CAB HAILING SOFTWARE

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

HARSH PRATAP SINGH (3122 22 5001 038)
KRISHNA VARUN R (3122 22 5001 064)
KISSHOR M (3122 22 5001 062)

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam – 603110

July 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled "**Cab Hailing Software**" is the bonafide work of "Harsh Pratap Singh(3122225001038), Krishna varun R(3122225001064) and Kisshor M (3122225001062)" who carried out the project work in the UCS2201 – Fundamentals and Practice of Software Development during the academic year 2022-23.

Internal Examiner External Examiner

Date: 18.07.2023

TABLE OF CONTENTS

Abstract :

The cab hailing is a software application designed to connect passengers with available drivers in real-time. The objective of this project is to provide a convenient and efficient way for passengers to request rides and for drivers to accept and complete those requests. The system is built using C programming language and utilizes various data structures and algorithms to manage ride requests, driver availability, and route optimization. The project also includes features such as fare estimation, payment processing, and driver ratings to ensure a seamless and satisfactory experience for both passengers and drivers. Overall, the cab hailing project in C aims to revolutionize the transportation industry by leveraging technology to improve accessibility, reliability, and safety.

PROBLEM STATEMENT

We need to develop a software system that efficiently assigns cabsto customers based on their location and requests.

- The system should calculate fares for customers considering a fixed base fare, distance-based fare, surge fees during peak demand,advance booking fees, and cancellation fees for ride cancellations.
- The objective is to create a seamless and transparent experiencefor both customers and cab drivers while ensuring accurate and fairpricing.

EXPLORATION AND ANALYSIS OF THE PROBLEM

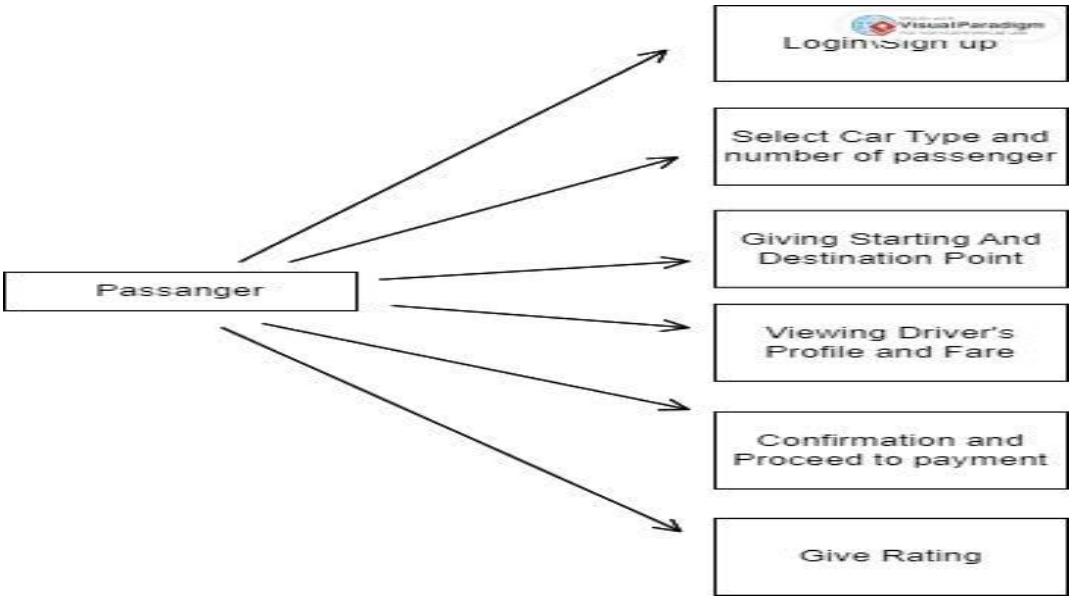
We did thorough, in-depth research on how existing cab hailing software's like "Uber", "Ola", "Rapido" work and tried to implementideas from them.

- We also explored on the following:
    1. Calculation of distance between any 2 given locations.
    2. Pricing and Fare structure.
- We created level 0,1 data flow diagrams and structure charts tofurther analyze our problem.

DIAGRAMS:

USE CASE DIAGRAMS:

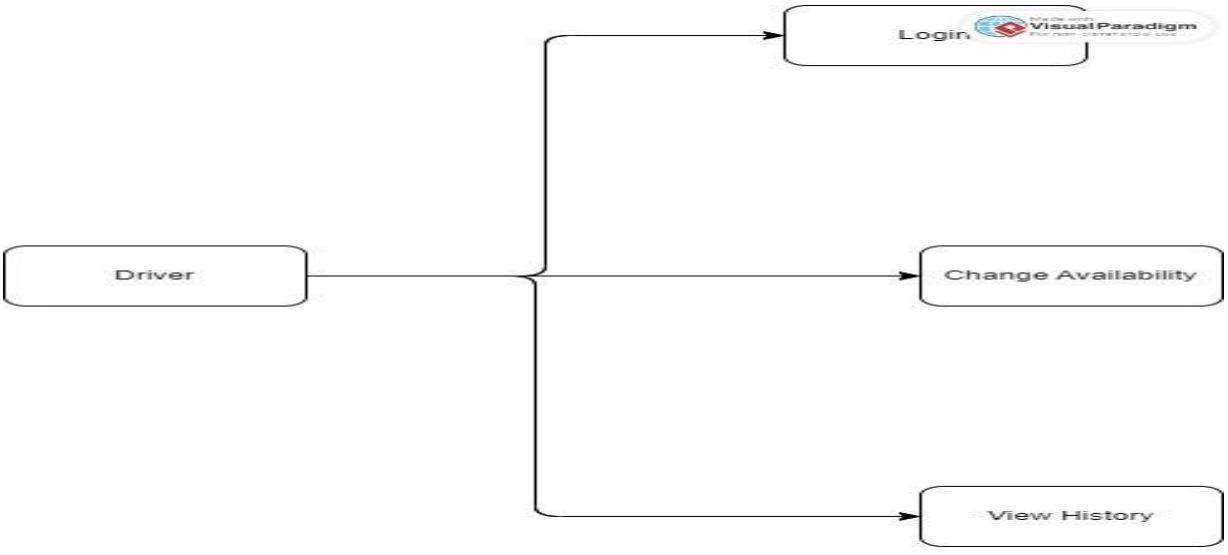
PASSENGER MODULE:



The Diagram Shows the Functions which a passenger can use in the Passenger module.

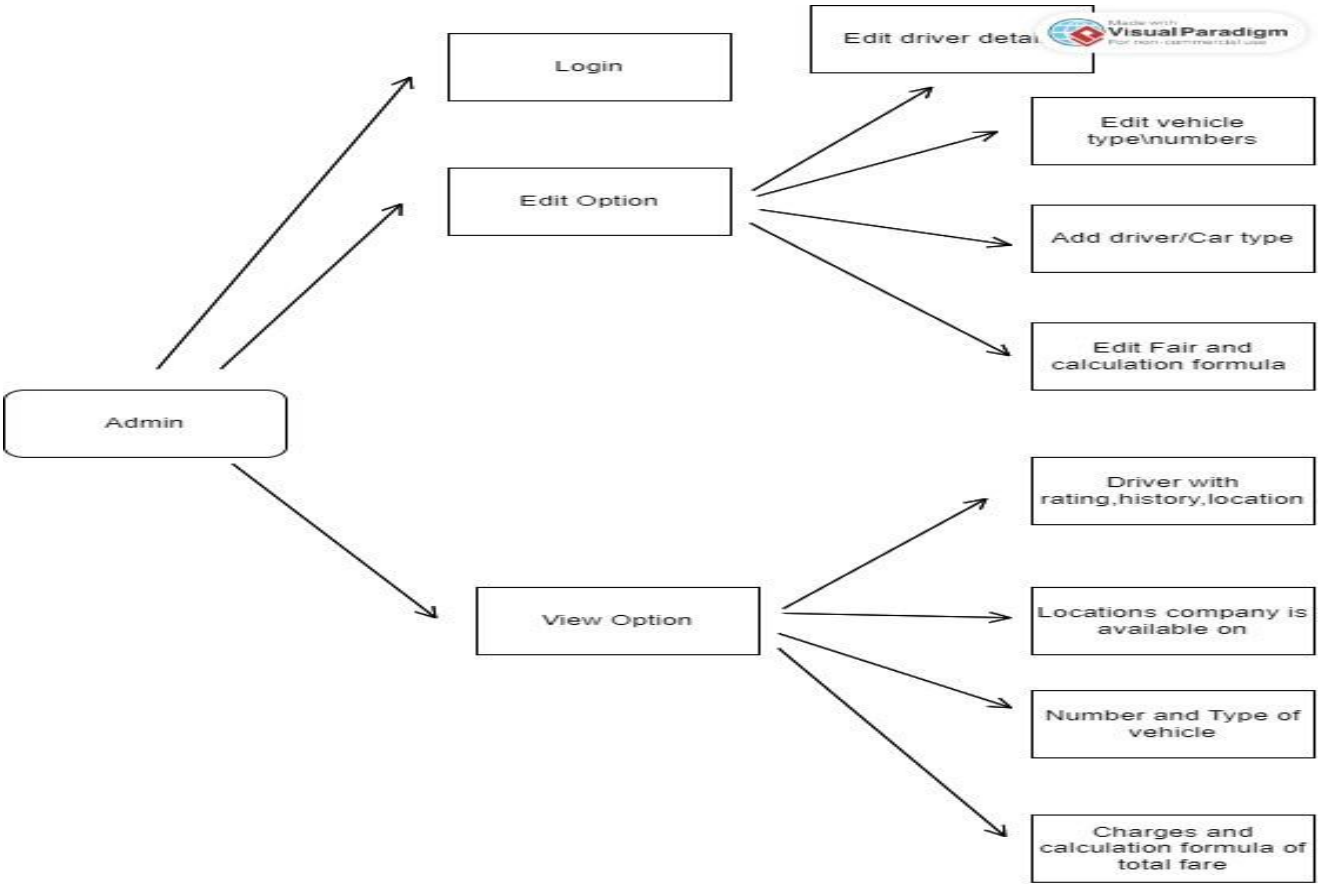
The Module helps the user to easily Know about the functionalities of the program. The Passenger After login or Signup Can give his location and destination and can choose the Kind of Car that passenger wants to book. And after That the driver assigned will appear with the history and with the fare details where passenger can either proceed or cancel the ride. If he has accepted the ride, then he will have to give the rating after the ride is over.

DRIVER :



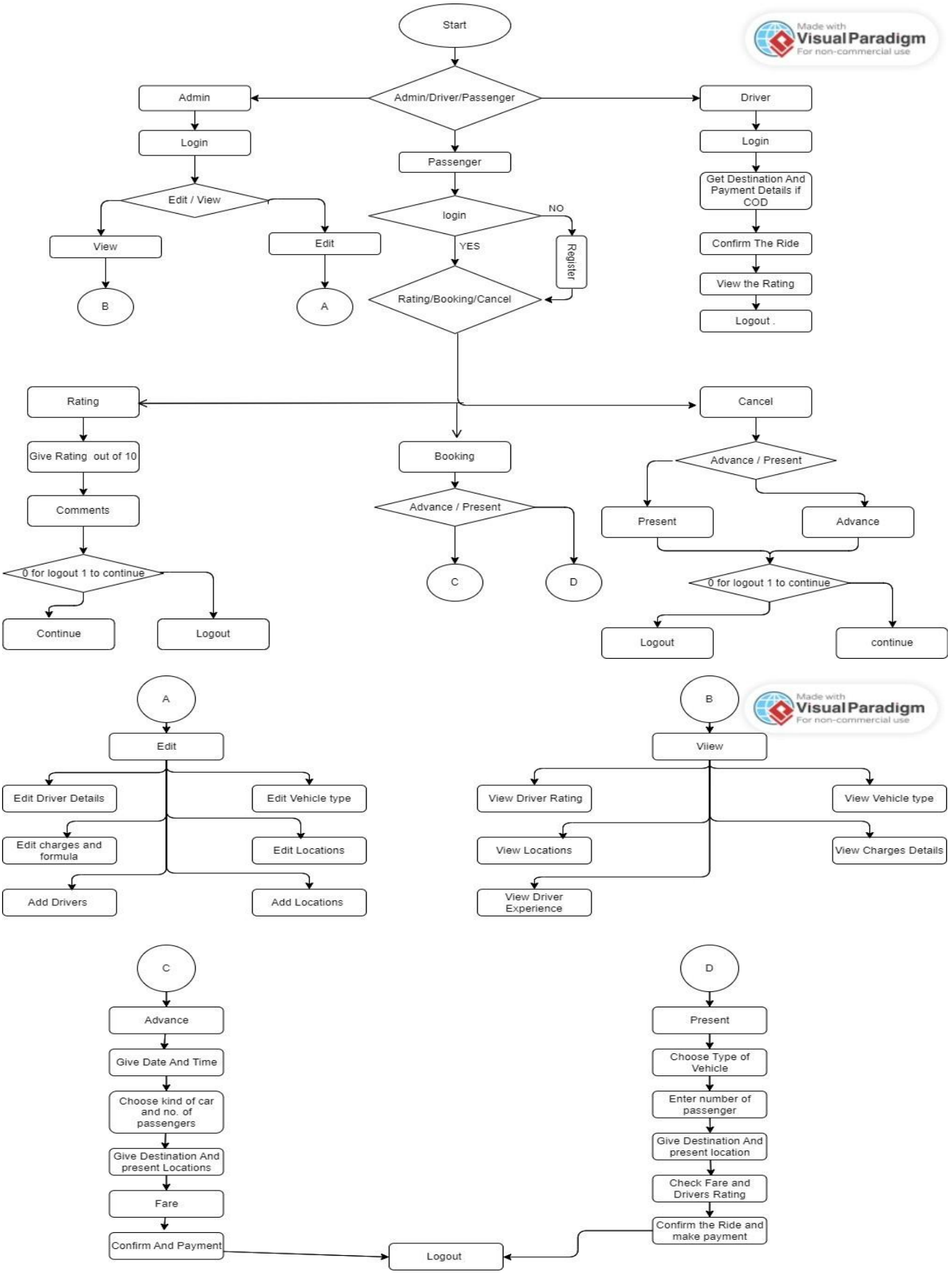
The Diagram explains the functionalities of the driver module. The driver will first have to login and then he will have two options either to view history or to change his availability if he wants holiday after connecting to the admin.

ADMIN:



The diagram shows the functionalities of the admin module. The Admin will have to login and then will functions to either edit or view the different datasets like viewing or editing the locations ,vehicle types , fares of different type etc.

ACTIVITY DIAGRAM:


Start
Admin/Driver/Passenger
Admin
Login
Edit / View
View
B
Edit
A
Passenger
login
NO
YES
Register
Rating/Booking/Cancel
Driver
Login
Get Destination And Payment Details if COD
Confirm The Ride
View the Rating
Logout .
Rating
Give Rating out of 10
Comments
0 for logout 1 to continue
Continue
Logout
Booking
Advance / Present
C
D
Cancel
Advance / Present
Present
Advance
0 for logout 1 to continue
Logout
continue
A
Edit
Edit Driver Details
Edit Vehicle type
Edit charges and formula
Edit Locations
Add Drivers
Add Locations
B
Viiew
View Driver Rating
View Vehicle type
View Locations
View Charges Details
View Driver Experience
C
Advance
Give Date And Time
Choose kind of car and no. of passengers
Give Destination And present Locations
Fare
Confirm And Payment
D
Present
Choose Type of Vehicle
Enter number of passenger
Give Destination And present location
Check Fare and Drivers Rating
Confirm the Ride and make payment
Logout

The class diagram here, represents the entire working of the applications. Firstly, the user will have the choice to enter as the Admin, Driver or passenger.

Case 1:

If the user enters as the admin he will have to login and then Modify or view the dataset. As explained in the use case diagram.

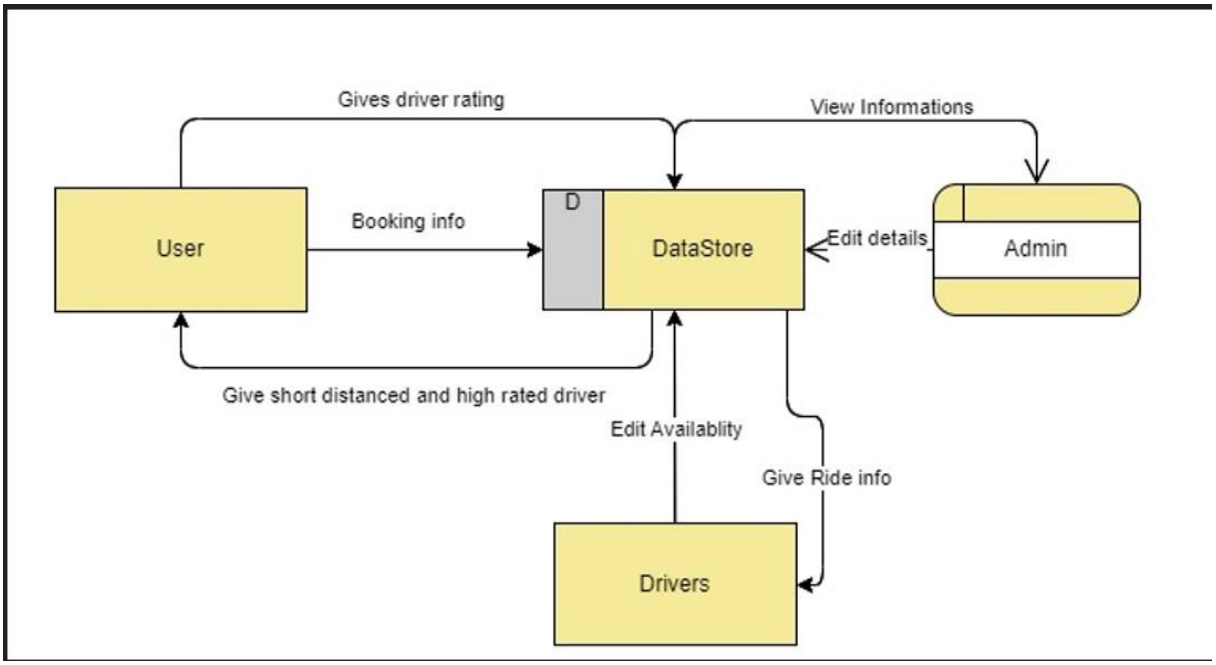
Case 2:

If the user enters as the passenger, he will be able to book the car, booking for advance, will be able to rent a car, cancel the present or the advance booking, View his details along with his history.

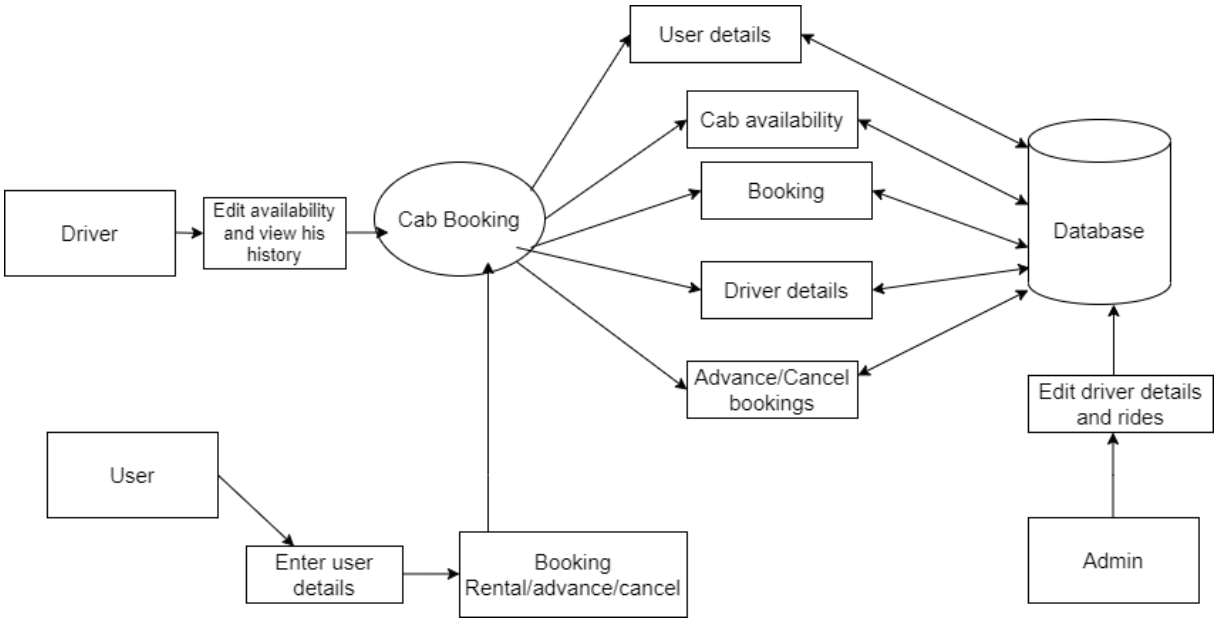
Case 3:

If the user enters as the driver, he will be able to change his availability in case of sickness or holiday after connecting to the admin and can also view his details.

DATA FLOW DIAGRAM LEVEL 0:

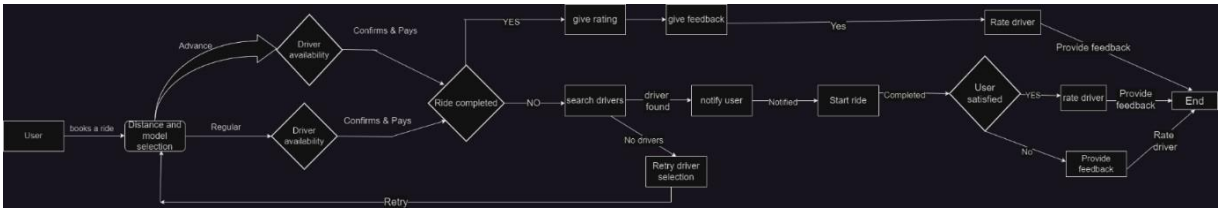


DATA FLOW DIAGRAM LEVEL - 1



The user inputs his details. Then the user details get stored in the database if he is a new user. If he is an existing user, he signs in by entering the allotted password for him which is also stored in the database. This process gets done by various functions in our program. Then the user chooses any one of the three options – regular booking, advance booking or cancellation. Then he gives the necessary inputs for the option he chose. Then we check from the database, if we have any rides similar to the one that the user asked. If there is one, we'll send the user a message. The driver can edit the car availability and also view his history. The admin can edit the driver's details and ride details from the database.

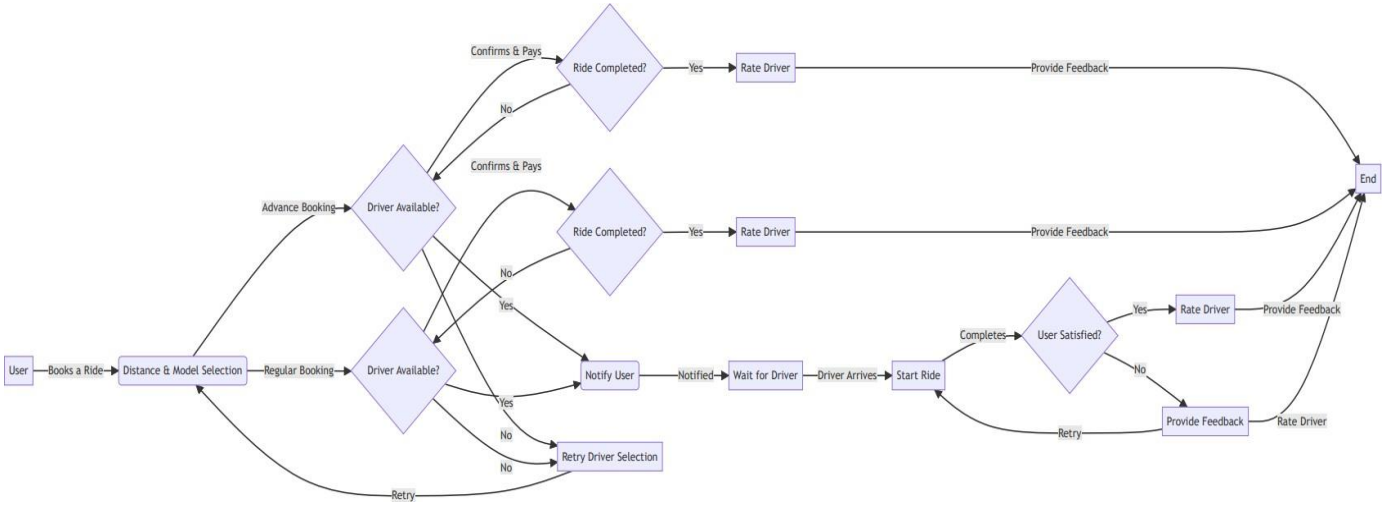
DATA FLOW DIAGRAM LEVEL - 2



The user enters his details. The user's details get checked. If he is an existing user, he signs in by entering his password. If he is a new user a new account is created for him, and his details get stored in the database. Then he books a ride by entering the start and drop location and then he selects the model of the car. Then he chooses whether he wants to book in advance, or it is a regular one. For both the cases We'll check the distance between the start and drop locations and we'll check if there is a driver is nearer to the user location and drives the model of the car entered by the user. If there is one, we'll notify him. If there is no driver, we'll send him a message that there's no driver

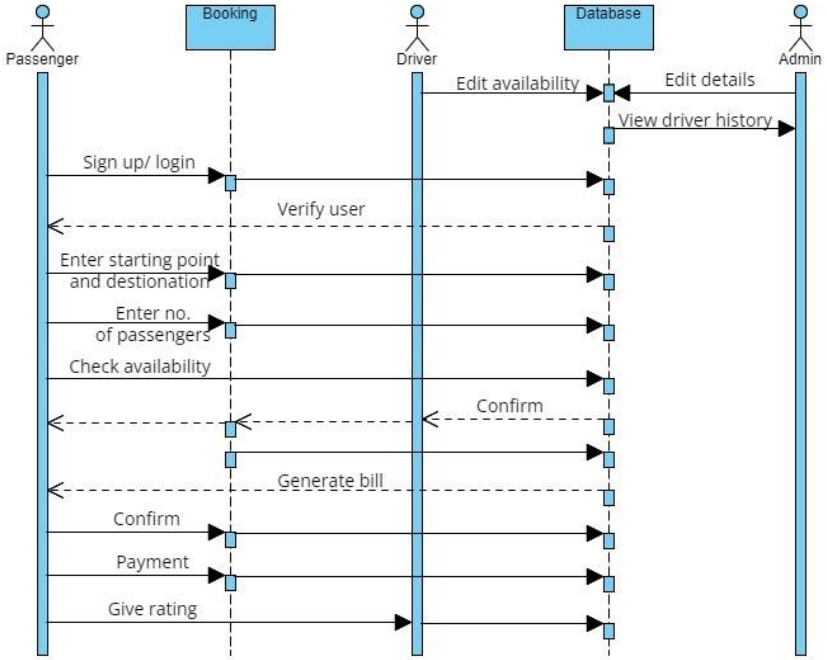
and he can try again. If a driver is found, he'll be notified. Then he makes the payment and takes the ride. Once the ride is completed, he'll rate the driver and gives the feedback.

ARCHITECTURE DIAGRAM :



The user enters his details. The user's details get checked. If he is an existing user, he signs in by entering his password. If he is a new user a new account is created for him and his details get stored in the database. Then he books a ride by entering the start and drop location and then he selects the model of the car. Then he chooses whether he wants to book in advance or it is a regular one. For both the cases We'll check the distance between the start and drop locations and we'll check if there is a driver is nearer to the user location and drives the model of the car entered by the user. If there is one, we'll notify him. If there is no driver, we'll send him a message that there's no driver and he can try again. If a driver is found, he'll be notified. Then he makes the payment and takes the ride. Once the ride is completed, he'll rate the driver. If the rating is below 2, we'll send him a message that we feel sorry for any inconvenience, and we'll ask him for feedback.

SEQUENCE DIAGRAM:

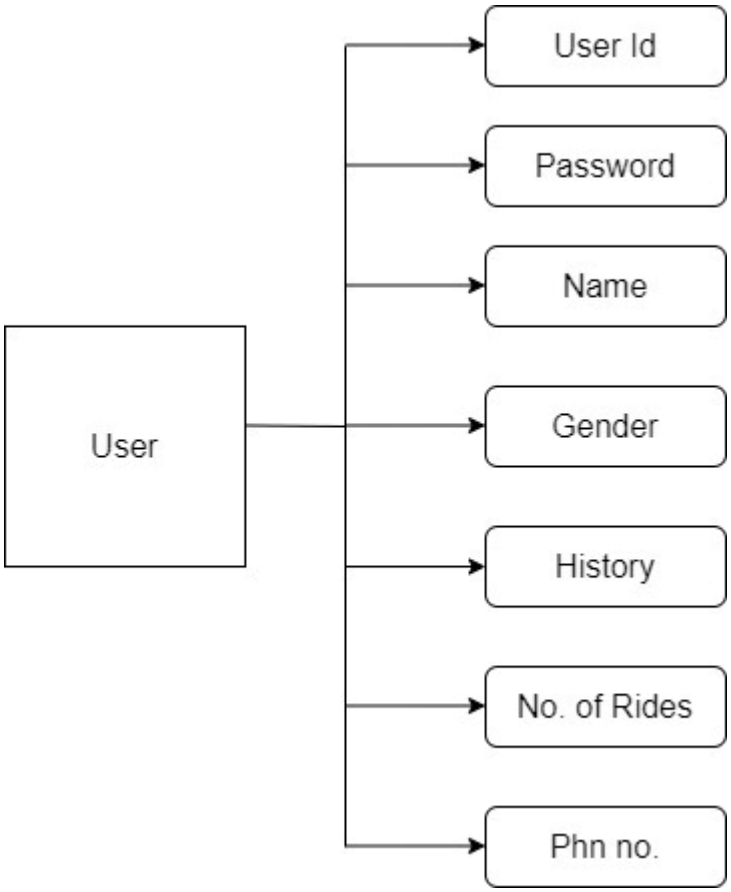


CONSTRUCT USED TO STORE DATA :

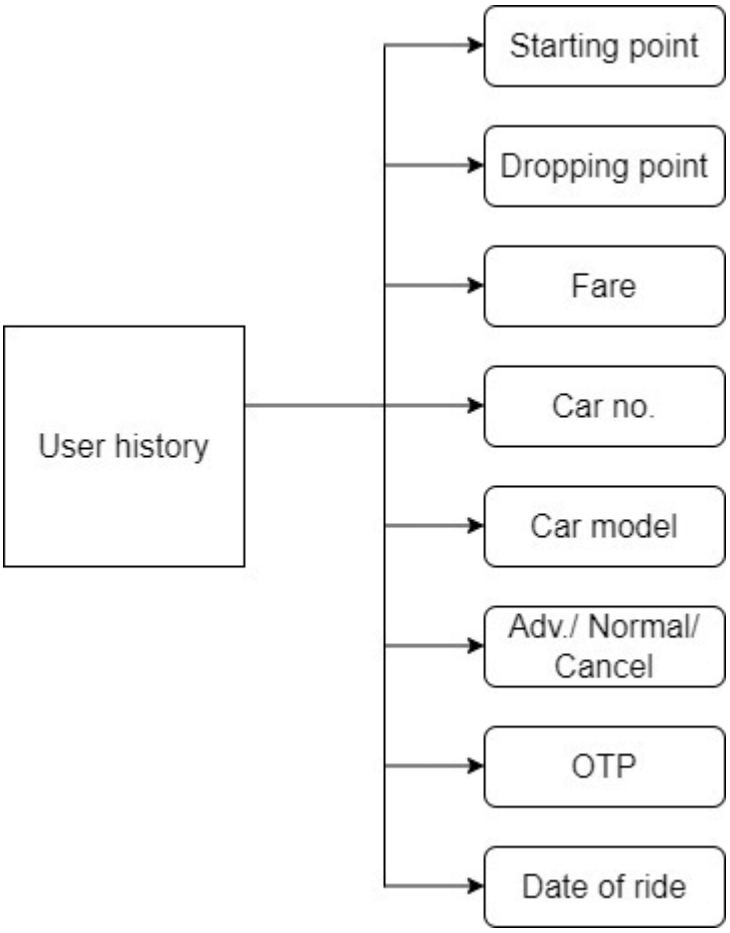
We used Files to access the data when we execute the program and the changes made in data during execution will be stored in the same files after closing the program. The data read in Files will be copied into structures, arrays, array of structures etc which will be explained in detail.

We used arrays of structure to store user, user history, driver history and fare data.

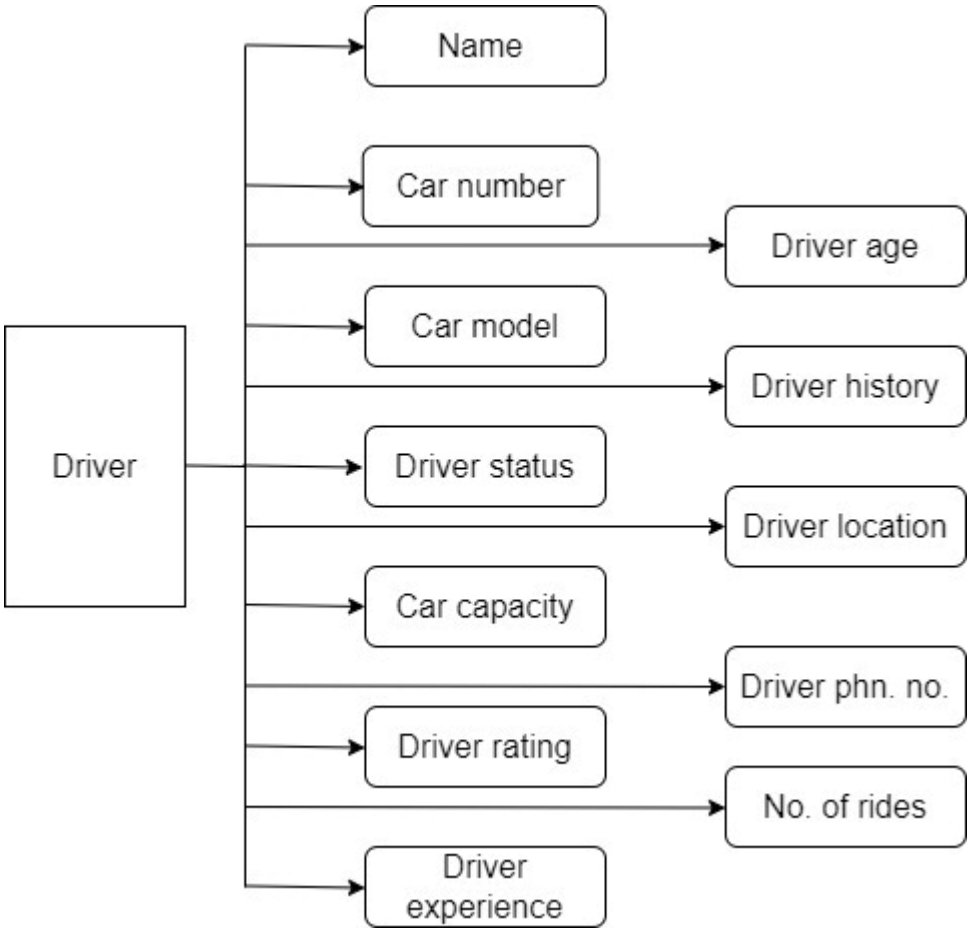
1. USER:



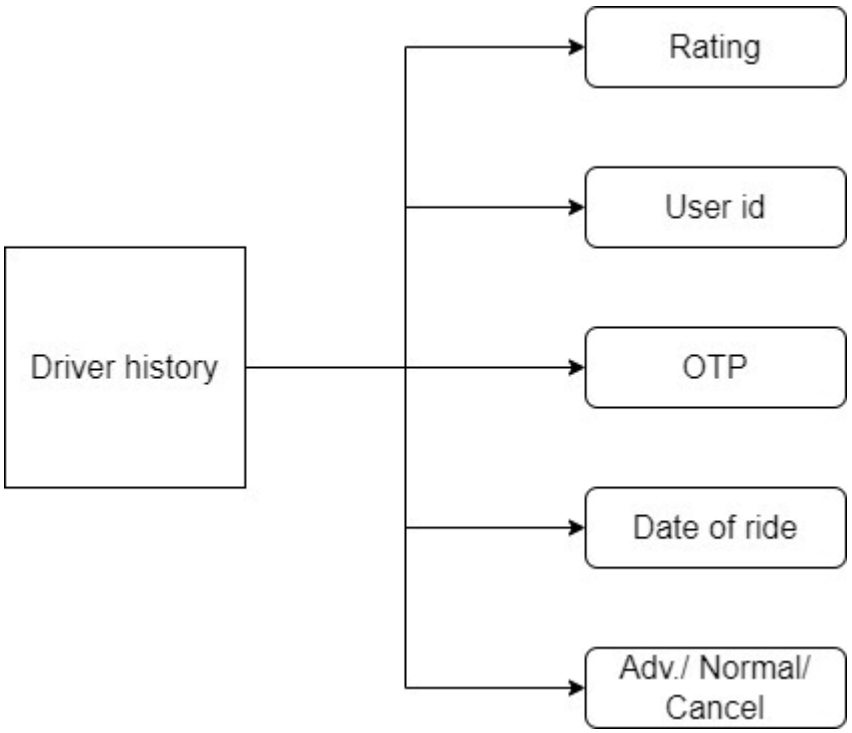
2. USER HISTORY:


User history
Starting point
Dropping point
Fare
Car no.
Car model
Adv./ Normal/ Cancel
OTP
Date of ride

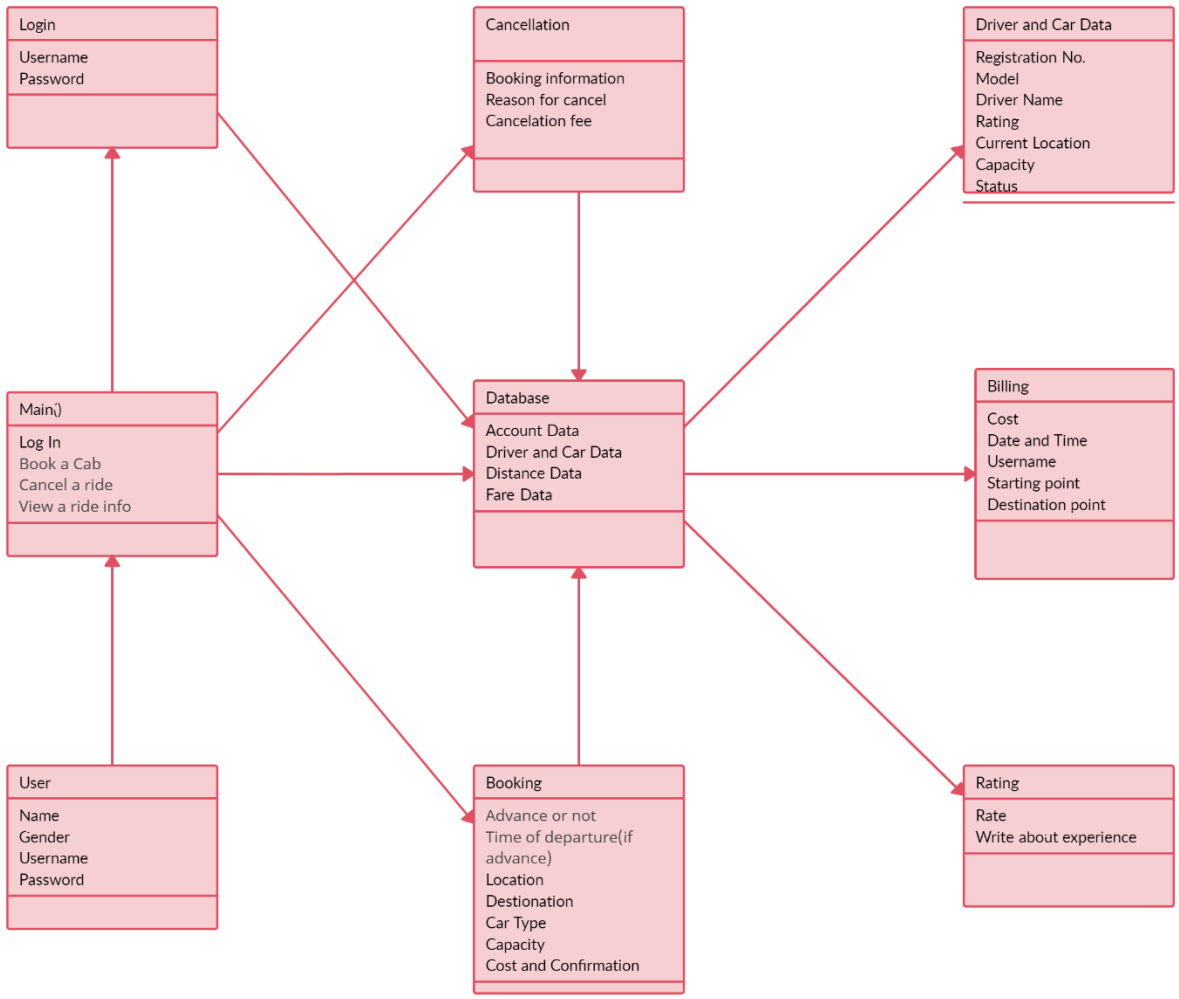
3. DRIVER :


Driver
Name
Car number
Driver age
Car model
Driver history
Driver status
Driver location
Car capacity
Driver phn. no.
Driver rating
No. of rides
Driver experience

4. DRIVER HISTORY :



STRUCTURE DIAGRAM:



IMPLEMENTATION :

LIBRARIES USED :

- stdio.h : We used it to use simple functions such as printf, scanf etc.....
- time.h : We used it for generating keys for rand() function.
- stdlib.h : We used it for accessing rand() function which is required for otp generation.
- string.h : We used it for accessing strings functions such as strcpy, strcmp, etc.....

APPLICATIONS USED FOR CODE DEVELOPMENT :

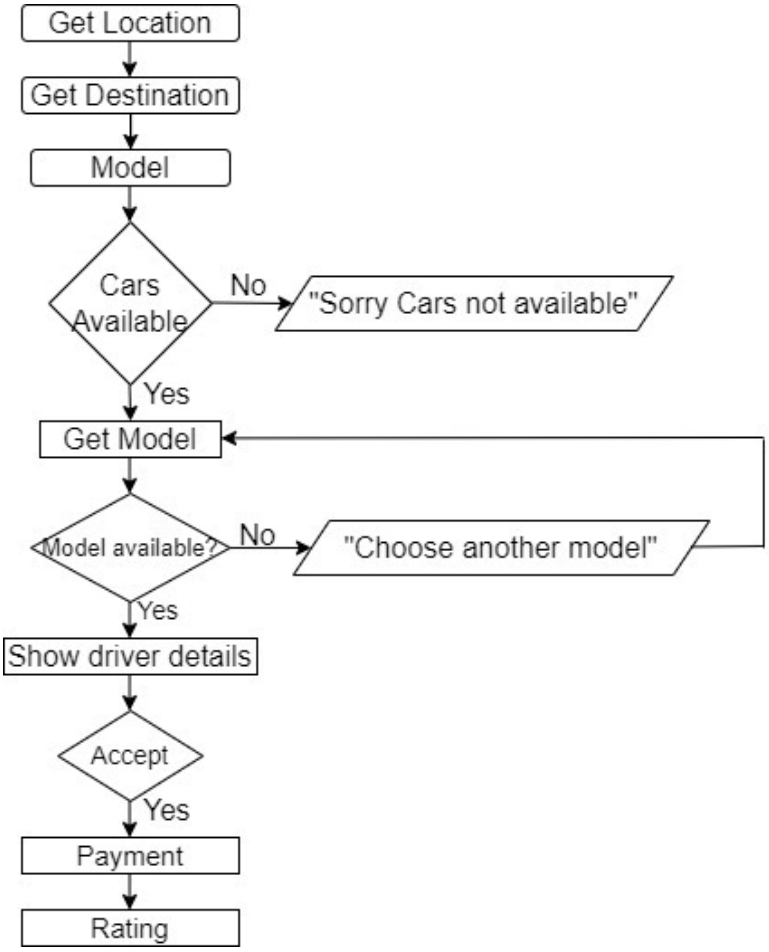
We used **DEVS C PROGRAMMING** for the code development as it is easy to work out test cases and detect errors in this applications.

CONSTRUCT USED TO STORE DATA :

We used Files to access the data when we execute the program and the changes made in data during execution will be stored in the same files after closing the program. The data read in Files will be copied into structures, arrays, array of structures etc which will be explained in detail

IMPORTANT MODULES:

BOOKING MODULE:



FUNCTIONS USED:

- loc(): It will take number of locations available as parameterand take the input of starting location of the ride.
- des(): It will take number of locations available as parameterand take the input of destination of the ride.
- distance(): It will take two location as parameter and returnsthe distance between them with help of data.
- highrated(): It will take an array of IDs of drivers asparameter and return the high rated Driver among them.
- fare(): It will take riding informations like model of cab, type of booking(advance or normal),and distance as parameter and calculatesand return the total cost of the ride

WORKING

- The user will give input of the starting point and destination ofthe ride through loc() and

des() function.

- The distance is calculated by the distance() function.
- The user will give the cab model. If the model is not available inthat location, the user are asked to give another model. If no cabsare available in that location then the user will be informed thatno cabs are available currently.
- The available driver IDs are stored in an array. Functionhighrated() will return the highly rated driver's id.
- Then the details of the drivers are shown to user with the help ofstructure and fare is calculated and shown to user by fare() function using booking details as parameters.
- The User can accept or reject the ride. If he reject he will be backto user module.
- If he/she accepts then the datasets are updated (like location ofdriver is changed to destination, etc).
- The Payment is to be made after the ride and user can rate theexperience.
- If the rating is less than 2 then he/she is asked to raise complaintthrough given email and the manager can act.

CODE:

```
void booking(int id, int nmodel, int nloc)
{
   int d;
   time_t t;
   int x,y,otp;
   getchar();
   x=loc(nloc);

   y=des(nloc);
   char str[15];
   strcpy(str, loc1);
   char des[15];
   strcpy(des, des1);
   d=distance(x,y);
   int i;
   int drl[50];
   char model[30];
   printf("Enter the model you need from the following lists \n");
   for(i=0; i<nmodel; i++)
   {
         printf("%s\n", m[i].model);
   }
   int p=1;
   int n,count,ans;
   while(p==1)
   {

     gets(model);
     count =0;
     n = 0;
```

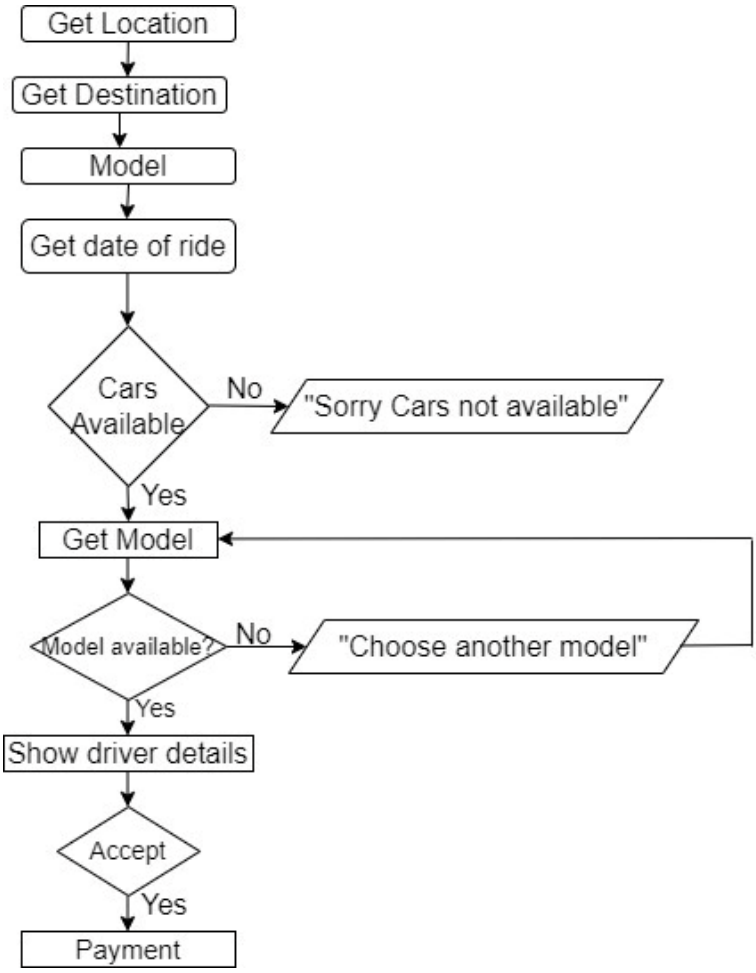
```
      ans;

      for (i=0; i<50; i++)
      {
          if(strcmp(dr[i].loc, str)==0 && dr[i].status == 1 )
          {
                count++;
              if(strcmp(dr[i].model, model)==0)
              {
                    drl[n]=i;
                    n++;
              }

          }



      }
      if (n==0 && count != 0)
      {
          printf("No cars in this model is available Choose another
model : \n");

      }
      else
      {
          p=0;
      }
     }
    if (count==0)
    {
          printf("SORRY NO CARS ARE AVAILABLE IN THE LOCATION\n");
     }
```

```
    else
    {
        ans = highrated(drl,n);
        printf("DRIVER FOUND\n");
        printf("ID : %d\n", dr[ans].car);
        printf("Name : %s\n", dr[ans].name);
        printf("Age : %d\n", dr[ans].age);
        printf("Experience : %d years\n", dr[ans].exp);
        printf("Average Rating : %.2f\n", dr[ans].rating);
        printf("Total charge = %f\n", fare(d,0,model));
        printf("****\n");
        printf("Press 1 to accept, Press 2 to cancel : \n");
        int res;
        scanf("%d",&res);
        if (res ==1)
        {
            a[id].rides++;
            srand(t);
            otp=rand()%1000;
            printf("The otp is %d\n", otp);
            printf("The Phone number of driver is %d\n",
dr[ans].phone);
            dr[ans].drhis[dr[ans].rides-1].otp=otp;
            a[id].his[a[id].rides-1].otp=otp;
            dr[ans].rides++;
            a[id].his[a[id].rides-1].car = dr[ans].car;
            strcpy(a[id].his[a[id].rides-1].drop, des);
            a[id].his[a[id].rides-1].fare =fare(d,0,model);
            strcpy(a[id].his[a[id].rides-1].model, dr[ans].model);
            strcpy(a[id].his[a[id].rides-1].start, str);
            dr[ans].status = 0;
            int gg=0;
```

```
		printf("\nEnjoy your ride. When the ride is over press
1\n");


		while(gg!=1)
		{
			scanf("%d", &gg);
		}


		dr[ans].status = 1;
		strcpy(dr[ans].loc, des);
		strcpy(dr[ans].drhis[dr[ans].rides-1].user, a[id].user);
		float rating;
		a[id].his[a[id].rides-1].adv =0 ;
		dr[ans].drhis[dr[ans].rides-1].adv=0;
		a[id].his[a[id].rides-1].adv=0;
		printf("Proceed payment through cash or through UPI id
2022\n");
		printf("…… \n");
		printf("Payment  Successful");
		printf("\nRate your experience(1 t0 5) : ");
		while(gg==1)
		{
			scanf("%f", &rating);
			if(rating<1 ||rating>5)
			{
				printf("Enter the values from 1 to 5 : ");
			}
			else
			{
				gg=0;
			}
		}
```

```
                dr[ans].rating=((dr[ans].rating * (dr[ans].rides-1)) +
rating)/dr[ans].rides;

                dr[ans].drhis[dr[ans].rides-1].rating = rating;

                if(rating <= 2)

                {

                      printf("Sorry for your bad experience, You can
contact our company through email abc@gamil.com or dial 99999 for
complain\n ");

                }

                else

                {

                      printf("Thank you for your rating:), share your
experience through email abc@gamil.com or dial 99999 \n");

                }

          }

    }

}
```

ADVANCE BOOKING MODULE



- loc() : It will take number of locations available as parameter andtake the input of starting location of the ride.
- des(): It will take number of locations available as parameter andtake the input of destination of the ride.
- distance(): It will take two location as parameter and returns thedistance between them with help of data.
- highrated(): It will take an array of IDs of drivers as parameterand return the high rated Driver among them.

fare(): It will take riding informations like model of cab, type of booking(advance or normal),and distance as parameter and calculates and return the total cost of the ride.

- advrating(): After the ride is over the user are asked to enter thisfunction manually and give rating. This function will update the datasets and make the driver available again.

WORKING:

- The user will give input of the starting point and destination ofthe ride through loc() and des() function.
- The distance is calculated by the distance() function.

- The user will give the date of the ride which will be sent to thedriver history structure.
- The user will give the cab model. If the model is not available inthat location, the user are asked to give another model. If no cabsare available in that location then the user will be informed thatno cabs are available currently.
- The available driver IDs are stored in an array. Functionhighrated() will return the highly rated driver's id.
- Then the details of the drivers are shown to user with the help ofstructure and fare is calculated and shown to user by fare() function using booking details as parameters.
- The User can accept or reject the ride. If he reject, he will beback to user module.
- If he/she accepts then the datasets are updated (like location ofdriver is changed to destination, etc).
- The Payment is to be made before the ride and user can rate the experience after the ride is over by manually entering advrating().
- advrating() will make the driver available.
- If the rating is less than 2 then he/she is asked to raise complaintthrough given email and the manager can act.

CODE:

```
void advance(int id, int nmodel, int nloc)
{

   int  d;
   time_t  t;
   int x,y;
   getchar();
   x=loc(nloc);
   y=des(nloc);
   char str[15];
   strcpy(str, loc1);
   char des[15];
   strcpy(des, des1);
   d=distance(x,y);
```

```
char date[10];
printf("Enter the date for your booking :\n");
gets(date);
int i;
int drl[50];
char model[30];
int otp;
printf("Enter the model you need from the following lists \n");
for(i=0; i<nmodel; i++)
{
      printf("%s\n", m[i].model);
}
int p=1;
int n,count,ans;
while(p==1)
{

  gets(model);
  count =0;
  n = 0;
  ans;

  for (i=0; i<50; i++)
  {
      if(strcmp(dr[i].loc, str)==0 && dr[i].status == 1 )
      {
            count++;
          if(strcmp(dr[i].model, model)==0)
          {
                drl[n]=i;
                n++;
          }
```

```
        }


    }
    if (n==0 && count != 0)
    {
        printf("No cars in this model is available Choose another
model : \n");


    }
    else
    {
        p=0;
    }
   }
  if (count==0)
  {
        printf("SORRY NO CARS ARE AVAILABLE IN THE LOCATION\n");
   }
   else
   {
        ans = highrated(drl,n);
        printf("DRIVER FOUND\n");
        printf("ID : %d\n", dr[ans].car);
        printf("Name : %s\n", dr[ans].name);
        printf("Age : %d\n", dr[ans].age);
        printf("Experience : %d years\n", dr[ans].exp);
        printf("Average Rating : %.2f\n", dr[ans].rating);
        printf("Total charge = %f\n", fare(d,1,model));
        printf("****\n");
        printf("Press 1 to accept, Press 2 to cancel : \n");
```

```
        int res;
        scanf("%d",&res);
        if (res ==1)
        {
              a[id].rides++;
              dr[ans].rides++;
              a[id].his[a[id].rides-1].car = dr[ans].car;
              strcpy(a[id].his[a[id].rides-1].drop, des);
              srand(t);
              otp=rand()%1000;
              printf("\n%s\n", date);
              printf("The otp is %d\n", otp);
              dr[ans].drhis[dr[ans].rides-1].otp=otp;
              a[id].his[a[id].rides-1].otp=otp;
              a[id].his[a[id].rides-1].fare =fare(d,1,model);
              strcpy(a[id].his[a[id].rides-1].date, date);
            strcpy(dr[ans].drhis[dr[ans].rides-1].date, date);
              strcpy(a[id].his[a[id].rides-1].model, dr[ans].model);
              strcpy(a[id].his[a[id].rides-1].start, str);
              dr[ans].status = 0;
              int gg=0;
              printf("Proceed advance payment through UPI id 2022\n");
              printf("…… \n");
              printf("Payment Successful\n");
              printf("The otp is %d\n", otp);
              printf("The Phone number of driver is %d\n",
dr[ans].phone);
              a[id].his[a[id].rides-1].adv =1;
              dr[ans].drhis[dr[ans].rides-1].adv=1;
              strcpy(dr[ans].loc, des);
              strcpy(dr[ans].drhis[dr[ans].rides-1].user, a[id].user);
              printf("Do not forget to rate once the ride is over
\n");
```

```
            }
        }
    }
```

EXTRA CONTENTS :

USER:

- User can we their history of rides such as car id, fare, starting point, destination date of ride, car model, rating, etc.
- OTP will also stored in the history so that it will be easy for the user.
- The rides are classified into 1. Normal booked and completed, 2.Advance booked and completed,3. Advance booked and not completed,4. Cancelled.
- When traveling interior of cities (ex inside Chennai), only fixed fee is applied.

DRIVER:

- Driver can edit availability with the permission of admin.
- When driver use this option, a call is directed to manager, to whom the driver should state the reason.
- Similar to user, driver can view his/her rides history.
- The location of driver will be changed to the destination of the ride after each rides.

ADMIN :

Admin can do the following :

- Add new driver/ car
- Add new location
- Edit driver/car details
- Edit fares
- Add new model
- View fare and driver information
- View available information

TEST CASES :

1. NORMAL BOOKING :

```
Press 1 to Book a cab
Press 2 for advance booking
Press 3 for renting a cab
Press 4 to rate your previously advance booked cab
Press 5 to cancelation
Press 6 to view history
Press any other number to log out
1
Chennai
Kelambakkam
Kalvakkam
Kovalam
Tambaram
Tiruporur

Please Enter the location name from the above cities
Chennai
Chennai
Kelambakkam
Kalvakkam
Kovalam
Tambaram
Tiruporur

Please Enter the destination name from the above cities
Tambaram
Enter the model you need from the following lists
Taxi
Luxury Car
Auto
SUV
Mini Car
Auto
DRIVER FOUND
ID : 1000
Name : Aananth
Age : 35
Experience : 7 years
Average Rating : 5.00
Total charge = 282.000000
```

```
Press 1 to accept, Press 2 to cancel :
1
The otp is 719
The Phone number of driver is 1275936408

Enjoy your ride. When the ride is over press 1
1
Proceed payment through cash or through UPI id 2022
..........
Payment Successful
Rate your experience(1 t0 5) : 4
Thank you for your rating:), share your experience through email abc@gamil.com or dial 99999
```

2. ADVANCE BOOKING :

```
Please Enter the location name from the above cities
Chennai
Chennai
Kelambakkam
Kalvakkam
Kovalam
Tambaram
Tiruporur

Please Enter the destination name from the above cities
Kovalam
Enter the date for your booking :
21/4/23
Enter the model you need from the following lists
Taxi
Luxury Car
Auto
SUV
Mini Car
SUV
No cars in this model is available Choose another model :
Mini Car
DRIVER FOUND
ID : 2001
Name : Kumar
Age : 32
Experience : 4 years
Average Rating : 3.90
Total charge = 554.000000
********
Press 1 to accept, Press 2 to cancel :
1

21/4/23
The otp is 719
Proceed advance payment through UPI id 2022
.........
Payment Successful
The otp is 719
The Phone number of driver is 1275937409
```

3. RENTAL BOOKING :

```
Welcome
We only rent for distance greater then 200 km
Enter the location
Chennai

Enter the destination
Agra
Enter the date :
23/11/23

Enter the distance
300
Taxi
Luxury Car
Auto
SUV
Mini Car

Enter the model names
Luxury Car
DRIVER FOUND
ID : 2001
Name : Kumar
Age : 32
Experience : 4 years
Average Rating : 3.90
The Fare will be : 9600
To accept enter 1 and to reject enter 2
1
You have booked it sucessfully
Please proceed for thr payment
.......
Payment sucessfull

23/11/23
The otp is 27
The Phone number of driver is 1275937409
Do not forget to rate once the ride is over
```

4. VIEWING DRIVER DETAILS AND HISTORY :

```
Welcome
Press 1 to Add driver
Press 2 to Edit driver
Press 3 to Add city
Press 4 to Add model
Press 5 to edit fare
Press 6 to view driver history
Press 7 to view fares
Press 8 to view available locations
Press any other number to log out
6
Enter driver number : (1 to 50)
0
Name : Aananth
ID : 1000
No. of rides taken : 2
Experience : 7 years
Avg rating : 5.000000
Phone no. : 1275936408
Status : 1
Current Location : Kovalam
1
User : Test
Rating : 5.000000
2
User : walter
Rating : 5.000000
```

5. ADD MODEL :

```
Press 1 to Add driver
Press 2 to Edit driver
Press 3 to Add city
Press 4 to Add model
Press 5 to edit fare
Press 6 to view driver history
Press 7 to view fares
Press 8 to view available locations
Press any other number to log out
4
Enter the model name :
Two Wheelers
Enter the maxdistance fee :
23
Enter min distance fee :
12
Enter fixed fee :
32
Enter cancellation fee :
3
Enter advanced fees :
23
Enter relaxation fees :
34
Enter surge fees :
43
Model added succesfully
```

6. ADD LOCATIONS:

```
Welcome
Press 1 to Add driver
Press 2 to Edit driver
Press 3 to Add city
Press 4 to Add model
Press 5 to edit fare
Press 6 to view driver history
Press 7 to view fares
Press 8 to view available locations
Press any other number to log out
3
Enter the name of the city :
Perungalathur
Enter the distance from Chennai :
12
Enter the distance from Kelambakkam :
10
Enter the distance from Kalvakkam :
19
Enter the distance from Kovalam :
30
Enter the distance from Tambaram :
5
Enter the distance from Tiruporur :
18
```

7. ADD DRIVERS :

```
Press 1 to Add driver
Press 2 to Edit driver
Press 3 to Add city
Press 4 to Add model
Press 5 to edit fare
Press 6 to view driver history
Press 7 to view fares
Press 8 to view available locations
Press any other number to log out
1
Write the name of driver :
Deva

Please Enter the Destination name from the above cities
Chennai
Kelambakkam
Kalvakkam
Kovalam
Tambaram
Tiruporur
Perungalathur
Perungalathur
Enter the model you need from the following lists
Taxi
Luxury Car
Auto
SUV
Mini Car
Two Wheelers
Two Wheelers
Enter the age of driver :
32
Enter the car id : 2812
Enter car capacity :
1
Enter the experience of driver :
3
Enter mobile number : 3725638293
Driver added successfully
Driver ID is 30
```

8. VIEW DETAILS :

```
Enter the model name :
Two Wheelers
Fixed Fee : 32
Advance Fee : 23
Cancellation Fee : 3
Long distance Fee : 23
Short distance Fee : 12
Relaxation Fee : 34
Surge Fee : 43


Welcome
Press 1 to Add driver
Press 2 to Edit driver
Press 3 to Add city
Press 4 to Add model
Press 5 to edit fare
Press 6 to view driver history
Press 7 to view fares
Press 8 to view available locations
Press any other number to log out
8
Chennai
Kelambakkam
Kalvakkam
Kovalam
Tambaram
Tiruporur
Perungalathur
```

9. VIEW USER HISTORY :

```
Press 1 to Book a cab
Press 2 for advance booking
Press 3 for renting a cab
Press 4 to rate your previously advance booked cab
Press 5 to cancelation
Press 6 to view history
Press any other number to log out
6
Name : Walter White
Gender : Male
Phone no. : 1362426988
No. of rides : 1
1
Starting point : Chennai
Destination : Kovalam
Car ID : 1000
Car model : Auto
OTP : 0
Fare : 372.000000
```

10. EDIT AVAILABLITY :

```
Press 1 to view history
Press 2 to change status
Press any other number to logout
2
Press 0 to turn off ride and 1 to turn on ride :
0
Contacting Manager
.................
```

ENVIRONMENTAL, ETHICAL, SOCIAL AND LEGAL ASPECTS :

Social perspective:

- Impact on traditional taxi drivers: The introduction of a cab hailing system may disrupt the traditional taxi industry, leading to job losses for taxi drivers who may find it difficult to compete with the convenience and affordability of the new system.

- Increase in employment : Many drivers can be employed in this system which increases economy of many people.

- Increase in Safety : As the background of the Drivers are checked, many crimes such as harassment, assault, or fraud can be avoided and passengers are more safe.

Legal perspective:

• Regulatory laws: Cab hailing systems need to comply with relevant laws and regulations, such as licensing requirements, safety standards, and data privacy laws.

• Data privacy: The cab hailing system should comply with data privacy laws and regulations and have clear policies for handling personal data, such as user profiles, location data, and payment information.

Environmental perspective :

1. Reduced emissions: By providing a more efficient and reliable transportation option, our services can reduce the number of cars on the road and the associated emissions from those vehicles. Additionally, we will be investing in electric or hybrid vehicles, which produce fewer emissions than traditional gasoline-powered cars.

2. Reduced need for parking: Our services can reduce the need for parking spaces, which can free up valuable land for other uses and reduce the environmental impact of parking infrastructure.

3. Improved traffic flow: Our services can also help to improve traffic flow by reducing the number of cars looking for parking or circling around looking for passengers. This can reduce congestion and associated emissions.

Ethical perspective :

1. Fair labor practices: Cab hailing companies have faced criticism for their treatment of drivers, including low pay, lack of benefits, and limited job security. Companies must ensure that they are treating their workers fairly and providing them with a living wage and other benefits.

2. Accessibility: Cab hailing services may not be accessible to all users, particularly those with disabilities or those living in areas without reliable internet access. Companies must ensure that their services are accessible to all users.

LIMITATIONS:

- In advance booked rides, we don’t know when the ride will be over. The user has to manually enter the advance rating module to confirm that the ride is over.
- In normal booking, we can’t estimate the time taken for the ride.
- In rental section the user has to manually enter the distance between locations.
- In rental section the starting point and destination typed cannot be verified (may be the destination is out of country or didn’t even exist).

LEARNING OUTCOMES:

- By doing this project we learned how to integrate different types of modules to develop a software.
- We learnt about the usage and implementation of structures and files.
- Learnt to analyse and solve a problem.
- Learnt about different types of errors and how to resolve them.

REFERENCE :

GeeksForGeeks, W3Schools, Tutorial Point, Schaum's Outlines Programming with C...