TEST SUITE

for

SoRaSu

A Transport Company

Computerization Software

Version 1.0 approved

Prepared by

Soukhin Nayek

Raj Parikh

Sukhomay Patra

Department of Computer Science and Engineering, IIT Kharagpur

April 3 , 2023

1. Test Cases for Address Class

---------- Constructor Testing ----------

GIVEN

city= “Kharagpur”

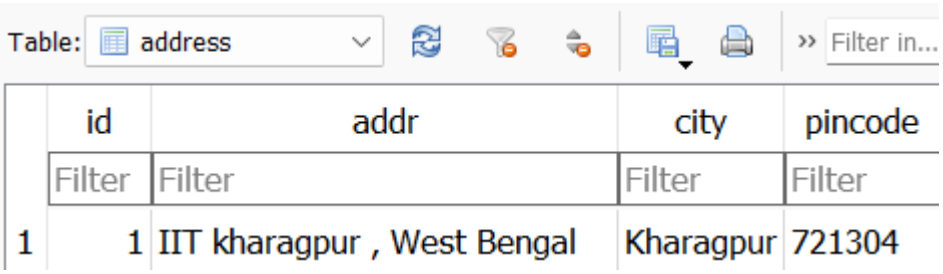
addr= “IIT kharagpur , West Bengal ”

Pincode = “721304”

THEN

assert if all values are correctly assigned or not.

This can be done by checking each attribute of Address class in sqlite software.



	id	addr	city	pincode
	Filter	Filter	Filter	Filter
1	1	IIT kharagpur , West Bengal	Kharagpur	721304

RESULT [pass]

GIVEN

city = “Gorakhpur”

addr = “A-76, Buddha Vihar”

THEN

assert if all values are correctly assigned to each attribute of Address class.

RESULT [pass]

GIVEN

city = “Kharagpur”

THEN

assert if all values are correctly assigned to each attribute of Address class.

RESULT [pass]

2. Test Cases for Bill Class

---------- Constructor Testing ----------

GIVEN

amount = 2500

branch_id= “12”

THEN

assert if all values are correctly assigned to each attribute of Bill class.

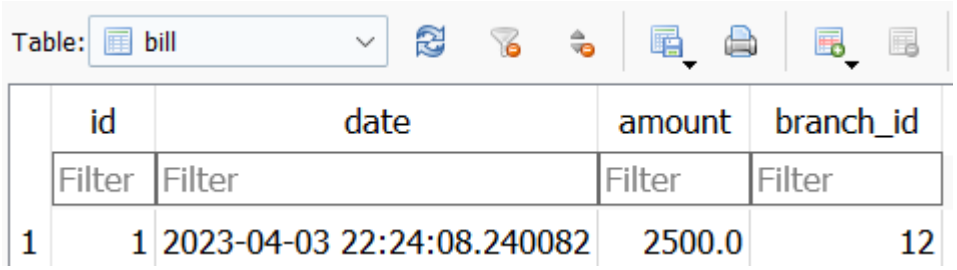


Table: bill

	id	date	amount	branch_id
	Filter	Filter	Filter	Filter
1	1	2023-04-03 22:24:08.240082	2500.0	12

RESULT [pass]

----------Method Testing ----------

1. **getDate()**

GIVEN

An object of the Bill class

THEN

assert that the date returned is the same as the date given at the time of constructing the object .
assert that the date is not a future date.

RESULT [pass]

2. **getAmount()**

GIVEN

An object of the Bill class

THEN

assert that the integer returned is the same as the amount given at the time of constructing the object .

RESULT [pass]

3. **getPaymentID()**

GIVEN

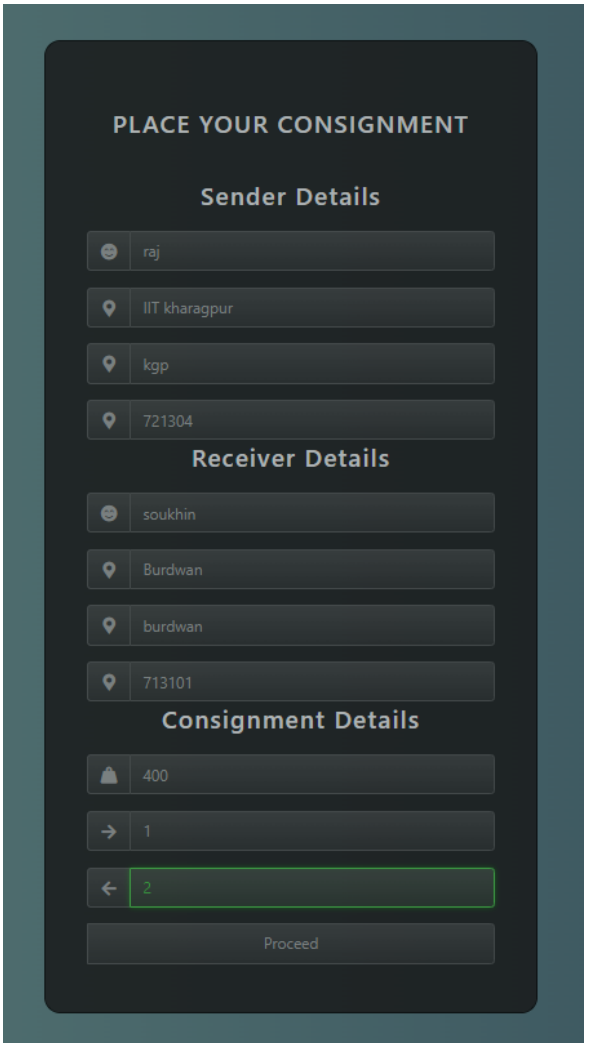
Two objects of the Bill class

THEN

assert that the string returned is the same as the paymentID given at the time of constructing each object.
assert that the paymentID of both the objects are unique.

RESULT [pass]

3. Test cases for Consignment Class



---------- Constructor Testing ----------

GIVEN

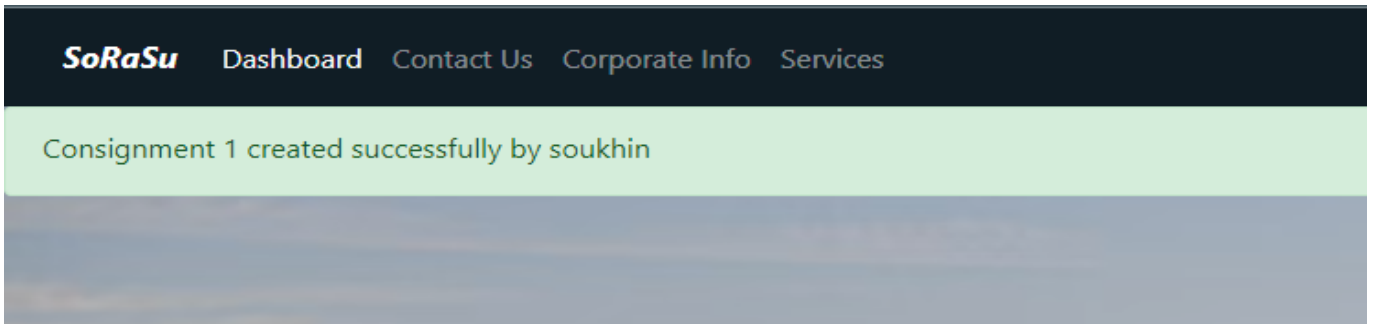
senderAddress= Address(city = “kgp”,addr= “IIT kharagpur ”Pincode = “721304”)
receiverAddress= Address(city = “burdwan”,addr= “Burdwan ”Pincode = “713101”)
bill = Bill(amount=100,branch_id=senderAddress.id)
volume = 400

consignment = Consignment(volume=volume,
bill_id=bill.id,
sender_name = “raj”,
receiver_name = “soukhin”,
senderAddress_id = senderAddress.id,
receiverAddress_id = receiverAddress.id,
sourceBranchID = 1,
destinationBranchID =2)

THEN

assert if all values are correctly assigned or not.

This can be done by checking each attribute of Address class (white-box-testing)



RESULT [pass]

----------Method Testing----------

1. **getID()**

GIVEN

Two Consignment class objects C1,C2

WHEN

Both objects are committed to database Db

THEN

assert that C1.getID() should not be equal to C2.getID()

assert that C1.getID() should be equal to 1.

RESULT [pass]

2. **getvolume()**

GIVEN

Get the volume

THEN

assert that given volume is required volume

2. **setTruckId(self,i)**

GIVEN

consignment.setTruckId(1)

WHEN

Set the truck_id=1

THEN

assert if all values are correctly assigned or not.

RESULT [pass]

2. **setStatus(self,i)**

GIVEN

consignment.setTruckId(1)

WHEN

Set the truck_id=1

THEN

assert if all values are correctly assigned or not.

RESULT [pass]

2. **getsourceBranchID()**

GIVEN

Get the source branch id

THEN

assert that given branch id is required branch id

2. **getdestinationBranchID()**

GIVEN

Get the destination branch id

THEN

assert that given branch id is required branch id

RESULT [pass]

4. Test Cases for Truck Class

---------- Constructor Testing ----------

GIVEN

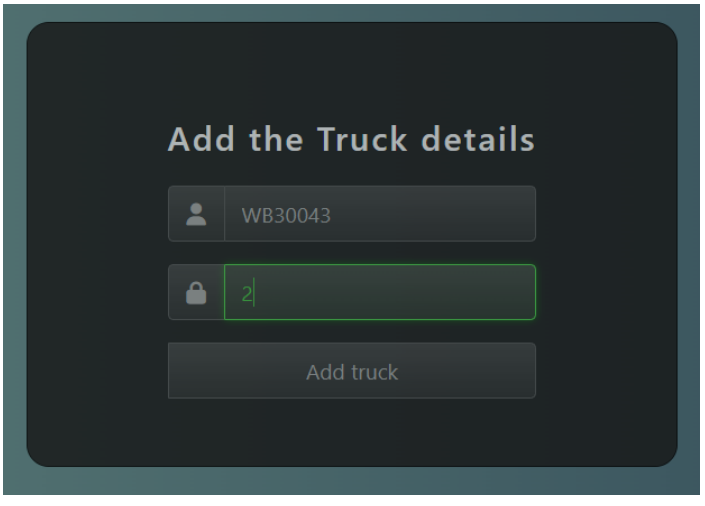
truckNumber = ”AB12C314”

branch_id = 1

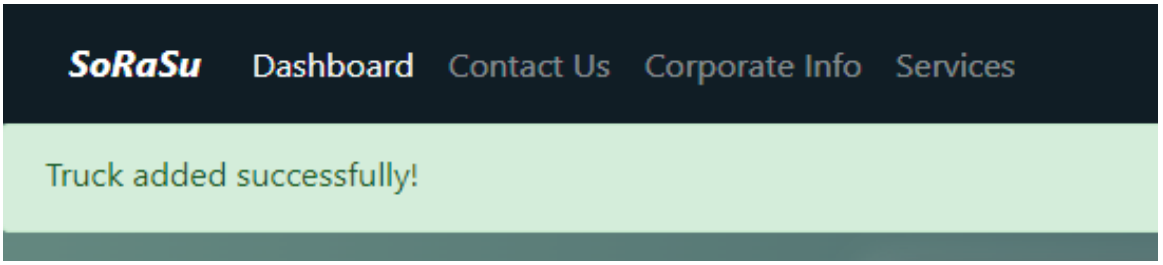
t1 = Truck(truckNumber , branch_id)

THEN

assert if all values are correctly assigned or not.



This can be done by checking each attribute of Address class (white-box-testing)



----------Method Testing ----------

1. **getTruckID()**

GIVEN

Two Consignment class objects C1,C2

WHEN

Both objects are committed to database Db

THEN

*assert that C1.**getTruckID**() should not be equal to C2.**getTruckID**()*

*assert that C1.**getTruckID**() should be equal to 1.*

RESULT [pass]

2. **getStatus()**

GIVEN

A Truck T with T.consignments = None

Case 1: When T is added to branch B, it's status is AVAILABLE

Case 2: When T is assigned any consignment C, it's status changes to ASSIGNED

Case 3: When T.isFull() is True, the truck is dispatched and it's status changes to ENROUTE

RESULT [pass]

3. **addConsignment()**

GIVEN

T = Truck(currentBranch = 1, volume = 0)

volume = 200

addr1 = Address(city = "Delhi")

addr2 = Address(city = "Kolkata")

C = Consignment(volume, addr1, addr2, destinationID)

WHEN

Truck T is assigned to a Consignment C

THEN

Truck is assigned such that it cannot take more than 500 units volume. More than one Truck will be assigned if its available volume is less than volume of a consignment. The consignments allotted to one truck will have the same destination.

4. **viewConsignments()**

GIVEN

T = Truck(currentBranch = 1, volume = 0)

volume = 200

addr1 = Address(city = “Delhi”)

addr2 = Address(city = “Kolkata”)

C1 = Consignment(volume, addr1, addr2, destinationID)

C2 = Consignment(volume, addr1, addr2, destinationID)

WHEN

Truck T is assigned to a Consignment C.

THEN

assert that the list T.consignments displays both consignments C1 and C2

RESULT [pass]

5. Test Cases for Employee Class

---------- Constructor Testing ----------

GIVEN

name = “Soukhin Nayek”

email = “soukhinkgp2@gmail.com”

branchID = 2

THEN

assert if all values are correctly assigned to each attribute of Employee class.

----------Method Testing ----------

1. **getName()**

GIVEN

An object of the Employee class E1

THEN

assert that the string returned is the same as the name given at the time of constructing the object . name is displayed in the navbar

RESULT [pass]

2. **getEmail()**

GIVEN

An object of the Employee class E1

THEN

assert that the string returned is the same as the email given at the time of constructing the object .

RESULT [pass]

3. **getBranchID()**

GIVEN

An object of the Employee class E1

THEN

assert that the integer returned is the same as the branchID given at the time of constructing the object .

RESULT [pass]

4. **set_password()**

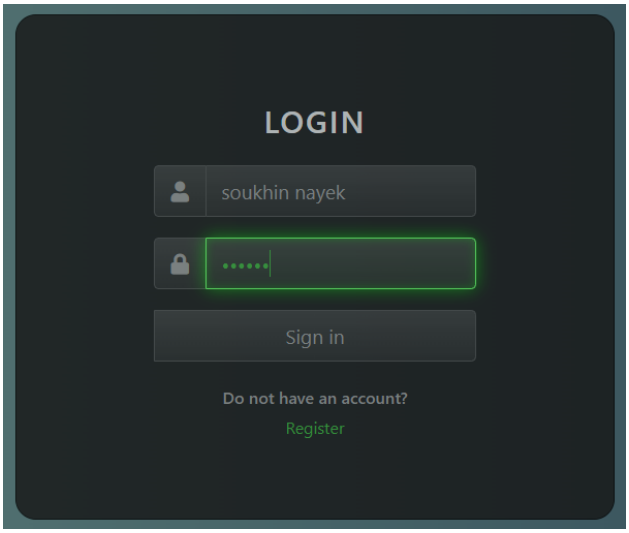
GIVEN

An object of the Employee class E1

A string which is to be set as password

THEN

assert that the password_hash is the same as the calculated hash value of the password string.



RESULT [pass]

5. **check_password()**

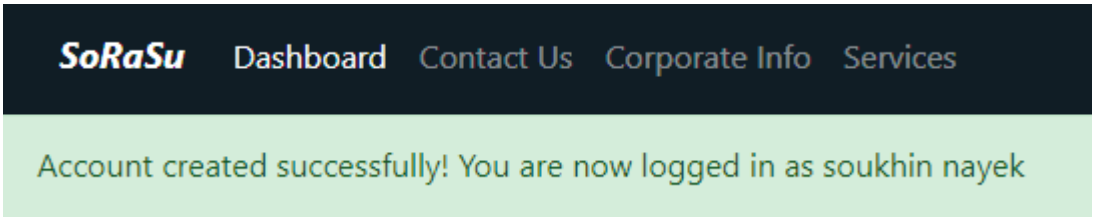
GIVEN

An object of the Employee class E1

A string which is to be set as password

THEN

assert that the function returns true if the hash value of the password string is equal to password_hash.



RESULT [pass]

6.view pending consignment

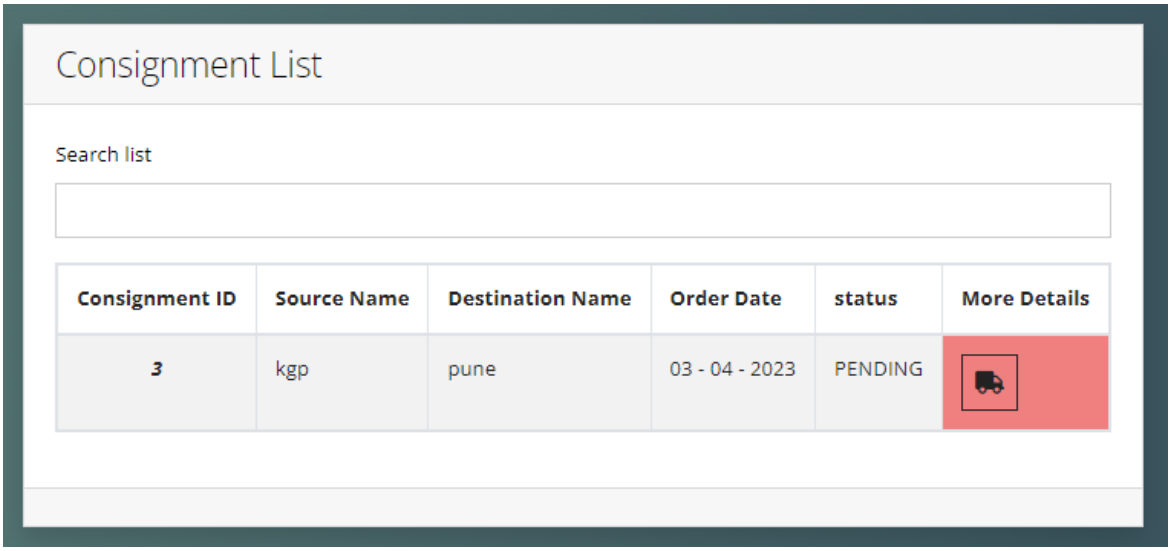
GIVEN

An object of the Employee class E1

Can check all the pending consignment in the branch

THEN

He will see a list of pending consignments where he can search.



RESULT [pass]

7.dispatch consignment

GIVEN

An object of the Employee class E1

Can check all the pending consignment in the branch

THEN

He will see a list of pending consignments.

RESULT [pass]

8.received consignment

GIVEN

An object of the Employee class E1

Can check all the pending consignment in the branch

THEN

He will see a list of pending consignments.

6. Test Cases for Manager Class

---------- Constructor Testing ----------

GIVEN

name = “Raj Parikh”

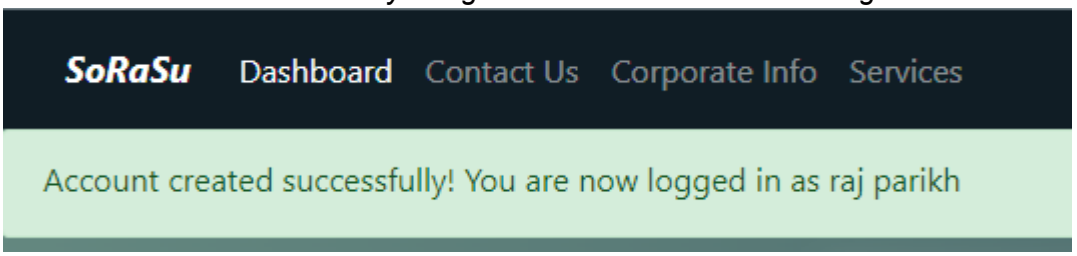
email = “ pmjindal@gmail.com ”

Position = “Manager”

branchID = 3

THEN

assert if all values are correctly assigned to each attribute of Manager class.



----------Method Testing ----------

Since Manager class is derived from Employee class, all the method tests in Employee class will comply with the Manager class.In addition to that, the following methods are tested

RESULT [pass]

1. **viewWaitingPeriod()**

GIVEN

An object of the Manager class M

THEN

assert that the average waiting period returned for a consignment is the same as the golden output.

RESULT [fail]

2. **viewWaitingTime()**

GIVEN

An object of the Manager class M

THEN

assert that the waiting time of a truck as returned by the function is correct.

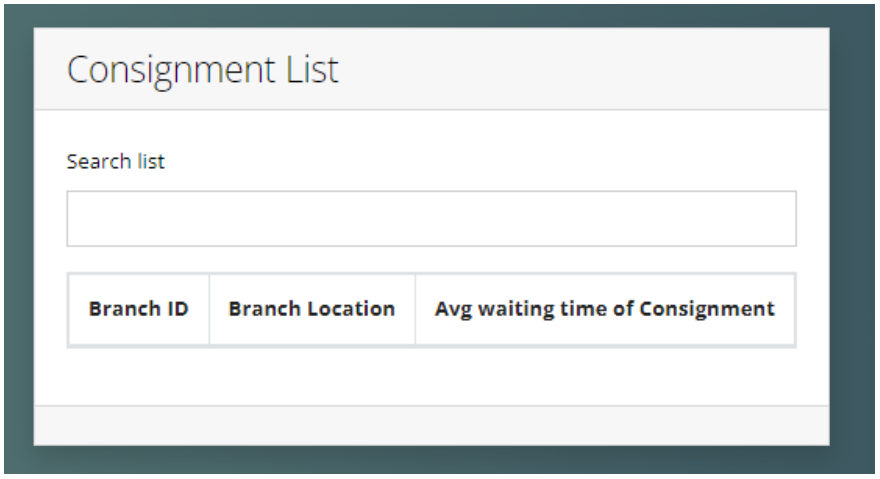
3. **viewIdleTime()**

GIVEN

An object of the Manager class M

THEN

assert that the idle time of a truck as returned by the function is correct.



RESULT [fail]

4. **changeRate()**

GIVEN

An object of the Manager class M and an integer variable storing the new rate

THEN

assert that the rate is changed to the given value .

RESULT [pass]

5. **buyNewTruck()**

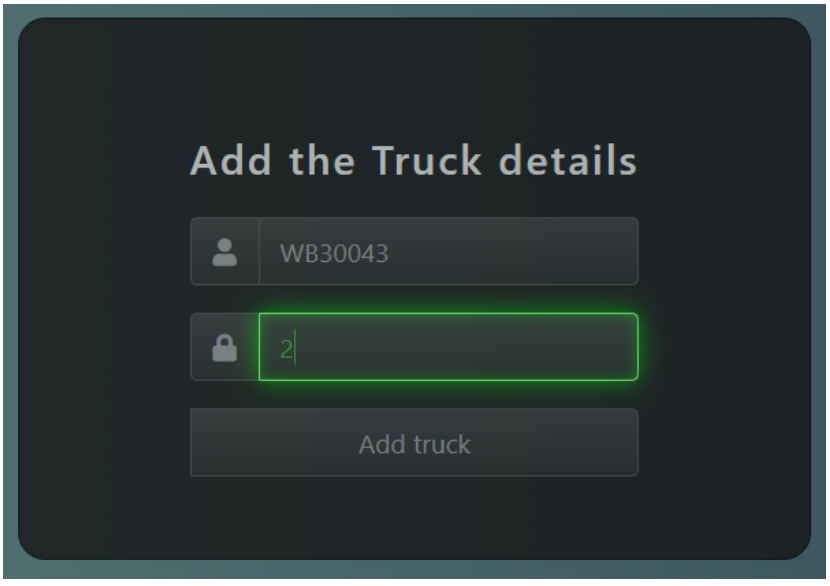
GIVEN

An object of the Manager class M B1 = BranchOffice(addr1, phone)

THEN

assert that the truck is added to the given branch.

RESULT [pass]



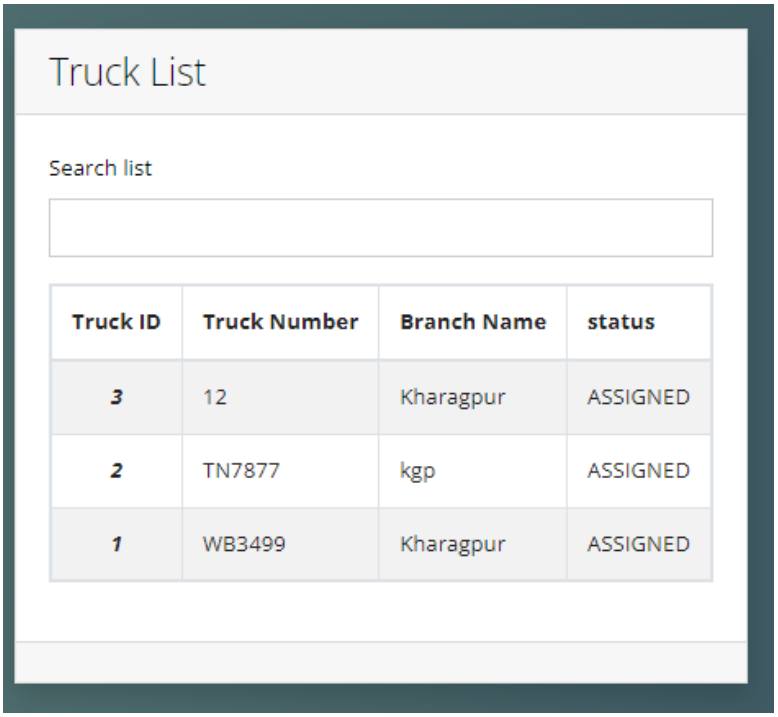
6. **TruckStatus()**

GIVEN

An object of the Manager class
M B1 = BranchOffice(addr1, phone)

THEN

assert that the status of the Truck is AVAILABLE.



RESULT [pass]

viewTruckUsage()

GIVEN

An object of the Manager class M An object of the Truck class T

THEN

assert that the usage time returned for the truck is correct.

RESULT [fail]

7. Test Cases for BranchOffice Class

---------- Constructor Testing ----------

GIVEN

address1 = Address(addr="vs",city="kgp",pincode="123445")

phone = “9876543210”

b1 = BranchOffice(rate='10',officeAddressID=address1.id,officePhone="9843843",idleTime=23)

THEN

assert if all values are correctly assigned to each attribute of BranchOffice class.

RESULT [pass]

----------Method Testing ----------

1. **getID()**

GIVEN

Two BranchOffice class objects B1, B2

WHEN

Both objects are committed to database Db

THEN

assert that B1.getID() should not be equal to B2.getID()

assert that B1.getID() should be equal to 1.

RESULT [pass]

2. **addEmployee()**

GIVEN

An Employee object E

WHEN

The employee is added to branch B1

THEN

assert that the employee is added with his branchID 1 (branchID of B1)

RESULT [pass]

3. **addTruck()**

GIVEN

A Truck object T

WHEN

The truck is added to branch B1

THEN

assert that the truck is added with his branchID 1 (branchID of B1

RESULT [pass]

5. **viewTransactions()**

GIVEN

A Branch object B1

THEN

assert that B1.viewTransactions() shows all the transactions of branch B1 only

RESULT [pass]

6. **receiveTruck()**

GIVEN

A Truck object T

WHEN

The truck is received at branch B1

THEN

assert that the Truck object T1 is properly added to list of other trucks in B1, the truck is empty and it's status is AVAILABLE. Also, all the consignments allotted to this truck should have its status DELIVERED

RESULT [fail]

8. Test Cases for HeadOffice Class

---------- Constructor Testing ----------

GIVEN

address1 = Address(addr="vs",city="kgp",pincode="123445")

phone = “9876543210”

h1 = HeadOffice (rate='10',officeAddressID=address1.id,officePhone="9843843",idleTime=23)

THEN

assert if all values are correctly assigned to each attribute of HeadOffice class.

----------Method Testing ----------

All the tests for the BranchOffice are to be complied for HeadOffice

1. **setRate()**

GIVEN

An object of the HeadOfficeClass and an integer variable rate

THEN

assert that the static constant of rate is correctly changed to the given value.

RESULT [pass]

9. Test Cases for Authorization Blueprint

----------Login ----------

GIVEN:

login credentials

URL of login

WHEN:

Employee/Manager/Driver/Customer tries to login

THEN:

Response should be appropriate with correct status code and RESPONSE HTML

Case 0:

Email doesn’t pass email validation test

email: pmjindal.com

password: aaaaaa

RESULT [pass]

Input:

POST METHOD

Response

Assert response.status == 401

Assert “invalid email” in response.data

Case 1:

Login credentials are correct

email:- pmjindal@gmail.com

password:-aaaaaa

Input:

POST METHOD

On submitting

Response:

Assert response.status == 200

Assert “Successfully logged in” in response.HTML

RESULT [pass]

Case 2:

Login credentials are wrong

email:- pmjindal@gmail.com

password:- aaaaab

Input:

POST METHOD with credentials

Response:

Assert response.status == 401

Assert “Incorrect password” in response.HTML

RESULT [pass]

Case 3:

email doesn’t exist

email:- pmjindsl@gmail.com

password: aaaaaa

Input:

POST METHOD

Response:

Assert response.status == 404

Assert “email not registered” in response.HTML

RESULT [pass]

5. Test Cases for Customer Class

---------- Constructor Testing ----------

GIVEN

name = “Soukhin Nayek”

email = “soukhinkgp2@gmail.com”

THEN

assert if all values are correctly assigned to each attribute of Customer class.

RESULT [pass]

----------Method Testing ----------

1. **getName()**

GIVEN

An object of the Customer class C1

THEN

assert that the string returned is the same as the name given at the time of constructing the object . name is displayed in the navbar

RESULT [pass]

2. **getEmail()**

GIVEN

An object of the Employee class E1

THEN

assert that the string returned is the same as the email given at the time of constructing the object .

RESULT [pass]

4. **set_password()**

GIVEN

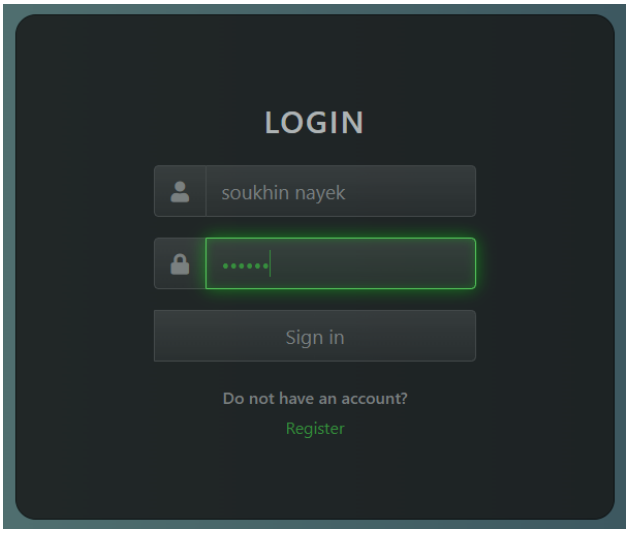
An object of the Employee class E1

A string which is to be set as password

THEN

assert that the password_hash is the same as the calculated hash value of the password string.

RESULT [pass]



5. **check_password()**

GIVEN

An object of the Employee class E1

A string which is to be set as password

THEN

assert that the function returns true if the hash value of the password string is equal to password_hash.

RESULT [pass]

Database Testing

Database testing will be done by creating objects of all model classes and committing them to the database using the ORM interface.This is followed by querying from the database back to check the validity of the commits.

1.Creating Consignment

db: Database ORM object (where everything will be committed)

GIVEN:

a = Address(addr=”MS Hall”,city = “Kharagpur”,pincode=”721302”)
b = Address(addr=”Trillium”, city = “Pune” , pincode=”411034”)

c = Consignment(volume = 100 , sender_name = ”Sukhomay” , receiver_name = ”Soukhin” , senderAddress_id = a.id receiverAddress_id = b.id , sourceBranchID = 2 , destinationBranchID = 3)

a1 = Address(addr=”VS Hall”,city = “Kharagpur”,pincode=”721302”)
b1 = Address(addr=”Jasminium”, city = “Pune” , pincode=”411034”)
c1 = Consignment(volume = 300 , sender_name = ”Raj” , receiver_name = ”Rahul” , senderAddress_id = a1.id receiverAddress_id = b1.id , sourceBranchID = 2 , destinationBranchID = 3)

WHEN:

a,b,c,a1,b1,c1 is committed to database db
db.session.add(a),db.session.add(b),db.session.add(c),db.session.add(a1),db.session.add(b1), db.session.add(c1)
db.session.commit()

THEN:

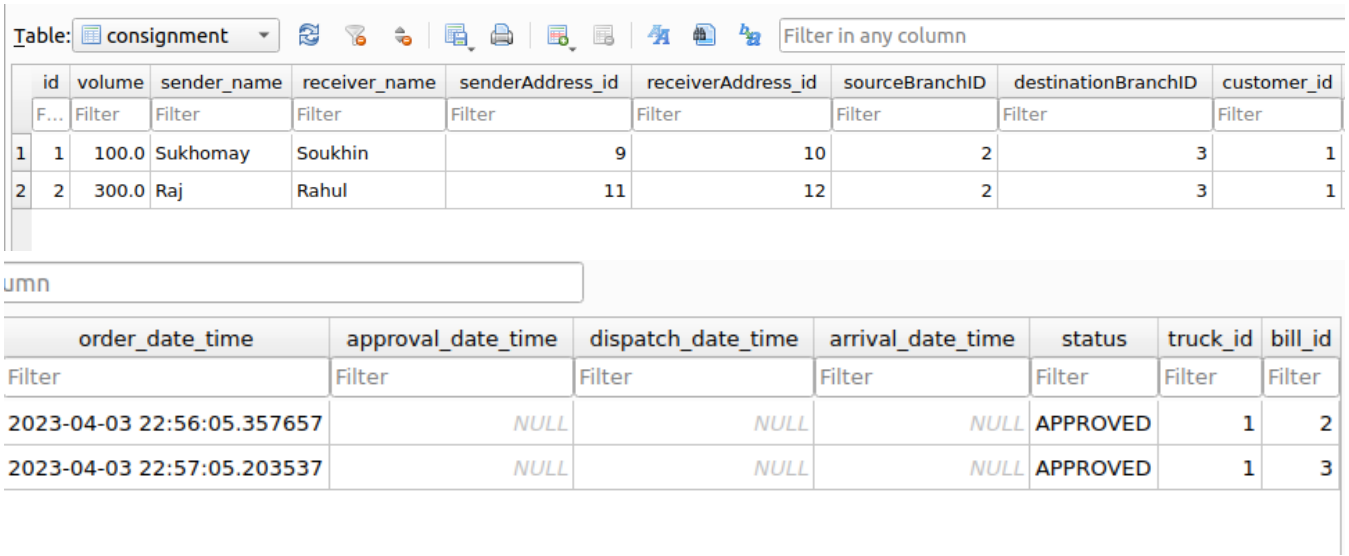


Table: consignment — Filter in any column

	id	volume	sender_name	receiver_name	senderAddress_id	receiverAddress_id	sourceBranchID	destinationBranchID	customer_id
1	1	100.0	Sukhomay	Soukhin	9	10	2	3	1
2	2	300.0	Raj	Rahul	11	12	2	3	1

umn

order_date_time	approval_date_time	dispatch_date_time	arrival_date_time	status	truck_id	bill_id
2023-04-03 22:56:05.357657	NULL	NULL	NULL	APPROVED	1	2
2023-04-03 22:57:05.203537	NULL	NULL	NULL	APPROVED	1	3

RESULT : PASS

2.Creating Customer
GIVEN:

customer = Customer(username = ‘rajparikh’ , name = “raj” , email_address = “raj@gmail.com” , password = “123456”)

WHEN:

customer is committed to database db

db.session.add(customer)
db.session.commit()

THEN:

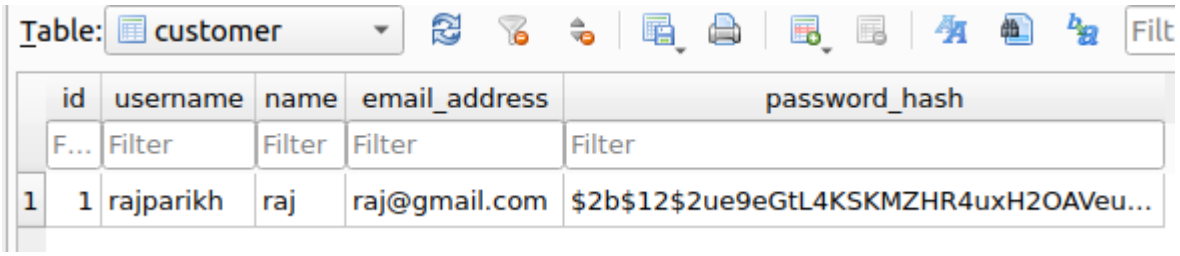


Table: customer

	id	username	name	email_address	password_hash
	F...	Filter	Filter	Filter	Filter
1	1	rajparikh	raj	raj@gmail.com	$2b$12$2ue9eGtL4KSKMZHR4uxH2OAVeu...

RESULT : PASS

3.Creating HeadOffice and Two BranchOffices
GIVEN:

address = Address(addr="IITKGP",city="Kharagpur",pincode="721302")
office = HeadOffice(rate=10,officeAddressID=address.id,officePhone="9090909")
address1 = Address(addr="vs",city="kgp",pincode="123445")
address2 = Address(addr="vs",city="pune",pincode="988989")
b1 = BranchOffice (rate = '10 ' , officeAddressID = address1.id , officePhone = "98438439880")
b2 = BranchOffice (rate = '10 ' , officeAddressID = address2.id , officePhone = "9088978989")

WHEN:

address,office,address1,address2,b1,b2 is committed to database db
db.session.add(address,address1,address2)
db.session.add(office)
db.session.add(b1,b2)Database when queried should return the object with the same fields.

THEN:

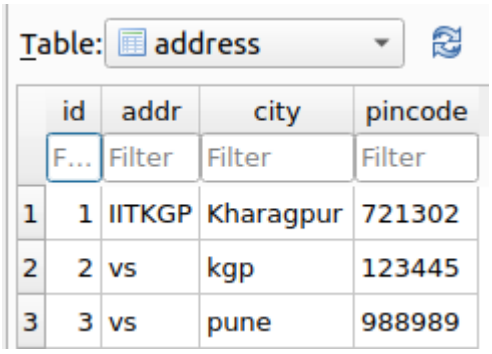


Table: address

	id	addr	city	pincode
	F...	Filter	Filter	Filter
1	1	IITKGP	Kharagpur	721302
2	2	vs	kgp	123445
3	3	vs	pune	988989

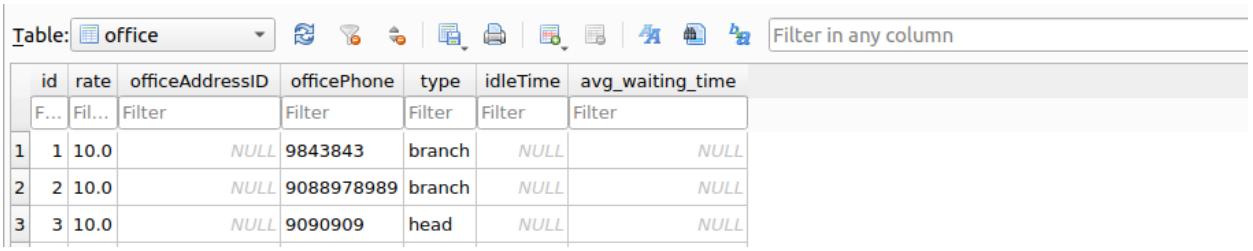


Table: office

Filter in any column

	id	rate	officeAddressID	officePhone	type	idleTime	avg_waiting_time
	F...	Fil...	Filter	Filter	Filter	Filter	Filter
1	1	10.0	NULL	9843843	branch	NULL	NULL
2	2	10.0	NULL	9088978989	branch	NULL	NULL
3	3	10.0	NULL	9090909	head	NULL	NULL

RESULT : PASS

4.Creating Manager and Employee(Branch Employee and Driver)
GIVEN:

manager = Manager(username = “man1” , name = “manager1” , email_address = “man1@gmail.com” , branchID = 1 , position = “Manager” , password = “123456”)
employee = Employee(username = “emp1” , name = “employee1” , email_address = “emp1@gmail.com” , branchID = 2 , position = “Employee” , password = “123456”)
driver = Employee(username = “dri1” , name = “dri1” , email_address = “dri1@gmail.com” , branchID = 2 , position = “Driver” , password = “123456”)

WHEN:

manager,employee,driver is committed to database db
db.session.add(customer)
db.session.add(employee)
db.session.add(driver)
db.session.commit()

THEN:

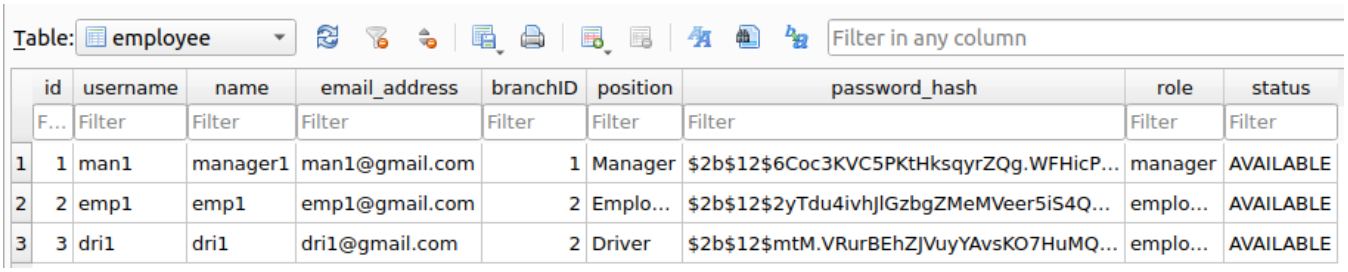


Table: employee

Filter in any column

	id	username	name	email_address	branchID	position	password_hash	role	status
	F...	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	man1	manager1	man1@gmail.com	1	Manager	$2b$12$6Coc3KVC5PKtHksqyrZQg.WFHicP...	manager	AVAILABLE
2	2	emp1	emp1	emp1@gmail.com	2	Emplo...	$2b$12$2yTdu4ivhJlGzbgZMeMVeer5iS4Q...	emplo...	AVAILABLE
3	3	dri1	dri1	dri1@gmail.com	2	Driver	$2b$12$mtM.VRurBEhZJVuyYAvsKO7HuMQ...	emplo...	AVAILABLE

RESULT:PASS

5.Creating Truck

GIVEN:

truck = Truck(branch_id = 2 , truckNumber = “MH047856”)

WHEN:

truck is committed to Database db
db.session.add(truck)
db.session.commit(truck)

THEN:

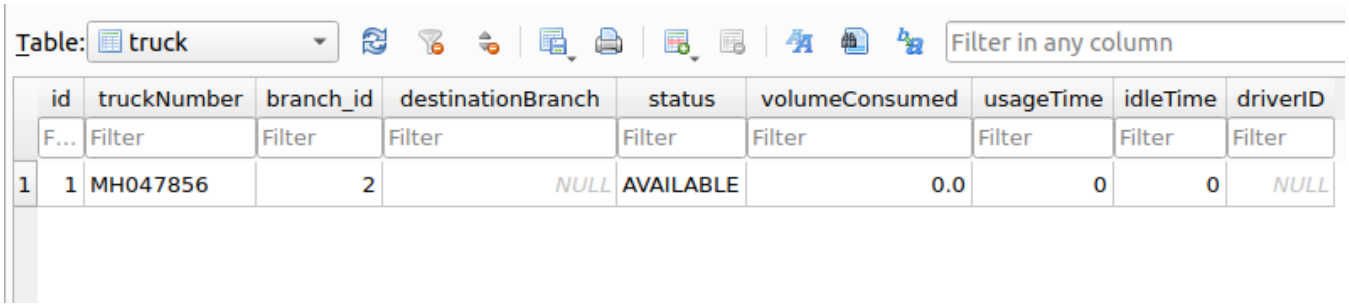


Table: truck

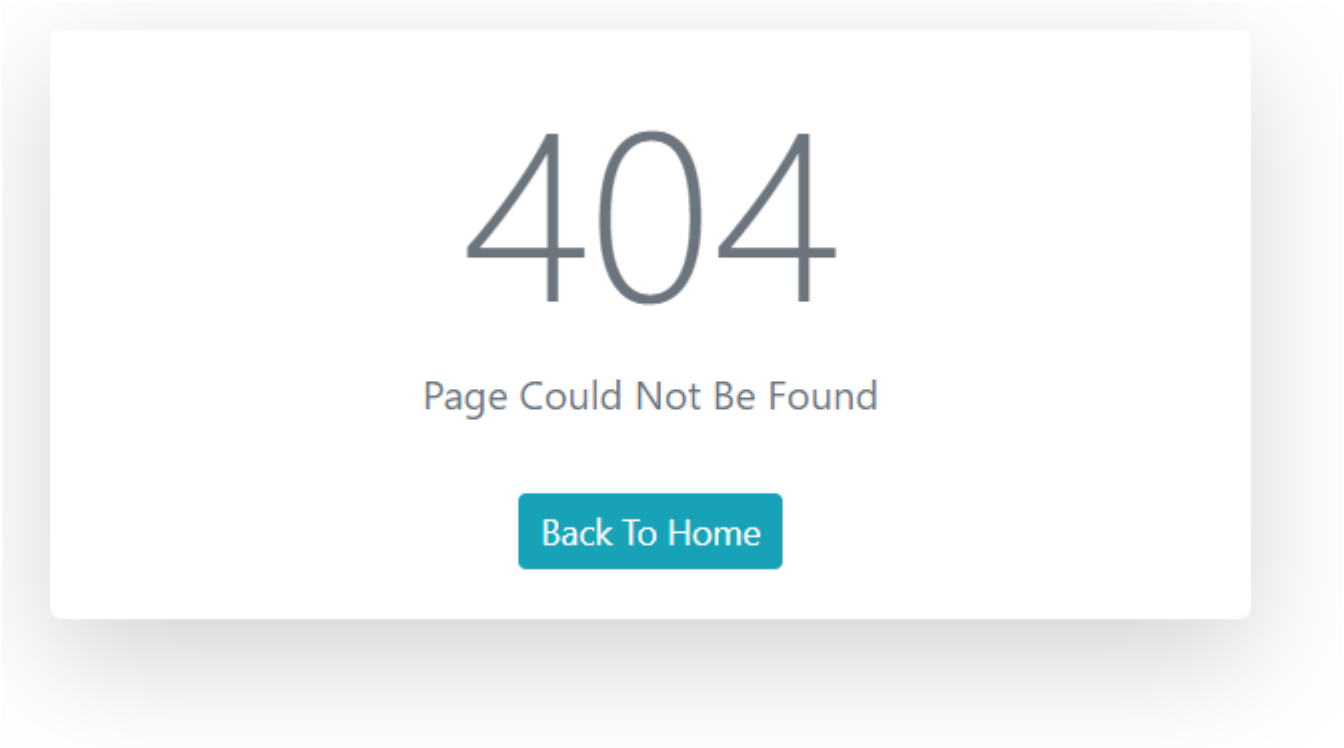
Filter in any column

	id	truckNumber	branch_id	destinationBranch	status	volumeConsumed	usageTime	idleTime	driverID
	F...	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	MH047856	2	*NULL*	AVAILABLE	0.0	0	0	*NULL*

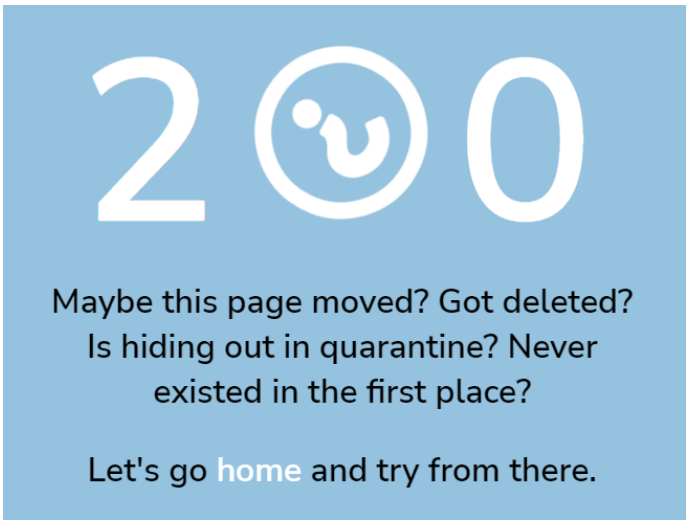
RESULT : PASS

Testing Web server and Database

1. **Error Handling**



2. Unit-test login



3. Unit-test Create Consignment

4. Unit-test Dispatch Truck

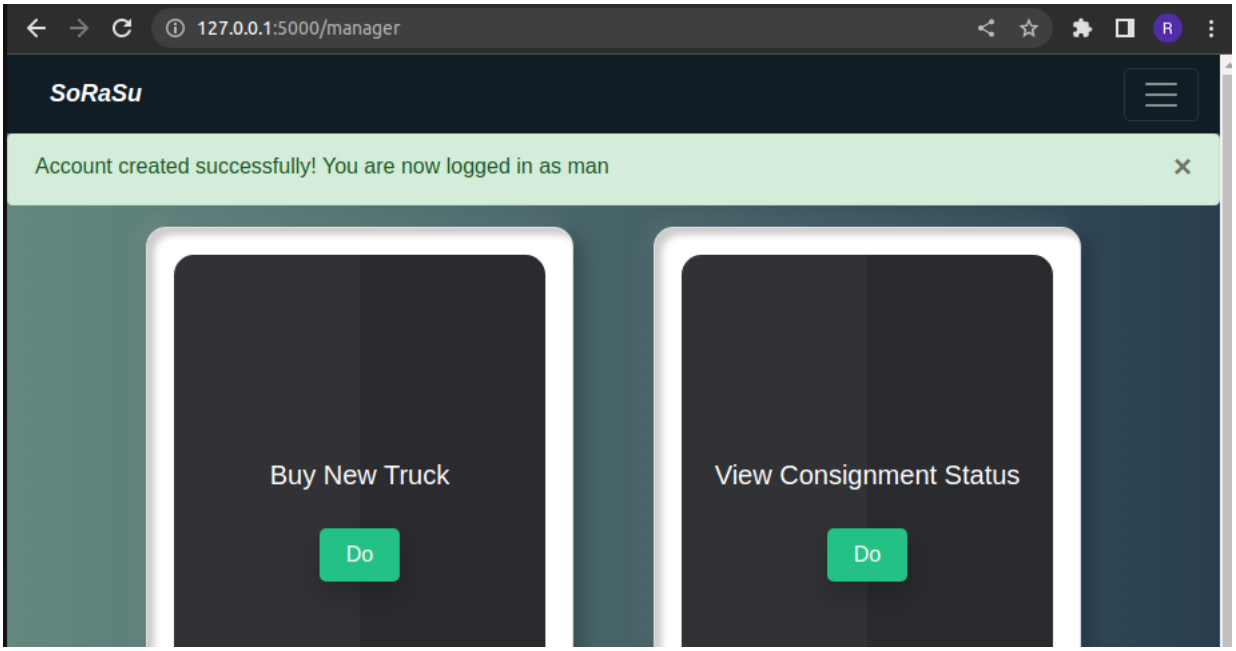
Test Cases for Application

1. When the Manager registers, it is checked that the email entered is valid and is not already present in the database.
   Manager
   M1:
   Name:
   man

   Email: man@gmail.com

   Password: 123456

2. The Manager logs in.
   username:man
   Password: 12345678



RESULT : PASS

3. E1:
   Name: emp

   username:emp

   Password: 123456

   Branch ID:

   2

Employee

E2 :

Name:

emp2

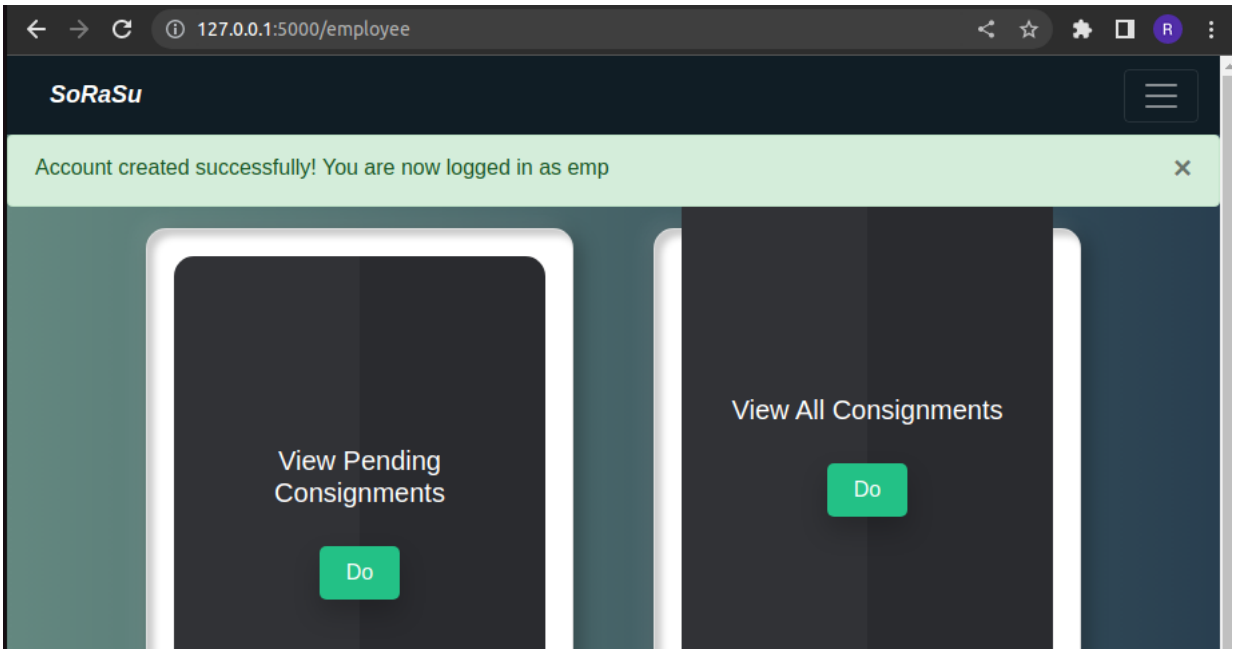
Email: emp2@gmail.com

Password: 123456

Branch ID: 2

10. The employees logs in. Login for Employee E1:
username:emp

Password: 123456
Login Successful

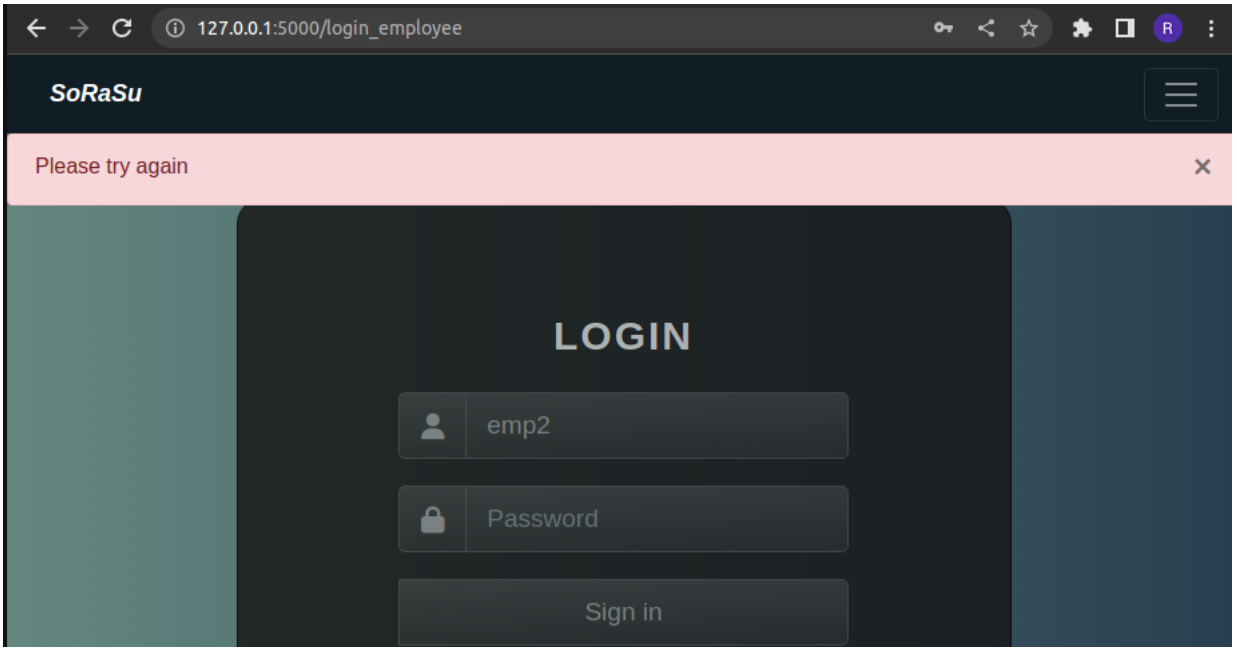


RESULT : PASS

Login for Employee E2:

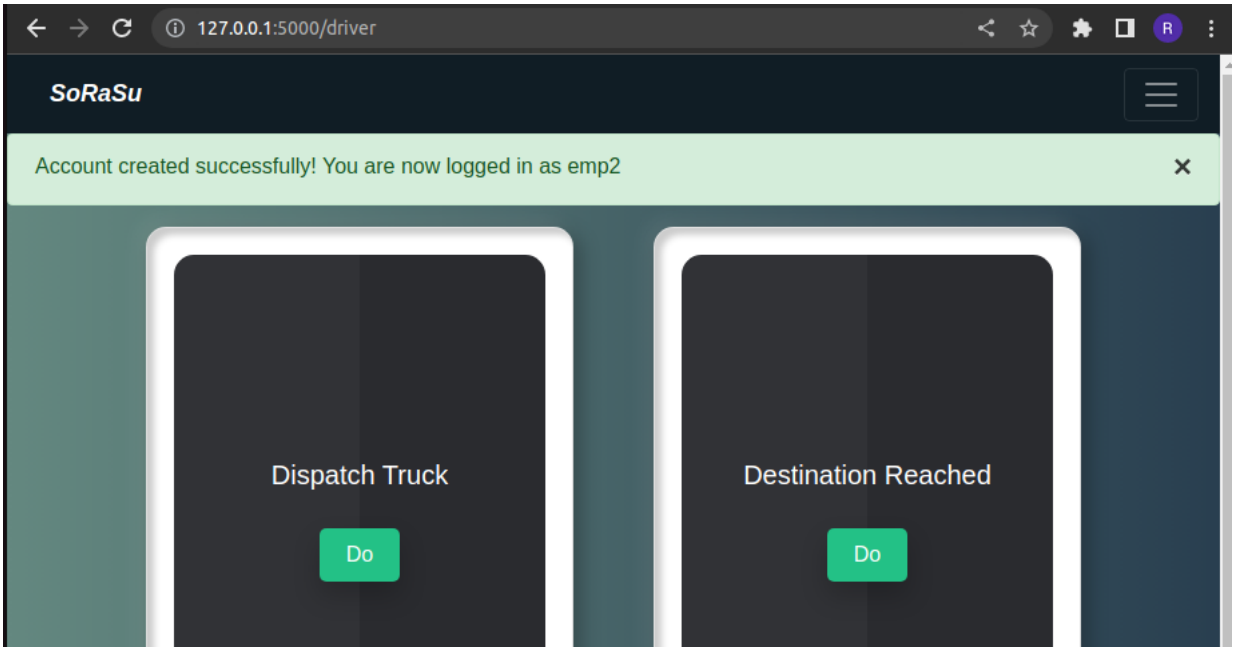
username:emp2

Password: 123132



RESULT : FAIL

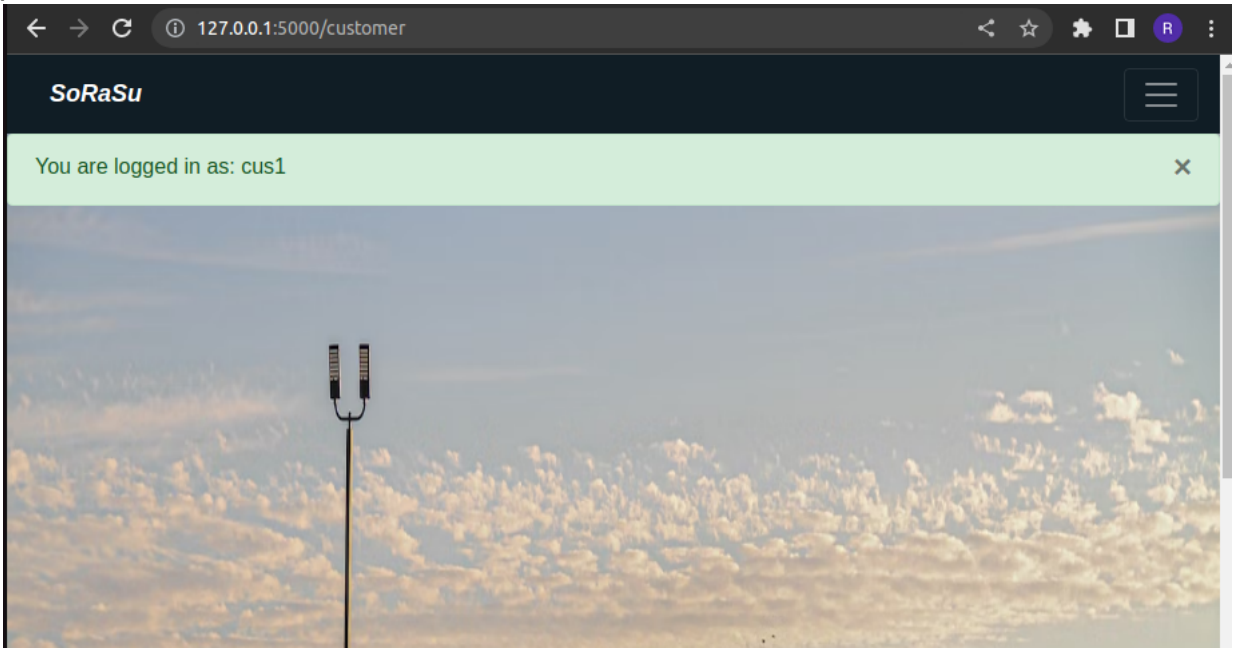
Login for Employee E2:
username: emp2
Password: 123456



RESULT : PASS

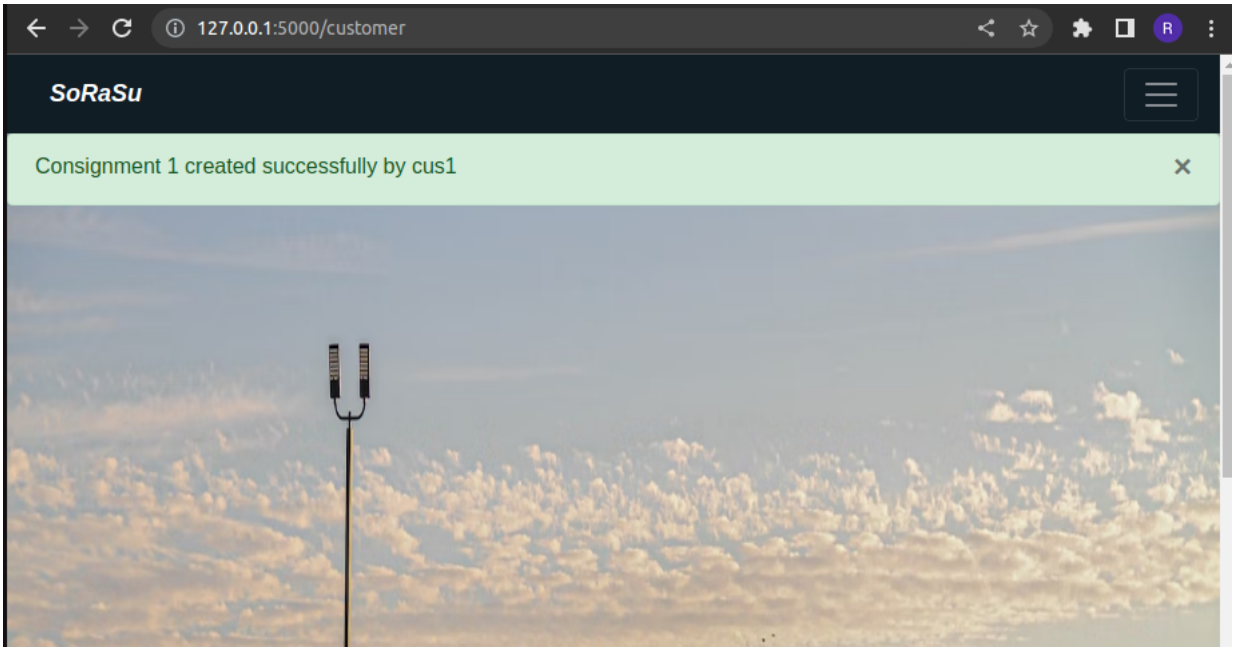
11. Customer Registers :
Username : cus1
Email address : cus1@gmail.com
Password:123456
12.Customer Login:
Username : cus1

password : 123456



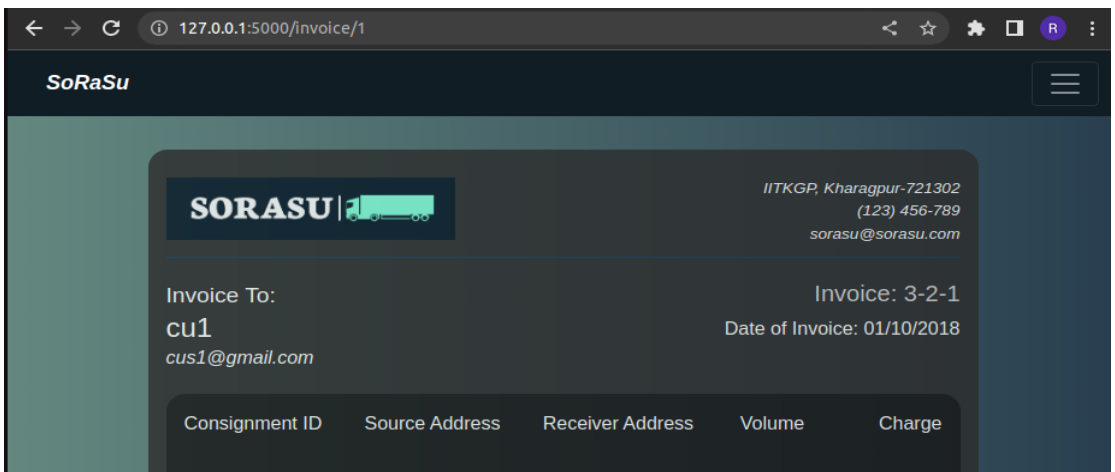
RESULT : PASS

12. Customer places a consignment:



RESULT : PASS

13. Customer view the generated bill



RESULT : PASS

14.• First Employee approves the truck
15.A truck’s status is changed from AVAILABLE to ASSIGNED as soon as a consignment is placed.
16. The consignment’s status is ALLOTED when a truck is assigned for it.
17.The truck’s status is changed to ENROUTE when the driver leaves.
18. When a truck is received, all the consignments’ status changed to DELIVERED
19. The truck is received by second employee.
20. Manager buys new truck for third branch
21. The consignment’s status is PENDING when no trucks are AVAILABLE in the branch
22.All employees logout from the application
23.Customer logout.