PLAESTRA FITNESS CENTER SYSTEM

GROUP MEMBERS
AMANDEEP KAUR AUJLA-300823928
VISHAVDEEP KAUR-300828926
SHAHBAZ SINGH-300817167

Contents

Fitness Center System

PROBLEM STATEMENT

PLALESTRA FITNESS CENTER is a fitness center which has manual system of maintaining records. Recently, the company has planned to expand its business by renting out gym equipment. Also, the company is experiencing huge customer traffic which makes it difficult to maintain records in excel sheets. The company is facing problems involving same people taking free trial a number of times or attending paid classes for free. Also, the company has introduced new packages and increased fee for new customers. Employees find it difficult to differentiate between new and old customers and are unable to charge customers appropriately.

The company also investing in advertisement to increase the number of customers and they require an information system to assist them in these decisions.

VISION STATEMENT

Upgrade the manual system of the fitness center to an automated computer system. Execute Information System to support and enhance day by day operation with expanded efficiency, while keeping a high level of Customer satisfaction.

SYSTEM CAPABILITIES

- Register customers and maintain records
- Manage payments
- Book trainers and sessions for members
- Manage schedule for members and trainers as well
- Maintain information about rented equipment
- Assist management in making decisions

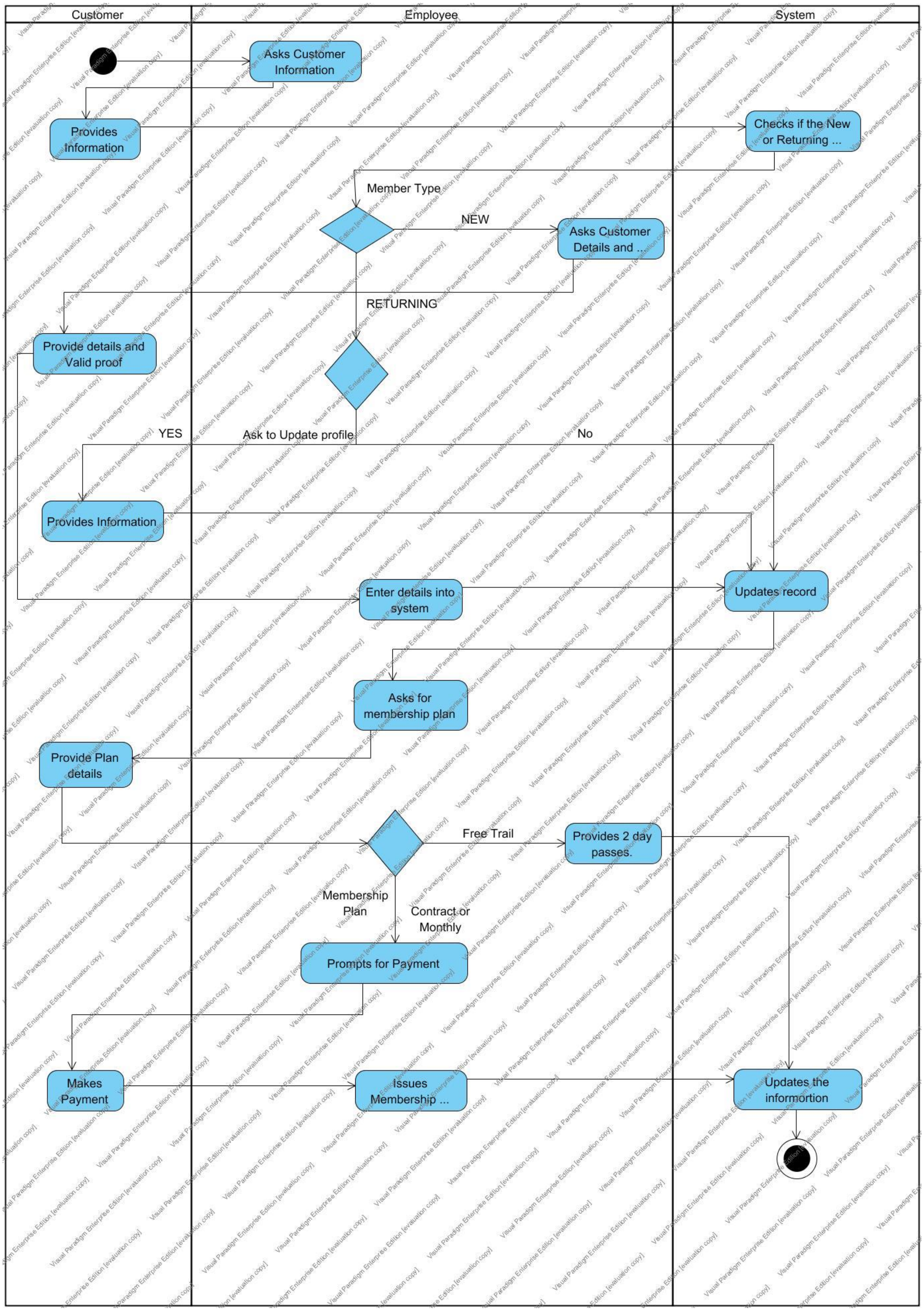
BUSINESS BENEFITS

- Increase in customer satisfaction due to fast processing
- Expansion of business by handling multiple trainers and sessions
- Lead to faster and efficient management decisions
- As opposed to the manual system the proposed information system will allow a more efficient management of the fitness center

WORK-FLOWS

1.0Member Registration

1.1 Employee asks for Customer Information
1.2 Customer provides information
1.3 Employee performs system check for previous record.
1.3.1.1 Employee asks details and valid proof to make a new record.
1.3.1.2 Customer provides details
1.3.1.3 Employee verifies the details.
1.3.1.4 System updates the record.
` 1.3.2 For returning customer, old profile is retrieved.
1.3.2.1 Employee ask to update info
1.3.2.2 Customer provides details
1.4 Employee asks for membership plan.
1.4.1 Customer makes payment if the membership plan is on Contract or month to month.
1.4.1.2 System updates the record
1.4.1.3 Employee activates the membership card
1.4.2 Employees provides 2 day passes if Customer wants free trial.


Customer
Employee
System
Asks Customer Information
Provides Information
Checks if the New or Returning ...
Member Type
NEW
Asks Customer Details and ...
RETURNING
Provide details and Valid proof
YES
Ask to Update profile
No
Provides Information
Enter details into system
Updates record
Asks for membership plan
Provide Plan details
Free Trail
Provides 2 day passes.
Membership Plan
Contract or Monthly
Prompts for Payment
Makes Payment
Issues Membership ...
Updates the informortion

2.0Member/Employee Check-In

2.1 Member/Employee approaches Check-in machine.

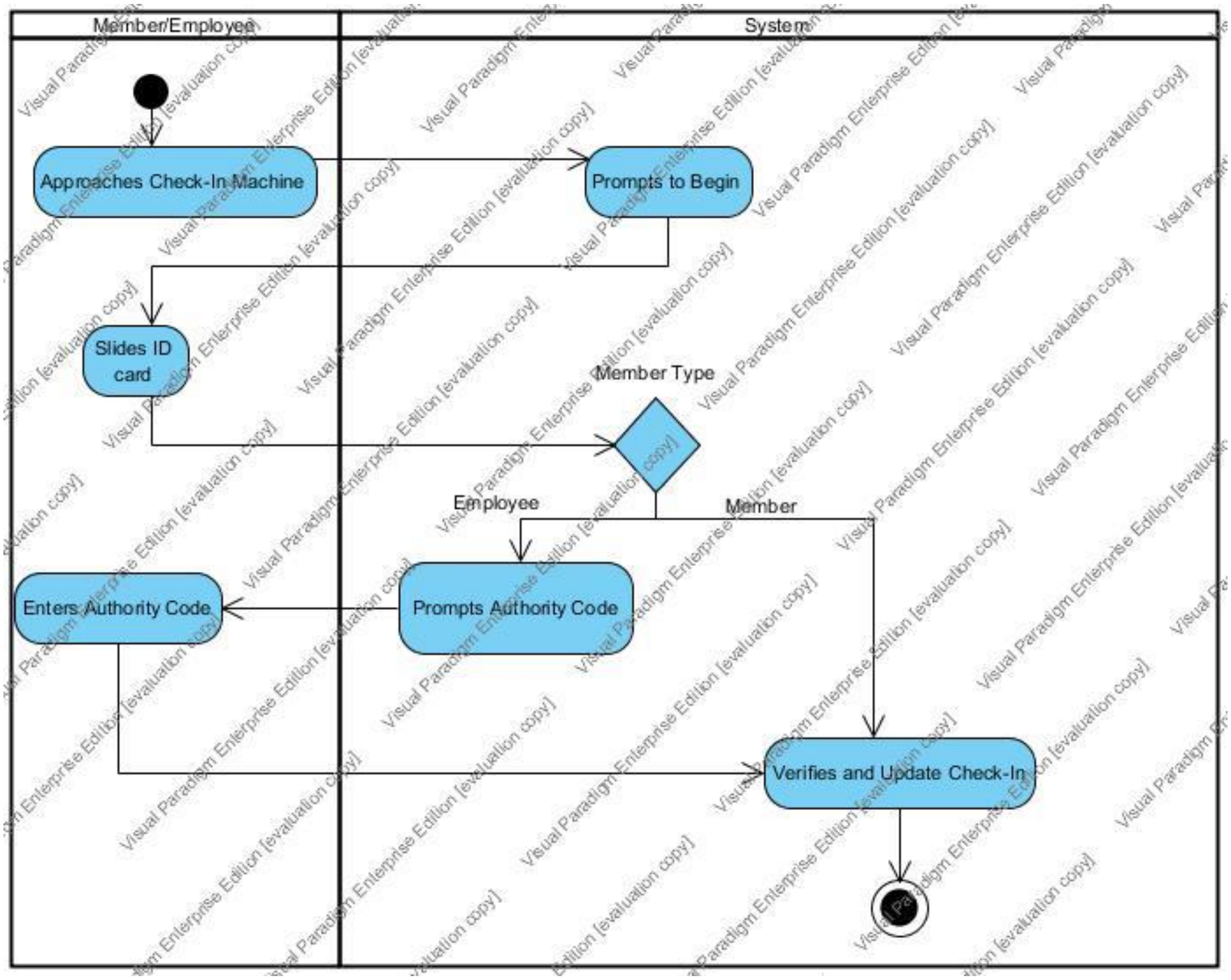
2.2 Slides the ID card.

2.3.1 For Employee, System prompts authority code.

2.3.1.1. Employees enters the authority code.

2.3.1.2 System verifies and update the check-in.

2.3.2 For Member, System update the check-in.



3.0Trainer Booking

3.1Customer asks for personal trainer.

3.2Employee asks the classification type and schedule.

3.1 Customer provides information.

3.2 Employee enters the details into the system

3.3 System automatically shortlist the available trainers.

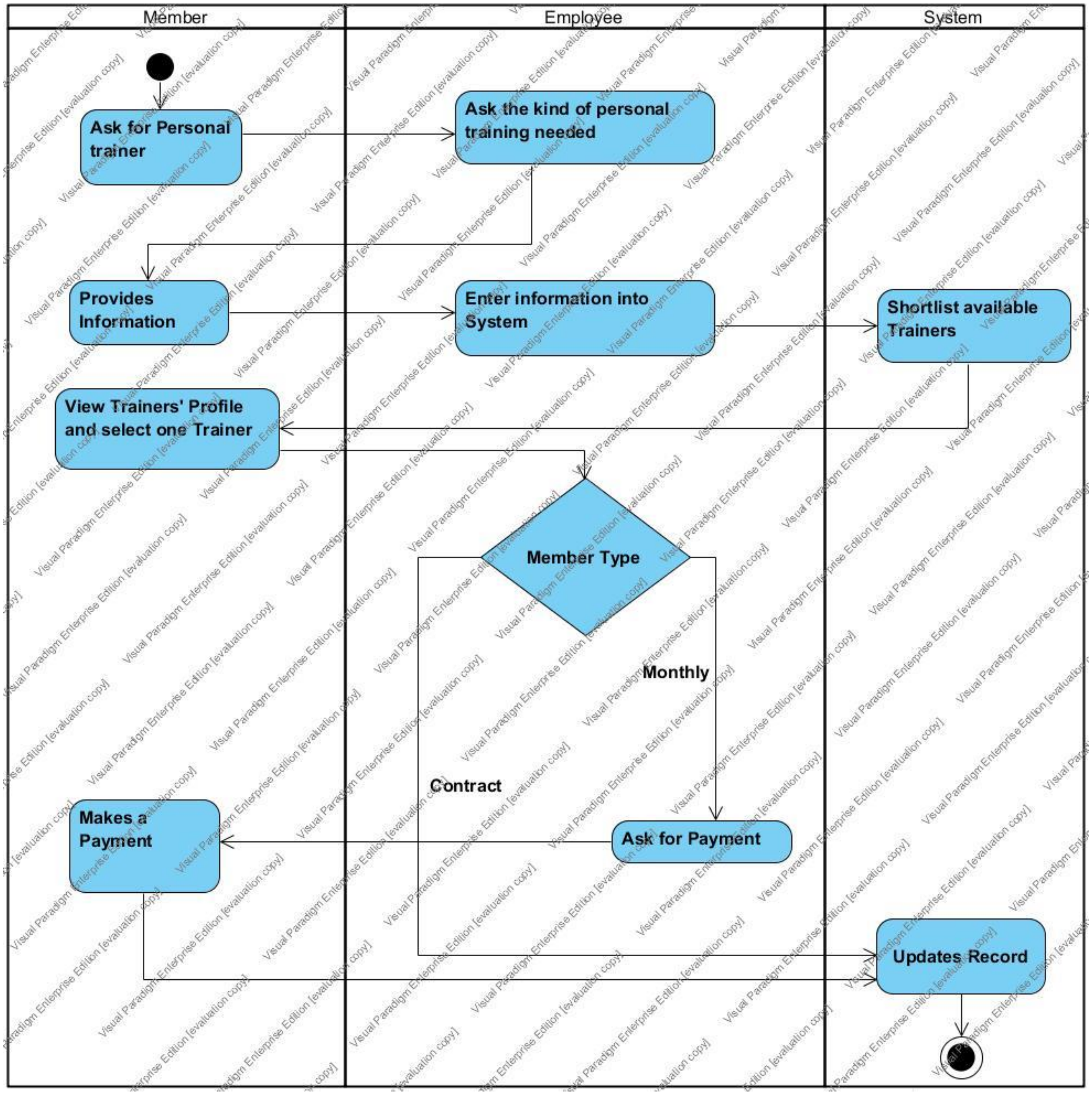
3.4 Customer selects a trainer.

3.7.1 Employee saves and updates the record into the system if the Customer is on Contract.

3.7.2 System generates total amount for the booking if the Customer on Monthly plan.

3.7.2.1 Customer makes payment.

3.7.2.2. System records the transaction.


Member
Employee
System
Ask for Personal trainer
Ask the kind of personal training needed
Provides Information
Enter information into System
Shortlist available Trainers
View Trainers' Profile and select one Trainer
Member Type
Monthly
Contract
Makes a Payment
Ask for Payment
Updates Record

4.0 Equipment Renting

4.1 Customer asks for an equipment(s).

4.2 Employee asks for Membership ID card.

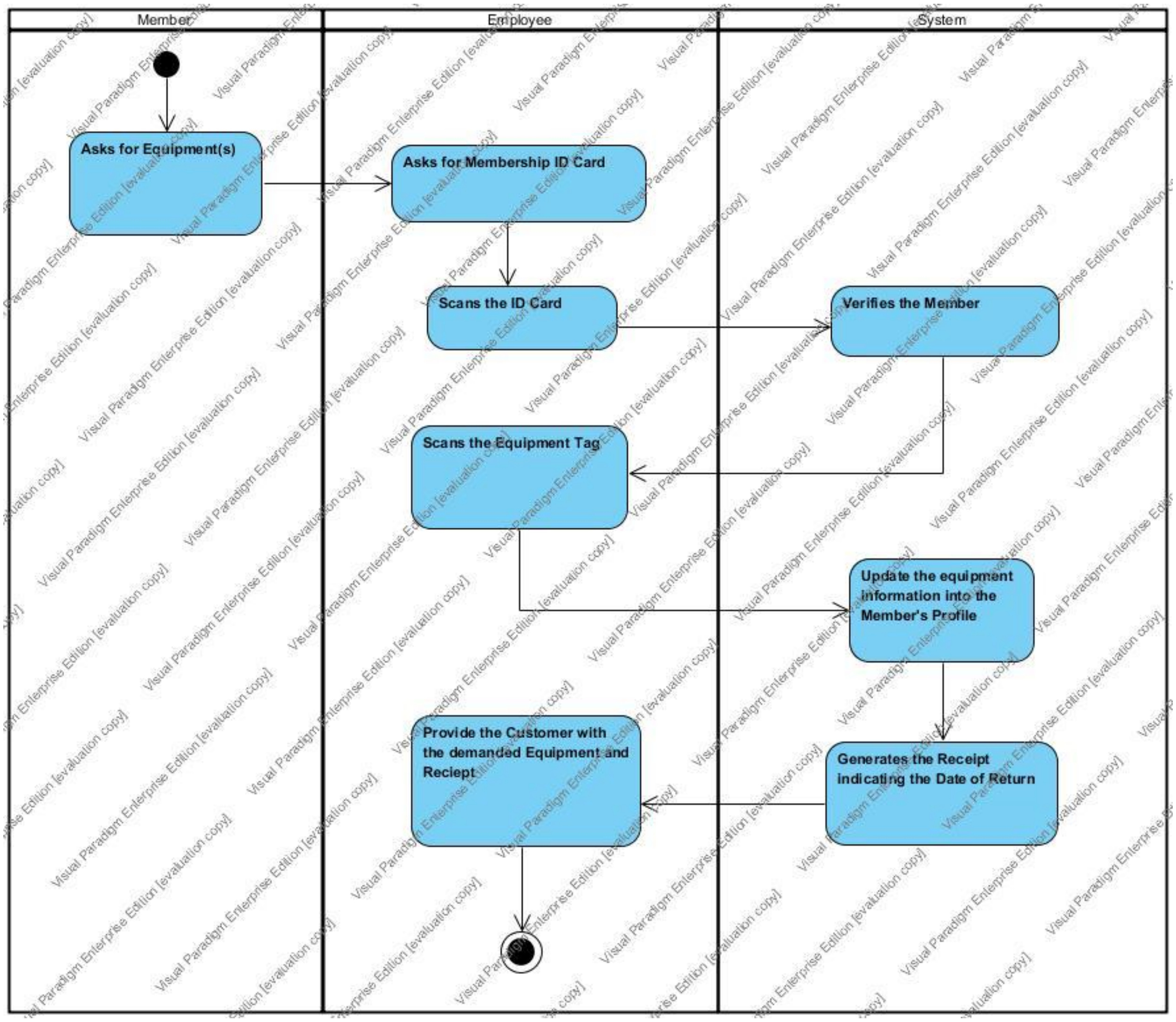
4.3 Employee scans the ID card.

4.4 Employee scans the equipment tag.

4.5 System update the equipment information into the Customers profile.

4.6 System generates the receipt indicating the due date.

4.7 Employee gives the equipment and receipt to the Customer.


Member
Employee
System
Asks for Equipment(s)
Asks for Membership ID Card
Scans the ID Card
Verifies the Member
Scans the Equipment Tag
Update the equipment information into the Member's Profile
Provide the Customer with the demanded Equipment and Reciept
Generates the Receipt indicating the Date of Return

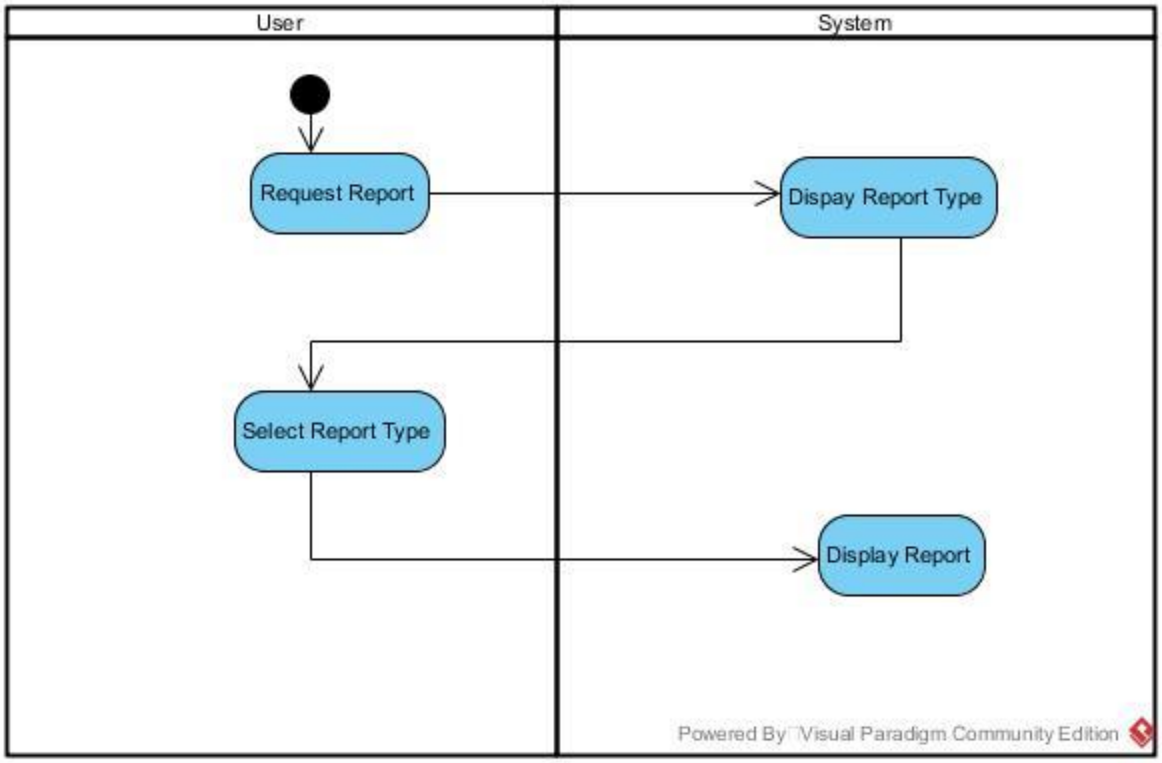
5.0 Patient and office reports

5.1 User (Employee or Manager) requests report.

5.2 System displays the different types of reports. (Monthly Member Reports, Monthly Staff Reports, Monthly Booking Reports

5.3 User selects report type.

5.4 System displays the report.



USE CASE

PROCESSES (Workflows):
1) Member Registration
2) Equipment Renting
3) Trainer booking

4) Member/Employee Check-In
5) Office Reports

The Processes correspond to the Functional areas of the Use Case Model shown below:

No.	Actor	Goal Use Case	<Include>	<Extend>	Generalization	Priority	Brief Description
1.1	Employee	Register as a member	• Get Payment • Activate Membership Card	• Book trainer • Book Session	• Member pays monthly • Member is on a contract	H	Employee registers customer as a new member
1.2	Employee	Register as a customer	• Verify customer info • Print two day passes			H	Employee registers a new customers and provides passes for free trial
1.3	Employee	Search member /customer details		• Add as a member • Search all customer/member data		M	Employee can search member details of a single or all members.
1.4	Employee	Maintain Payment Details		Print Payment Details		L	Employee can view or alter payment details and print them
1.5	Employee	Update member/ customer details				M	Employee can modify customer or member details as required

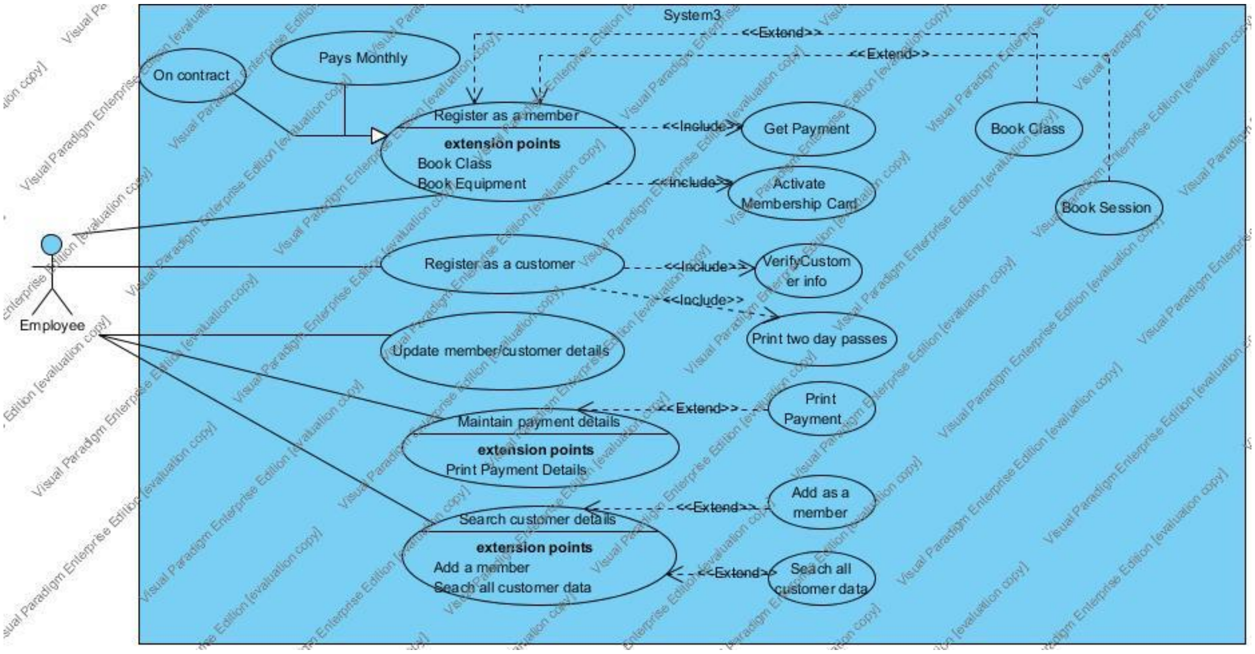
2.1	Employee Member	Check-In.	Scan membership/staff card			M	Members or employee can check-in and mark their attendance
2.2	Employee Staff	View Member Attendance Details				M	Employee or Manager can view attendance details of members at fitness center
2.3	Manager	View Staff Attendance				M	Manager can view attendance details of staff
3.1	Employee	Book Trainer	• Confirm Membership type • Check member schedule • Check trainer schedule	• Get Payment		H	Employee books trainer for members by considering schedules of both trainer and member. Also if the member pays monthly and is not on contract then employee gets payment from member
3.3	Employee	Update training schedule	•Check member schedule •Check trainer schedule • Get current training schedule				Employee makes changes on training schedule depending on member or trainer request

3.4	Employee	View members under a trainee				H	Employee can view members under a trainee when asked by a member or a trainee
3.5	Employee	View Trainer Details				L	Employee can view trainer details when requested by a member
3.6	Employee	Cancel Booking				L	Employee cancels training as requested by trainer or employee
4.1	Employee	Rent Equipment	• Scan Equipment • Confirm Membership • Update on member account	Generate receipt		H	Employee rents equipment by scanning the equipment and confirm the membership status of member. Employee then updates the rented equipment on member’s account.

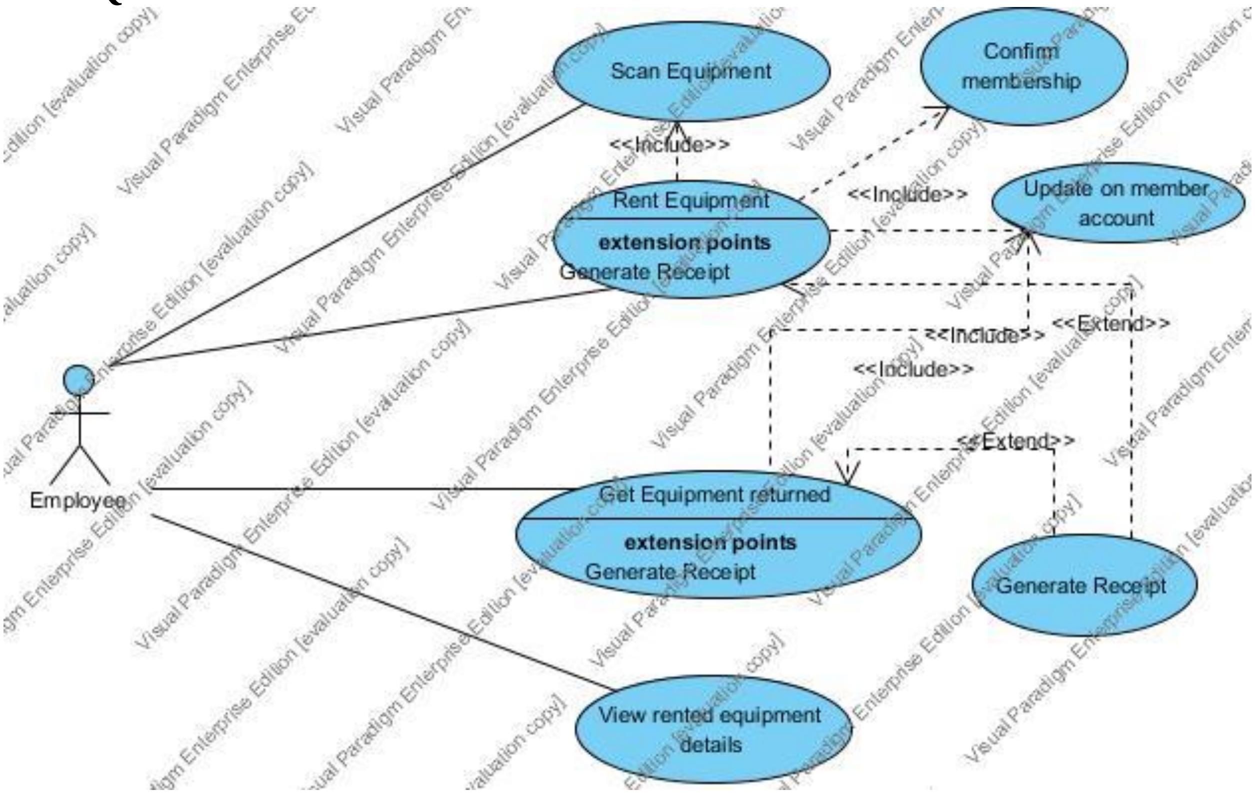
4.2	Employee	Get Equipment Returned	• Update on member account	Generate receipt		M	Employee scans the equipment and takes off it from the member account
4.3	Employee	View Rented Equipment Details				H	View Rented Equipment Details: Employee can view details of rented equipment.
5.1	Employee Management	View Monthly Booking Reports				H	Employee and members can view Monthly Booking Reports
5.2	Employee Management	View Monthly Member Reports				H	Employee and members can view Monthly Member Reports
5.3	Management	View Monthly Staff Reports				H	Management can view monthly staff reports.

USE CASE DIAGRAM

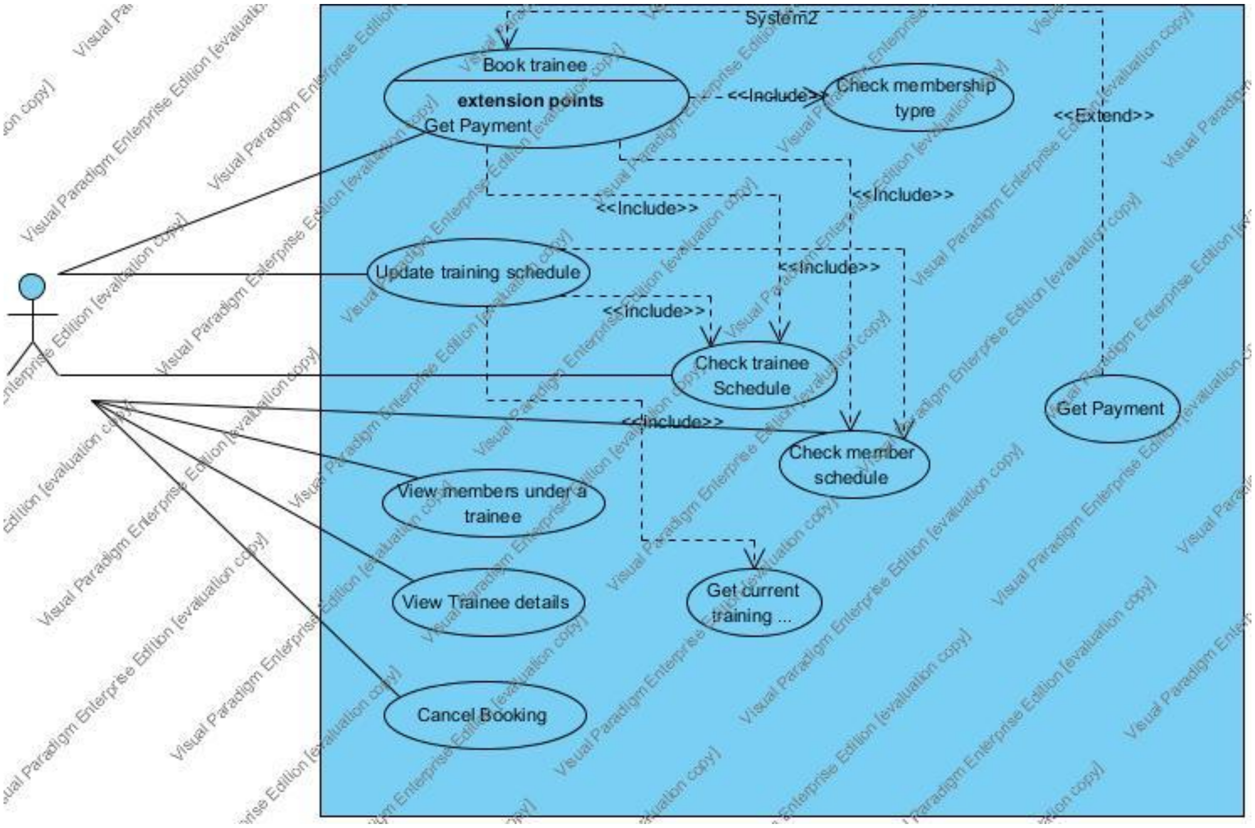
1.1 MEMBER REGISTRATION



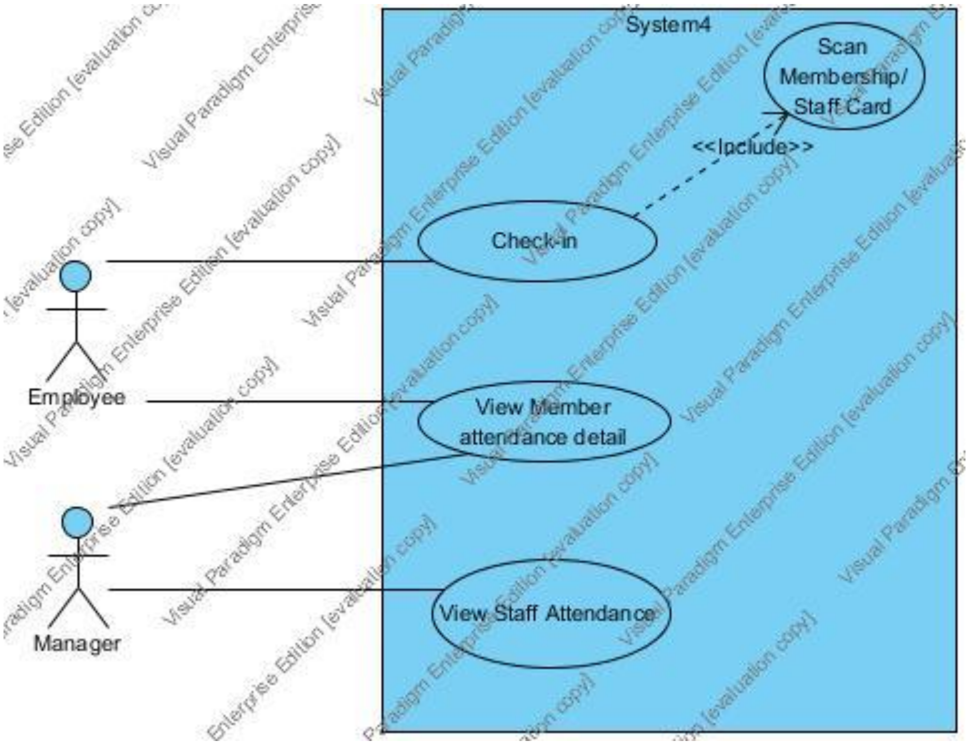
1.2–EQUIPMENT RENTING



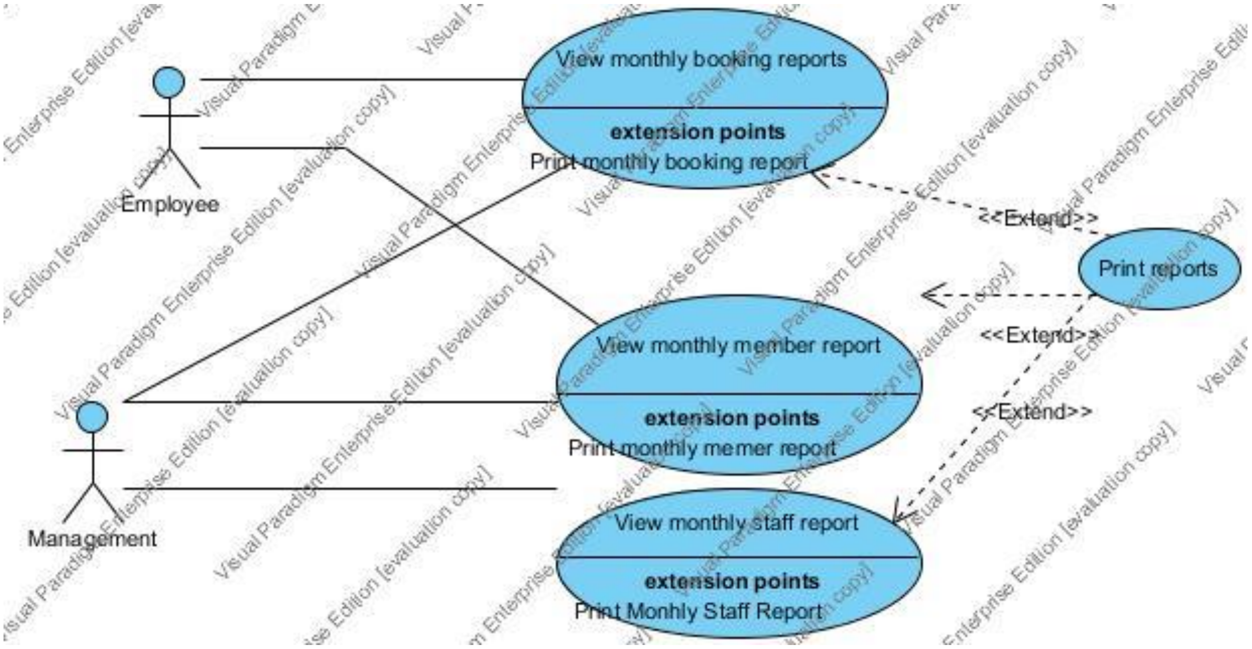
1.3–TRAINER BOOKING



1.4-MEMBER/EMPLOYEE CHECK-IN



1.5-OFFICE REPORT


View monthly booking reports
extension points
Print monthly booking report
Employee
<<Extend>>
Print reports
View monthly member report
<<Extend>>
<<Extend>>
extension points
Print monthly memer report
Management
View monthly staff report
extension points
Print Monhly Staff Report

USE CASE DESCRIPTION

PACKAGE 1-MEMBER REGISTRATION

REGISTER A MEMBER
1.1
Customer can be registered as a member if he pays for the service. Member can pay monthly or pay the whole fee at once. If they pay the whole fee at once would be then treated as on contract and they can book trainer or any other session for free. When member is registered his membership card is activated. While registration, member can also book a session or a trainee.

REGISTER A CUSTOMER
1.2
Employee registers a new member as a customer after verifying that he was not registered before or hasn’t taken free trials before. Employee then prints two day free passes for the customer.

UPDATE MEMBER/CUSTOMER DETAILS
1.3
Employee can view payment details of members and based on these details he can send emails regarding fee payment. He can also print out the details.

UPDATE MEMBER/CUSTOMER DETAILS
1.3

Employee can view payment details of members and based on these details he can send emails regarding fee payment. He can also print out the details.

SEARCH CUSTOMER DETAILS
1.4
Employee can search customer data. He can view a particular customer data and change his status to member from a customer or he can search all customer data.

PACKAGE 2-EQIUPMENT RENTING

Rent Equipment
2.1
Employee can rent equipment by scanning the equipment and confirming that the customer is a member. Employee then updates the rent equipment info on member s account. Employee can also print ARECEIPT to be given to the member.

Get Equipment returned
2.2
Employee scans the equipment and takes off it from the member account

View Rented Equipment Details
2.3
Employee can view details of rented equipment.

PACKAGE 3- TRAINER BOOKING

Book Trainee
3.1
Employee books trainer for members by considering schedules of both trainer and member. Also if the member pays monthly and is not on contract then employee gets payment from member

View Member Under Training
3.2
Employee can view members under a trainee when asked by a member or a trainee

View Tainee Details
3.3

Employee can view trainer details when requested by a member

Update Training Schedule
3.4
Employee makes changes on training schedule depending on member or trainer request

Cancel Booking
3.5
Employee cancels training as requested by trainer or employee

PACKAGE 4- EMPLOYEE/MEMBER CHECK-IN

Check –IN
4.1
Employee or member can mark their attendance by scanning their membership or staff cards respectively. If the person is Employee, the System will prompt for authority code to enable the process. If the person is Customer, System scans the card and update the record. The Check-In tracks the record of number of people utilizing the Center facilities.

View Staff Details:
4.2
Manager can view staff details as well as attendance details

View Member details:
4.3
Manager or employee can view member details as well as attendance details

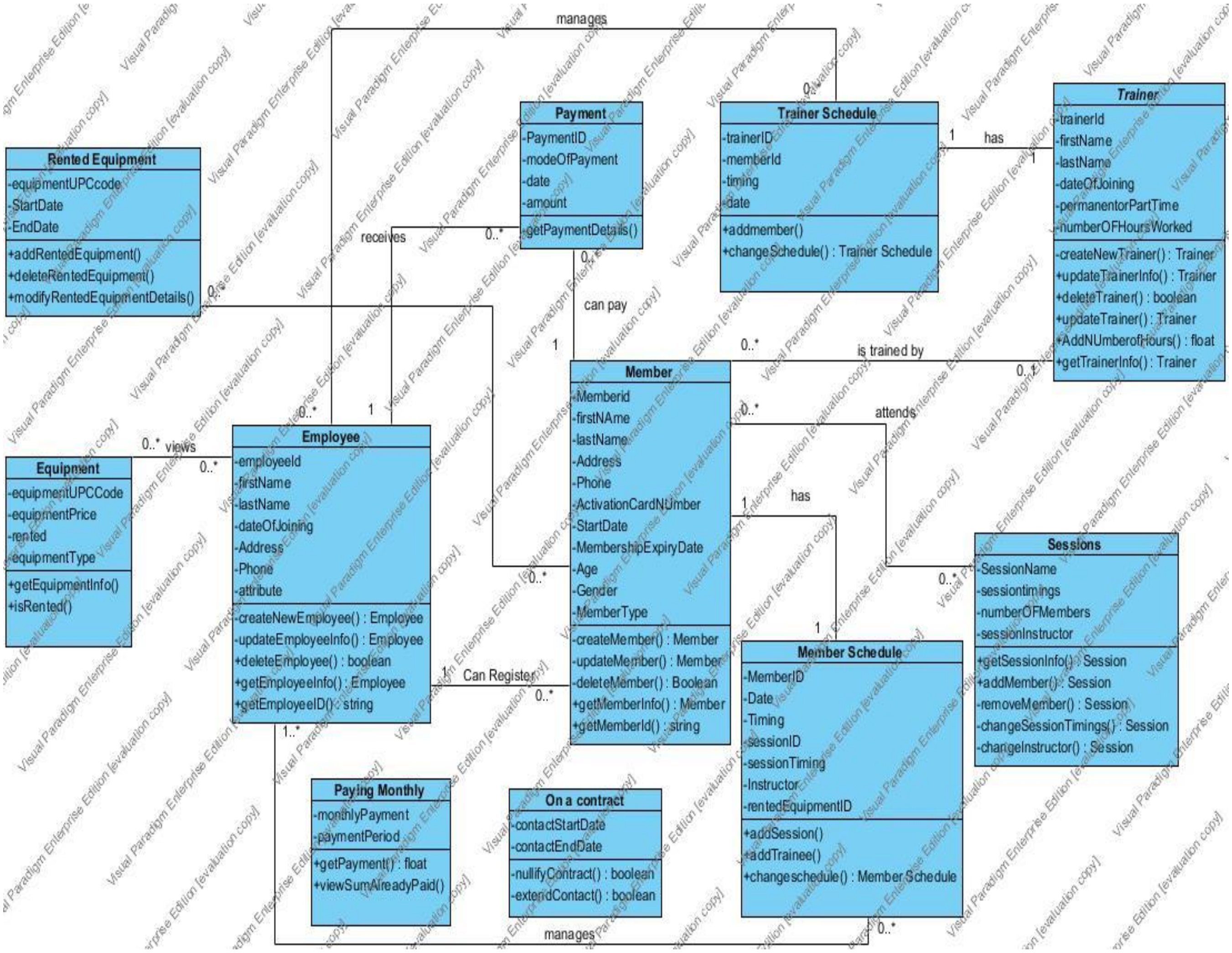
PACKAGE 5- OFFICE REPORT

View Monthly Booking Reports
5.1
Employee and members can view Monthly Booking Reports. Record can be printed.

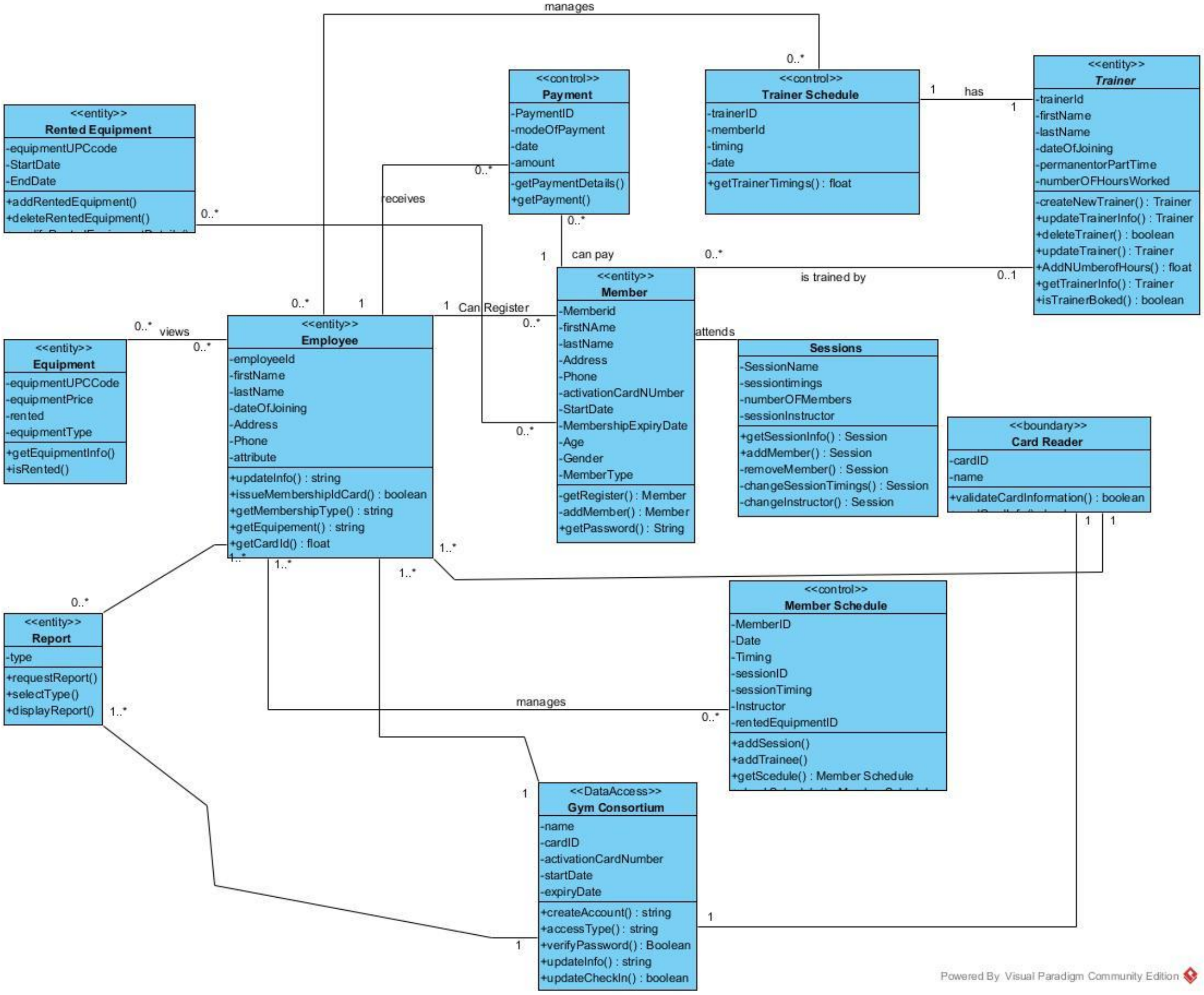
View Monthly Member Reports
5.2
Employee and members can view Monthly Member Reports. Record can be printed.

View Monthly Staff Reports
5.3
Management can view monthly staff reports. Record can be printed.

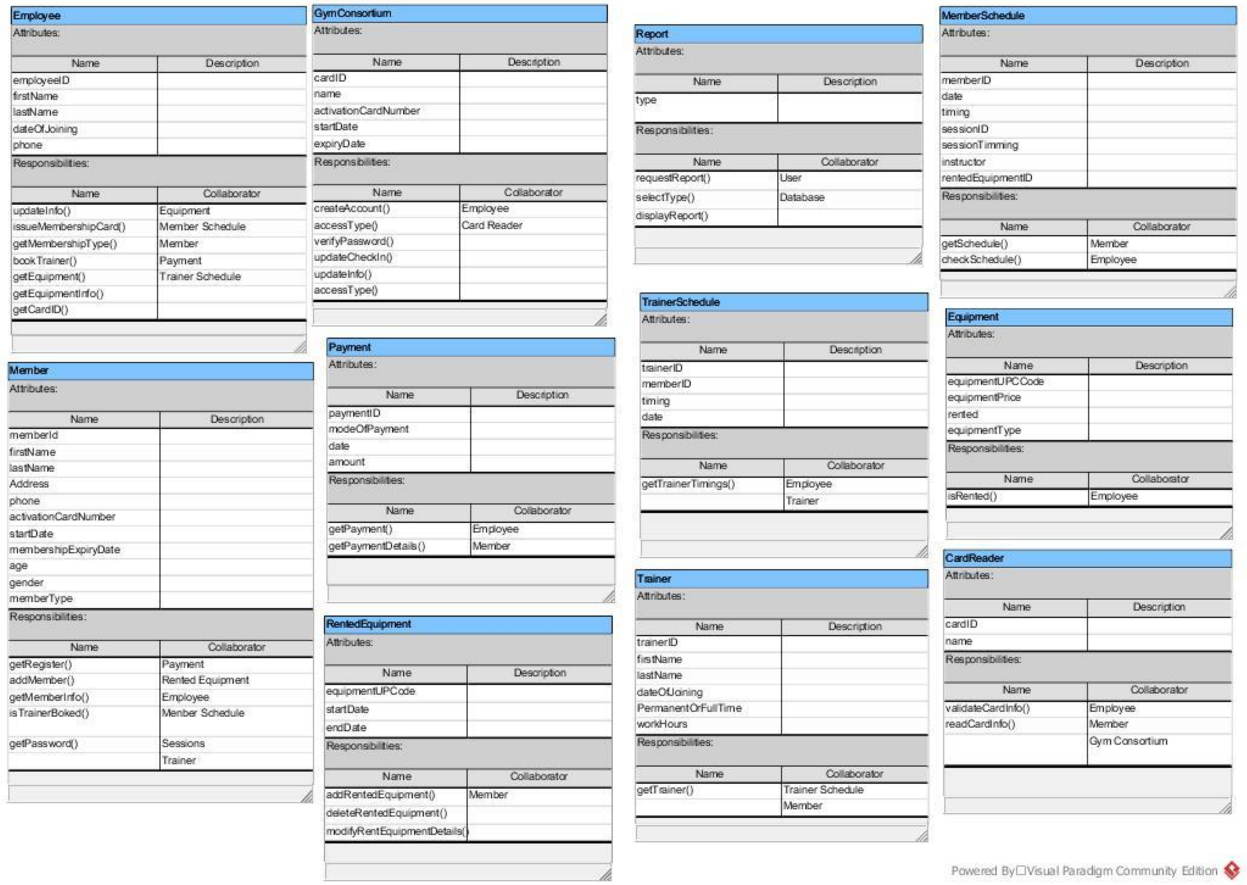
DOMAIN CLASS DIAGRAM



DESIGN DETAIL CLASS DIAGRAM



CRC CARDS



Employee

Attributes:

Name	Description
employeeID	
firstName	
lastName	
dateOfJoining	
phone	

Responsibilities:

Name	Collaborator
updateInfo()	Equipment
issueMembershipCard()	Member Schedule
getMembershipType()	Member
bookTrainer()	Payment
getEquipment()	Trainer Schedule
getEquipmentInfo()	
getCardID()	

Member

Attributes:

Name	Description
memberId	
firstName	
lastName	
Address	
phone	
activationCardNumber	
startDate	
membershipExpiryDate	
age	
gender	
memberType	

Responsibilities:

Name	Collaborator
getRegister()	Payment
addMember()	Rented Equipment
getMemberInfo()	Employee
isTrainerBoked()	Menber Schedule
getPassword()	Sessions
	Trainer

GymConsortium

Attributes:

Name	Description
cardID	
name	
activationCardNumber	
startDate	
expiryDate	

Responsibilities:

Name	Collaborator
createAccount()	Employee
accessType()	Card Reader
verifyPassword()	
updateCheckIn()	
updateInfo()	
accessType()	

Payment

Attributes:

Name	Description
paymentID	
modeOfPayment	
date	
amount	

Responsibilities:

Name	Collaborator
getPayment()	Employee
getPaymentDetails()	Member

RentedEquipment

Attributes:

Name	Description
equipmentUPCode	
startDate	
endDate	

Responsibilities:

Name	Collaborator
addRentedEquipment()	Member
deleteRentedEquipment()	
modifyRentEquipmentDetails()	

Report

Attributes:

Name	Description
type	

Responsibilities:

Name	Collaborator
requestReport()	User
selectType()	Database
displayReport()	

TrainerSchedule

Attributes:

Name	Description
trainerID	
memberID	
timing	
date	

Responsibilities:

Name	Collaborator
getTrainerTimings()	Employee
	Trainer

Trainer

Attributes:

Name	Description
trainerID	
firstName	
lastName	
dateOfJoining	
PermanentOrFullTime	
workHours	

Responsibilities:

Name	Collaborator
getTrainer()	Trainer Schedule
	Member

MemberSchedule

Attributes:

Name	Description
memberID	
date	
timing	
sessionID	
sessionTimming	
instructor	
rentedEquipmentID	

Responsibilities:

Name	Collaborator
getSchedule()	Member
checkSchedule()	Employee

Equipment

Attributes:

Name	Description
equipmentUPCCode	
equipmentPrice	
rented	
equipmentType	

Responsibilities:

Name	Collaborator
isRented()	Employee

CardReader

Attributes:

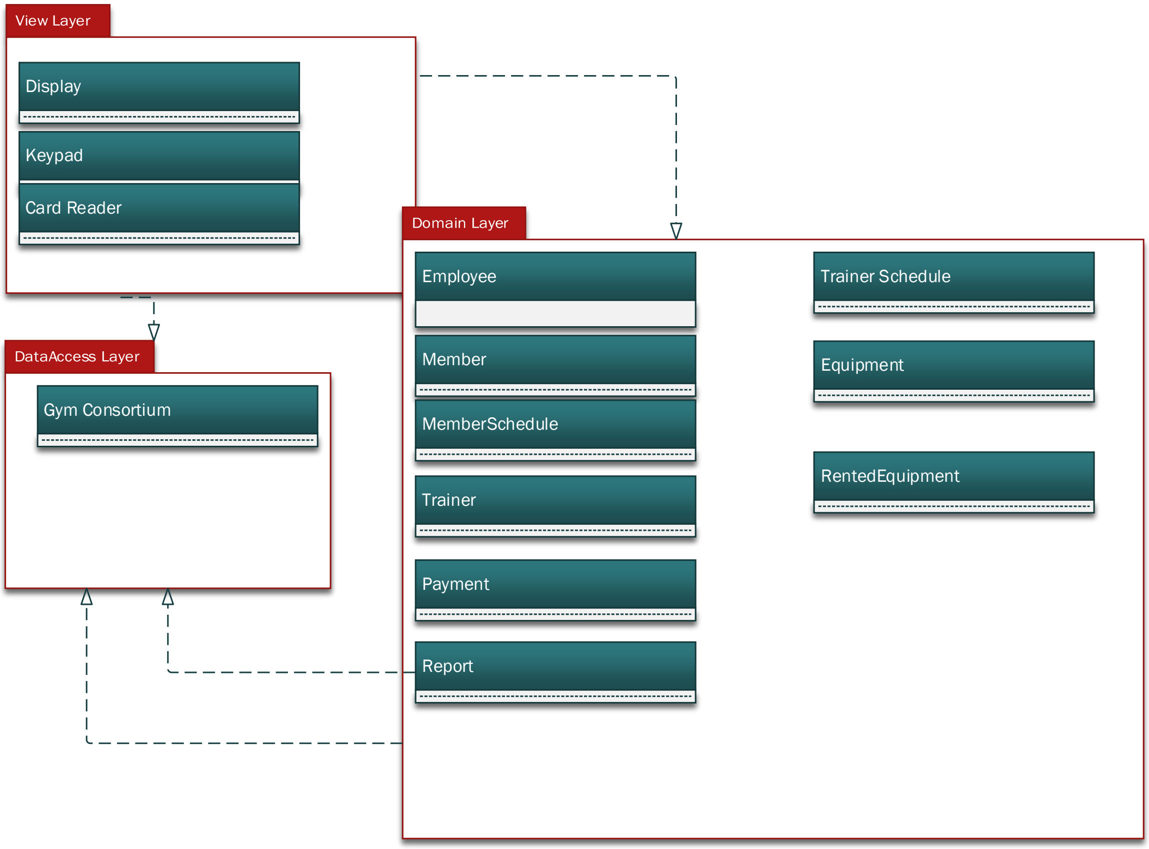
Name	Description
cardID	
name	

Responsibilities:

Name	Collaborator
validateCardInfo()	Employee
readCardInfo()	Member
	Gym Consortium

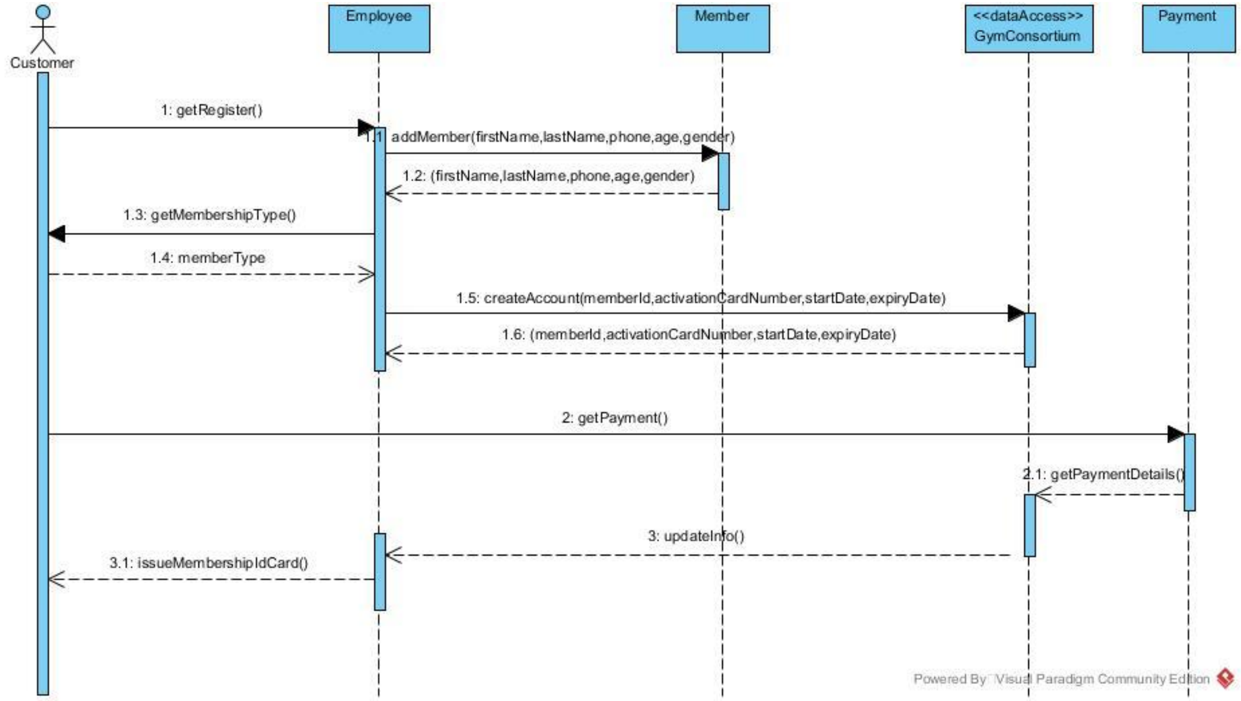
Powered By□Visual Paradigm Community Edition

PACKAGE DIAGRAM

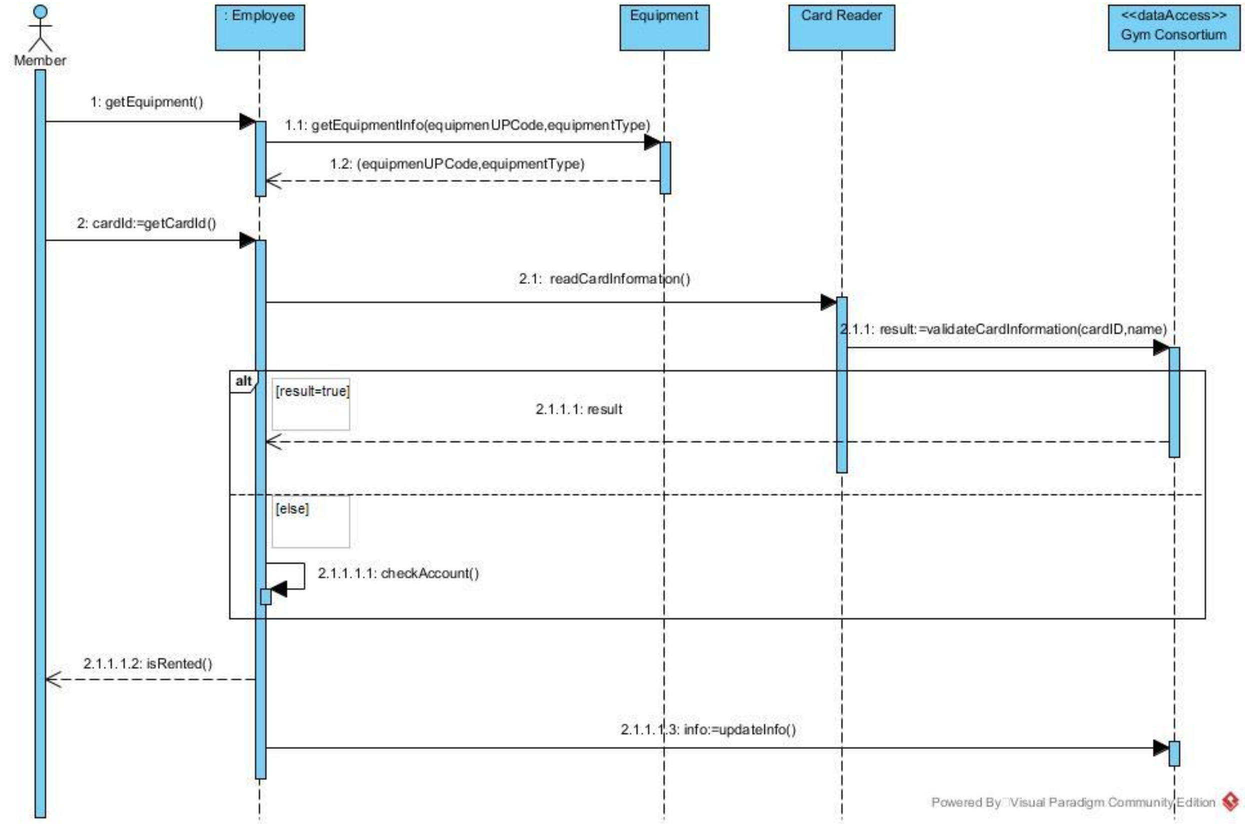


SEQUENCE DIAGRAMS

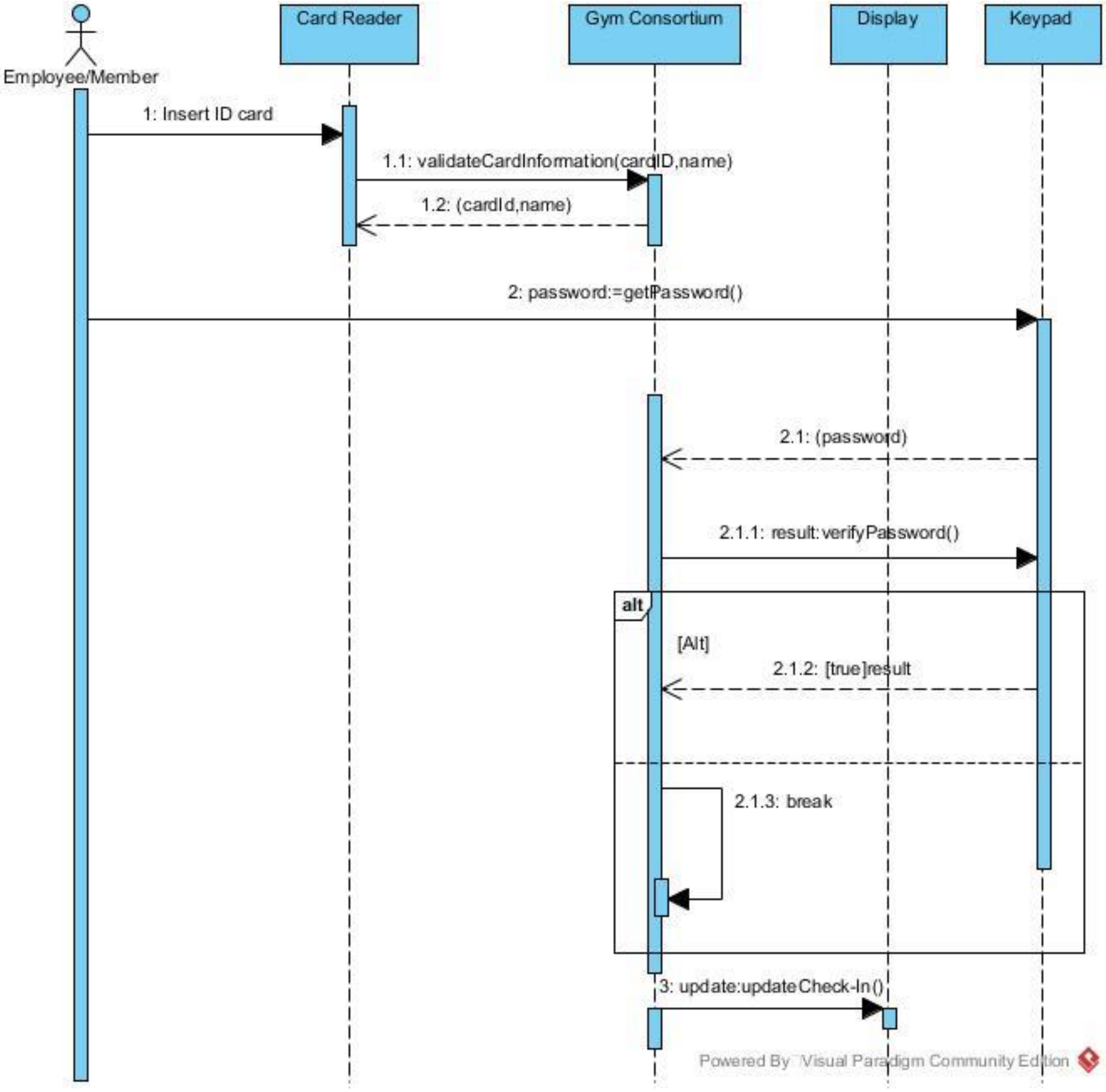
1. Member Registration


Customer
Employee
Member
<<dataAccess>>
GymConsortium
Payment
1: getRegister()
1.1: addMember(firstName,lastName,phone,age,gender)
1.2: (firstName,lastName,phone,age,gender)
1.3: getMembershipType()
1.4: memberType
1.5: createAccount(memberId,activationCardNumber,startDate,expiryDate)
1.6: (memberId,activationCardNumber,startDate,expiryDate)
2: getPayment()
2.1: getPaymentDetails()
3: updateInfo()
3.1: issueMembershipIdCard()
Powered By Visual Paradigm Community Edition

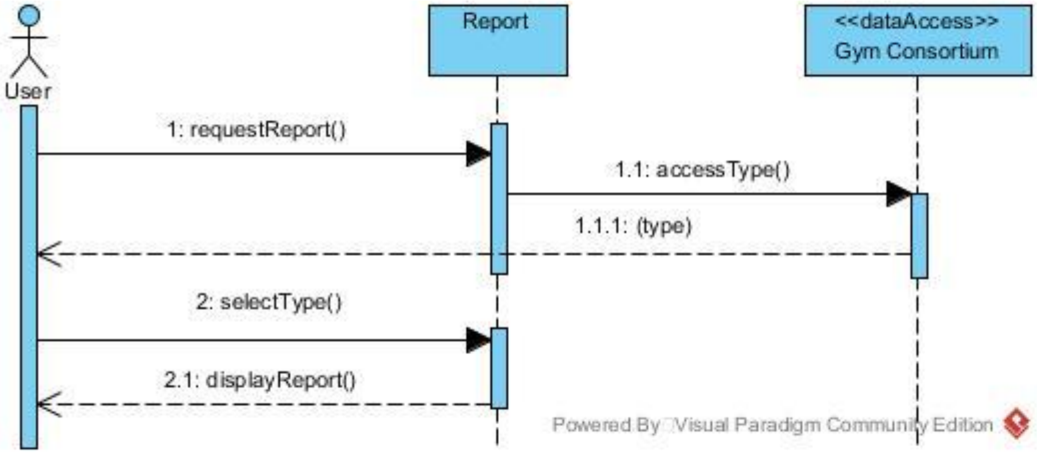
2. Rent Equipment


Member
: Employee
Equipment
Card Reader
<<dataAccess>>
Gym Consortium
1: getEquipment()
1.1: getEquipmentInfo(equipmenUPCode,equipmentType)
1.2: (equipmenUPCode,equipmentType)
2: cardId:=getCardId()
2.1: readCardInformation()
2.1.1: result:=validateCardInformation(cardID,name)
alt
[result=true]
2.1.1.1: result
[else]
2.1.1.1.1: checkAccount()
2.1.1.1.2: isRented()
2.1.1.1.3: info:=updateInfo()
Powered By Visual Paradigm Community Edition

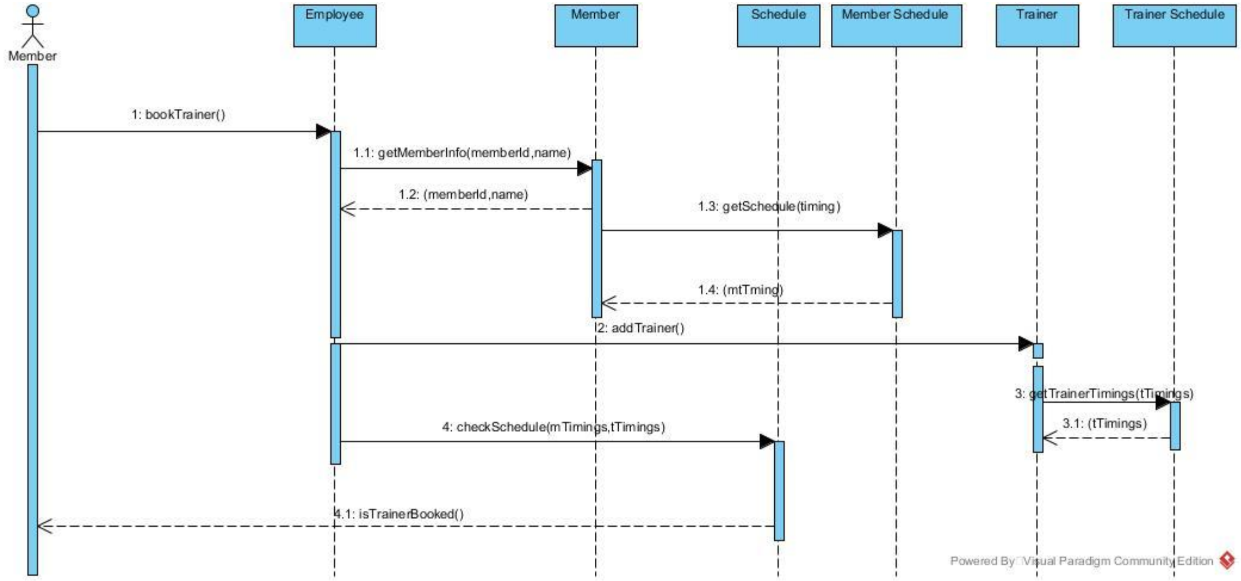
3. Member/Employee Check-In


Employee/Member
Card Reader
Gym Consortium
Display
Keypad
1: Insert ID card
1.1: validateCardInformation(cardID,name)
1.2: (cardId,name)
2: password:=getPassword()
2.1: (password)
2.1.1: result:verifyPassword()
alt
[Alt]
2.1.2: [true]result
2.1.3: break
3: update:updateCheck-In()
Powered By Visual Paradigm Community Edition

4. Office Report

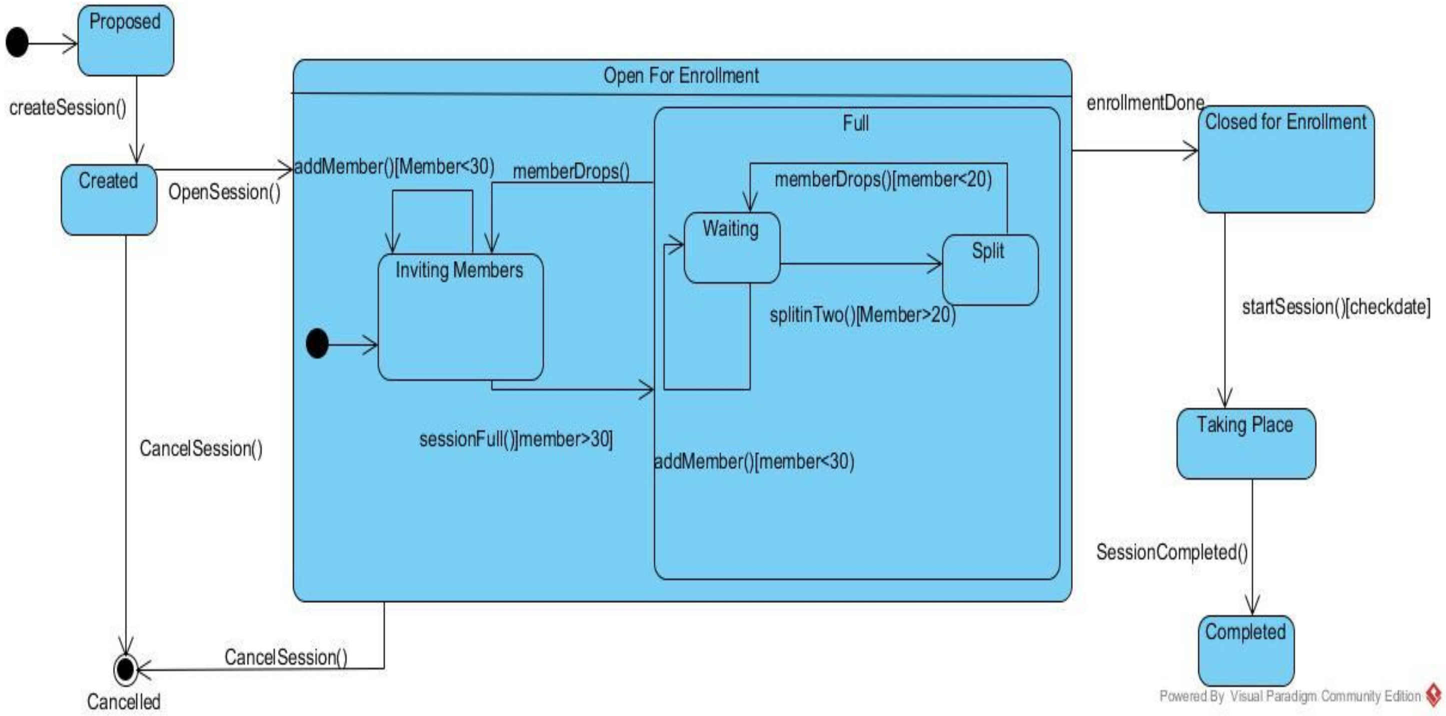


5. Book Trainer

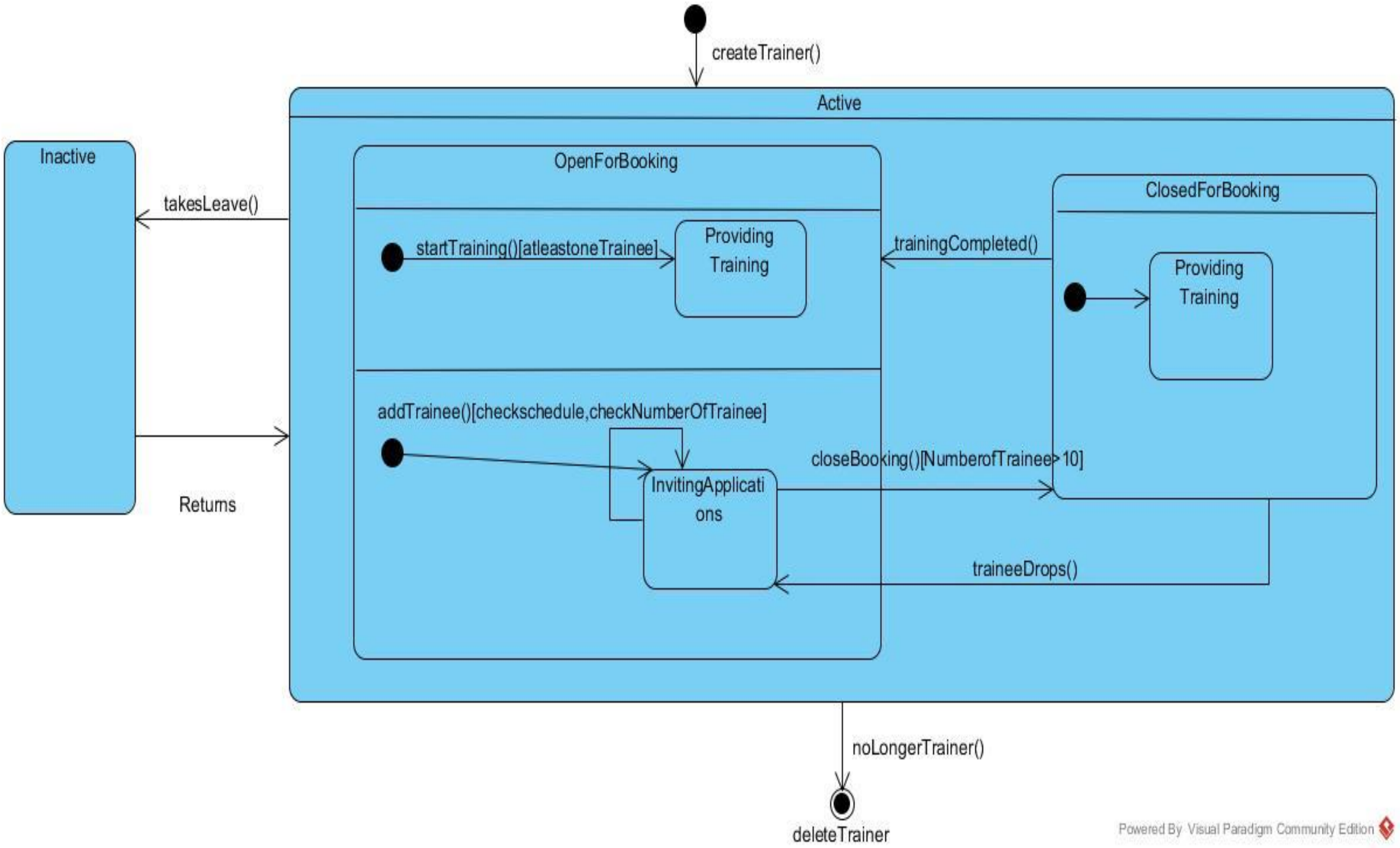


STATE DIAGRAM

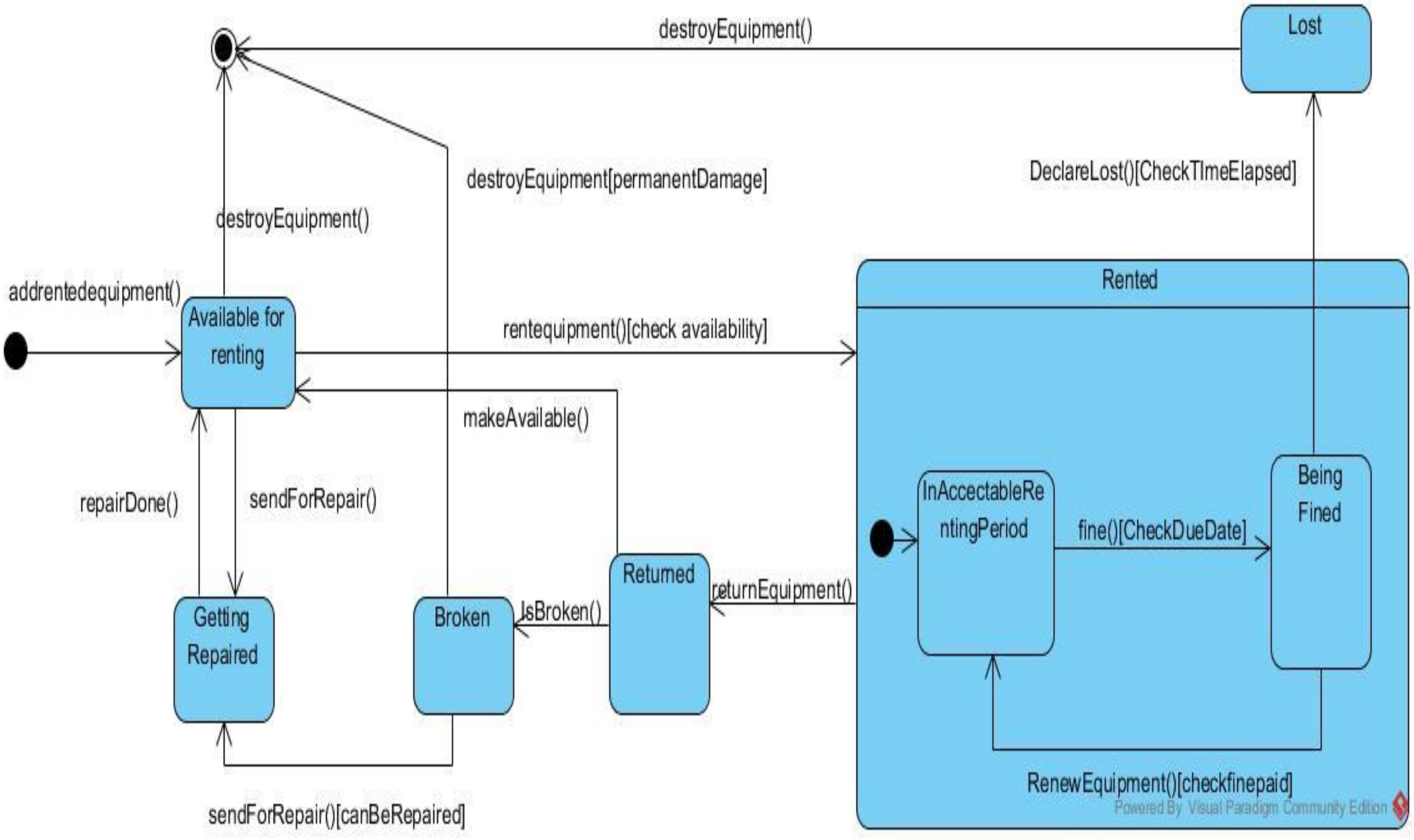
Sessions



Trainer (Trainer can provide training even when he is open for booking or close for booking)

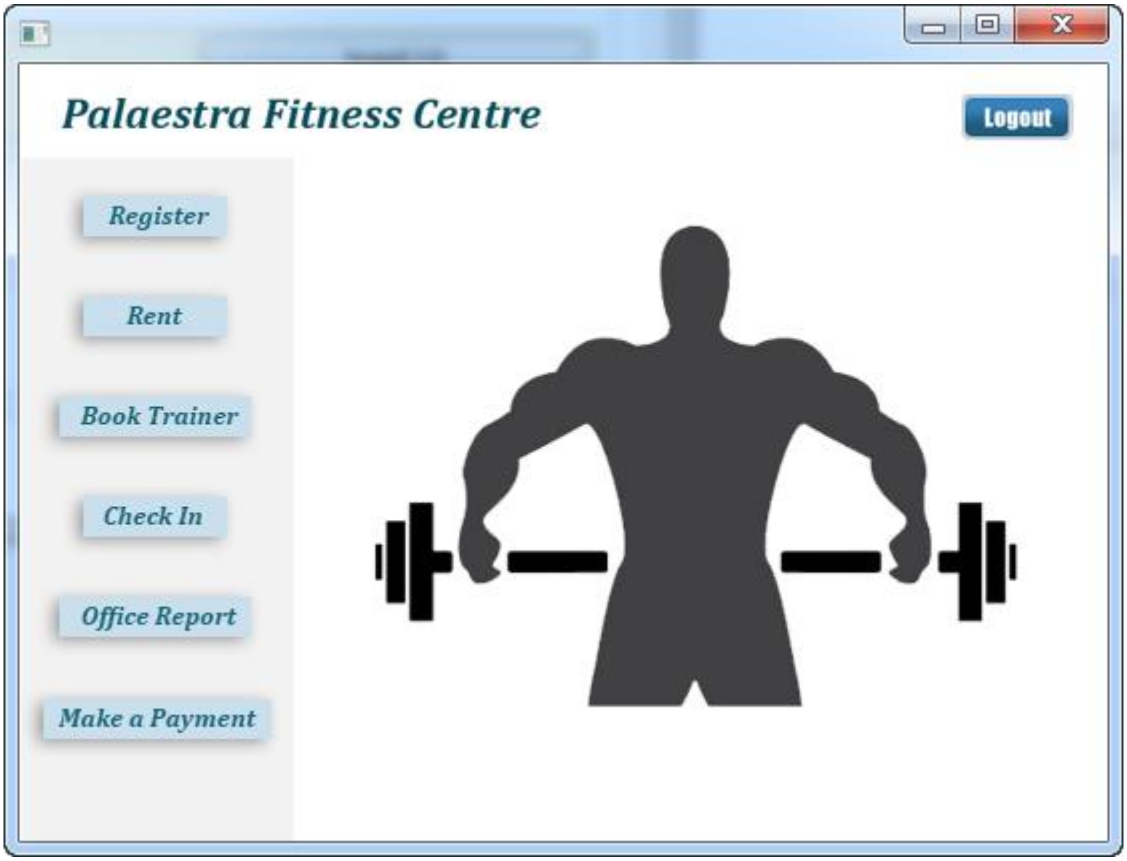


Equipment


destroyEquipment()
Lost
destroyEquipment[permanentDamage]
DeclareLost()[CheckTImeElapsed]
destroyEquipment()
addrentedequipment()
Available for renting
rentequipment()[check availability]
Rented
makeAvailable()
repairDone()
sendForRepair()
InAccectableRentingPeriod
fine()[CheckDueDate]
Being Fined
Returned
returnEquipment()
Getting Repaired
Broken
IsBroken()
RenewEquipment()[checkfinepaid]
Powered By Visual Paradigm Community Edition
sendForRepair()[canBeRepaired]

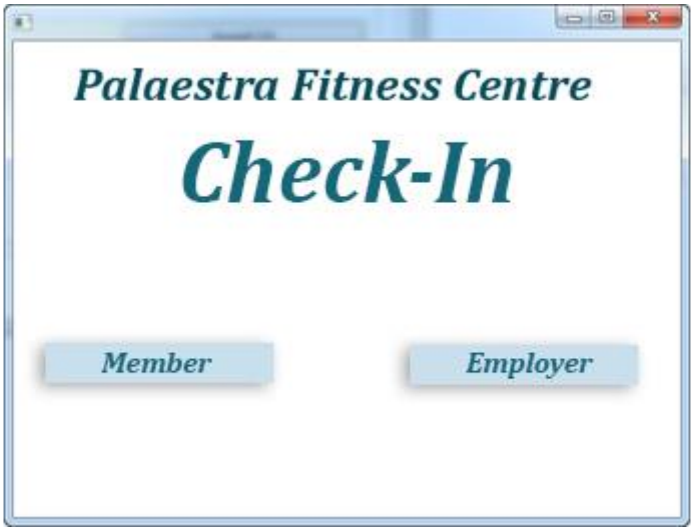
MOCK-UP USER INTERFACES

MAIN-MENU

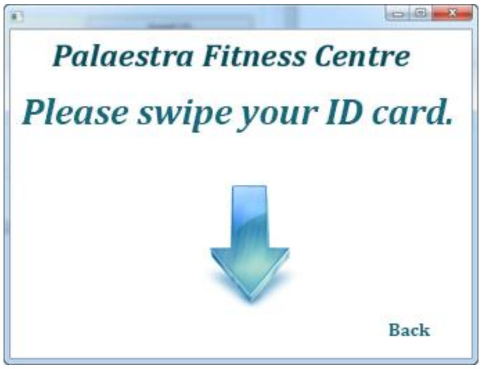


MEMBER-CHECK-IN

1. Check-In dialog box. User will click as according to its identity.



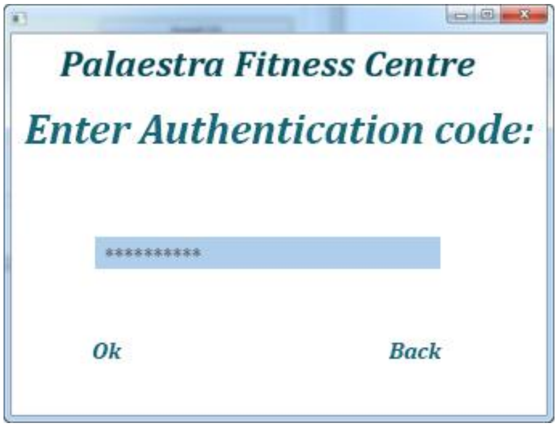
2. User will swipe ID card.



Palaestra Fitness Centre

Please swipe your ID card.

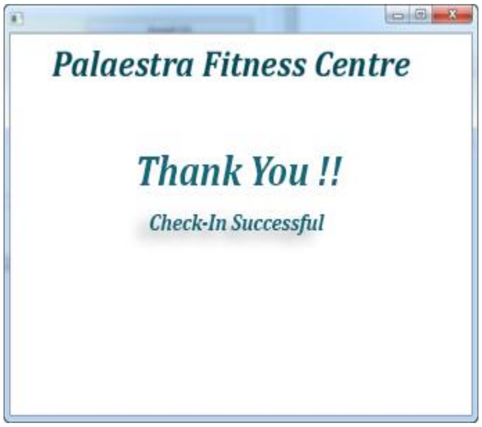
3.1 If user is Employee, he has to enter authentication code.



Palaestra Fitness Centre

Enter Authentication code:

****

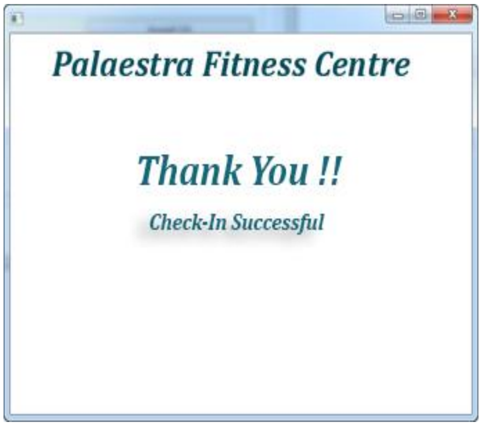


Palaestra Fitness Centre

Thank You !!

Check-In Successful

3.2 If user is Member, then directly record will be updated.

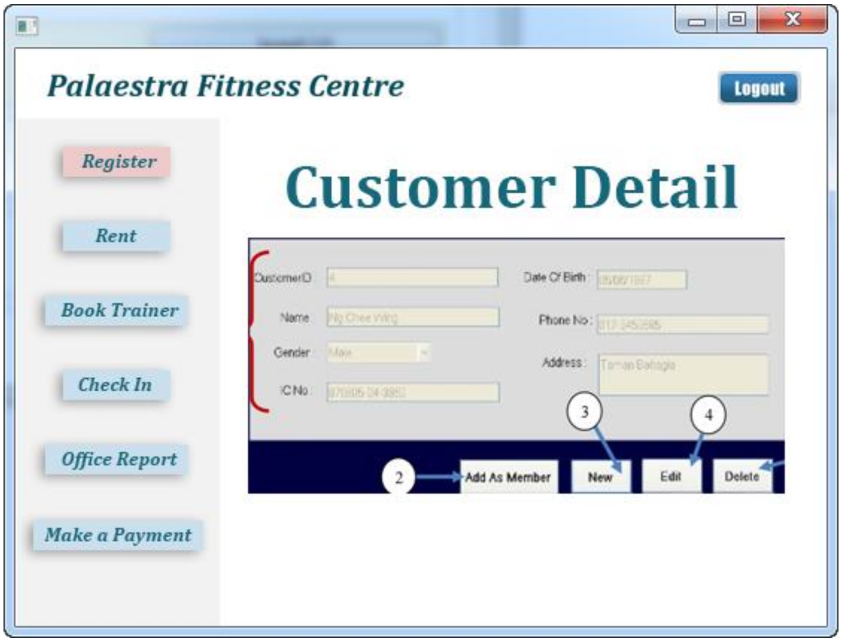


Palaestra Fitness Centre

Thank You !!

Check-In Successful

REGISTER



CustomerID : 4

Name : Ng Chee Wing

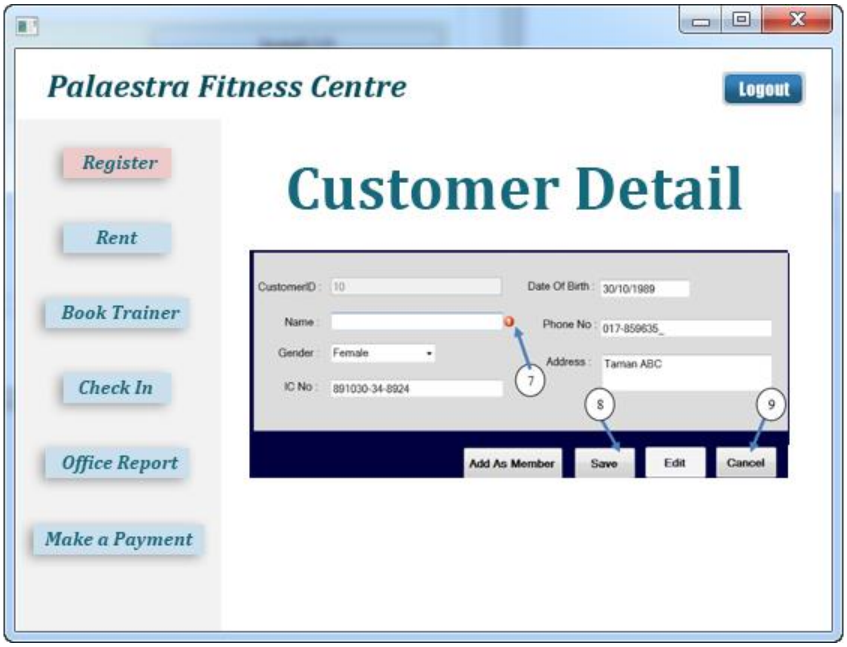
Gender : Male

IC No : 870905-04-3950

Date Of Birth : 05/06/1987

Phone No : 017-3453595

Address : Taman Bahagia



CustomerID : 10

Name :

Gender : Female

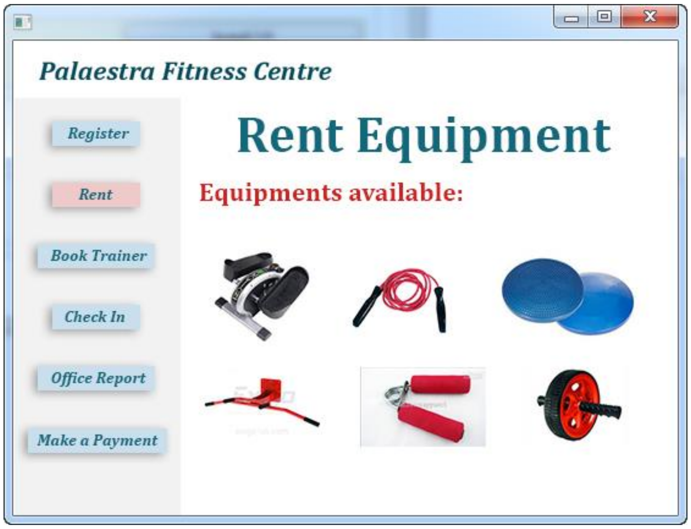
IC No : 891030-34-8924

Date Of Birth : 30/10/1989

Phone No : 017-859635_

Address : Taman ABC

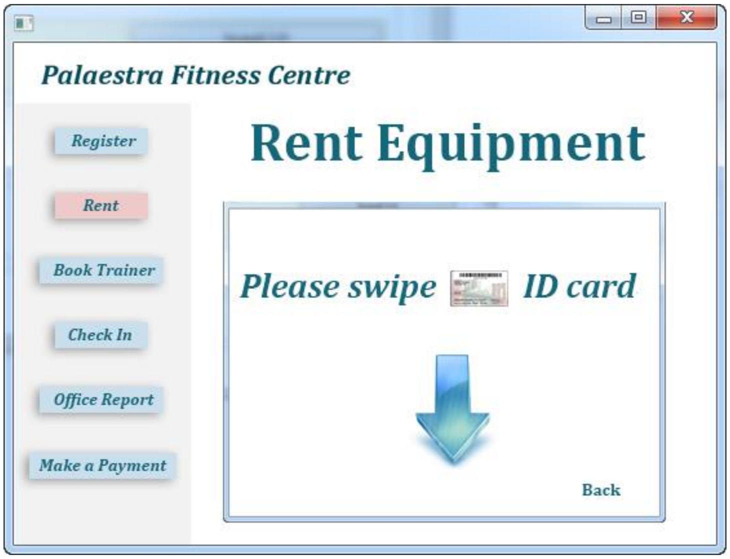
RENT EQUIPMENT



Palaestra Fitness Centre

Rent Equipment

Equipments available:

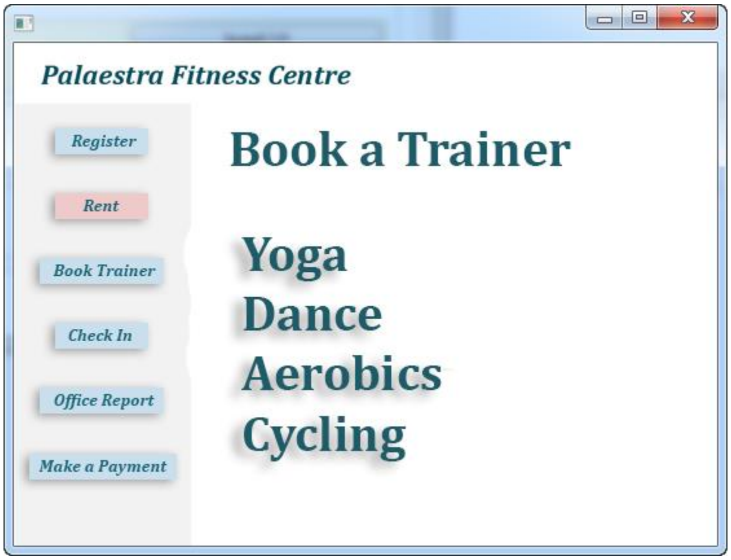


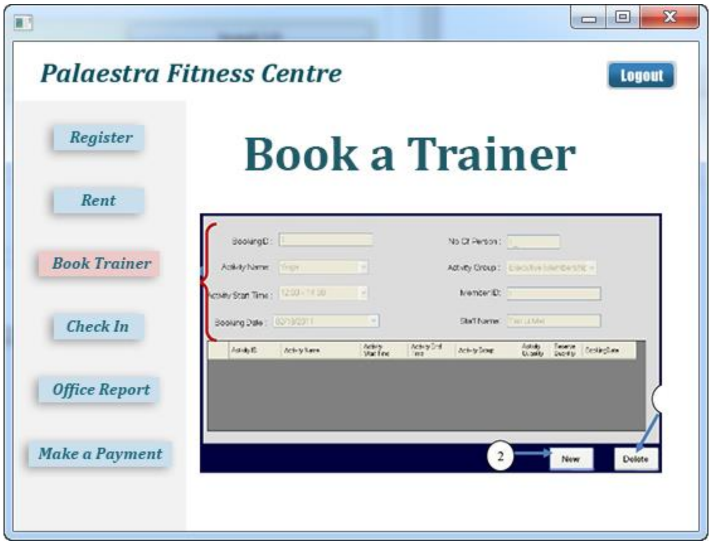
Palaestra Fitness Centre

Rent Equipment

Please swipe ID card

BOOK MEMBER





BookingID : 1

No Of Person : 1

Activity Name : Yoga

Activity Group : Executive Membership

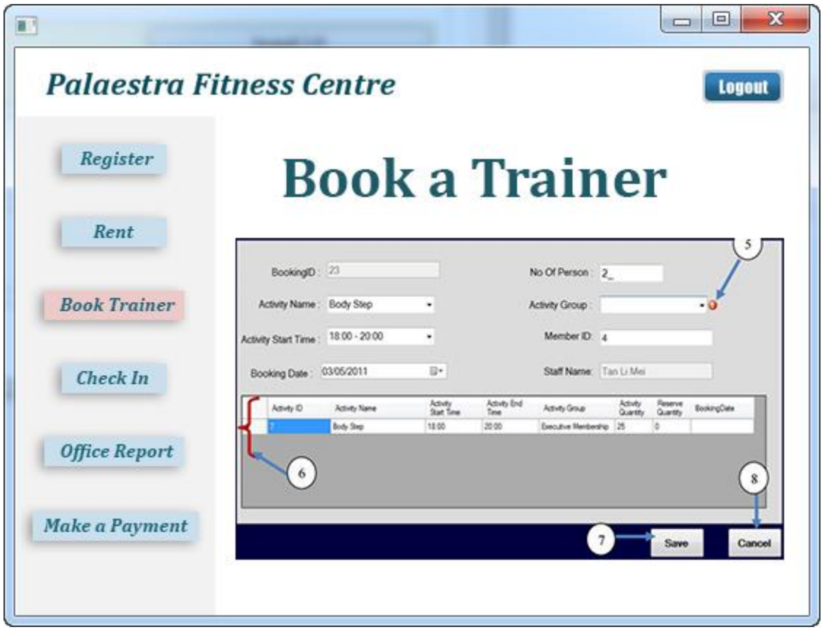
Activity Start Time : 12:00 - 14:30

MemberID : 1

Booking Date : 02/19/2011

Staff Name : Tan Li Mei

	Activity ID	Activity Name	Activity Start Time	Activity End Time	Activity Group	Activity Quantity	Reserve Quantity	BookingDate



BookingID : 23

No Of Person : 2_

Activity Name : Body Step

Activity Group :

Activity Start Time : 18:00 - 20:00

Member ID: 4

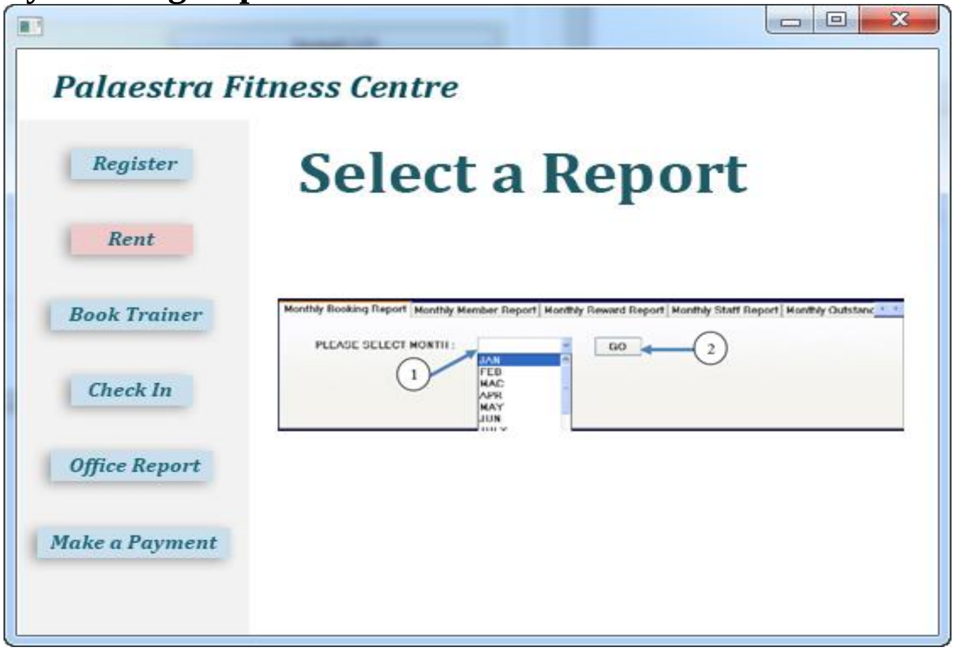
Booking Date : 03/05/2011

Staff Name: Tan Li Mei

Activity ID	Activity Name	Activity Start Time	Activity End Time	Activity Group	Activity Quantity	Reserve Quantity	BookingDate
7	Body Step	18:00	20:00	Executive Membership	25	0	

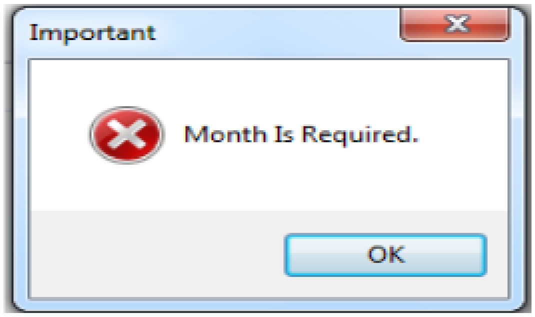
OFFICE REPORTS

Monthly Booking Report



PLEASE SELECT MONTH :

1. Click Month drop down list to select the Month.

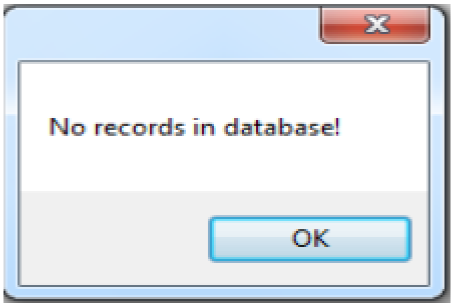


Important

Month Is Required.

This message will show when user do not select the month.

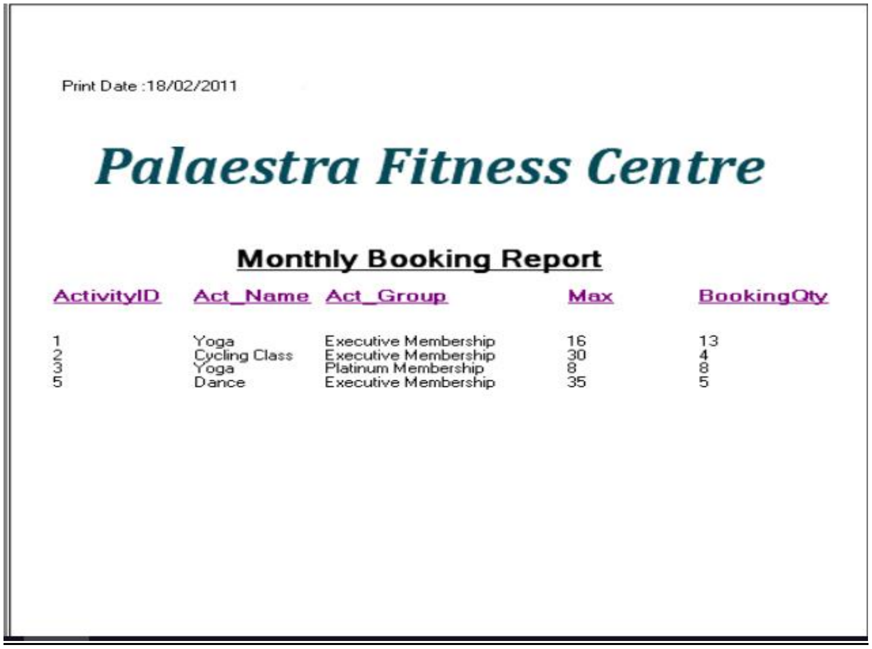
2. Click to search monthly booking data.



No records in database!

This message will show when no record in database.

Monthly Booking Report



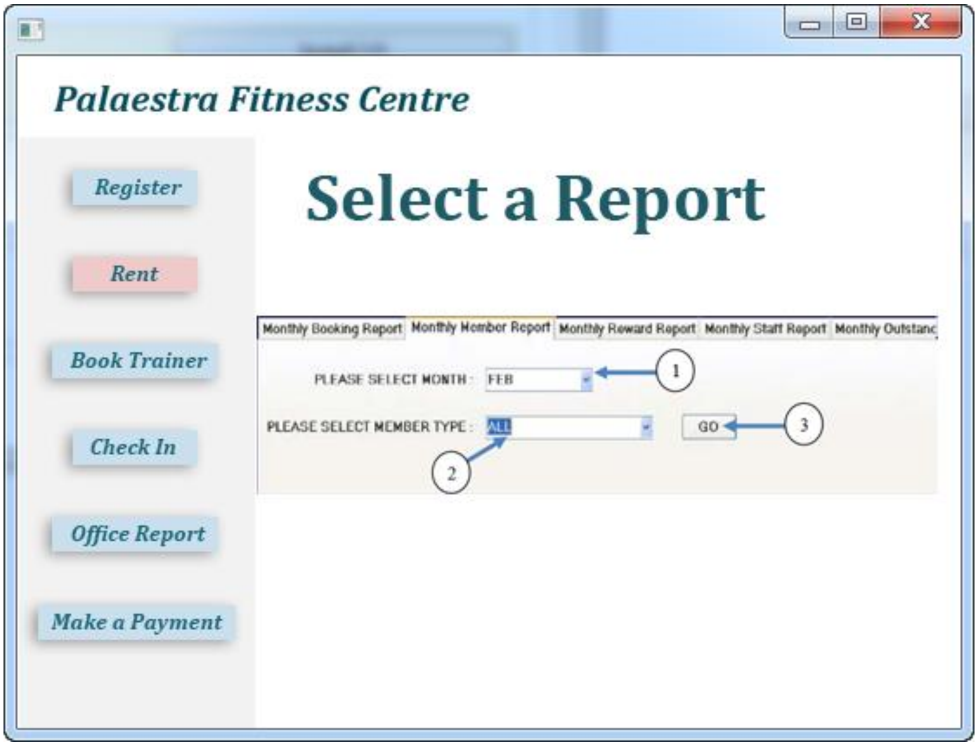
Print Date :18/02/2011

Palaestra Fitness Centre

Monthly Booking Report

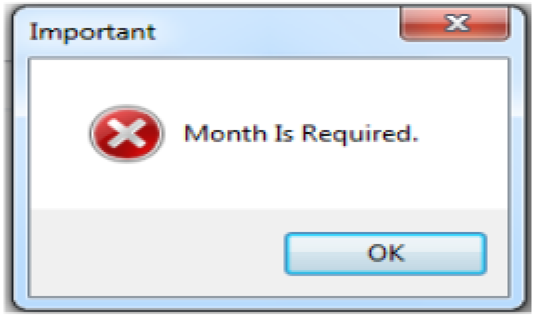
ActivityID	Act_Name	Act_Group	Max	BookingQty
1	Yoga	Executive Membership	16	13
2	Cycling Class	Executive Membership	30	4
3	Yoga	Platinum Membership	8	8
5	Dance	Executive Membership	35	5

Monthly Member Report



1. Click Month drop down list to select the Month

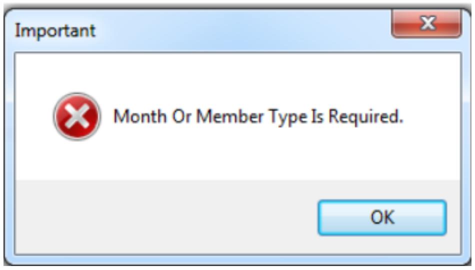
   This message will show when user do not select the month.



Important

Month Is Required.

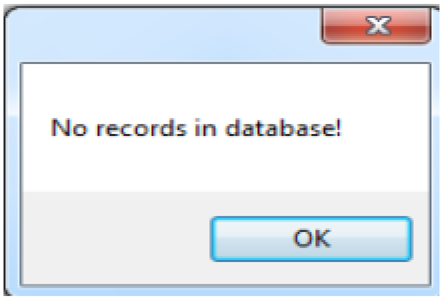
2. Click Member Type drop down list to select the member type.



Important

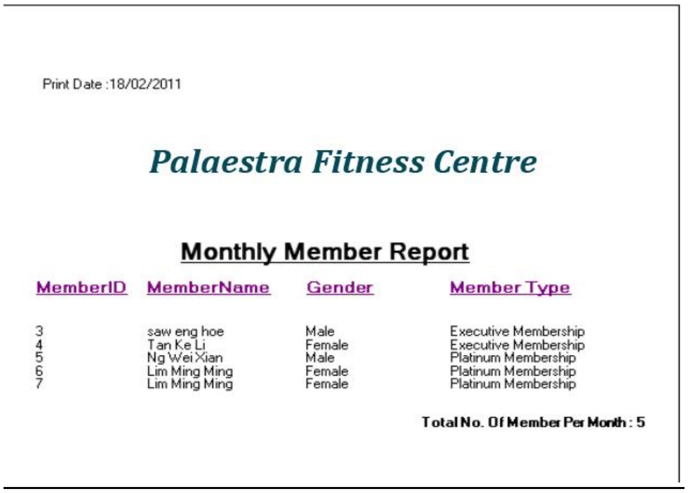
Month Or Member Type Is Required.

3. Click to search monthly member data


No records in database!

This message will show when no record in database.

Monthly Member Report


Print Date :18/02/2011

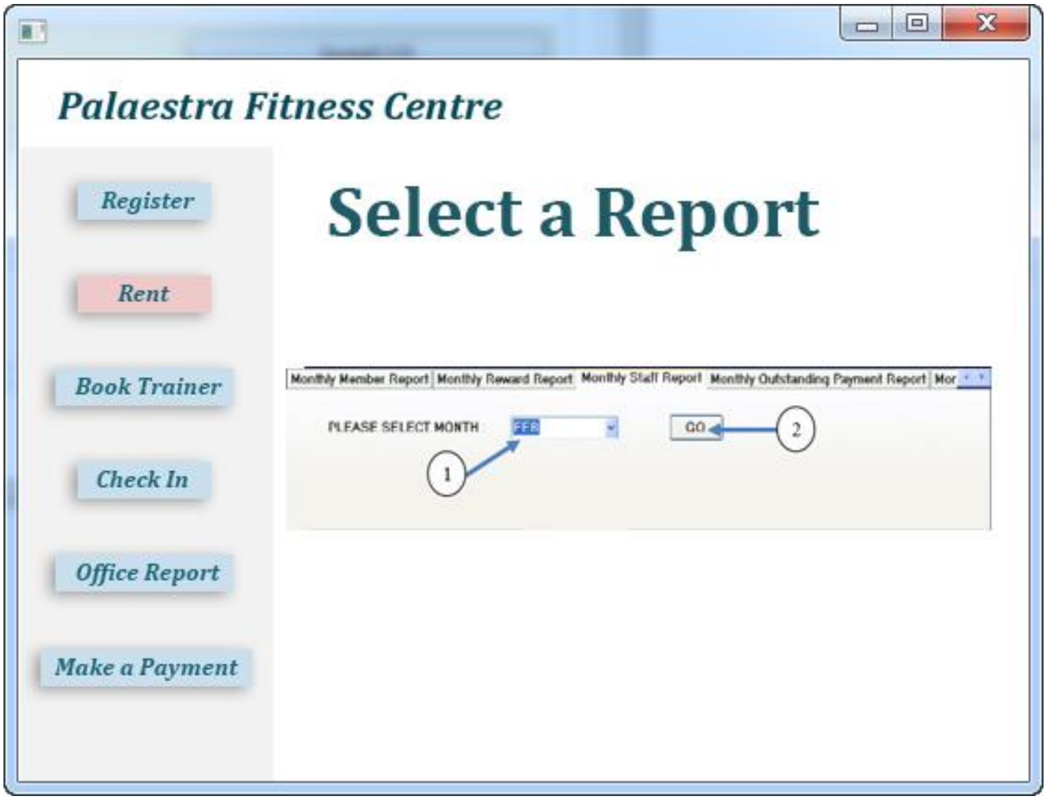
Palaestra Fitness Centre

Monthly Member Report

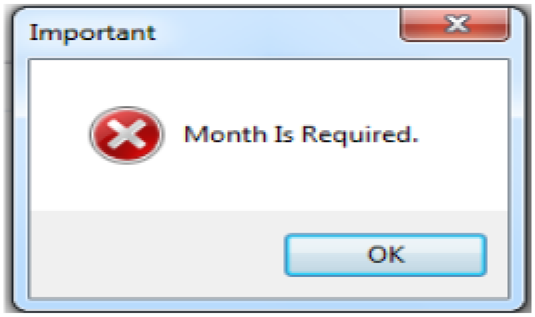
MemberID	MemberName	Gender	Member Type
3	saw eng hoe	Male	Executive Membership
4	Tan Ke Li	Female	Executive Membership
5	Ng Wei Xian	Male	Platinum Membership
6	Lim Ming Ming	Female	Platinum Membership
7	Lim Ming Ming	Female	Platinum Membership

Total No. Of Member Per Month : 5

Monthly Staff Report



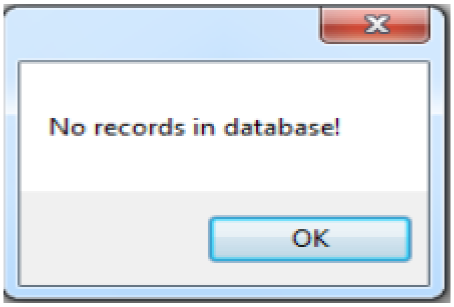
1. Click Month drop down list to select the Month.



Month Is Required.

This message will show when user do not select the month.

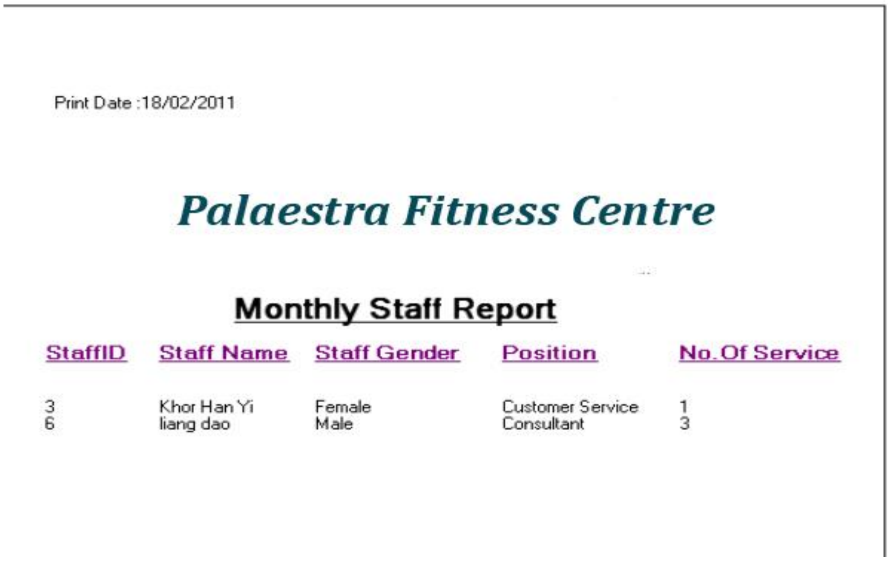
2. Click to search monthly booking data.



No records in database!

This message will show when no record in database.

Monthly Member Report



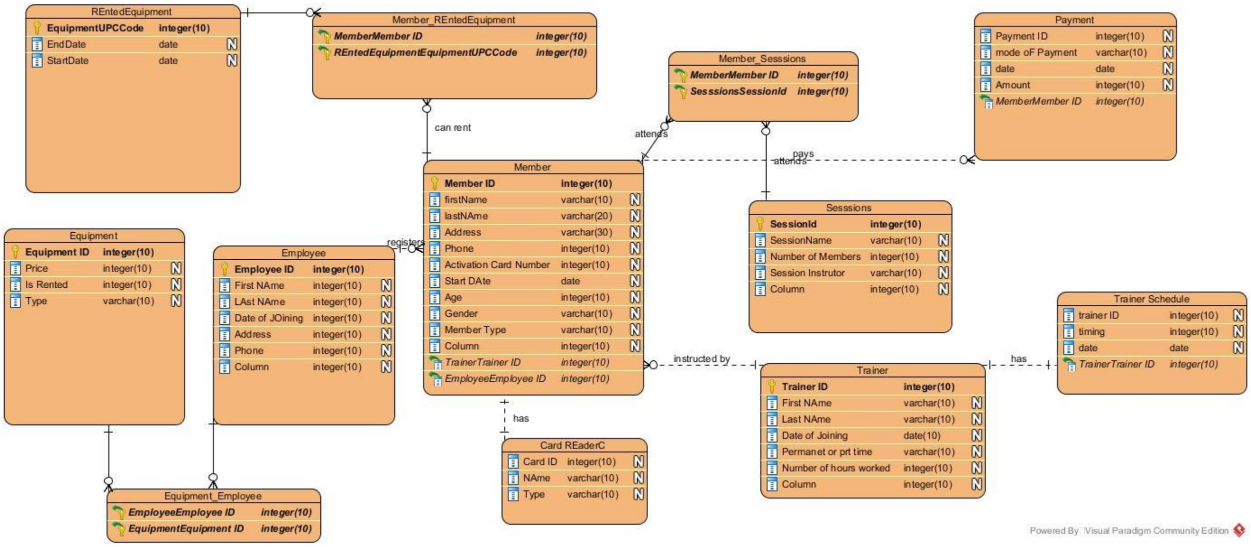
Print Date :18/02/2011

Palaestra Fitness Centre

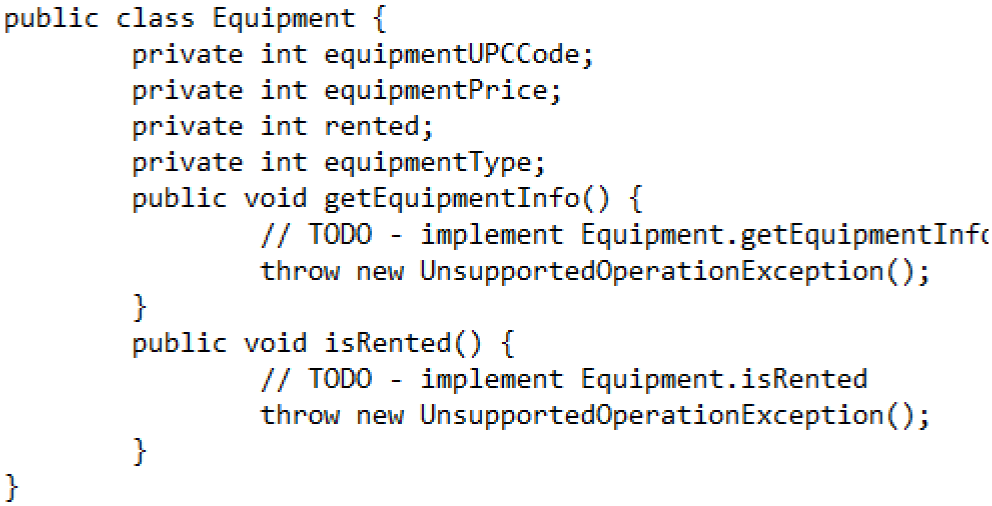
Monthly Staff Report

StaffID	Staff Name	Staff Gender	Position	No.Of Service
3	Khor Han Yi	Female	Customer Service	1
6	liang dao	Male	Consultant	3

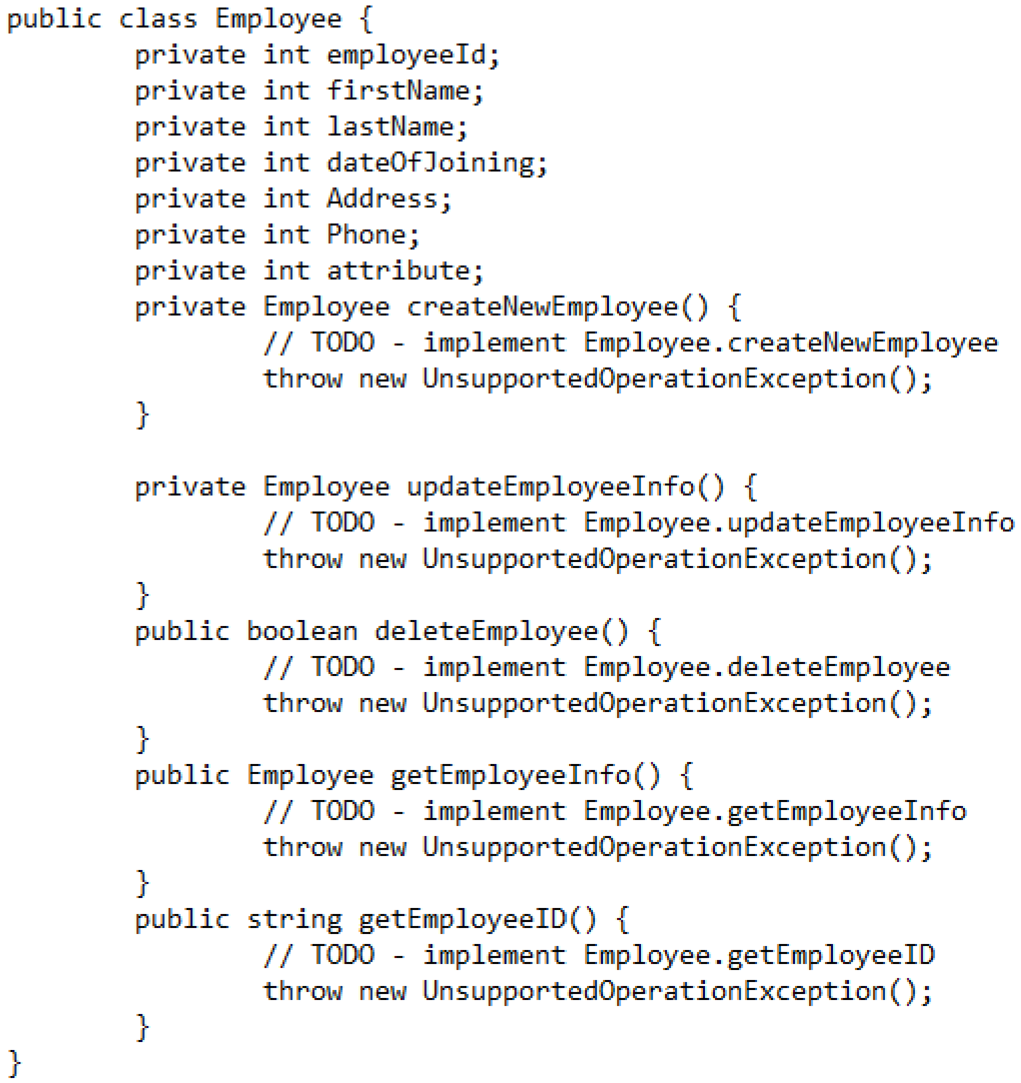
ERD



STUB CODE

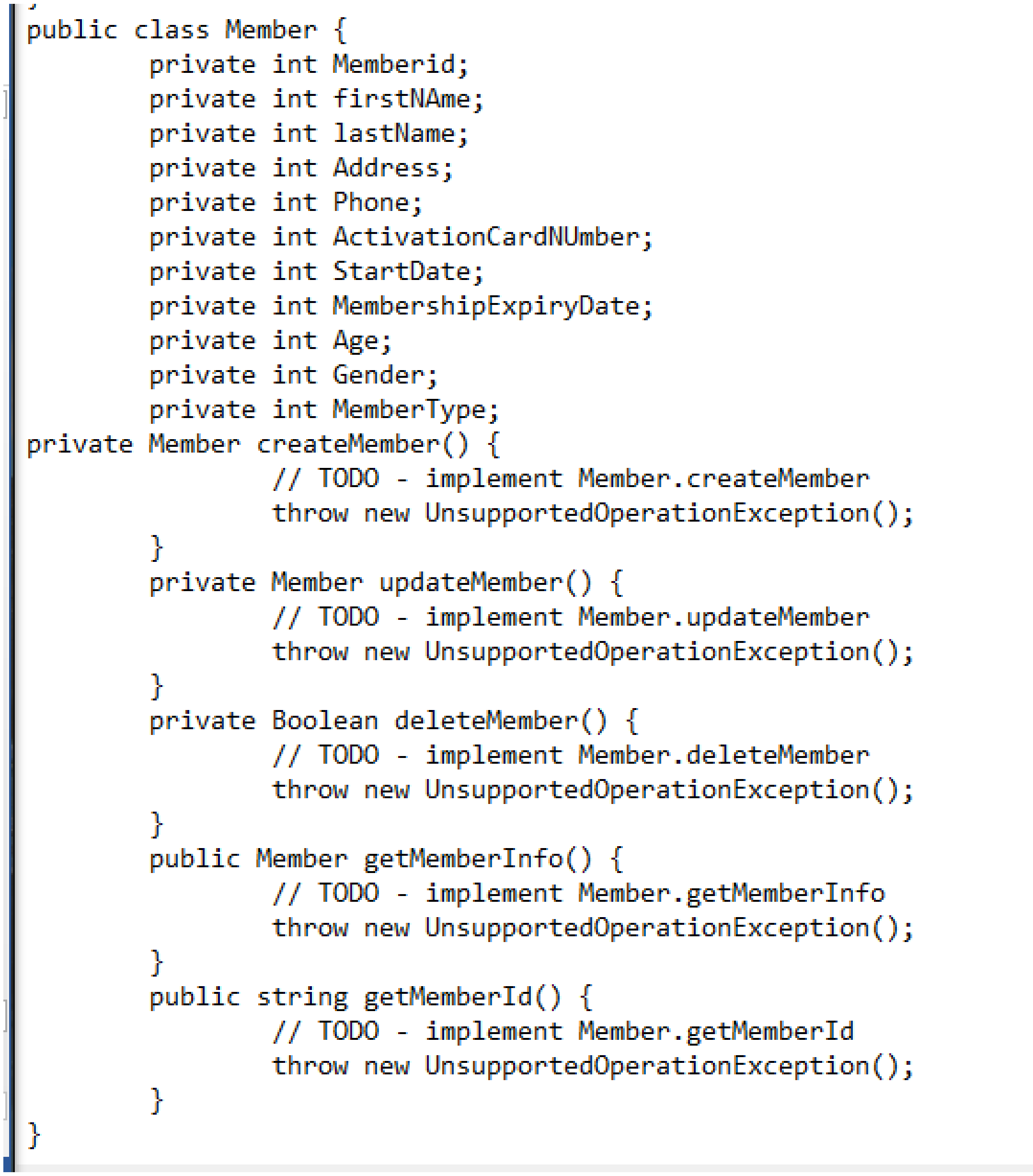


```
public class Equipment {
        private int equipmentUPCCode;
        private int equipmentPrice;
        private int rented;
        private int equipmentType;
        public void getEquipmentInfo() {
                // TODO - implement Equipment.getEquipmentInf
                throw new UnsupportedOperationException();
        }
        public void isRented() {
                // TODO - implement Equipment.isRented
                throw new UnsupportedOperationException();
        }
}
```

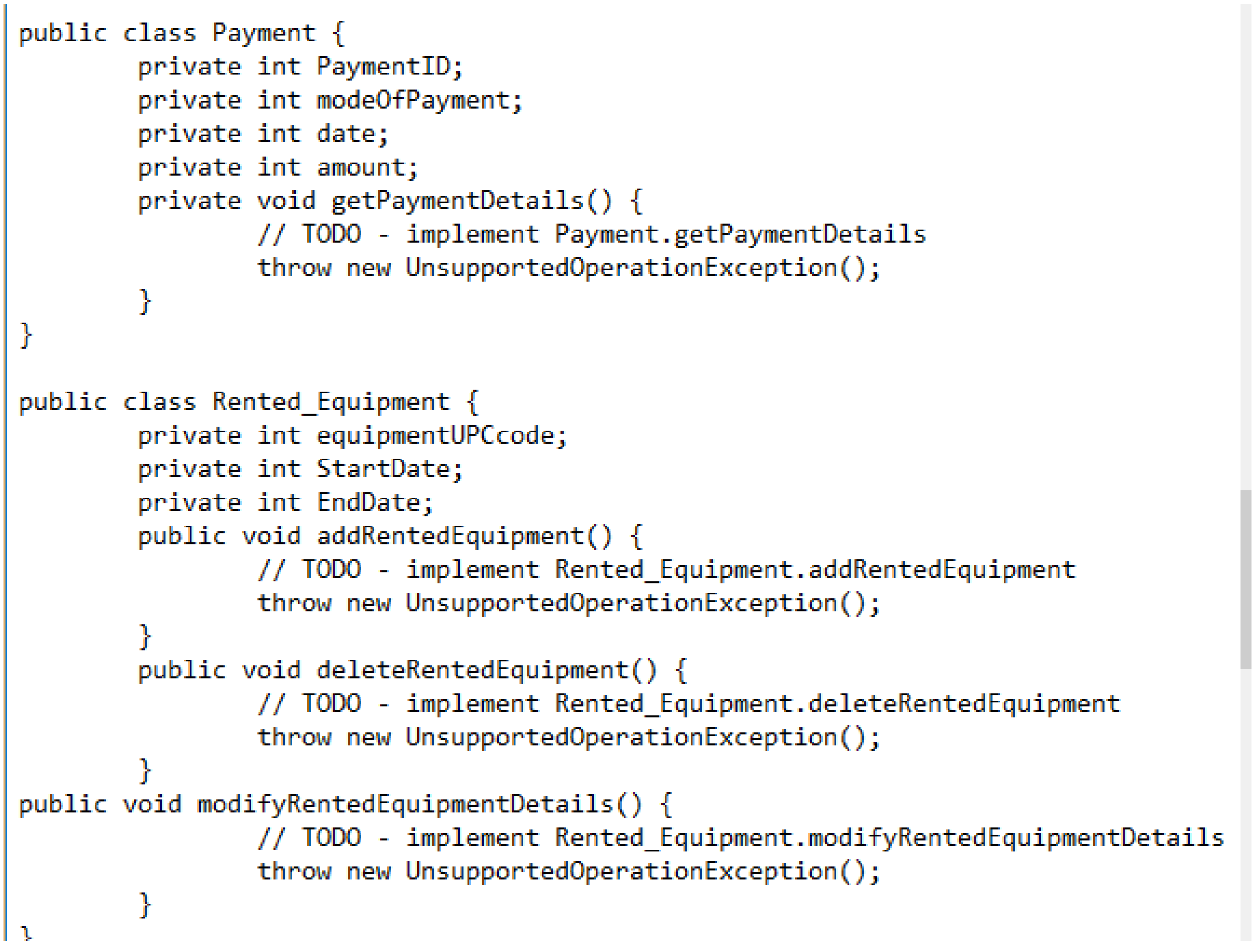


```
public class Employee {
        private int employeeId;
        private int firstName;
        private int lastName;
        private int dateOfJoining;
        private int Address;
        private int Phone;
        private int attribute;
        private Employee createNewEmployee() {
                // TODO - implement Employee.createNewEmployee
                throw new UnsupportedOperationException();
        }

        private Employee updateEmployeeInfo() {
                // TODO - implement Employee.updateEmployeeInfo
                throw new UnsupportedOperationException();
        }
        public boolean deleteEmployee() {
                // TODO - implement Employee.deleteEmployee
                throw new UnsupportedOperationException();
        }
        public Employee getEmployeeInfo() {
                // TODO - implement Employee.getEmployeeInfo
                throw new UnsupportedOperationException();
        }
        public string getEmployeeID() {
                // TODO - implement Employee.getEmployeeID
                throw new UnsupportedOperationException();
        }
}
```

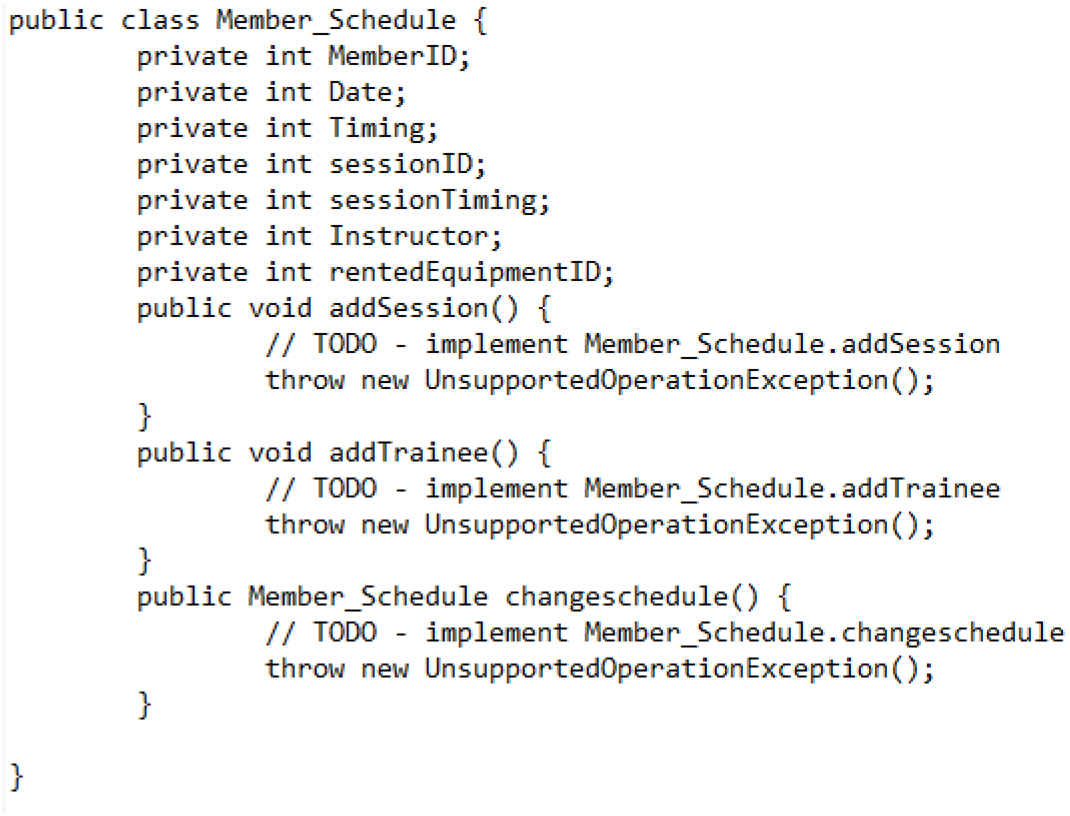


```
public class Member {
        private int Memberid;
        private int firstNAme;
        private int lastName;
        private int Address;
        private int Phone;
        private int ActivationCardNUmber;
        private int StartDate;
        private int MembershipExpiryDate;
        private int Age;
        private int Gender;
        private int MemberType;
private Member createMember() {
                // TODO - implement Member.createMember
                throw new UnsupportedOperationException();
        }
        private Member updateMember() {
                // TODO - implement Member.updateMember
                throw new UnsupportedOperationException();
        }
        private Boolean deleteMember() {
                // TODO - implement Member.deleteMember
                throw new UnsupportedOperationException();
        }
        public Member getMemberInfo() {
                // TODO - implement Member.getMemberInfo
                throw new UnsupportedOperationException();
        }
        public string getMemberId() {
                // TODO - implement Member.getMemberId
                throw new UnsupportedOperationException();
        }
}
```



```
public class Payment {
        private int PaymentID;
        private int modeOfPayment;
        private int date;
        private int amount;
        private void getPaymentDetails() {
                // TODO - implement Payment.getPaymentDetails
                throw new UnsupportedOperationException();
        }
}

public class Rented_Equipment {
        private int equipmentUPCcode;
        private int StartDate;
        private int EndDate;
        public void addRentedEquipment() {
                // TODO - implement Rented_Equipment.addRentedEquipment
                throw new UnsupportedOperationException();
        }
        public void deleteRentedEquipment() {
                // TODO - implement Rented_Equipment.deleteRentedEquipment
                throw new UnsupportedOperationException();
        }
public void modifyRentedEquipmentDetails() {
                // TODO - implement Rented_Equipment.modifyRentedEquipmentDetails
                throw new UnsupportedOperationException();
        }
}
```



```java
public class Member_Schedule {
        private int MemberID;
        private int Date;
        private int Timing;
        private int sessionID;
        private int sessionTiming;
        private int Instructor;
        private int rentedEquipmentID;
        public void addSession() {
                // TODO - implement Member_Schedule.addSession
                throw new UnsupportedOperationException();
        }
        public void addTrainee() {
                // TODO - implement Member_Schedule.addTrainee
                throw new UnsupportedOperationException();
        }
        public Member_Schedule changeschedule() {
                // TODO - implement Member_Schedule.changeschedule
                throw new UnsupportedOperationException();
        }

}
```

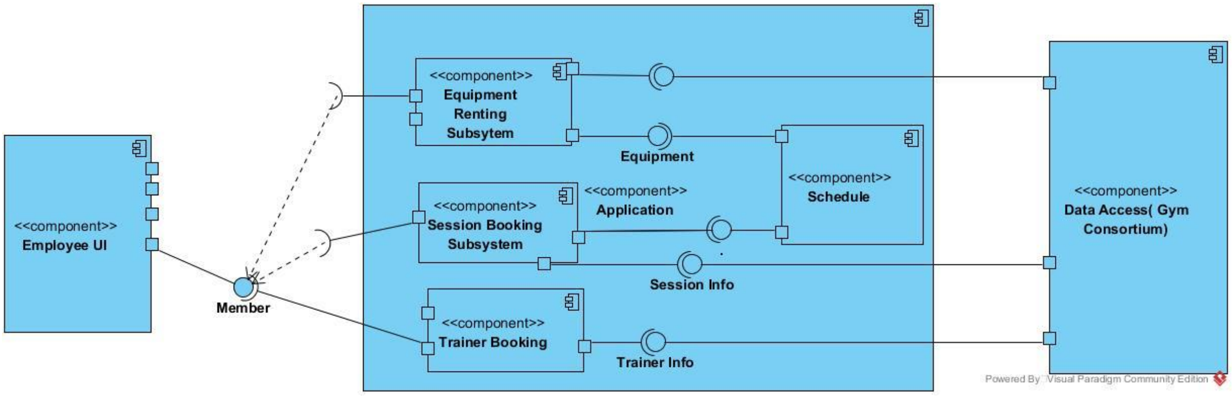
```java
public class Trainer_Schedule {
        private int trainerID;
        private int memberId;
        private int timing;
        private int date;
        public void addmember() {
                // TODO - implement Trainer_Schedule.addmember
                throw new UnsupportedOperationException();
        }

        public Trainer_Schedule changeSchedule() {
                // TODO - implement Trainer_Schedule.changeSchedule
                throw new UnsupportedOperationException();
        }
}
```

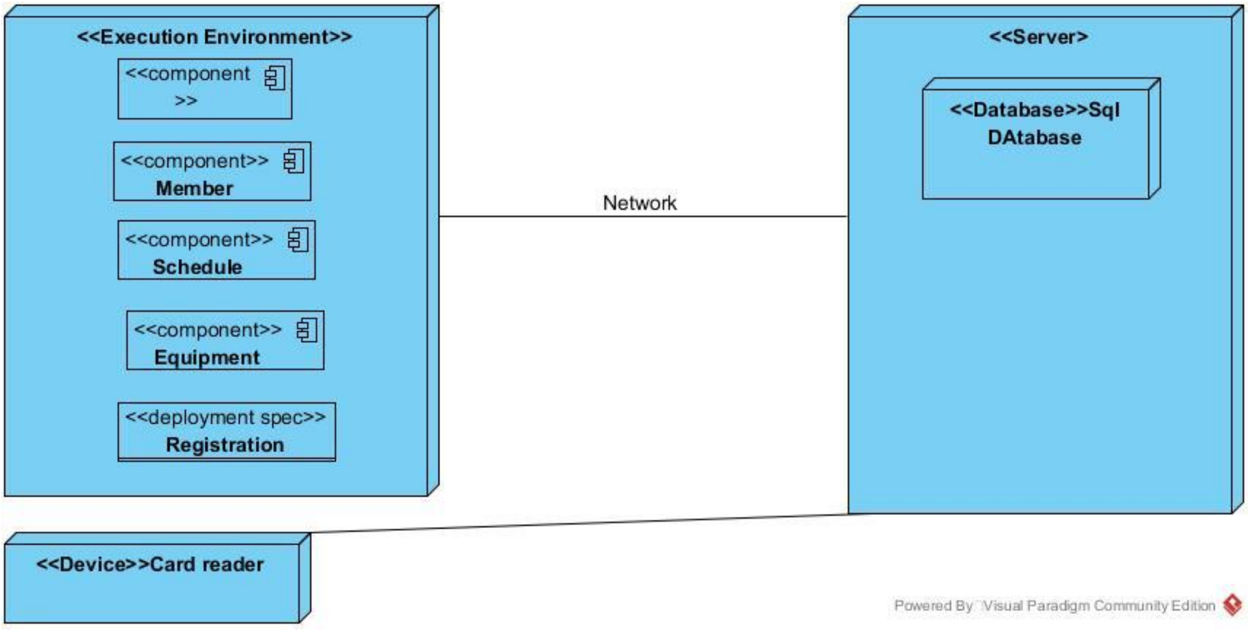
```
public abstract class Trainer {
private int trainerId;
        private int firstName;
        private int lastName;
        private int dateOfJoining;
        private int permanentorPartTime;
        private int numberOFHoursWorked;
        private Trainer createNewTrainer() {
                // TODO - implement Trainer.createNewTrainer
                throw new UnsupportedOperationException();
        }
        public Trainer updateTrainerInfo() {
                // TODO - implement Trainer.updateTrainerInfo
                throw new UnsupportedOperationException();
        }
        public boolean deleteTrainer() {
                // TODO - implement Trainer.deleteTrainer
                throw new UnsupportedOperationException();
        }
        public Trainer updateTrainer() {
                // TODO - implement Trainer.updateTrainer
                throw new UnsupportedOperationException();
        }
        public float AddNUmberofHours() {
                // TODO - implement Trainer.AddNUmberofHours
                throw new UnsupportedOperationException();
        }
        public Trainer getTrainerInfo() {
                // TODO - implement Trainer.getTrainerInfo
                throw new UnsupportedOperationException();
        }
}
```

```
public class Sessions {
        private int SessionName;
        private int sessiontimings;
        private int numberOFMembers;
        private int sessionInstructor;
        public Session getSessionInfo() {
                // TODO - implement Sessions.getSessionInfo
                throw new UnsupportedOperationException();
        }
        public Session addMember() {
                // TODO - implement Sessions.addMember
                throw new UnsupportedOperationException();
        }
        private Session removeMember() {
                // TODO - implement Sessions.removeMember
                throw new UnsupportedOperationException();
        }
        private Session changeSessionTimings() {
                // TODO - implement Sessions.changeSessionTimings
                throw new UnsupportedOperationException();
        }
        private Session changeInstructor() {
                // TODO - implement Sessions.changeInstructor
                throw new UnsupportedOperationException();
        }
}
```

COMPONENT DIAGRAM



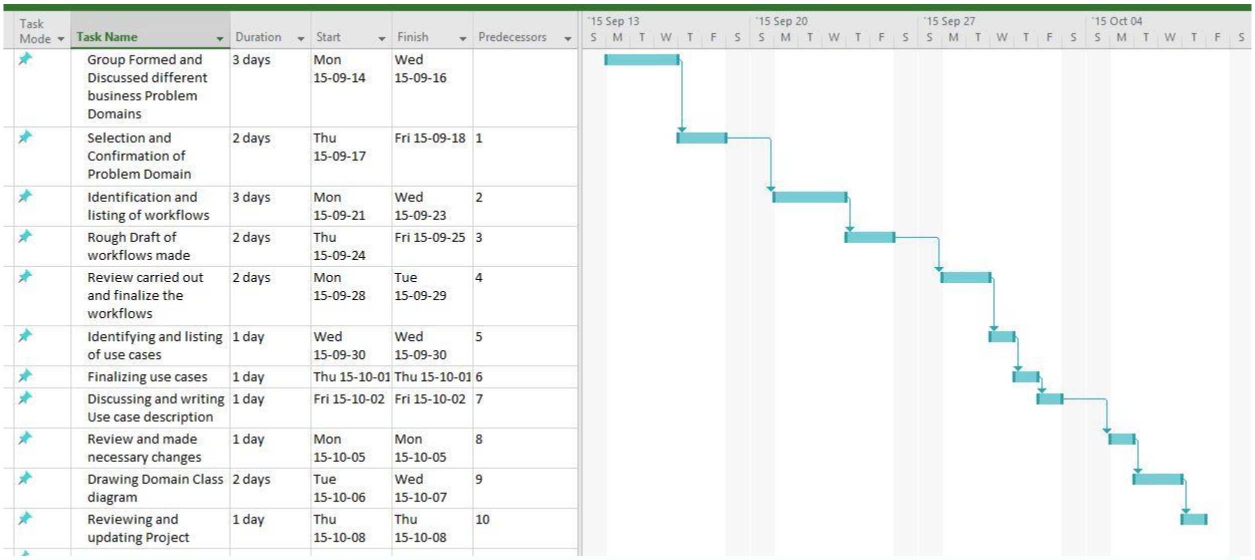
DEPLOYMENT DIAGRAM



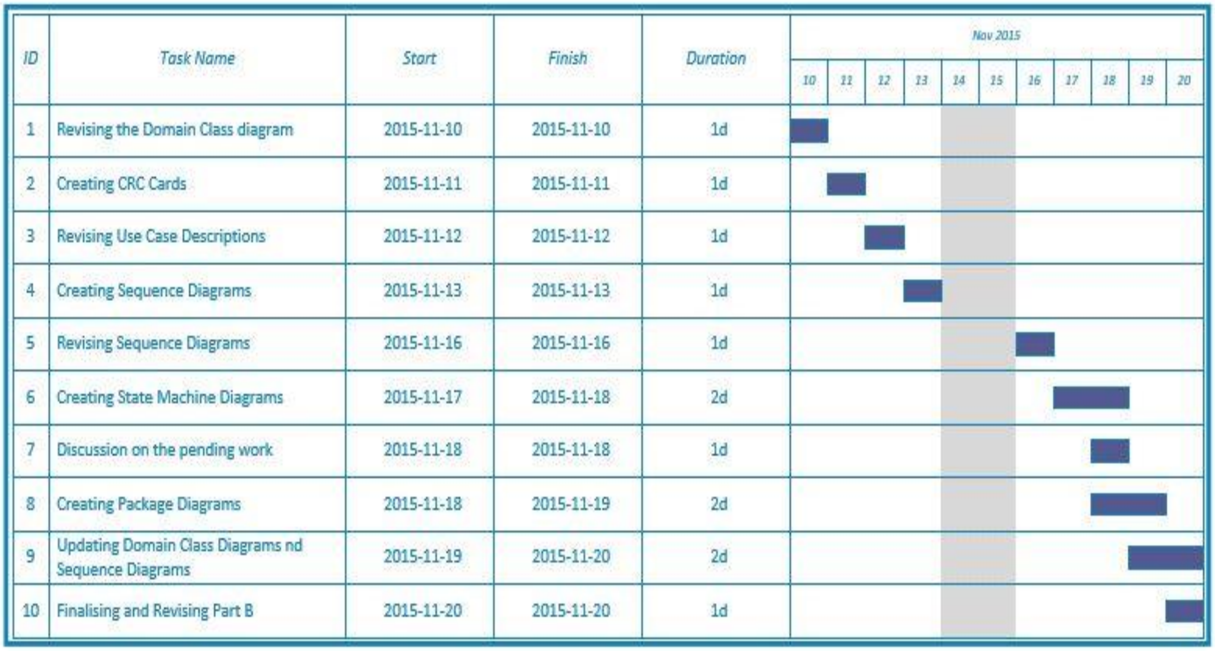
List a set of technology/development tools

3-Tier Architecture	Development Tools	Comments
Presentation Tier	Java Adobe Photoshop CC	Developers use to create applications with a web plug-in that allows you to design and test the apps.
Business Logic Tier	Java XML	Major features to accommodate functional-style programming and attaining commands over logical statements.
Data Tier	SQL Server	It is used to store and update information that needs to be changed in various states.

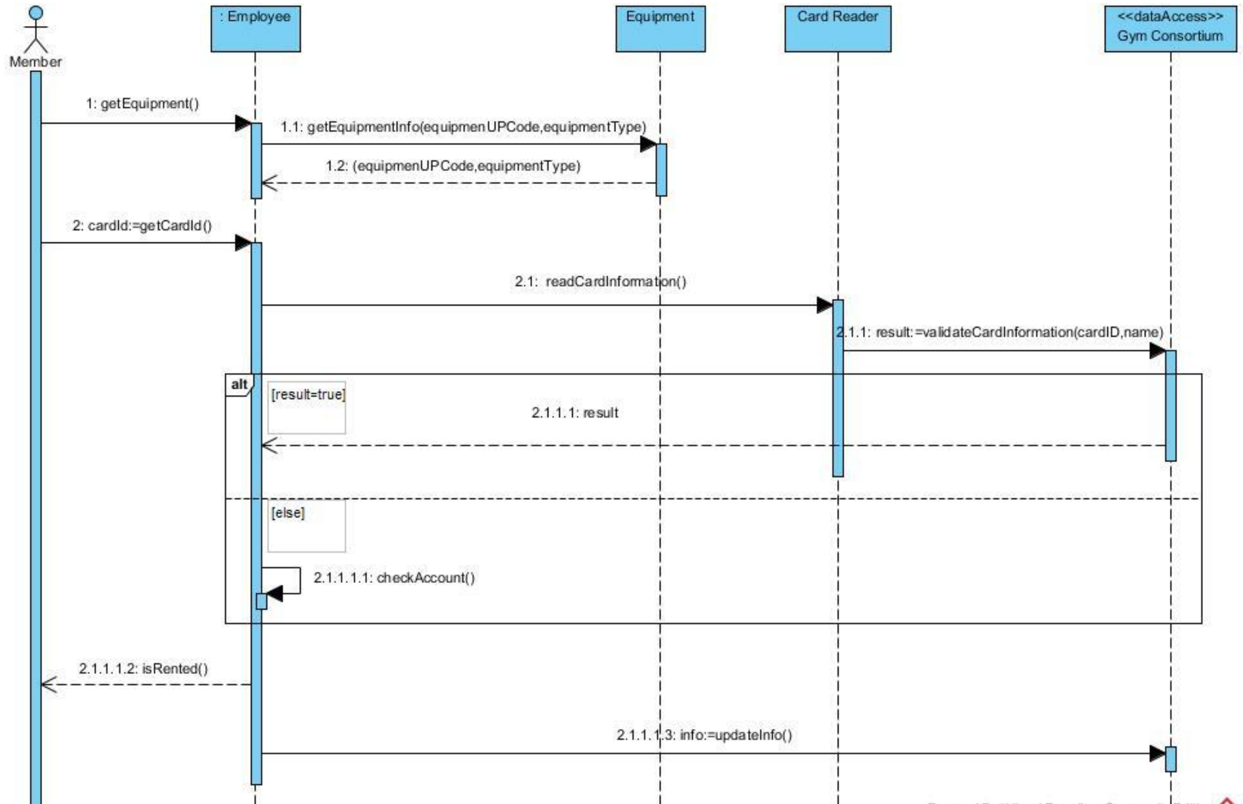
GANTT CHART

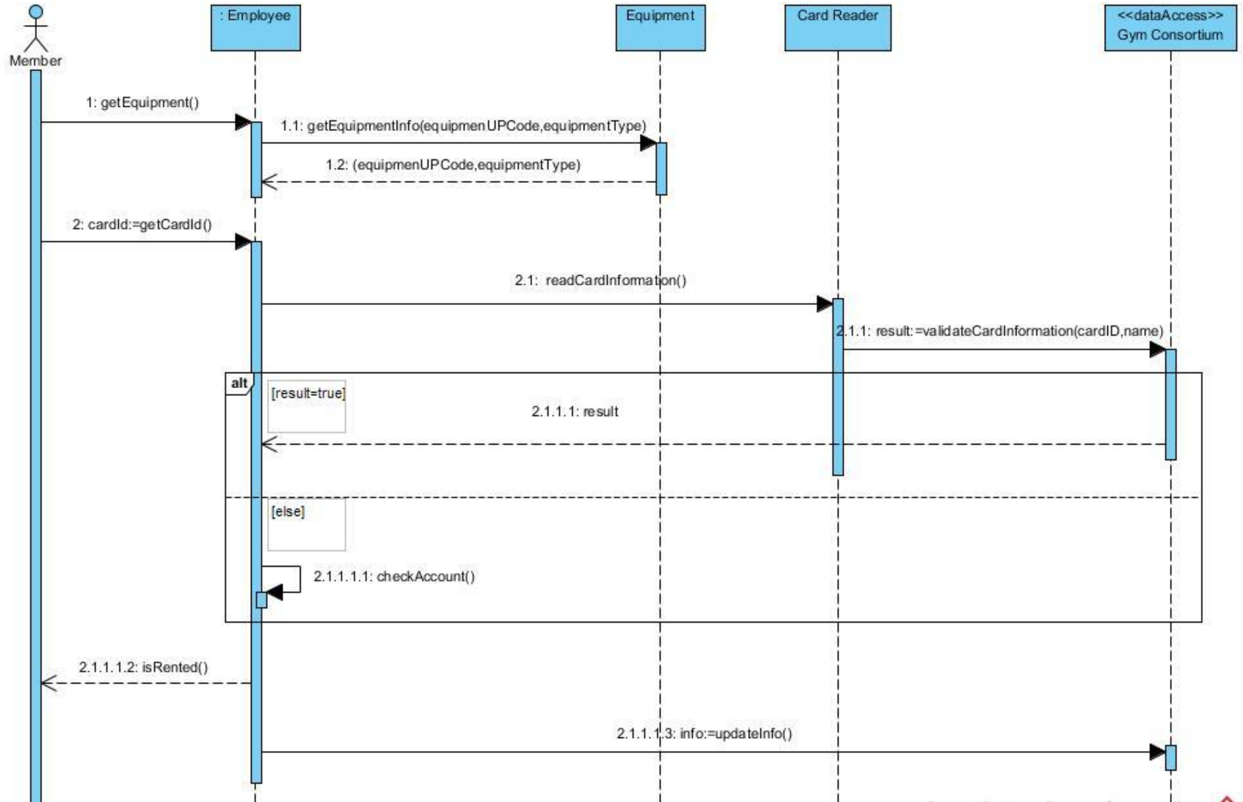


Task Mode	Task Name	Duration	Start	Finish	Predecessors
📌	Group Formed and Discussed different business Problem Domains	3 days	Mon 15-09-14	Wed 15-09-16	
📌	Selection and Confirmation of Problem Domain	2 days	Thu 15-09-17	Fri 15-09-18	1
📌	Identification and listing of workflows	3 days	Mon 15-09-21	Wed 15-09-23	2
📌	Rough Draft of workflows made	2 days	Thu 15-09-24	Fri 15-09-25	3
📌	Review carried out and finalize the workflows	2 days	Mon 15-09-28	Tue 15-09-29	4
📌	Identifying and listing of use cases	1 day	Wed 15-09-30	Wed 15-09-30	5
📌	Finalizing use cases	1 day	Thu 15-10-01	Thu 15-10-01	6
📌	Discussing and writing Use case description	1 day	Fri 15-10-02	Fri 15-10-02	7
📌	Review and made necessary changes	1 day	Mon 15-10-05	Mon 15-10-05	8
📌	Drawing Domain Class diagram	2 days	Tue 15-10-06	Wed 15-10-07	9
📌	Reviewing and updating Project	1 day	Thu 15-10-08	Thu 15-10-08	10



ID	Task Name	Start	Finish	Duration
1	Revising the Domain Class diagram	2015-11-10	2015-11-10	1d
2	Creating CRC Cards	2015-11-11	2015-11-11	1d
3	Revising Use Case Descriptions	2015-11-12	2015-11-12	1d
4	Creating Sequence Diagrams	2015-11-13	2015-11-13	1d
5	Revising Sequence Diagrams	2015-11-16	2015-11-16	1d
6	Creating State Machine Diagrams	2015-11-17	2015-11-18	2d
7	Discussion on the pending work	2015-11-18	2015-11-18	1d
8	Creating Package Diagrams	2015-11-18	2015-11-19	2d
9	Updating Domain Class Diagrams nd Sequence Diagrams	2015-11-19	2015-11-20	2d
10	Finalising and Revising Part B	2015-11-20	2015-11-20	1d


Member
: Employee
Equipment
Card Reader
<<dataAccess>>
Gym Consortium
1: getEquipment()
1.1: getEquipmentInfo(equipmenUPCode,equipmentType)
1.2: (equipmenUPCode,equipmentType)
2: cardId:=getCardId()
2.1: readCardInformation()
2.1.1: result:=validateCardInformation(cardID,name)
alt
[result=true]
2.1.1.1: result
[else]
2.1.1.1.1: checkAccount()
2.1.1.1.2: isRented()
2.1.1.1.3: info:=updateInfo()


Member
: Employee
Equipment
Card Reader
<<dataAccess>>
Gym Consortium
1: getEquipment()
1.1: getEquipmentInfo(equipmenUPCode,equipmentType)
1.2: (equipmenUPCode,equipmentType)
2: cardId:=getCardId()
2.1: readCardInformation()
2.1.1: result:=validateCardInformation(cardID,name)
alt
[result=true]
2.1.1.1: result
[else]
2.1.1.1.1: checkAccount()
2.1.1.1.2: isRented()
2.1.1.1.3: info:=updateInfo()