

**Introduction:** In our country, there are a lot of transports on the road but we don't know which transport will go where and which road it will be used and where the transport right at this moment. Again if we want to go someplace but we don't know which road will be shortest for our journey? So trying to solve this type of problem, I build an android app called "***Transport Tracking App***". In this system, every transport must be needed to register under a transport company and those company will be registered to the govt. (Admin). There will be some fixed transport stoppage and transport must stop at that stoppage only. Furthermore, this ***Transport Tracking App*** is enough user-friendly app so anyone can easily use this app and make their life easier.

### **Technologies Used:**

#### **Language:**

- a) Java
- b) Android
- c) PHP
- d) MySQL

#### **Compiler:**

- a) Android Studio
- b) Xampp

#### **Others:**

- a) Google Map API
- b) Smart Phone (Virtual Device)

### **Hardware:**

- a) Processor: Intel(R) Core(TM) i3-6100U CPU @2.30GHz 2.30 GHz
- b) RAM: 4.00GB

**Methodologies:** In this System, there will be 4 sections and this section are given below:

- a) Admin
- b) Transport Company
- c) Driver
- d) User

Relationship between these sections are:

- a) Admin will provide the transport company account.
- b) When the transport company adds the driver account, this account must be needed to approve by the admin otherwise the driver can't use their account on these apps.

**Sign in page:** To use this app everyone must need to sign in (Figure 1) and if the user doesn't have any account they can create an account by clicking on the signup option (Figure 6). When we sign in, the apps will decide our category (Admin, Transport Company, Driver, and User) and it will show the corresponding activity page to us.

So when the user clicks on the sign-in button, the apps immediately call the ***SignInActivity.java*** with sending two parameters user mail address and password. In ***SignInActivity.java*** there are two URL.

```
String reg_url = "http://192.168.42.149/register.php";
```

```
String login_url = "http://192.168.42.149/login.php";
```

Since I used my laptop as my database server so here the **IP Address** is my laptop own IP Address and register.php, login.php this type of file are kept in the **Xampp -> htdocs** folder. So when the network will change, the IP address must need to be changed. In this ***SignInActivity.java*** the user mail address and password are send to the PHP file and match to the database file if the user is valid ***SignInActivity.java*** will send to the next page according to the user category otherwise it will show a message and return to the Sign-In page.

**Admin section:** In this activity generally, the admin location, all the transport stoppage, and the connected stoppage road will be shown (Figure 2). The admin can select other activity from the menu bar option (Figure 3). When the admin selects an option it will call the ***Search\_Database.java*** to search the database.

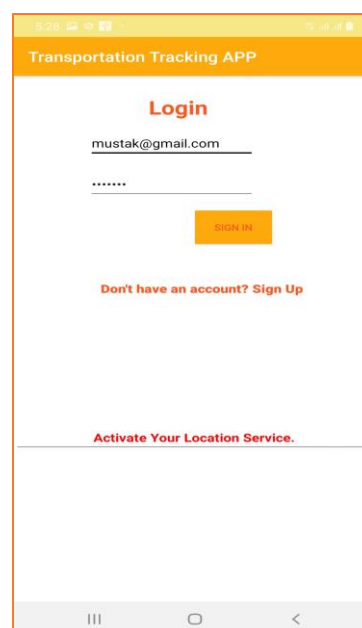


Figure 1: Sign In Page

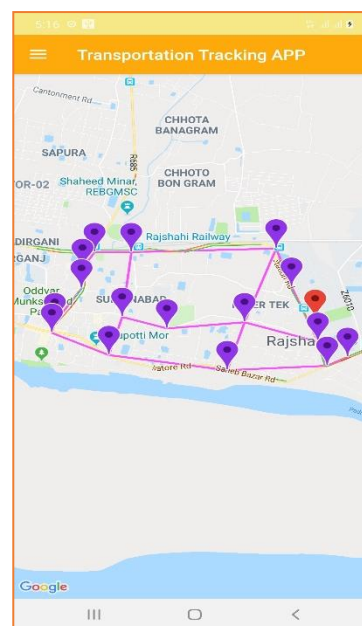


Figure 2: Admin Main Activity

```
Search_Database search_database = new Search_Database(this);
search_database.execute(type,"company");
```

After searching the databases the desired result will put as **extras** to the **Intent** and transferred to the desired activity.

```
Intent intent = new Intent(context,
AdminActivity.class);
```

```
intent.putExtra("result", result);
```

```
intent.putExtra("id", id);
```

```
context.startActivity(intent);
```

And the desired activity will receive the intent extras such as

```
Intent intent=getIntent();
```

```
user_id=intent.getStringExtra("id");
```

```
result=intent.getStringExtra("result");
```

Menu bar options-

- In the Profile activity, the admin can see his information only (Figure 4).
- In the Others Admin activity, the admin can see the others admin info and add the other admin to this system by providing proper info (Figure 5, Figure 6). In this activity, the table is **dynamic** which shows the admins information.
- In the transport company activity, the transport company information is shown in a dynamic table (Figure 7) and admin can add the new admin by clicking on the new admin button and this will show register activity (Figure 6).
- In the Driver Info. / Approval activity, the admin can see the driver info in a dynamic table and at the top of the right corner of the page, the number of the driver approval request will be shown. If the number of the requested driver approval is more than one a **red circle** will be shown with the number otherwise it will show a **green circle** and the number will be 0 (Figure 8). So when the admin click on the number the driver approval activity page will be shown and when the admin activates the approval button the driver will be added and right at the moment if the admin inactivates the approval button the driver approval will be negative (Figure 9).
- In the Stoppage location, the admin can see the stoppage name, location on this page (Figure 10). When new location button is clicked location insert will page will open and

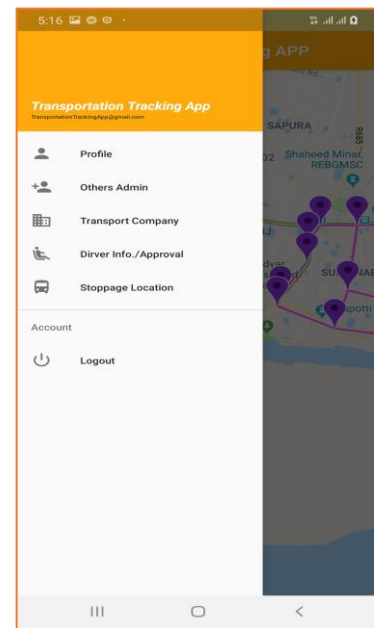


Figure 3: Admin Menu Bar



Figure 4: Profile

the admin must give the location name, latitude, longitude, division and connected stoppage name (Figure 11).

Name	Mail Address	Phone No.
jakir	jakhos97@gmail.com	01521334719
Imam Hossain	imambuet420@gmail.com	01552929499

Figure 5: Others Admin

Figure 6: Register

Figure 11: Location Insert

Company Name	Company Mail	Address	Company Phone
sagor	sagor@gmail.com	01521330713	
Asif	asif@gmail.com	121212121	
adib	adib@gmail.com	2121212121	

Figure 7: Transport Company

Company Name	Driver Name	Driver Mail	Driver Phone	Driver NID
sagor	Asif	Asif@gmail.com	01682485219	01010101010
sagor	Shob	shob@gmail.com	01521334718	23456786543
Asif	kazi	kazi@gmail.com	01521334718	23232323
sagor	bulbul	bulbul@gmail.com	0121212120	123444444
sagor	apu	apu@gmail.com	0211314150	456444444

Figure 8: Driver Info.

Place Name	Latitude	Longitude
mugda	23.7295	90.4285
komlapur	23.7285	90.4254
arambag	23.7315	90.421
motijil	23.7266	90.4215
polton	23.7302	90.4121
kakrail	23.7378	90.4091
malibag	23.7438	90.4143
rajarBag	23.7401	90.42
bashabo	23.74	90.4276
buddho mondir	23.7356	90.4287
ruet	24.3626	88.629
talaimari	24.3616	88.627
hadir mor	24.3611	88.6172
zero point	24.3653	88.6
sagor para bottola	24.3671	88.6069
sadur mor	24.3665	88.6189
tika para	24.3658	88.6113
northern mor	24.3643	88.6261
filling station	24.3705	88.6235
bodra	24.3748	88.622
railway station	24.3744	88.6078
raigate	24.3744	88.6042
new market	24.3726	88.603
aolokar mor	24.3703	88.6029
gonokpara	24.3665	88.6003
sagor para	24.3628	88.6056

Figure 10: Stoppage Location

Company Name	Driver Name	Driver Mail	Driver Phone	Driver NID	Approval
sagor	abir	abir@gmail.com	015000000	0192837465	<input type="checkbox"/>
sagor	zahid	zahid@gmail.com	012882737	0120804836768792	<input type="checkbox"/>
sagor	tutul	tutul@gmail.com	12901239023	01208048367687924	<input type="checkbox"/>
sagor	onik	onik@gmail.com	2342342342	45654767867453	<input type="checkbox"/>

Figure 9: Driver Approval

**Transport Company:** The sign in page will transfer to the **Transport\_Company.java** activity if the account is valid. In this page generally, the transport information will be shown under the company account (Figure 12). To activate the transport, the company need to click on the status **inactive** option and it will show an **alert dialog box** where the company needs to assign the driver id number. Again when the company clicks on the status **active** option the transport driver id will be empty and the status will be changed to **inactive**.



Transport Name	Transport License	Driver ID	Status
Hanif	abcd	1	Active
Desh Travels	acbde		Inactive
Akota	mnop		Inactive
Hanif	bbse	5	Active

Figure 12: Transport Company

There is 4 button at the top off the table. To add a new driver the company will need to click on the new driver button and give the proper information. The driver information will be shown on the driver info activity (Figure 13) and the transport company profile will be showed after clicked on the profile button (Figure 4). To sign-out from the account, the sign-out option needs to be clicked.



Name	Email	Phone	NID	Approval
Arif	Arif@gmail.com	01682485219	0101010101010	Yes
shoib	mnzshoib@gmail.com	01521334705	23456786543	Yes
bulbul	bulbul@gmail.com	0121212120	1234444444	Yes
apu	apu@gmail.com	0211314150	4564444444	Yes
abir	abir@gmail.com	015000000	0192837465	No
zahid	zahid@gmail.com	012882737	0120804836768792	No
tutul	tutul@gmail.com	12901239023	01208048367687924	No
onik	onik@gmail.com	2342342342	45654767867453	No

Figure 13: Driver Information

**Driver:** In this driver activity, when the driver sign-in the apps will start to update the driver location in the database. I create a function called **requestLocation ()**, in this function when the driver position is changed to 1 meter then 1 sec later it will call the **onLocationChanged()** function to update the location on the database.

```
locationManager.requestLocationUpdates (provider, 1000, 1, this);
```

There is two more option in this activity, one profile and the other is the sign-out button. On clicking the profile button the driver profile will be shown (Figure 4) and the sign-out button will sign-out the driver to his account.

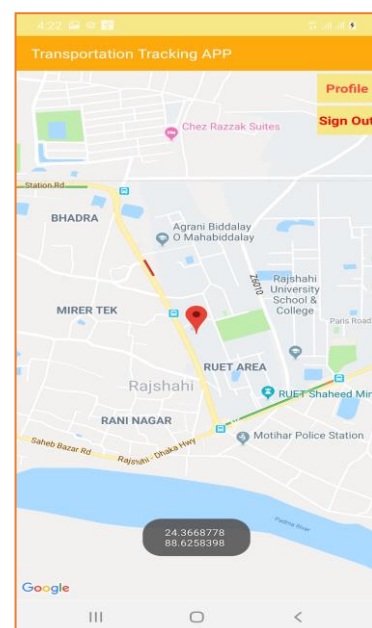


Figure 14: Driver

**User Section:** When the user sign-in to these apps, the UserActivity.java will be shown which is just like the admin activity (Figure 2 and Figure 3). In the menu bar option, there will be 5 options-

- a) Profile
- b) User Search Location
- c) User Select Transport
- d) Transport Info
- e) Sign out

These five activities description are given below:

- In the profile activity, the user profile will be shown (Figure 4).
- In the User\_Search\_Location.java activity, the user can search the best shortest road from a source stoppage to the destination stoppage. When the user searches the road the apps will show the shortest road on the map as **red color** and write the total distance and shortest road transport stoppage on the map (Figure 15). To implement this I write the **Dijkstra's Shortest Path Algorithm** in User\_Search\_Location.java activity and create a **Node type class**

```
class Node{
    int start=0;
    int stop=0;
    double cost=0;
}
```

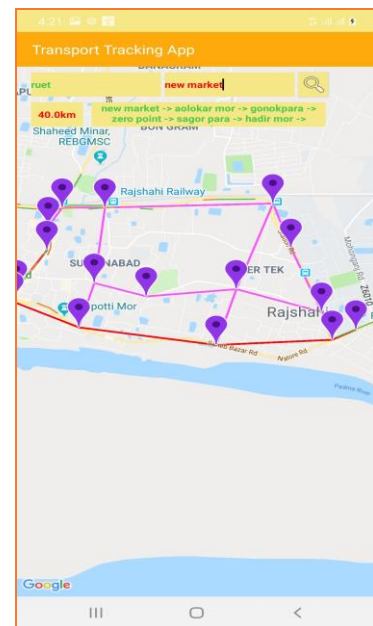


Figure 15: User Search Location

To store the connected node an **ArrayList<Node>[] adj\_node** is used which is **Node type** and to store the shortest distance from the source stoppage to all the stoppage I used **distance ArrayList** and to store the parent stoppage of every stoppage from source to a destination I used **parent ArrayList**. So when the user clicks on the search button all this ArrayList will be cleared and store the data according to the Dijkstra's Algorithm and then call the **map fragment** to draw the shortest road on the map. After calling the map fragment the map will be cleared and redraw the road on the map.

- In the User Select Transport option, the user can select a transport name and according to this transport name, all the transport location in that name will be shown on the map if any transport on that name is active (Figure 16).
- In the Transport Info option, all the transport traveling stoppage information will be shown according to their name in a dynamic table. So the user can easily know which transport will go in which stoppage (Figure 17).



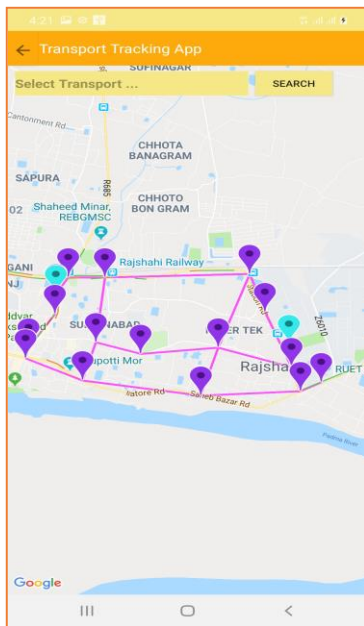


Figure 16: User Select Transport



Figure 17: Transport Info.

**Database:** In the database, I create a database called tta and in this database, there are 5 tables and these tables are almost redundancy-free.

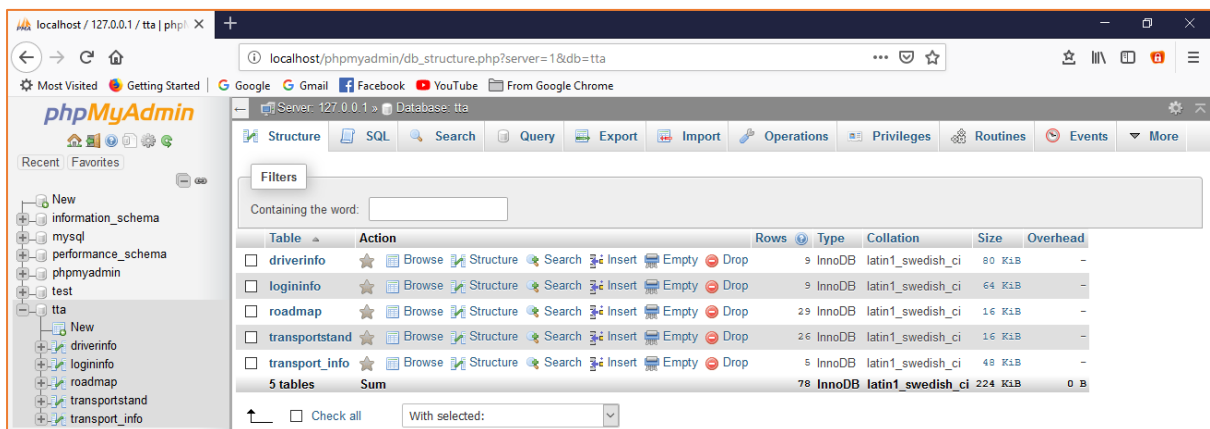


Figure 18: Database

To connect to this database we need to keep our PHP page in the **Xampp -> htdocs folder** and write the IP Address in five java files. This five java files are-

- SignInActivity.java (reg\_url, login\_url)
- Database\_insert.java (search\_url)
- Database\_update.java (search\_url)
- Search\_Database.java (search\_url)
- TStandBackgroud.java (login\_url)

**PHP Page:** There is six PHP page, we just need to keep this page on the Xampp -> htdocs folder. The database java file will connect to this page and pass the data between them. This PHP page will connect to the database and perform the query properly and return the result to the Java page.

**Future work:** In the future, I will try to add more facility like

1. Try to update the algorithm to find the shortest path more quickly
2. Try to make the system more secure
3. Try to provide the user with the best transport suggestion according to their location to travel with lower cost and distance

**Conclusion:** By using this app, all the transport may be controlled under a system and may reduce the transport problem in our country. As my apps are user-friendly, so anyone can easily use this app and I will try to make this app more user-friendly and add more facility to the apps and secured.