Software Tools and Techniques
(COMP1618)

Final Report

Submitted By
Tufayel Ahmed Chowdhury
001150759-4
MSc Computer Forensics and Cyber Security

Module leader: **Dr. Tuan Vuong**
Submission Date:17-December-2021

Contents

List of Figures

List of Tables

Greenwich Cabs Project Report

Submitted By
Tufayel Ahmed Chowdhury
001150759-4
MSc Computer Forensics and Cyber Security

Submission Date:17-December-2021

1 Introduction

The Greenwich Cabs project for a small minicab company operating with only 10 cars and this company has four dispatcher employees. This coursework is a brief explanation of my coursework-1 which was to design and implement stand-alone object-oriented programming (OOP) Java application with a graphical user interface that is allowing the dispatcher to record journey details of customers and also allow to show the earning details of a driver in a particular day that is providing from the Greenwich Cabs company. The main functionalities of the dispatchers are to record the journeys and the fares. These records are for legal reasons as well as calculate the percentage that the drivers pay. The company provides 20% of the fare to the driver for the work. Basically, this system has four dispatchers and 10 drivers and the dispatcher store the data using MS Excel and now the company is looking for a software solution which allows dispatchers to store data as well as can see the data. In this project, I have used MySQL database for storing data and for graphical user interfaces (GUIs) have become the user interface of choice for making software more interactive. Here I used Java.
NetBeans IDE is a free and open source integrated development environment for application development on a different operating systems such as Windows, Mac, Linux, and Solaris. I used NetBeans swing to design the graphical user interface which produce the most interactive GUI. For programming, I used OOP java and keep the records by using MySQL database.
Moreover, most of the functionalities of the application have already been discussed in the previous report and most of the minor steps are avoid repetition in this report. Therefore, this is highly recommended to read this report complemented with coursework-1.
As discussed in the previous report, Application Lifecycle Management (ALM) has been adopted the entire coursework for planning, designing, development, all

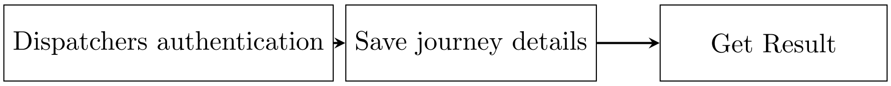


Figure 1: Process Diagram

testing, and extension. The UML diagrams have been designed and developed for most of the classes in coursework-1 and the rest of the classes are included in this report as well as this coursework will deliver the designing, development, and testing of some additional classes and their result.

2 Design

I have analysed the requirements of the company and from the requirement analysis I collected the main entities and attributes and created the classes. Below I describe the classes that I found:

2.1 Dispatcher Class

This class is responsible to identify the valid dispatcher who can access the system. Though this system has four dispatchers, these four dispatchers can access the system and create the records for the journey as well as they can see the earning details of a driver for a particular date.

2.2 Journey Details class

A dispatcher can create journey records for passengers as well as can delete or update the journey for a passenger with their mobile number and the delete is only possible within 24 hours in this system for security purposes.

2.3 Driver Class

In driver class, it contains driver details which include the total job has done for a particular driver on a particular date. Also, the total amount of money he earned, would be counted.

2.4 Earning Details Class

In earning details class will be executing the total number of jobs have done and the total amount of money earned on a particular date. Finally, it will show the details of the statistics for a particular date.

2.5 Conceptual UML diagram

There have been some changes made to some of the new classes included in the system as well as some of the class diagrams designed in coursework-1. These are shown below the conceptual design of how dispatchers handle the system. It is highly recommended to coursework-1 for detailed design and explanation of all other classes and their functionalities. Moreover, from my all classes and functionalities I have designed the conceptual UML diagram as shown in figure 2.

2.6 Database Diagram

From the entities and attributes and UML class diagram I have designed the database diagram as shown in figure 3.

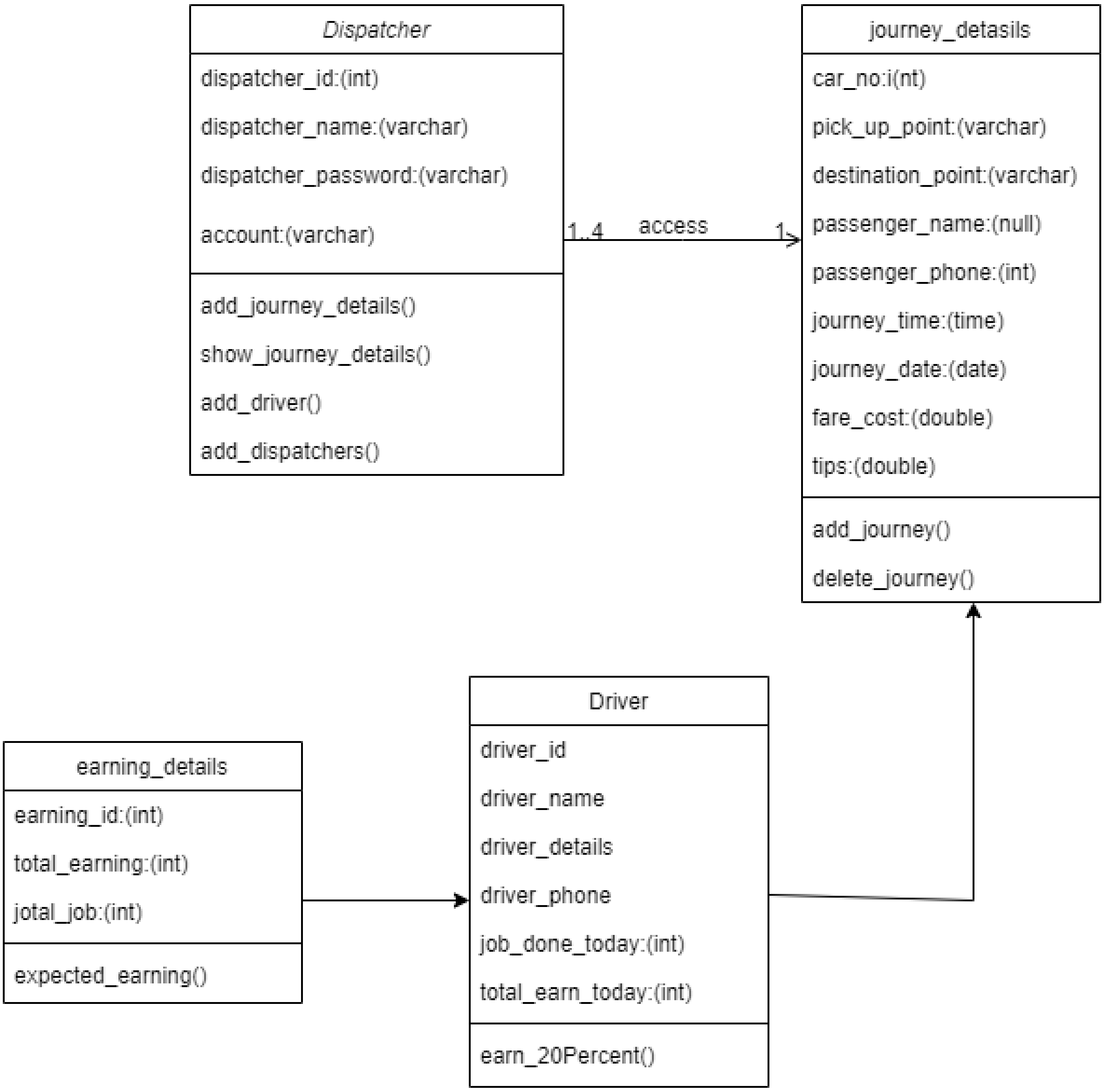


Figure 2: UML class diagram.

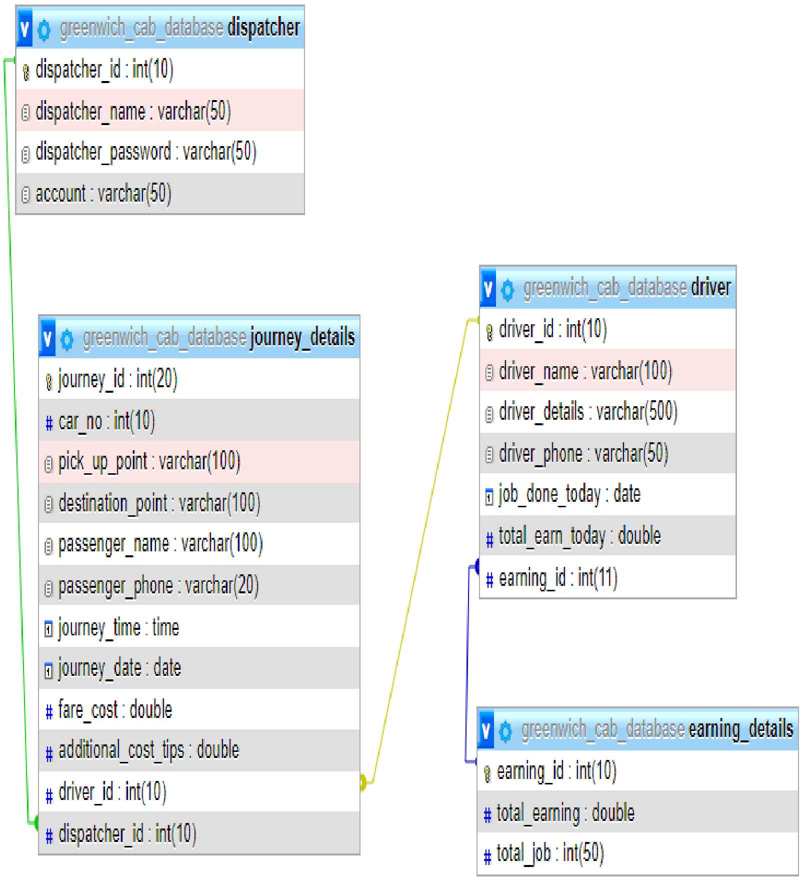

greenwich_cab_database dispatcher
dispatcher_id : int(10)
dispatcher_name : varchar(50)
dispatcher_password : varchar(50)
account : varchar(50)
greenwich_cab_database journey_details
journey_id : int(20)
car_no : int(10)
pick_up_point : varchar(100)
destination_point : varchar(100)
passenger_name : varchar(100)
passenger_phone : varchar(20)
journey_time : time
journey_date : date
fare_cost : double
additional_cost_tips : double
driver_id : int(10)
dispatcher_id : int(10)
greenwich_cab_database driver
driver_id : int(10)
driver_name : varchar(100)
driver_details : varchar(500)
driver_phone : varchar(50)
job_done_today : date
total_earn_today : double
earning_id : int(11)
greenwich_cab_database earning_details
earning_id : int(10)
total_earning : double
total_job : int(50)

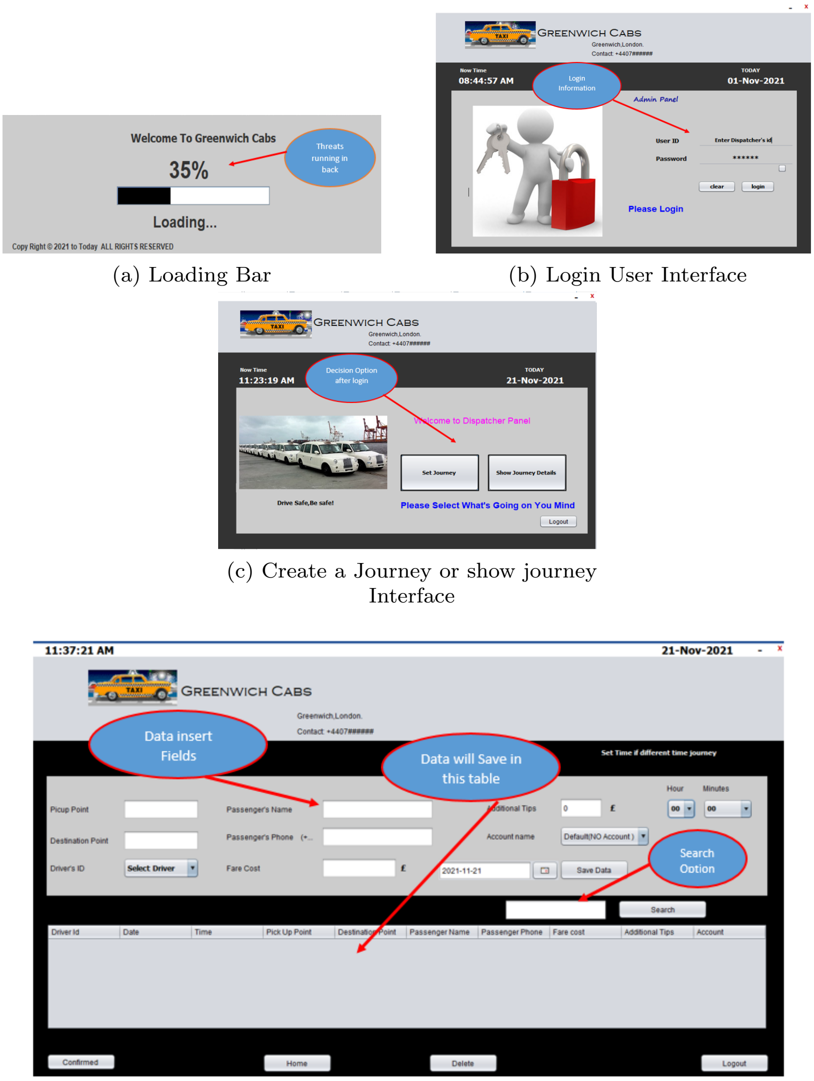
Figure 3: UML class diagram.

3 Application Development

3.1 Implementing all Classes

Most of the classes used in this system have developed in coursework-1. This section will describe more details about each class and its functionalities. In coursework-1, I described details about the dispatcher class and its user interphase part.

4 Citation examples



(a) Loading Bar

(b) Login User Interface

(c) Create a Journey or show journey Interface

(d) Interface for creating a journey

Figure 4: Major Overview of GUI Design

In my previous report, I included details about my all GUI design. Figure 4 shows the most common GUI design of my project. For loading bar GUI design, I used the thread sleep method for using to pause the execution of the current thread for specified milliseconds. In my software, I used threat sleep 30 milliseconds.

According to the previous report, after verifying the dispatcher's authentication, a dispatcher will be able to access the system and can see the details of the journey of the driver. If the dispatcher Id and password wrong the error message will be pop up that I mentioned details in my coursework-1 report. After login, a dispatcher can see the decision GUI design that a dispatcher wants such as can create a journey or can show the journey details that are our primary requirements for this project. From this page, a dispatcher can set a journey or can show details of the journey. If a dispatcher wants to set the details, click the journey details button and set the journey details.

After giving the details of a journey, the dispatcher first has to save the data into the table. Next, by the confirmed button, he can confirm the booking of the journey and it will be recorded. A dispatcher can delete of edit or update the table by clicking on the row of the table and can update the data. Also, if any passenger wants to change the journey before this particular date, he can update the journey or cancel his journey. For this, the dispatcher needs to know the passenger's phone number for searching. Also, for security purposes, I have implemented that if the dispatcher wants to delete any journey, it should have to be the same date of the journey created. In addition, there is a time input filed so that dispatcher can set time and advanced date otherwise by default it takes the current date and time automatically.

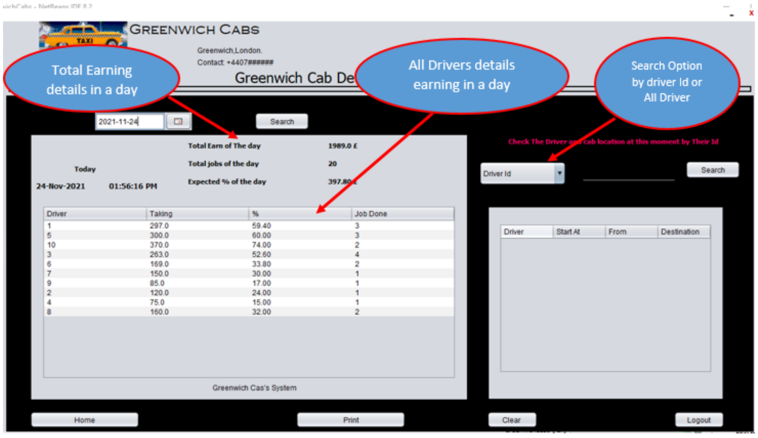
Also the dispatcher can see the details of the drivers how much they earn on a particular date. It will be calculated how much job is done and how much money earn on a particular date as well as will be calculated how much will be in 20% as expected earnings in total cost. Also, there is a table which will show the specific driver how many jobs has completed and how much he has earned, and in 20% how much he expected money earned on a particular date. The dispatcher also can see the same details of any date by selecting a date. I also set print facilities for dispatchers that they can save or print and share with the owner of the company.

On the right-hand side, the dispatcher can track the driver when he started the journey and his pickup point and destination point by his driver ID. Also, in the dropdown button if the dispatcher selects "select all", it will show all details of the driver when they started the journey, their pickup point, and destination point.

4.1 Checking Interface

I need to test the validation for the dispatcher. If any dispatcher makes any mistake such as wrong username, password or invalid character, etc. it should show some error message. Also, when data is executed successfully into the database the successful message will be shown with the passenger's name and his booking time and date.

If the dispatcher didn't enter a valid passenger phone number that he entered data in that day that's why the error message showed. I also keep restrictions for all input fields if the dispatcher makes any mistake the error message will


Greenwich Cabs
Greenwich,London.
Contact: +4407######
Greenwich Cab De
Total Earning details in a day
All Drivers details earning in a day
Search Option by driver Id or All Driver
Search
Check The Driver and cab location at this moment by Their Id
Driver Id
Search
Greenwich Cas's System
Home
Print
Clear
Logout

2021-11-24

Total Earn of The day 1989.0 £

Total jobs of the day 20

Expected % of the day 397.80 £

Today

24-Nov-2021 01:56:16 PM

Driver	Taking	%	Job Done
1	297.0	59.40	3
5	300.0	60.00	3
10	370.0	74.00	2
3	263.0	52.60	4
6	169.0	33.80	2
7	150.0	30.00	1
9	85.0	17.00	1
2	120.0	24.00	1
4	75.0	15.00	1
8	160.0	32.00	2

Driver	Start At	From	Destination

Figure 5: Showing all saved data in a particular date and driver details.

appear. For example, phone number, fare cost, and additional tips must be number and pickup point and a destination point will never only number it can be number and character or only characters. Also, tips could be 0 to any number but the fare cost must start more than 4 pounds.
There is also a delete button dispatcher that can delete any passenger's journey on the same date when the passenger purchased the journey.
According to my data insertion for a specific date 24-12-2021, we can see each driver how much job has been done and how much they have been taken and it is shown in figure 6. Also, we can see the expected amount of money for 20% company policy. For example, driver 1 has taken a total of 297 by completing 3 jobs and his 20% expectation is 297£. For each drive the value is correct. Also, we can see the total jobs have done by all drivers is 20 and total earning is 1989 £ and expected amount of 20% is 397.80 £ which is performed accurately as like a dispatcher insert the data.

According to figure 5, it is noticed that by clicking the search button we can see the driver's details when he starts as well as his pickup and destination point. The same details we can see by just selecting the Driver id and putting the driver ID as shown in figure 5.

5 Testing

Testing is an extremely crucial part of any software. In m y case, I have tested my all classes with all possible valid and invalid input and create the below testing table.

Table 1: Software testing evaluation

Test ID	Class Name	Test Description	Input	Expected Output	Actual Output	Result Status

1	loginPage.java	Valid dispatcher id and password	Dispatcher id:111111 Password:99999 Click login button	Successful login	Successful login	Pass
2	loginPage.java	Invalid dispatcher id and password	Dispatcher id: admin Password: admin Click login button.	Show a popup error message.	Show a popup error message.	pass
3	LoginPage.java	valid dispatcher id and invalid password	Dispatcher id :111111 Password: admin Click login button	Show a popup error message.	Show a popup error message.	Pass
4	LoginPage.java	Invalid dispatcher id and valid password	Dispatcher id: Admin Password: 99999 and Click login button	Show a popup error message.	Show a popup error message.	Pass
5	LoginPage.java	Empty dispatcher id and password.	Empty dispatcher id and password Click login button	Show a popup error message.	Show a popup error message.	pass
6	LoginPage.java	Empty dispatcher id and valid password.	Empty dispatcher id and password :99999 Click login button	Show a popup error message.	Show a popup error message.	pass
7	allPage.java	“set journey” button verification	Click “set journey”	Go to journey setup page	Go to journey setup page	Pass
8	allPage.java	“journey details” button verification Click	“journey details” button	Go to journey details page	Go to journey details page	Pass
9	allPage.java	“logout” button verification	Click “logout” button	Go to login page	Go to login page	pass
10	Journey.java	Set pickup point empty	Put all valid input except Pickup point (make it empty) and click save button	Show a popup error message.	Show a popup error message.	pass
11	Journey.java	Empty input field validation	Put all valid input except destination point (make it empty) and click save button	Show a popup error message.	Show a popup error message.	pass

12	Journey.java	Not select any Driver id	Set all valid input except the driver id not selected from combo box and click save button	Show a popup error message.	Show a popup error message.	pass
13	Journey.java	Valid fare cost	Set fare cost more than 5 and other valid input and click save button	Save data into the table	Save data into the table	pass
14	Journey.java	invalid fare cost	Set less than 5 or try to set string character and click save button	Show a popup error message.	Show a popup error message.	pass
15	Journey.java	Valid tips	Set 0 or any number value and click save button	Save data into the table	Save data into the table	pass
16	Journey.java	Invalid tips	Set empty or try to set string character or less than 0 and click save button	Show a popup error message.	Show a popup error message.	pass
17	Journey.java	Valid data save into database.	Click Confirmed button	show successful message for data save into database.	show successful message for data save into database.	pass
18	Journey.java	Clear all fields or table	Click clear button	Clean everything	Clean everything	pass
19	Journey.java	Go back previous page	Click home page	Go to the home page(allPage.java class)	Go to the home page(allPage.java class)	pass
20	Journey.java	logout verification	Click logout button	Go to the login page	Go to the login page	pass
21	Journey.java	Valid search	Set 09876543 phone number into search field And click search button	Show the details of journey of in the table.	Show the details of journey of in the table.	pass
22	Journey.java	In Valid search validation	Set 976543 phone number into search field And click search button	Show message customer not found	Show message customer not found	pass

23	Journey.java	Delete any journey	Set 09876543 phone number into search field and click search button. Select data from the table And click delete button.	Successful delete message show	Successful delete message show	pass
24	detailsPage.java	Selected date journey show	Select the data “2021-11-08” in data field and click search button	Show journey details of “2021-11-08”	Showed journey details of “2021-11-08”	pass
25	detailsPage.java	Show all driver’s current journey details of any date	Select “check all” from check box and click search button beside the search field.	Show all journey of driver	Show all journey of driver	pass
26	detailsPage.java	Show any driver’s current journey details of a particular date	Select “Driver Id” from check box such as set 1 in the input field. Next, click search button beside the search field.	Show driver 1 details into the below table such as starting time, pickup point and destination point and Google Map will appeared beside the table as shown in figure above.	Show all details and Google MAP	pass
27	detailsPage.java	Print verification	Click print button	save all details of selected date as a pdf file.	save all details of selected date as a pdf file.	pass

6 Critical evaluation

I have discussed some restrictions as well as the strengths of protect in coursework1. Now I am going to explain more details about it.

6.1 Strengths

6.1.1 Good user interface

This project maintains the quality of a software with good colour combination which is user friendly as well as I used some images which enhanced the good interface for the user.

6.1.2 User friendly

I reduced some input field and exchange it with combox so that the users do not need too much typing and it protects users from typing mistake. Also I used JCalender and jCombobox which are help the users to select any data quickly without making mistake.

6.1.3 Versioning

Throughout the project implementation I followed the versioning. So, anytime this project can be updated with new requirement features.

6.1.4 Protected Mistake

In all input fields are validated. So, users are safe from making mistakes. If any wrong input occurs, the error message will show what mistake happened.

6.1.5 Map facilities

In this project, dispatchers can check the journey details such as when driver start journey, pickup point and destination point of any driver's or all drivers' details who already have taken the passengers.Also, the Google Map will will be appeared beside the table by replacing the calculation table.

6.2 Weakness

While this project is 100% bug free and it is quite impossible to find out any error. However, there have some limitations from my normal observation and it was not the requirements of this software. In addition, if it can be reduced, the software will be more user-friendly.
-Most of the input fields need typing and there not any mentioning option. For instance, if I used the google API for pickup and destination point, it would be more user-friendly.
-There is no option for taking the money by using bank card or oyster card. So, there is a great limitation in transaction method.
-There have not any option for adding new dispatcher or any new driver because of their requirements was to make it for 10 drivers and four dispatchers.

6.3 Future plan

I had faced plenty of challenges while implementing the software. I recovered all the challenges and acquired new knowledge and skills. In future, I will work with this project's limitation that how to make it more user-friendly and how resolve the challenges that is mentioned in above weakness section.Also, if I deliver this software the company, I will include the Google API into my google map which will show the real time driver location.

7 Conclusion

Greenwich Texi Cab is a small mini cab company and company's four dispatchers are recording all the details of the journey by using spread sheet for handling their 10 cars. For a permanent solution I implemented a software using java for programming and MySQL for database which can also track records of the driver. Dispatcher can see the expected amount of money that how much a driver daily earns. Also, dispatcher can see the when driver start the journey and end the journey. In addition, there is a Google Map facilities where dispatcher can see the current location of the driver.However, there is a limitation of using google Map that the Google API is not free. Therefore, for getting this facilities, company need to busy this google api.Finally, this software is user-friendly which protects user from mistake and it is a 100% bug and error free software.

8 References

References

Oracle help center Java Tutorial (n.d.). https://docs.oracle.com/cd/E14507_01/apirefs.1112/e14133/java001.htm#BENAD385. Accessed: 2021-09-15.

Stack Over Flow (n.d.). https://stackoverflow.com/. Accessed: 2021-09-15.

Tutorial Point Java Programming (n.d.). https://www.tutorialspoint.com/java/index.htm. Accessed: 2021-09-15.

You Tube Java GUI (n.d.). https://www.youtube.com/watch?v=Kmgo00avvEw. Accessed: 2021-09-15.

A Appendix

1. ALM: Application Lifecycle Management: I have maintained various stages of a software application during my software development.

2. Conceptual Class Diagram: I have analysed the requirements of the Greenwich cab company and described the UML which includes various class attributes, operations and relationships among the related classes.

3. Database Connection: In my case, I have used MySQL database for storing data and below the major code of my database connection and here I put use as root and null password that we can secure by putting username and password. static final String DATABASE$_URL =$

$”jdbc : mysql : //localhost/greenwich_cabs”;$
$staticfinalStringDATABASE_USER = ”root”;$
$staticfinalStringDATABASE_PASS = ””;$
$staticConnectionconnection = null; //managesconnection$
$staticStatementstatement = null; //querystatement$
$staticResultSetresultSet = null;$
$privateObjectrollField;$
$privateObjectbrk;$
$Connectioncondb;$

void getConnection() {

try {

connection = (Connection)
DriverManager.getConnection(DATABASE$_URL, DATABASE_USER, DATABASE_PASS);$
$statement = (Statement);$
$connection.createStatement();$
$System.out.println(”ConnectHoise...”);$

} catch (SQLException e) {
System.err.println(”Connection Error”);
loginPage ob = new loginPage();
ob.setVisible(true);

}
}

4. GUI Design: I used NetBeans swing to design the graphical user interface which produces the most interactive GUI.

5. Unit test: In my case, I have analysed manually all the input fields for the validation test and checked the individual all components of my software.

6. Expected result calculation: I applied nested query for retrieving the data from the database. After getting the driver's information for a particular data I calculated. Here, I write major part of calculation of my code:

7. double tempJob = Double.parseDouble(job);
double tempFare$_cost = Double.parseDouble(fare_cost);$
$doubletempTips_cost = Double.parseDouble(tips_cost);$
$TempTotal_fare = tempTips_cost + tempFare_cost;$
$total_job = (int)(total_job + tempJob);$
$StringexpectedPerDriver = df.format((0.20) * TempTotal_fare);$
$total_fare = total_fare + TempTotal_fare;$
$if(tempJob > 0)\{$
$ObjectrowData[] = driver_id, TempTotal_fare, expectedPerDriver, job;$
$DefaultTableModelmodel = (DefaultTableModel)dataTable.getModel();$
$model.addRow(rowData);$

$\}StringexpectedOnTotal = df.format((0.20) * total_fare);$

$System.out.println(total_fare + ”TempTotal_fare” + TempTotal_fare);$

$totalMoney.setText(String.valueOf(total_fare) + ””);$

$totalJobs.setText(String.valueOf(total_job));$

$expected.setText(String.valueOf(expectedOnTotal) + ””);$

8. In my project, I used MySQL database. Threfore, it is highly recommended to used Xammp software. First start the Xammp server.Next, by importing my given database file ”greenwich$_cabs.sql”, usercanusethesoftware.$