DRIVEIO

Naisha Mahiat Chowdhury
ID: 190204089

Fairuz Sharika Tasnim
ID: 190204102

Most. Mahazabin Prome
ID: 190204091

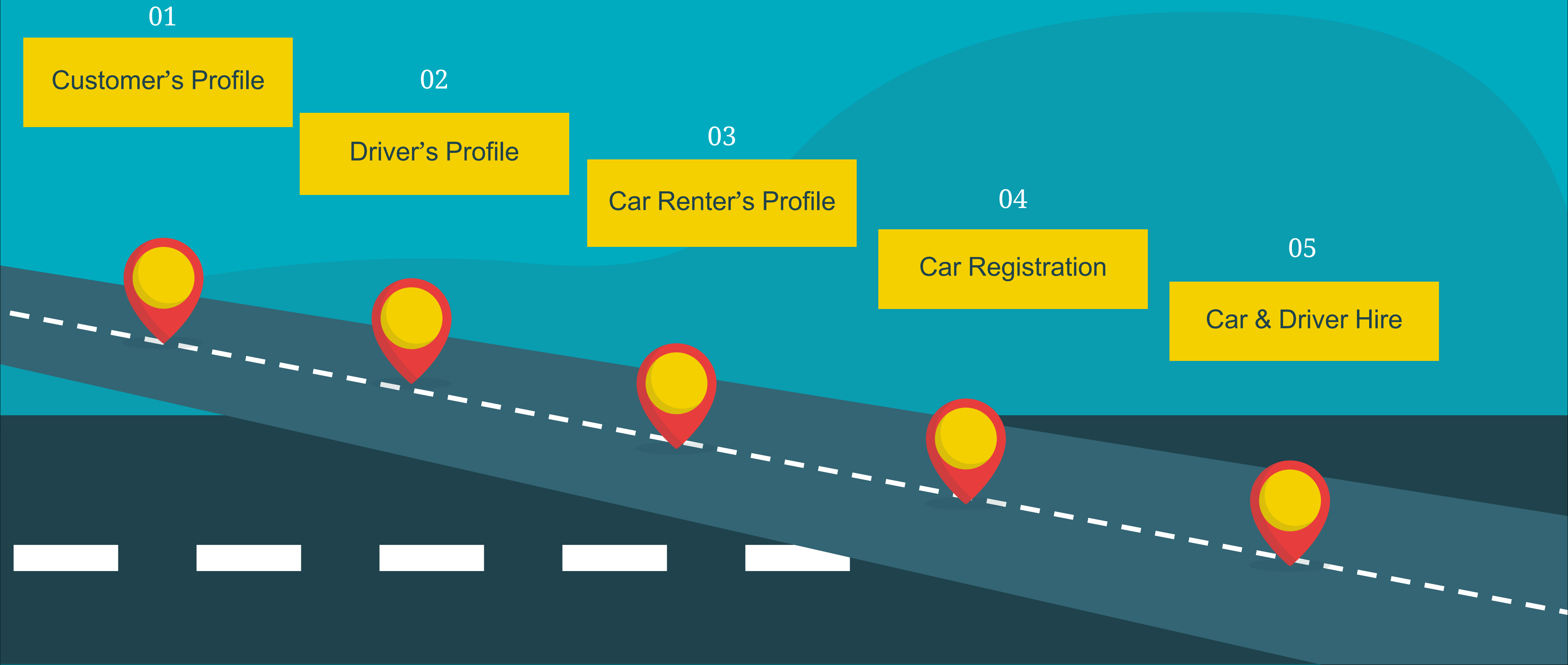
Fariha Hassan
ID: 190204109

Marzia Binta Monir
ID: 190204114

Introduction

The project we worked on is called ‘**Driveio**’. The main goal of our project was to make the transportation system easier for our users. The main users of our mobile application are car owners, drivers and the people who need a vehicle in terms of emergency. A car owner can rent his car to someone in need. A customer can borrow a car for some hours to days for his personal uses. Besides, someone can also hire a driver for few hours to drive their car.

Features of DRIVEIO



01 Customer’s Profile

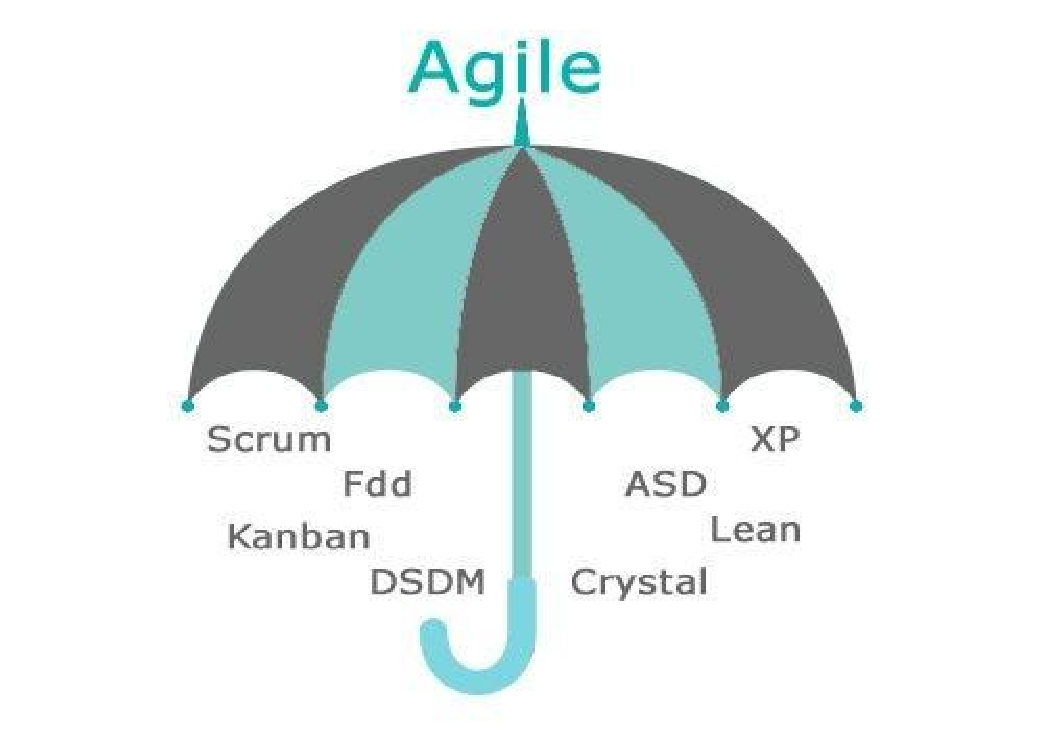
02 Driver’s Profile

03 Car Renter’s Profile

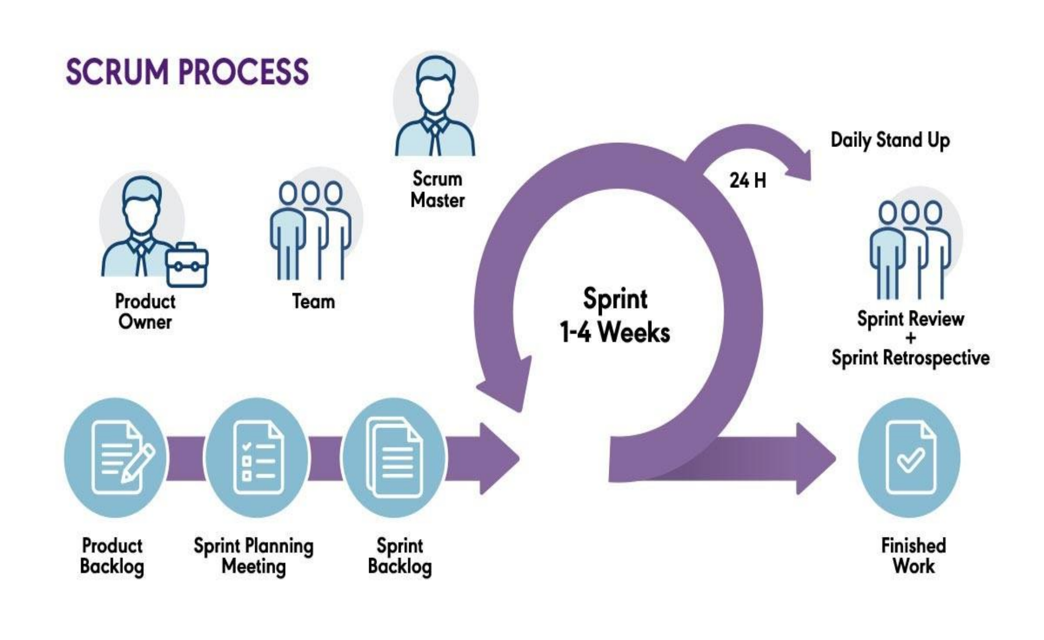
04 Car Registration

05 Car & Driver Hire

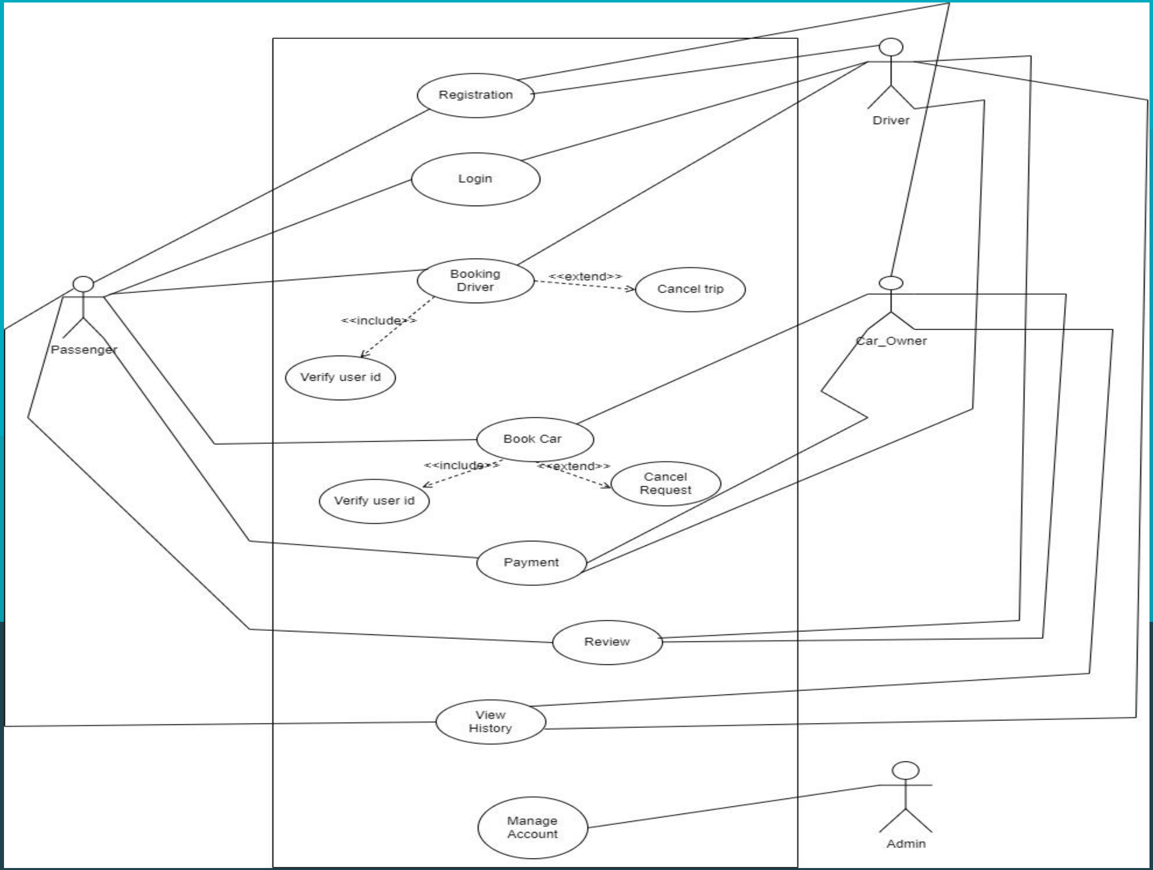
Process Model



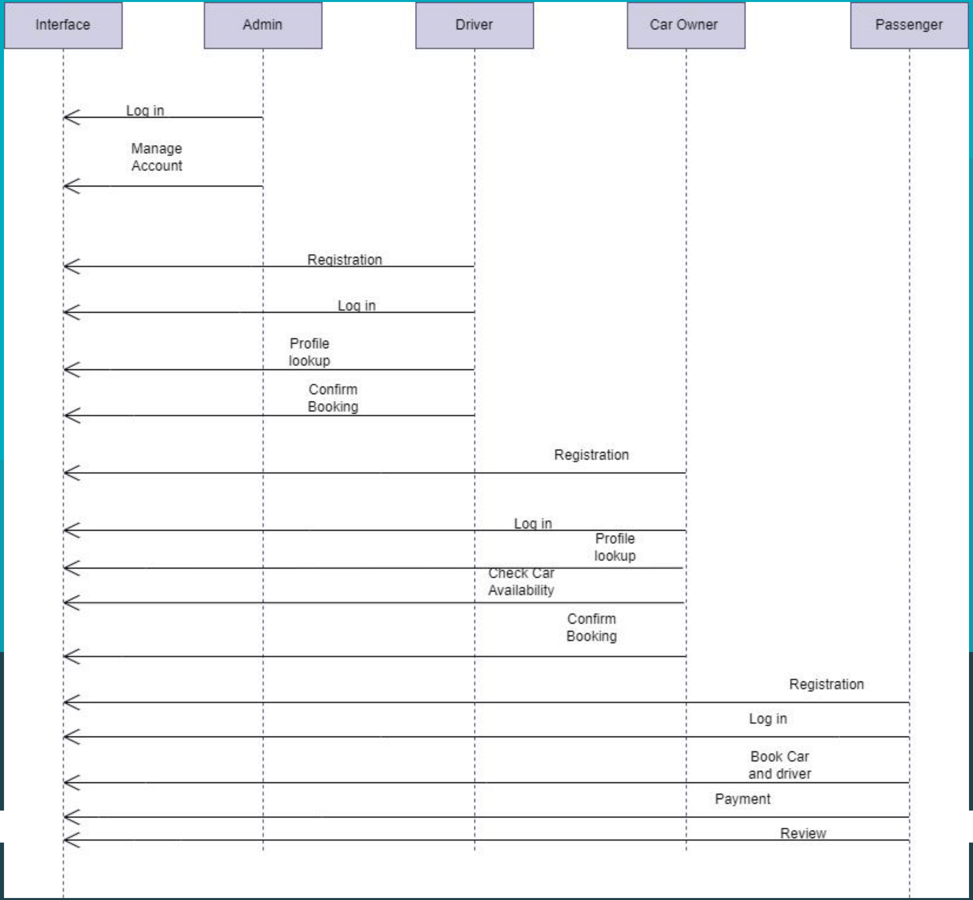
Process Model



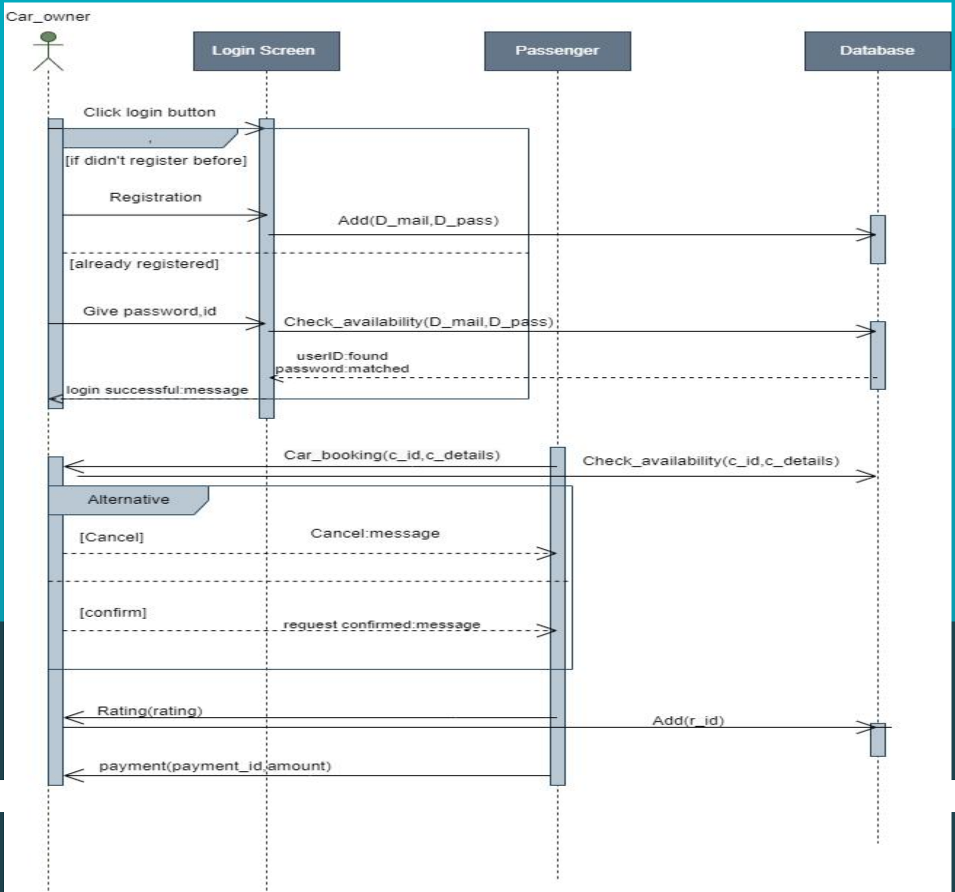
Use Case Diagram



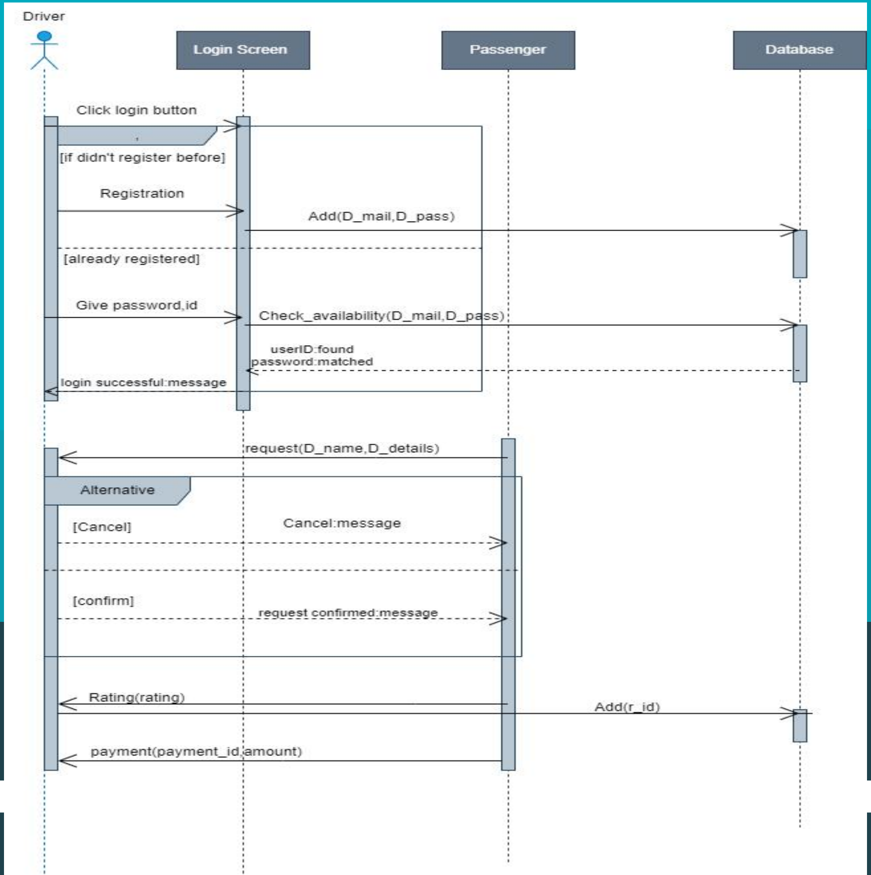
Sequence Diagram



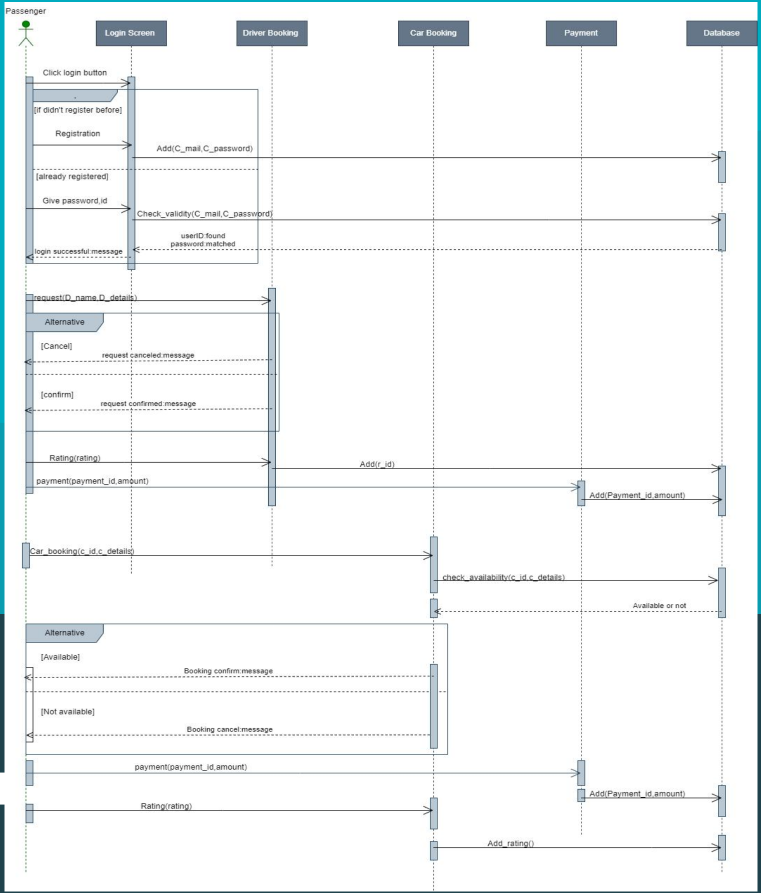
Sequence Diagram



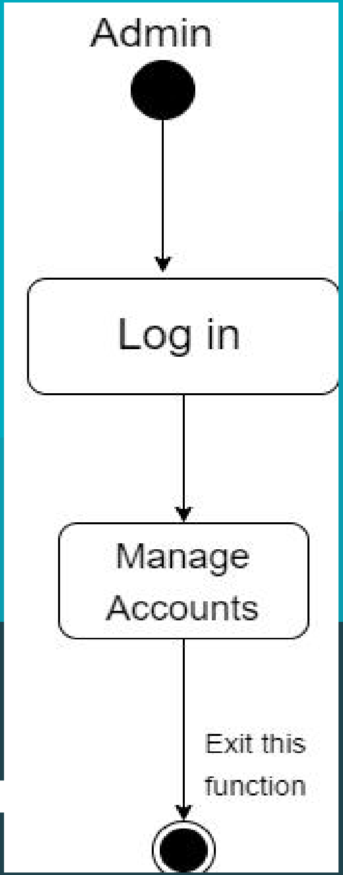
Sequence Diagram



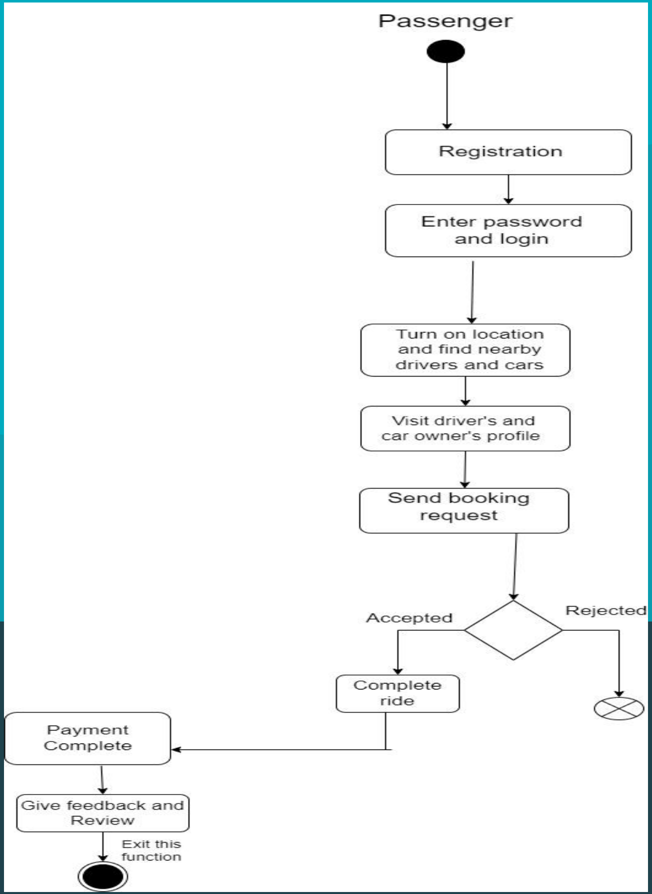
Sequence Diagram



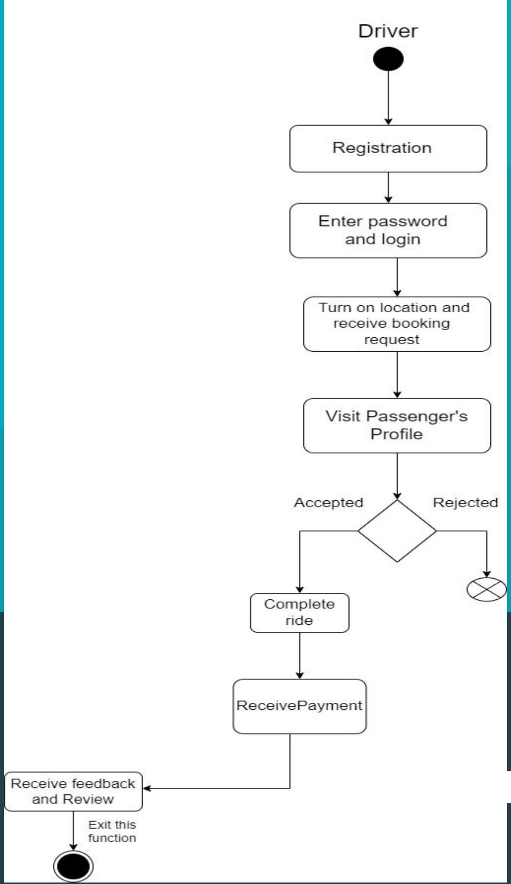
Activity Diagram



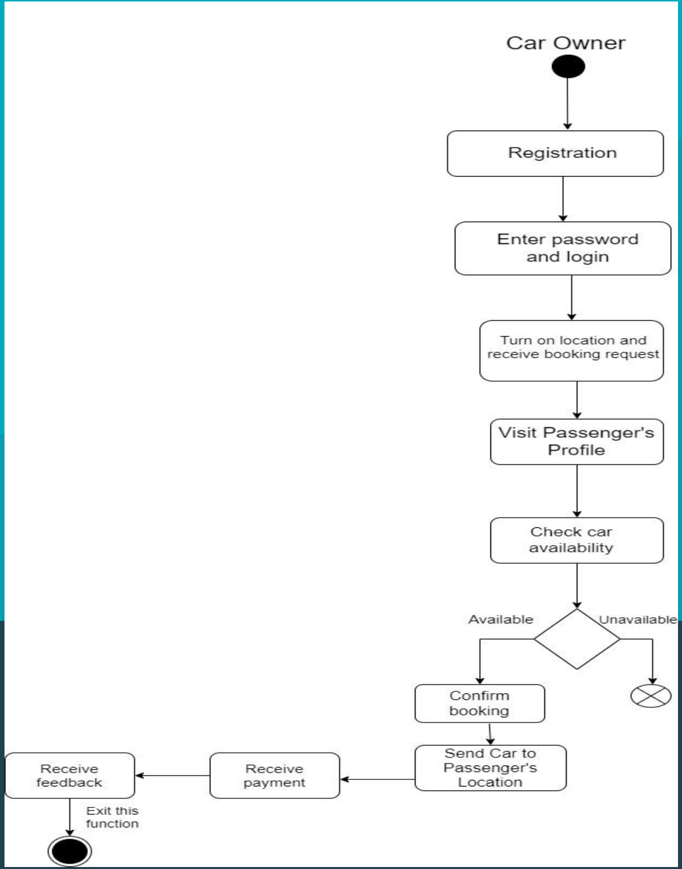
Activity Diagram



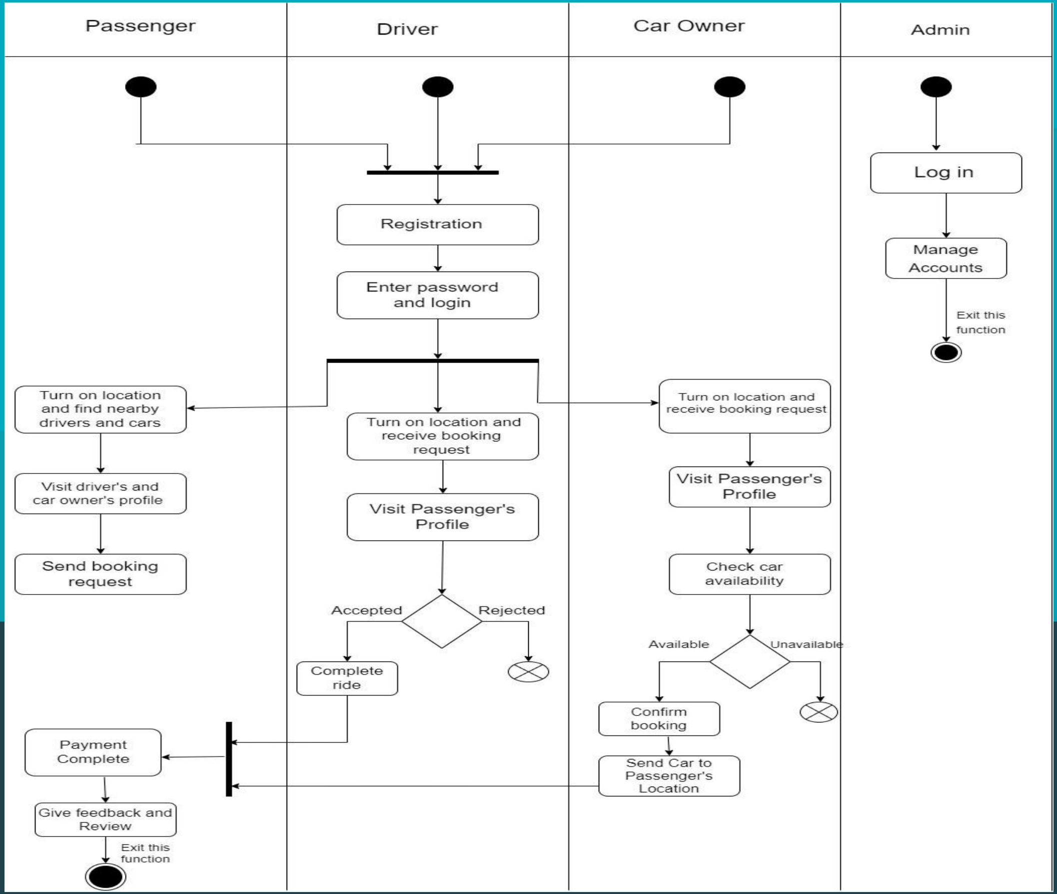
Activity Diagram



Activity Diagram

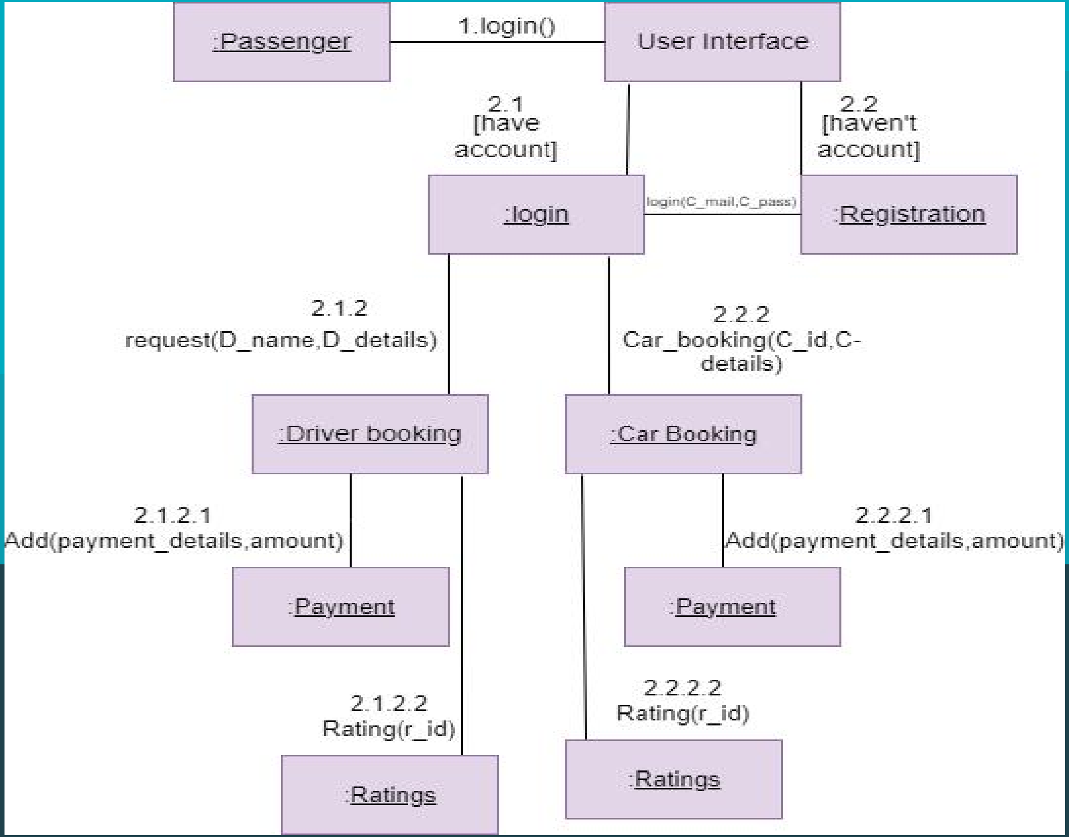


Swim Lane Diagram



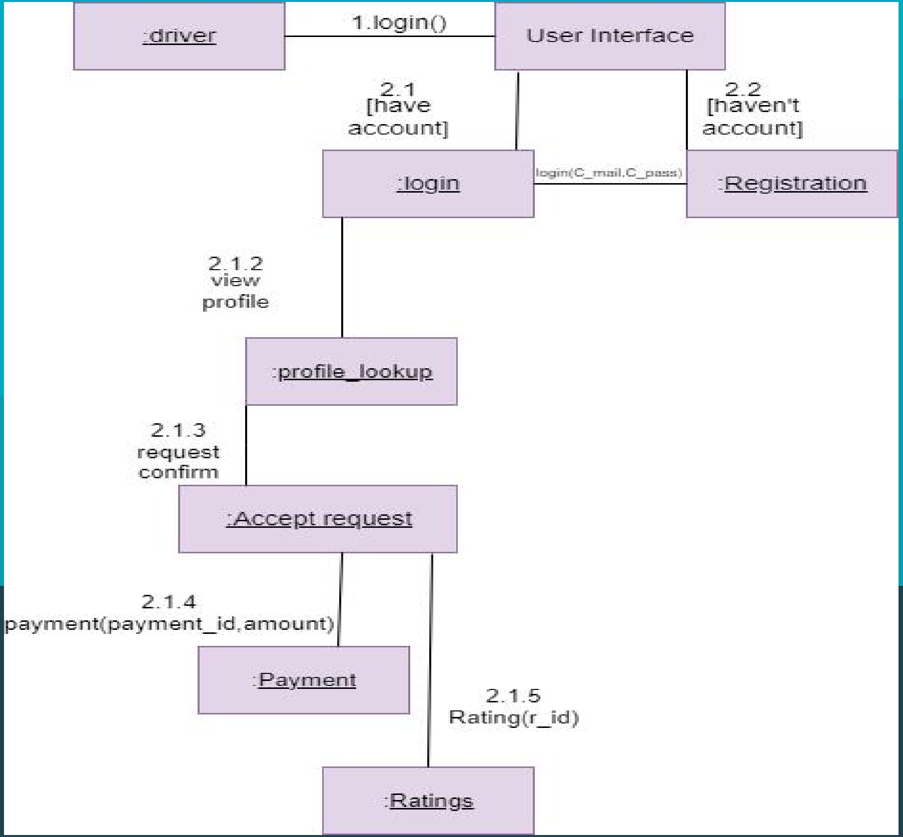
Collaboration Diagram

For passengers:



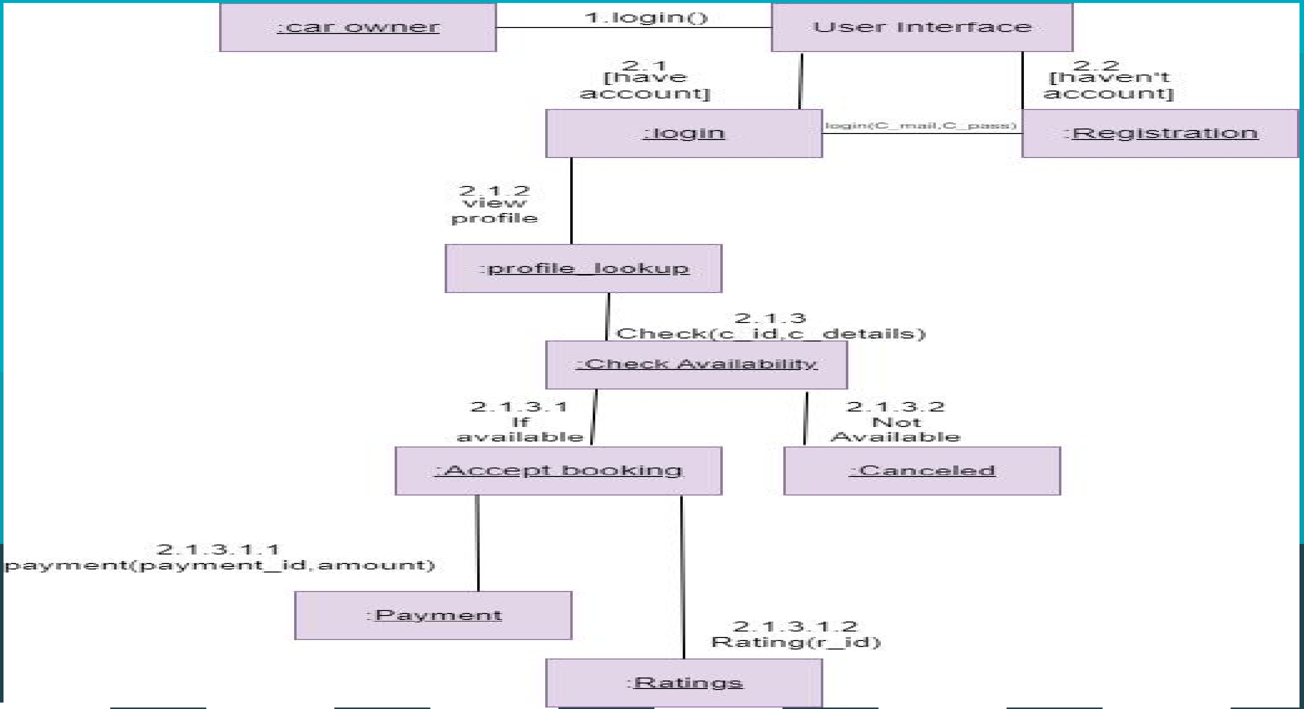
Collaboration Diagram

For drivers:



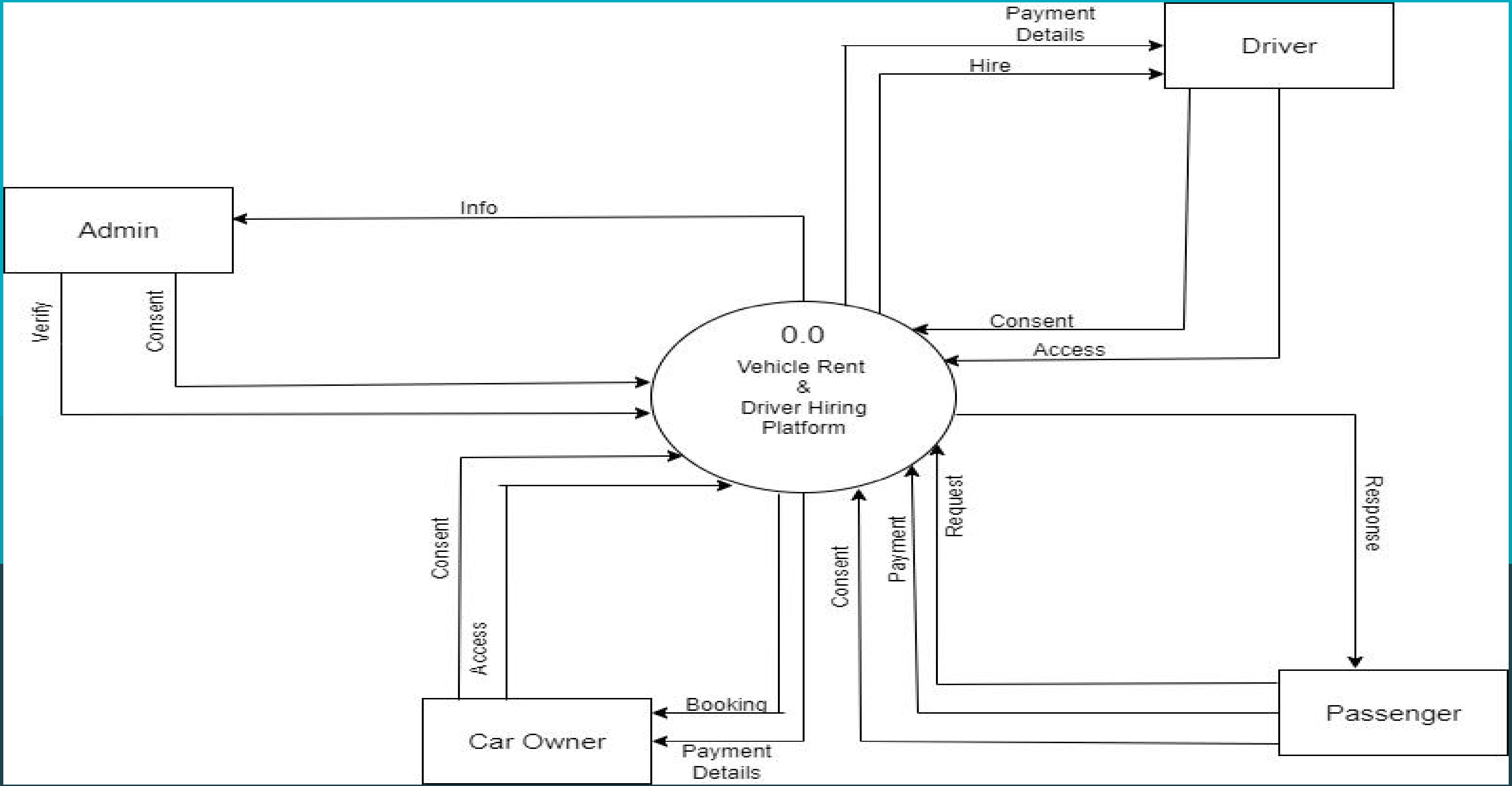
Collaboration Diagram

For car owners:



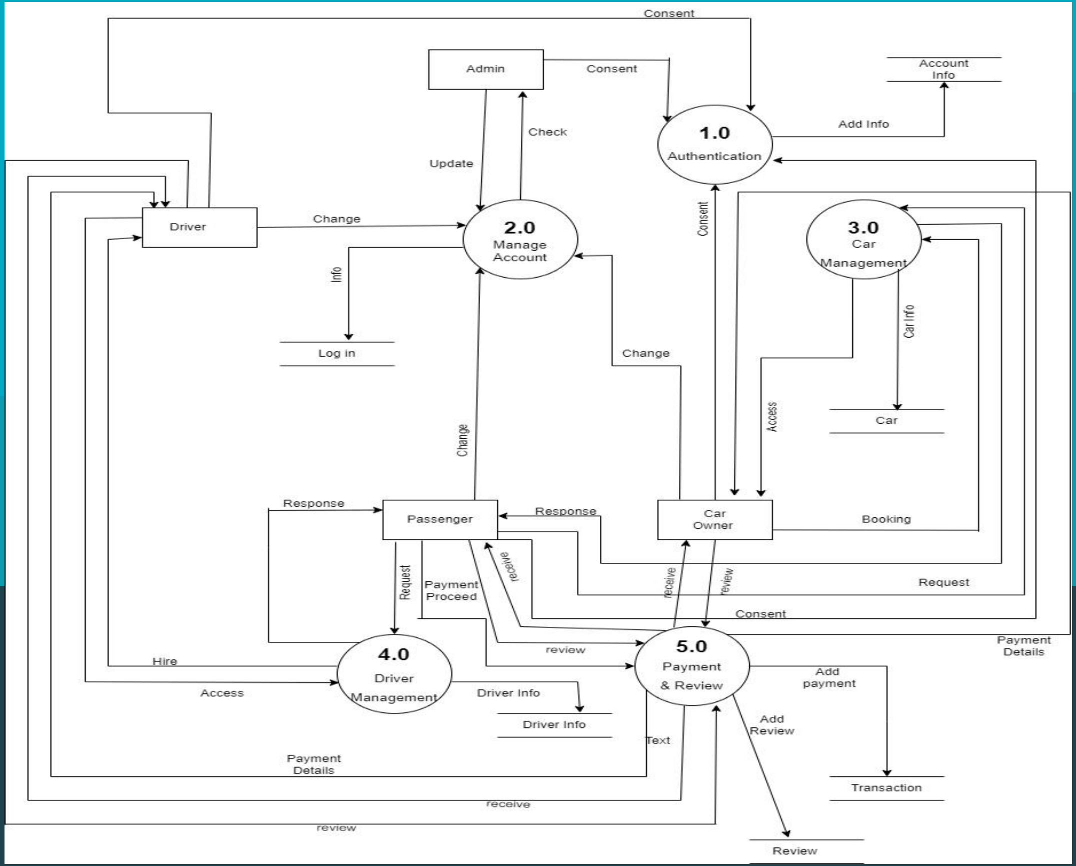
Data Flow Diagram

Level 0 context diagram:



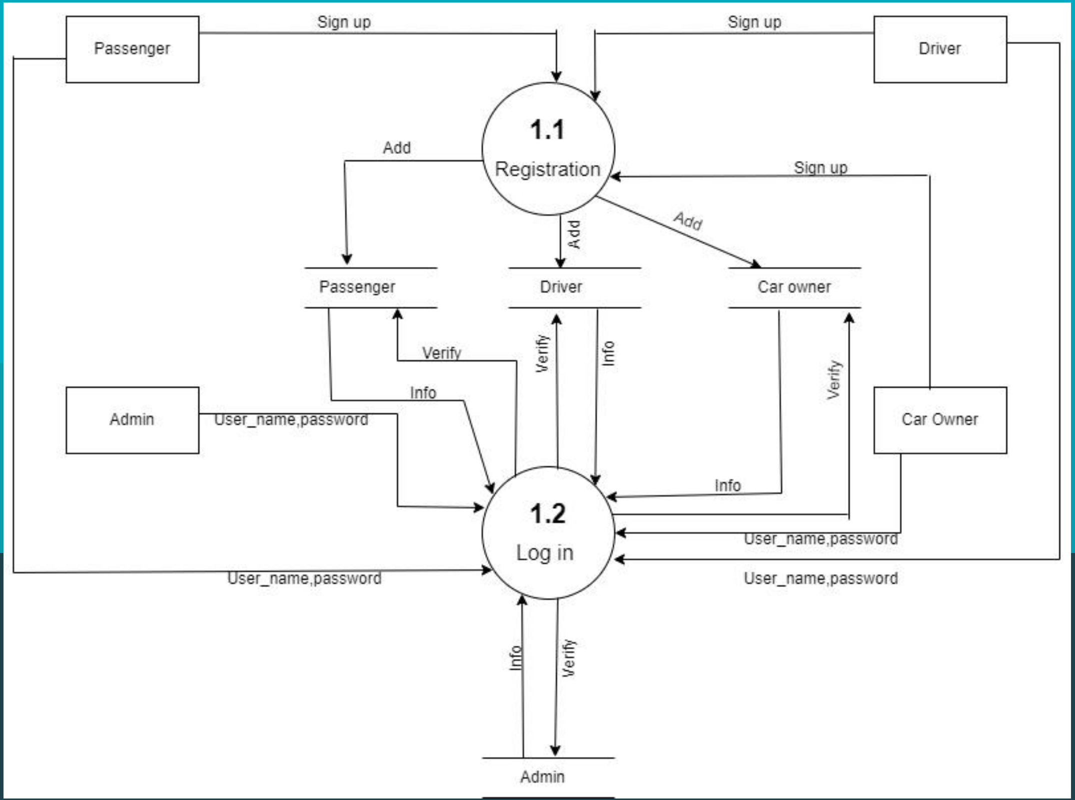
Data Flow Diagram

Level 0 construction diagram:



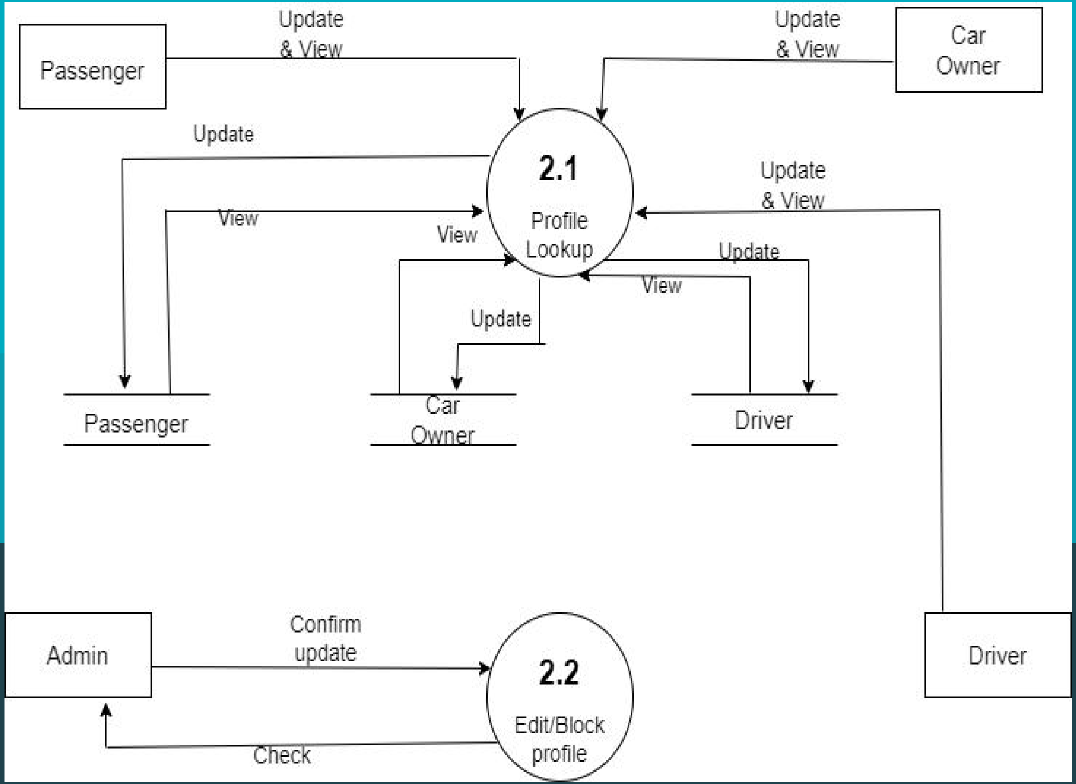
Data Flow Diagram

Level 1-1 diagram:



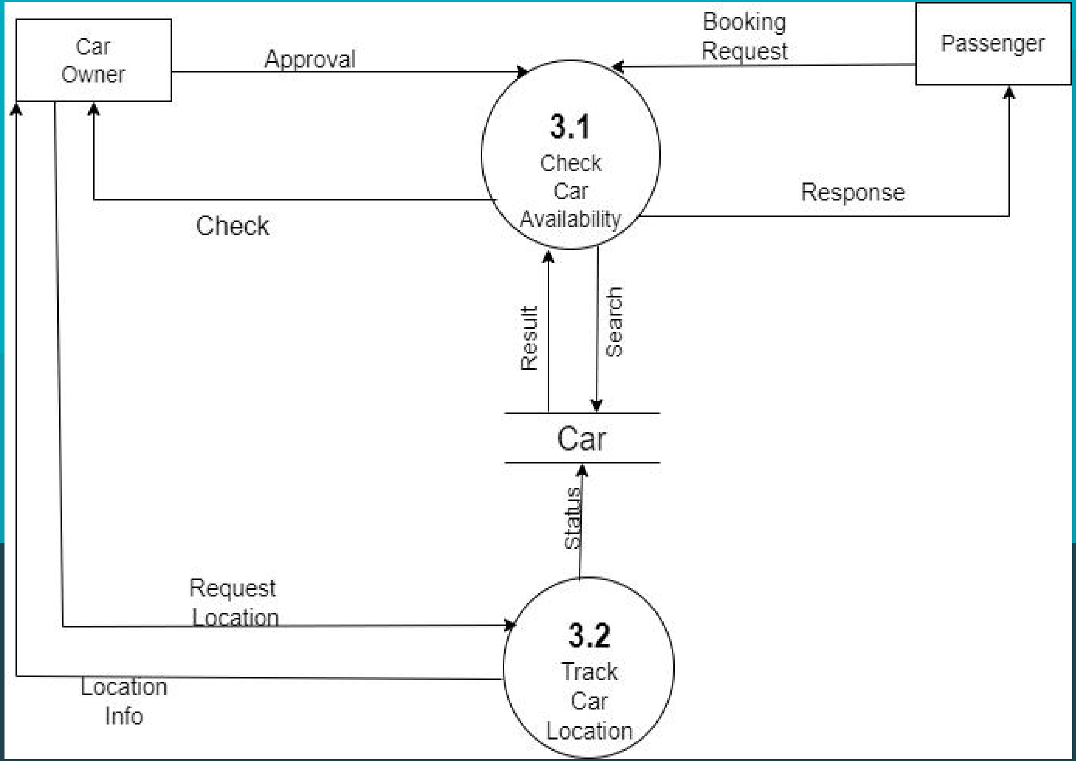
Data Flow Diagram

Level 1-2 diagram:



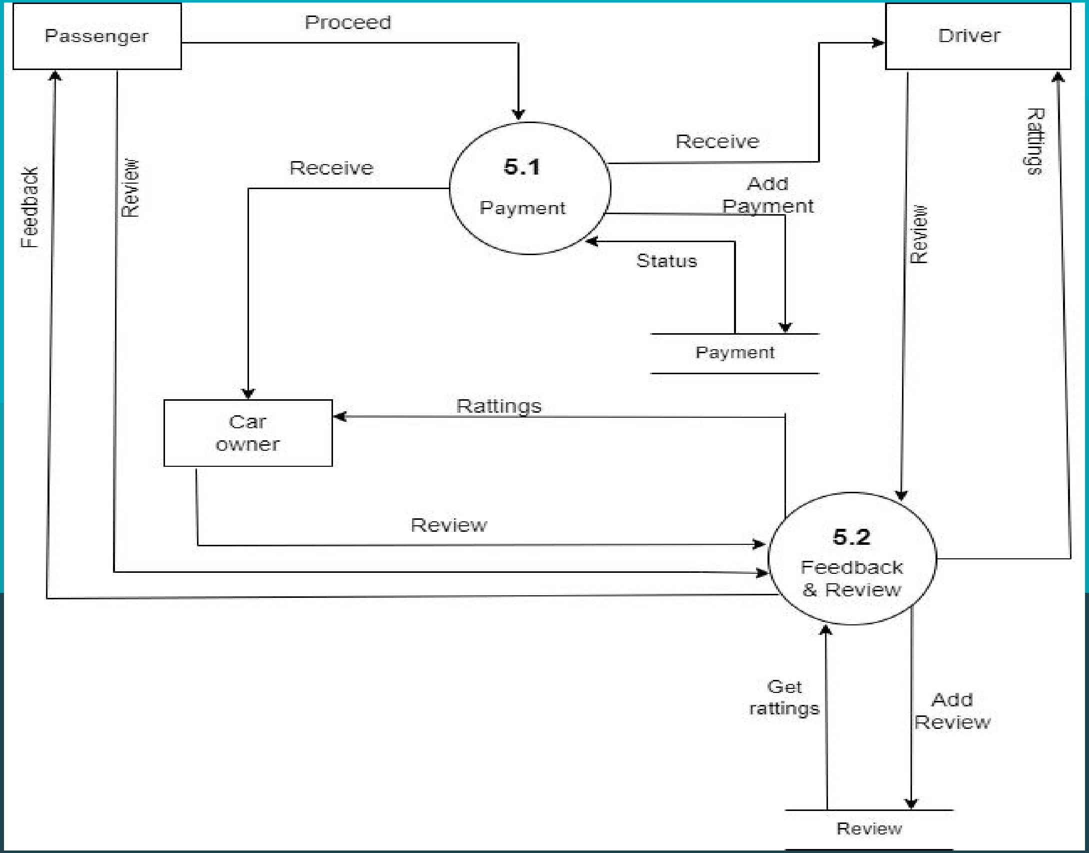
Data Flow Diagram

Level 1-3 diagram:



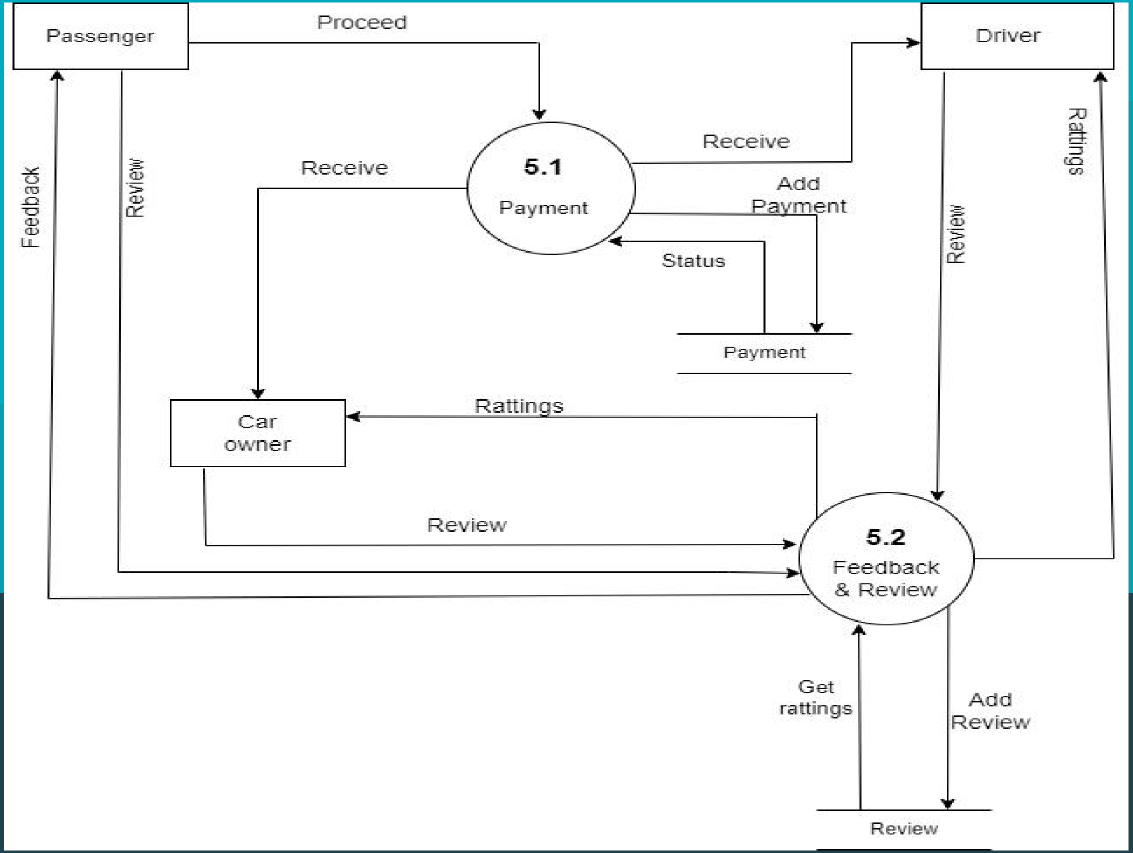
Data Flow Diagram

Level 1-4 diagram:

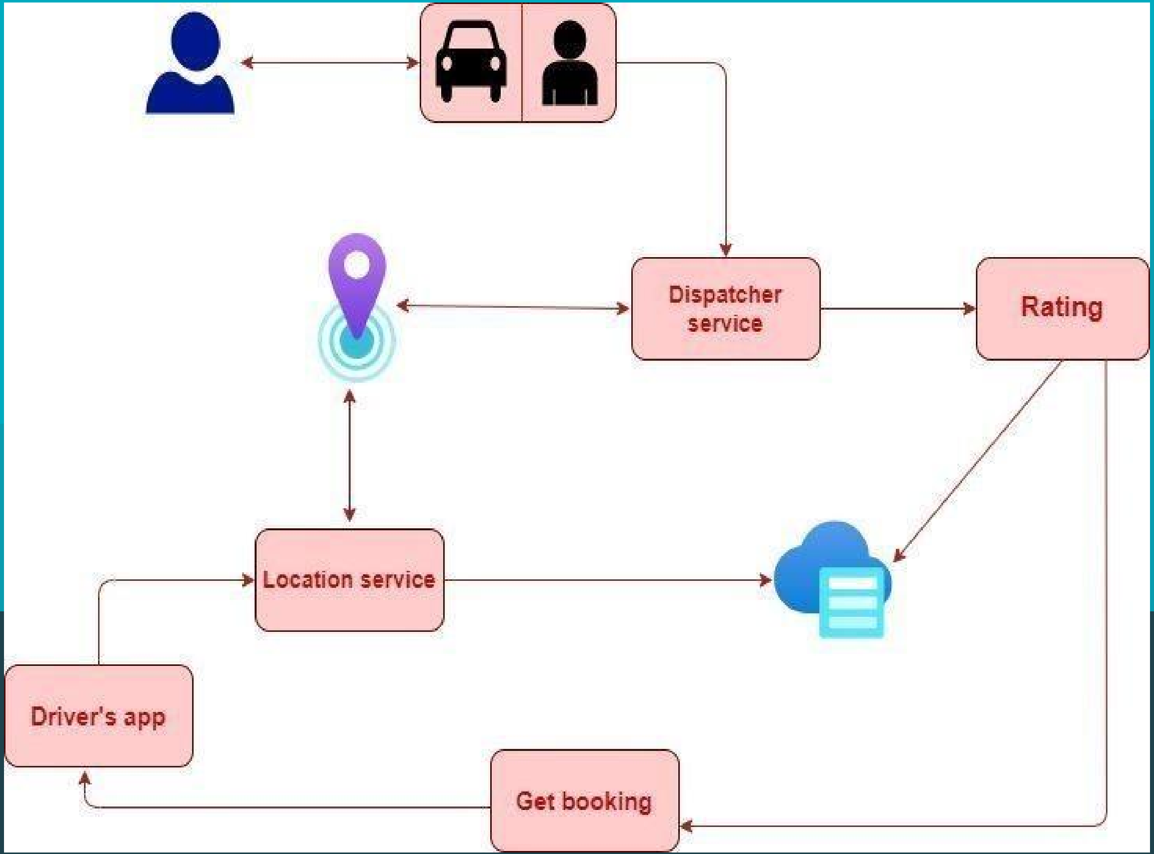


Data Flow Diagram

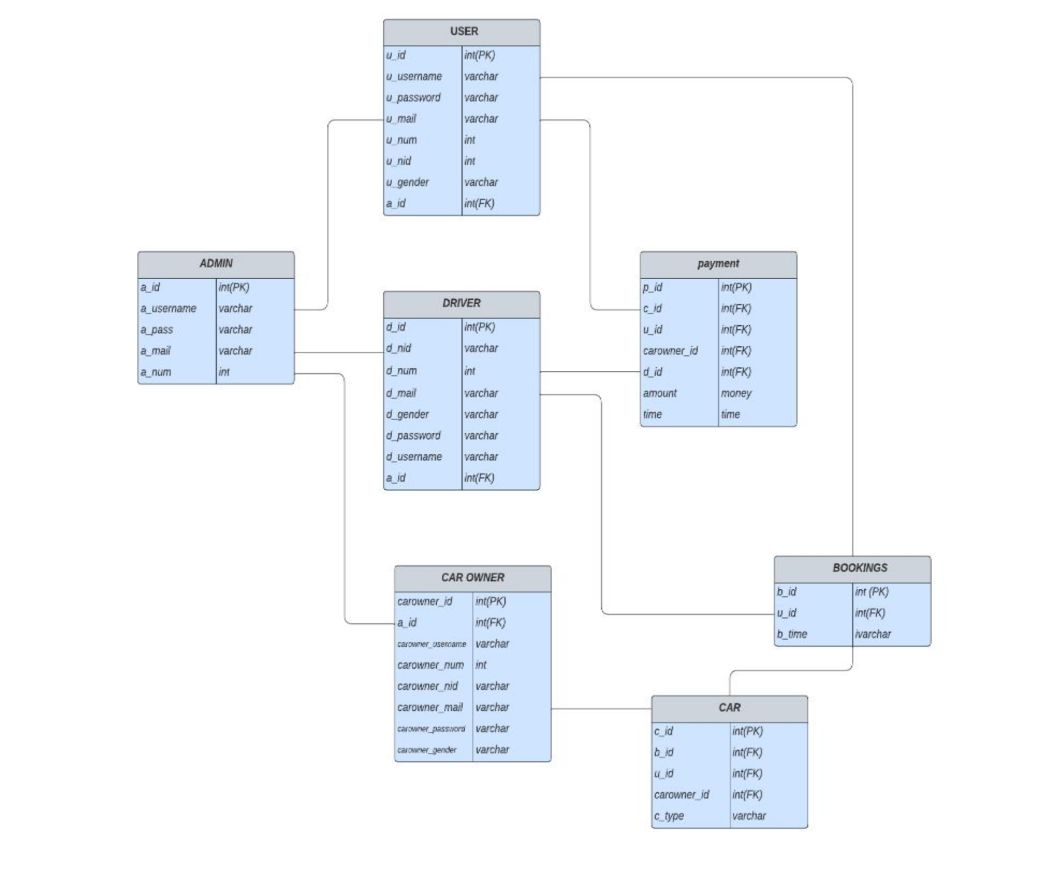
Level 1-5 diagram:



Architecture Flow Diagram



Schema Diagram



USER	
u_id	int(PK)
u_username	varchar
u_password	varchar
u_mail	varchar
u_num	int
u_nid	int
u_gender	varchar
a_id	int(FK)

ADMIN	
a_id	int(PK)
a_username	varchar
a_pass	varchar
a_mail	varchar
a_num	int

DRIVER	
d_id	int(PK)
d_nid	varchar
d_num	int
d_mail	varchar
d_gender	varchar
d_password	varchar
d_username	varchar
a_id	int(FK)

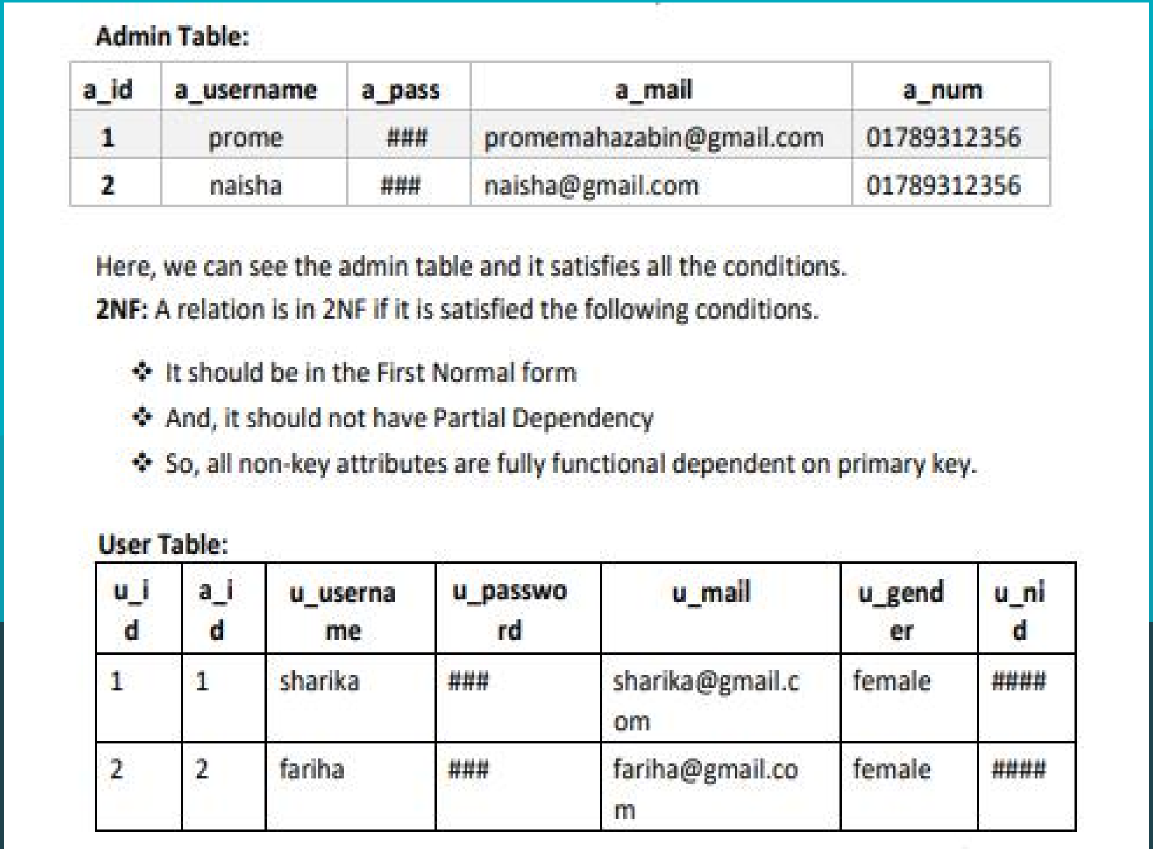
payment	
p_id	int(PK)
c_id	int(FK)
u_id	int(FK)
carowner_id	int(FK)
d_id	int(FK)
amount	money
time	time

BOOKINGS	
b_id	int (PK)
u_id	int(FK)
b_time	ivarchar

CAR OWNER	
carowner_id	int(PK)
a_id	int(FK)
carowner_username	varchar
carowner_num	int
carowner_nid	varchar
carowner_mail	varchar
carowner_password	varchar
carowner_gender	varchar

CAR	
c_id	int(PK)
b_id	int(FK)
u_id	int(FK)
carowner_id	int(FK)
c_type	varchar

Normalization



Admin Table:

a_id	a_username	a_pass	a_mail	a_num
1	prome	###	promemahazabin@gmail.com	01789312356
2	naisha	###	naisha@gmail.com	01789312356

Here, we can see the admin table and it satisfies all the conditions.

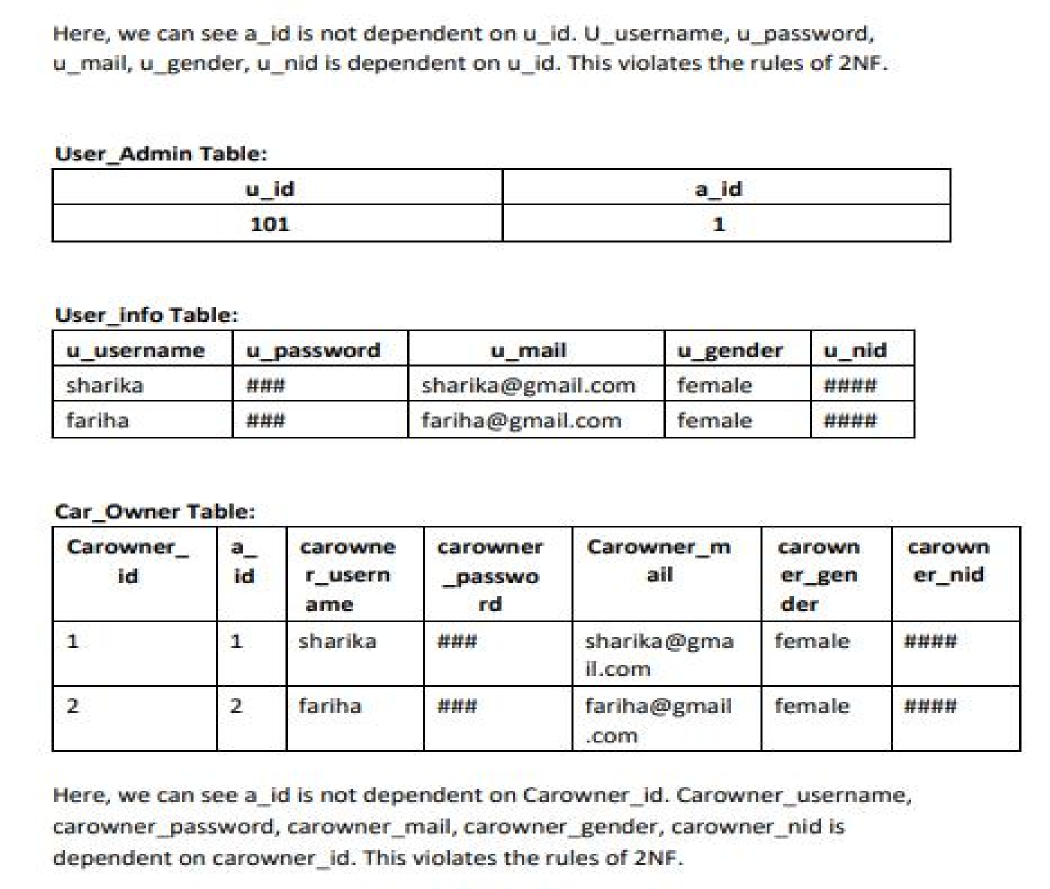
2NF: A relation is in 2NF if it is satisfied the following conditions.

- It should be in the First Normal form
- And, it should not have Partial Dependency
- So, all non-key attributes are fully functional dependent on primary key.

User Table:

u_id	a_id	u_username	u_password	u_mail	u_gender	u_nid
1	1	sharika	###	sharika@gmail.com	female	####
2	2	fariha	###	fariha@gmail.com	female	####

Normalization



Here, we can see a_id is not dependent on u_id. U_username, u_password, u_mail, u_gender, u_nid is dependent on u_id. This violates the rules of 2NF.

User_Admin Table:

u_id	a_id
101	1

User_info Table:

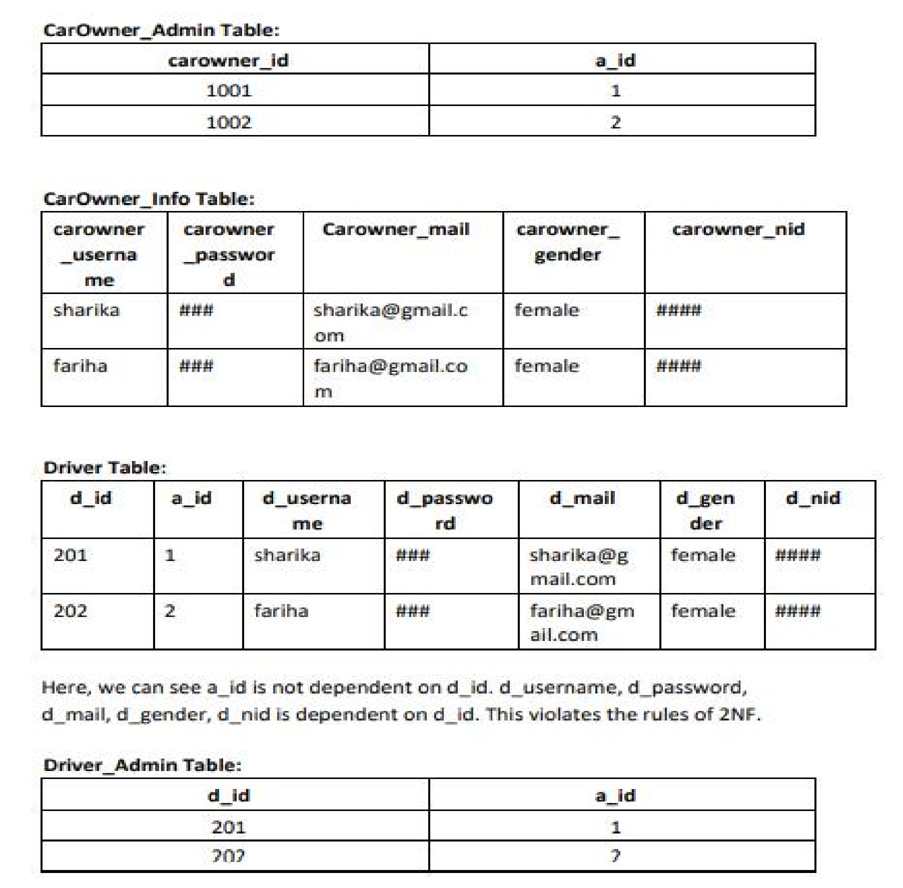
u_username	u_password	u_mail	u_gender	u_nid
sharika	###	sharika@gmail.com	female	####
fariha	###	fariha@gmail.com	female	####

Car_Owner Table:

Carowner_id	a_id	carowner_username	carowner_password	Carowner_mail	carowner_gender	carowner_nid
1	1	sharika	###	sharika@gmail.com	female	####
2	2	fariha	###	fariha@gmail.com	female	####

Here, we can see a_id is not dependent on Carowner_id. Carowner_username, carowner_password, carowner_mail, carowner_gender, carowner_nid is dependent on carowner_id. This violates the rules of 2NF.

Normalization



CarOwner_Admin Table:

carowner_id	a_id
1001	1
1002	2

CarOwner_Info Table:

carowner_username	carowner_password	Carowner_mail	carowner_gender	carowner_nid
sharika	###	sharika@gmail.com	female	####
fariha	###	fariha@gmail.com	female	####

Driver Table:

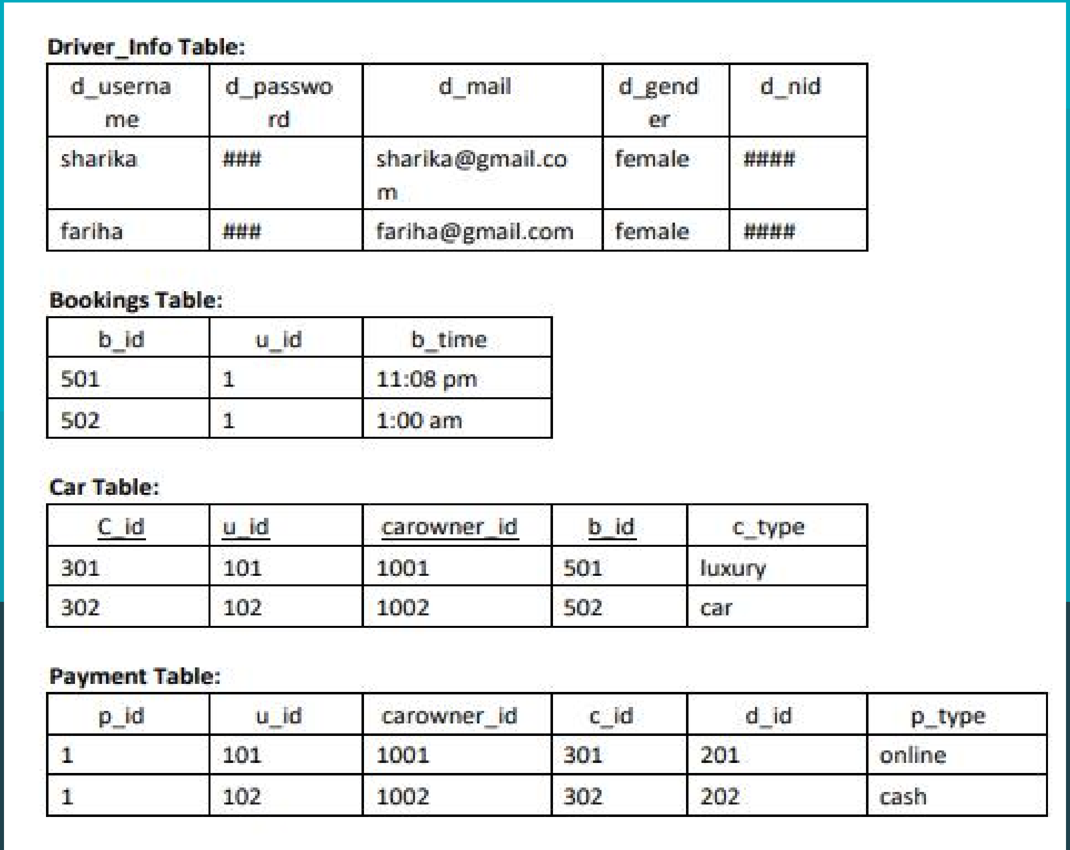
d_id	a_id	d_username	d_password	d_mail	d_gender	d_nid
201	1	sharika	###	sharika@gmail.com	female	####
202	2	fariha	###	fariha@gmail.com	female	####

Here, we can see a_id is not dependent on d_id. d_username, d_password, d_mail, d_gender, d_nid is dependent on d_id. This violates the rules of 2NF.

Driver_Admin Table:

d_id	a_id
201	1
202	2

Normalization



Driver_Info Table:

d_username	d_password	d_mail	d_gender	d_nid
sharika	###	sharika@gmail.com	female	####
fariha	###	fariha@gmail.com	female	####

Bookings Table:

b_id	u_id	b_time
501	1	11:08 pm
502	1	1:00 am

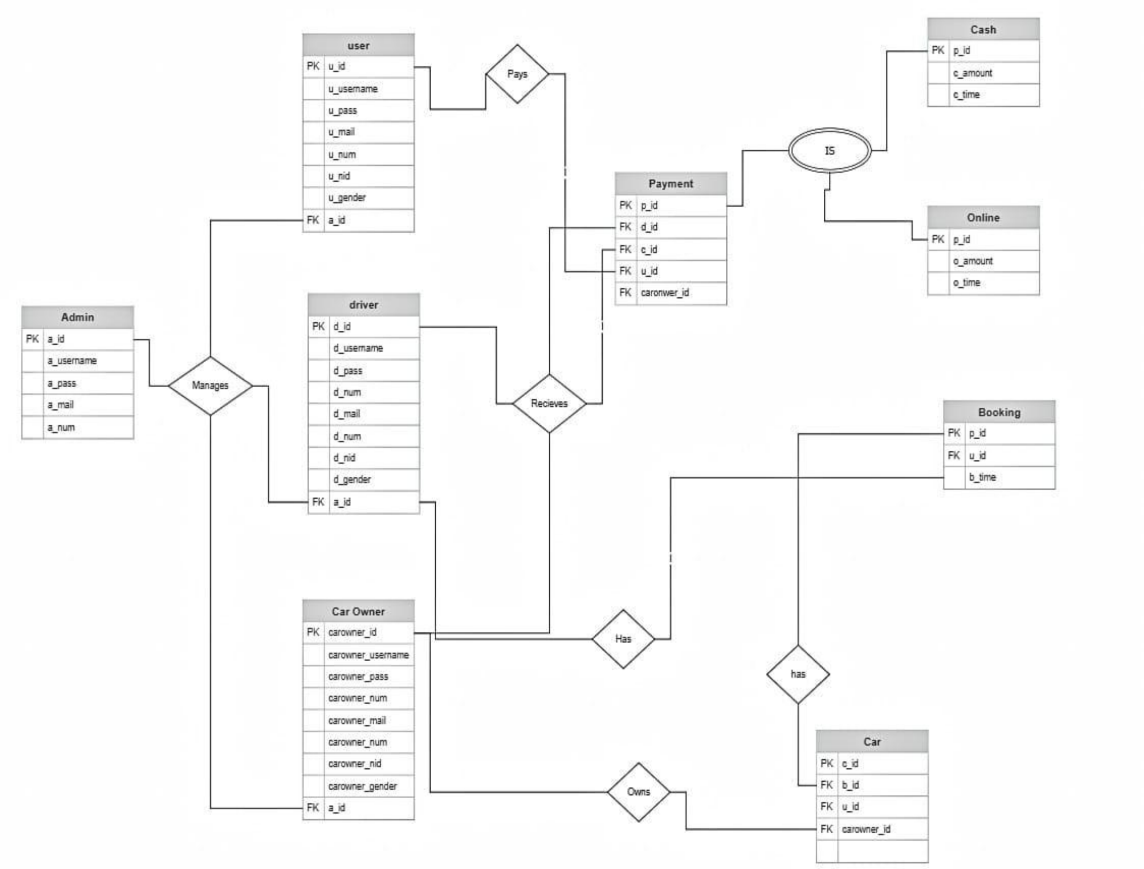
Car Table:

C_id	u_id	carowner_id	b_id	c_type
301	101	1001	501	luxury
302	102	1002	502	car

Payment Table:

p_id	u_id	carowner_id	c_id	d_id	p_type
1	101	1001	301	201	online
1	102	1002	302	202	cash

Normalization



	user
PK	u_id
	u_username
	u_pass
	u_mail
	u_num
	u_nid
	u_gender
FK	a_id

	Cash
PK	p_id
	c_amount
	c_time

	Payment
PK	p_id
FK	d_id
FK	c_id
FK	u_id
FK	caronwer_id

	Online
PK	p_id
	o_amount
	o_time

	Admin
PK	a_id
	a_username
	a_pass
	a_mail
	a_num

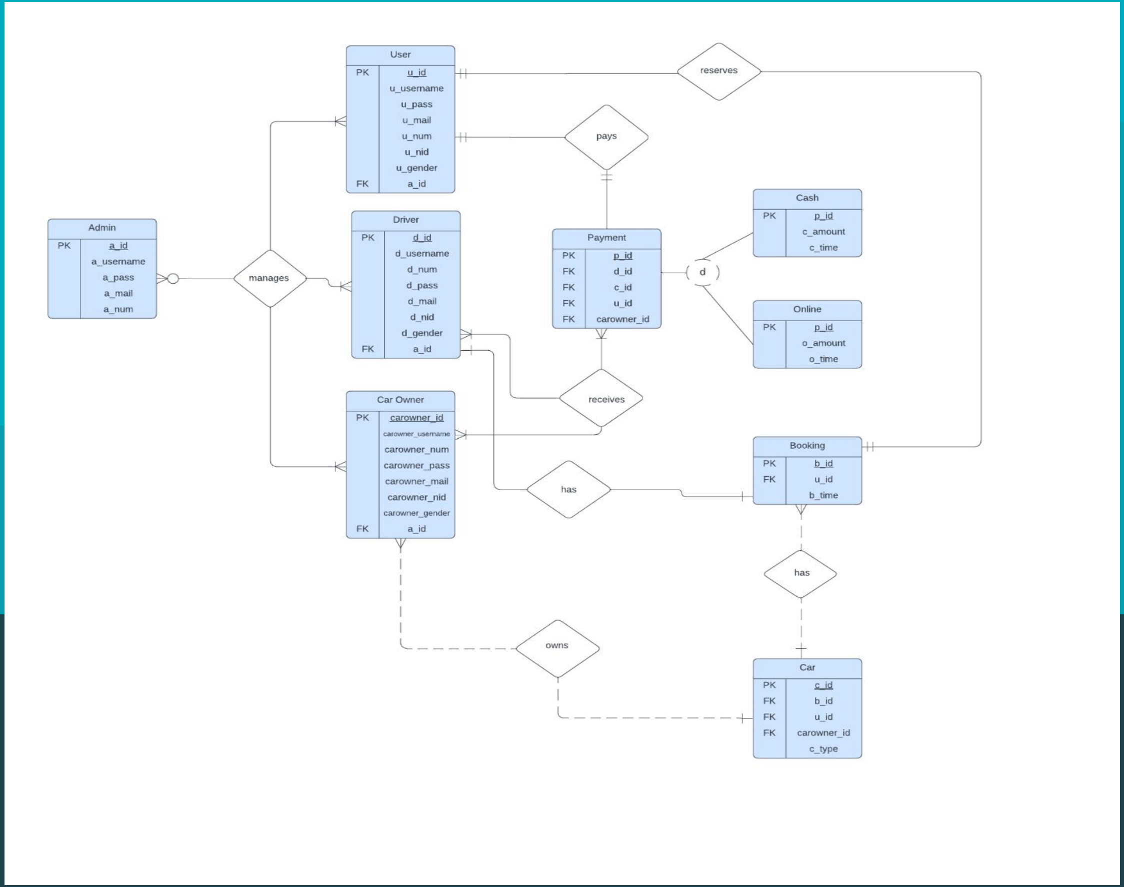
	driver
PK	d_id
	d_username
	d_pass
	d_num
	d_mail
	d_num
	d_nid
	d_gender
FK	a_id

	Booking
PK	p_id
FK	u_id
	b_time

	Car Owner
PK	carowner_id
	carowner_username
	carowner_pass
	carowner_num
	carowner_mail
	carowner_num
	carowner_nid
	carowner_gender
FK	a_id

	Car
PK	c_id
FK	b_id
FK	u_id
FK	carowner_id

ERD



User	
PK	u_id
	u_username
	u_pass
	u_mail
	u_num
	u_nid
	u_gender
FK	a_id

Admin	
PK	a_id
	a_username
	a_pass
	a_mail
	a_num

Driver	
PK	d_id
	d_username
	d_num
	d_pass
	d_mail
	d_nid
	d_gender
FK	a_id

Payment	
PK	p_id
FK	d_id
FK	c_id
FK	u_id
FK	carowner_id

Cash	
PK	p_id
	c_amount
	c_time

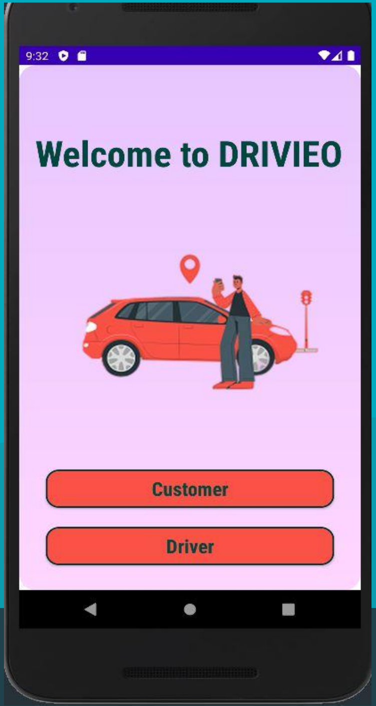
Online	
PK	p_id
	o_amount
	o_time

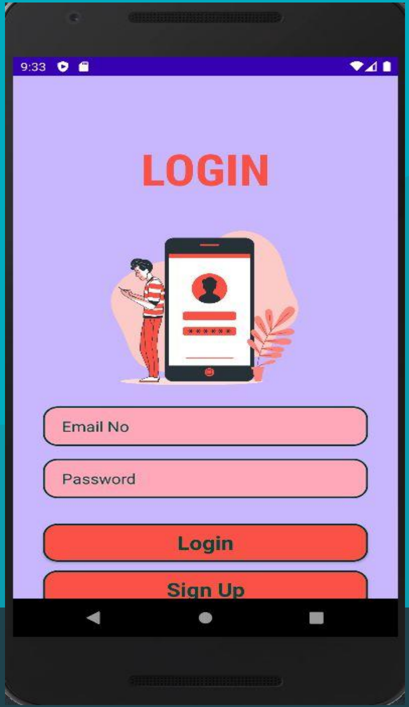
Car Owner	
PK	carowner_id
	carowner_username
	carowner_num
	carowner_pass
	carowner_mail
	carowner_nid
	carowner_gender
FK	a_id

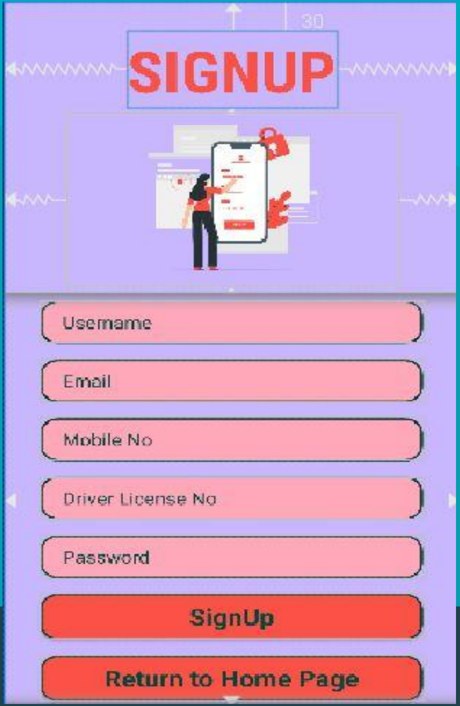
Booking	
PK	b_id
FK	u_id
	b_time

Car	
PK	c_id
FK	b_id
FK	u_id
FK	carowner_id
	c_type

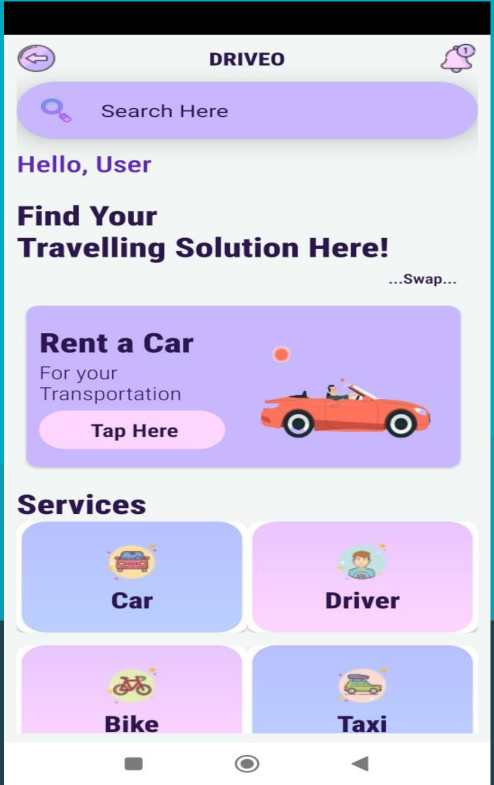
Ui/Ux Design


9:32
Welcome to DRIVIEO
Customer
Driver


9:33
LOGIN
Email No
Password
Login
Sign Up


30
SIGNUP
Username
Email
Mobile No
Driver License No
Password
SignUp
Return to Home Page

Ui/Ux Design



DRIVEO

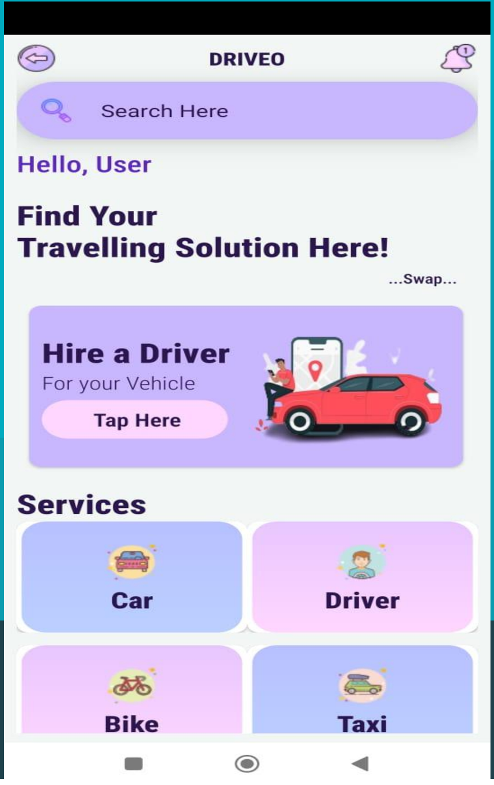
Search Here

Hello, User

Find Your
Travelling Solution Here!

...Swap...

Rent a Car
For your Transportation



DRIVEO

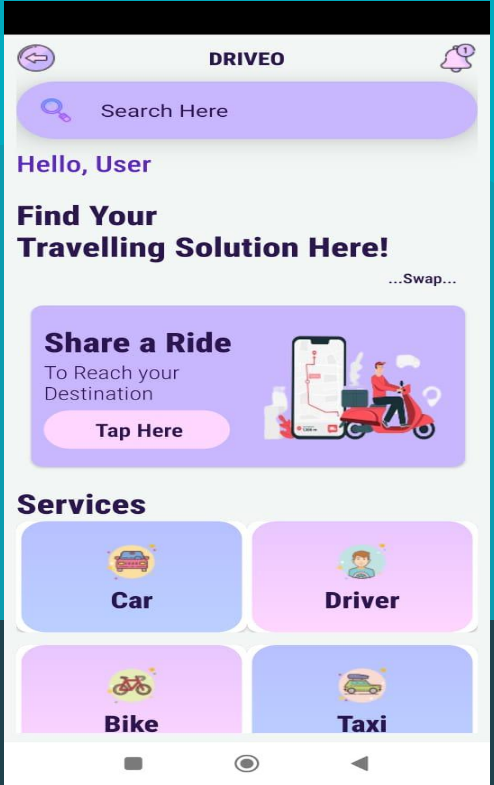
Search Here

Hello, User

Find Your
Travelling Solution Here!

...Swap...

Hire a Driver
For your Vehicle



DRIVEO

Search Here

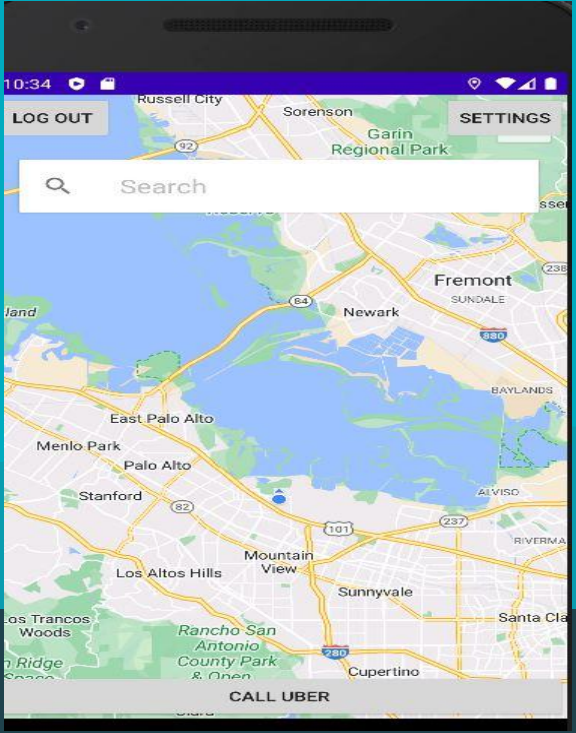
Hello, User

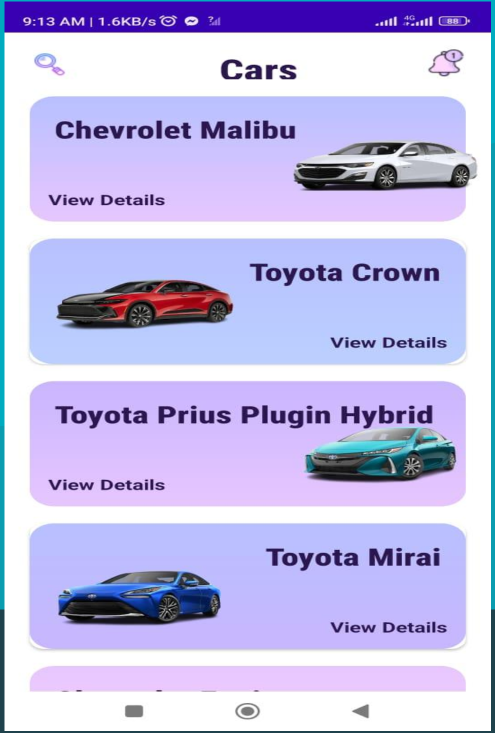
Find Your
Travelling Solution Here!

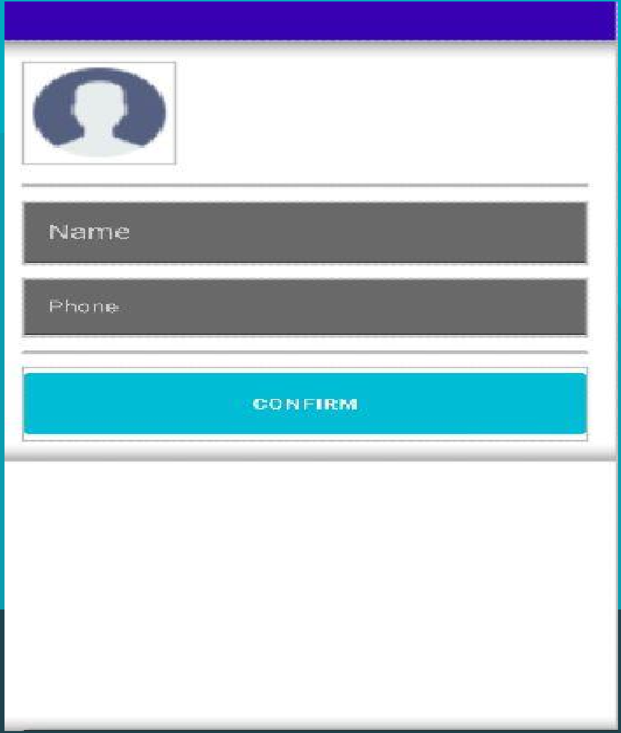
...Swap...

Share a Ride
To Reach your Destination

Ui/Ux Design


10:34
LOG OUT
SETTINGS
Search
Russell City
Sorenson
Garin Regional Park
Fremont
SUNDALE
Newark
BAYLANDS
East Palo Alto
Menlo Park
Palo Alto
Stanford
ALVISO
Mountain View
Los Altos Hills
Sunnyvale
Santa Cla
Los Trancos Woods
Rancho San Antonio County Park & Open
Cupertino
CALL UBER


9:13 AM | 1.6KB/s
Cars
Chevrolet Malibu
View Details
Toyota Crown
View Details
Toyota Prius Plugin Hybrid
View Details
Toyota Mirai
View Details


Name
Phone
CONFIRM

Conclusion:

Our app will make travelling easier. We won't have to worry about transportation. We won't have to worry if our car driver doesn't come. We can simply login to this app and call our desired transportation or driver anytime and anywhere. And not only this, we can also earn some money by registering our cars to the system or by registering as a driver.

Thank you!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

Any Question?