DBS PROJECT REPORT

Title:

“CYCLIFY, rent a bike!”

Team Members: CSE B, Batch 1

Pranav Kulkarni	Roll no. 14	Reg no. 170905078
Manan Shah	Roll no. 15	Reg no. 170905080

Abstract:

MIT, Manipal is an engineering college in Karnataka. It has a significant student strength of 9000. On the weekends and holidays, students often plan to explore famous spots around here. There exist motorbike and car renting businesses near the campus which are suitable for longer trips, for example, to Mangalore.

However, there exist many scenic spots in and around Manipal itself which are away from the highways and the main roads, places where a motorbike can't be driven to. We have observed that there is a considerable demand to rent bicycles on the holidays. However, there are no source which offers bicycles for rent in Manipal or Udupi.

Among the 9000 MIT students, there are around 500 who own and use their bicycles on campus. During the weekends and holidays, we estimate at least half of these 500 cycles (around 250) just sit unused as the owners are busy with their academic or club work.

Using our application, these owners can post their cycles for rent whenever they do not foresee themselves using them. They can give details about their bikes. They can post a minimum rate per hour that they are willing to accept.

The prospective renters can input a date when they want a cycle. They will then get a list of cycles available, from which they can then make a choice.

Our solution solves the problem of the absence of a cycle-renting shop. It also provides a small income source to the bike owners.

Goal:

We aim to connect the bike owner with the prospective bike renter.

Problem Statement:

The data required for us to achieve our goal would be the user details. For both, the **owners and renters**, we will need their **contact information**, **registration number**, **names** and their **login details**. Moreover, for owners, we will need the data concerning what is the **make of their cycle**, their **cycle id**, registered with security section, the **color** and some other specifications. Next, we will need the data when each particular **cycle is available** to be rented, how much the owner wants to **charge** and **till what time** it is free. Furthermore, we need to have a table to keep all possible **timeslots** when the cycles could be rented. Lastly, we need a record of all the **transactions** that have taken place using our app.

The functional requirements would be that when the owner enters a cycle as available for a particular day or time, any renter is able to see the available cycle for the day and able to select whichever one he wants, once selected the renter will get owner's number and contact him and borrow the cycle. Hence this way, we ensure that the owner's free cycle is given to someone and the renter also gets to use a cycle for the time being.

Sample Data

owners -

170905078 'pranav kulkarni' 9741108130 'room 302, block 8, mit'
170905080 'manan shah' 9824060542 'room 626, block 14, mit'
170905011 'teja setia' 9000010011 'room 501, block 23, mit'
170905012 'sumit agarwal' 9000100012 'room 501, block 23, mcods'
170905013 'ishan pitty' 9000100013 'room 502, block 23, doc'
170905014 'tanya sharma' 9000100014 'room 502, block 23, des'
170905015 'ronson lobo' 9000100015 'room 503, block 23, icas'
170905016 'riya subedar' 9000100016 'room 503, block 23, mcon'
170905017 'syed nabil' 9000100017 'room 504, block 23, mit'
170905018 'ashwin mathur' 9000100018 'room 504, block 23, mcops'
170905019 'sarthak arora' 9000100019 'room 505, block 23, mcods'
170905020 'ansh raj' 9000100020 'room 505, block 23, mit'

renters -

170905000 'ankit baral' 9000000000 'room 420, block 13, soc'
170905011 'aniket baral' 9000000001 'room 421, block 13, soc'
170905001 'mehul garg' 9000100001 'room 420, block 13, kmc'
170905002 'ojas shah' 9000100002 'room 420, block 13, mcops'
170905003 'rajat dumir' 9000100003 'room 420, block 13, doc'
170905004 'harsh dubey' 9000100004 'room 420, block 13, mcods'
170905005 'rehan jain' 9000100005 'room 420, block 13, wgsha'
170905006 'rhea mehra' 9000100006 'room 420, block 13, mit'
170905007 'aditi mittal' 9000100007 'room 420, block 13, sca'
170905008 'arib ali' 9000100008 'room 420, block 13, tapmi'
170905009 'rahul shaw' 9000100009 'room 420, block 13, des'
170905010 'pranav jain' 9000100010 'room 420, block 13, foa'

cycles -

'sdt-101' 'hercules a75' 'blue' 'AUG' 1 30 170905012
'sdt-102' 'hercules a50' 'red' 'FEB' 0 32 170905011
'sdt-103' 'montra climber' 'black' 'JUL' 1 34 170905013
'sdt-104' 'montra rockrider' 'green' 'MAR' 1 36 170905014
'sdt-105' 'hercules a75' 'orange' 'APR' 1 38 170905015
'sdt-106' 'hercules a50' 'pink' 'DEC' 0 40 170905016
'sdt-107' 'hero streamliner' 'purple' 'SEP' 0 42 170905017
'sdt-108' 'mercedes a65' 'white' 'MAY' 1 44 170905078
'sdt-109' 'bmw b75' 'brown' 'JUN' 1 46 170905019
'sdt-110' 'hercules a50' 'violet' 'NOV' 0 48 170905017

timeslot -
1904011200 '04-01-2019 12:00'
1904011300 '04-01-2019 13:00'
1904011400 '04-01-2019 14:00'
1904011500 '04-01-2019 15:00'
1904011600 '04-01-2019 16:00'
1904011700 '04-01-2019 17:00'
1904021200 '04-02-2019 12:00'
1904021300 '04-02-2019 13:00'
1904021400 '04-02-2019 14:00'
1904021500 '04-02-2019 15:00'
1904021600 '04-02-2019 16:00'
1904021700 '04-02-2019 17:00'

available -
'sdt-101' 1904011300 1904011400
'sdt-102' 1904011400 1904011700
'sdt-103' 1904011400 1904011600
'sdt-104' 1904011600 1904011700
'sdt-105' 1904011500 1904011700

transcation -

170905001 'sdt-102' 1904011400 1904011600 64
170905007 'sdt-103' 1904011500 1904011600 34
170905011 'sdt-104' 1904011600 1904011700 36

DDL Commands

```
create table owners
(
	regno numeric (9),
	name varchar2 (20),
	phoneno numeric (14),
	address varchar2 (50),
	primary key (regno)
);

create table renters
(
	regno numeric (9),
	name varchar2 (20),
	phoneno numeric (14),
	address varchar2 (50),
	primary key (regno)
);



create table cycles
(
	cycleid varchar2 (8),
	make varchar2 (50),
	colour varchar2 (20),
	monthofpurchase numeric (4),
	--MMYY
	gearednotgeared numeric (1),
	rate numeric (4),
	ownerid numeric (9) references owners on delete cascade,
	primary key (cycleid)
);


create table timeslot
(
	tid numeric (10),
	--YYMMDDHHMinMin
	dateandtime timestamp,
	primary key(tid)
);

create table available
(
	cycleid varchar2 (8) references cycles on delete cascade,
	starttid numeric (10) references timeslot on delete cascade,
	endtid numeric (10) references timeslot on delete cascade,
	primary key (cycleid,starttid,endtid)
);

create table transaction
(
	renterid numeric (9) references renters on delete cascade,
	cycleid varchar2 (8) references cycles on delete cascade,
	starttid numeric (10) references timeslot on delete cascade,
```

```
        endtid numeric (10) references timeslot on delete cascade,
        amount numeric (5),
        primary key (renterid,cycleid,starttid)
);

create table ownerslogin
(
        regno numeric (9) references owners on delete cascade,
        pass varchar2 (30),
        primary key (regno)
);

create table renterslogin
(
        regno numeric (9) references renters on delete cascade,
        pass varchar2 (30),
        primary key (regno)
);

create table owners_old_data
(
        time_of_delete timestamp,
        regno numeric (9),
        name varchar2 (20),
        phoneno numeric (14),
        address varchar2 (50)
);



CREATE OR REPLACE trigger owners_trigger
before delete on owners

FOR EACH ROW
BEGIN

        insert into owners_old_data values
        (
                systimestamp,
                :old.regno,
                :old.name,
                :old.phoneno,
                :old.address
        );

END;
/


create table renters_old_data
(
        time_of_delete timestamp,
        regno numeric (9),
        name varchar2 (20),
        phoneno numeric (14),
        address varchar2 (50)
);

CREATE OR REPLACE trigger renters_trigger
```

```
before delete on renters

FOR EACH ROW
BEGIN

	insert into renters_old_data values
	(
		systimestamp,
		:old.regno,
		:old.name,
		:old.phoneno,
		:old.address
	);

END;
/

create or replace function getTS (day number, month number, year number, t1 number)
return number as tot number;
begin
--YYMMDDHHMinMin

tot:=0+year-2000; tot:=tot*100;
tot:=tot+month; tot:=tot*100;
tot:=tot+day; tot:=tot*100;
tot:=tot+t1; tot:=tot*100;
return tot;
end;
/

/*
DECLARE
BEGIN
	DBMS_OUTPUT.PUT_LINE(getTS(12,11,2045,20));
END;
/
```

List of Queries used

Login page:

- "SELECT * FROM ownerslogin where regno=" + regno
- "SELECT * FROM renterslogin where regno=" + regno

Sign Up page:

- "insert into " + acctype + "s values (" + regno + ",'" +name + "'," + phno + ",'" + addr + "')"
- "insert into " + acctype + "slogin values (" + regno + ",'" +pass + "')"

Renters page:

- "SELECT * FROM renters where regno=" + OID

Renter Remove Account:

- "delete from renterslogin where regno = " + OID
- "delete from renters where regno = " + OID

Browse Cycle:

- "SELECT * FROM owners where regno=" + OID

Remove Cycle:

- "delete from transactions where cycleid = " + CID
- "delete from owners where regno = " + OID

Owner Remove Account

- "delete from ownerslogin where regno = " + OID
- "delete from owners where regno = " + OID

Add Cycle:

- "insert into cycles values ('" + CID + "','" + Make + "','" + Color + "'," + mop + "," + gear + "," + rate + "," + OID + ")"

Java Code for functional design (DB connectivity and access)

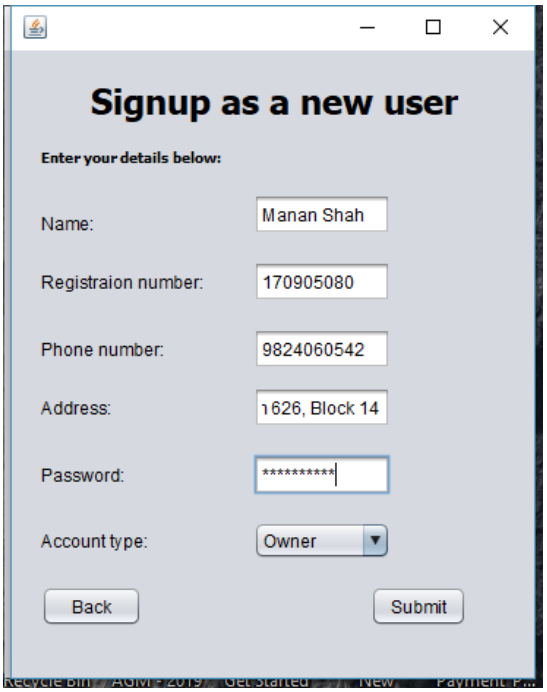
For connection of DB and Front End:

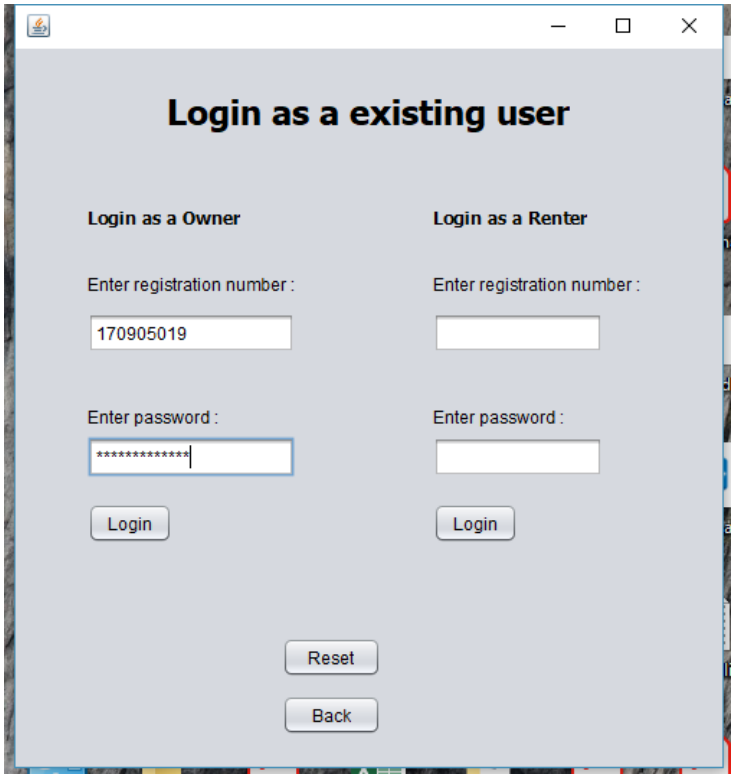
```
  Class.forName("oracle.jdbc.driver.OracleDriver");

      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost", "system",
"prajant");

      Statement stmt = con.createStatement();
```

Back Button for every screen:

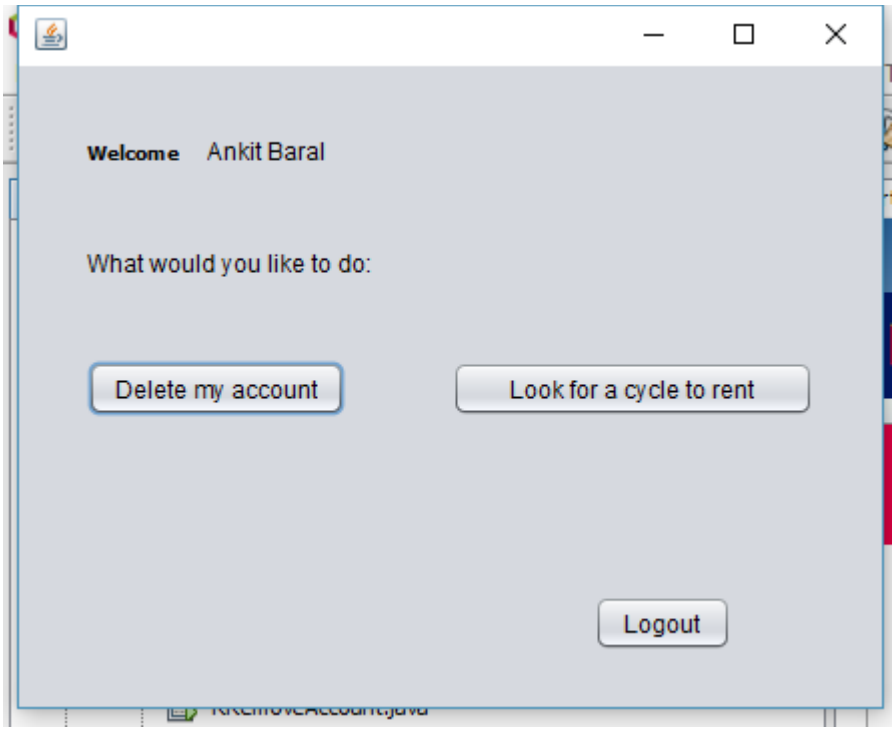
```
private void backbuttonActionPerformed(java.awt.event.ActionEvent evt) {

      loginorsignup ob = new loginorsignup();

      ob.setVisible(true);

      this.setVisible(false);

   }
```

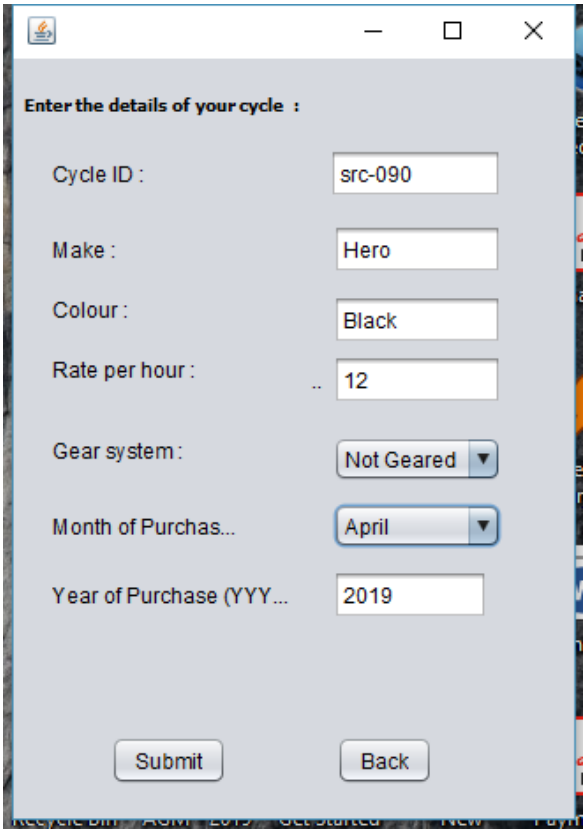
UI –Screenshots

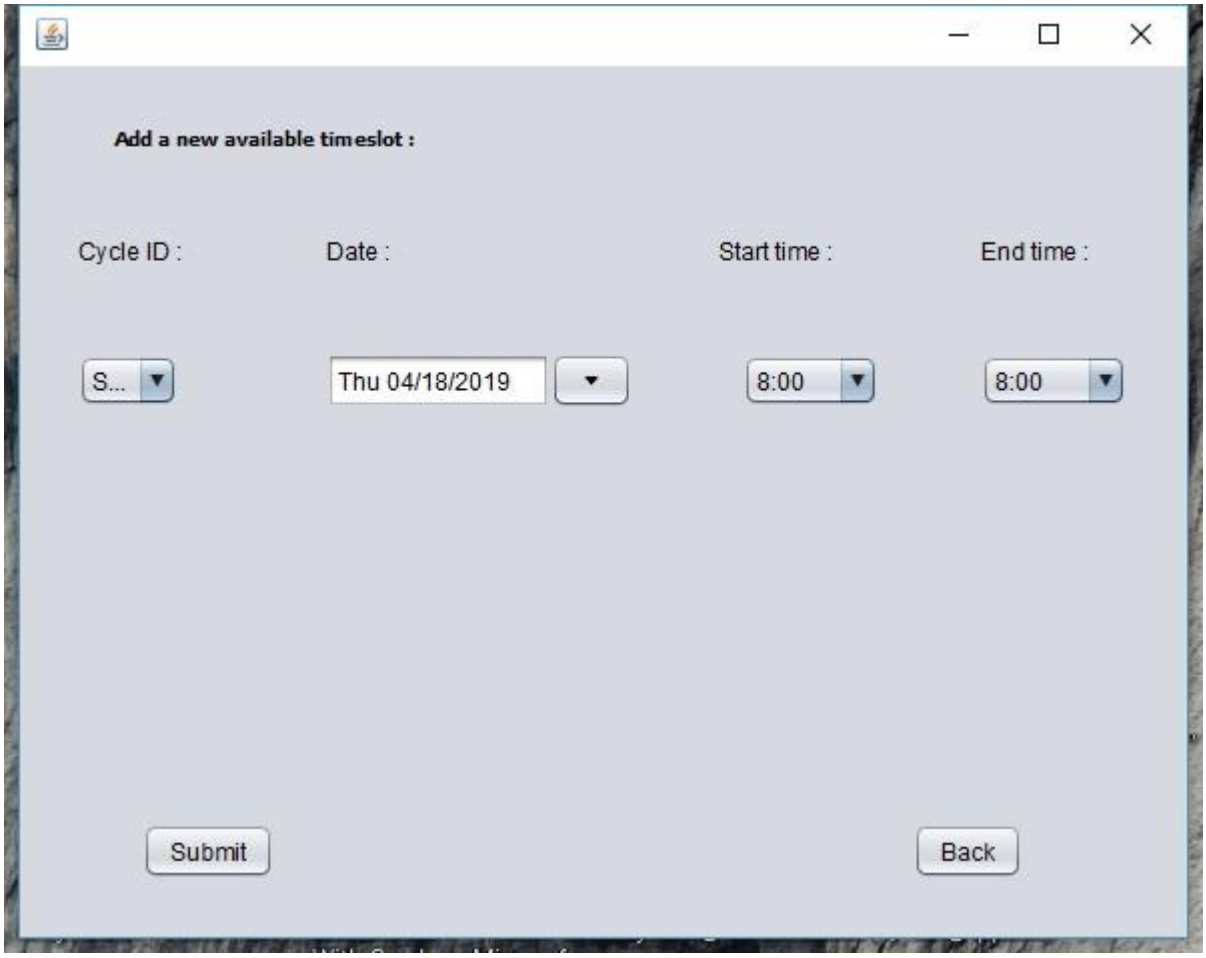


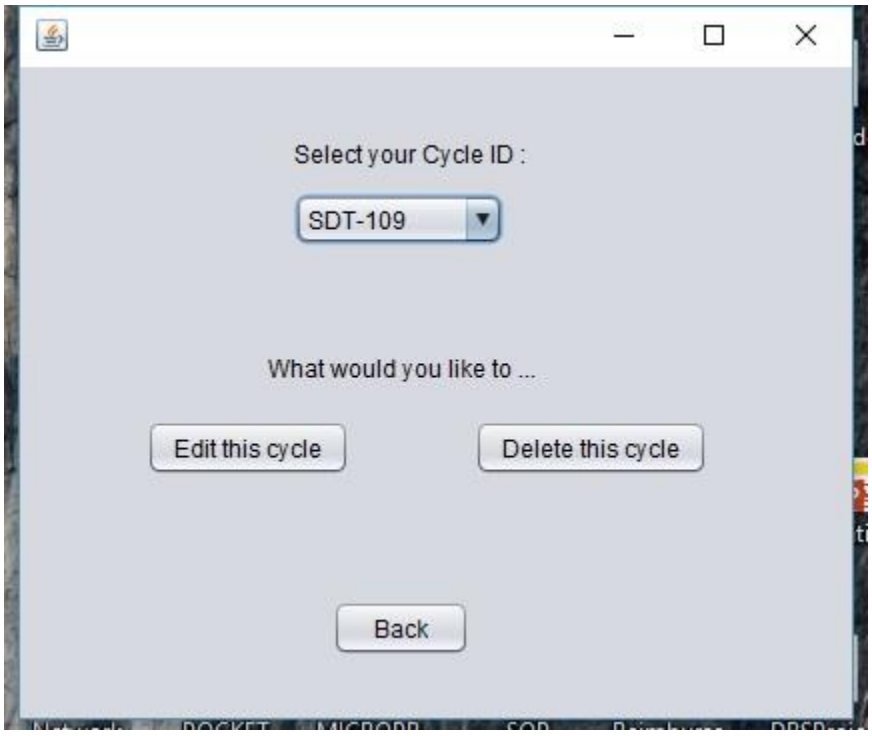

Login as a existing user
Login as a Owner
Login as a Renter
Enter registration number :
170905019
Enter registration number :
Enter password :
Enter password :
Login
Login
Reset
Back

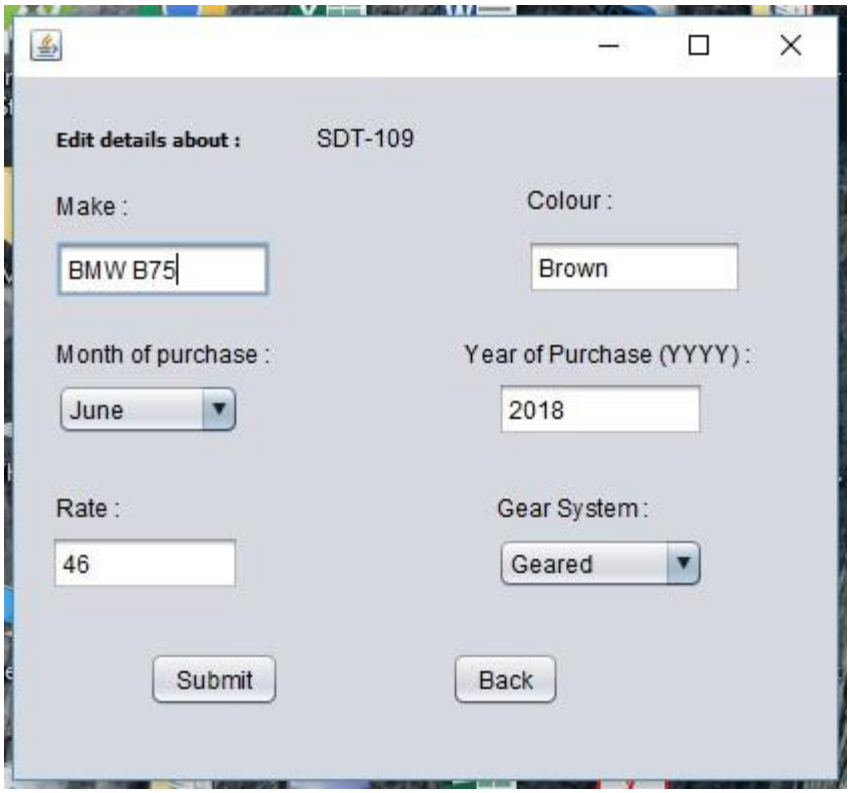

Welcome Sarthak Arora
What would you like to do :
Add a new cycle
Edit or remove an existing cycle
Add a new available timeslot
Delete my account
Logout

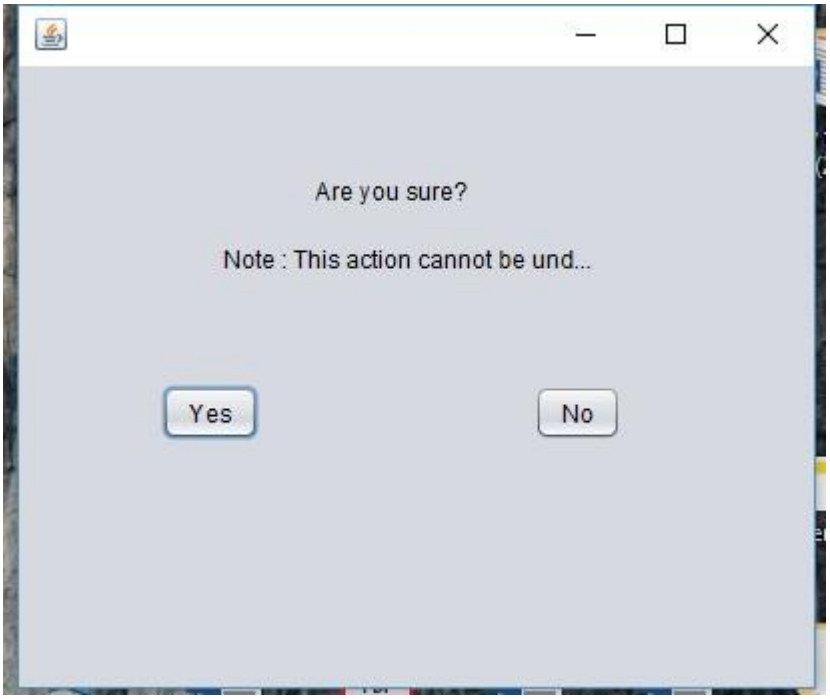

Welcome Ankit Baral
What would you like to do:
Delete my account
Look for a cycle to rent
Logout

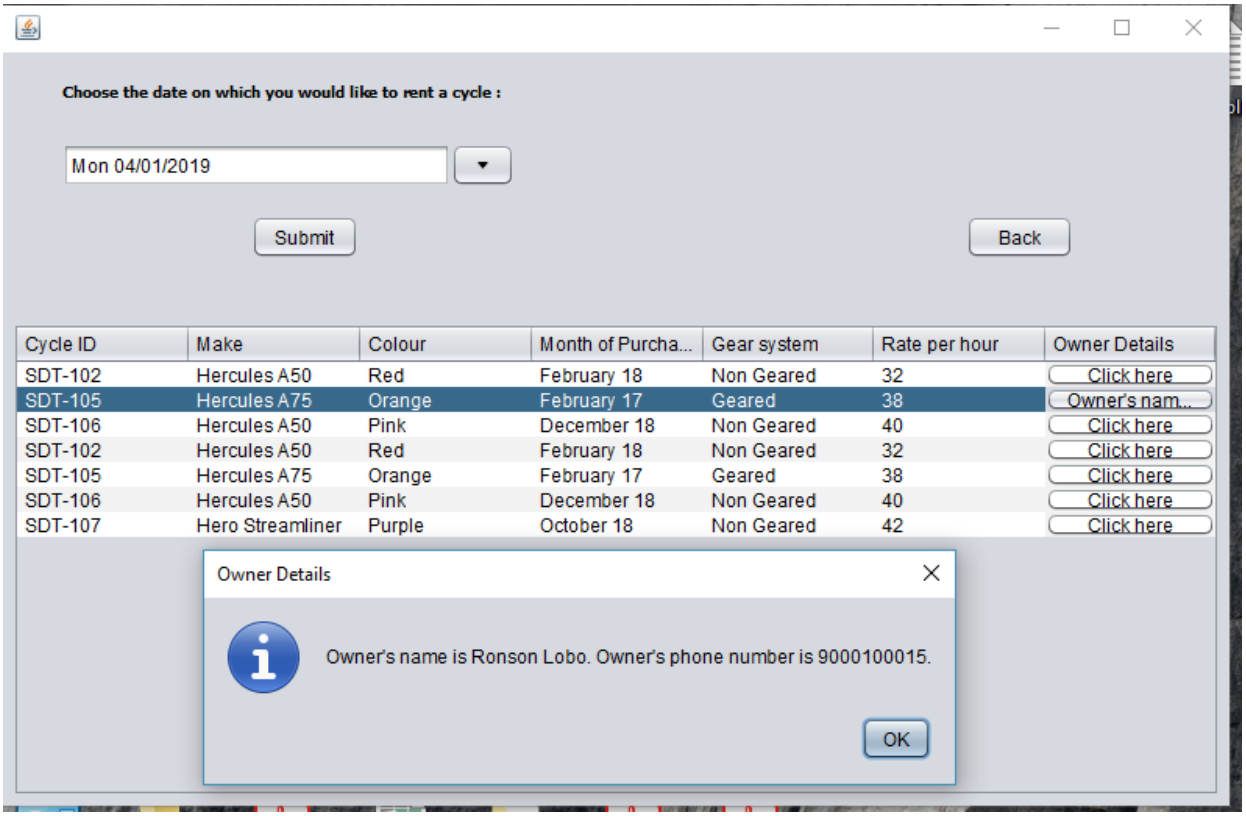

Enter the details of your cycle :
Cycle ID :
src-090
Make :
Hero
Colour :
Black
Rate per hour :
12
Gear system :
Not Geared
Month of Purchas...
April
Year of Purchase (YYY...
2019
Submit
Back


Add a new available timeslot :
Cycle ID :
Date :
Start time :
End time :
S...
Thu 04/18/2019
8:00
8:00
Submit
Back


Select your Cycle ID :
SDT-109
What would you like to ...
Edit this cycle
Delete this cycle
Back


Edit details about :
SDT-109
Make :
BMW B75
Colour :
Brown
Month of purchase :
June
Year of Purchase (YYYY) :
2018
Rate :
46
Gear System :
Geared
Submit
Back


Are you sure?
Note : This action cannot be und...
Yes
No


Choose the date on which you would like to rent a cycle :
Mon 04/01/2019
Submit
Back
Cycle ID
Make
Colour
Month of Purcha...
Gear system
Rate per hour
Owner Details
SDT-102 Hercules A50 Red February 18 Non Geared 32 Click here
SDT-105 Hercules A75 Orange February 17 Geared 38 Owner's nam...
SDT-106 Hercules A50 Pink December 18 Non Geared 40 Click here
SDT-102 Hercules A50 Red February 18 Non Geared 32 Click here
SDT-105 Hercules A75 Orange February 17 Geared 38 Click here
SDT-106 Hercules A50 Pink December 18 Non Geared 40 Click here
SDT-107 Hero Streamliner Purple October 18 Non Geared 42 Click here
Owner Details
Owner's name is Ronson Lobo. Owner's phone number is 9000100015.
OK