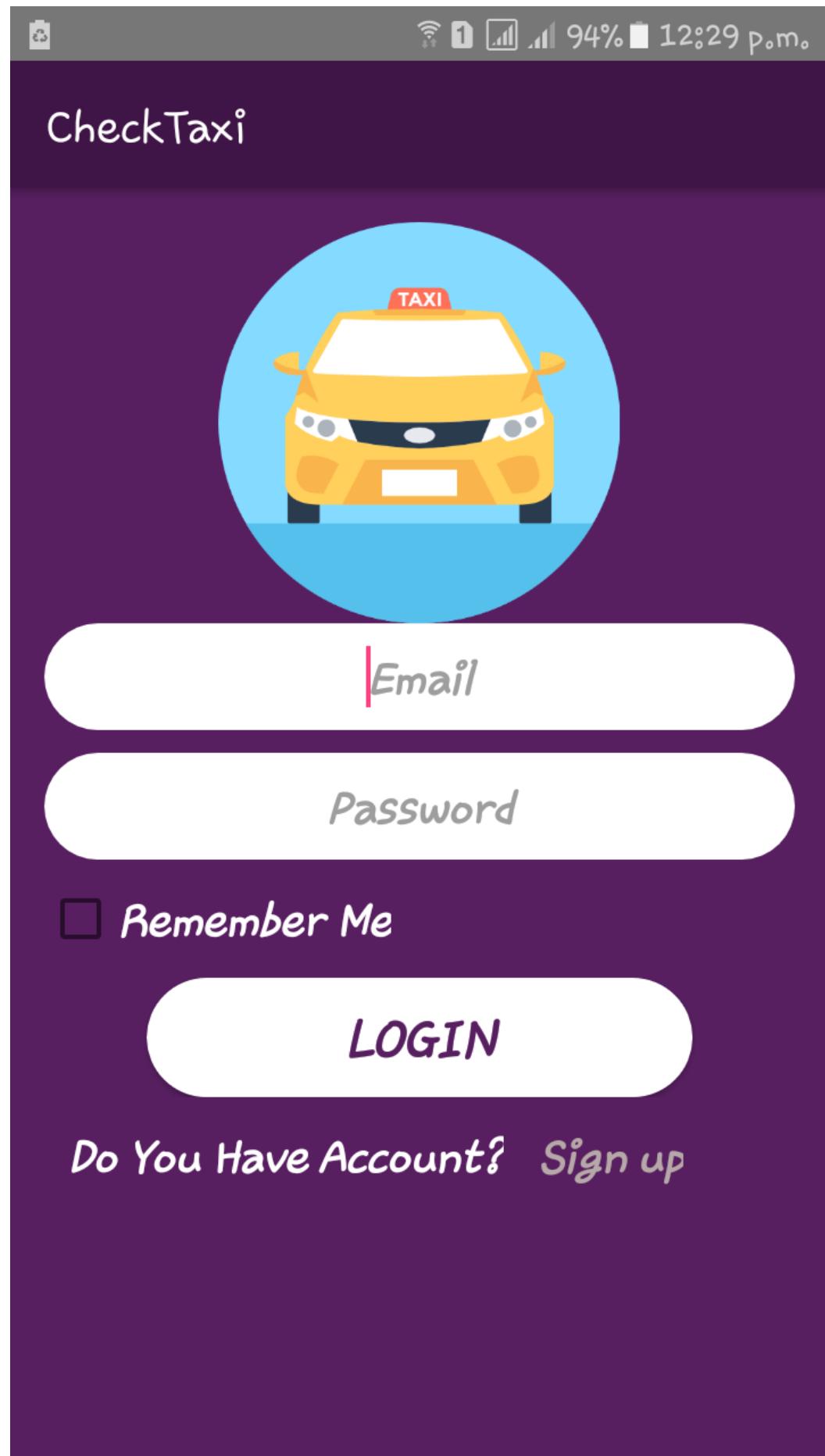
Introduction:-

In this chapter, we will talk in the second part of the project, which is used by police officers in the traffic to know the expiry of licenses for drivers, which provides the policemen time and effort and facilitates the process of traffic in ambush police and roads

6.1.1 log in

- ❖ Policeman login in the application with username and password through this page.
- ❖ if the police man dosen't have email ,he must sign up and register in the database.

(Figure 17) log in



6.1.2 searching for the car:-

- Police man search for the car by car_number to explore the expiry of licenses for drivers.

OR

- Turn on the beluttoth for the mobile of policeman and connect with beluttoth module in the car for appearing the car_number ,car name and car_color.



76%

2:02 p.m.

PoliceMan

Search about user

Car Number

Scan Via Bluetooth

Logout

Car Name: Daewoo

Car Num: 7541rqt

Car Name: Daewoo

Car Num: 7541rqt

Car Name: Hover

Car Num: 6041aqq

Car Name: lansir

Car Num: 5432bjy

Car Name: nissan

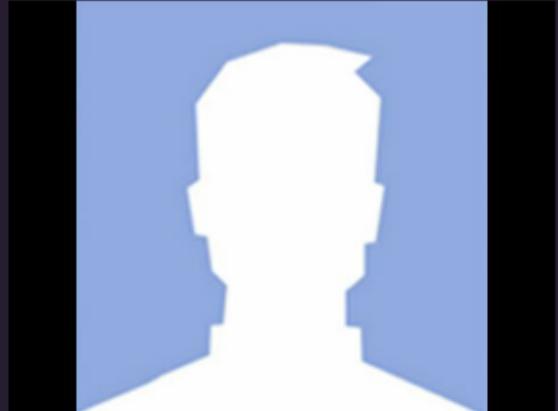
6.1.3 After finding the car:-

It appear for police man information about the car and driver like:

Name , national _id, driver_end licences,driver picture

Car end licences, car_check data through these things he explore the expiry of the licence of driver.

PoliceMan



Name: **** * * * * * * * * * * * *

National Id: * * * * * * * * * * * *

Driver End Linces: 1-1-2019

Car End Linces: 2019

Car check Data: 1-1-2020

OK

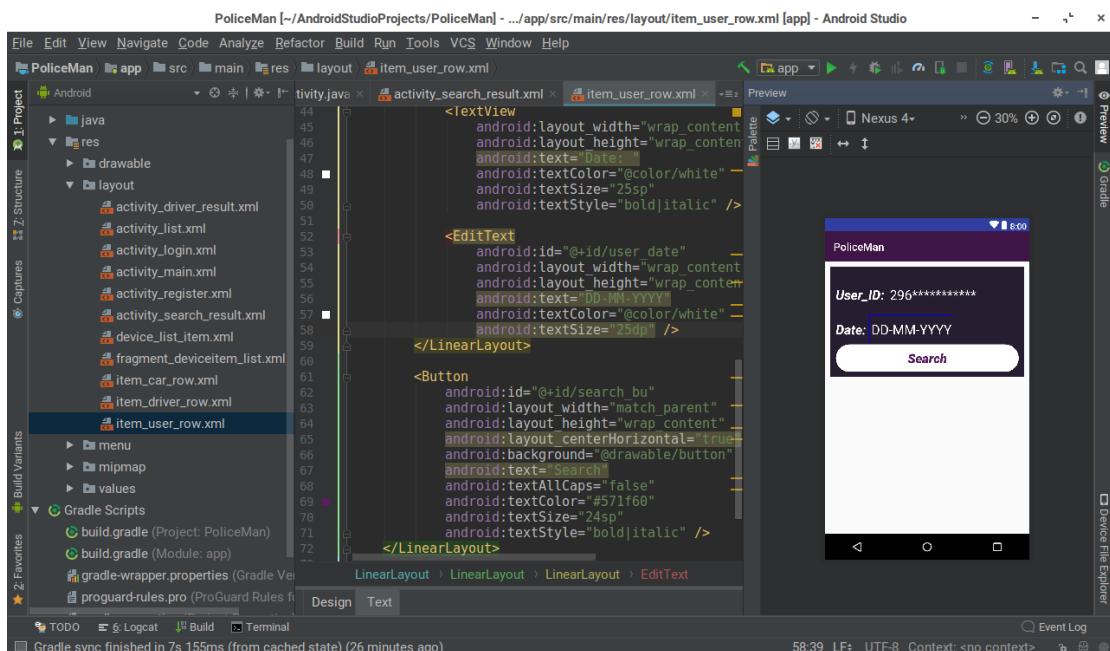
Section 6.2:

6.2.1 Search for user:

If any person in danger and dosen't send message for end the destination :-

- After 10seconds it sends notification that for suring finishing the destination .
- If doesn't finish and appear any danger and any person report we take from person

User_id for this person and the date and then search for this person.



6.2.2 Maps in the application of police man:-

After the policeman enter the user_id and date it determines the path of the car which it drive through this we can determine the place of the person and there is problem it doesn't store the path in the server it can drop the server so that we can store it in the data base.

The screenshot shows the phpMyAdmin interface for the database 'id5115588_checktaxi'. The left sidebar lists databases like 'New', 'car_and_driver', 'tawsila', 'trip_pass', and 'user'. The 'trip_pass' table is selected in the main area. The 'Structure' tab is active, displaying the following table definition:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	<code>int(11)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>		<code>AUTO_INCREMENT</code>	<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>
2	<code>trip_id</code>	<code>int(6)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>			<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>
3	<code>lang</code>	<code>varchar(45)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>			<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>
4	<code>lat</code>	<code>varchar(45)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>			<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>
5	<code>user_id</code>	<code>int(14)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>			<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>
6	<code>user_date</code>	<code>varchar(100)</code>	<code>utf8_unicode_ci</code>		No	<code>None</code>			<code>Change</code> <code>Drop</code> <code>Primary</code> <code>Unique</code> <code>More</code>

Below the table structure, there is an 'Indexes' section with one index defined:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<code>Edit</code> <code>Drop</code>	<code>PRIMARY</code>	<code>BTREE</code>	Yes	No	<code>id</code>	0	<code>A</code>	No	

At the bottom, there is a button to 'Create an index on 1 columns'.

(Figure 20):- trip pass.