



BINUS UNIVERSITY BINUS INTERNATIONAL

Assignment Cover Letter

(Teamwork)

Student Information:

	Surname	Given Names	Student ID Number
1.	Alifio	Rasyid	2201798295
2.	Muchsin	Hisyam	2201797430

Course Code : COMP6571 **Course Name** : Data Structures & Algorithms

Class : L2AC **Name of Lecturer(s)** : Kartiko Eko Putranto

Major : CS

Title of Assignment : Bus
(if any) **Reservation**

Type of Assignment : Final Project

Submission Pattern

Due Date : 29 - 06 - 2019 **Submission Date** : 29 - 06 - 2019

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

1. Alifio Rasyid 2. Muchsin Hisyam

CONTENTS

1. Problem Discussion	3
2. Proposed Alternative Solution	3
3. Theoretical Analysis	3
3.1. Implementation	3
3.2. How it works	3
4. Program Manual	4
4.1. Main page	4
4.2. Add reservation.....	4
4.3. Cancel/delete reservation	6
4.4. Edit reservation.....	7
4.5. View all reservations	9
4.6. Save file.....	10
4.7. Read file	10
5. References	11

1. Problem Discussion

Nowadays the development of technology is getting faster, one of which is in the field of computers. Now, computers play an important role in facilitating the completion of a job, increasing work efficiency and increasing the creativity and activities of employees so they have good skills or abilities. So we decided to make a simple bus reservation program that not trouble some the user / admin when using it. This program already created reservation data automatically for the next 7 days, and 5 bus for each days. So, the user / admin doesn't need to worry about manipulating the reservation and bus data.

2. Proposed Alternative Solution

So, we have tried to make this bus reservation application using an array. But array is not efficient for this application, so we decided to use Linked List, Vector, and Map on this application.

3. Theoretical Analysis

3.1. Implementation

- **ADT Linked List**, is a linear data structure in which the elements are pointing to the next elements and not stored at contiguous memory locations (dynamic array).
- **<vector>**, is a dynamic arrays that can resize itself automatically when the data or elements is inserted or deleted.
- **<map>**, is a containers that store elements in which each elements has a key and value, and can't have same key.
- **<string>**, is a header that introduces string and have function for string types.
- **<ctime>**, is a header that can get and manipulate date and time data or information.
- **<ctype>**, is a header that contains function to validate characters.
- **<fstream>**, is a header that can manipulate data to file such as read data from files, and save data to files.

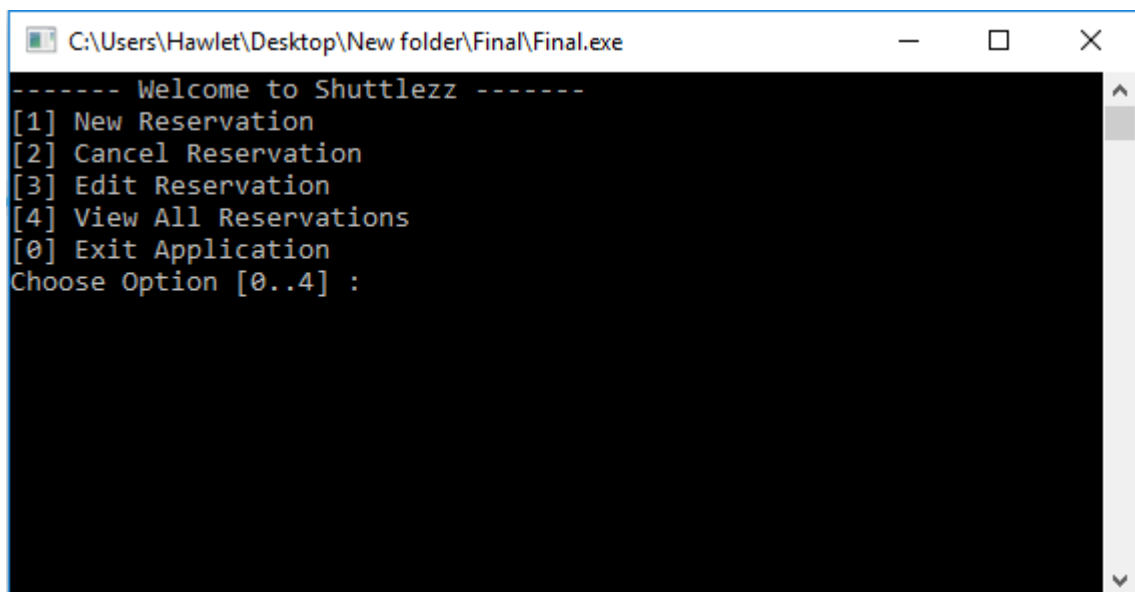
3.2. How it works

In this application, we use Linked List data structure for saving all reservation data. Because Linked List are good for inserting, deleting and editing data in the front or back (constant time). We need to use `<map>` to save the date data, so there will be 5 buses per day and have 20 seats each buses. So, the key for the map is the date, and the value is the `vector<int>`. We import `<string>` only for using the `to_string()` function for convert the numerical value or integer value to string value, for example the integer day, month, and year to string. And we also use `<ctype>` to validate the input data, such as `seat_no` can't be alphabetic, and use `<fstream>` to save the `LinkedList` nodes to ".txt" file when the program close, and load the data on ".txt" when the program started.

4. Program Manual

4.1. Main page

This is the main page of the program, so basically this main page is running the `run();` function on `Business` class, this function print all of the menu that can be runned by the program. The user (admin) can select the menu based on numbers that the `run();` function printed, and the input menu that the user(admin) selected/inserted will run the next function based by the selection.



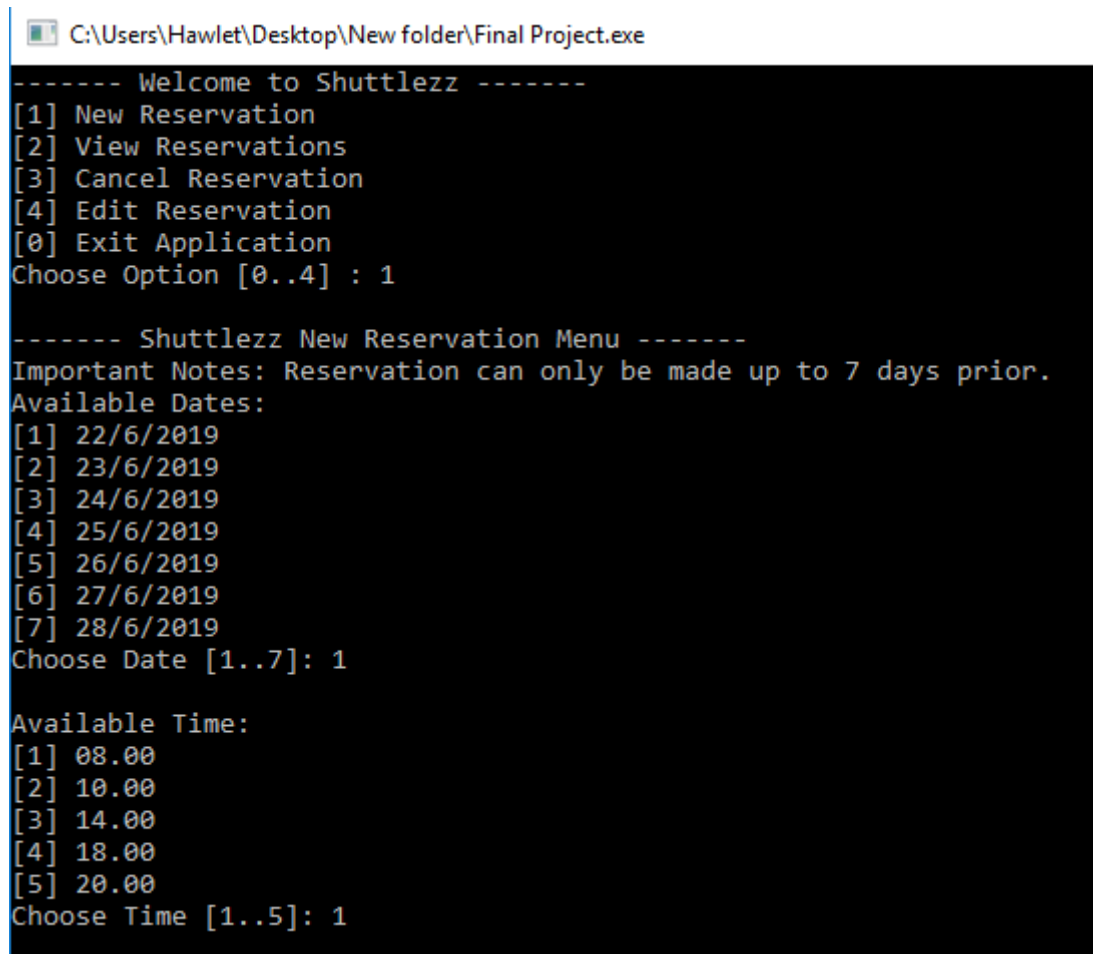
```
C:\Users\Hawlet\Desktop\New folder\Final\Final.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] Cancel Reservation
[3] Edit Reservation
[4] View All Reservations
[0] Exit Application
Choose Option [0..4] :
```

Image 4.1. Main page

4.2. Add reservation

The user (admin) can add the details reservation data of the customer's reservation and save it the `LinkedList<Reservation> AllReservation` if they select the "[1] New Reservation" menu at the main page. So, on this menu the program ask the user (admin) to input the customer's reservation data such as "first_name" variable for saving the customer's first name that will be inputed by the user (admin) , "last_name" variable for saving the customer's last name that will be inputed by the user (admin), "date" variable that will be selected by the user (admin) from the 7 days forward from current date when the `addReservation()` function runs, "time" variable that will be selected by the user (admin) later after

the program ask the date, "seat_no" variable for choosing the bus's seats that will be inputed by the user (admin).



```
C:\Users\Hawlet\Desktop\New folder\Final Project.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] View Reservations
[3] Cancel Reservation
[4] Edit Reservation
[0] Exit Application
Choose Option [0..4] : 1

----- Shuttlezz New Reservation Menu -----
Important Notes: Reservation can only be made up to 7 days prior.
Available Dates:
[1] 22/6/2019
[2] 23/6/2019
[3] 24/6/2019
[4] 25/6/2019
[5] 26/6/2019
[6] 27/6/2019
[7] 28/6/2019
Choose Date [1..7]: 1

Available Time:
[1] 08.00
[2] 10.00
[3] 14.00
[4] 18.00
[5] 20.00
Choose Time [1..5]: 1
```

Image 4.2.1. Add reservation

```

No of Available Seats: 20
Seats Available: ([!!] means seat are occupied)
[01][-----]
      [02][03][04]
[05]      [06][07]
[08]      [09][10]
[11]      [12][13]
[14]      [15][16]
[17][18][19][20]
Choose Seat [1..20] :1

Please Fill in the Customer's Information
First Name: Alifio
Last Name: Rasyid
Seat 1 reserved for Alifio Rasyid
Press Enter to continue...

```

Image 4.2.2. Add reservation

4.3. Cancel/delete reservation

The user (admin) can cancel or delete the reservation data of the customer's reservation on the `LinkedList<Reservation> AllReservation` if they select the "[3] Cancel Reservation" menu at the main page. So on this menu the program show the current date until 7 days forward, and the user (admin) can cancel or delete the customer's reservation data by those selected date. Then, the program will show all reservations based on that date, and the user (admin) will inputted the certain reservation data, and the program will delete that data.

```
C:\Users\Hawlet\Desktop\New folder\Final Project.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] View Reservations
[3] Cancel Reservation
[4] Edit Reservation
[0] Exit Application
Choose Option [0..4] : 3

----- Shuttlezz Cancel Reservation Menu -----
Available Dates:
[1] 22/6/2019
[2] 23/6/2019
[3] 24/6/2019
[4] 25/6/2019
[5] 26/6/2019
[6] 27/6/2019
[7] 28/6/2019
Choose Date [1..7]: 1

[1] Name: Alifio Rasyid    Date: 22/6/2019    Time: 08.00    Seat No: 1
Choose Option: 1

Press Enter to continue...
```

Image 4.3. Cancel/delete reservation

4.4. Edit reservation

The user (admin) can edit the reservation data of the customer's reservation on the `LinkedList<Reservation> AllReservation` if they select the "[4] Edit Reservation" menu at the main page. So, on this page is similar like the previous cancel or delete menu, so the program will print the current date until 7 days forward then it will print the customer's reservation data based by date, then the user (admin) select which data that they want to edit, this edit reservation only can edit the "first_name", "last_name", and "seat_no" variables.

```
C:\Users\Hawlet\Desktop\New folder\Final Project.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] View Reservations
[3] Cancel Reservation
[4] Edit Reservation
[0] Exit Application
Choose Option [0..4] : 4

----- Shuttlezz Edit Reservation Menu -----
Important Notes: Only name and seat numbers are editable.
Available Dates:
[1] 22/6/2019
[2] 23/6/2019
[3] 24/6/2019
[4] 25/6/2019
[5] 26/6/2019
[6] 27/6/2019
[7] 28/6/2019
Choose Date [1..7]: 1

[1] Name: Alifio Rasyid      Date: 22/6/2019      Time: 08.00      Seat No: 1
Choose Option: 1
```

Image 4.4.1. Edit reservation

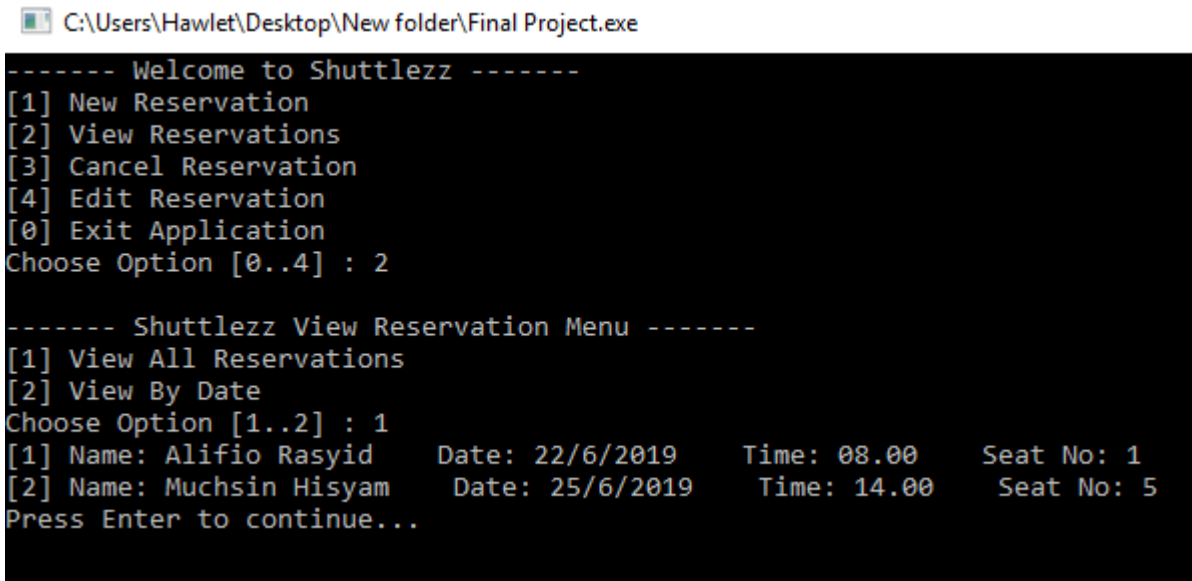
```
0Seats Available: ([!!] means seat are occupied)
[01][-----]
      [02][03][04]
[05]      [06][07]
[08]      [09][10]
[11]      [12][13]
[14]      [15][16]
[17][18][19][20]
Choose Seat [1..20] :20

Please Fill in the Customer's Information
First Name: Muchsin
Last Name: Hisyam
Seat 20 reserved for Muchsin Hisyam
Press Enter to continue...
```

Image 4.4.2. Edit reservation

4.5. View all reservation

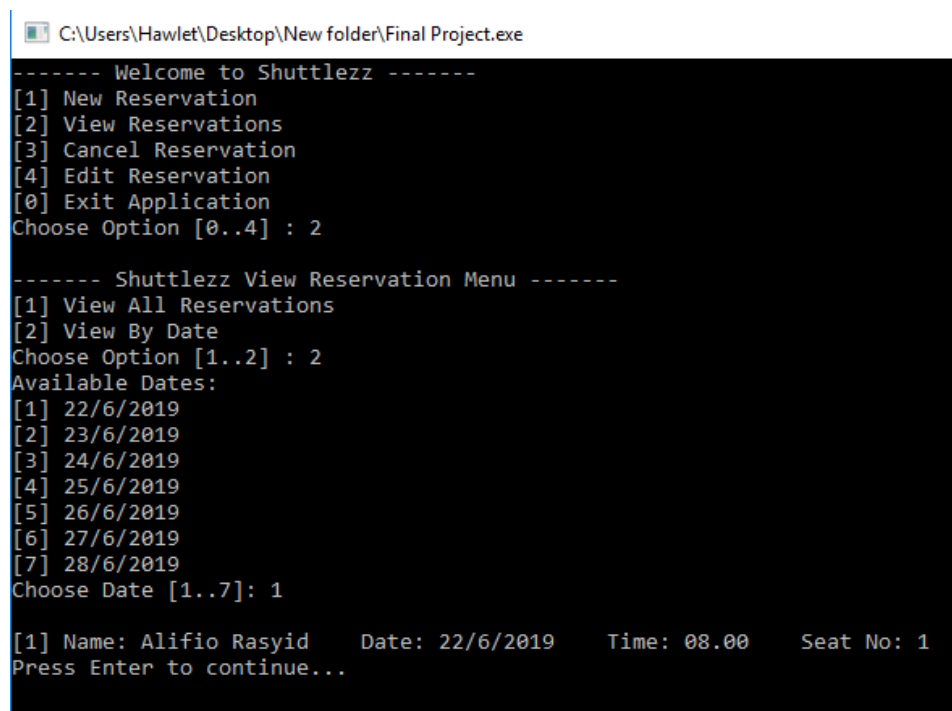
The user (admin) can print all the reservations data of the customer's reservations on the `LinkedList<Reservation> AllReservation` if they select the "[2] View All Reservations" menu at the main page. So, on this menu the user (admin) can select 2 option, print all the reservations data or print reservations data by date.



```
C:\Users\Hawlet\Desktop\New folder\Final Project.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] View Reservations
[3] Cancel Reservation
[4] Edit Reservation
[0] Exit Application
Choose Option [0..4] : 2

----- Shuttlezz View Reservation Menu -----
[1] View All Reservations
[2] View By Date
Choose Option [1..2] : 1
[1] Name: Alifio Rasyid      Date: 22/6/2019      Time: 08.00      Seat No: 1
[2] Name: Muchsin Hisyam    Date: 25/6/2019      Time: 14.00      Seat No: 5
Press Enter to continue...
```

Image 4.5.1. View all reservation



```
C:\Users\Hawlet\Desktop\New folder\Final Project.exe
----- Welcome to Shuttlezz -----
[1] New Reservation
[2] View Reservations
[3] Cancel Reservation
[4] Edit Reservation
[0] Exit Application
Choose Option [0..4] : 2

----- Shuttlezz View Reservation Menu -----
[1] View All Reservations
[2] View By Date
Choose Option [1..2] : 2
Available Dates:
[1] 22/6/2019
[2] 23/6/2019
[3] 24/6/2019
[4] 25/6/2019
[5] 26/6/2019
[6] 27/6/2019
[7] 28/6/2019
Choose Date [1..7]: 1

[1] Name: Alifio Rasyid      Date: 22/6/2019      Time: 08.00      Seat No: 1
Press Enter to continue...
```

Image 4.5.2. View all reservation by date

4.6. Write file

On this step, when the user (admin) select the exit menu, the program will save the `LinkedList<Reservation> AllReservation` data to a ".txt" file through the `WriteFile()` ; function, then the program will close.

```
// Read and Write
void Business::WriteFile(){
    ofstream outFile("data.txt");

    if (outFile.fail()){
        cerr << "Error Opening File" << endl;
        exit(1);
    }

    if (outFile.is_open()){
        for (LinkedList<Reservation>::Iterator it = AllReservations.begin(); it != AllReservations.end(); it++){
            string first_name = (*it).GetFirstName();
            string last_name = (*it).GetLastName();
            string date = (*it).GetDate();
            string time = (*it).GetTime();
            int seatno = (*it).GetSeatNo();
            outFile << first_name << " " << last_name << " " << date << " " << time << " " << seatno << endl;
        }
    } else {
        cout << "Cannot open file" << endl;
    }
}
```

Image 4.6. writeFile() function code

4.7. Read file

Same like the save file but this is the opposite, when the user (admin) open the program, the program will take all data from ".txt" file and set those data to the `LinkedList<Reservation> AllReservation` through the `readFile()` ; function, then the program will run the main page.

```
void Business::ReadFile(){
    int dateindex, timeindex;
    string first_name, last_name, date, time;
    int seatno;

    ifstream inFile("data.txt");
    if (inFile.fail()){
        cerr << "Error Opening File" << endl;
        exit(1);
    }

    if (inFile.is_open()){
        while(inFile >> first_name >> last_name >> date >> time >> seatno) {
            if (getDateIndex(date) != -1){
                dateindex = getDateIndex(date);
                timeindex = getTimeIndex(time);
                bookSeat(dateindex, timeindex, seatno-1);
                Reservation tempData(first_name, last_name, date, time, seatno);
                AllReservations.addNode(tempData);
            }
        }
    } else {
        cout << "Cannot open file" << endl;
    }
}
```

Image 4.7. readFile() function code

5. References

- <https://www.geeksforgeeks.org/data-structures/linked-list/>
- <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>
- <https://www.geeksforgeeks.org/vector-in-cpp-stl/>
- <http://www.cplusplus.com/doc/tutorial/files/>
- <http://www.cplusplus.com/reference/ctime/>
- <http://www.cplusplus.com/reference/cctype/>
- <http://www.cplusplus.com/reference/string/>
- http://www.cplusplus.com/reference/string/to_string/
- <https://stackoverflow.com/questions/19039972/linked-list-vs-vector>

Our Github Project's Link : <https://github.com/muchsinhisyam/Bus-Reservation>