Online Bus Ticketing System

Conference Paper · August 2016

DOI: 10.13140/RG.2.2.30979.71208

1 author:

Syed Hasan
Asia Pacific University of Technology and Innovation
20 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Table of Contents

ACKNOWLEDGEMENT

First of all I would like to thank my lecturer Ms. SALASIAH BINTI SULAIMAN who taught me the basic of system development. She helped us to learn about different types of software development methodologies along with benefits. She also taught us the basic knowledge of UML Design.

Lastly, I am very glad that I have successfully able to complete our assignment on time. I appreciate all the helpers for helping out along the way of this development. I thank all of them cordially for their helpful attitude.

1.0 INTRODUCTION

In today's tech era, IOT (Internet of things) has become much popular around the world. Almost all the devices, which are known as smart device, can connect to the internet and access data from any corner of the world. There was a time when people used to waste their valuable times just to get a piece of information. Now the technology is more advance then compare to any previous times. One of the blessings of technology is web application. It allows users to interact with the system from anywhere as long as they are connected to the internet.

EZB SDN BHD is a bus service company located in the heart of Malaysia, Kuala Lumpur. Usually they do paper works for keep their customer's details and booking reservation records. If any customer need to reserve seat he or she need to call them or walk in to their counter which is consider as wasting their valuable times. Sometimes the phone line also keep busy and customers unable to reserve seats for them. Besides, EZB need to keep records of the payments made by customers in papers and quite impossible for them to keep track on payment issues. Currently EZB is planning to replace their old booking system with new system which is online based. So they want to implement an online web based bus ticketing system which will be easier for customers to book from home and abroad as well as for them to manage their overall business smoothly.

Here the system I am going to discuss is "Bus Ticketing System" which is completely a web application. As we already discussed above that internet has made the user's interaction through the system easier, so this web application can connect to respective servers for accessing data which will surely help users to purchase the bus ticket or reserve their seats online without waiting on queue. Moreover, in recent decays peoples are like to travel to get some relief from their monotonous life. So, they want to travel without any hesitations. In this modern tech era they want a system that will enhance the portability, accessibility as well as user friendly. So here, we are going to implement a web system, which we already stated above, having all the features that will make it more user friendly and accessible.

Below I will describe the assumptions, features and how the system is going to be designed using UML (Uniform Modelling Language). Along with that I will also provide some strong and weakness (limitations) of the system.

2.0 FEATURES

According to our system we are having some features which will make users to access the system smoothly. Here we will describe what are going to implement in this bus ticketing system. We will integrate the features with the user's interaction to make it clearer.

2.1 Users List

a. User: Customer

Here customers can purchase the ticket or reserve their seats by accessing the system as a guest or as a member. With every purchasing or reservation the system will generate a reservation id or purchased id for both guest and members at a time. Once the payment is done customers won't be able to send the request through the system to refund the payment if they want to cancel their purchase. Customers can update their account details as for example: contact number, email address, mail address etc. If the customers is not registered members then the system will not have any account for the guest to update at any time they want. So the information as for example name, address, contact, and email they will provide while reservation need to make sure they don't change for any reasons. Registered Customers are able to cancel their booking although there will be no refund.

b. User: Staff

Employee is another user of this system. They will access the system for reservation and booking purposes. Every employee can login to the system by entering their individual username and password. Depending on customers preferences they will purchase the ticket on behalf of them or reserve the seats and send them an email as a confirmation or notification.

c. User: Manager

Manager will be able to access the system to check the reports of booking or purchasing details of individual customers. He can cancel the booking on request. He will also check the employees working hours completed. Manager will also be able to generate the sales report depending on months or years. He can also update the bus schedule.

d. User: Admin

Admin is the key users of this entire web system. He will register new staff, new manager and even new admin (if necessary) by entering their details into the system. He will also be able to manage the booking of the registered customers. Again, changing the user's credentials like username and password is another important function that only an admin will be allowed to do that. In a nutshell admin will be able to access each and every functionalities of the system. Besides admin can also delete users account.

3.0 ASSUMPTION

Here we are going to illustrate some of the assumption regarding the systems functionalities. We will discuss some of the advantages, disadvantages of the system itself. We will also show how we will design or modelling the system using UML (Uniform Modelling Language) for examples: Use Case, Activity (Swim Lane) and Class Diagram (Draft). For further description in details we will proceed step by step so that it will help us to understand the overall development process.

Now we will list out some advantages and disadvantages regarding the system functionalities.

3.1 Advantages

- The system having online payment functions. So that it will be easier for customers to pay online rather than going to the counter and pay by cash.
- The system will generate a voucher code for registered customers every 15 days. So, using this voucher customers will have some discount while purchasing ticket online.
- Customers will be able to see the available bus depending on the schedule they select along with the price.
- To make the seat reservation clearer the system having the picture of available and reserved bus seats so that customers will know which seats they can reserve before they go for purchasing tickets. Below is a sample of a bus reservation seats GUI ((SHOHOZ.COM, 2014):

Figure 1 BUS SEATS RESERVATION UI

Above is a sample of bus sets reservation UI which shows the available and booked or purchased seats by color.

- If there is any upcoming or on going promotion then the system will show it for customers whoever accessing.
- As the system having online transaction it will not save the customer's credit or debit details like card numbers, expiry dates and cvv code for security issues while purchasing or reserving the seats.
- If the registered customers want to add their credit or debit card as a default payment methods then the system will have the sanction where it will encrypt the card details and allow them to use it while purchasing without entering any card details.
- If the users forget their password the system will have the panel where they can apply for the new password for logging in to the system thus the new password will be send to their registered email address promptly.
- In case of emergency, the system will ask users to add their secondary email address for password recovery. It is also an extra advantages the system will provide as a better security for users.
- For enhancing the security while online transaction the system will be secured by SSL (Secured Socket Layer) having certification from Veri Sign & MacAfee.
- For the last not for least, every data provide by the users will be encrypted by 128 bits.

Although the system having some advantages, it will also have some disadvantages too. Below are some disadvantages:

3.2 Disadvantages

- The system will not be available for 24 * 7 (Every Sunday). For that customers need to go to the counter and purchase tickets by cash.
- The system will not allow customers to refund.
- The system is accessible only for limited hours (6:00 am – 12:00 am)

- If any of the registered customers do not access the system for more than 6 months then the system will delete their account from the database to keep their database free from excessive files.
- System will not support any offline data accessibility.
- System will not support PayPal transfer.
- Admin will not be able to access reservation sanction of the system.
- Once booking confirmed customers will not be able to cancel.
- All the data need to backup manually at the end of day.

4.0 SOFTWARE DEVELOPMENT MODEL

A software development model or system development methodology in software engineering is a framework that is used to structure, plan as well as control the process of developing an information system. There are few software development models which we usually follows. They are:

- Agile Software Development
- Extreme Programming
- JAD (Joint Application Development) Waterfall
- WSDM (Web Semantic Design Method)
- Design Improvement
- Small Releases

Changes in software requirements is a common in every software development. As a developer we need to make sure that whatever the changes take place our system should meet the overall requirements of the stakeholders. Here I am going to choose XP (Extreme Programming) methodology to develop this system.

Extreme Programming

Extreme Programming is one of the agile methodology for system development. Here changes in requirements while iteration is possible compare to others like (Scrum, RAD etc.). It improves the quality and responsiveness of system application.

There are 5 phases in this agile methodology. They are:

- Planning
- Designing
- Coding
- Testing
- Listening

We are going to discuss how we relate those phases in this development in details.

Planning

The first step we are going to take is planning. Here we the development team will involve customers or users to create the system requirements or user stories. Here the users of our system are Admin, Staff, Manager and the registered customers. We will convert their requirements into the iteration which is usually a small parts of the system functionalities. We will focus on dead line of the iteration so that we will be able to complete the requirements on time.

Design

After the planning we will start designing the overall system. Here we will include object oriented design models (UML) to illustrate the overall functionalities of the system. This design models will show the active users of the system and their involvement in every process. As for examples: registered customers login, customer registration, payment, reservation, booking updates and so on.

Coding

This is the most important phases we will take into account while developing the system. Here we will do pair coding for developing the system using one machine so that at the end we will be able to produce high quality code and at the same time reduce the overall cost. It will also help us to

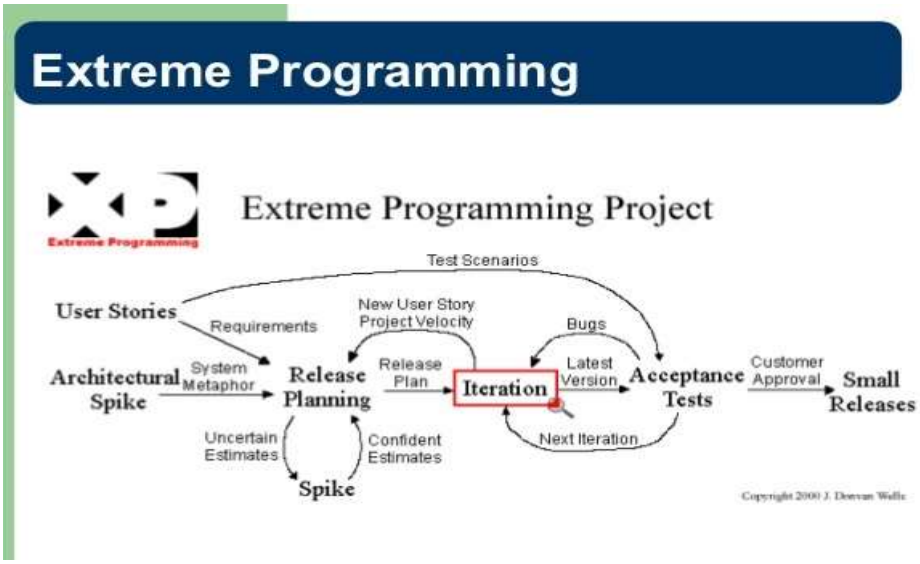


Figure 2 Extreme Programming

Reduce the conflict and optimize the functionalities of the system.

Testing

Here we will do testing while developing rather than waiting for ending the overall development phases. We will test the codes which will lead us to eliminate the system bugs and make sure that the code passes all the unit test before we release it to our stakeholders (EZB).

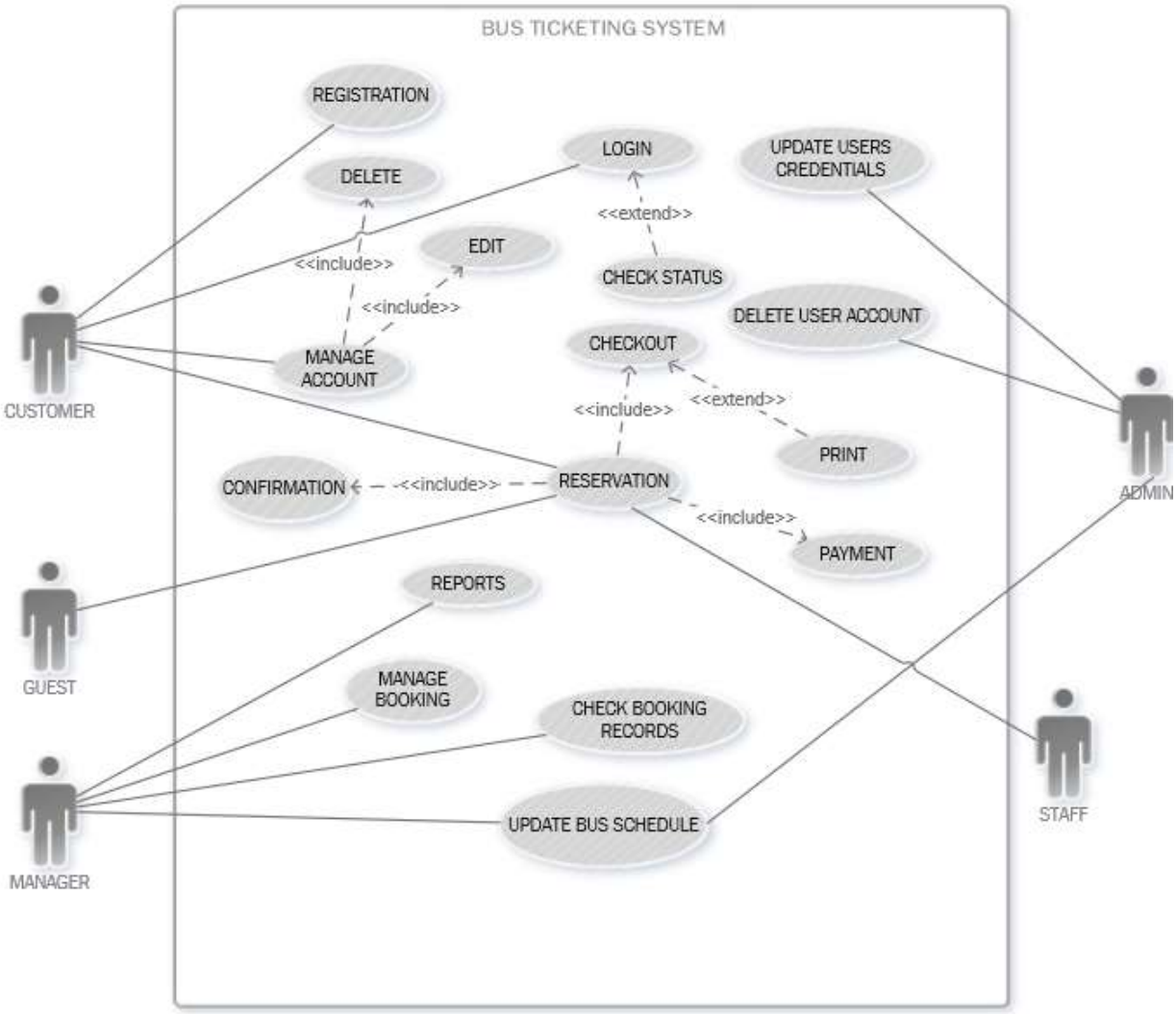
Besides, we will also take care of users acceptance testing which is completely based on what we gathered the system requirements from EZB. We will provide the results of the testing and also demonstrate the system to make it more familiar and user friendly.

Listening

One of the advantage of using this methodology is involvement of our stakeholders. We will involve users in development phases just to receive their feedback regarding the requirements and the functionalities of the system. In every iteration customers well provide their feedback and when they satisfied we will procced to next iteration.

5.0 ANALYSIS MODEL

5.1 UML (Uniform Modelling Language)



5.2 Use Case Specification

Use Case Name	Registration
Summary	Customer enter their details into the system. Details of the customers used to be validated by the system before saving into the database.
Actor	Customers
Precondition	Customer need to access the system through web browser
Main Sequences	1. Customers provide their details. 2. System validate customer's details. 3. System then save customers details into the database. 4. System register new customers. 5. System shows the message after saving the details successfully.
Alternative Sequences	N/A
Post Condition	System create account for the new customer.

Use Case Name	Login
Summary	Customer enter their username and password to access their account
Actor	Customers
Precondition	Customer need to register as a member
Main Sequences	1. Customers enter user credentials. 2. System verify users. 3. System redirect to the user page.

	4. System shows the message after successful login.
Alternative Sequences	1(a) If customers are unable to login to the system then the system will show customers "Forget Password" link where they can change their password.
Post Condition	System redirect customers to their account.

Use Case Name	Update Users Credentials
Summary	Admin can update users (staff, manager) credentials such as username and password if necessary.
Actor	Admin
Precondition	Admin need to login to the system.
Main Sequences	1. Admin login to the system. 2. Admin enter new username or password for staff and manager. 3. System validate the details provided by admin. 4. System save data into the database. 5. System update details. 6. System shows message successful after saving the details into the database.
Alternative Sequences	N/A
Post Condition	System update username or password for staff and manager.

Use Case Name	Delete User Account
Summary	Admin can delete user (staff, manager, and customer) account if necessary.
Actor	Admin
Precondition	Admin need to login to the system.
Main Sequences	1. Admin login to the system. 2. Admin access the account file of users. 3. System shows users account. 4. Admin select users account. 5. Admin click the button "Delete Account". 6. System delete the user account. 7. System shows message successful after deleting account.
Alternative Sequences	N/A
Post Condition	System delete account from database.

Use Case Name	Manage Account
Summary	Customer can manage their account through the system.
Actor	Customer
Precondition	Customers need to login to the system.
Main Sequences	1. Customer login to the system. 2. Customer access the "Manage Account" section. 3. Customer update their account details such as address, contact, email, credit card details etc.

	4. System validate customer's new details. 5. System save customers new details into the database. 6. System update customers account. 7. System shows the successful message after updating database.
Alternative Sequences	N/A
Post Condition	System update customer's details and save it in database.

Use Case Name	Reservation
Summary	Customers can reserve or book their seats using the system. Staff can also do reservation through the system.
Actor	Customer & Staff
Precondition	Customers need to login to the system as a member or access it as a guest.
Main Sequences	1. Customer (registered) login to the system. 2. Customer check the available bus schedule. 3. Customer select date and time from the bus schedule. 4. System check for the availability. 5. System confirm the availability. 6. Customer click the button "RESERVE". 7. System save reservation details into the database. 8. System reserve seats for customers.

	9. System shows message "Reservation Successful".
Alternative Sequences	N/A
Post Condition	System update customer's details and save it in database.

Use Case Name	Reports
Summary	Manager can generate reports using the system
Actor	Manager
Precondition	Manager need to login to the system
Main Sequences	1. Manager login to the system. 2. Manager access the report section into the system. 3. System shows the records of reports. 4. Manager check the reports records from the section. 5. Manager select date and click "Generate Reports". 6. System generate reports. 7. The system shows the message "Reports Generated Successfully".
Alternative Sequences	N/A
Post Condition	System generate the reports.

Use Case Name	Manage Booking
Summary	Manager can edit, update, and delete booking records of customers.

Actor	Manager
Precondition	Manager need to login to the system
Main Sequences	1. Manager login to the system. 2. Manager access the Manage Booking section into the system. 3. System shows the booking records. 4. Manager select booking id. 5. Manager edit, delete or update the details of the customers booking records. 6. System save the new booking details of the customers into the database. 7. System update the booking details. 8. System shows message "Updated Successful".
Alternative Sequences	N/A
Post Condition	System update, delete or edit the customer booking records.

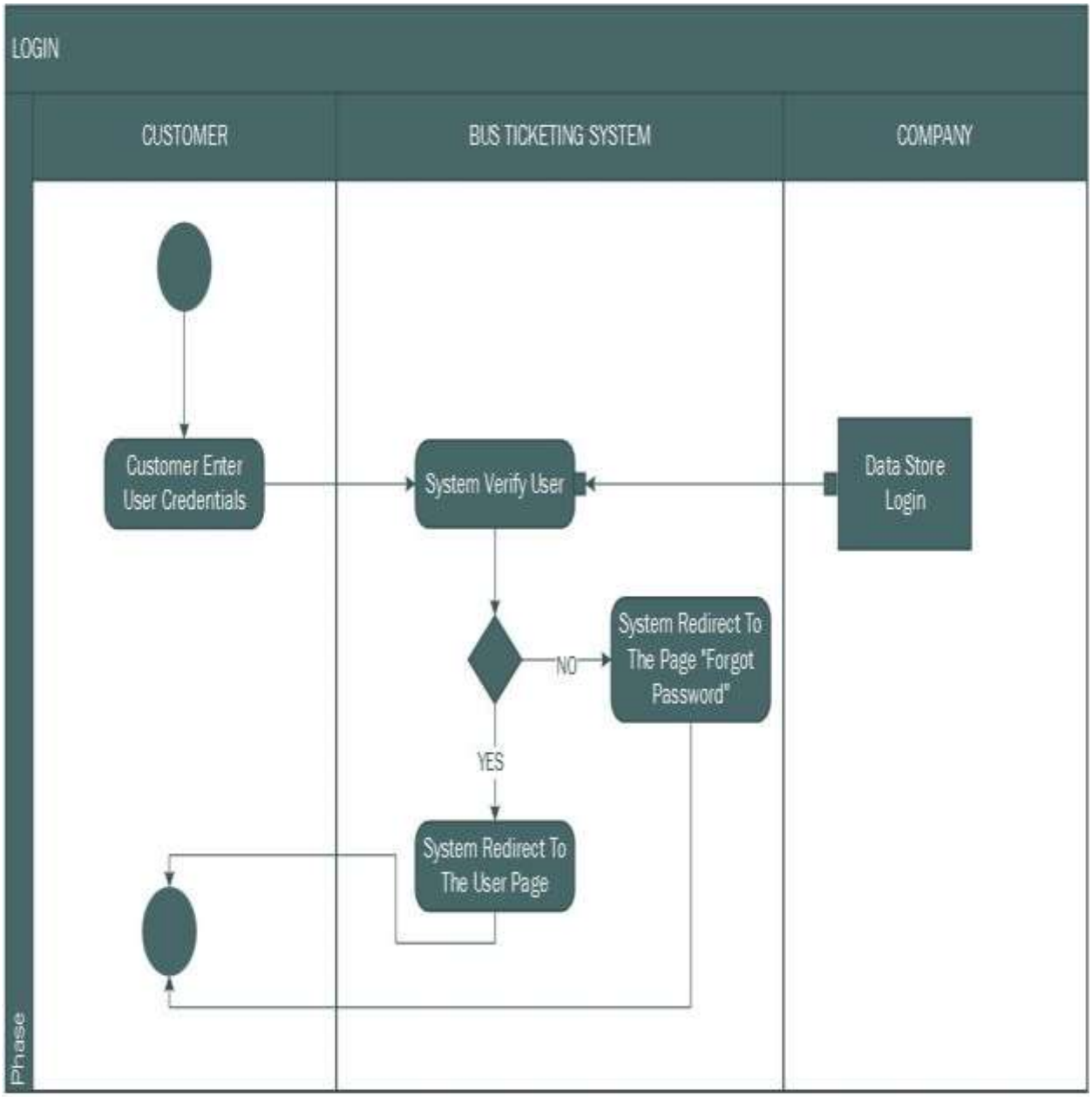
Use Case Name	Check Booking Records
Summary	Manager can check the booking records of each customers
Actor	Manager
Precondition	Manager need to login to the system
Main Sequences	1. Manager login to the system. 2. Manager access the Check Booking Records section in the system. 3. System shows all the booking records.
Alternative Sequences	N/A

Post Condition	System allows manager to check the booking records of the customers.

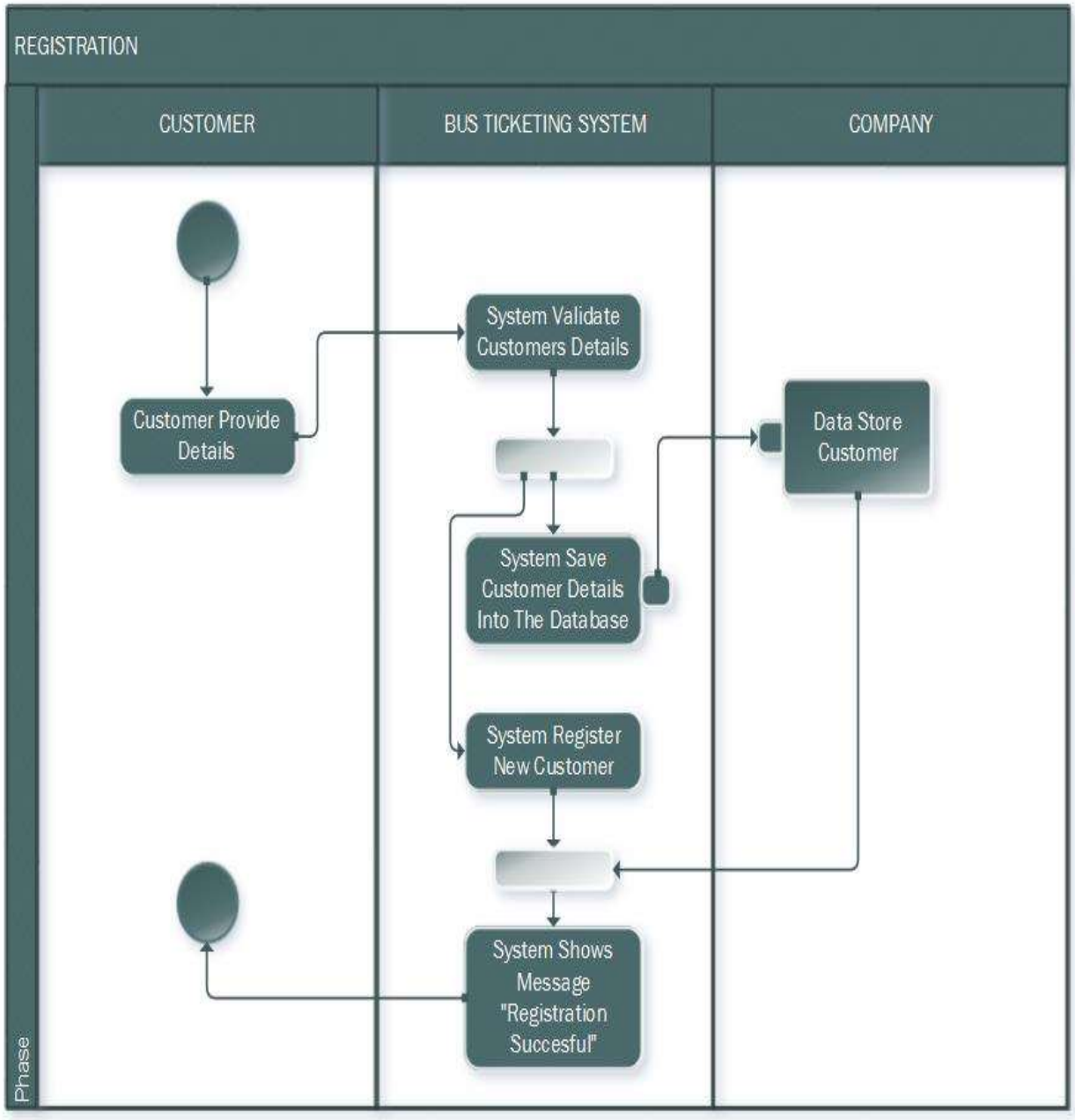
Use Case Name	Update Bus Schedule
Summary	Manager & Admin can update bus schedule
Actor	Manager & Admin
Precondition	Manager need to login to the system.
Main Sequences	1. Manager login to the system. 2. Manager access the Update Bus Schedule function through the system. 3. System shows the bus schedule. 4. Manager update the bus schedule. 5. System save new bus schedule into the database. 6. System update the new bus schedule. 7. System shows successful message after schedule update.
Alternative Sequences	N/A
Post Condition	System update the bus schedule.

5.3 Activity Diagram (Swimlane)

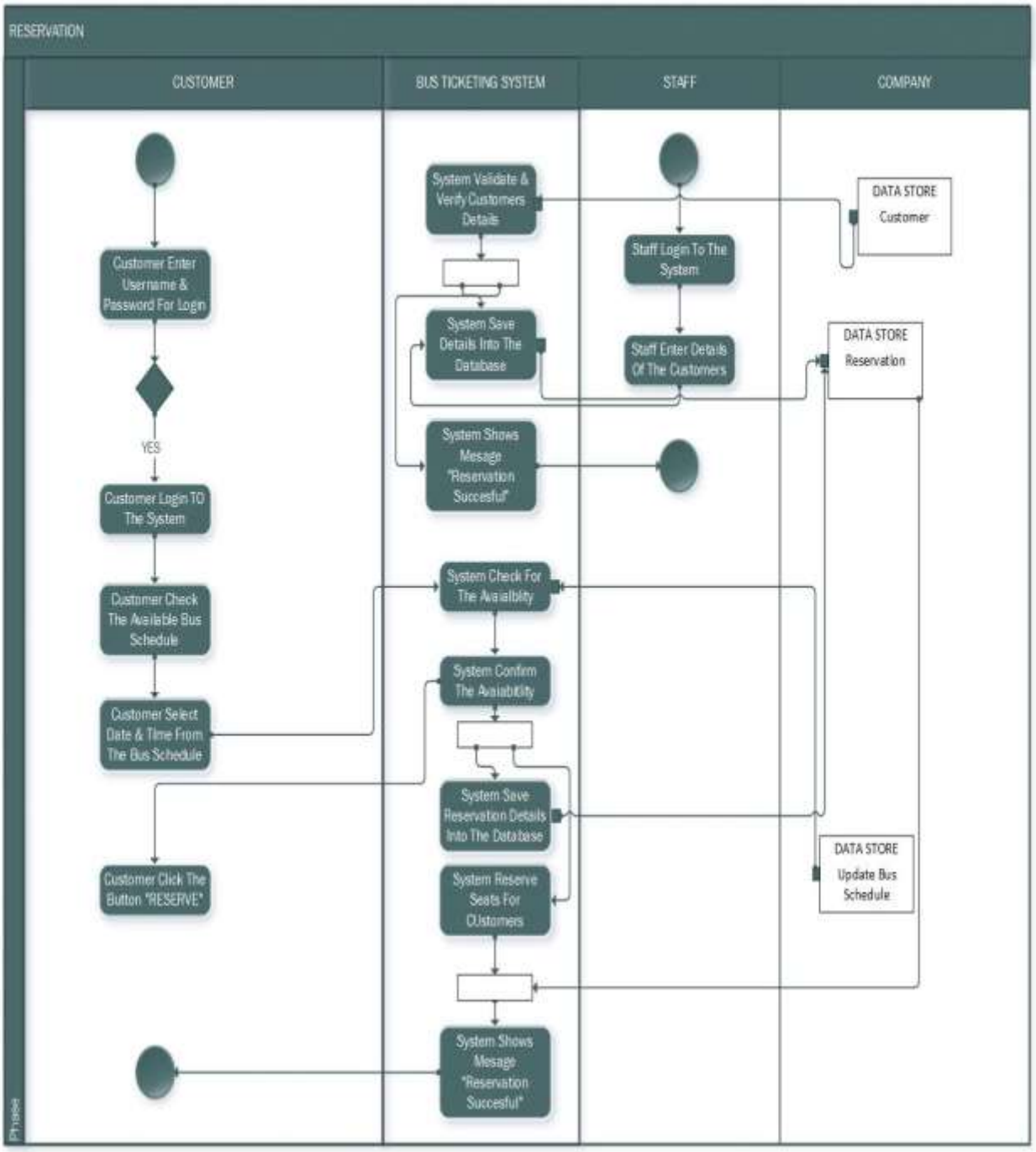
a) Login



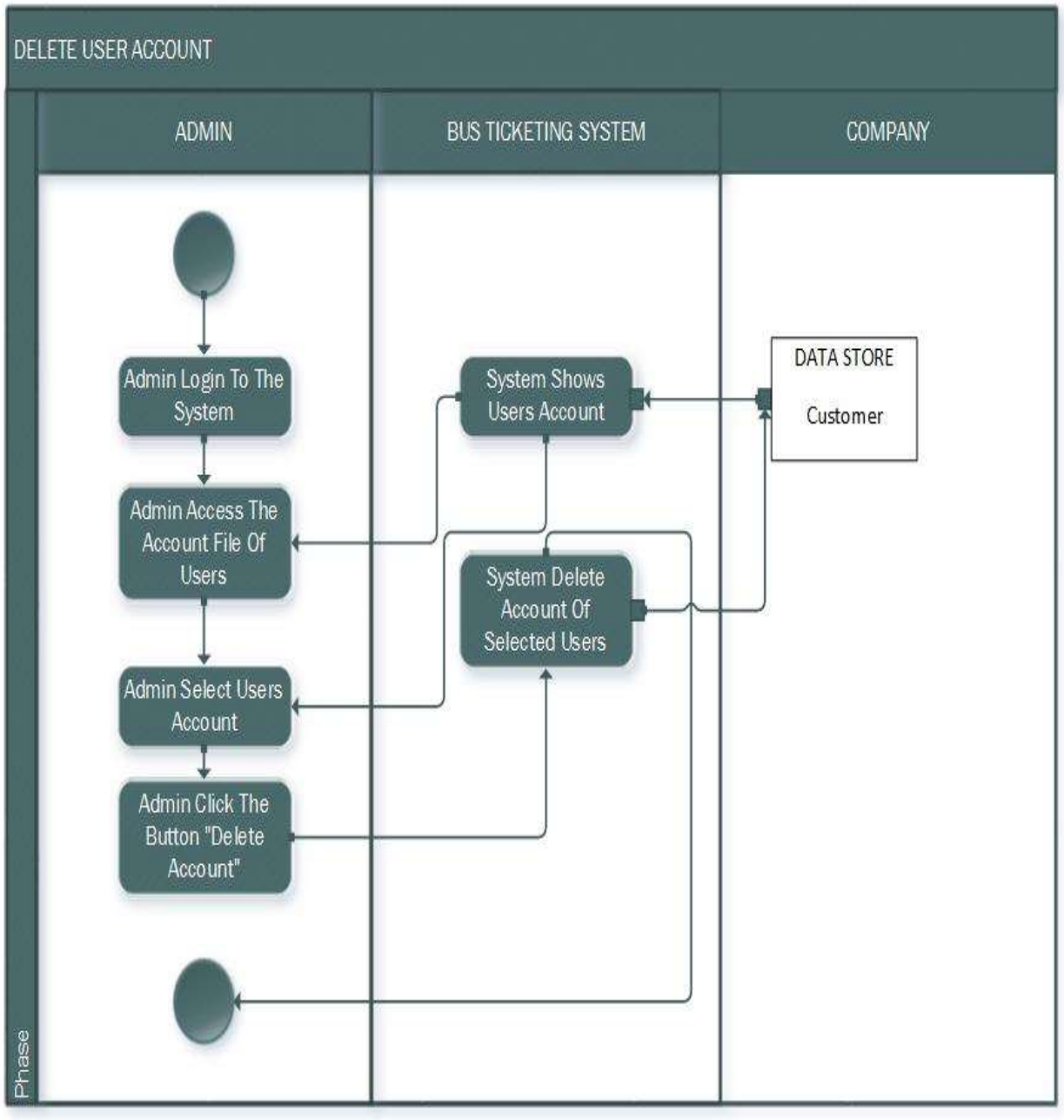
b) Registration



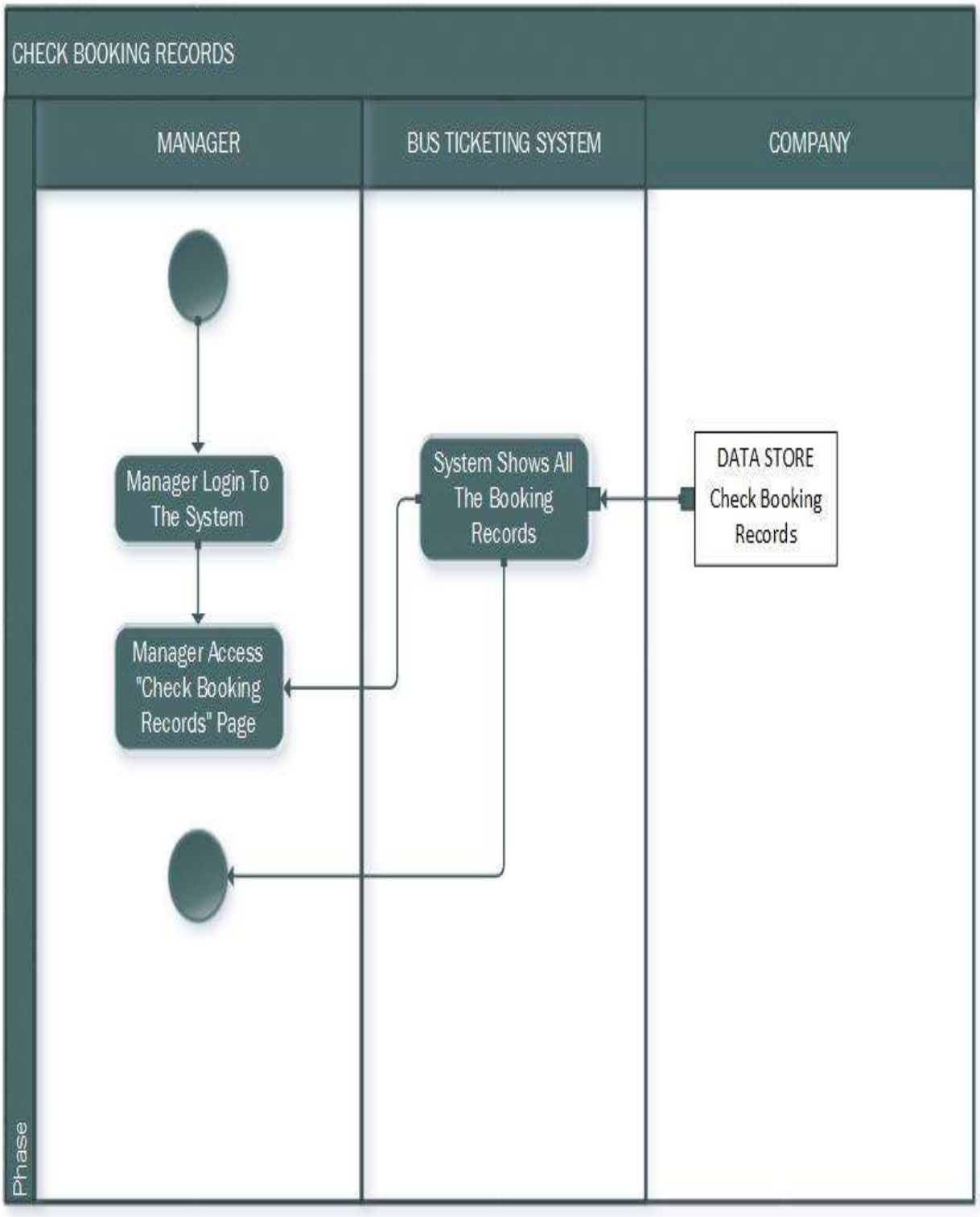
c) Reservation



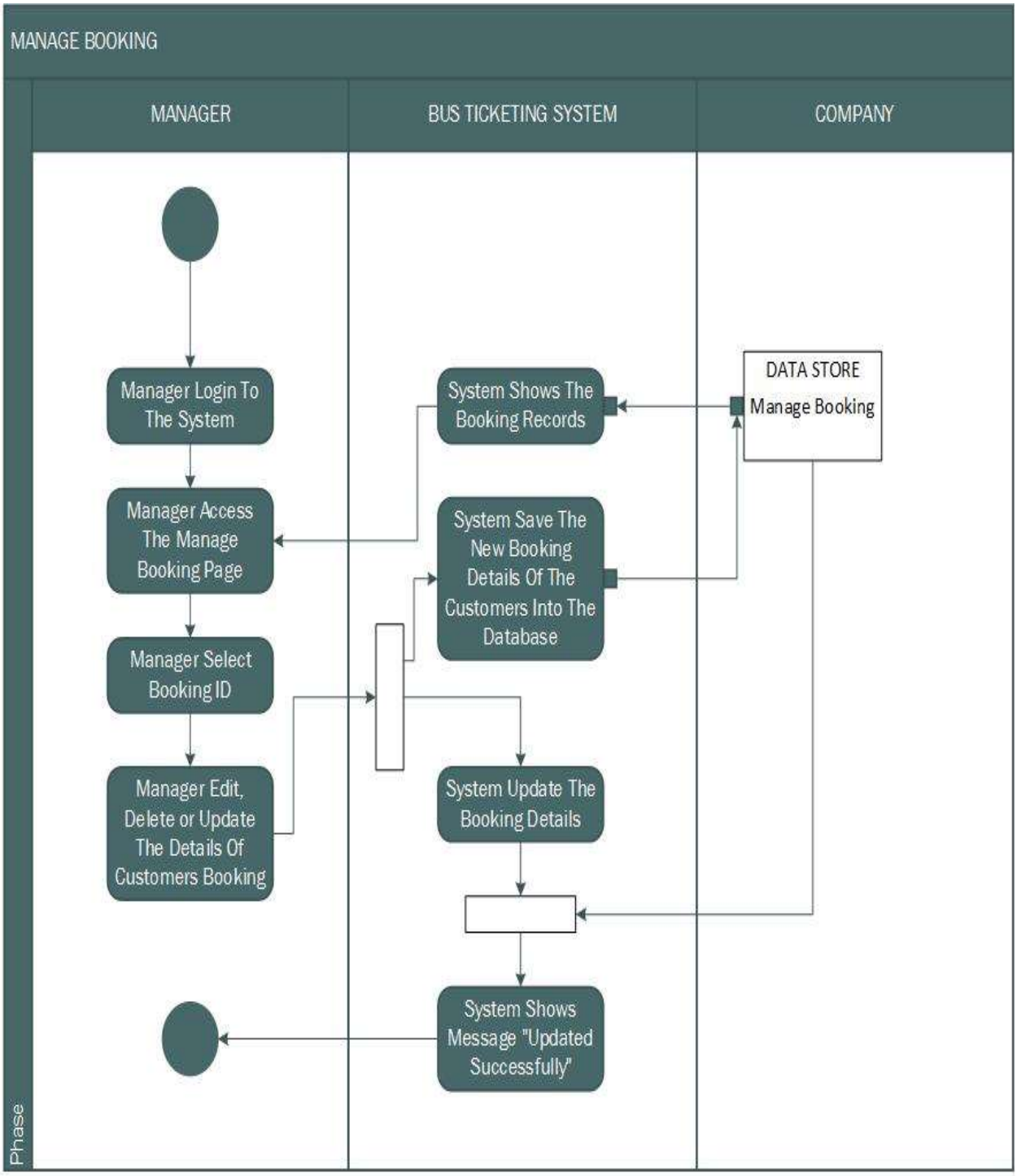
d) Delete User Account



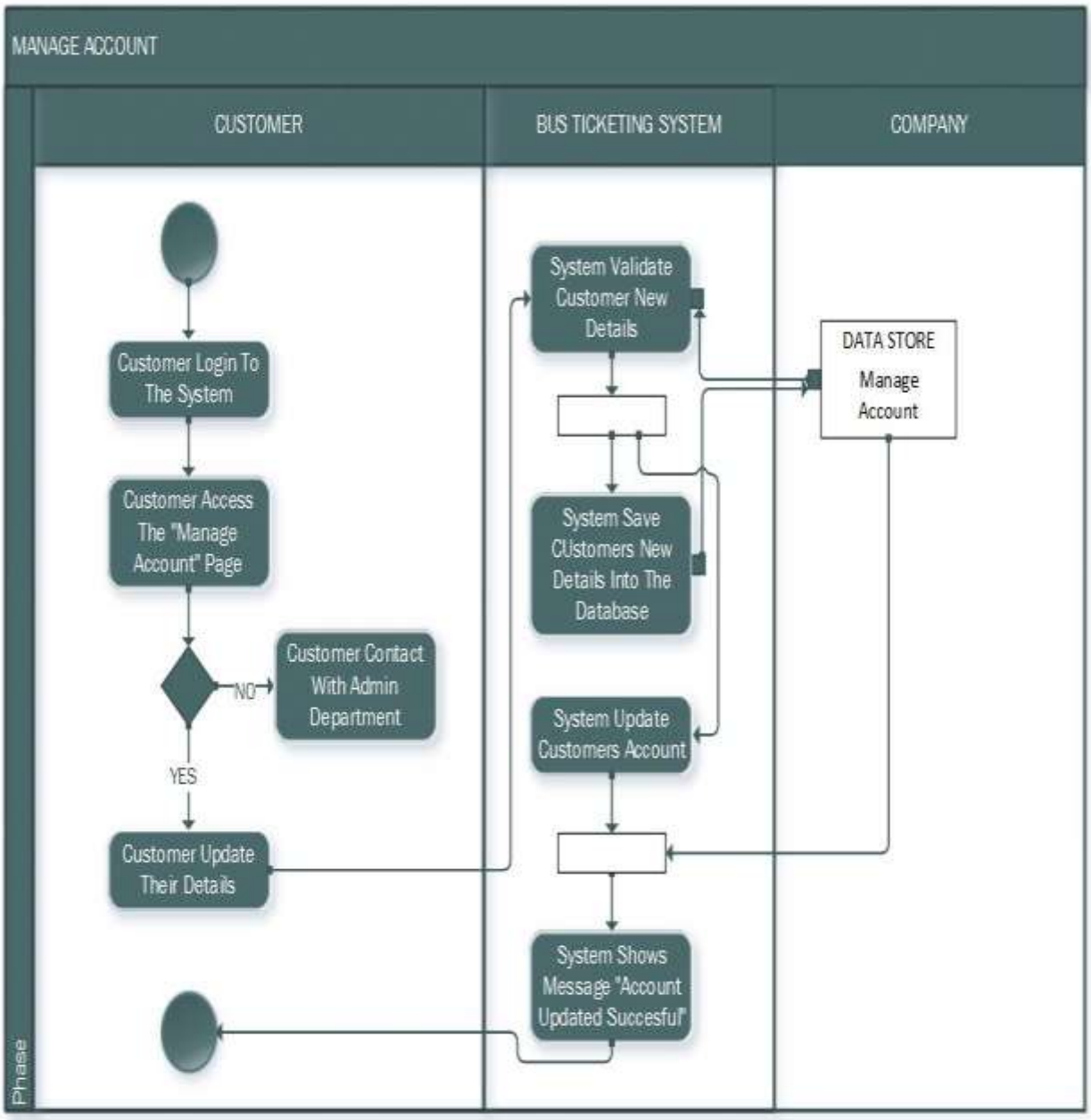
e) Check Booking Records



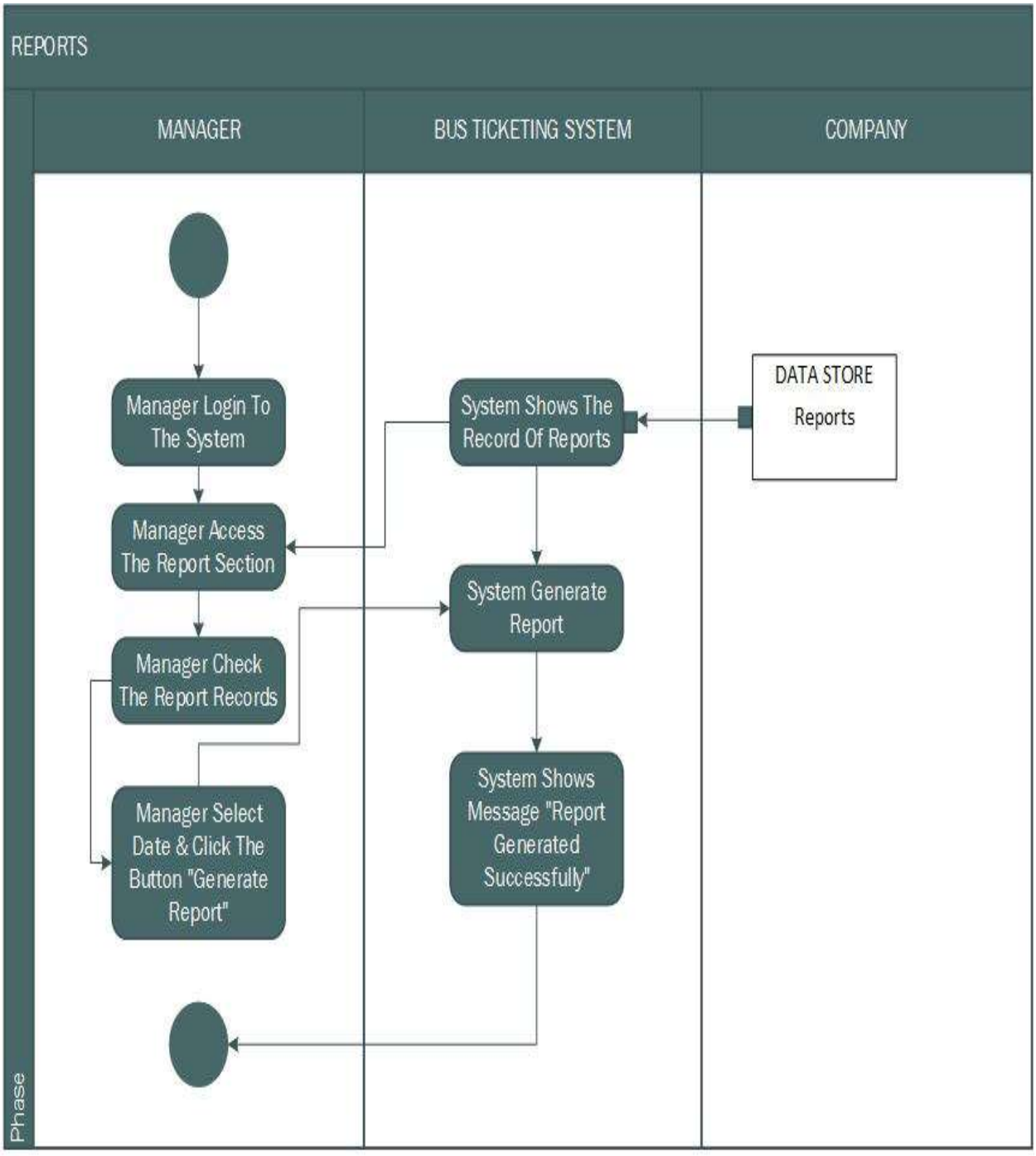
f) Manage Booking



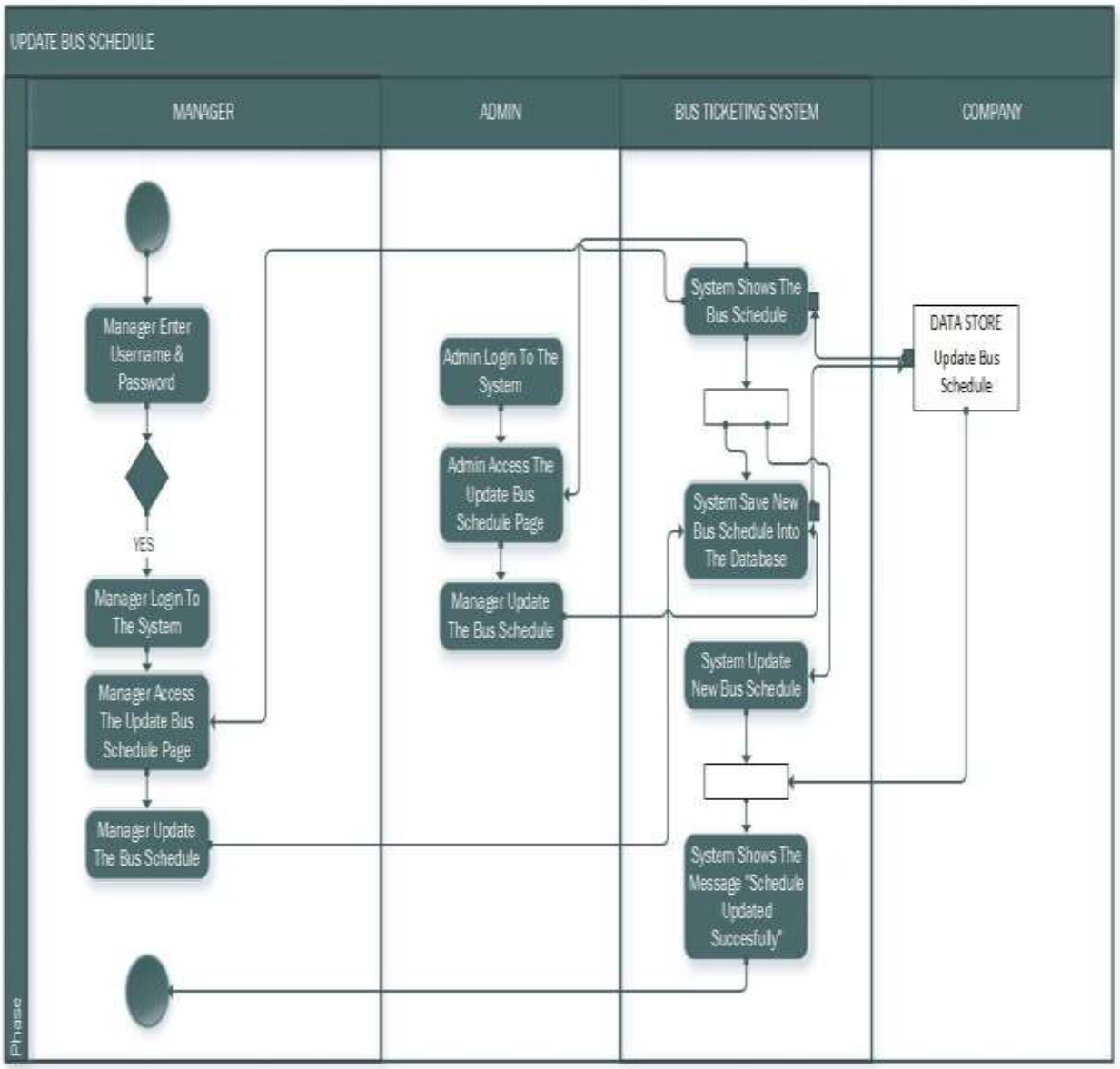
g) Manage Account



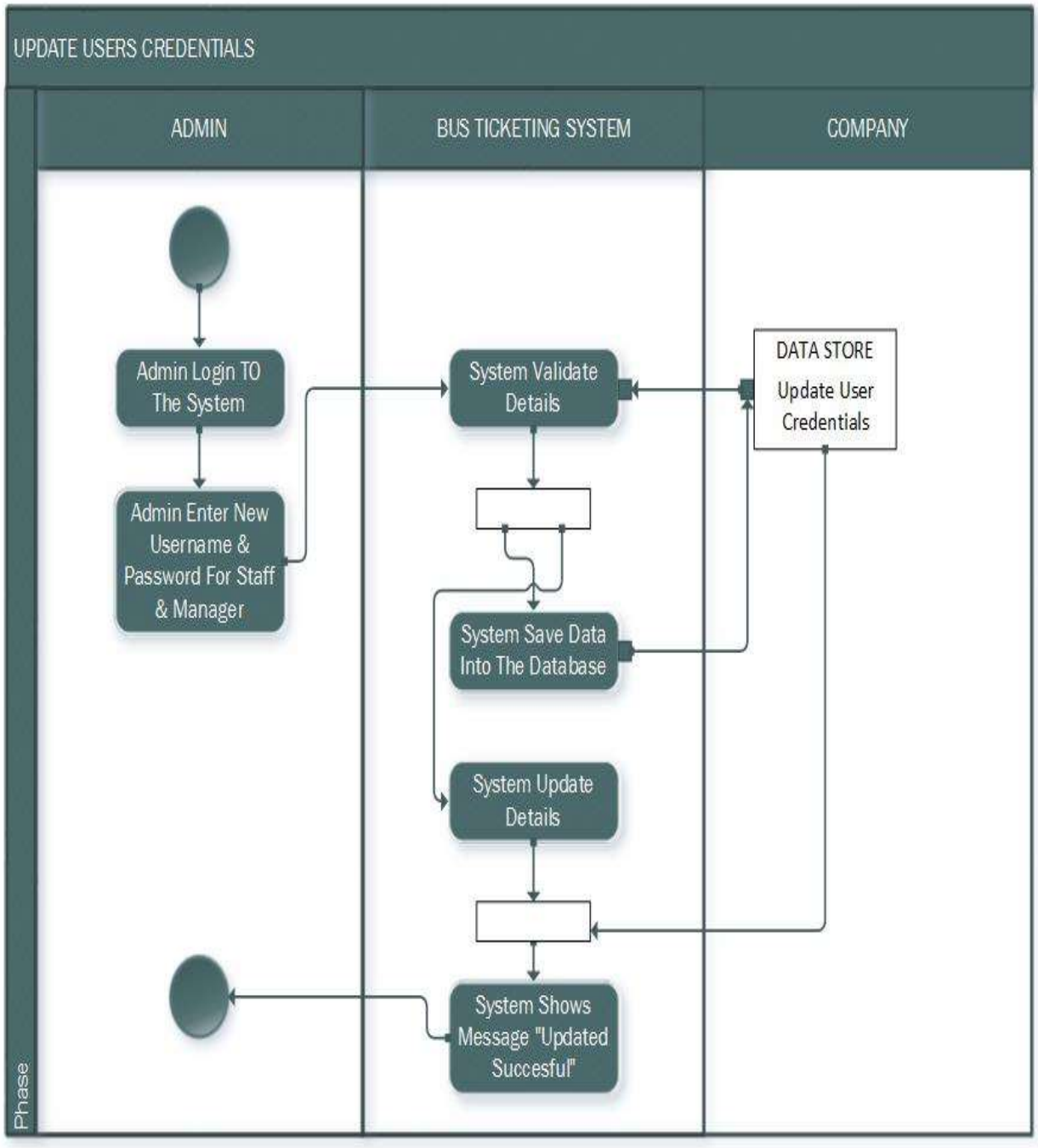
h) Reports


REPORTS
MANAGER
BUS TICKETING SYSTEM
COMPANY
Manager Login To The System
System Shows The Record Of Reports
DATA STORE
Reports
Manager Access The Report Section
System Generate Report
Manager Check The Report Records
System Shows Message "Report Generated Successfully"
Manager Select Date & Click The Button "Generate Report"
Phase

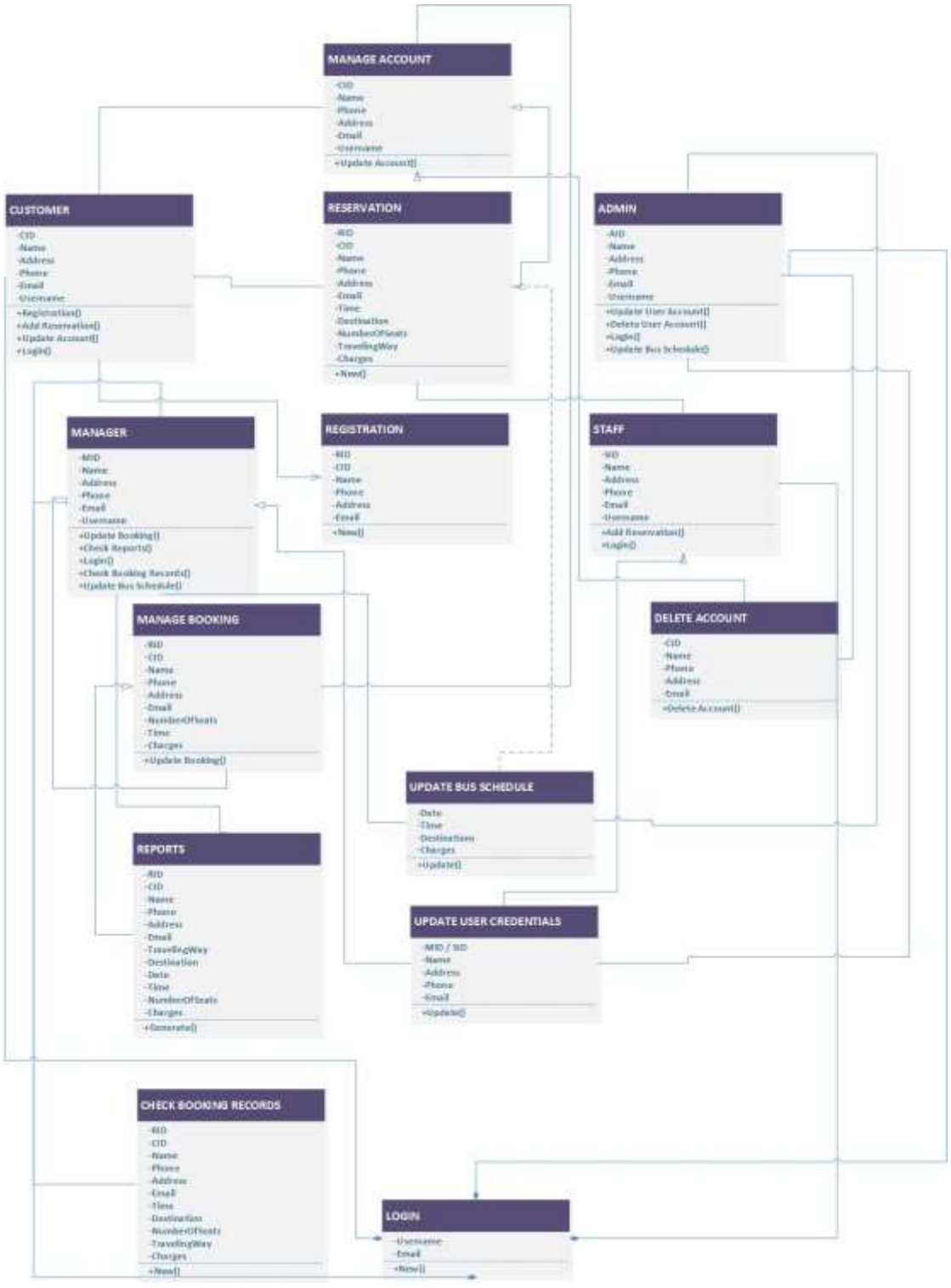
i) Update Bus Schedule



j) Update Users Credentials

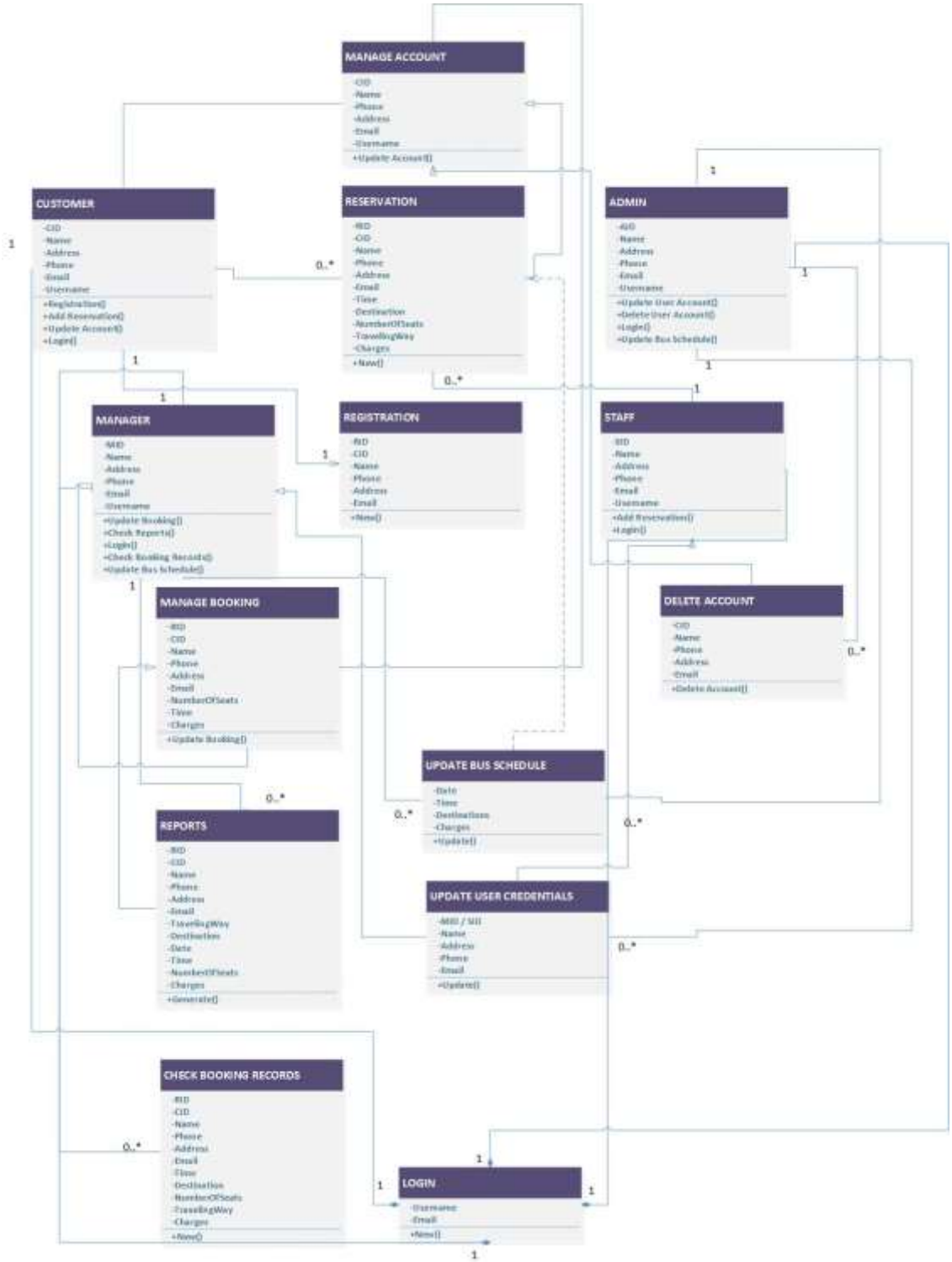


5.4 Class Diagram (Draft)



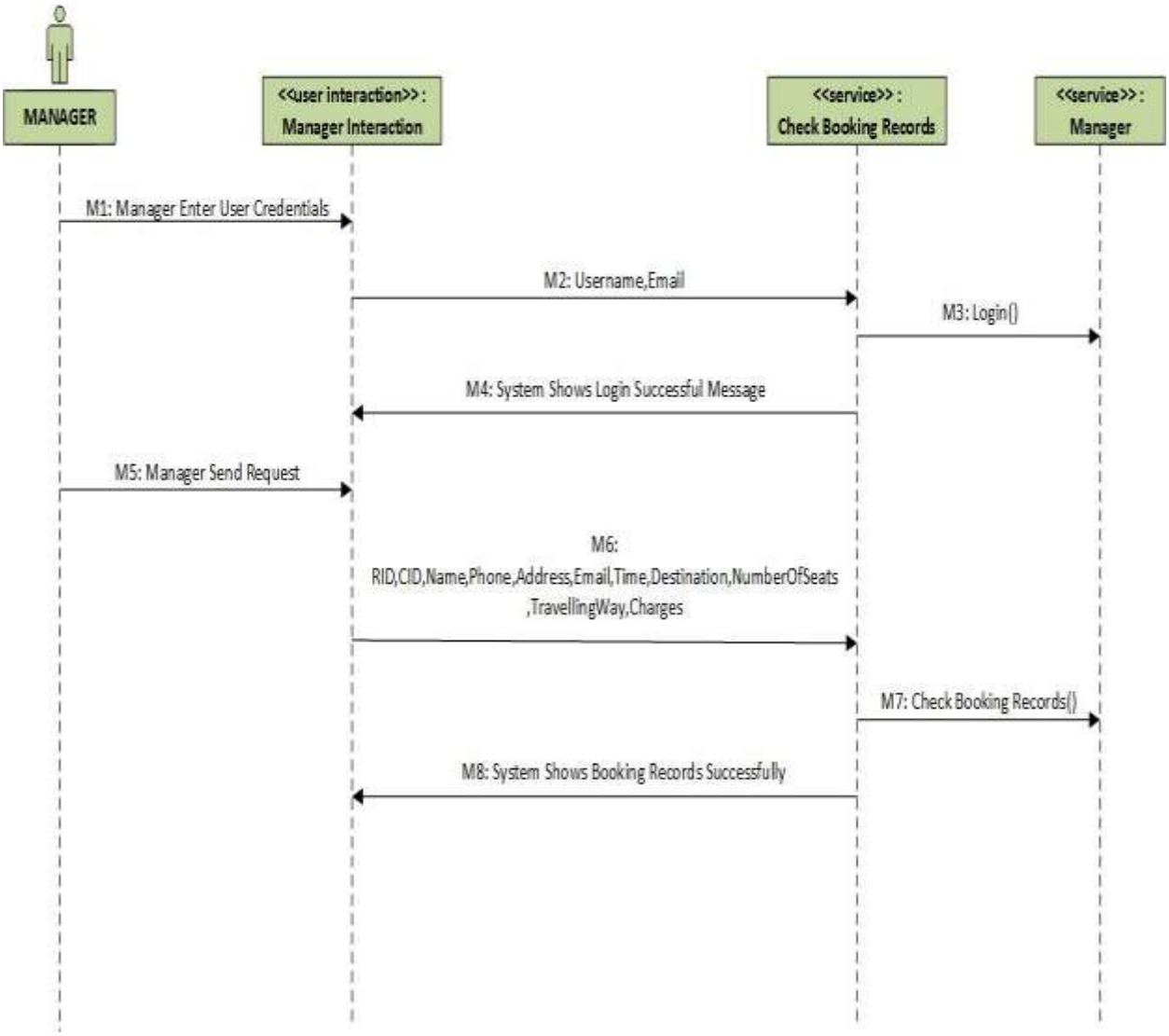
6.0 DESIGN MODEL

6.1 Class Diagram

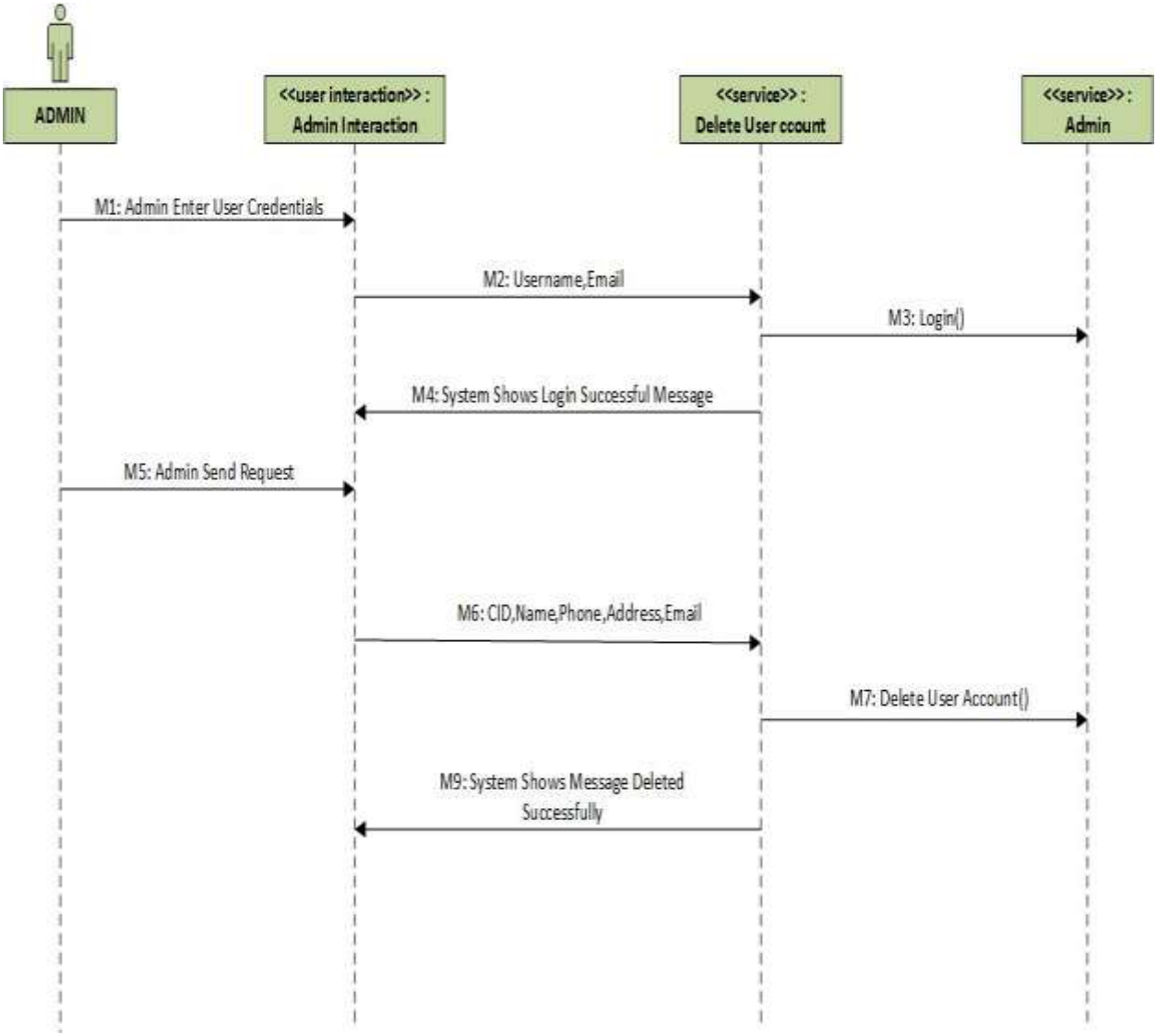


6.2 Sequence Diagram

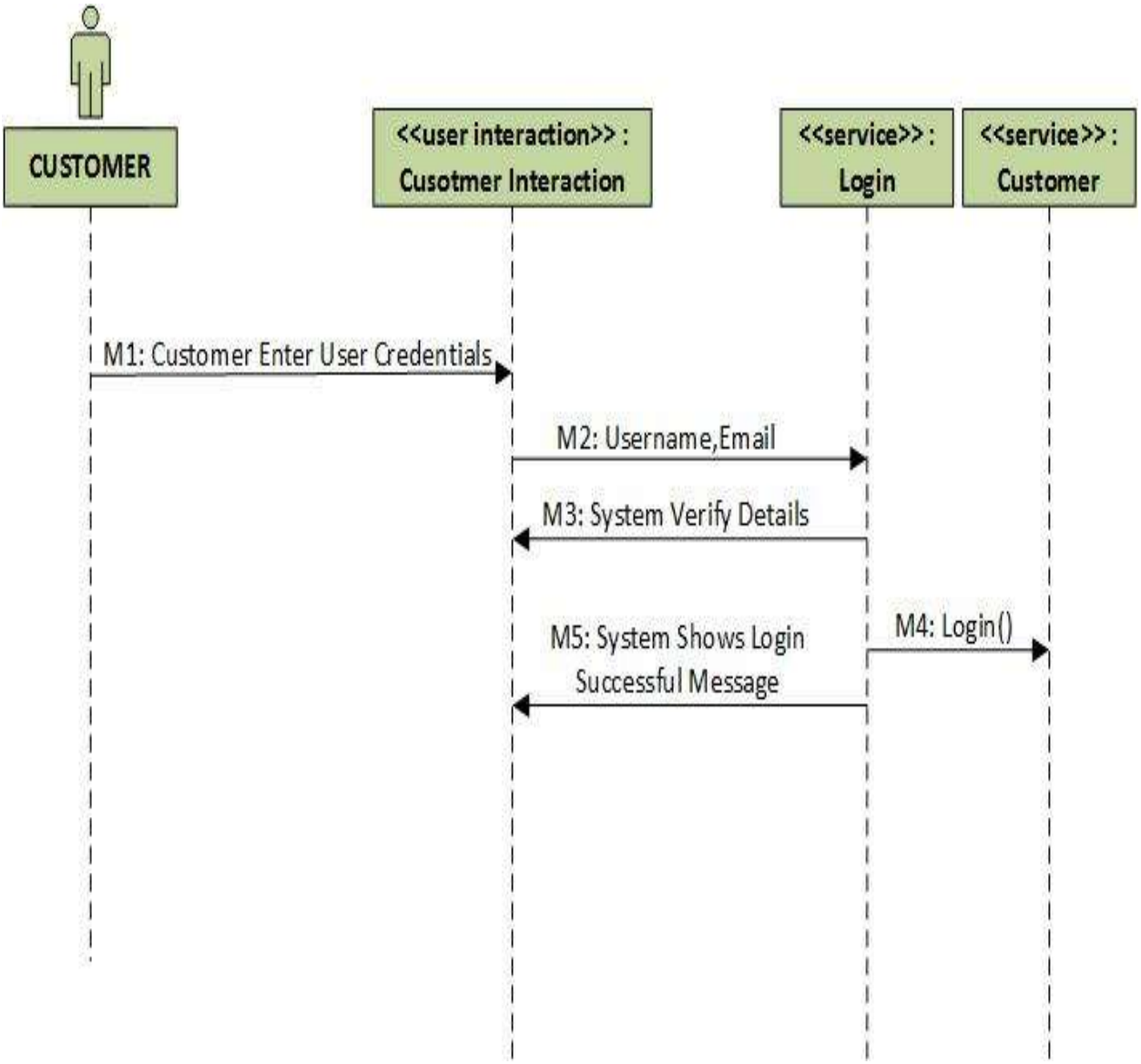
a) Check Booking Records



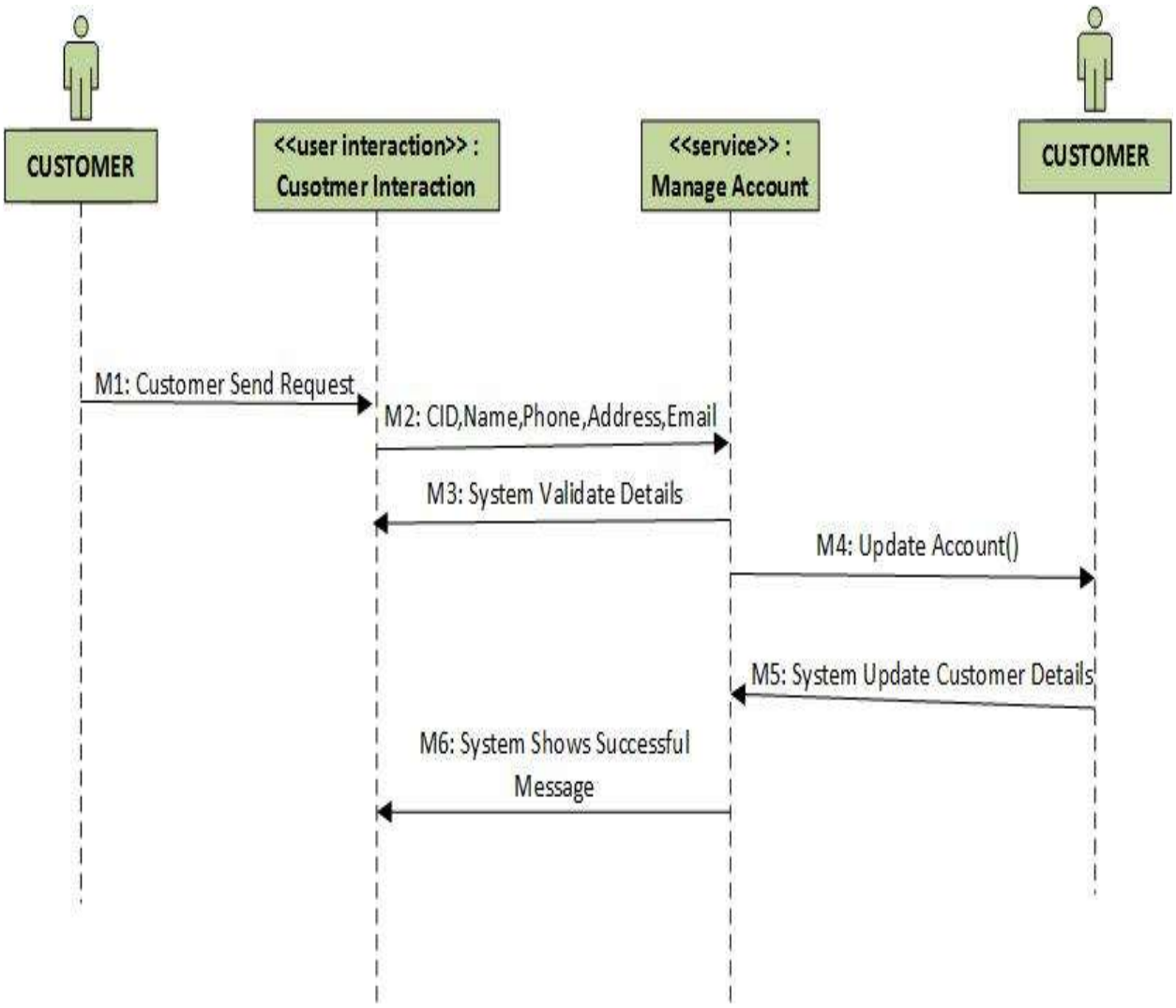
b) Delete User Account



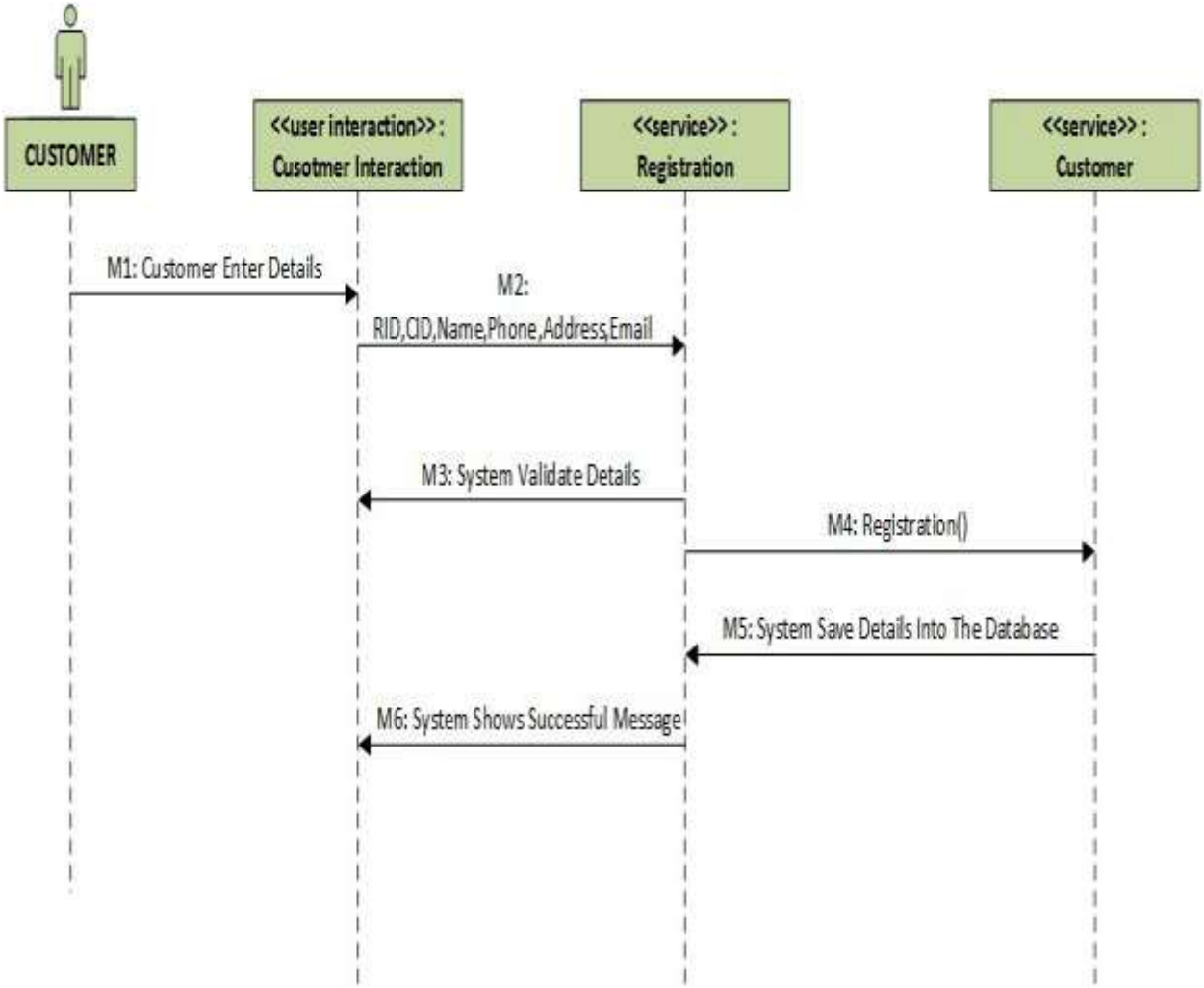
c) Login


CUSTOMER
<<user interaction>> : Cusotmer Interaction
<<service>> : Login
<<service>> : Customer
M1: Customer Enter User Credentials
M2: Username,Email
M3: System Verify Details
M4: Login()
M5: System Shows Login Successful Message

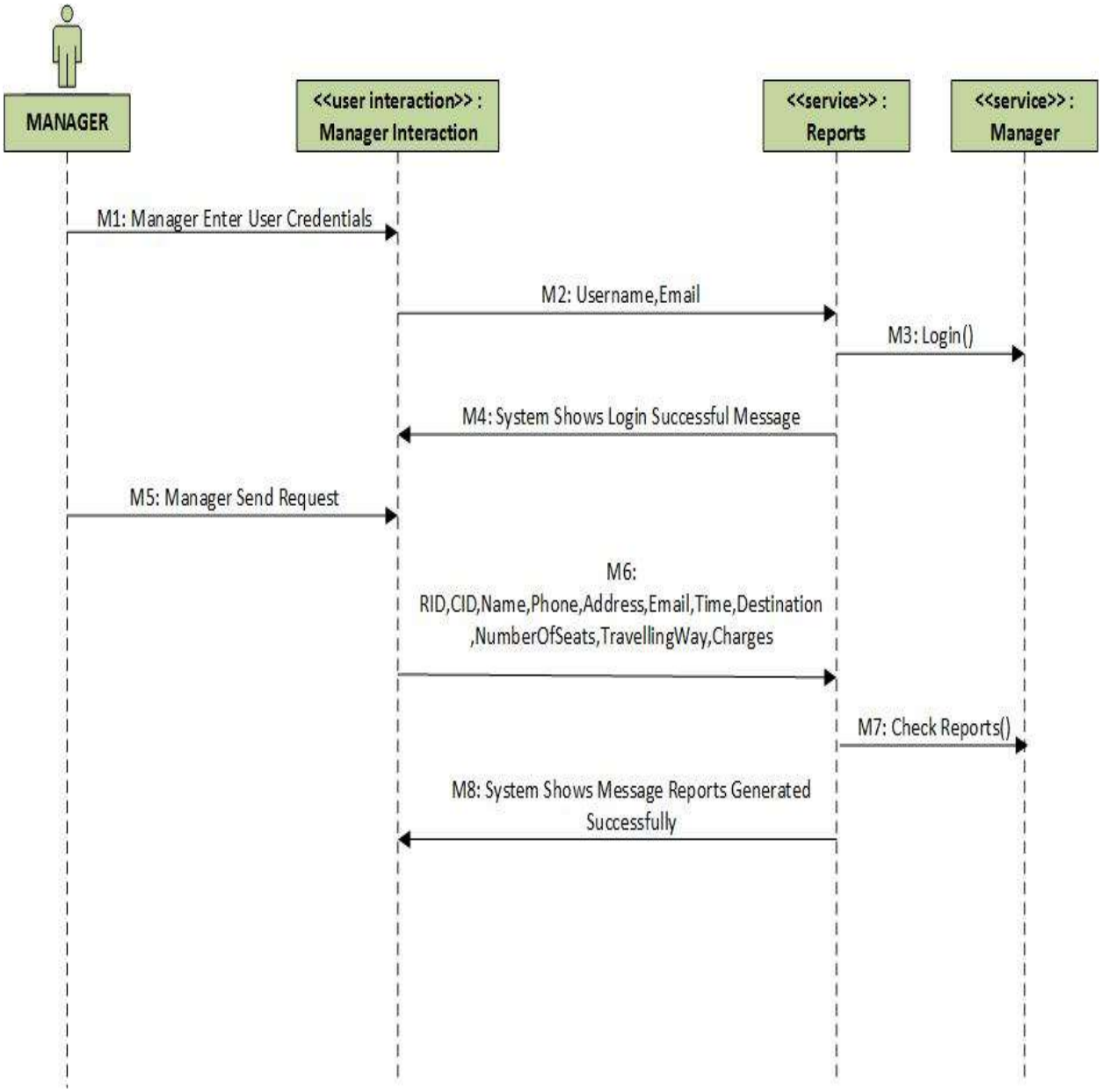
d) Manage Account



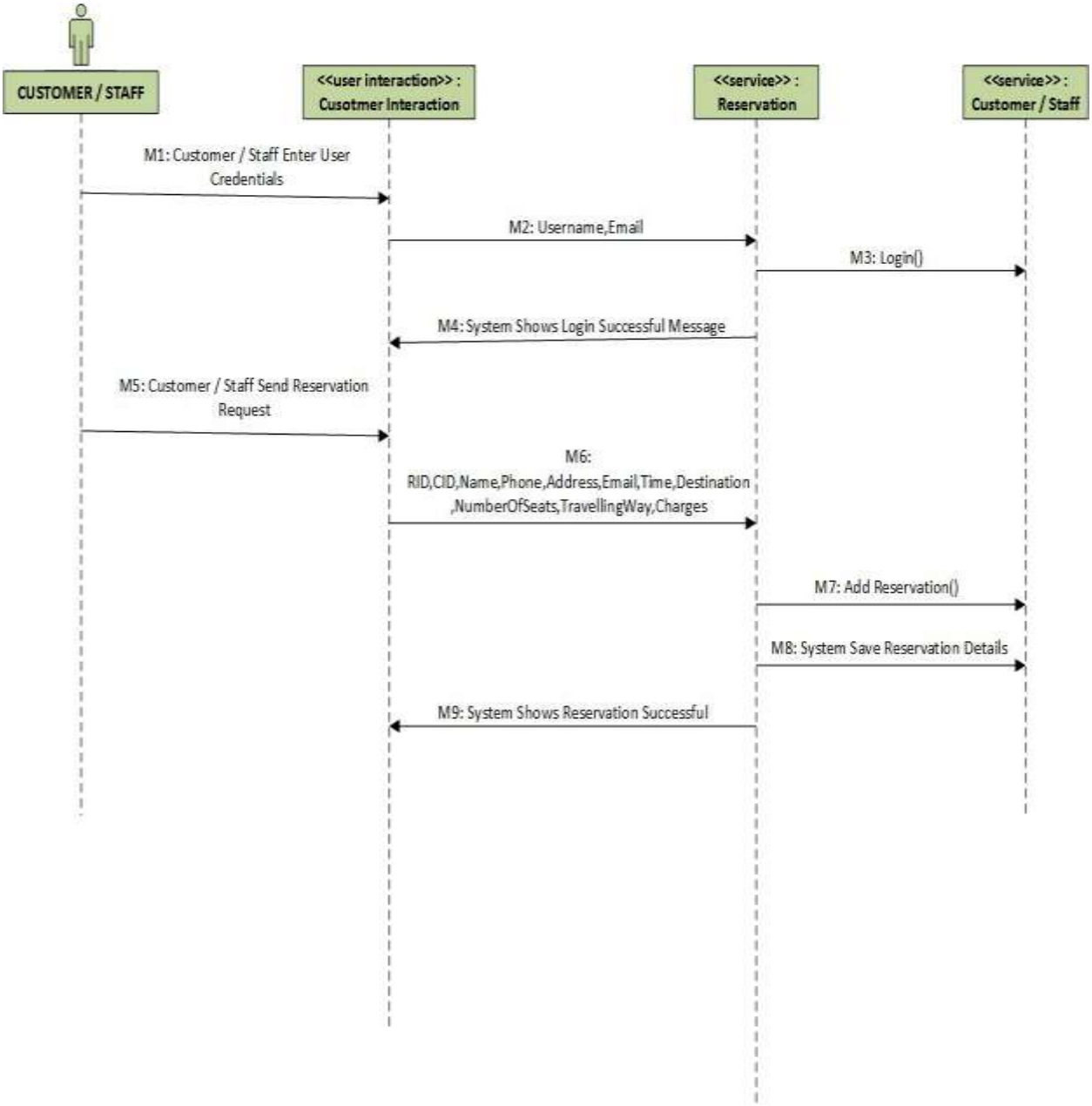
e) Registration



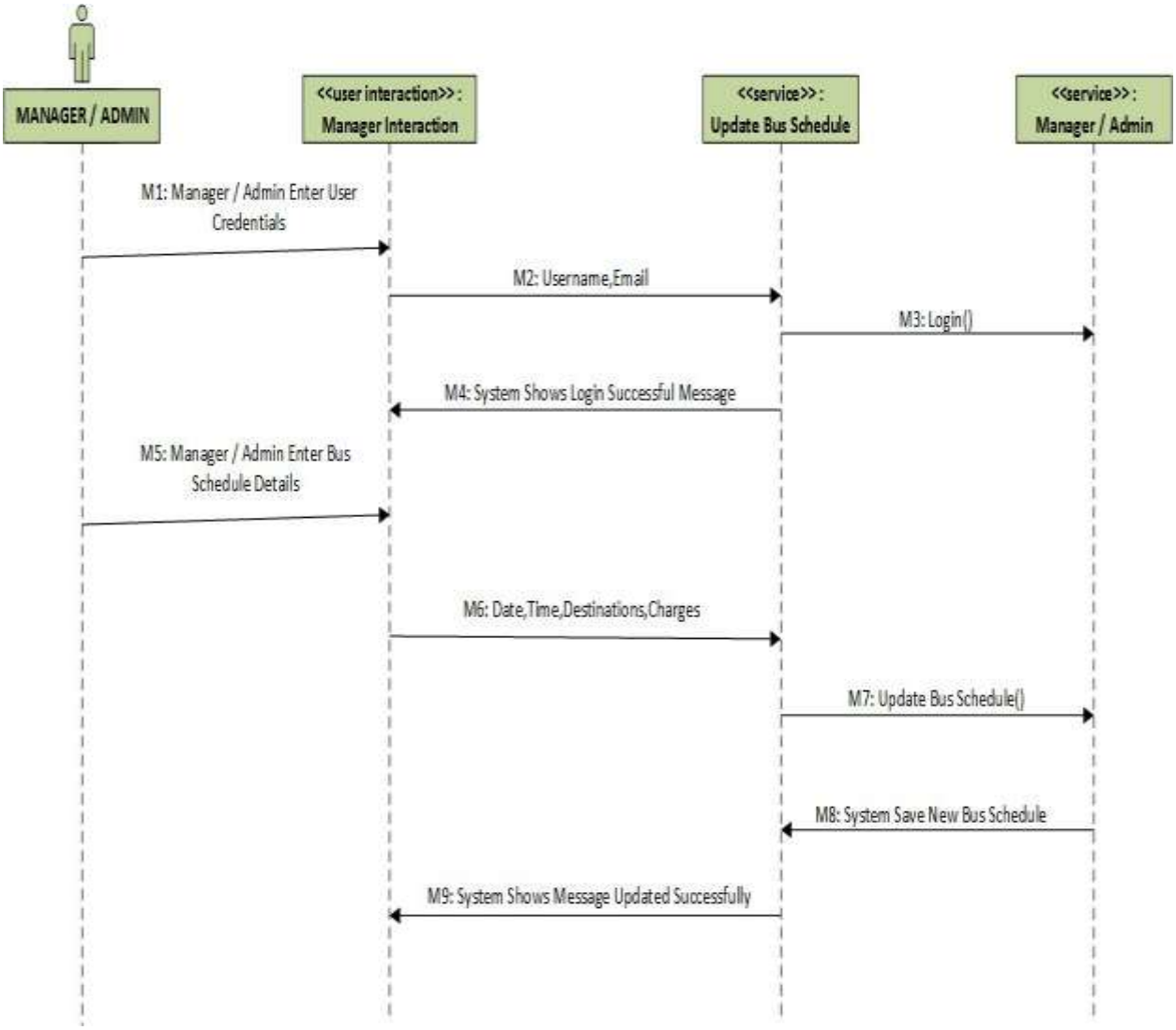
f) Reports


MANAGER
<<user interaction>> : Manager Interaction
<<service>> : Reports
<<service>> : Manager
M1: Manager Enter User Credentials
M2: Username,Email
M3: Login()
M4: System Shows Login Successful Message
M5: Manager Send Request
M6: RID,CID,Name,Phone,Address,Email,Time,Destination ,NumberOfSeats,TravellingWay,Charges
M7: Check Reports()
M8: System Shows Message Reports Generated Successfully

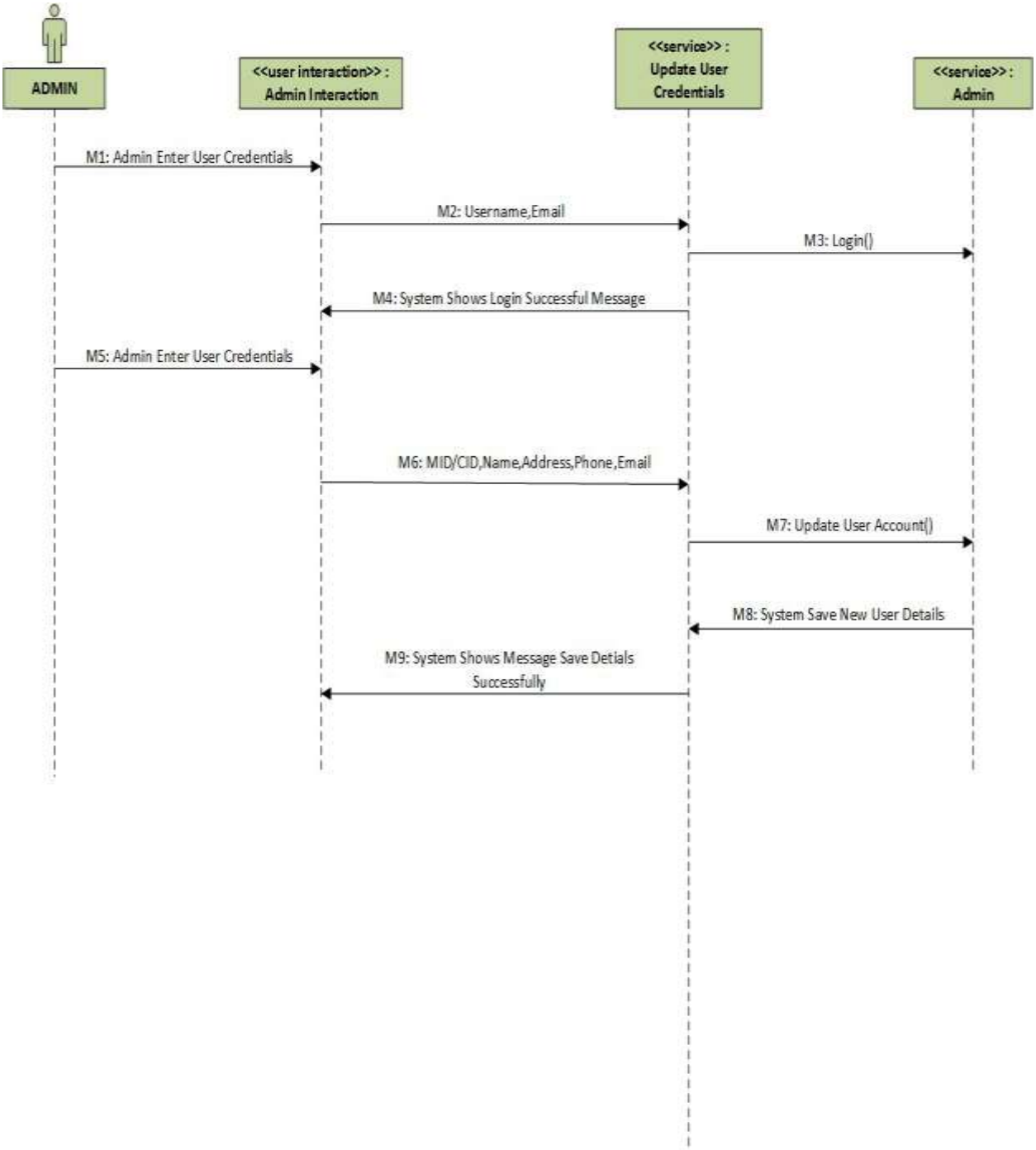
g) Reservation



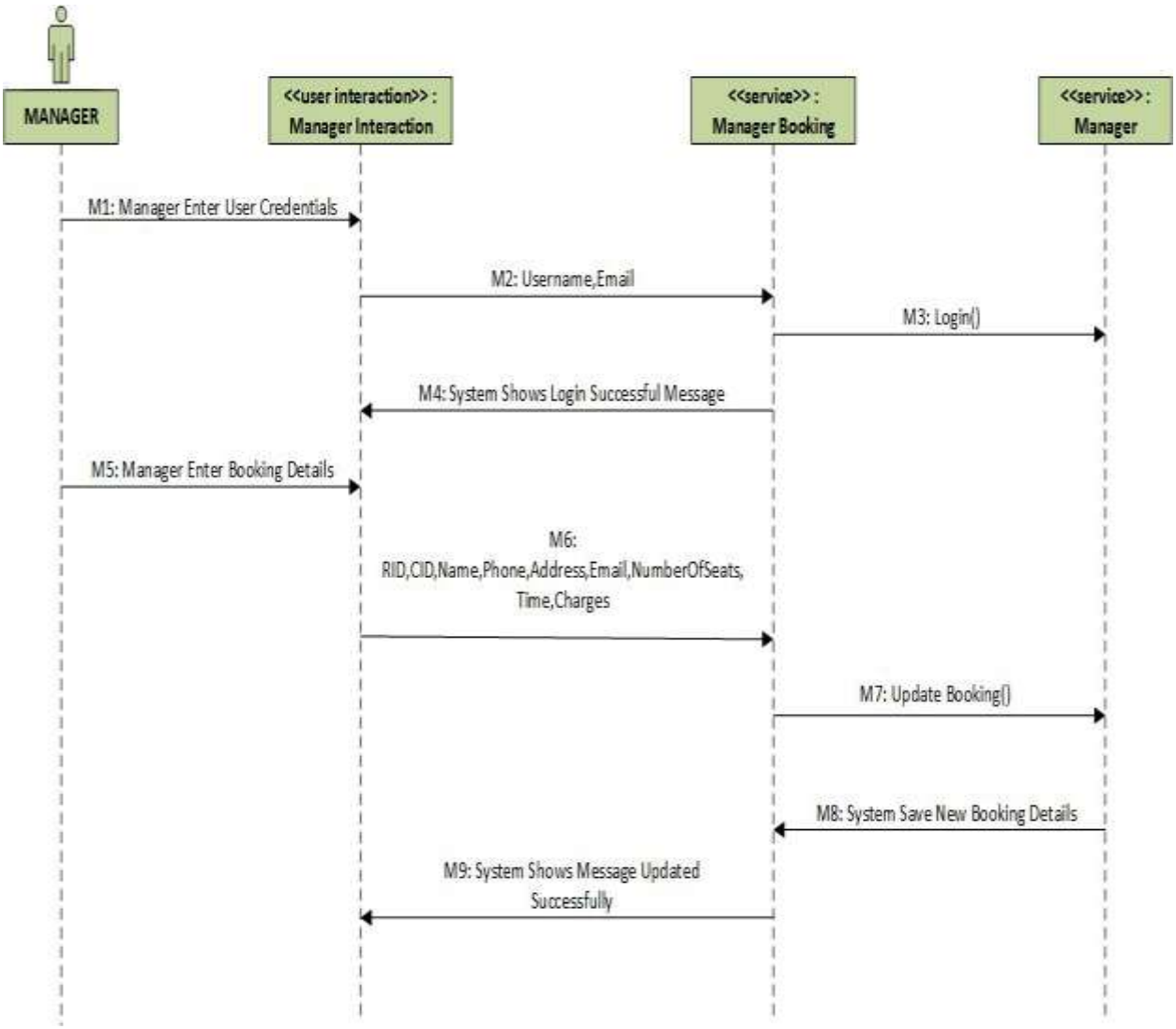
h) Update Bus Schedule



i) Update User Credentials

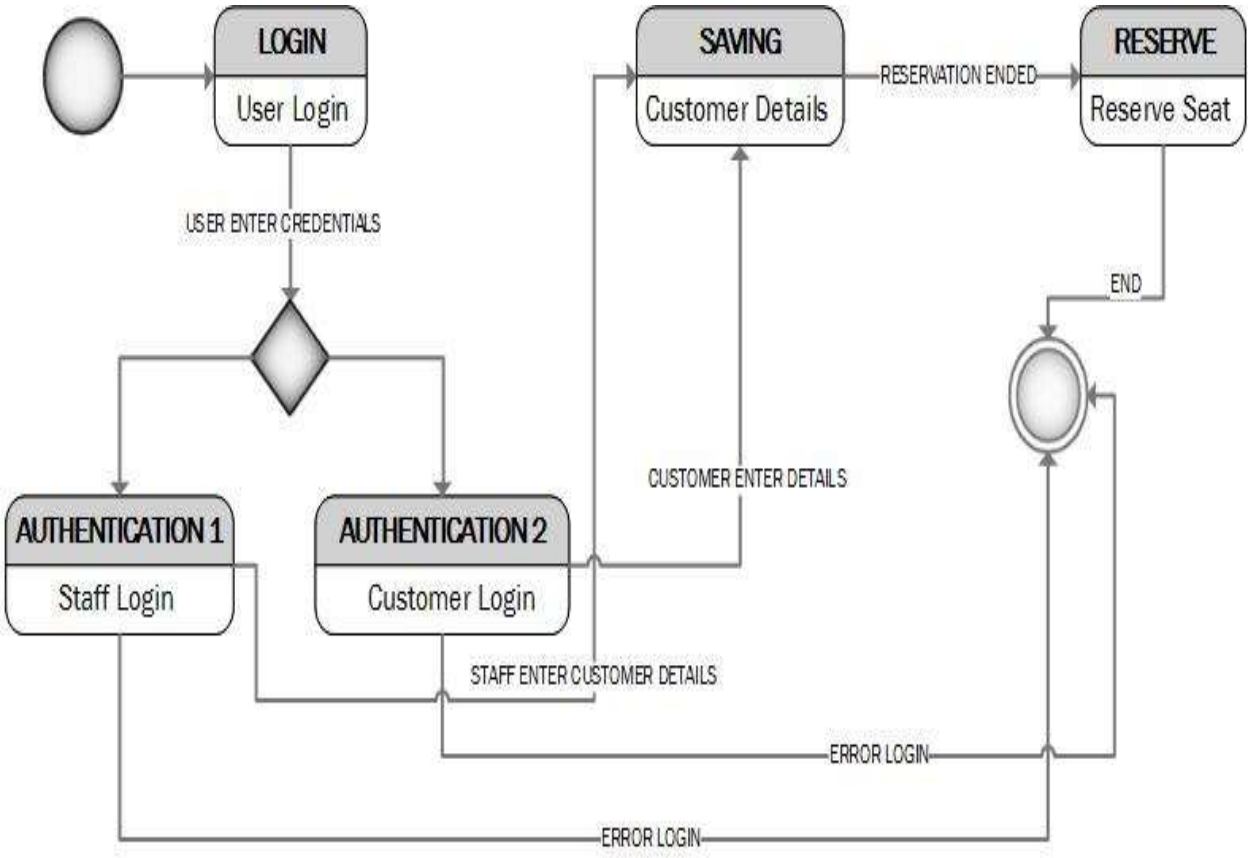


j) Manage Booking



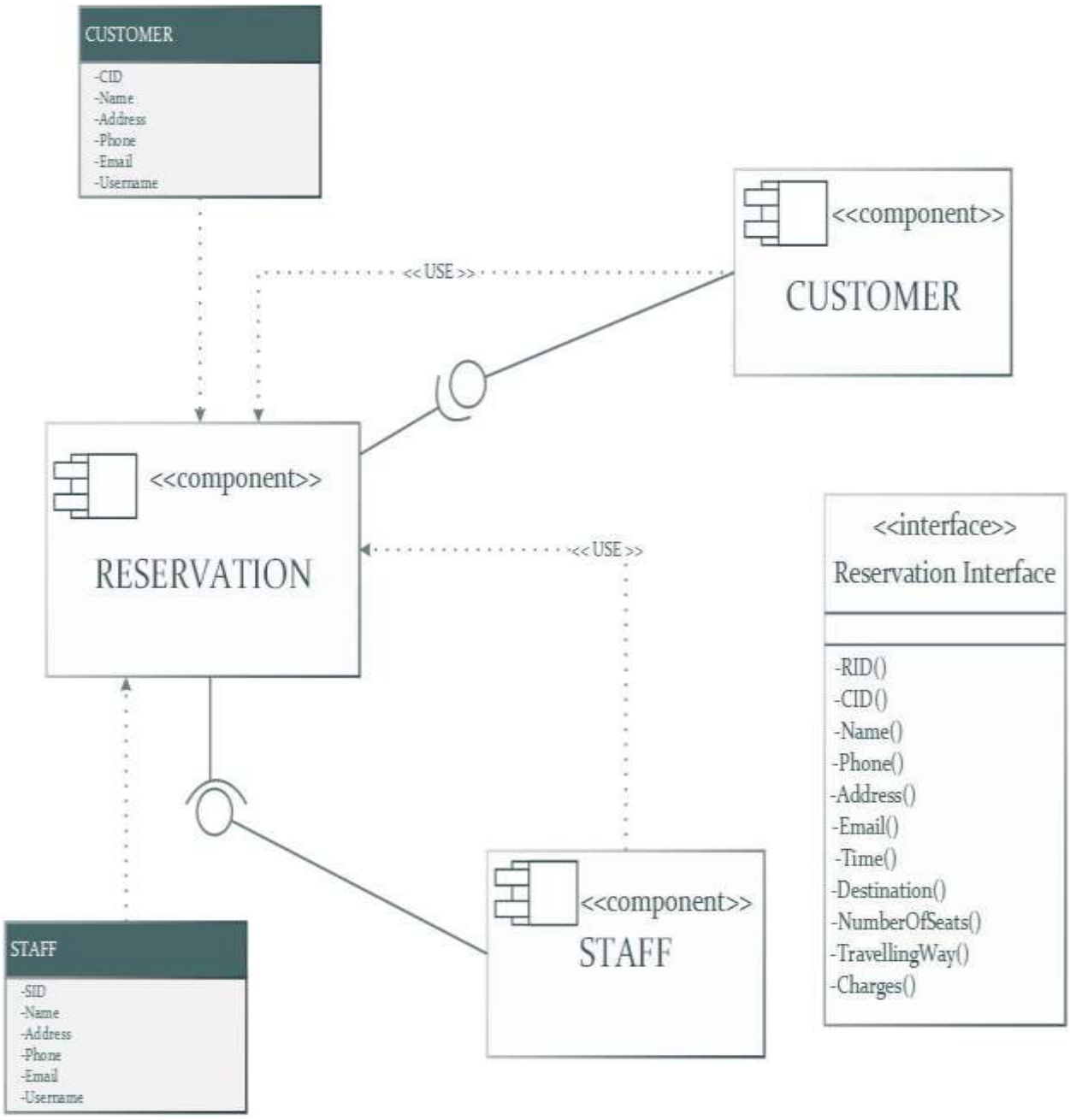
6.3 State Machine Diagram

a) Reservation



6.4 Component Diagram

a) Reservation



7.0 CONCLUSION

Although the overall system will bring some advantages for EZB, some of the facts needed to be considered. As we said earlier the system is not going to available for 24*7 and there will be a limited operation hours, EZB may lose some customers in future due to the booking time restriction. If the internet connection become down then the system will halt and no offline data can be accessed through it. Changing customers mind wont effect the system as there will be no refund after booking.

The pros of the system we can consider here is like EZB can run the business smoothly as there will be no hustle regarding managing customers details, booking details, payments and so on. This system will increase the productivity of their business as well. For the future enhancement, there will be a scope for international customers also to book from home and abroad which will boost up their business profit and also make them renown worldwide.

Besides, they also need to provide hands on training to their staffs so that they will become more familiar to the system and also make it easy for them to book seats on behalf of customers. Restriction in PayPal money transfer is also an issue. So customers from abroad will face some difficulties on payment.

Auto backup will be implemented in future as currently we designed the backup system manually. If the system crash then all the data will be erased so staff should take care for keeping the data safe by saving it every day after office hours.

From the above discussion we can say that although the system will have some cons, somehow it will have some positive impact of EZB business process. Comparing to current system which is just a manual book keeping, this online web system will automate their day to day business process.

8.0 CODE

```
#include<iostream>
#include<string.h>
using namespace std;

class Customer{
public:
    void setID(string id){
        CID = id;
    }

    string getID(){
        return CID;
    }

public:
    void setName(string name){
        Name = name;
    }

    string getName(){
        return Name;
    }
```

```
void setAddress(string address){
    Address = address;
}

string getAddress(){
    return Address;
}

void setPhone(int phone){
    Phone = phone;
}

int getPhone(){
    return Phone;
}

void getEmail(string email){
    Email = email;
}
```

```
    string setEmail(){
        return Email;
    }

    void getUsername(string usn){
        Username = usn;
    }
    string setUsername(){
        return Username;
    }
private:
    string CID;
    string Name;
    string Address;
    int Phone;
    string Email;
    string Username;
};
```

```
class Staff{

public:
    void setID(string id){
        SID = id;
    }

    string getID(){
        return SID;
    }

public:
    void setName(string name){
        Name = name;
    }

    string getName(){
        return Name;
    }
```

```
void setAddress(string address){
    Address = address;
}

string getAddress(){
    return Address;
}

void setPhone(int phone){
    Phone = phone;
}

int getPhone(){
    return Phone;
}

void getEmail(string email){
    Email = email;
}
```

```
    string setEmail(){
        return Email;
    }

    void getUsername(string usn){
        Username = usn;
    }
    string setUsername(){
        return Username;
    }
private:
    string SID;
    string Name;
    string Address;
    int Phone;
    string Email;
    string Username;

};
```

```
class Manager
{
public:
    void setID(string id){
        MID = id;
    }

    string getID(){
        return MID;
    }

public:
    void setName(string name){
        Name = name;
    }

    string getName(){
        return Name;
    }
```

```
    void setAddress(string address){
        Address = address;
    }

    string getAddress(){
        return Address;
    }

    void setPhone(int phone){
        Phone = phone;
    }

    int getPhone(){
        return Phone;
    }

    void getEmail(string email){
        Email = email;
    }
```

```
    string setEmail(){
        return Email;
    }

    void getUsername(string usn){
        Username = usn;
    }
    string setUsername(){
        return Username;
    }
private:
    string MID;
    string Name;
    string Address;
    int Phone;
    string Email;
    string Username;
};
```

```
class Admin{
public:
    void setID(string id){
        AID = id;
    }

    string getID(){
        return AID;
    }

    void setName(string name){
        Name = name;
    }

    string getName(){
        return Name;
    }
```

```
void setAddress(string address){
    Address = address;
}

string getAddress(){
    return Address;
}

void setPhone(int phone){
    Phone = phone;
}

int getPhone(){
    return Phone;
}

void getEmail(string email){
    Email = email;
}
```

```
string setEmail(){
    return Email;
}

void getUsername(string usn){
    Username = usn;
}
string setUsername(){
    return Username;
}
void setPhone(int phone){
    Phone = phone;
}

int getPhone(){
    return Phone;
}
```

```
private:
    string AID;
    string Name;
    string Address;
    int Phone;
    string Email;
    string Username;
};

class Reservation{
public:
    void setrid(string Rid){
        RID = Rid;
    }
    string getrid(){
        return RID;
    }
    void setcid(string Cid){
        CID = Cid;
    }
    string getcid(){
        return CID;
    }

void setName(string name){
    Name = name;
}

string getName(){
    return Name;
}
void setAddress(string address){
    Address = address;
}

string getAddress(){
    return Address;
}
void getEmail(string email){
    Email = email;
}

string setEmail(){
    return Email;
}
```

```
void settime(string TIME){
    Time =TIME;
}
string gettime(){
    return Time;
}
void setdestination(string DESTINATION){
    Destination = DESTINATION;
}
string address(){
    return Destination;
}
void setPhone(int phone){
    Phone = phone;
}

int getPhone(){
    return Phone;
}
void setnumberofseat(int NUMBEROFSEAT){
    NumberOfSeats = NUMBEROFSEAT;
}
```

```
    int getnumberofseat(){
        return NumberOfSeats;
    }
    void settravellingway(string TRAVEL){
        TravellingWay = TRAVEL;
    }
    string gettravellingway(){
        return TravellingWay;
    }
    void setCharges(int charges){
        Charges = charges;
    }

    int getCharges(){
        return Charges;
    }
private:
    string RID;
    string CID;
    string Name;
    int Phone;
```

```
    string Email;
    string Time;
    string Destination;
    int NumberOfSeats;
    string TravellingWay;
    int Charges;
};

class Manage_Account:public Customer{
};

class Registration :public Customer{

};

class Delete_Account :public Manage_Account{

};

class Update_BusSchedule : public Reservation{

};


class Update_UserCredential : Manager{

};

class Reports : public ManageBooking{

};

class ManageBooking : public Manage_Account{

};

class login : public Customer, Admin, Staff{

};

void Exit(){
    exit(0);
}

void main(){}
```

REFERENCES

AgileModelling, 2014. *AgileModelling.* [Online] Available at: http://agilemodeling.com/artifacts/componentDiagram.htm [Accessed 5 August 2016].

CatchThatBus, 2015. *CatchThatBus.* [Online] Available at: http://www.catchthatbus.com/?locale=en_us&utm_campaign=191348963&utm_medium=9079555163&utm_term=online%20ticket%20booking%20bus&utm_source=adwords&device=c&gclid=CjwKEAjwz4u9BRCbioK3stnBznESJADA75xbMYEzeQsDtlSIQz7-ycB_mh-KeByjYBVfH0Ucii8W4xoCXCTw_wcB [Accessed 5 August 2016].

Donald Bell, 2004. *IBM DEVELOPERWORKS.* [Online] Available at: http://www.ibm.com/developerworks/rational/library/dec04/bell/ [Accessed 31 July 2016].

GOMMA, H., 2011. *Software Modelling & Design.* 1st ed. Cambridge: Cambridge University Press.

MakeMyTrip, 2016. *MakeMyTrip.* [Online] Available at: https://www.makemytrip.com/bus-tickets/ [Accessed 5 August 2016].

Mircorosoft, 2016. *Microsoft Support.* [Online] Available at: https://support.microsoft.com/en-us/kb/837294 [Accessed 31 July 2016].

SHOHOZ.COM, 2014. *SHOHOZ.COM.* [Online] Available at: https://lh3.googleusercontent.com/M0QOHtEnUXsPWiWJQYZ4nncQe2w-iITGyF-HOQuE826uh0GJrwYtUzrUAcjFkWwnCw=h900 [Accessed 90 May 2016].

Sparx System, 2016. *Sparx System.* [Online] Available at: http://www.sparxsystems.com/resources/uml2_tutorial/uml2_componentdiagram.html [Accessed 31 July 2016].

TraceModeler, 2008. *TraceModeler.* [Online] Available at:

http://www.tracemodeler.com/articles/a_quick_introduction_to_uml_sequence_diagrams/ [Accessed 5 August 2016].

UML DIAGRAMS, 2014. *UML DIAGRAMS.org.* [Online] Available at: http://www.uml-diagrams.org/component-diagrams.html [Accessed 31 July 2016].