ICT Engineering

SEP 1Y A16

Project Report

Group 1

Andreea Buturca 253691

Karolina Beliharova 253810

Marek Lowy 253652

Martin Janosik 253934

Supervisors

Steffen Vissing Andersen

Birgitte von Fyren Balsløv

Date: 16. 12. 2016

**Table of contents**

[1 Abstract 5](#_Toc469412647)

[2 Introduction 6](#_Toc469412648)

[3 Analysis 7](#_Toc469412649)

[3.1 Requirements 7](#_Toc469412650)

[3.2 Use Case Diagram 9](#_Toc469412651)

[3.3 Use Case Description 11](#_Toc469412652)

[3.4 Activity Diagram 12](#_Toc469412653)

[4 Design 13](#_Toc469412654)

[4.1 Graphical User Interface (GUI) 13](#_Toc469412655)

[4.2 Database Design 14](#_Toc469412656)

[4.3 Model class diagram 15](#_Toc469412657)

[4.4 Class diagram 16](#_Toc469412658)

[4.5 Sequence diagram 17](#_Toc469412659)

[5 Implementation 18](#_Toc469412660)

[6 Testing 19](#_Toc469412661)

[6.1 Testing methods according to Use Case 19](#_Toc469412662)

[6.2 Test of GUI 19](#_Toc469412663)

[7 Results 20](#_Toc469412664)

[8 Conclusion 21](#_Toc469412665)

[9 References 22](#_Toc469412666)

[9.1 Books 22](#_Toc469412667)

[9.2 Additional resources 22](#_Toc469412668)

[10 Appendices 23](#_Toc469412669)

[10.1 Appendix 1 – Activity diagram 23](#_Toc469412670)

[10.2 Appendix 2 – Use Case Description 27](#_Toc469412671)

[10.3 Appendix 3 – Class Diagram 34](#_Toc469412672)

[10.4 Appendix 4 – User Guide 35](#_Toc469412673)

**List of figures and tables**

[Figure 1: Priority Use Case Diagram 9](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412891)

[Figure 2: Class Diagram 16](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412892)

[Figure 7: Class Diagram 34](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412893)

[Table 1: Manage buses 27](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412894)

[Table 2: Register a new chauffeur 28](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412895)

[Table 3: Create tour 29](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412896)

[Table 4: Make reservation 30](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412897)

[Table 5: Search Reservation/Tour 31](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412898)

[Table 6: Edit Reservation/Tour 32](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412899)

[Table 7: Tours Overview 33](file:///C:\Users\Karolina\Documents\GitHub\SEP1\Documentation\ProjectReport.docx#_Toc469412900)

# Abstract

*VIA Bus is a company located in Horsens, Denmark with Trip Driver as the manager. At VIA Bus you can either rent a bus with a chauffeur – driving to a destination of your choice, or travel by bus to one of the predefined locations, i.e. a few European countries and some Danish sites and events.*

*VIA Bus is need of some kind of system to keep track of the tours, chauffeurs, bus routes and customers, and therefore the manager Trip Driver decided to contact us to create the system for him. System based on Java must meet the requirements of the customer.*

# Introduction

The project presented to us is based on an interview with VIA Bus Company having wishes for their booking system. VIA Bus is a company that offers different types of tours for customers. They organize predefined tours or make a new trip by renting a bus with a chauffeur.

The program must meet their requirements, run on Java, which means it has to be used on various platforms. Simple and usability tested GUI have to be created to make sure the employees will have no problems using it. It must be single user system, running on one single front desk.

The VIA Bus company wants a user system that will be easy and clear for using. We need to create simple, but powerful tool that will handle every aspect which the employees of the company might encounter. The employee wants to have an easy way to reserve tour. The employee also wants to store information like customer and chauffeur data in a database. The employees should be able to find available chauffeurs and busses and assign them new trips according to their wishes. The system will automatically calculate prices for each reservation according to travel type, extra services etc. All data must be saved in the computer using database which will be easy to back up in case of losing data.

# Analysis

## Requirements

1. The employee should be able to search through the list of destinations for the two kinds of tours: standard trips and travels, and private trips and travels with a private bus and a chauffeur.
2. The employee should be able to see all available seats for each tour.
3. The employee should be able to make a reservation for available seats for given destination, date, and time.
4. The employee should be able to see details about reservation as price, extra services, discount, amount of passengers, customer’s name, address, email, phone number, the number of points given to each customer, passenger’s name, birthday, destination, departure time and date, bus type and chauffeur’s dates.
5. For private customers, the employee should be able to register the name, address, the passenger’s birthday, and when it is permitted the email address for the newsletter.
6. For companies, the employee should be able to register the name, address, phone number of the person that is making the reservation and the company’s name.
7. The employee should be able to identify the profile customer: company or private person, including at least name, address, email, and phone number.
8. The employee should be able to find data about customer in the list of customers, to check the number of points and if it’s a frequent customer.
9. The employee should be able to register chauffeurs by: name, address, email, phone, date of birthday, employee ID, preferences for tour distance: long, medium, short, for bus type: classic, mini, party luxury.
10. The employee should be able to identify the chauffeurs with full time contract and vikars.
11. The employee should be able to see all the details about the chauffeurs in the list of chauffeur.
12. The employee should be able to register all the busses by selecting the type: classic, mini, party luxury, registering the number plates and number of seats per bus.
13. The employee should be able to see all the details about the busses in the list of busses.
14. The employee must be able to register the hour, date and place for the tours’ arrival, departure and stops.
15. The employee should be able, in creating tour, to select which bus is available for next tour, selecting in the list of busses one bus type.
16. The employee should be able to find an available chauffeur after the suitable bus was found and time and date for the tour were chosen.
17. The employee should be able to search through the list of chauffeurs: full time or vikars.
18. The employee should be able to hire a bus-and-chauffeur, for a private tour, after searching through the list of tours.
19. The employee should be able to add extra services such as: food, accommodation, tickets, suitable for the private tours, according to customer requirements.
20. The employee must be able to find, edit or remove the tours.
21. The employee must be able to find, edit or remove the reservations.
22. The employee should be able to add additional notes for each reservation, in order to set the correct price for each customer.
23. The employee should be able to set the prices for extra services for each reservation according to passenger’s age.
24. The employee should be able to set a lower price for extra services for persons under 18.
25. The employee should be able to reward and give discounts to customers before setting the total price.
26. The employee should be able to save reservation if there are some changes.
27. The system stores all the data in database.
28. The system must be implemented in Java.

## C:\Users\Karolina\Desktop\SEP\usecase.PNGUse Case Diagram

Figure 1: Priority Use Case Diagram

**Tours Overview:** All tours will be shown.

**Create tour:** Employee is asked to enter information about new tour: date, time, departure, destination, distance, he has to choose bus from the bus list and chauffeur from the chauffeur list, fill in extra services, price, bus stop and time for break. If it is private tour, employee has to fill data about customer, otherwise tour is added to the list.

**Register a new chauffeur:** Employee is asked to enter information about new chauffeur: name, address, phone number, email address, employee ID (5-digit number), and type of contract: full-time or vicar. If it is full-time, employee specify wishes for trip and the chauffeur is added to the list.

**Manage buses:** Employee is asked to enter information about new bus: type, registration plate and number of seats and the bus is added to the list. Employee is able to remove bus from the list.

**Make reservation:** Employee can make a reservation.

**Find Reservation/Tour:** Employee can search reservation by customer data: name / company name, address, email and phone number. Employee is able to remove reservation. Employee can find tour search by tour information: destination, departure, date. Employee is able to remove tour from the list.

**Edit reservation/tour:** Employee can change some information about reservation/tour.

## Use Case Description

****“Create tour” is the most important part of the system because the purpose of this project is to make a program that rents a bus with a chauffeur driving to a destination of customer’s choice, or travel by bus to one of the predefined locations. The documentation for all other classes and Use Cases can be found in the appendices. The following pages of the documentation (sequence diagram, activity diagram, and implementation) will be orientated to the “Create tour” use case.

Table 1: Create Tour

## Activity Diagram

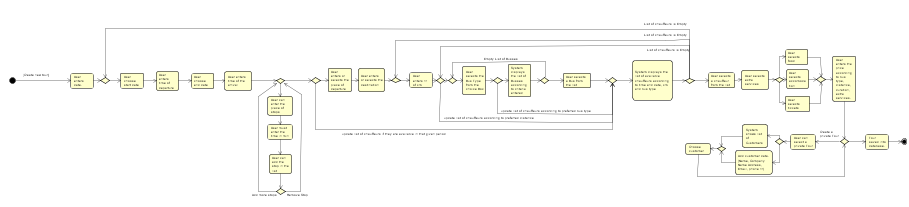
****The Create tour activity diagram displays step-by-step the process of creating new tour.

Figure 2: Create Tour

# Design

## Graphical User Interface (GUI)

## Model class diagram

This basic model shows us the structure of the system. “Bus” class is used to create new vehicle in the system and information about it and through this class we get information about bus type from classes “LuxuryBus, PartyBus, ClassicBus, MiniBus”. “Bus list” class stores all busses and information about them. “Person” class is extended by classes “Passenger, Customer, and Chauffeur” and is used to store data about a person. “ChauffeurList” is used to hold a list of chauffeurs and it can split them according to criteria. “CustomerList” is used to hold a list of customers and split them according to criteria and the same is for “PassengerList” with passengers. Every object of class “Reservation” stores information about each reservation. “ReservationList” is used to hold a list of reservation and it can filter them by various criteria. Information in class “Destination” is about place and stops and it is stored to “DestinationList”. In class “Trip” is created new tour and all data is saved to “TripList”. “Main” class contains main functions of our system. “DataHandler” is used for connecting database with the “Main” class and help us to have fixed connection with database when we do some changes like add bus/reservation/chauffeur.

## Class diagram

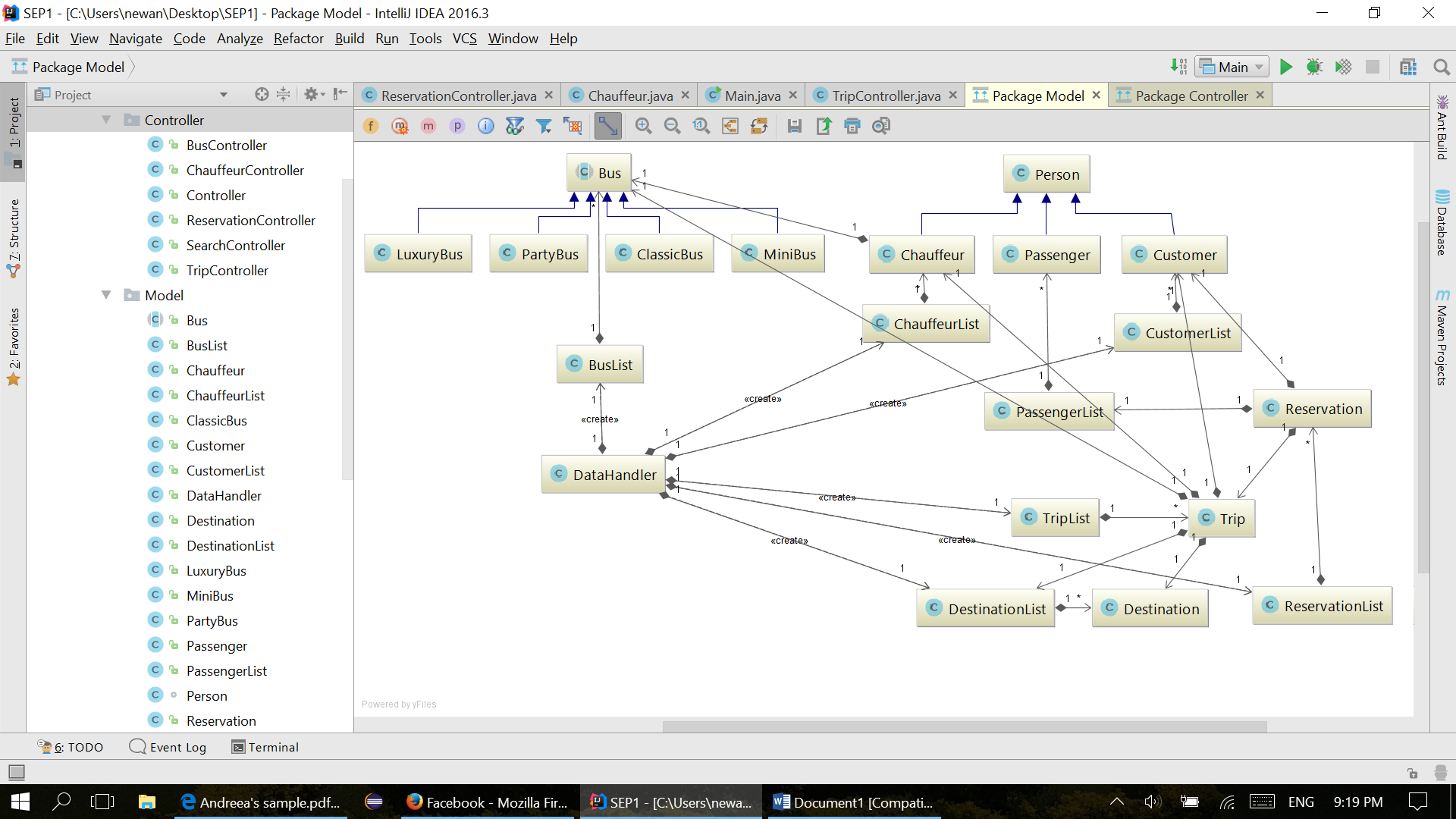
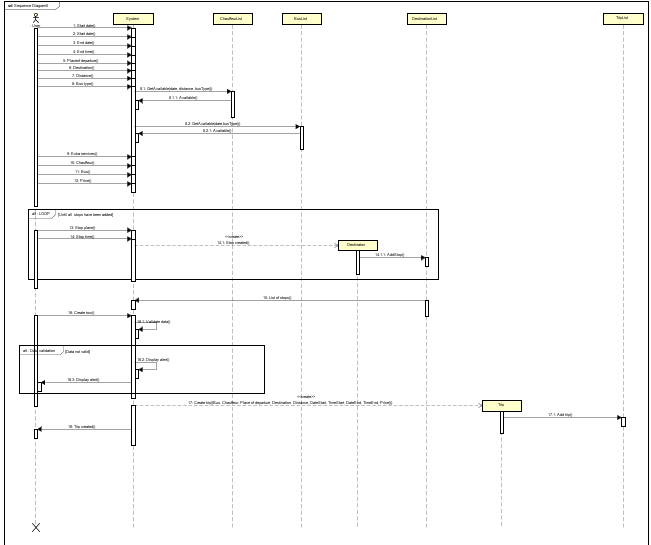


Figure 3: Class Diagram

## Sequence diagram



# Implementation

# Testing

## Testing methods according to Use Case

## Test of GUI

# Results

# Conclusion

Main goal of our project was to create software for VIA Bus company having wishes for their booking system. With VIA Bus company you can rent bus and chauffeur for a rent that means driving to destination of customer’s choice, or travel by bus to one of the predefined locations. Program is prepared for easy and clear using and can be used by different actors for different actions. Our goal was create program that complies all customers‘ requirements. Every Use Case description and Activity Diagram were checked and compared to the program. This program is comprehensive, fast and user does not need any specific computer knowledge.

During working on project some problems came. We had mainly problems with Java code and different types of errors in program. We always needed to decide how it should work and agree on it together.

As a conclusion, project is successfully completed and our system complies all the requirements. We created powerful software that will be modest for using. The final version of the program is designed and created in unique way for specific customer.

# References

## Books

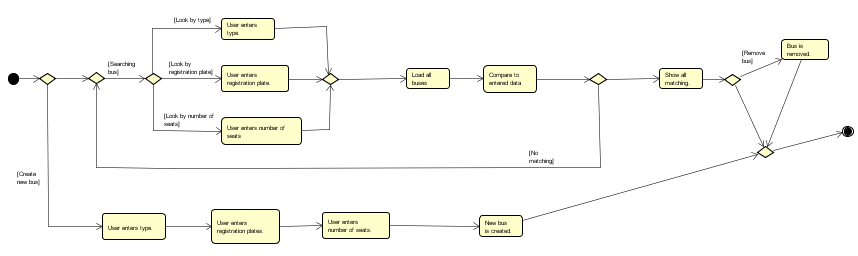
## Additional resources

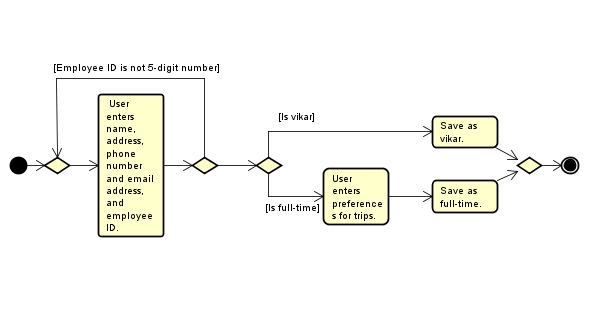
Presentations from classes

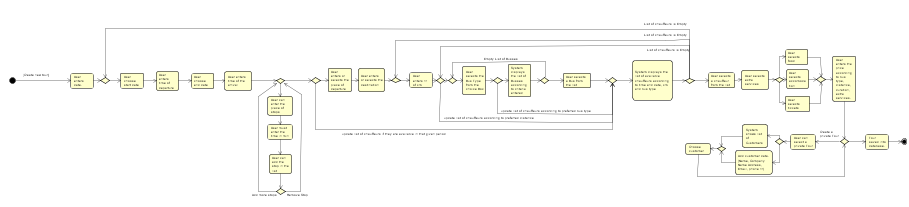
JAVA 1 and UML (SDJ1)

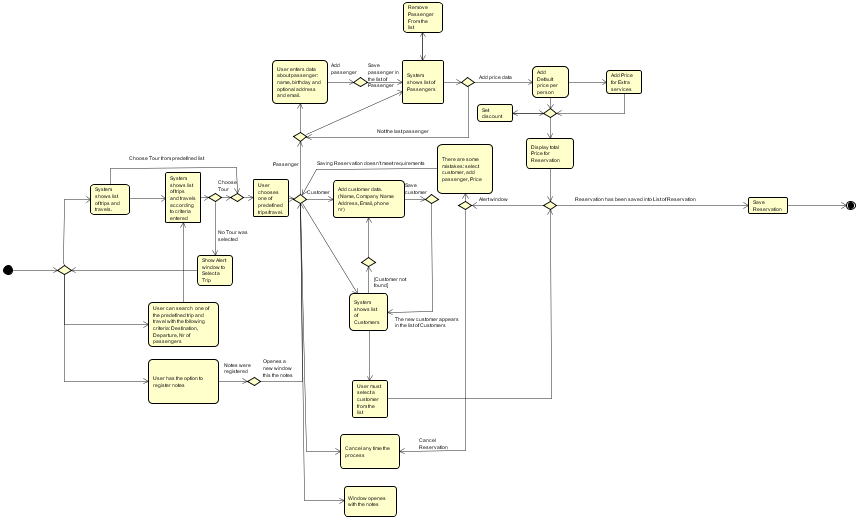
# Appendices

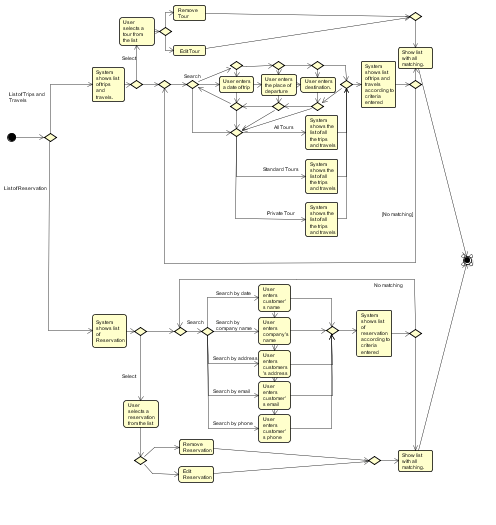
## Appendix 1 – Activity diagram

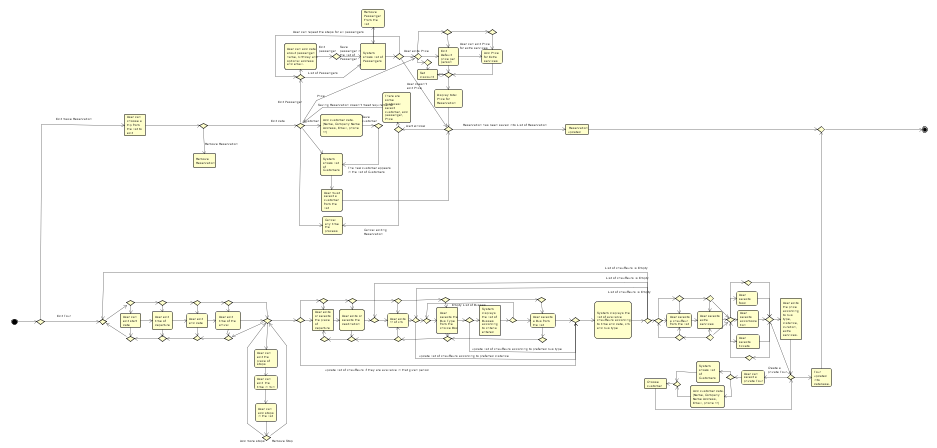
**Manage buses:**

**Register a new chauffeur:**

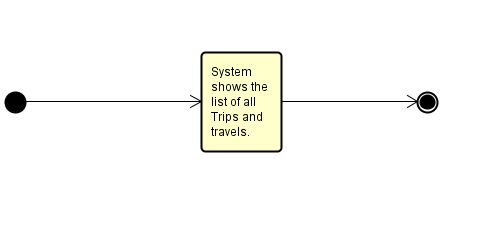
**Create a tour:**

**Make reservation:**

**Search Reservation/Tour:**

**Edit Reservation/Tour:**

**Tours Overview:**

****

## Appendix 2 – Use Case Description

**Manage buses:**

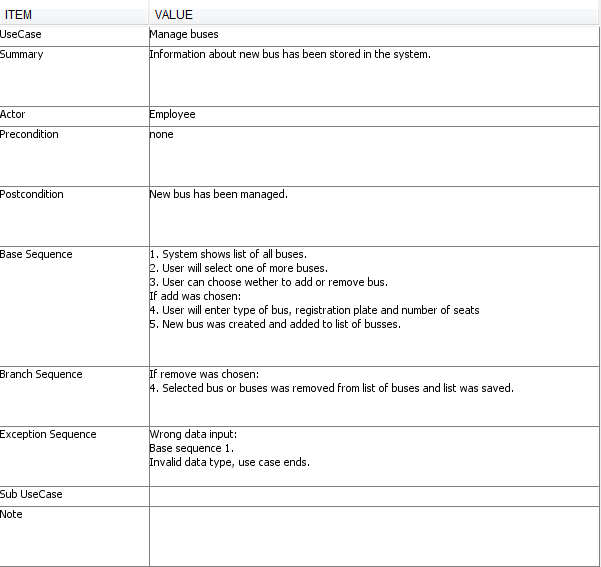


Table 2: Manage buses

**Register a new chauffeur:**

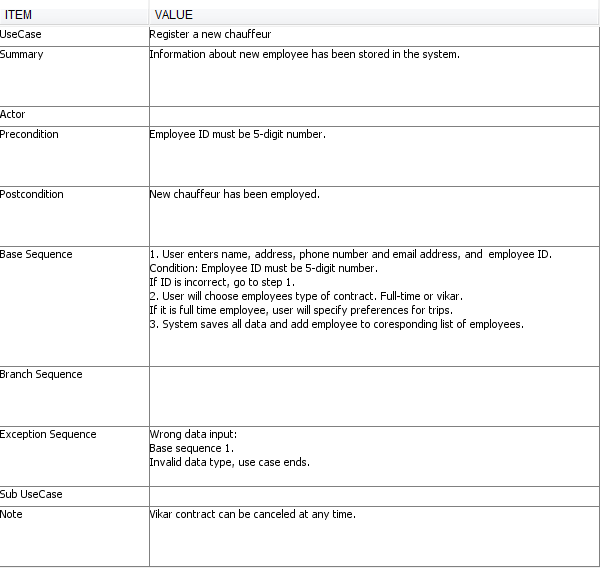
****

Table 3: Register a new chauffeur

**Create a tour:**

****

Table 4: Create tour

**Make reservation:**

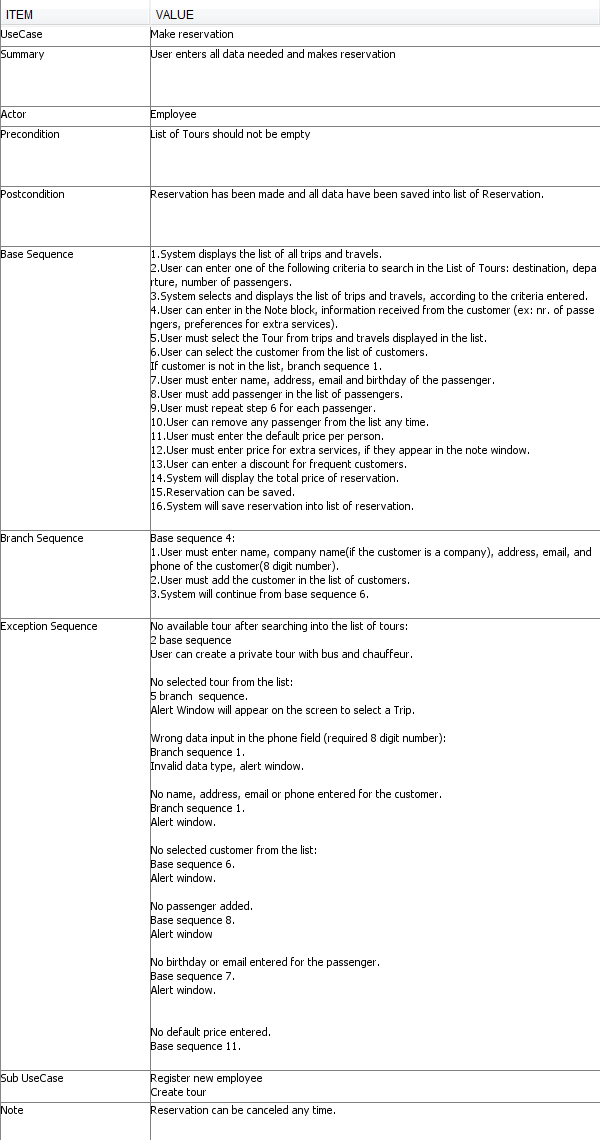
****

Table 5: Make reservation

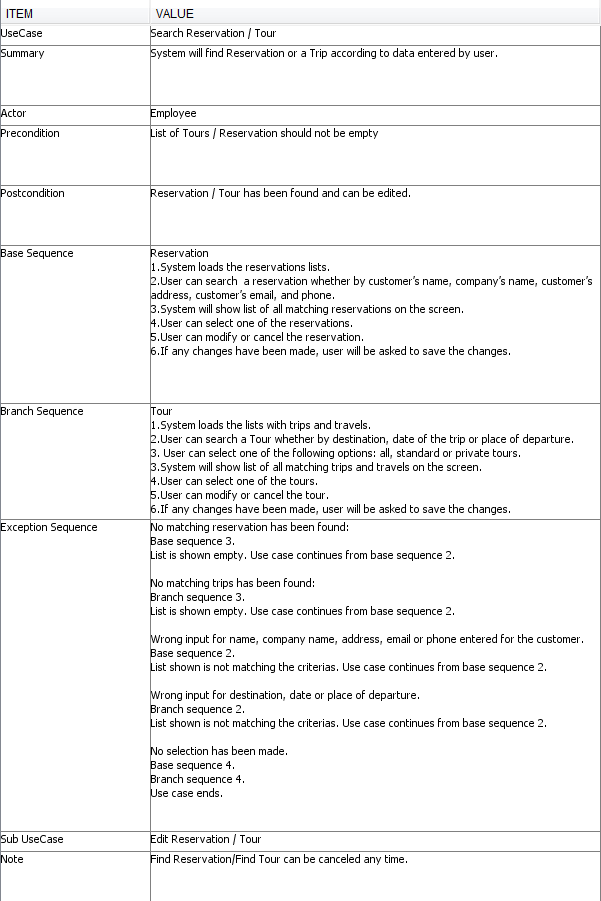
**Search Reservation/Tour:**

Table 6: Search Reservation/Tour

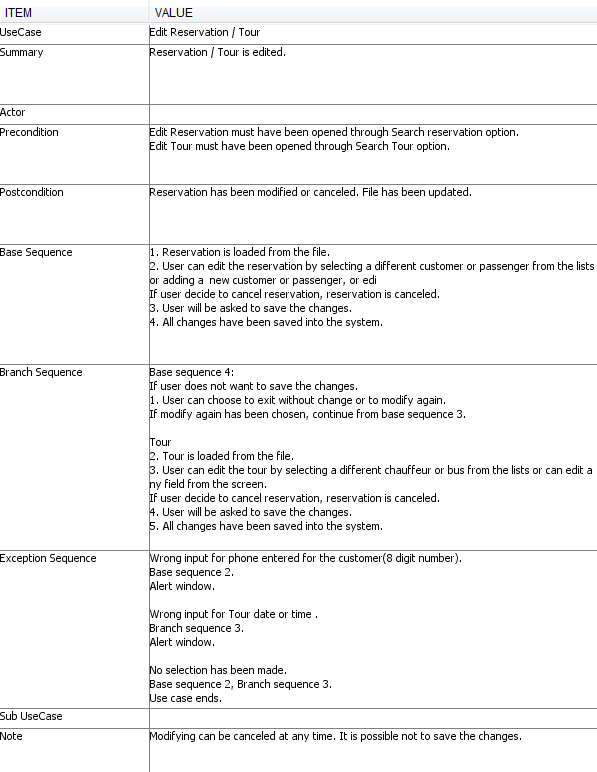
**Edit Reservation/Tour:**

Table 7: Edit Reservation/Tour

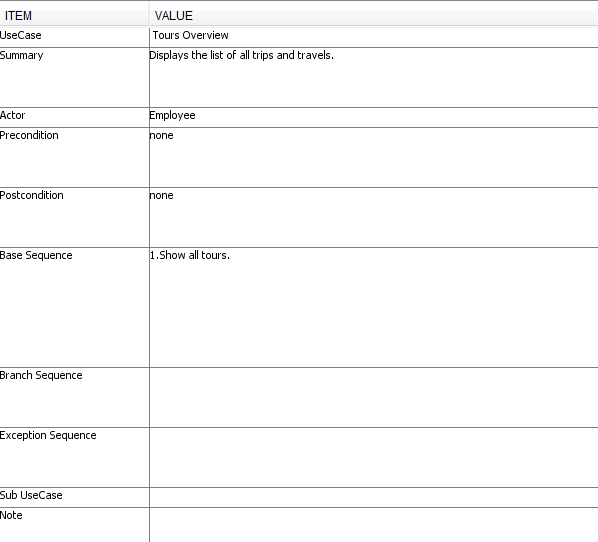
**Tours Overview:**

Table 8: Tours Overview

## Appendix 3 – Class Diagram

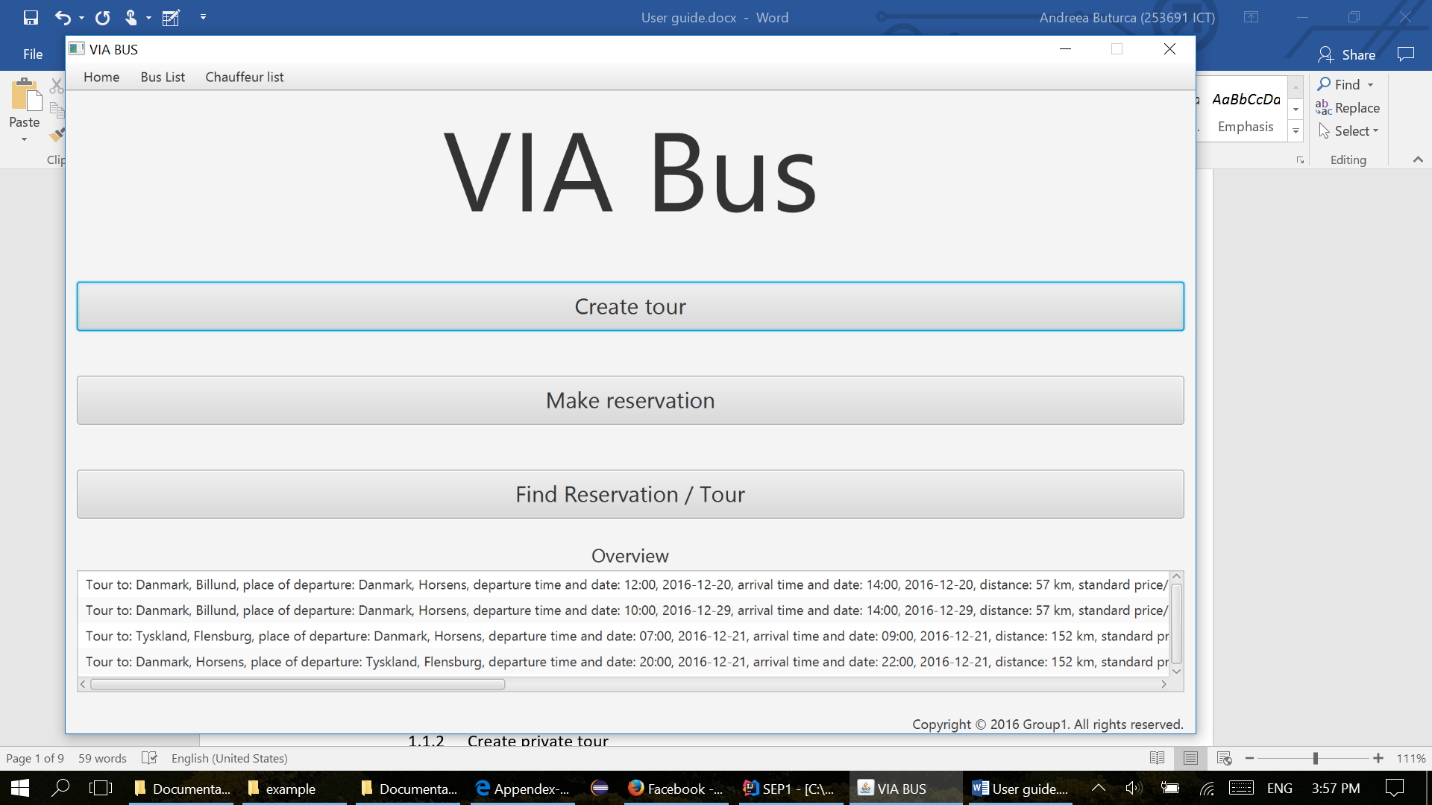
Figure 4: Class Diagram

## Appendix 4 – User Guide

**Home**

The home page is displayed when the program is opened. The Home page allows user to access Create tour, Make reservation, Find reservation / tour and displays the list of tours.

On the following pages it will be elaborated all the functions of the program.



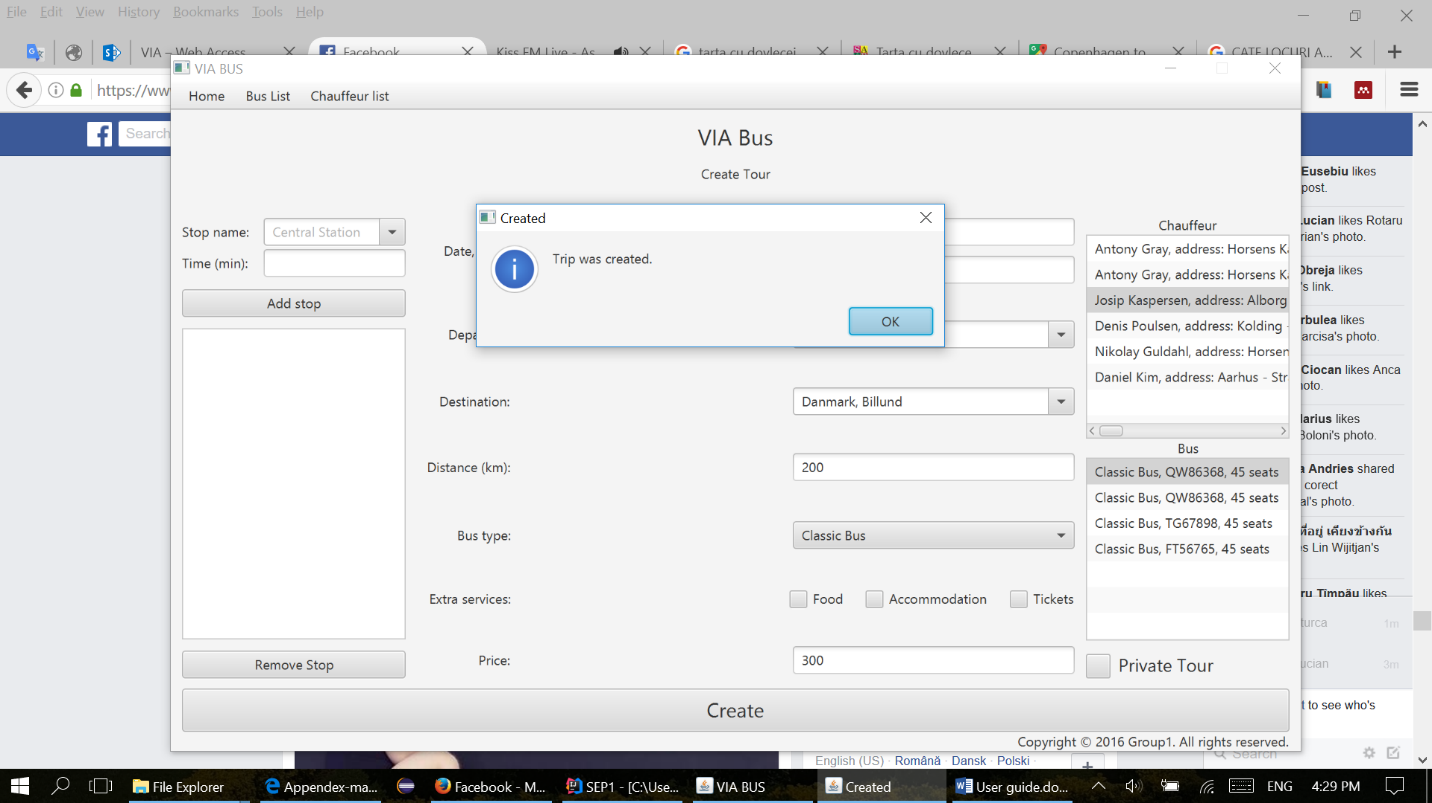
**Create tour**

**Create tour – standard**

The user must fill in all the dates about start time(format example 10:00), date of the departure (can be picked up or filled in with the following format : month/day/year: 12/01/2016), end time and date of the arrival, the departure place, destination place, distance in km, the buss type can be choosen from the list, price. The chauffeur list is sorting after user enters the date, time, and after the bus type is selected. The bus list is sorting after user selects the bus type. In order to create a new Tour, the user must select a chauffeur and a bus from the lists. In case one of the fields is not filled in, an alert window will be displayed on the screen with the name of the field that should be filled in properly

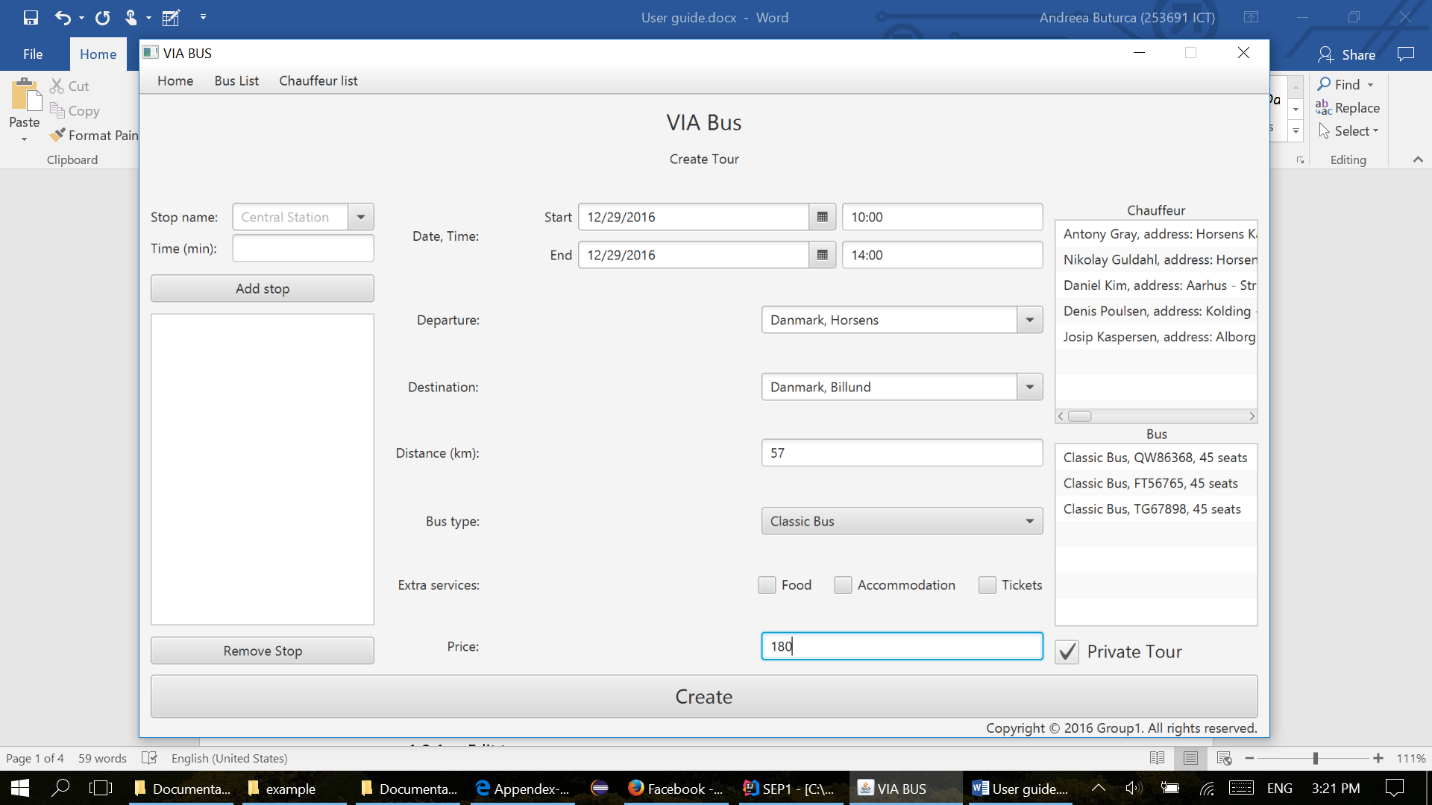
The check boxes for extra services and adding stops fields are not mandatory.



In order to finish creating a new tour, the user must press the Create button and an alert window will appear on the screen, like the one displayed bellow.

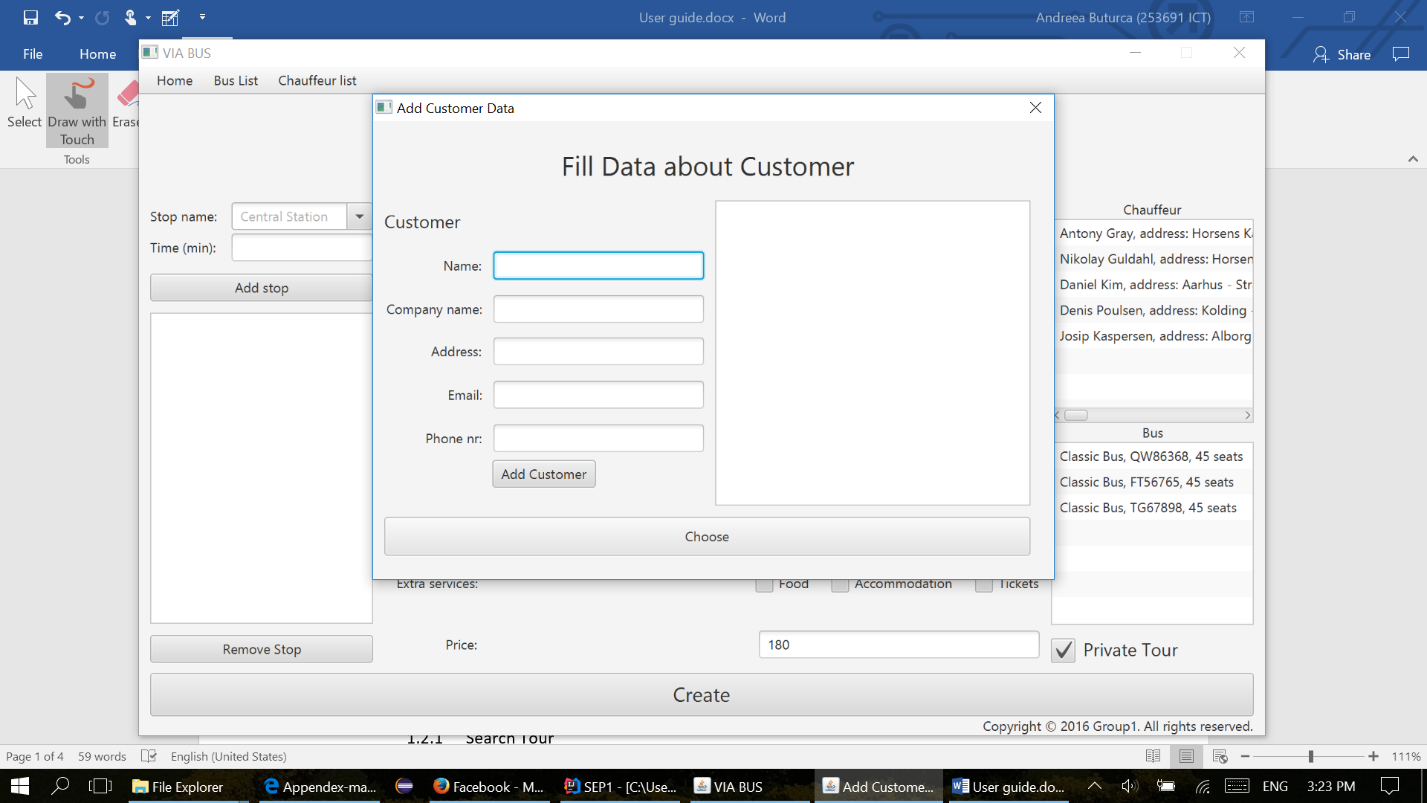
**Create private tour**

If the user wants to create a private tour, he must follow the steps from Create tour, and press at the end the button Private Tour.

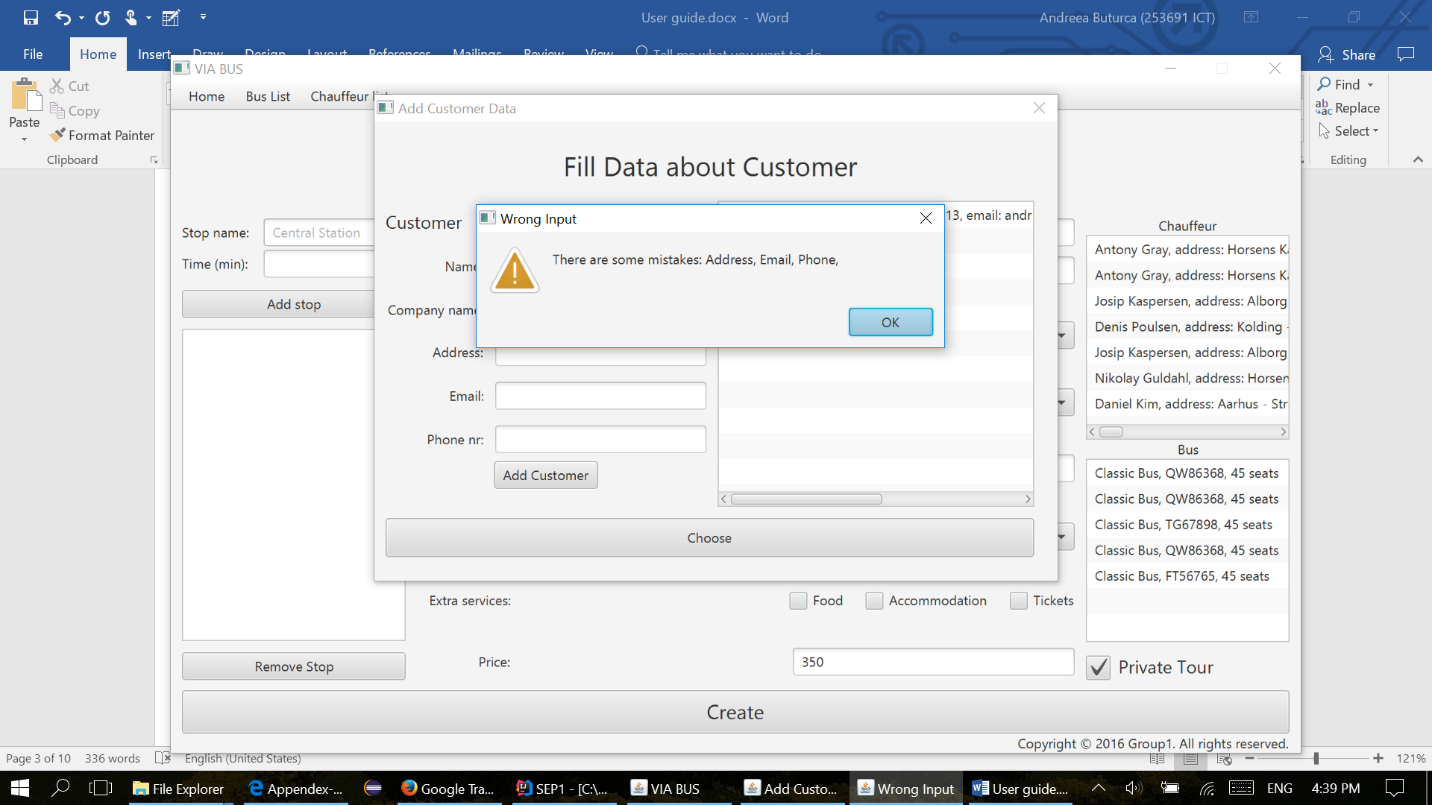




After this, another window will appear on the screen with: Fill data about customer.

This window will enable the user either to choose from the list of customers displayed on the right side of the window, or to fill in all the dates about customer: name, company name (only if the order is made by a company), address, email, and phone number

If the user doesn’t fill all the mandatory fields about the customer, an alert window will appear, like the one below, indicating which field must be filled in properly.



**Make reservation**

Make reservation appears in the Home tab, on the third place of the list.

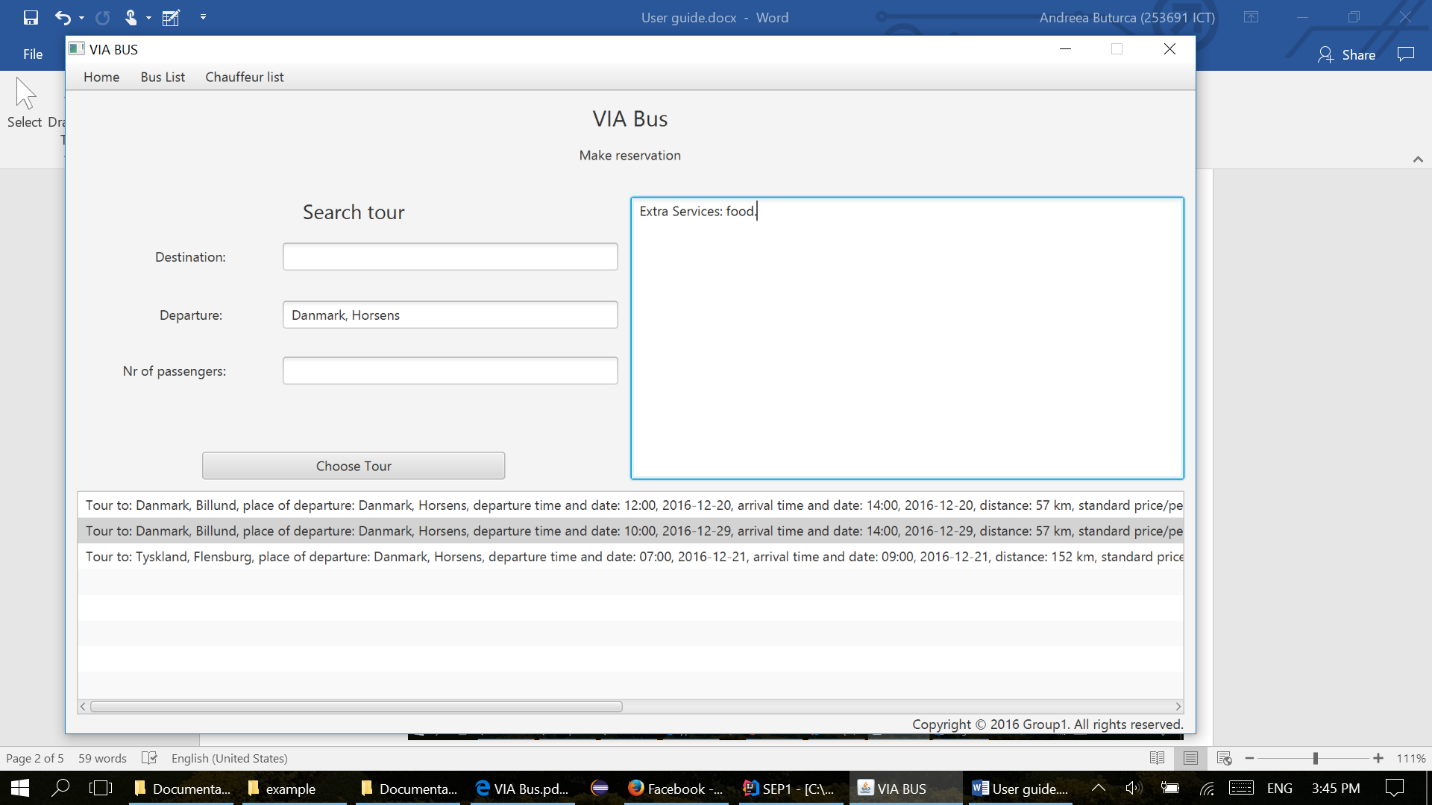
**Make reservation - Search Tour**

The screen Make reservation its split in 3 sections. First section has three fields used to search in the list of tours with the following parameters: destination, departure, number of passengers. The user can pick either one, or all the fields in order to find the best tour, according to customer requirements.

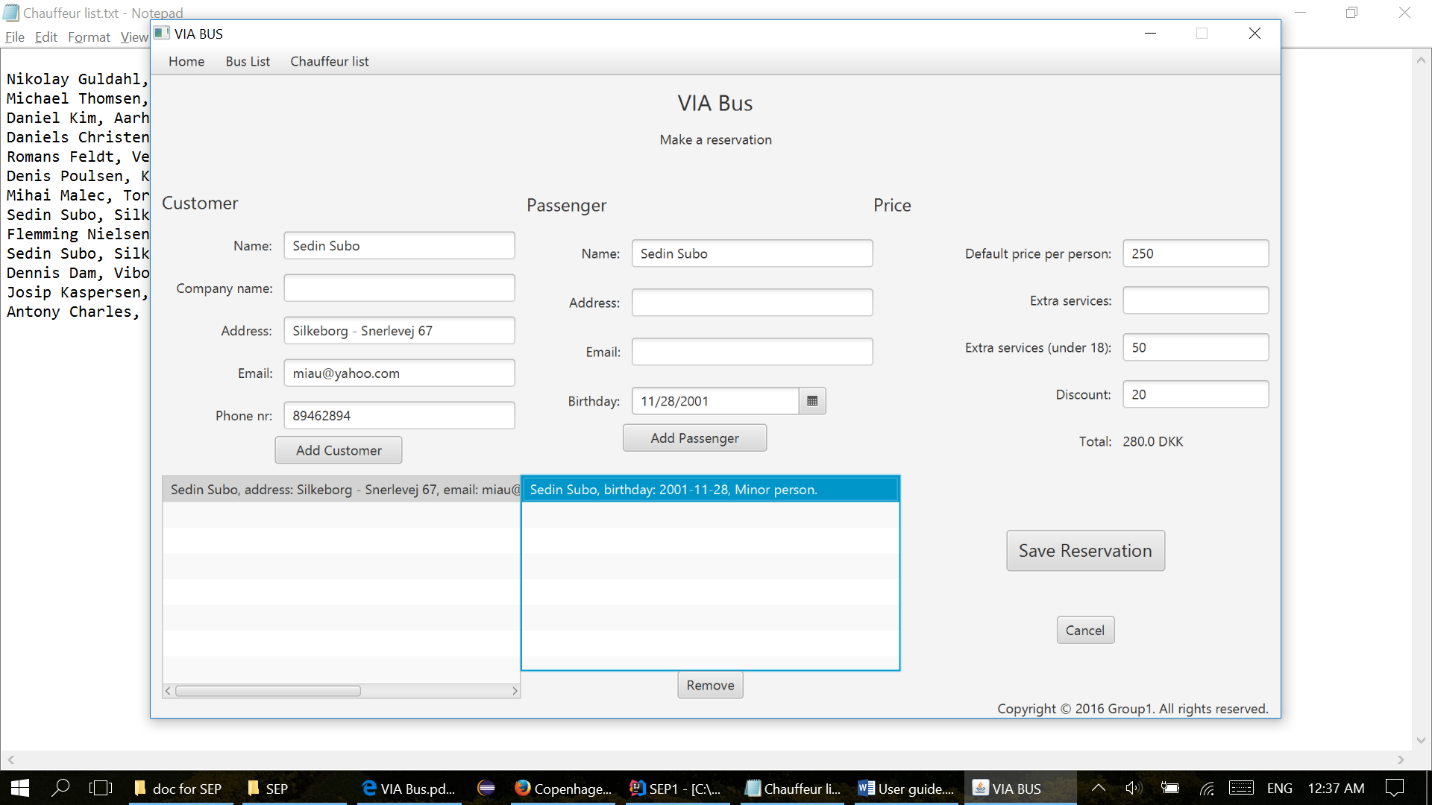
Second section, it enables the user to make notes in the right side of the screen, by simply clicking on the text field and typing in. The notes will automatically appear in the next step of making reservation, with the same data, only if the user typed into the note section. Notes are a convenient way of storing information which the user needs in the workspace.

The third section of this screen is the list of the Tours displayed in the bottom of the screen. The list is updated automatically according to the searches made by the user.

Before the user presses the button Choose tour, he must select a tour from the list.



**Make reservation**

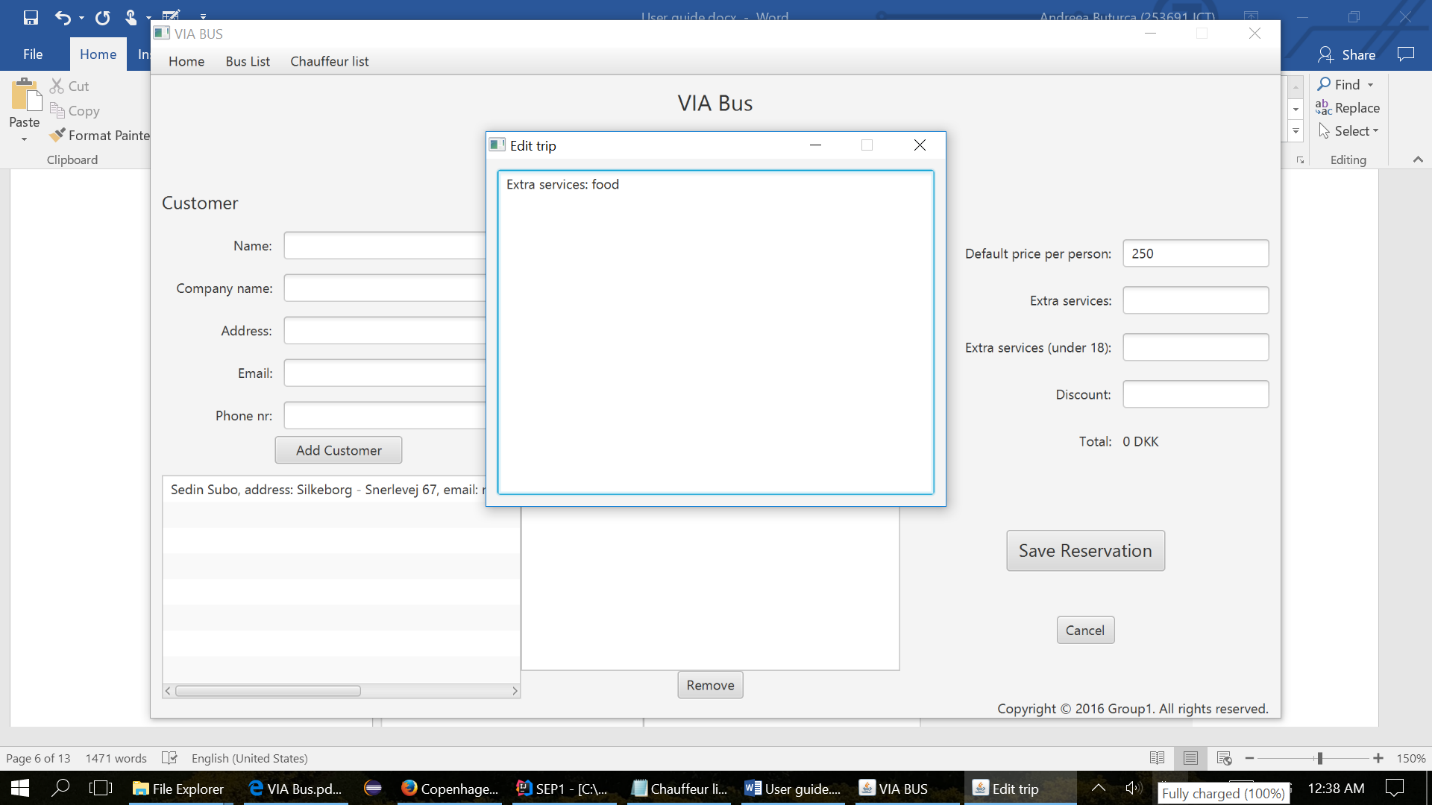
The user must must select the customer, from the list of customers, wich is dispalyed under the Customer section, otherwise he must fill in all the dates about the customer, if the customer is at the first order, with the following dates: name, company name, address, email, phone number. All the fields about customer are mandatory, the field company name must be filled in, only if the order is made by a company.

The next step in “Making a reservation” is to add the passengers, one by one, by pressing Add passenger. Each passenger will appear in the list of passengers displayed under the passenger section. The fields name and birthday are mandatory. The fields address and email are used only if the customer wants to give more details to receive the newsletter.

The next step is to fill in the price per person for extra services. Remark that the field default price per person is already filled in with the price took from the selected tour field.

The user will adjust the price for extra services according to the notes he took in step 1 of “Make a reservation”. This notes appear like in the picture displayed bellow. The user can also give a discount to the customer if necessary in the field called Discount.

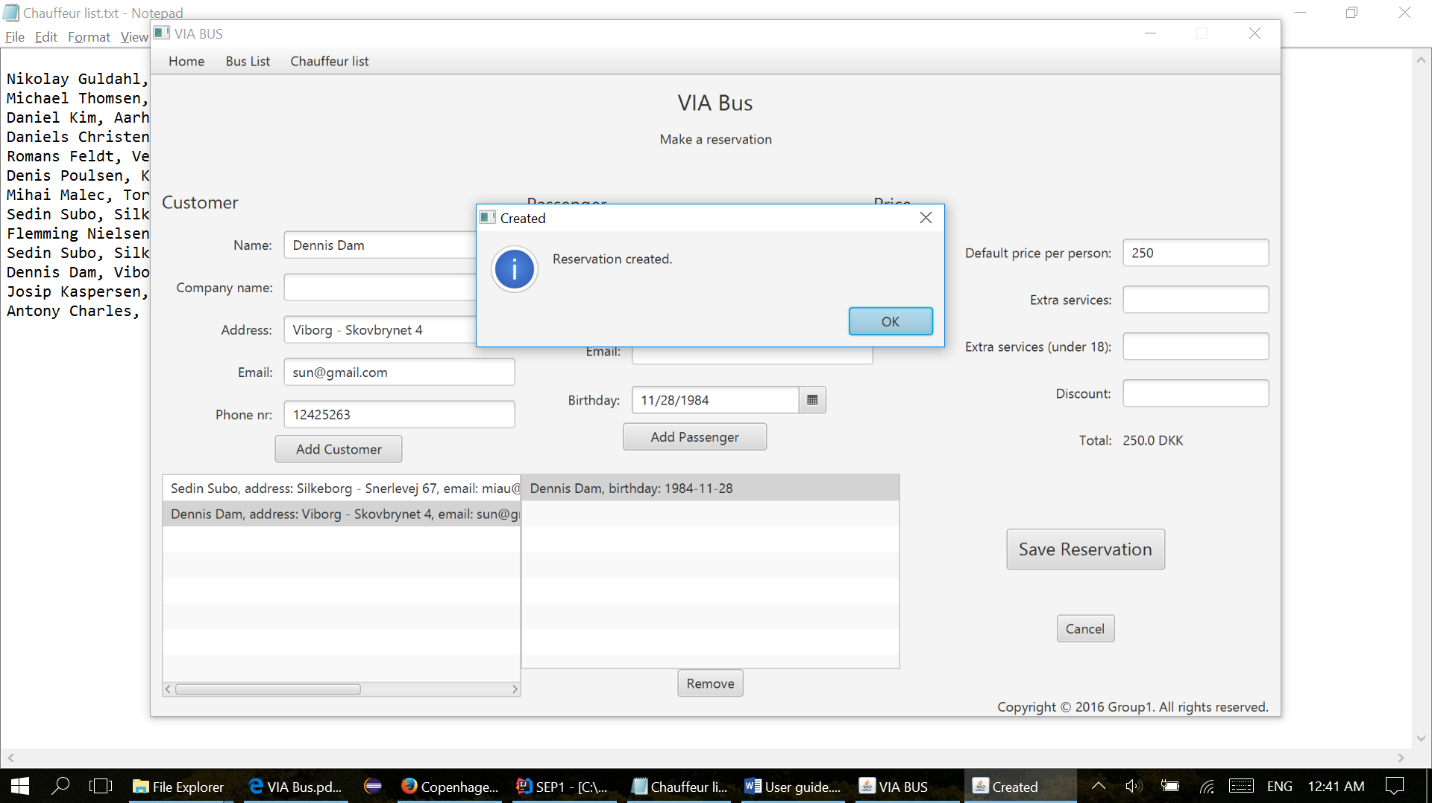
All the prices must be typed in per person and the total price of the reservation will appear in the Total section.



The user can any time press cancel reservation, and he will be directed to the home screen.

After all the steps enumerated above, the user can save the reservation, and an alert window will appear on the screen, which will confirm that the reservation was created.

The alert window will be exactly like the one displayed bellow.

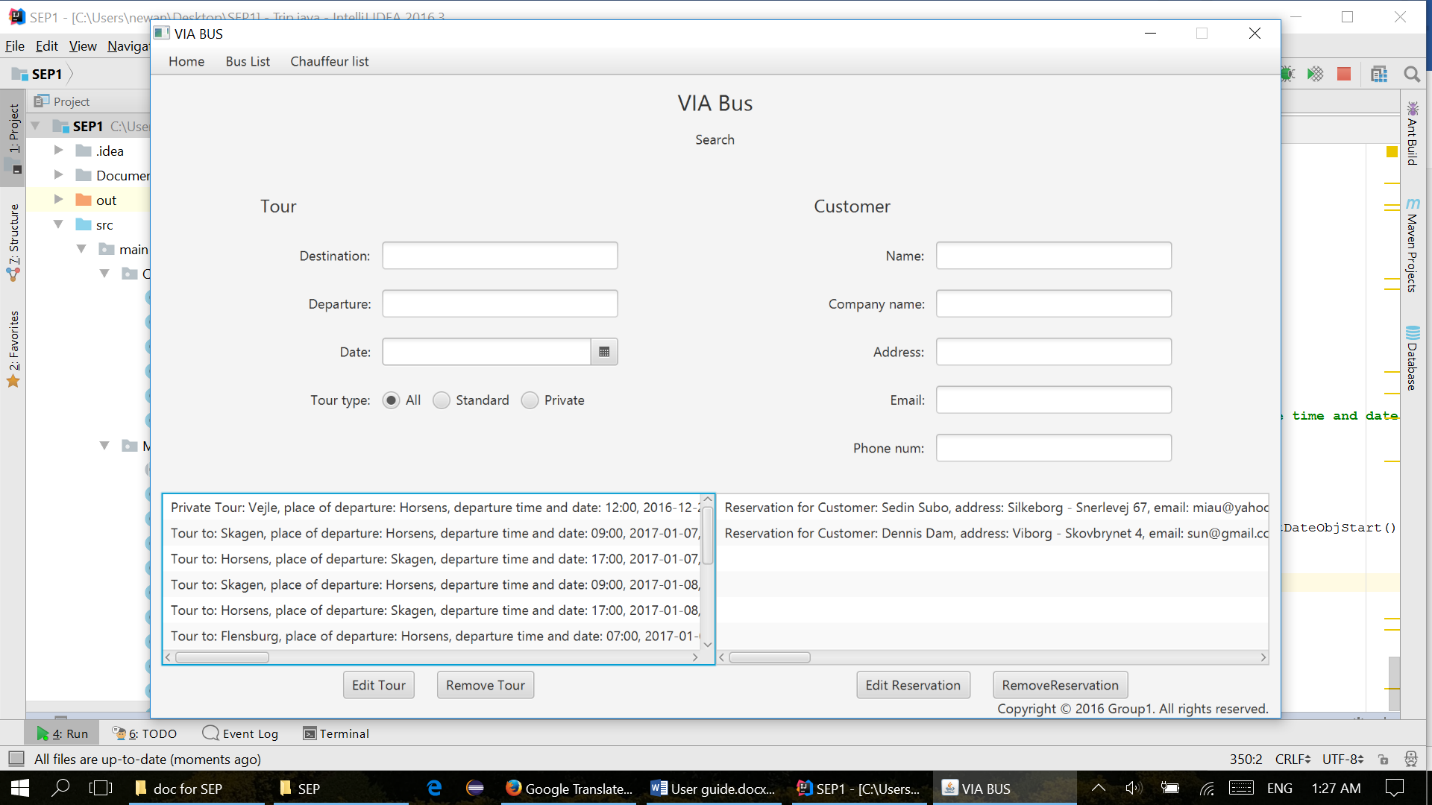


**Find Reservation/Tour**

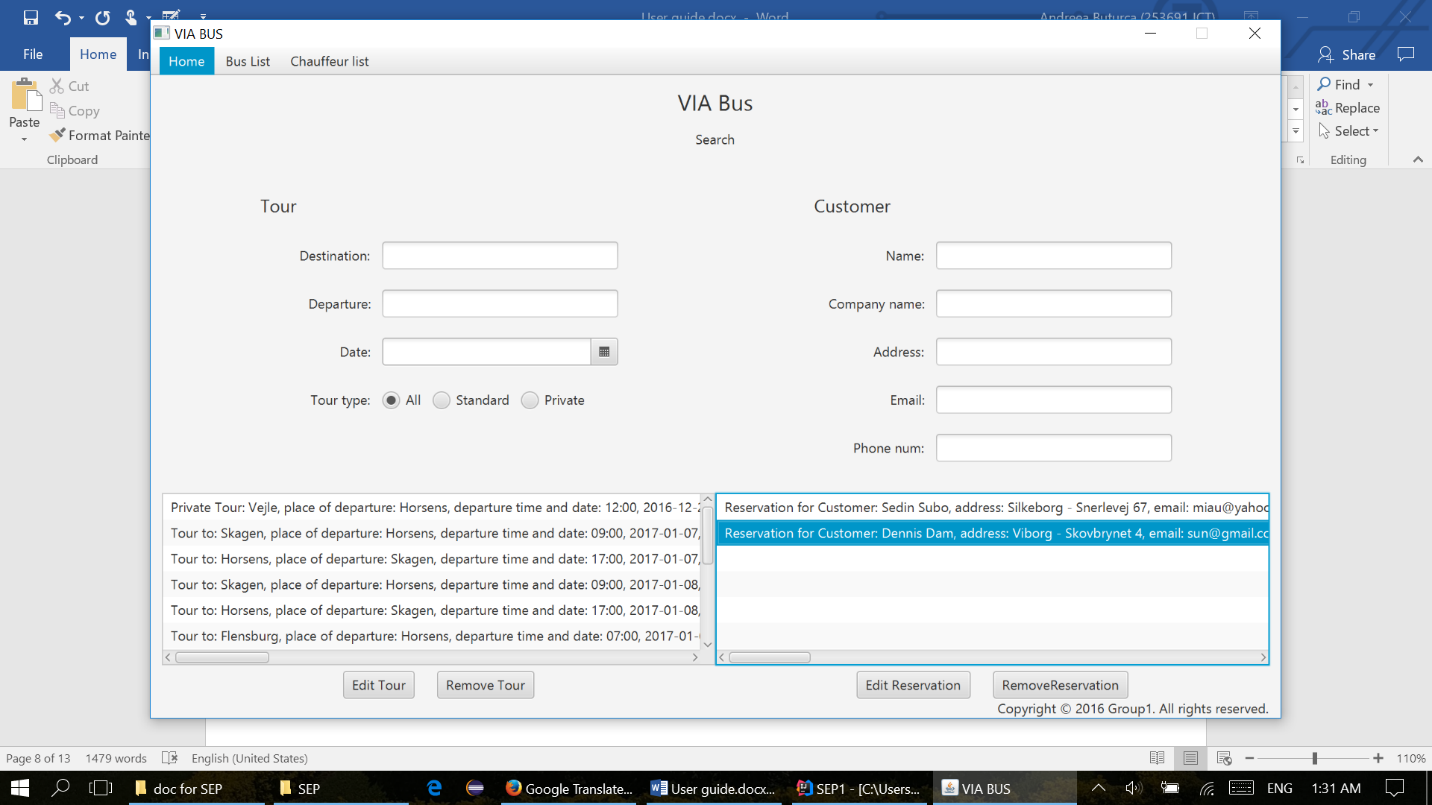
Find Reservation/Tour appears in the Home tab, on the last place of the list. The screen it’s split in 2 sections, the left half of the page, is for Tours and the right half of the page is for Reservations.

The tour section, has three fields used to search in the list of tours with the following parameters: destination, departure, and date. The user can use either one, or all the fields in order to find the tour, according to customer requirements. And, three radio buttons for all tours, standard or private tours. The list is updated automatically according to the fields used by the user.

In order to go on edit tour or remove tour, the user must select one from the list.



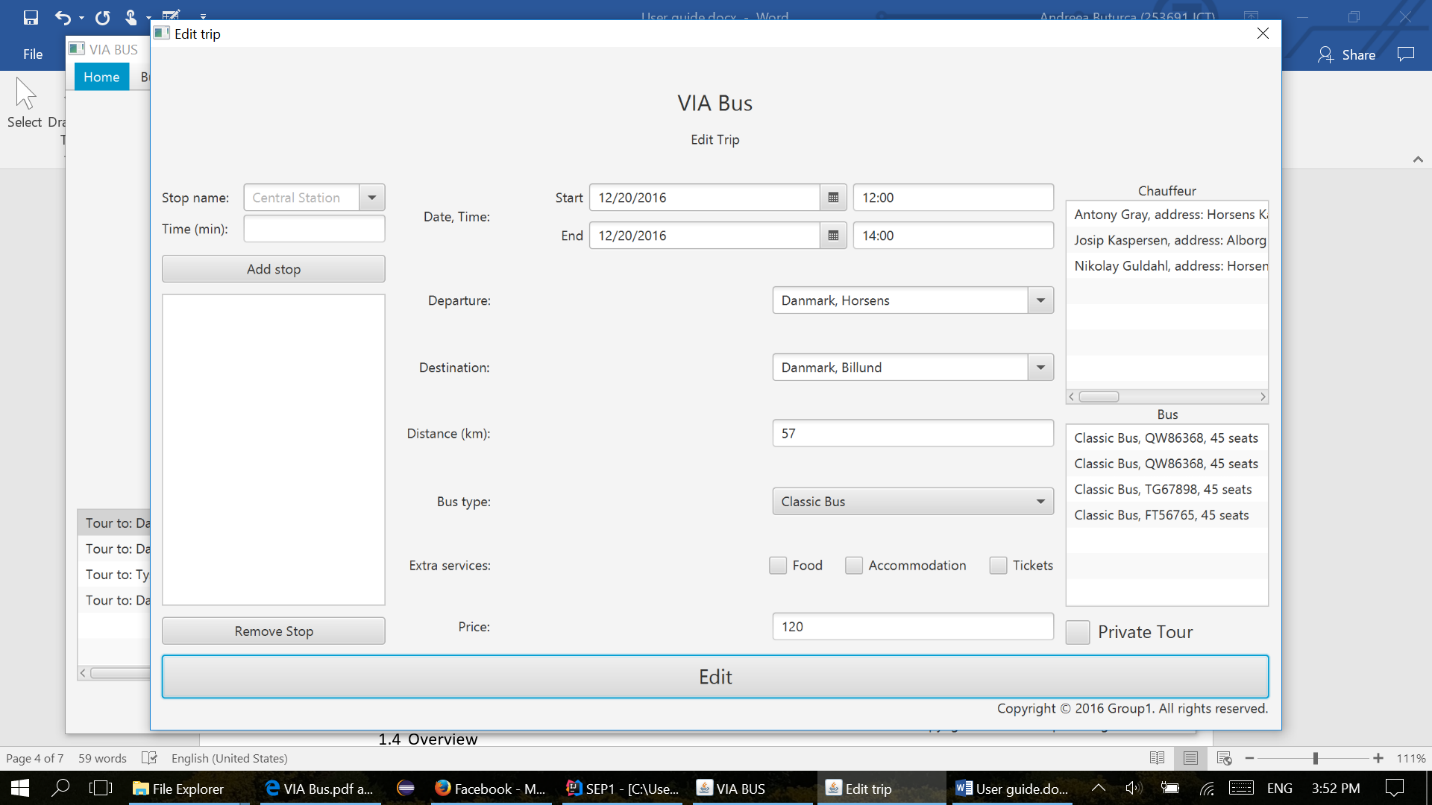
The Reservation section, has five fields used to search in the list of reservations with the following parameters about the customer that made the reservation: name, company name, address email, phone number. The user can use either one, or all the fields in order to find the right reservation. The list is updated automatically according to the fields used by the user.

In order to go on edit reservation or remove reservation, the user must select one from the list.

**Edit tour**

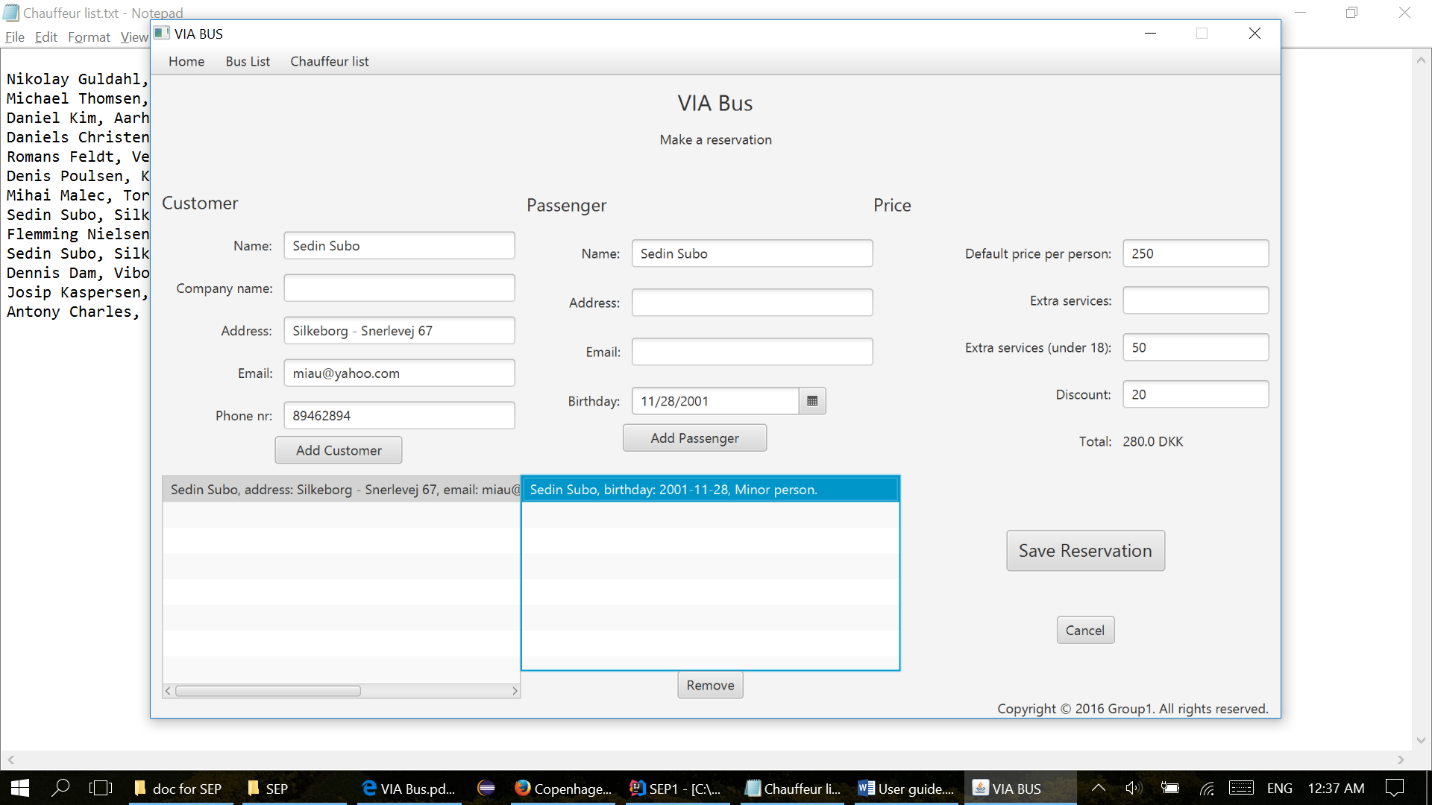
When the user selects edit tour, a new window will appear on the screen with the same interface like in create tour, but in this case, all the fields will be filled in with the dates from reservation selected in the previous step.

The user can edit any of the fields he wants to change, but in order to succeed, it is mandatory to select again the chauffeurs and the buss from the list.

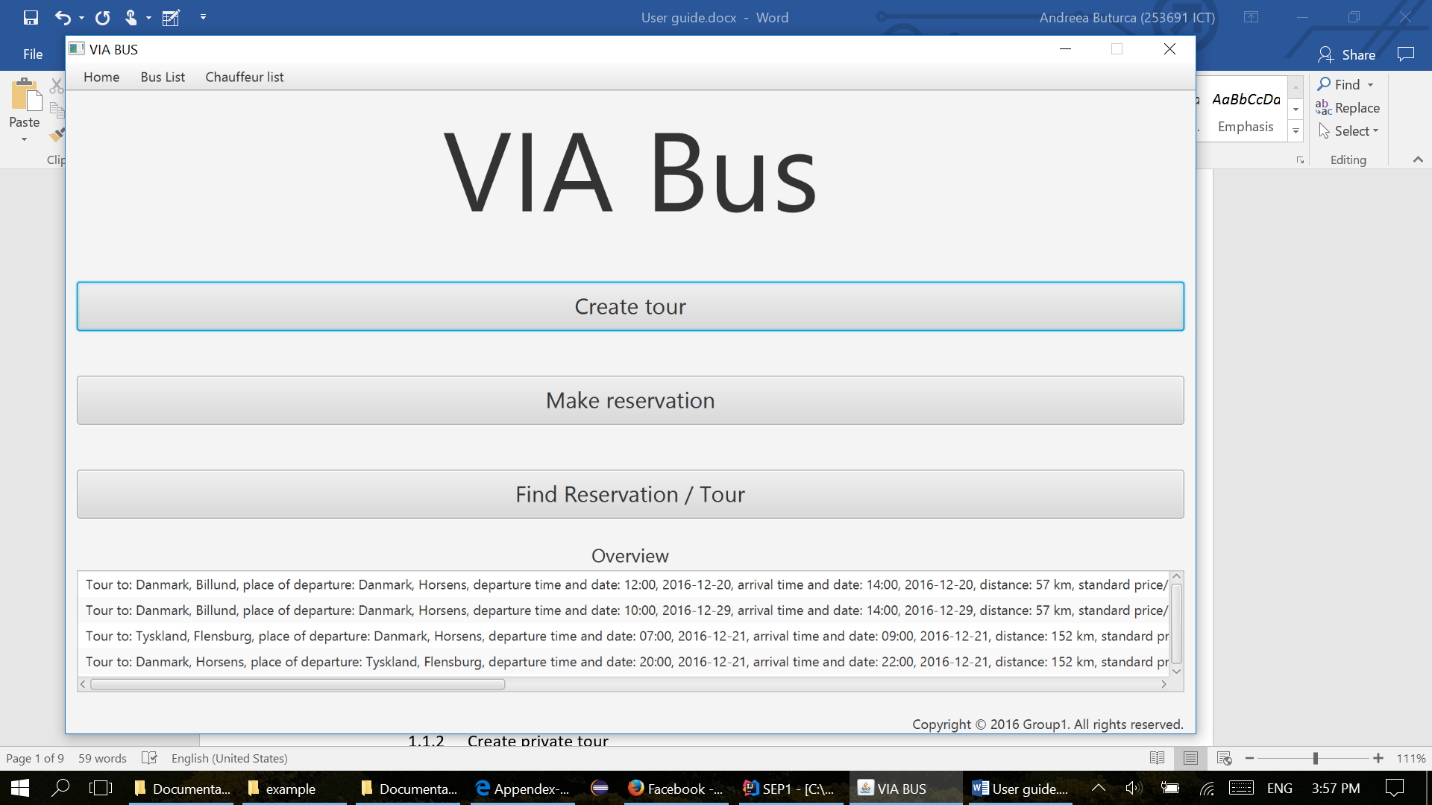


**Edit reservation**

When the user selects edit reservation, a new window will appear on the screen with the same interface like the one in Make Reservation. The user can add new customer, passengers, remove passengers or set new price or discount. To save the changes, it is mandatory to select again the customer from the list.

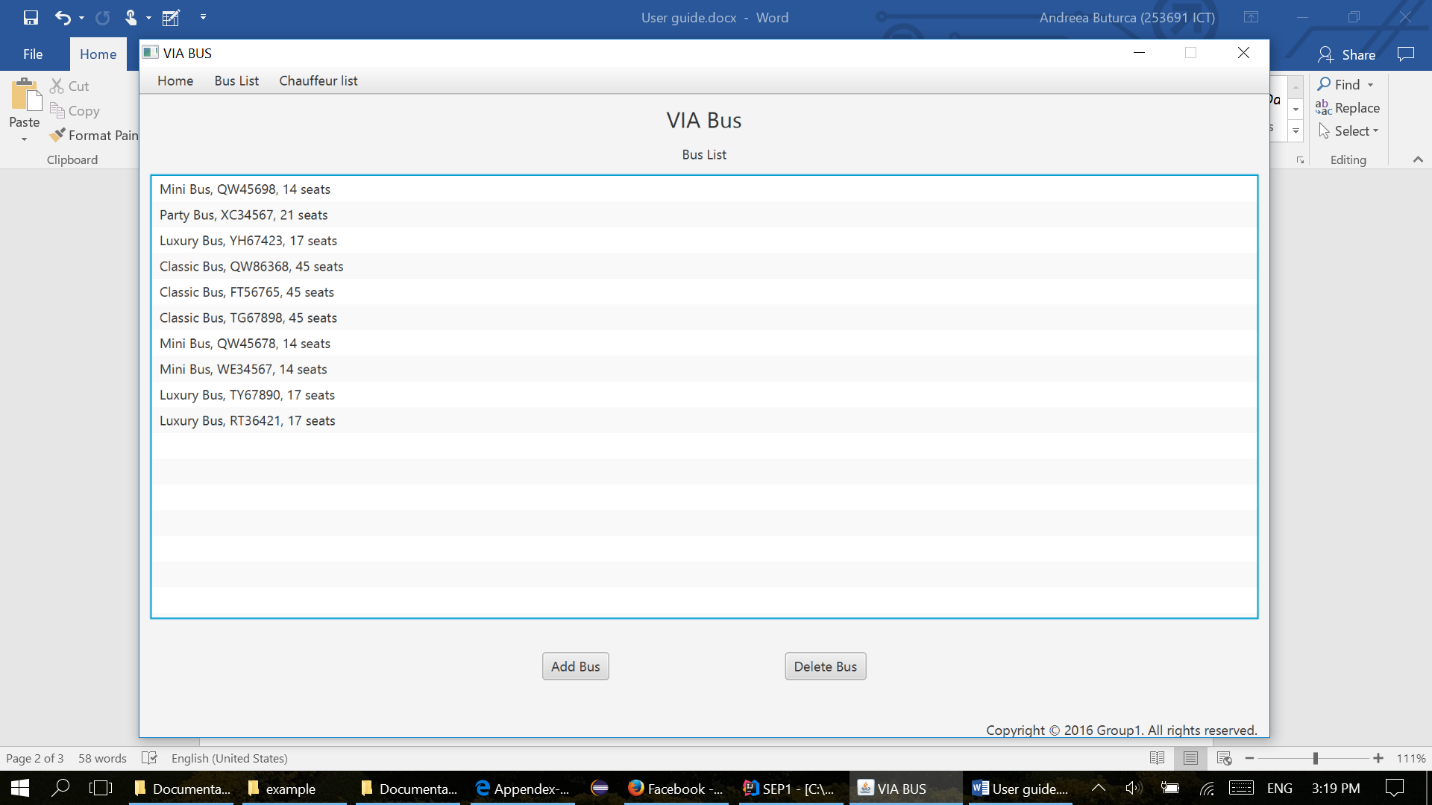


**Overview**

The main page shows in the bottom of the page the list of tours made by the VIA Bus company, sorted by date.

**Bus List**

**Bus List**

The user must go on the second tab of the menu and select Bus List in order to see all busses, displayed in a list, like the one bellow.

**Add buss**

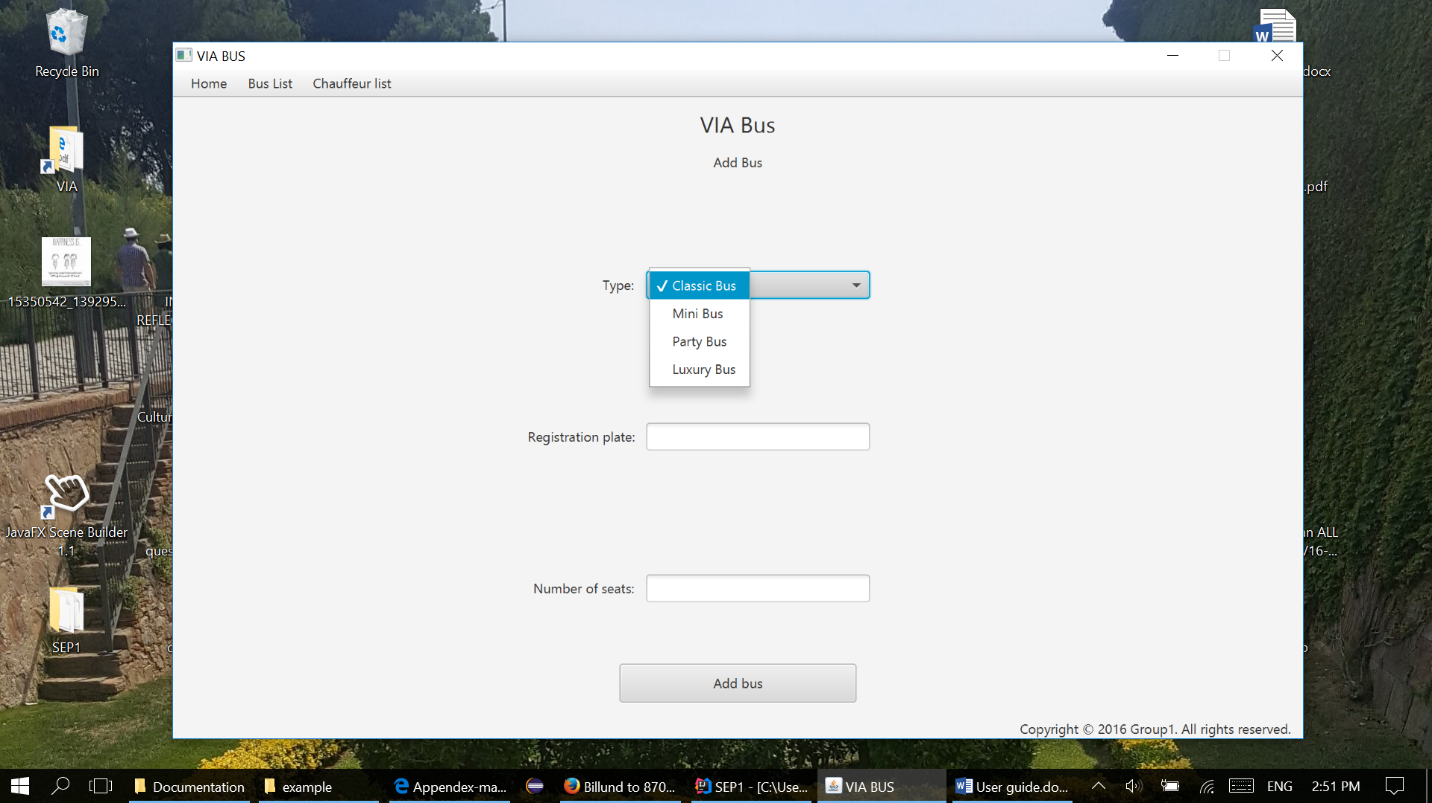
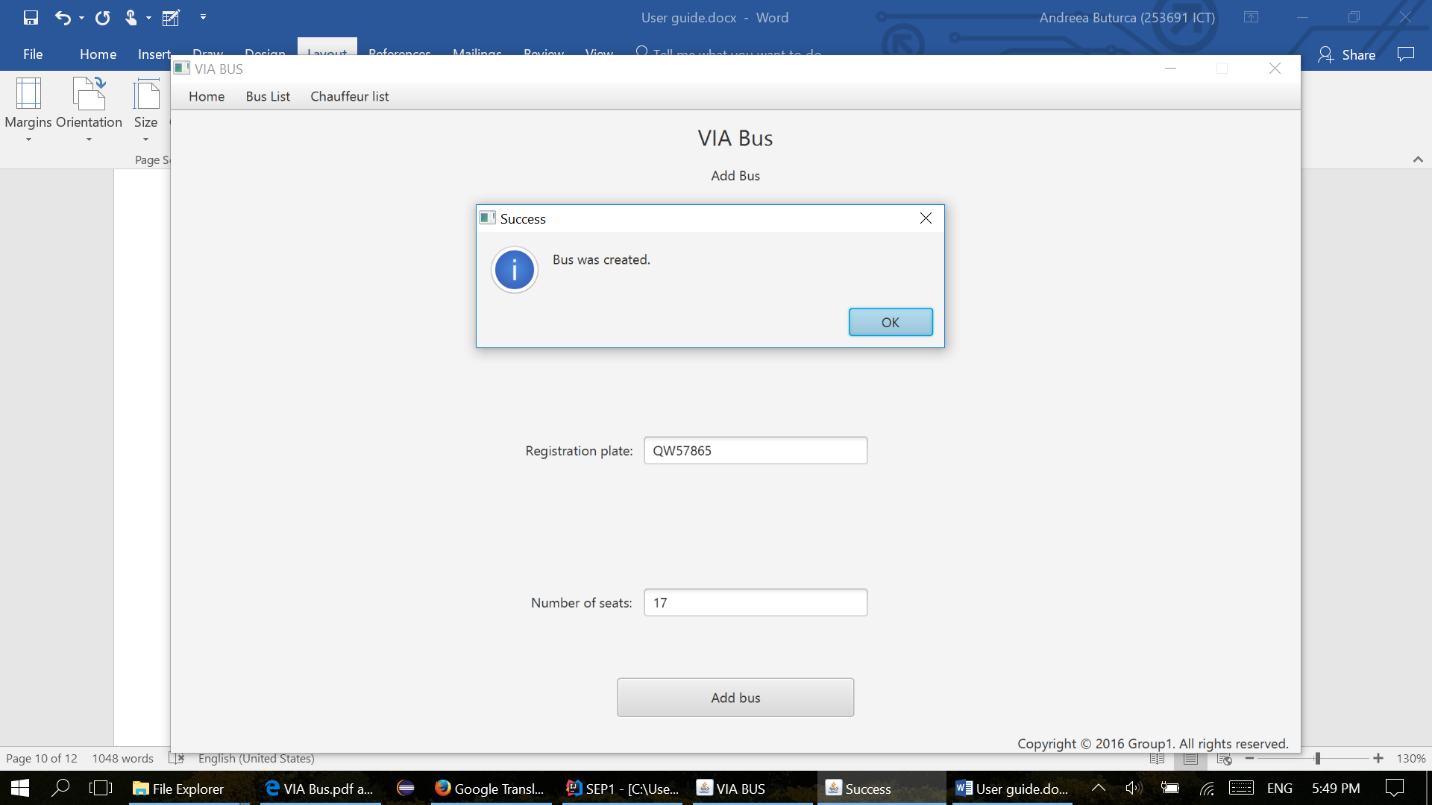
First step, the user must select one of the busses: classic bus, mini bus, party bus or luxury bus.

The second step is to add a registration plate number, which must respect to following format: 2 uppercase letters and 5 digit number (example: AS12345).

Third step: enter the number of seats, per buss.

Last step: press add button.

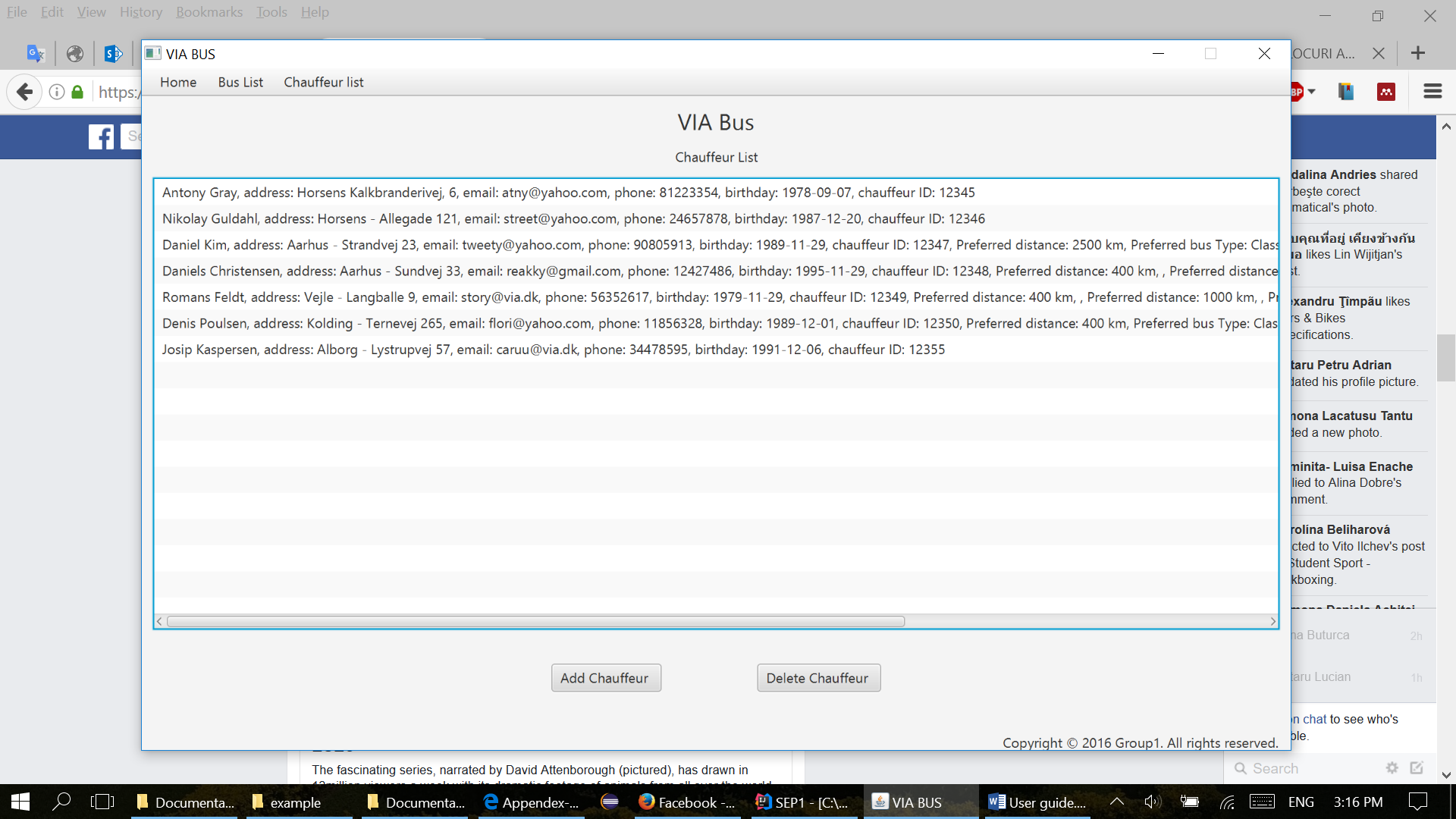
If one of the above steps it’s not respected, an alert window will appear on the screen, saying which field must be filled properly, otherwise a Succes window will be displayed, showing that the bus was created.



**Chauffeur List**

**Chauffeur List**

The user must go on the last tab of the menu and select Chauffeur List in order to see all chauffeurs displayed in a list



**Add chauffeur**

In the Add chauffeur, the user must fill in all the fields with the following mandatory data: name, address, email, phone number (8 digit), date of birthday(can be picked up or filled in with the following format month/day/year), employee ID (5 digit). The user can also select either one, or all the check boxes about chauffeur’s preferrences: distance and bus type. For long distance, the chauffeur prefers only tours with a distance bigger or equal with 2500 km, for medium between 400 and 2500 km and for short distance less, or equal with 400 km. If the user adds a vikar chuffeur, no preferrences are needed to be selected.

