ICT Engineering

Process report

VIA Bus

Group 8

PREPARED BY:

Robert Chalcak, ID 253723
Florian Kesten, ID 254114
Chavdar Hristov, ID 253912
Maja Petrusic, ID 253899

COURSE:

IT-SEP1Y-A16

SUPERVISORS:

Brigitte von Fyren Balsløv
Steffen Vissing Andersen

DATE:

16 December 2016

CONTENTS:

Before

During

After

1. GROUP POLICY

Mission

- Our study group exists because we want to learn how to work in a team how to cooperate with each other and to be successful. To complete every task we are given and to do our best in school overall. We want to develop our communication skills and to help each other to overcome their weaknesses.

Goals

- Do project, prepare for the exam, do day to day assignments, do day to day hand ins and presentations, improve our English skills - especially writing.

Commitments

We agree to:

1. Come to class.
2. Make sure that when we miss class that we contact the others in our group.
3. That we will work on group assignments collaboratively.
4. Be honest and realistic in reporting project scope and schedule.
5. Show up at meetings.
6. Keep other members of the team informed about what are you doing.
7. Complete assignments before group meetings.
8. Help others if they have problems in assignments.
9. Meet up on a regular basis.
10. Write code clearly and understandably for others.
11. Ask other members for help if I don't understand something.

Team meeting group rules

When we have meeting we will:

1. Respect each other.
2. Be open to new approaches and listen to new ideas.
3. Be clear and to the point.
4. Listen carefully other people's opinions.
5. Seek to find some common ground for agreement.
6. Communicaqte with other only in English.

7. Respect other people's work.
8. Celebrete success.
9. Seek first to understand and then to be understood.
10. Be honest and open during meetings.
11. Always be on time.
12. Be friendly and nice.
13. Divide the tasks so everyone does what he or she does best.
14. Don't create conflicts without reason.
15. Keep discussions on track.

*the group policy was signed by all members of the group

2. SWOT ANALYSIS

2.1 Individual SWOT Analysis

Florian Kesten

Strengths:

Creative, objective, critical, experience with programming and projects

Weaknesses:

Too critical, often get bored

Opportunities:

improve communication skills, learn more about programming, gain more motivation

Threats:

Being tired, bad weather

Chavdar Hristov

Strengths:

creative, practical, mature, persistent.

Weaknesses:

not that much experience with JAVA, sometimes I have lack of motivation.

Opportunities:

improve my programming skills, improve in communicating in a group project.

Threats:

bad weather of course, sometimes I may not have that much time to spend on project.

Maja Petrusic

Strengths:

Creative, dynamic, turns ideas into actions.

Weaknesses:

over-optimistic, can lose interest, language barrier, not that much experience with Java.

Opportunities:

Improve programming and communication skills.

Threats:

Illness.

Robert Chalcak

Strengths:

Open minded person, willing to learn, always on time.

Weaknesses:

Not enough experience with programming.

Opportunities:

Learn something new and interesting, improve programming and communication skills.

Threats:

Lack of available information to solve problems.

2.1 Group SWOT Analysis

Strengths:

Creativity, positive working enviromment, group work.

Weaknesses:

not much experience, easily distracted.

Opportunities:

gain experience in Java, become better in communication, improve working in a group, prove ourselves that we can finish this project before a deadline.

Threats:

not enough time for the project, we don’t leave in the same place, weather, focus, sickness.

3. CONSIDERATION AND ACADEMIC WRITING:

Target audience: mixed audience

Level of formality: informal style of writing

We have chosen the mixed audience it means that the main message of our project should be understandable for everybody, but on the other hand our project includes special terms from programming. We also decided that the user manual should be written simply and clearly so the VIA bus employees can see how the system works without any problems to figure how everything functions. Because of this, our project is written in an informal style.

4. LIST OF TASKS

Class Tour	Florian, Robert
Class MyDate	Maja, Florian
Class Reservation	Florian, Robert
Class Customer	Florian, Robert
Class PassengerList	Florian
Class Passenger	Florian, Robert
Class Chauffeur	Florian, Robert
Class Bus	Florian, Robert
Class CustomerList	Florian
Class TourList	Florian, Maja
Class ChauffeurList	Florian
Class BusList	Florian
Class CustomerListAdapter	Florian
Class TourListAdapter	Florian
Class ChaufferListAdapter	Florian

Class BusListAdapter	Florian, Robert
Class VIABUS	Florian, Chavdar
Class GUI	Chavdar, Florian

Use Case Diagram	Maja, Chavdar
Sequence Diagram	Florian, Maja
Activity Diagrams	Chavdar, Robert
UML Diagrams	Florian, Chavdar
Cover Page	Maja
Introduction	Chavdar, Maja
Design	Maja
Implementation	Chavdar
Test	Chavdar, Florian
Analysis	Robert, Chavdar, Florian, Maja
Result	Chavdar
Conclusion	Florian, Chavdar
References	Florian, Chavdar, Maja, Robert
Appendices	Florian
Group policy	Maja, Florian, Rober, Chavdar
SWOT	Maja, Florian, Rober, Chavdar
Belbin analyses	Maja
Considerations and academic writing	Maja
List of tasks	Maja
Daily log	Maja
Reflections	Maja, Florian, Robert, Chavdar
Blooms profile	Maja, Florian, Robert, Chavdar
User Manual	Florian, Chavdar, Maja

5. DAILY LOG

1. Meeting

Date: 7 November 2016, Monday

Present: Chavdar, Robert and Florian

- We made a group contract
- We discussed group roles and personal responsibilities

2. Meeting

Date: 10 November 2016, Thursday

Present: Florian, Chavdar, Robert

- We made list of requirements
- We talked about Use Case diagram
- General discussion about project

3. Meeting

Date: 21 November 2016, Monday

Present: Florian, Chavdar, Robert and Maja

- We met as a group of four members for the first time and we have updated the group contract because Maja joined our group after we made initial contract.
- New contract was signed by all of the members.
- We drew up the use-cases for all requirements
- We made use case description for the use case diagram
- We agreed to make a facebook group

4. Meeting

Date: 1 December 2016, Thursday

Present: Florian, Chavdar, Robert and Maja

- We drew up Activity diagrams for the requirements
- General discussion about project
- We decided that we should have some feedback from our SDJ supervisor

5. Meeting

Date: 6 December 2016, Thursday

Present: Florian, Chavdar, Robert and Maja

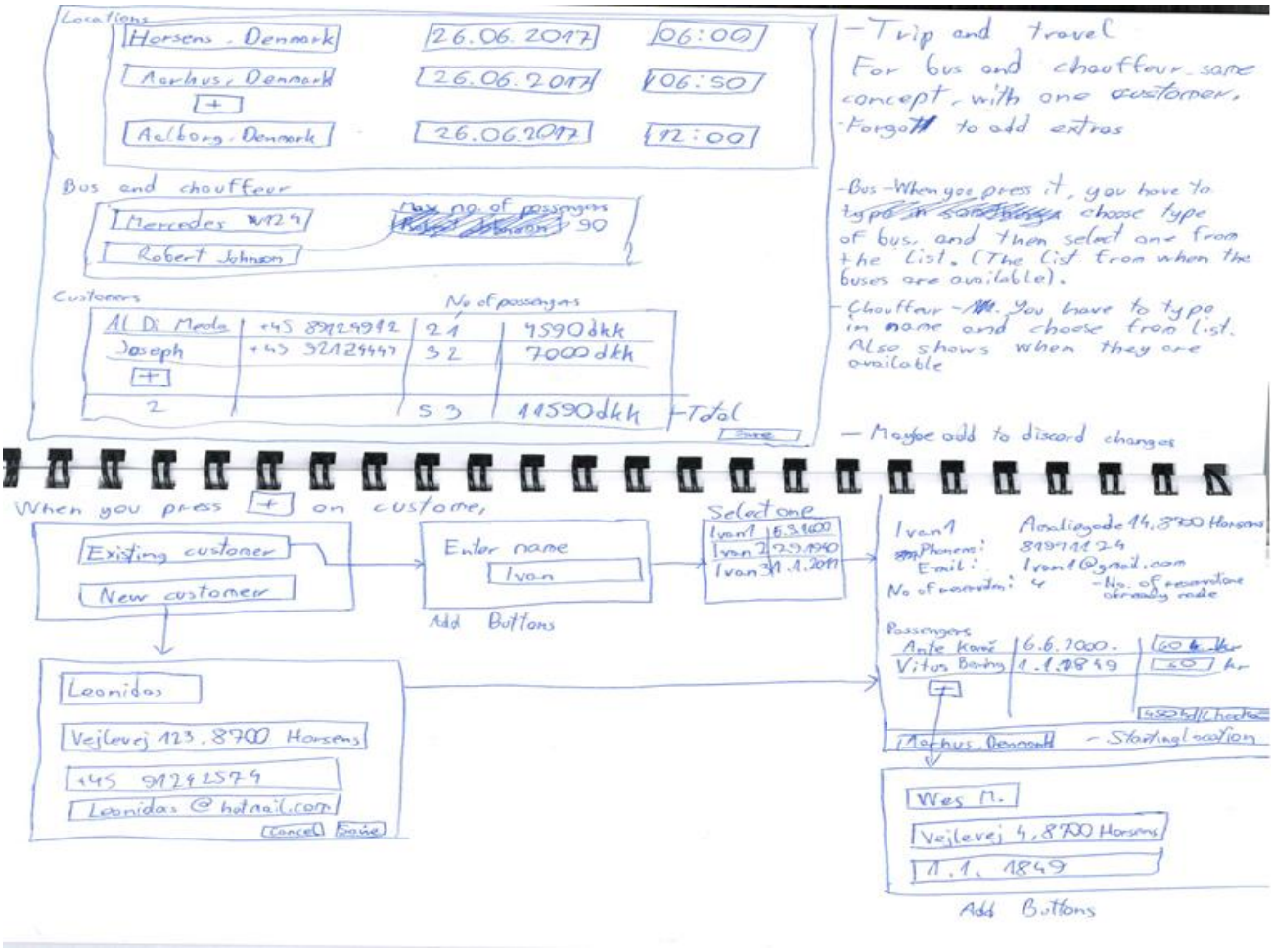
- After getting feedback we decided to update requirements, diagrams and use case description
- We drew the basic class diagram
- We talked about how will we split up the work

6. Meeting

Date: 9 December 2016, Thursday

Present: Florian, Chavdar, Robert and Maja

- We code the basic diagrams
- We made a sketch of GUI
- We began work on GUI



Initial sketches of the GUI

7. Meeting

Date: 11 December 2016, Sunday

Present: Florian, Chavdar, Robert and Maja

- Working on GUI
- Complex programming
- Starting thinking about the report layout
- We wrote an introduction
- Decided to change some things about the way the system will work
- Start programing MyDate class
- Work on sequence diagram
- Testing code
- Implementing classes from the problem domain into the GUI

8. Meeting

Date: 14 December 2016, Wednesday

Present: Florian, Chavdar, Robert and Maja

- Complex programming
- Finish work on the GUI
- Implementing classes from the problem domain into the GUI
- Each group member worked on individual SWOT analysis
- Personal process reports
- Working on project report

9. Meeting

Date: 15 December 2016, Thursday

Present: Florian, Chavdar, Robert and Maja

- Implementing classes from the problem domain into the GUI
- Finishing program
- Working on project and process report
- Finishing up the final things for the project and grouping it together

10. Meeting

Date: 16 December 2016, Friday

Present: Florian, Chavdar, Robert and Maja

- Checking the spelling and grammar
- Finished project and reports

6. REFLECTIONS

Chavdar Hristov

In the beginning I thought the project won't be that hard. We were confident in ourselves but that was our problem.

We faced the fact we had I lot to do on our project. Because we didn't start earlier, now it was difficult to initiate the project . The time we had was limited. After meetings with our supervisors we knew what we had to do. Soon we managed to divide the work in our group to different pieces. Everyone was given the task in which we were feeling comfortable and confident. During the project period we had more meetings with our supervisors and our group. We had disagreements but that is normal for group work. Eventually we found the right way of doing the project.

I've gained knowledge and experience from the project. I feel happy now that we've succeeded. We've worked hard and managed to successfully finishing the project on time.

Robert Chalcak

First time I read the project assessment, I was a little bit scared that it will be too difficult. I have read this assessment many times, and I was thinking about this for a long time. I tried to divide this project to small pieces and got the idea that it is not so difficult as it looks. After the first official seminar at school, I have started programming. By reading the assessment and diagrams I started programming with universal and basic classes, like: MyDate, Person, Chauffeur, Passengers, Customer and etc. I tried to share my outcome with other members in our group. Also during lessons I asked the teacher or other members for their opinion. Also I used a lot of materials from the book and from the internet and of course skills which I got during lessons.

The next part was little bit more difficult. It was about making classes with will be store the passengers, customers and chauffeurs. Sometimes it was little bit hard, because I got a lot of error messages during testing. Of course I tried to solve the problem so I looked for advices in the study material on Study net and fixed it. I was successful. I always shared my outcome with other members of my group. During programming I also did documentary links for further documentation. After each meeting with supervisor and with other members I tried to reshape my code, according to the requirement and class diagram. Sometimes I wrote more codes than necessary, because it is always better to have more than less, and it is the possibility to make further correction or reduction.

During the processing of the project I began to understand how the programming language works better and I also understood algoritme better. This project has given me a lot of experience in developing the software. I tried to cooperate with other people and create something that will work and I am happy that we were successful. Now I know how the structure of basic software looks and how each part is connect to other and how each part is very important for proper functionality and, when there is something wrong how we can fix it. This assessment has also given me the idea that more heads know more than only one, and also if more people work together and each member works on his task and we connect all the tasks together, it can rise single user system, which will be work.

Maja Petrusic

Before starting project I was very worried because I had problems with the first group. But tenkfully I joined this group. I felt tat doing project in this group won't be that hard. I was quite confident in myself and my team.

Soon we realized that we have lot more work than we thought and that we have not begun to do the project on time. I had the feeling that we will not finish the project on time. But after meeting with supervisor we divide the work in our group to different pieces. After that it was easier to move forward. The team atmosphere was quite pleasant, everyone was open-minded and I think that helped a lot when we stumbled along some problems.

Now we know each other much better than at the beginning, also we have gained a lot of experience with programming and documentation of projects. All in all, it was a really nice experience

Florian Kesten

Before the project period started, I wasn't really excited about the project, because I was feeling a bit unmotivated at that time, but I was sure we would finish it easily and well before the deadline.

When the project period started, I realised it would be harder than I thought, but I still thought we would have more than enough time, and that we could start doing the project a bit later. As the deadline was getting closer and closer, we started doing the project, and were sure that we would enough time. We put our ideas on the table, but we weren't sure how to divide the roles between us. We split parts of the project, and everyone started doing what they were assigned to. Our meetings were laid back, and the atmosphere was relaxed, but the roles weren't that clear, so we couldn't implement every requirement.

All in all, it was a pleasant experience, I learnt a lot about projects, documentation, reports, and gained insight into what being an ICT engineer would look like. But most importantly, I got more motivated about my studies and I'm excited about my future projects both in my studies and outside of them.

7. BLOOMS PROFILE

Chavdar Hristov

Before:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5															
Good	4															
	3															
	2															
Basic	1															
No knowledge	0									X						

After:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5			X		X		X								
Good	4		X				X									
	3	X													X	
	2															X
Basic	1															
No knowledge	0									X						

Robert Chalcak

Before and after:

x- before - after Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6													X		X
	5		X			X	X								X	
Good	4	X									X	X	X			
	3							X								
	2			x							X	X	X			
Basic	1								x	X						
No knowledge	0				X											

Maja Petrusic

Before:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5														■	
Good	4							■				■				
	3		■	■		■	■		■		■			■		■
	2	■											■			
Basic	1									■						
No knowledge	0				■											

After:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5		■										■		■	
Good	4					■	■	■			■	■				
	3	■		■					■					■		■
	2															
Basic	1									■						
No knowledge	0				■											

Florian Kesten

Before:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5															
Good	4															
	3															
	2															
Basic	1															
No knowledge	0															

After:

Fill in this form – include it in your portfolio – discuss it with the rest of the group Date.....	Bloom's level	Keeping a portfolio	Reflecting on learning	System development	SCRUM	Java Programming	Object-oriented design and programming	UML	Web Programming	Database design	Written English	Spoken English	Team working	Sharing knowledge	Project planning	Presentation / exam skills
Excellent	6															
	5															
Good	4															
	3															
	2															
Basic	1															
No knowledge	0															

ICT Engineering

Project report

VIA Bus

Group 8

PREPARED BY:

Robert Chalcak, ID 253723
Florian Kesten, ID 254114
Chavdar Hristov, ID 253912
Maja Petrusic, ID 253899

COURSE:

IT-SEP1Y-A16

SUPERVISORS:

Brigitte von Fyren Balsløv
Steffen Vissing Andersen

DATE:

16 December 2016

Contents

Introduction

The objective of our program is to create a functional system for VIA bus. VIA bus is a company for bus tours, located in Horsens. The company needs a system which should keep track of the tours, chauffeurs, buses and customers.The requirments for the system will be based on an interview with manager Trip Driver.

After interview with Mr. Driver the major problems were identified. Since the company did not have any system before it was a problem to keep track of their chauffeurs, so the company needs a computer program to have a better overview of when each of the chauffeurs is available. And also to register their tours, customers, passengers and reservations. These problems have been expertly tackled while developing our program.

Also, a user friendly graphic interface would be developed for the program, so the user could easily navigate the various tabs using only a mouse and a keyboard. It is important to mention that program should be use only by employees and manager.

Project Analyses 2

2.1 Functional System Requirements:

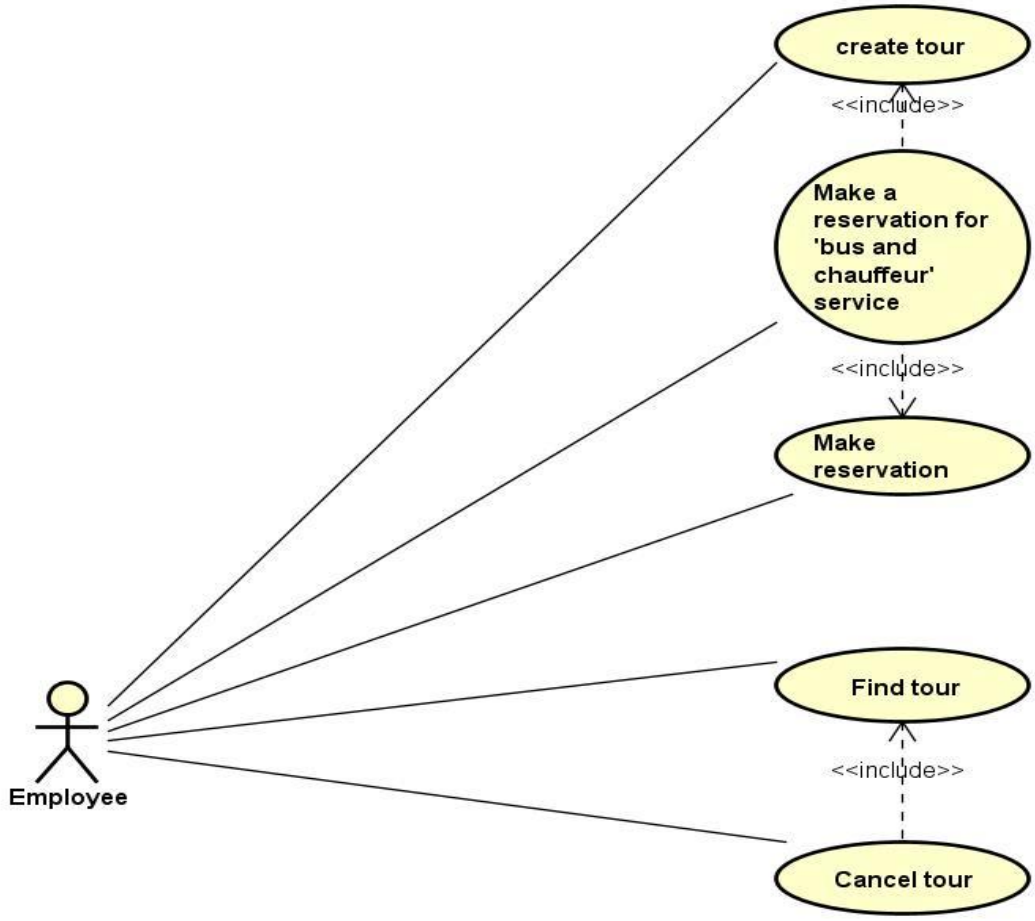
1. The employee should be able to create a new tour.
2. The employee should be able to set the the locations, dates and times for the locations, bus and chauffeur for the tour.
3. The user should be able to choose the type of the tour.
4. The user should be able to choose the type of bus for the tour.
5. The system should be able to save the tour.
6. The employee should be able to create one or more reservations for a tour.
7. The employee should be able to set the customer, passengers, and price for each reservation.
8. The employee should be able to write down the name, address, phone number, and email address for a customer.
9. The employee should be able to write down the name, address and age for a passenger.
10. The employee should be able to create a 'Bus & Chauffeur' reservation.
11. The employee should be able to set the locations, dates and times for the locations, bus, chauffeur, customer, number of passenger, and price for the 'Bus & chauffeur ' reservation.
12. The system should be able to store the name, address, 5-digit employee ID for a chauffeur, when he is on a trip, his phone number and email address.
13. The employee should be able to cancel a reservation.
14. The system should be able to display when each chauffeur is available.
15. The system should be able to store information about regular customers.
16. The system should be able to store chauffeur wishes.
17. The user should be able to search for a tour by destination.
18. The employee should be able to enter a company or organization name in case they are the customers .

19. The employee should be able to add extras to a reservation.

2.2 Non-functional System Requirements

1. System must be developed in JAVA programming language.
2. System must be able to be controlled by mouse and keyboard.

2.3 Use case modeling

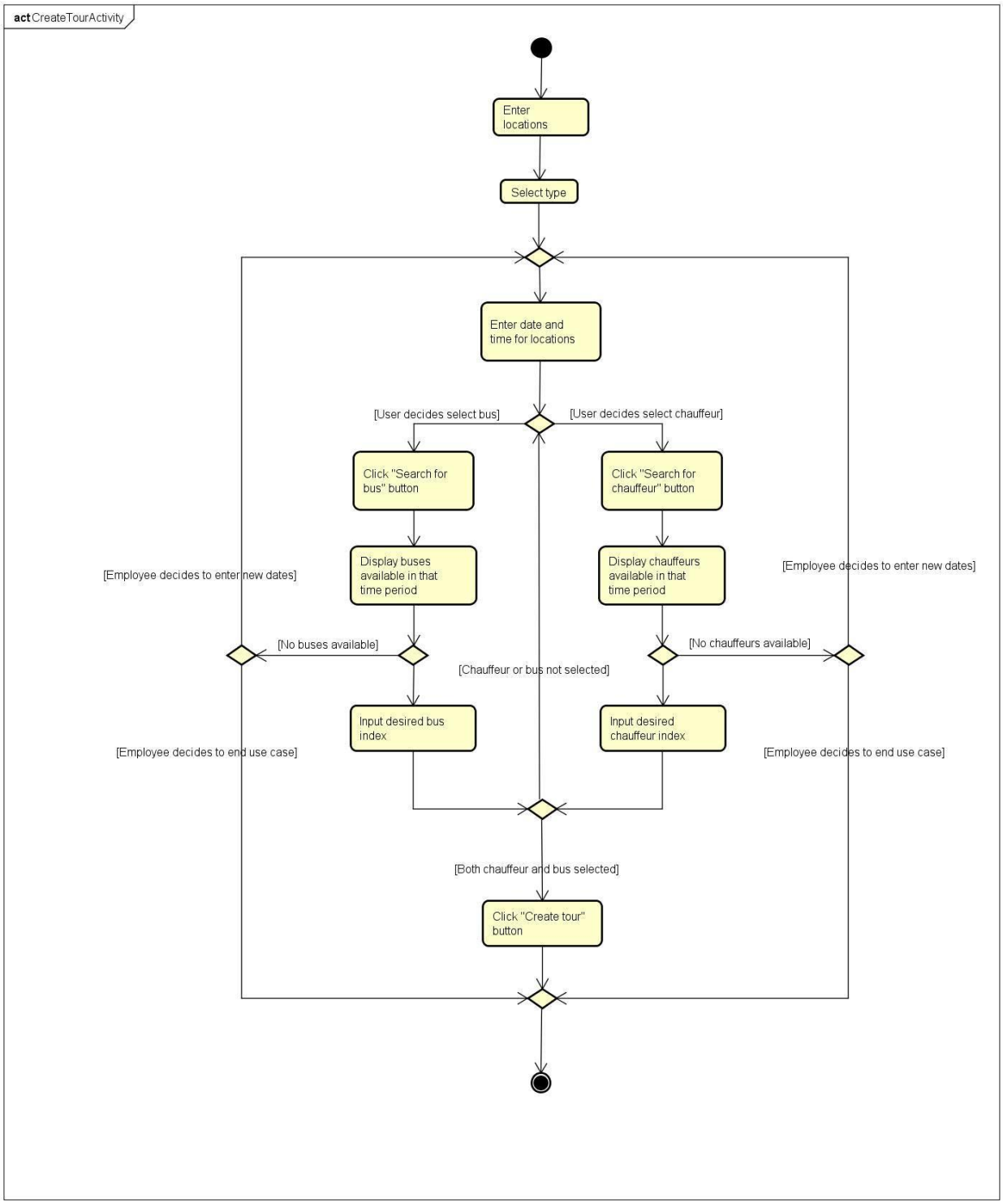


This figure shows the potential situations that the user might face during his work. The diagram also gives idea what the system needs to do.

The following figure has a table with the use case description of "Create tour". This will be the case that the user will come across during the work.

ITEM	VALUE
UseCase	create tour
Summary	An employee creates a tour.
Actor	Employee
Precondition	None
Postcondition	A tour has been made, with the data for the ch auffeur, bus, locations, and dates stored in the system
Base Sequence	1. Choose type of tour. 2. Enter locations. 3. Enter date and time for each location. 4. Search for available buses of a type. 5. System returns a list of available buses of th at type (Use Case: List available buses). 6. If no buses are available, go to step 3, or go to exception 1. 7. Select a bus from the list. 8. Search for available chauffeurs. 9. System returns a list of available chauffeurs (Use Case: List available chauffeurs). 10. If no chauffeurs available, go to step 3. 11. Select a chauffeur from the list. 12. Show reservation details and save data in s ystem
Branch Sequen ce	
Exception Sequ ence	No available buses of that type: 1-4 as base sequence Returns empty list Use case ends No available chauffeurs: 1-8 as base sequence. Returns empty list Use case ends
Sub UseCase	Manage tour data Manage passengerdata
Note	

The following diagram illustrates the activity diagram for the "Create tour" use case.

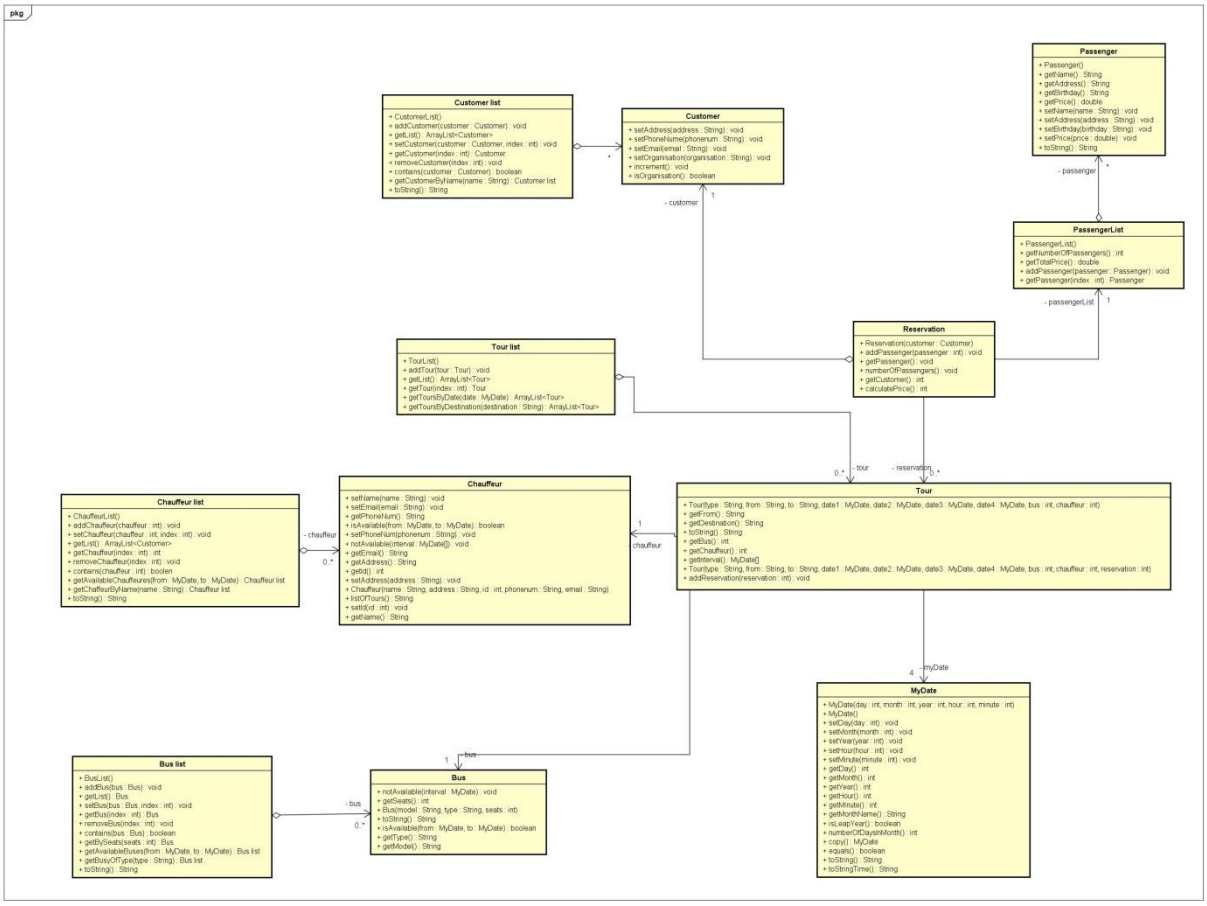


This figure shows the “Create tour” situation. You can see the steps that the user must take so the system reacts properly. First the user must enter locations for the tour, after he/she selects type of the tour. Next he should type in the dates and times for the tour. The user clicks a button and the system searches for available buses or chauffeurs for the time he wants to create a tour. If there are available drivers and buses, the user must select the desired index of chauffeur or bus. If there are no available buses or chauffeurus, the user cancels the tour or

returns back to the dates and times and arranges different dates and times so when buses and chauffeurs are available. After the bus and chauffeur is selected the tour can be made.

For additional diagrams and use case descriptions, check the appendix.

The next illustration shows the classes from the system and a few important methods.



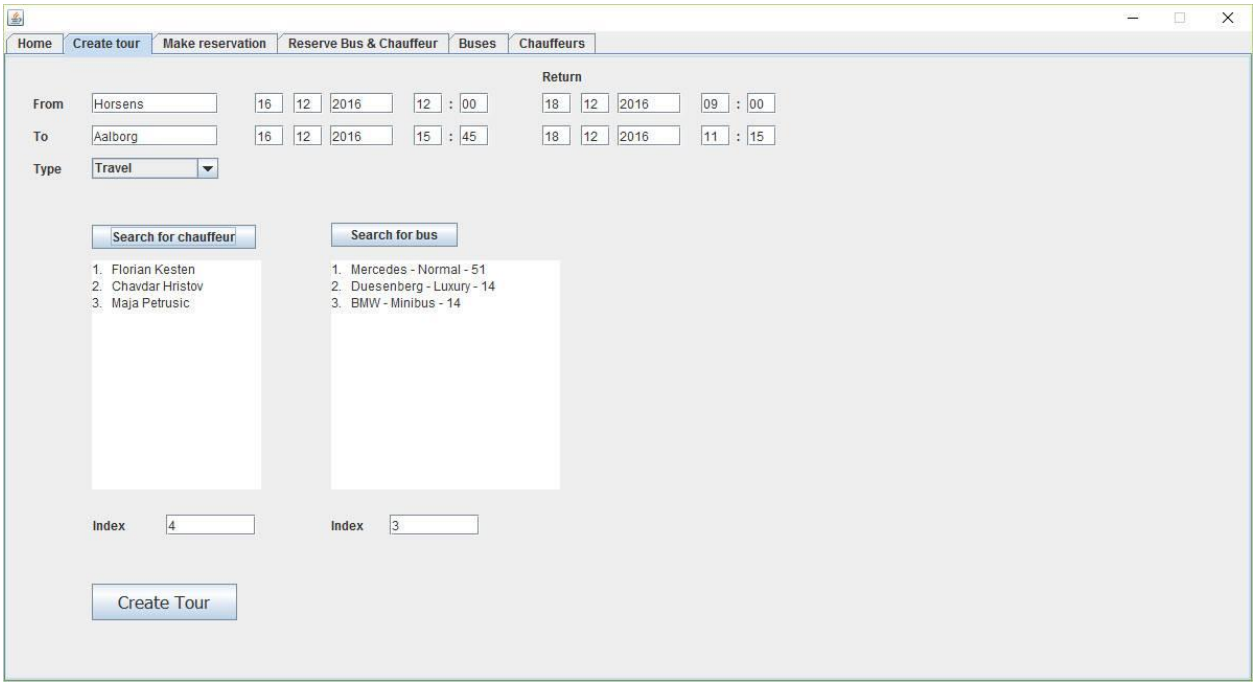
This figure illustrates some of the important methods used in the system. The “Reservation” class creates an object of the Customer which contains all the customer’s information. The “Bus” class contains information about the buses, the type of bus, the number of seats, the model. “Chauffeur” class holds the information about the chauffeur(Name, Address, Phone number, etc.). The “Bus” and “Chauffeur” classes are conected with an assoiciation to the “Tour” class.

Design 3

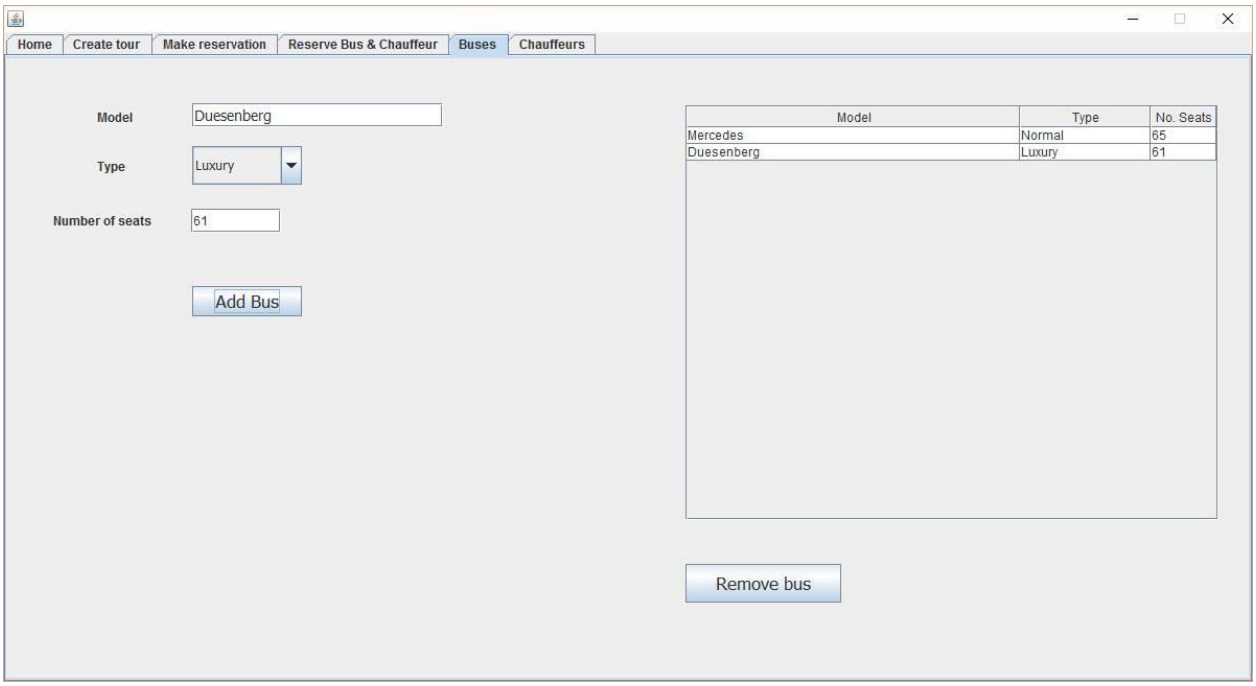
3.1 GUI design

This part of the report portraits the graphical user interface of the system for creating a tour.

When starting the program there are six tabs you can select.

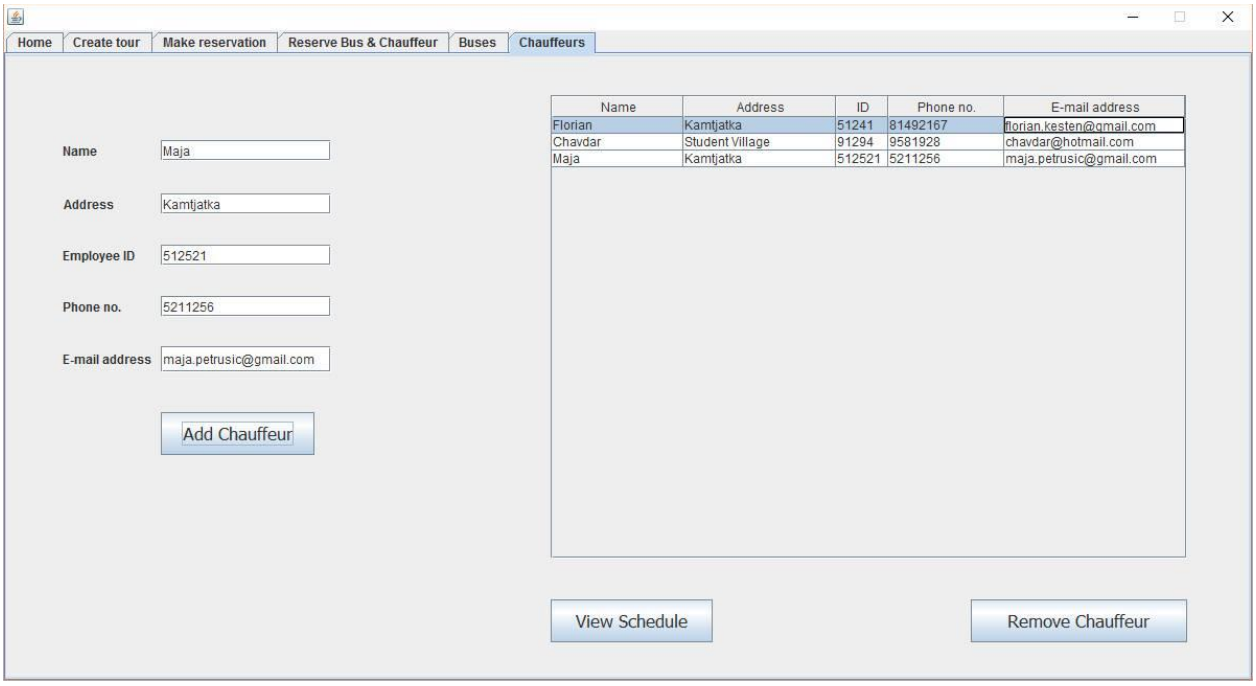


The user enters the desired data information in the text areas at top of the window. The text lists provides information about available chauffeurs and buses for the period that has been typed in before that. When all the data is entered the button “Create Tour” saves the information and makes a new tour.



From the “Buses” tab the user is capable of adding a new bus. The user fills out the text fields and selects the information from the dropdown list to complete the requirements. Once the bus is added, for the user is possible to use it or remove it.

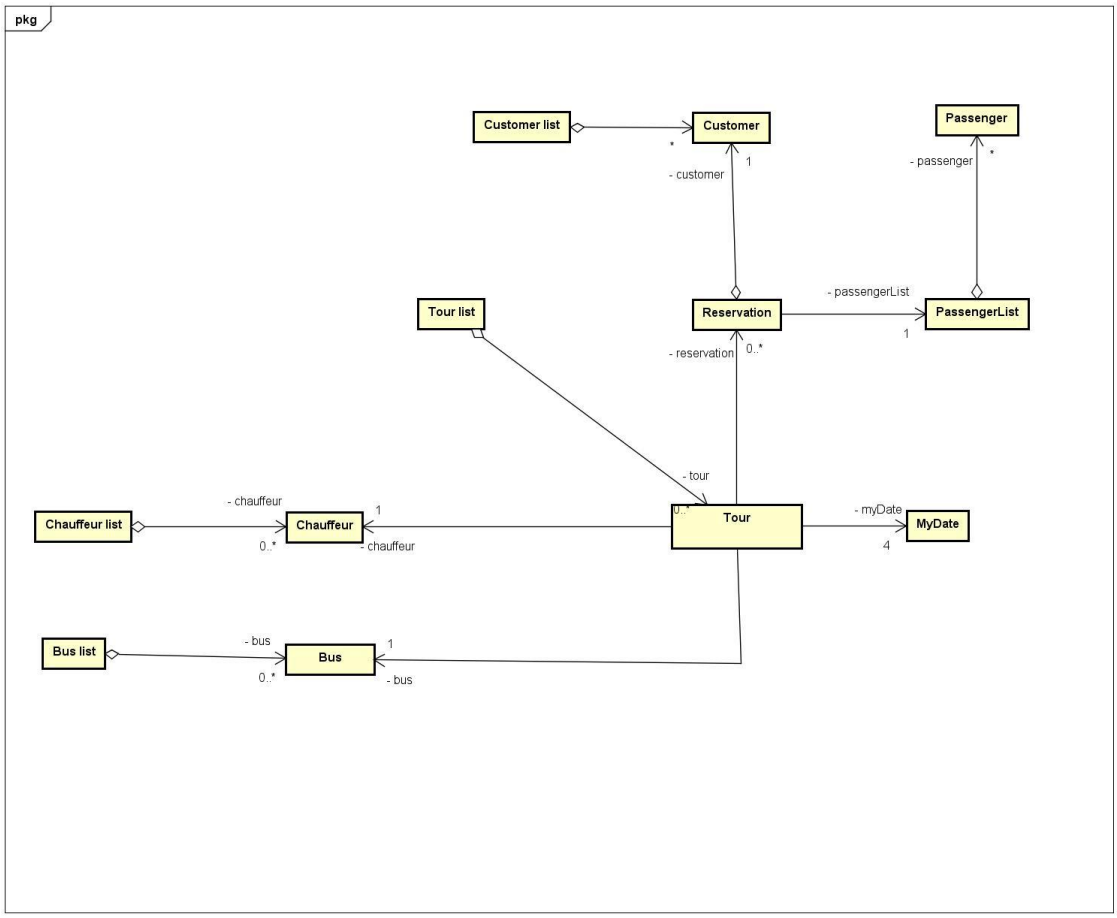
Once the user has buses, he can navigate to the “Chauffeurs” tab as it is seen on the next picture.



From here the user can add chauffeurs employees. First he needs to fill out the text areas with required information about the driver. Once the chauffeur has been added, he is part of the schedules for the tours. The schedule is called by pressing the “View Schedule” button. Also the user is capable of removing the chauffeurs if they are no longer part of the company.

3.2 Class Diagrams

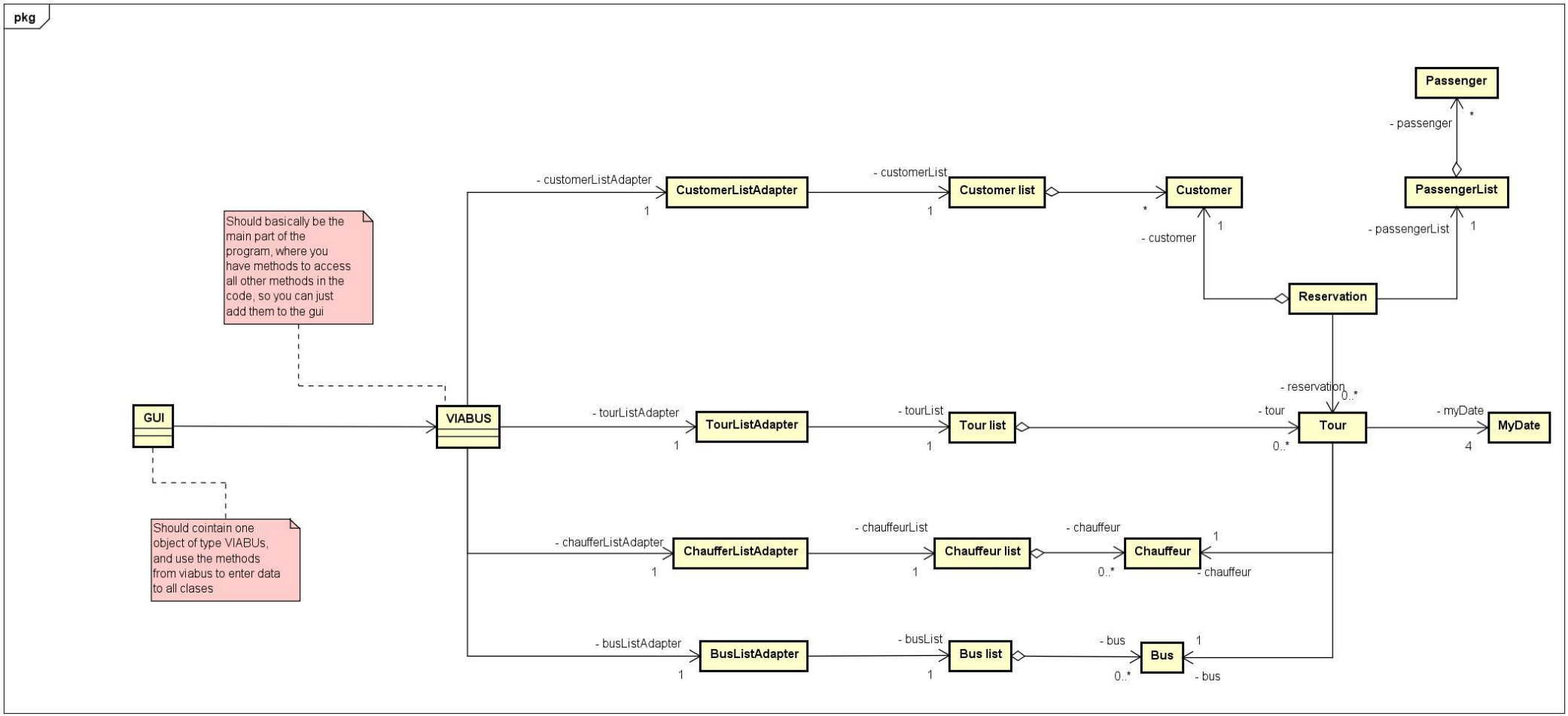
The figure … shows the analysis class diagram as basis with some important methods and attributes to better understand the design.



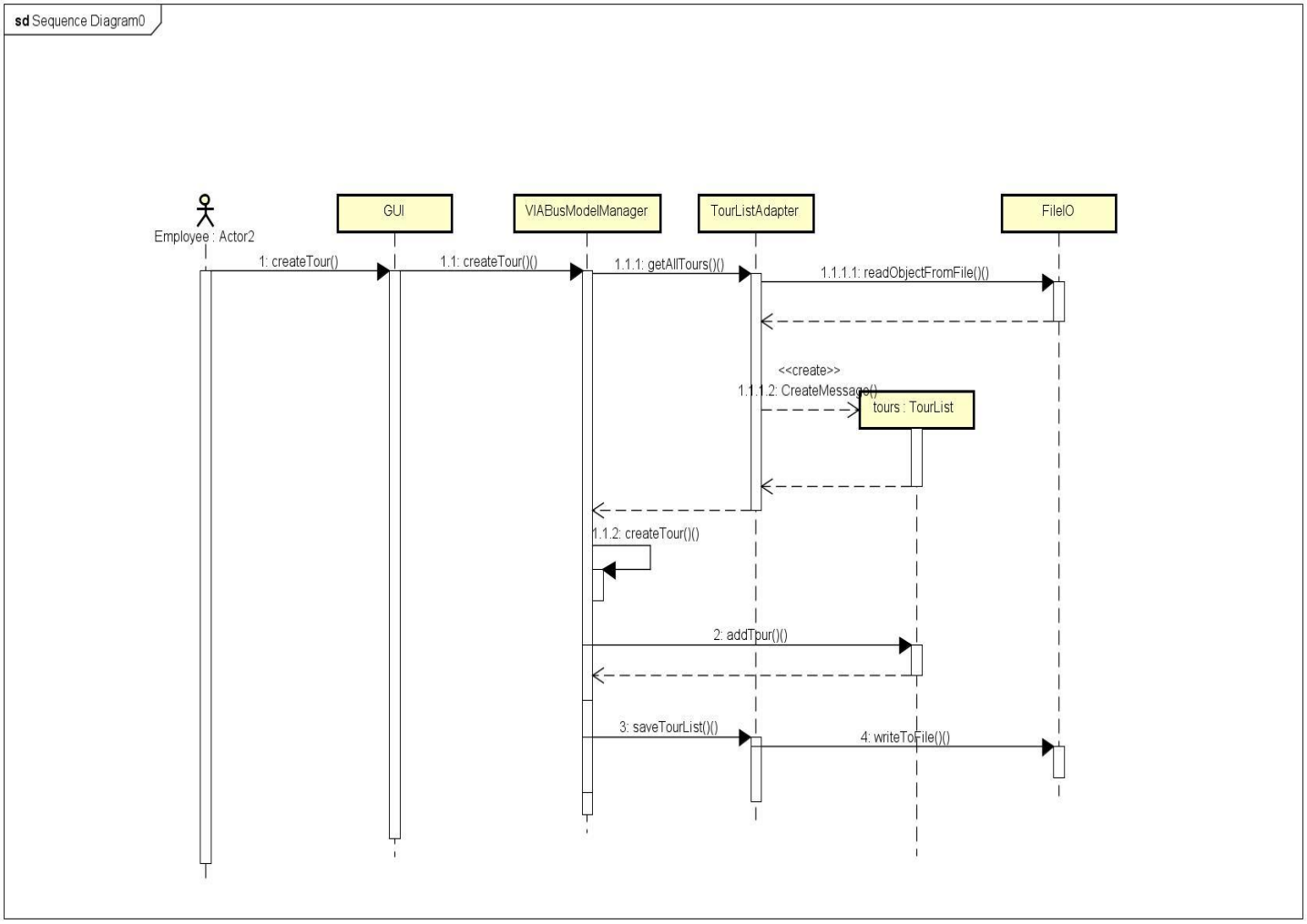
The diagram shown presents the object approach used in developing the program. Also the relationships between the classes are indicated

The “Tour” class has objects from the “Bus”, “Chauffeur”, “MyDate” and “Reservation” classes. The “Tour” is connected with the “TourList” class which contains tour objects. “BusList” contain an array list of “Bus” type objects, “ChauffeurList” has the array list of “Chaffeur” type objects. The “CustomerList” and “PassengerList” also contain array lists therefor with object of type Passenger and Customer. The “Reservation” class has the access to those objects and through it the “Tour” class has the access to all the date obects.

The diagram next illustrates all the classes along with the file adapters and VIA Bus Model. It has all important methods for the system classes. Detailed class diagram with all the connection can be seen on the next page.


pkg
Passenger
- passenger
*
PassengerList
- passengerList
1
Customer
- customer
1
*
- customerListAdapter
CustomerListAdapter
1
- customerList
Customer list
1
Should basically be the main part of the program, where you have methods to access all other methods in the code, so you can just add them to the gui
Reservation
- reservation
0..*
GUI
VIABUS
- tourListAdapter
TourListAdapter
1
- tourList
Tour list
1
- tour
0..*
Tour
- myDate
MyDate
4
Should cointain one object of type VIABUs, and use the methods from viabus to enter data to all clases
- chaufferListAdapter
ChaufferListAdapter
1
- chauffeurList
Chauffeur list
1
- chauffeur
0..*
Chauffeur
1
- chauffeur
- busListAdapter
BusListAdapter
1
- busList
Bus list
1
- bus
0..*
Bus
1
- bus

3.3 Sequence diagram



User creates tour, GUI calls method createTour for VIABUSModelManager, the manager gets all tours from the TourlistAdapter. The adapter creates a TourList and returns it to the manager. The manager creates Tour object, adds it to TourList, returns the list to the adapter, which then writes the list to FileIO

Implementation 4

Every object we wanted to store in a file, needed to have it's class implement Serialisable.

Serialisable allows an object of a class to be stored in a binary file, and also allows it to be read form the same file, and converterd back to an object of a class.

```
import java.io.Serializable;

public class TourList implements Serializable
```

The following code shows the implementation of the method isAvailable(from,to).

When we call this method an object of type bus or chauffeur, with two mydate parameters, it checks if the dates are in between any of the dates stored in an arraylist of type mydate[] which hold the objects dates when they are on tour, and returns true if they are not on a tour in the entered time period.

```
public boolean isAvailable(MyDate from, MyDate to)
  {
    for (int i = 0; i < dates.size(); i++)
    {
      if ((from.isBetween(dates.get(i)[0], dates.get(i)[1]))
          || (to.isBetween(dates.get(i)[0], dates.get(i)[1]))
          || (from.equals(dates.get(i)[0]))
          || (from.equals(dates.get(i)[1]))
          || (to.equals(dates.get(i)[0])) || (to.equals(dates.get(i)[1]))
          || (dates.get(i)[0].isBetween(from, to))
          || (dates.get(i)[1].isBetween(from, to)))
      {
        return false;
      }
    }
    return true;
```

REFERENCES:

Books

- Tony Gaddis, Starting Out with Java: Early Objects, Fifth Edition, 2015.

Websites

- Studienet: https://studienet.via.dk/Class/IT-SEP1Y-A16/Session%20Material/VIA%20Bus.pdf
- Java2s: http://www.java2s.com/Tutorial/Java/CatalogJava.htm
- Oracle: http://docs.oracle.com/en/
- Stackoverflow: http://stackoverflow.com/