

Admission of patient in 2020 (Q1)

ID: 1711012
Topic: Show more in front

Name: Sumiya Afrin

CSE213 Final exam, Autumn 2020.

Section: 02

Project Title: Bangladesh Medical Association.

If not noted max 10 marks

1. Name of Doctor

2. Name of Hospital

3. Date

4. Name of patient

5. Name of doctor

6. Name of hospital

② ①
1.a) User : Ministry of Health

Goal : Generate Report

Workflow :

e1. Select hospital name.

e2. Select branch name.

e3. View to generate report

e4. Save button for save gene
generate report.

1.b) Ministry of Health

a) Approve the Committee list

Workflow:-

e1) e1. Select Committee list

e2. To list the members of the committee

e3. To select the members of the committee.

e4. Then approve the Committee list

e5. click back button.

b) Login as a ~~member~~ ^{Ministry} of Health

Workflow:-

e1. The ministry

③

b) Login as a ministry of Health

Workflow:-

R1. The ministry of Health members

e1. Can only login.

e2. See the all information

e3. Click back button.

c) See every member info.

Workflow:-

e1. See the member information.

e2. The ministry of Health will be able to view the information of all BMA members.

e3. click back button.

d) ~~Feedback to any members Report~~

~~Workflow:-~~

e1. Select ~~Feedback Option.~~

e2. if any members report whether his report is correct or not

e3. if any members report seems

incorrect then tell them

to change their report.

e4. click back button.

5

e) Selection the members problem

Workflow:-

e1. Select Problem Solution option.

e2. If they are facing any issue,

~~We~~ We will solve it.

e3. ~~Back~~ Click back button.

Final result -

⑥

1. d) ~~not~~
model class

Public class MinistryofHealth extends User

{ Private String phone No;

Public static ObservableList<Data> generate

GenerateAndReturnChartDataDef()

{ File f = null;

ObservableList<Data> list;

Scanner sc;

String str = " ";

String [] token;

7

try {
 Scanner sc = new Scanner(new File("ministryofHealth.txt"));
 while (sc.hasNextLine()) {
 String line = sc.nextLine();
 System.out.println(line);
 }
}

f = new File("ministryofHealth.txt");
Scanner sc = new Scanner(f);

f = new File("ministryofHealth.txt");
Scanner sc = new Scanner(f);

Scanner sc = new Scanner(f);

Scanner sc = new Scanner(f);

if (f.exists()) {

{
 Scanner sc = new Scanner(f);
 while (sc.hasNextLine()) {

String str = sc.nextLine();

tokens = str.split(",");

list = FXCollections.observableArrayList();

new MerchantData(tokens[0]),

⑧

```
Integer.parseInt(tokens[1])),  
    ); }  
return list; }  
else  
    System.out.println("ODPS! Emp. ID does  
not exist ..."); }  
catch (IOException ex)  
{  
    Logger.getLogger(MainFXMLController.  
        class.getName()).log(Level.SEVERE,  
        null, ex); }  
finally { } }
```

(5) (10)

1.0 Controller Class

Public class ministryofHealthReportController implements Initializable - (this method)()

{

@FXML Private PieChart;

@FXML Private LoadPieChartButton;

@FXML Private Stakeable;

@Override

Public void initialize(URL url, ResourceBundle rb){}

10.

@FXML

private void generateReportButtonOnlick
() (ActionEvent event)

{ Observablelist<Data> list =

client.generateAndReturnChancelData();

PieChart.setData(list); }

}

3 (for)

String s

S

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

2. Package abstractpackage;
import instancepackage.TransportVehicle;
Public interface Insurable
{
 void insureVehicle(TransportVehicle t);
 void renewPolicy (TransportVehicle t);
}
Package abstractpackage;
Public interface Maintainable
{
 void selfDiagnose();
 void ~~self~~ alertDriver();
}

Package instancePackage;

Public class Bus extends PassengerVehicle

{

@overried

void setVehicleInfo()

{
 }
 }

@overried

void showVehicleInfo()

{
 }

@overried

Public void selfDiagnose()

{
 }

@Override

public void actionPerformed()

{

}

}

(Compilator: Bitte nicht löschen)

Package instancepackage;

Public class MicroBuses extends PassengerVehicle

{

int workingHourPenalty;

int fairRentHour;

@Override

void selfVehicleInfo() {

}

15

@Override

void showVehicleInfo()

{ }

@Override

~~void~~ public void selfDiagnose()

{ }

@Override

public void sendDriver()

{ }

{ }

} Total class

15 15

package instancepackage;

public abstract class PassengerVehicle

extends TransportVehicle

{ int roofseat; }

String licensenb, area, destination; }

public void totalIncome()

()

~~Book~~

else ()

else ()

(16)

Package interfacepackage;

import abstractpackage.maintainable;

Public abstract class TransportVehicle

implements maintainable

{
String regNo, brand, Price, etc;

String fuelType;

float getTotalIncome()

{ return totalIncome; }

abstract void setVehicleInfo();

abstract void showVehicleInfo();

Public String getRegNo()

{ return regNo; }

Package instancepackage;

Public class Truck extends TransportVehicle

{ String category, licenseNb, area, destination;

@overriden

void setVehicleInfo() {

throw new UnsupportedOperationExceptionException

{ ("not supported yet."); }

~~@Overriden~~

void showVehicleInfo()

throw new unsupportedOperationException

("not supported yet.");

~~@Overriden~~

~~Did~~

public void selfDiagnose()

throw new unsupportedOperationException

("not supported yet.");

(19)

@overridden

public void overDriven() {
throw new UnsupportedOperationException
("not supported yet"); } }

package instancepackage;

import java.util.Scanner;

public class DriverClass

{ public static void main(String[] args) {
TransportCompany tc = new TransportCompany();

20

~~Scanner SC = new Scanner(System.in);~~

first choice;

choice = sc.nextInt();

do

{ if (choice == 1)

d' merobes me. ³ z new Micocacabell.

mc.setVehicleInfo(),

+ addTransport(me)

F. Haevercade (mc), 7

(21)

else if (choice == 2)

{ Bus b = nearBus();

b. setVehicleInfo(); }

t. setAddATransport(b); }

t. ignoree (b); }

else if (choice == 3)

{ t. StockInfo(); }

~~else if Cho~~

{ Trueu tas new Track();

ta. setVive iCteInfo();

92

t.addATraxport(ta);
t.insurevehicle(ta); } }

elseif (choice == 4) {

{ t.showVehicleInfo(); }

elseif (choice == 5)

{ }

{ }

elseif (choice == 6)

{ t.showVehicleInfo(); }

else if (choice == 7)

{ ~~if isenreable~~

+ ShowTotalIncome(); }

else if (choice == 8)

{

}

while (choice != 9)

{

} }