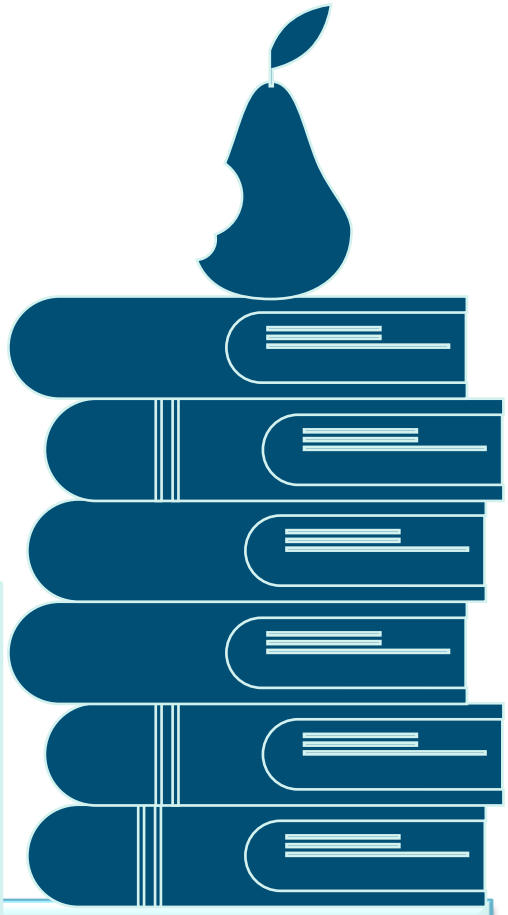
Software Engineering Class Fall 2020/2021
Project Report
Software requirement specification

Central Library Organization and Management Software (CLOMS)



SEPTEMBER 7

Team Number: 3

Prepared by:

-Rahma Mohamed Makram Saied.
Section 2
Student Mail: Rahma.Saeid.82@h-eng.helwan.edu.eg

-Rana Mohamed Hussein Mohamed Bekheet.
Section 3
Student Mail: Rana.Bekheet.82@h-eng.helwan.edu.eg

-Sarah Gamal El Deen Mohamed.
Section 3
Student Mail: Sarah.Mohamed.48@h-eng.helwan.edu.eg

-Solwan Shokry Ahmed Mohamed (Team Leader)
Section 3
Student Mail: Solwan.Mohamed.89@h-eng.helwan.edu.eg

Breakdown of individual contributions:

Contributions of: Rahma Mohamed Makram

- Analyzing the customer's problem statement and deduced the system's functional requirements.
- Deduced the actors and goals of the system from the problem statement.
- Use Case diagram description:
  - Use case 01
  - Use case 07
  - Use case 08
  - Use case 09
  - Use case 11
  - Use case 12
  - Use case 13
  - Use case 15
- Use case diagram
- Sequence Diagram:
  - Register visitor Account.
  - Lending books and resources.
- Class Diagram and interface specification.
- Class Diagram description:
  - Explaining the UML attributes and operations notations in plain language
- Entity Relationship Diagram.
- Data Flow Diagram.

Contributions of: *Rana Mohamed Hussein*

- Shared in stating the customer's problems to be analyzed.
- Analyzing the customer's problem statement and deduced the system's functional requirements.
- Deduced the stakeholders of the system from the problem statement.
- Use Case diagram description:
  - Use case 01
  - Use case 02
  - Use case 04
  - Use case 05
  - Use case 06
  - Use case 08
  - Use case 14
  - Use case 15
- Use case diagram.
- Sequence Diagram:
  - Register visitor Account.
  - Searching E-Books.
  - Sending membership expiration dates.
- Class Diagram and interface specification.
- Class Diagram description:
  - Explaining the UML relations between classes notations in plain language
- Entity Relationship Diagram.
- Data Flow Diagram.

Contributions of: *Sarah Gamal El-Deen Mohamed*

- Deduced the stakeholders of the system from the problem statement.
- Analyzing the customer's problem statement and deduced the system's functional requirements.
- Use Case diagram description:
  - Use case 03
  - Use case 05
  - Use case 06
  - Use case 07
  - Use case 08
  - Use case 10
  - Use case 14
  - Use case 16
- Use case diagram.
- Sequence Diagram:
  - Register visitor Account.
  - Searching E-Books.
  - Booking internal computers.
- Class Diagram and interface specification.
- Class Diagram description:
  - Explaining the UML relations between classes notations in plain language.
- Entity Relationship Diagram.
- Data Flow Diagram.

Contributions of: *Solwan Shokry Ahmed Mohamed*

- Shared in stating the customer's problems to be analyzed.
- Deduced the actors and goals of the system from the problem statement.
- Analyzing the customer's problem statement and deduced the system's functional requirements.
- Use Case diagram description:
  - Use case 01
  - Use case 02
  - Use case 03
  - Use case 04
  - Use case 09
  - Use case 11
  - Use case 12
  - Use case 13
  - Use case 15
  - Use case 16
  - Use case diagram
- Sequence Diagram:
  - Register visitor Account.
  - Checking Books and resources availability.
- Class Diagram and interface specification.
- Class Diagram description:
  - Explaining the UML attributes and operations notations in plain language
- Entity Relationship Diagram.
- Data Flow Diagram.

1. Customer Problem Statement

Many library systems are operated manually by group of people. In such situations many people involved in the process of managing the library such a way that to keep records regarding the books & students (borrowers), check the books manually, keep records on issued books etc. all these things have to be carried out manually & if the library is large in content handling is also a problem. On the other hand, keeping large amount of maintenance workers may cost a lot & it will not be efficient for a library. Manual record keeping is also not a reliable method as people tend to forget things. On the borrower ’s point of view, in manual system borrower can’t find a book exactly at once as they are not ordered well.

Sometimes user might be searching for a book that is not available in the library in such situations people get annoyed or depressed. Therefore, there should be a reliable way to manage the library system. The feature of recording and retrieving user

and book history details is going to throw some very interesting results regarding the books most popular among the users and also their favorite genre according to age group etc.

Problem in Existing System

The existing system that is being used in majority of public libraries completely manual in nature. Information about all the books and members/users is maintained separately in data-entry registers. The entries made in each register are having a serial number corresponding to the register name and year. Also, there are separate registers for purchase and sale of books and for different user plans. This often leads to redundant information too. Though some libraries do have computerized systems that provide basic features such as adding books, user etc. and that of issuing the books, but what they lack is the user involvement and interactivity.

Limitations of existing system:

- Majority of libraries are dependent on paperwork which turns out to be very inefficient, and data backup is very difficult and tedious
- Users or members in such systems either become dependent on the librarian when they enquire about the books existing in the library or find themselves helpless when they try to search the library for their book of interest unknown of even its availability!!
- Present day systems involve the users very little in the entire process, and also do not consider the user wants for newer books etc. This non-involvement of end users also means that either they have to manually ask the librarian whether a particular book is already issued by some other user or they have to browse through the library.

2. System Requirements

a. Enumerated Functional Requirements

Requirement Number	Requirement brief description
Req.1	Adding library employees' data on the system.
Req.2	Entering the visitors' information to the system such as: • Visitor's ID. • Visitor's name. • Visitor's phone number. • Visitor's address.
Req.3	Assign membership to library visitors.
Req.4	Adding members' specific information such as: • Their educational state and their major (if students). • Their fields of interests. • Their E-mail addresses.
Req.5	Giving each library member a specified membership number to give to the receptionist to simplify the recognition processes.
Req.6	Checking books and resources' availability in the library.
Req.7	Managing the lending process of the library's books and resources.
Req.8	Booking a session on the library's internal computers.

Req.9	Checking if there are any available computers at the moment.
Req.10	Updating the number of computers used on the system.
Req.11	Updating library members on the newly arrived books and resources to the library according to their majors or fields of interest.
Req.12	Send membership expiration date warning to the members whom memberships are about to expire 2 weeks prior the due date.
Req.13	Keep track on the members' complains and suggestions.
Req.14	Keep track on the member's reservations for upcoming events.

b. Enumerated nonfunctional Requirements

3. Functional Requirements Specification

a. Stakeholders

Software Engineer

- Responsible for applying the user's requirements.
- Responsible for system development.
- Responsible for support and deployment.

Staff

- Responsible for all the Sales process.
- Responsible for organizing events.
- Responsible for creating customers' membership.
- Responsible for customers lending.
- Responsible for customers' reservations.
- Responsible for customers complains.

Administrator

1. Responsible for adding new staff members.
- Responsible or removing staff members.
- Responsible for updating the staff data.

Manager

- Views all the data.

b. Actors and Goals

Software Engineer

- Gain experience in building software systems and its development.

Staff

- Easily add members on the system.
- To be able to organize events easier and more properly.
- Manage the sales more proficiently.
- Easily know what isn't in stock.

Administrator

- To be able to reach all the data concerning the staff easily.

Manager

- To be able to monitor all data easily.

c. USE CASES

I. Use case Description

Use Case ID: 01

Title: Register Employee

Actor: HR Administrator

Stakeholders and their needs:

- Library's HR Administrator; needs to register the data of the new hired employees to the data base of the library's software system.
- Library's newly hired employees; needs to be registered on the system as a new staff member.

Trigger: A new employee is hired in a specific job description in the library.

Pre-conditions:

- The system must be active on the HR administrator's PC.
- All the data are provided in the new employee's resume to be registered on the system.

Post-conditions:

- The data base is updated, and the new data are added to the system.
- The newly hired employee is provided with his own access key to the system.

Basic flow:

1- A new employee joined the Library's staff members.

2- The Administrator takes the employee's information and enter them in the system.

3- The Administrator inters the data provided in the employee's resume.

4- The data base is updated, and the new data are added to the system.

5- The newly hired employee is provided with his own access key to the system.

Extensions:

The resume provided by the newly hired employee is incomplete, then the Administrator does not complete the registration until he is provided with the complete information which need to be added to the system.

Author: Solwan Shokry Ahmed- Rana Mohamed Hussein.

Use Case ID: 02

Title: Register Account

Actor: Staff

Stakeholders and their needs:

- Library's visitor needs to register a new account to be able to use the library's books and resources.
- The library's staff needs the new visitor's data to insert them to the system's data base to keep track on all the library's properties that the new visitors interact with and to manage all the possible services to be provided to said new visitor.

Trigger: Staff needs to register account for a visitor.

Pre-conditions:

- The system must be active on the staff member's PC.
- A new visitor visits the library and applies for a new data registration.

Post-conditions:

- The visitor's data is entered to the system's data base.
- The visitor has access to the library's resources.

Basic flow:

1- A new visitor visits the Library and asks the reception employee for an account.
2- The Staff member takes data from the visitor which is his name, ID, phone number and address.
3- The Staff registers an account for the visitor and give him/her the username and password.
4- After the visitor is provided with his account access information**,** he is free to read or borrow any of the available resources of the library, unless it is limited to people with membership.

Extensions:

1- The Visitor wants to upgrade his/her account to have a membership, they should be registered in the Library's system as visitors first.
2- The Staff upgrades the visitor's account after taking the visitor's email, education state and his/her fields of interest.
3- The Staff registers the membership account and gives the member a new username and password for his membership account.

Author: Rana Mohamed Hussein, Solwan Shokry Ahmed.

Use Case ID: 03

Title: **Register Membership**

Actor: Staff

Stakeholders and their needs:

- Library's visitor needs to register a membership to be able to unlimitedly use the library's books and resources and to have the privileges provided for members of the library.
- The library's staff needs the visitor's data to insert them to the system's data base to keep track on all the library's properties that the new visitors interact with and to manage all the possible services to be provided to said new member.

Trigger: A visitor wants to register for a membership.

Pre-conditions:

- The member already has a visitor account, and his basic information exists in the system's data base.
- The staff has all the member's data required to complete the membership registration.

Post-conditions:

- The member's account is registered in the system's data base.
- The member is provided with his membership number.

Basic flow:

1. The visitor asks the staff to upgrade his account.
2. The staff takes the visitor's membership additional data such as his E-Mail, his educational status, and his fields of interests.
3. The staff upgrades the visitor's account to a member account.
4. The staff provides the member with his membership number.
5. The account is now upgraded, and the new member has more privileges.

Extensions:

1. The visitor does not have an account registered in the system in the first place.
2. The staff then asks for all the information to build the member account, including the basic visitor's account data.
3. The member already has a registered member account on the system**,** but his account is expired.
4. The staff only processes the payment matters and renews the account for him without changing his major data, and then the account's status is reactivated**,** and all his privileges are provided to him.

Author: Solwan Shokry Ahmed- Sarah Gamal El deen Mohamed

Use Case ID: 04

Title: **Checking Books and Resources Availability**

Actor: Staff

Stakeholders and their needs:

Trigger: Staff needs to check books/resources availability.

Pre-conditions:

- The books and resources data base are continuously updated.
- Each book or resource borrowed from the library must be recorded on the system.

Post-conditions:

- No changes are made in the system.
- All data remain the same.

Basic flow:

1. A visitor/member visits the Library to read a book/resource.
2. The Staff checks the availability of the given book name.
3. If the book/resource is available**,** the staff informs the user and tell him the book section.

Extensions:

1. A visitor/member visits the Library to read a book/resource.
2. The Staff checks the availability of the given book name.
3. If the book/resource is not available**,** the staff informs the visitor.
4. If the book is not available and the visitor is a member the staff emails him/her when the book is available.

Author: Rana Mohamed Hussein, Solwan Shokry Ahmed.

Use Case ID: 05

Title: **Lending Books and Resources**

Actor: Staff

Stakeholders and their needs: The system is required to manage the lending of books and resources process by recording who borrowed the books, their details, and the end date of lending process

Trigger: The staff employee is asked to lend books to a member.

Pre-conditions:

- The person requesting to borrow from the library must be a registered member on the library system

Post-conditions:

- The system listed the lending process, the borrowed books who borrowed them and the date in which it was borrowed
- The system updated the amount of the available books in the library
- The system is ready to accept another lending process

Basic flow:

1. A member visits the Library to borrow a book/resource.
2. The staff must validate the membership.
3. The Staff must check the availability of the given book name first.
4. If the book/resource is available, the staff sets a duration for the lending which is 2 weeks.
5. The amount of that book availability is decreased by 1.
6. The staff hand in the member/employee the book/resource.

Extensions:

1. The person requesting to borrow a book is not registered or not a valid member.
2. the employee will ask this person to register.
3. The book required to be borrowed is not found.
4. the employee will cancel the lending process and apologizes for the inconvenience.

Author:

Rana Mohamed Hussein – Sarah Gamal El deen Mohamed

Use Case ID: 06

Title: **Newly Arrived Books and Resources Update**

Actor: **Staff**

Stakeholders and their needs: It is required to update the available books and resources in the library

Trigger: The arrival of new books and resources.

Pre-conditions:

- The already available material in the library is listed in the system and categorized by their book name, author, and genre.

Post-conditions:

- The system is updated with the amount of each of the available material and their copies in the library.

Basic flow:

1- Staff adds the name, author, genre, amount, and the date of arrival of the new book/resource.

2- The number of this available book is increased by one

3- The System's data base is now updated with the new book.

4- If the book/resource is available, the staff informs the user and tell him the book section.

Extensions:

1- The newly arrived book is the first copy in the library.

2- The employee then will register the book and its info in the system.

3- The same flow will occur.

Author:

Sarah Gamal El deen Mohamed, Rana Mohamed Hussein

Use Case ID: 07

Title: **Checking available Computers**

Actor: Staff

Stakeholders and their needs: The system is required to keep a record of the available computers, the computers in use and the duration of each session.

Trigger: Staff needs to check availability of an internal computer for a visitor/member.

Pre-conditions:

- The total number of the computers in the library is registered previously in the system.

Post-conditions:

- The system is updated with the computers in booked sessions and their duration and the number of the available computers in specific time.

Basic flow:

1- Visitor/member visits the Library and ask for an internal computer in a specific time.

2- The staff checks the availability of an internal computer in this time.

3- If an internal computer is available in this time**,** the staff informs the user of the available computer number.

Extensions:

1- All the computers are not available.

2- The employee searches for the nearest time a computer will be available and informs the person who requested a computer session.

Author:
Rahma Mohamed Makram - Sarah Gamal El deen Mohamed

Use Case ID: 08

Title: **Booking Internal Computer session**

Actor: Staff

Stakeholders and their needs: The system is required to manage booking internal computer sessions by recording who booked, their details and the session time

Trigger: Staff needs to book an internal computer for a visitor/member.

Pre-conditions:

- The person booking the session must be registered in the system and the number of available computers is listed.

Post-conditions:

- The system listed the booked computers, who booked them and the session time.
- The system is ready to accept another Booking process
- A computer session has started, and the new number of available computers is updated.

Basic flow:

1- The Staff member checks the internal computers availability on the system.

2- If there are empty computers, the staff books a session for the visitor/member

3- The staff employee registers the session duration on the system

4- The Staff gives the visitor the computer number to use.

Extensions:

1- The Staff member checks the internal computers availability on the system.

2- There are no available computers, the staff checks the nearest session to end on the system.

3- The Staff informs the visitor to wait.

Author:
Sarah Gamal El deen Mohamed – Rana Mohamed Hussein – Rahma Mohamed Makram.

Use Case ID: 09

Title: **Managing Events and Reservations**

Actor: Staff

Stakeholders and their needs: Staff need to manage events and reservations for members

Trigger: A member wants to reserve an event.

Pre-conditions:

- The person requesting to reserve an event in the library must be a registered on the library system.

Post-conditions:

- The system is updated with the upcoming events and reservations.

Basic flow:

1. A member visits the Library to reserve an event.
2- The staff must validate the membership
3- The Staff must check the availability of seats.
4- If available, the staff reserve the event for the member.

Extensions:

1. All the tickets of the event are reserved.
2. The staff informs the member that the event's tickets are sold out.

Author:

Rahma Mohamed Makram- Solwan Shokry Ahmed

Use Case ID: 10

Title: **Send Membership Expiration Date**

Actor: Staff, E-mail System

Stakeholders and their needs:
It is required to send an e-mail warning member whose memberships are about to end in two weeks' time, to renew the membership before the expiration date.

Trigger: There are only 14 days left before the membership of a member expires.

Pre-conditions:
- The date of the membership subscription, the membership duration, and the due date in which the membership will expire must be listed previously in the system.

Post-conditions:
- The system is ready to trace the due date of the next membership.

Basic flow:
1- The Library system detects that the expiration date is after two weeks from today's date.
2- The Library system sends a warning notification to the library's staff employee with the details.
3- The employee sends the warning e-mail to the member via the email system.

Extensions:
1- The calendar of the library's system is not correctly calibrated.
2- The e-mail is not sent due to poor network connection.
3- The employee did not see the warning notification; therefore, the warning e-mail is not sent.

Author:
Sarah Gamal El-Deen Mohamed

Use Case ID: 11

Title:

Send to Members Book recommendations according to their field of interest

Actor: : Staff , Email System

Stakeholders and their needs: send an email to the member with set of most trended books according to their interests.

Trigger: —

- Members set their fields of interests.
- Arrival of new books.

Pre-conditions:

- The member asked for a book that was not available, and now it's on library books list.
- There are newly arrived books with the same author or with the same genre.

Post-conditions:

- The member is ready to read or borrow the book

Basic flow:

1- Member set his field of interest.

2- The library system detects his field of interest and the category of books he usually reads or borrows.

3- If there is a book that matches his interests, the library system sends a recommendation mail to the member.

Extensions:

1- Not all the members had fields of interest assigned to them on the system.

2- The employee excludes them from the emails and sets a reminder to ask them about the missing information in their accounts.

Author:
Rahma Mohamed Makram- Solwan Shokry Ahmed.

Use Case ID: 12

Title: **Respond to Members suggestions and complains use case**

Actor: Staff

Stakeholders and their needs: they want to get the visitor/member feedback to fix and improve what member complains and give him the best service

Trigger: — A member submitted a suggestion or a complain on one of the library's internal computers.

Pre-conditions: —

- The person who made suggestion or complain in the library must be registered on the library system.
- The member who will be writing a complain or suggesting a recommendation must be using one of the library's internal computers.
- The submitted feedback appears on the staff's interface of the system.

Post-conditions:

- The system is updated with member/visitor complains and suggestions.
- The staff has taken an action regarding the feedback.
- A reply is sent to the member using the e-mail system.

Basic flow:

1. A member visit a library
2. The member must login in with his username and password to the System.
3. The member is asked if there is any suggestion to improve the service
4. If yes, he should state his complains or suggestion on the internal computer.
5. The feedback is directed to the staff's interface of the system.
6. The staff reviews the feedback and takes an appropriate action regarding it.
7. The staff replies to the member who assigned the feedback using the e-mail system.

Extensions:

1- There is no specified e-mail in the feedback form sent to the system.
2- The staff takes an action, nonetheless.

Author:

Rahma Mohamed Makram- Solwan Shokry Ahmed

Use Case ID: 13

Title: **Login**

Actor: Staff, visitor, and member

- **Stakeholders and their needs:** The need of registration the data of the new hired employees to the data base of the library's software system.
- The need of registration the data of new visitor /members to the data base of the library's software system
- Staff, members have access to library system by their username and password

Trigger: A person needs to access the library system

Pre-conditions:

- The person who is signing-in to the system is already registered and has an account.
- The username and password given to him matches the ones recorded on the system's data base.
- The account is valid and not expired.
- An internal computer is available and ready for them to sign in.

Post-conditions:

- The system is opened on the used device and is ready to be operated on.

Basic flow:

1- A person, either a staff, a visitor or a member wants to use the library's system.
2- A staff goes to his desk and operates the device containing the software.
3- A visitor or a member checks for internal computers availability.
4- If there are any available computers, they use them and operate the device.
5- The system screen shows a login boxes to fill.
6- They fill the boxes with their username and passwords.
7- The system checks the entered data.
8- If the data are correct, the person is authenticated to enter the system.
9- The system opens and is ready to be operated on.

Extensions:

1- The user enters an incorrect data, or the account is expired.
2- The system shows an error message specifying the problem and asking for a valid username and password.
3- If the user's data are wrong, they rewrite them.
4- If the user's account is expired, he goes back to the staff to register or renew their account.

Author: Rahma Mohamed Makram- Solwan Shokry Ahmed.

Use Case ID: 14

Title: **Search E-Books and e-Resources**

Actor: Member

Stakeholders and their needs:
The member to be able to search through the Available eBooks on the internal computer of the library.

Trigger: A member wants to read an eBook/resource on the internal computers

Pre-conditions:
The member has booked a computer session previously.

Post-conditions:
The member finished reading and closed the eBook.

Basic flow:

1- The visitor/member must request the employee to book a session on the internal computer.
2- The visitor/member must login in with his username and password to the System.
3- The member may now search for the eBook/resource he/she wants to read on the System.
4- The visitor/member types the book and author name to search for the system to search for it.
5- The System allows the visitor/member to read the book if found.

Extensions:

1- The visitor/member request the employee to book a session on the internal computer.
2- The visitor/member login in with his username and password to the System.
3- The visitor/member types the book and author name to search for the system to search for it.
4- The eBook/resource was not found.
5- The visitor/member search for another book to read.

Author:
Rana Mohamed Hussein – Sarah Gamal El deen Mohamed

Use Case ID: 15

Title: **Write suggestions or complains**

Actor: Member

Stakeholders and their needs: improvement of library system with respect to member complains and suggestions

Trigger: A member wants to write suggestions or complains through internal computers

Pre-conditions:

- A person who writes a complain might be registered as a member.
- The member is using an internal computer.
- The member has a valid membership.
- The member has logged-in the system.

Post-conditions:

- The system is updated with members complains and suggestions and ready to improve the library's services.

Basic flow:

1- The Member requests the employee to book a session on the internal computer.
2- The member logs-in in with his username and password to the System.
3- The member may now add his/her suggestion or complains to the System.
4- The feedback is directly forwarded to the staff's interface of the system.
5- The staff starts his duty of managing the received feedback.

Extensions:

1- The member's account has expired.
2- No authentication is given to the member to access the system.
3- The member has the freedom to renews his membership.

Author

Rahma Mohamed Makram- Solwan Shokry Ahmed- Rana Mohamed Hussein.

Use Case ID: 16

Title: **Reserve an event**

Actor:
Member

Stakeholders and their needs:
The member can reserve an event manually through the system of the internal computer.

Trigger: The member wants to reserve an event.

Pre-conditions:

- The member is logged in his account and the event's details is previously registered on the system.

Post-conditions:
A reservation is made by the member and this reservation is recorded on the system.

Basic flow:

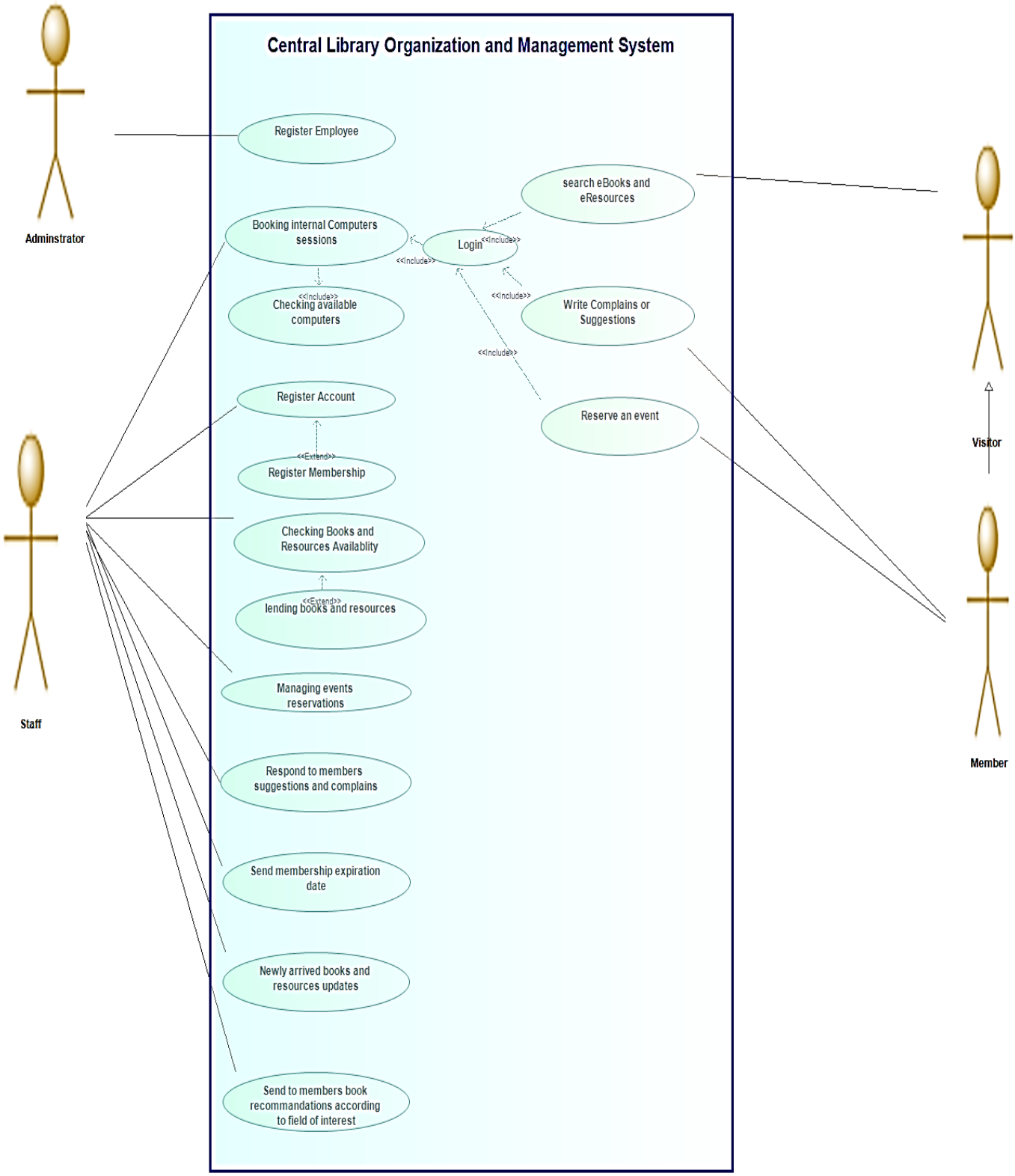
1- The member logs in his account on the computer system.
2- The member checks the available events.
3- The member reserve the desired event.

Extensions:

1- The member cannot log into his account.
2- The event's tickets are all sold-out.

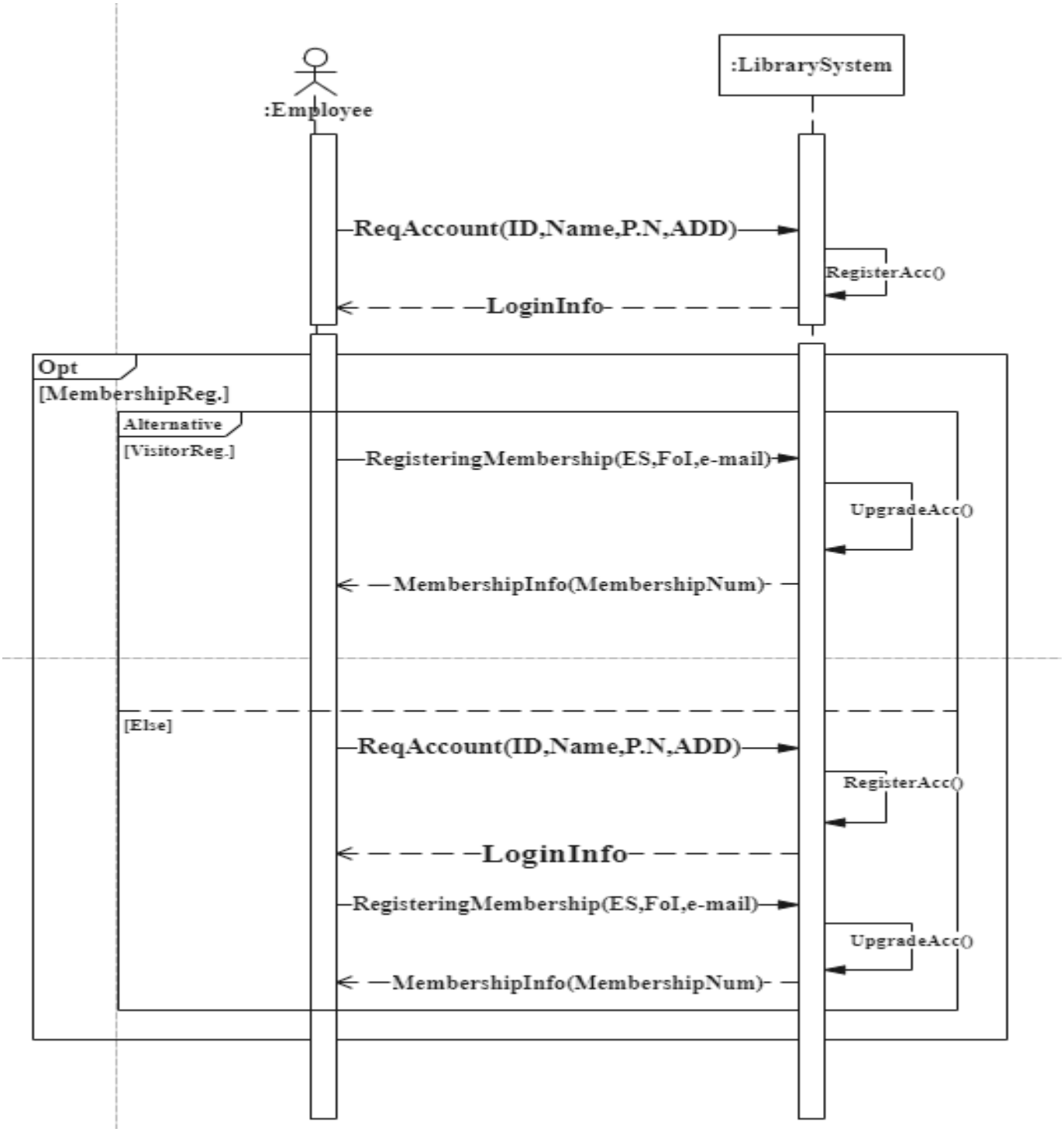
Author:
Sarah Gamal El deen Mohamed – Solwan Shokry Ahmed

II. Use Case Diagram

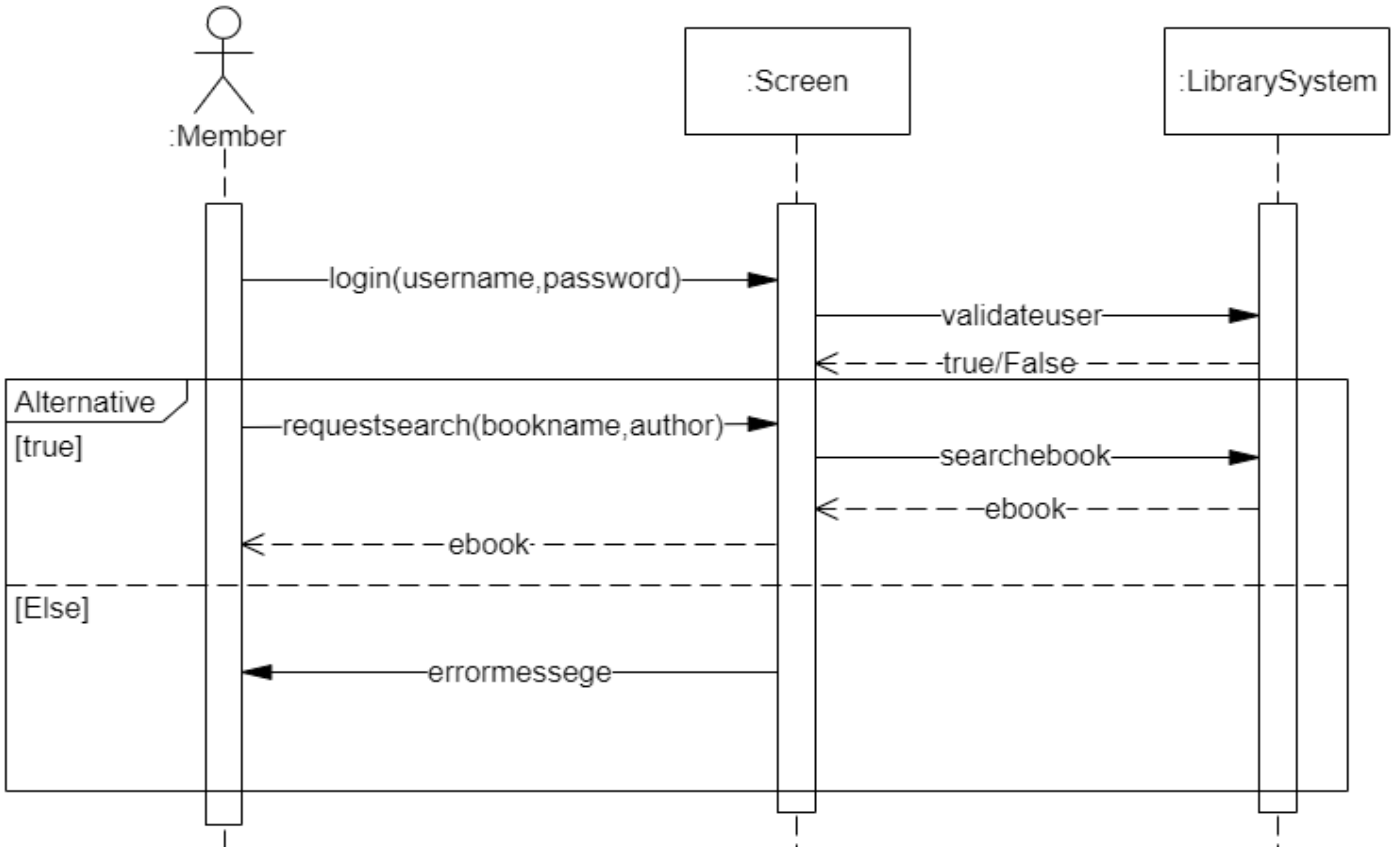


d. System Sequence Diagrams

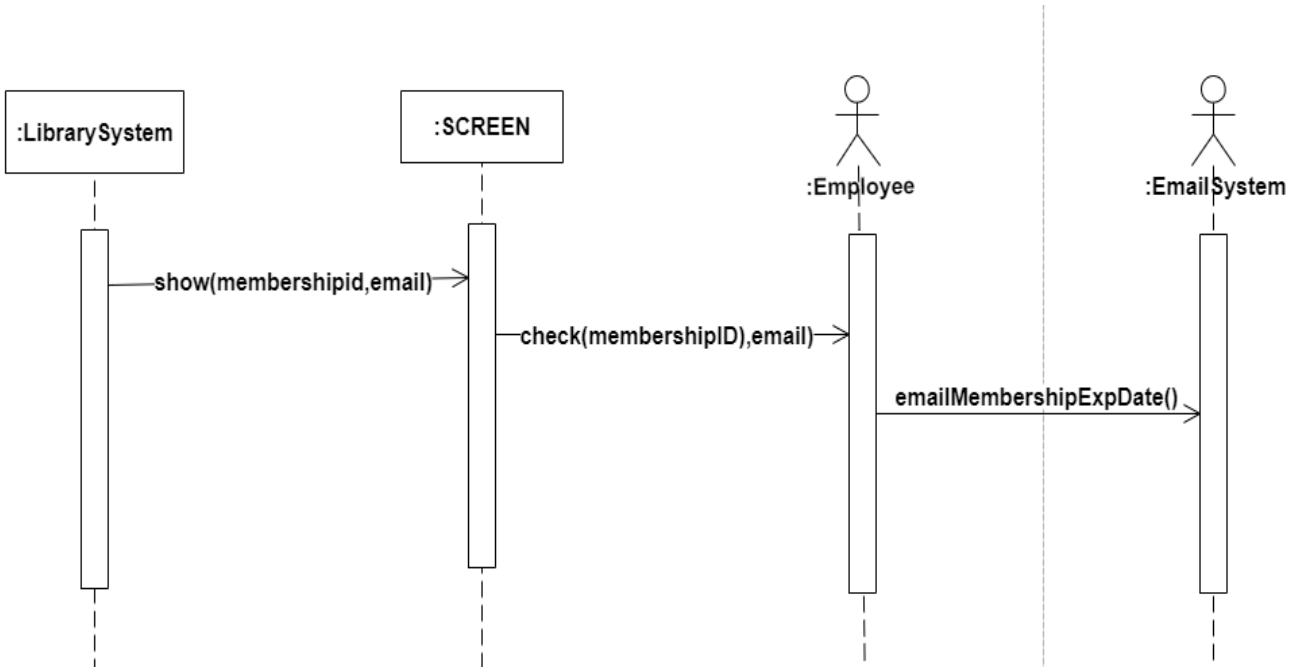
1- Sequence Diagram for Register Visitor Account use case:



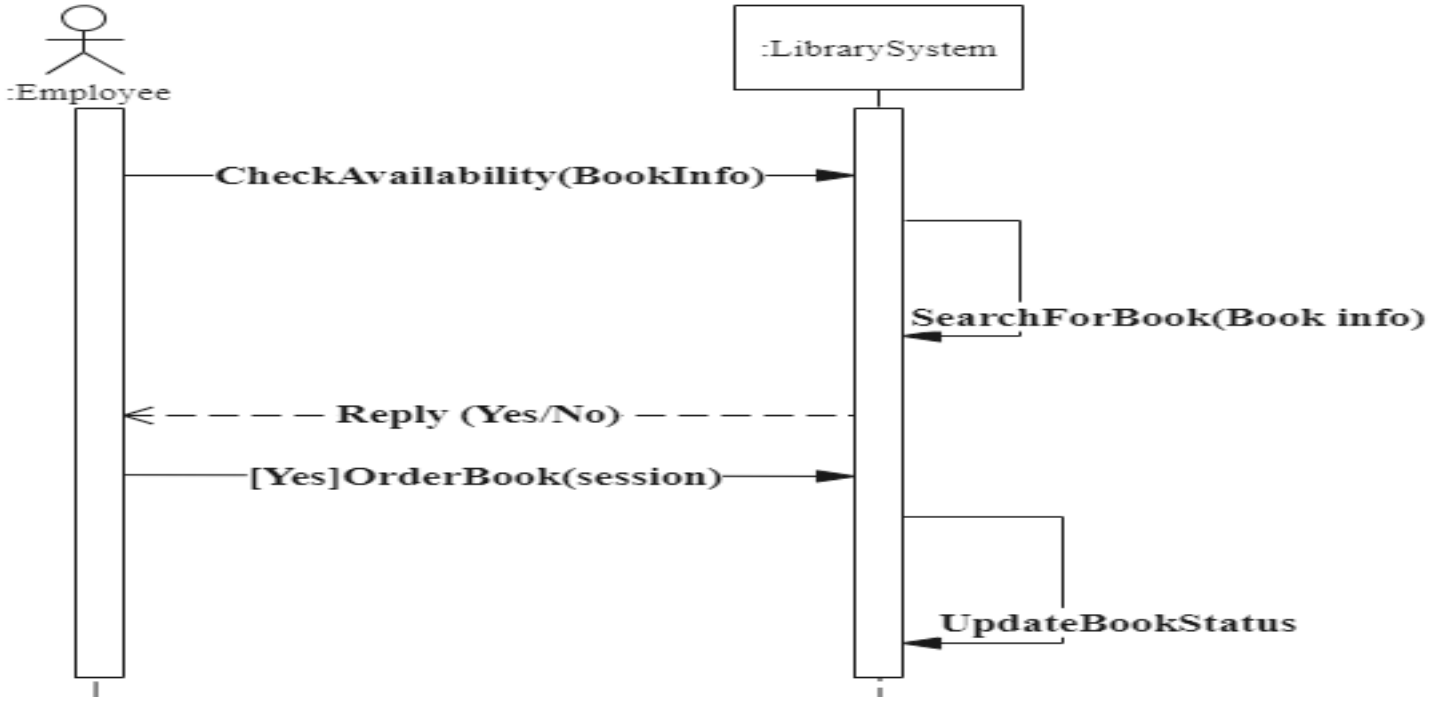
2- Sequence Diagram for searching eBooks:



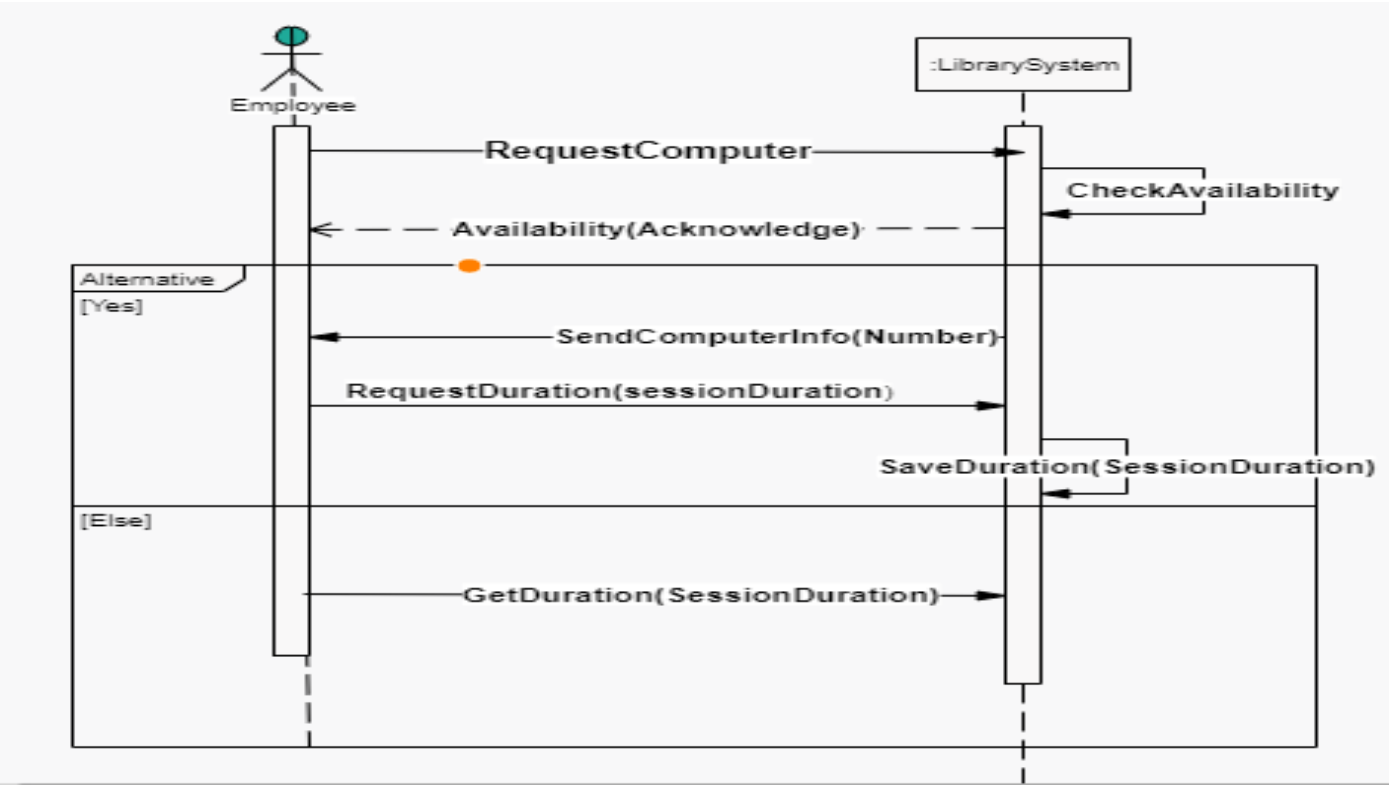
3-Sequence Diagram for Sending Membership Expiration Date use case:



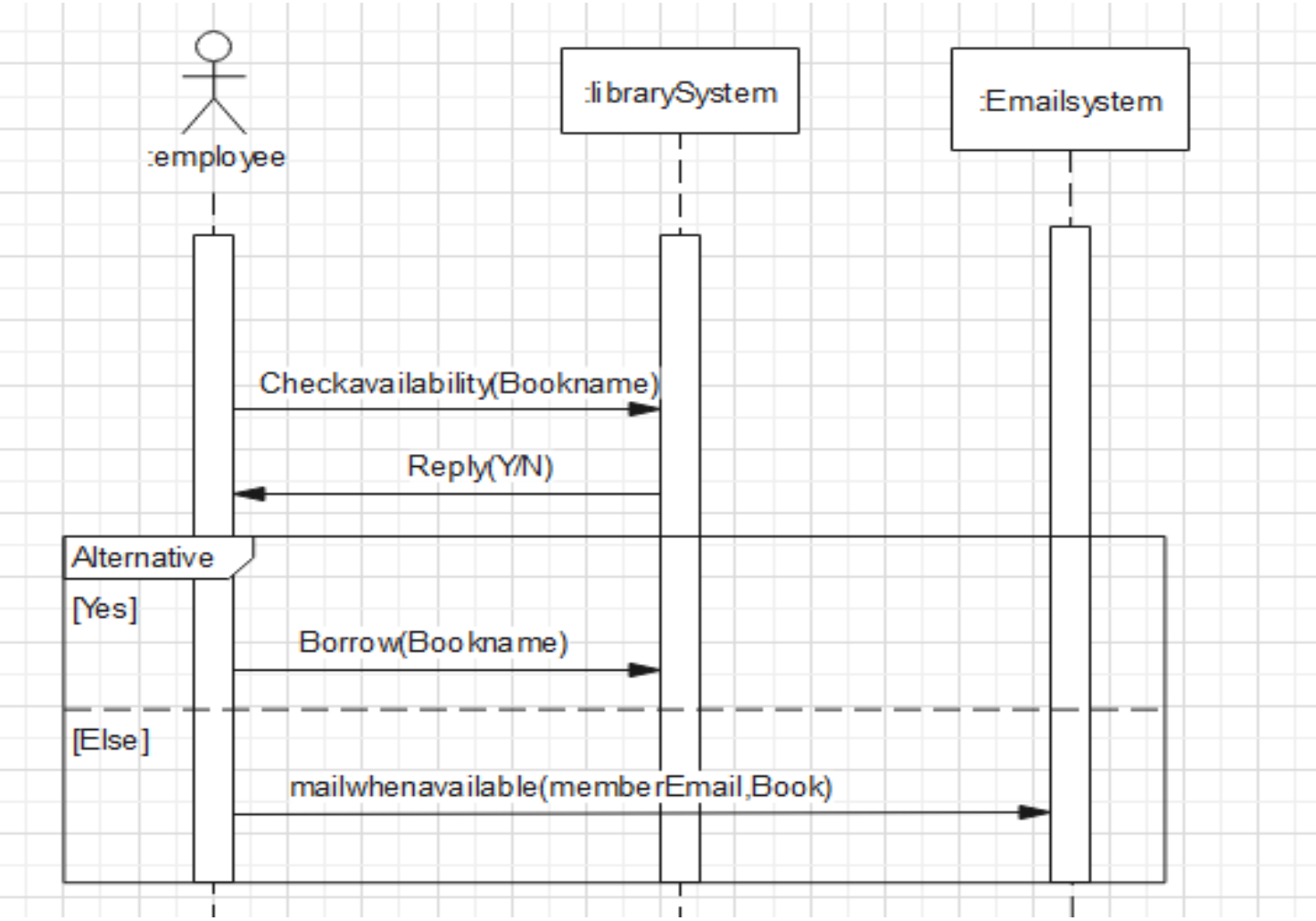
3-Sequenc Diagram for Checking Book and Resources Availability use case:



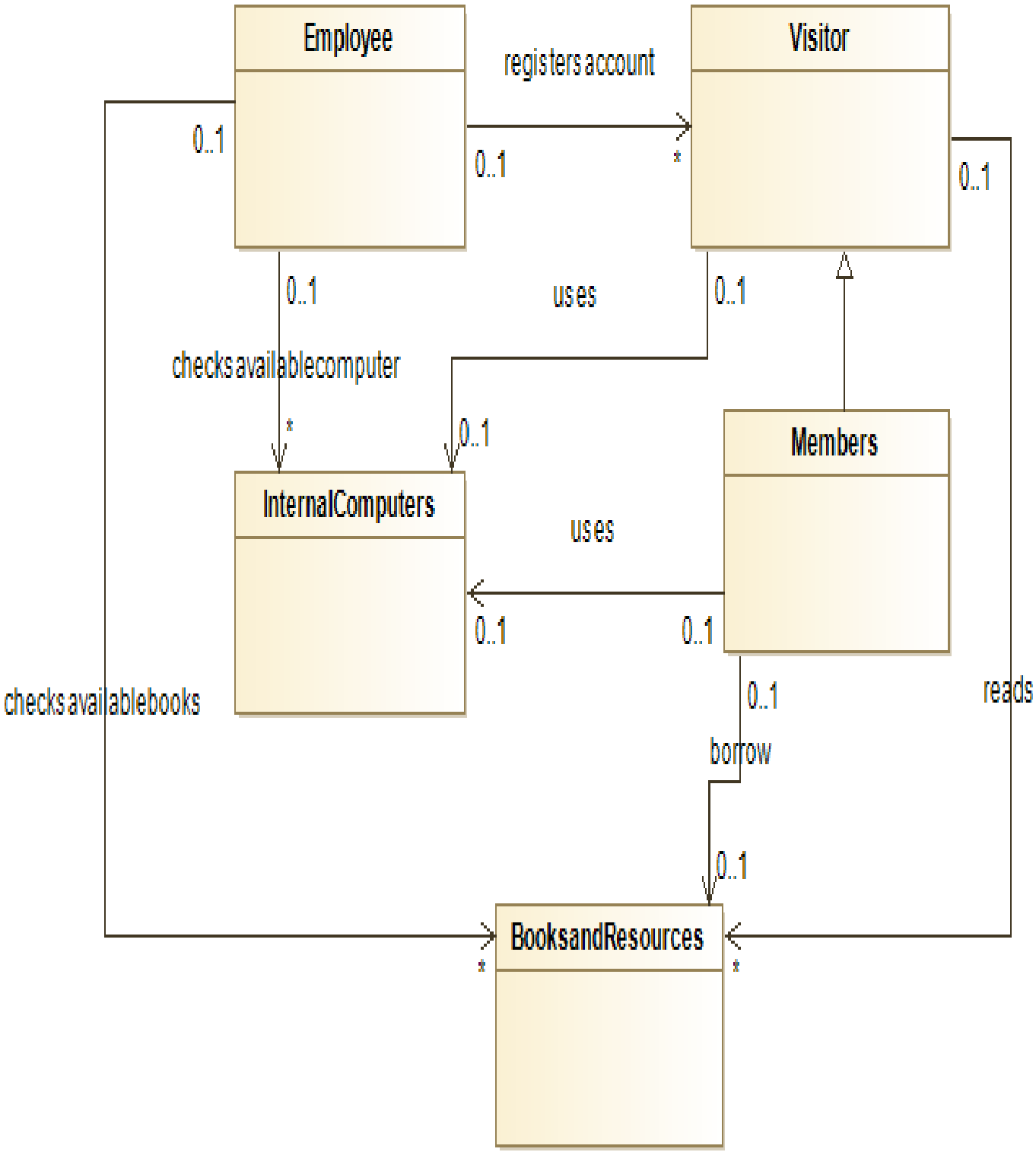
4- Sequence Diagram for Booking Internal Computers:

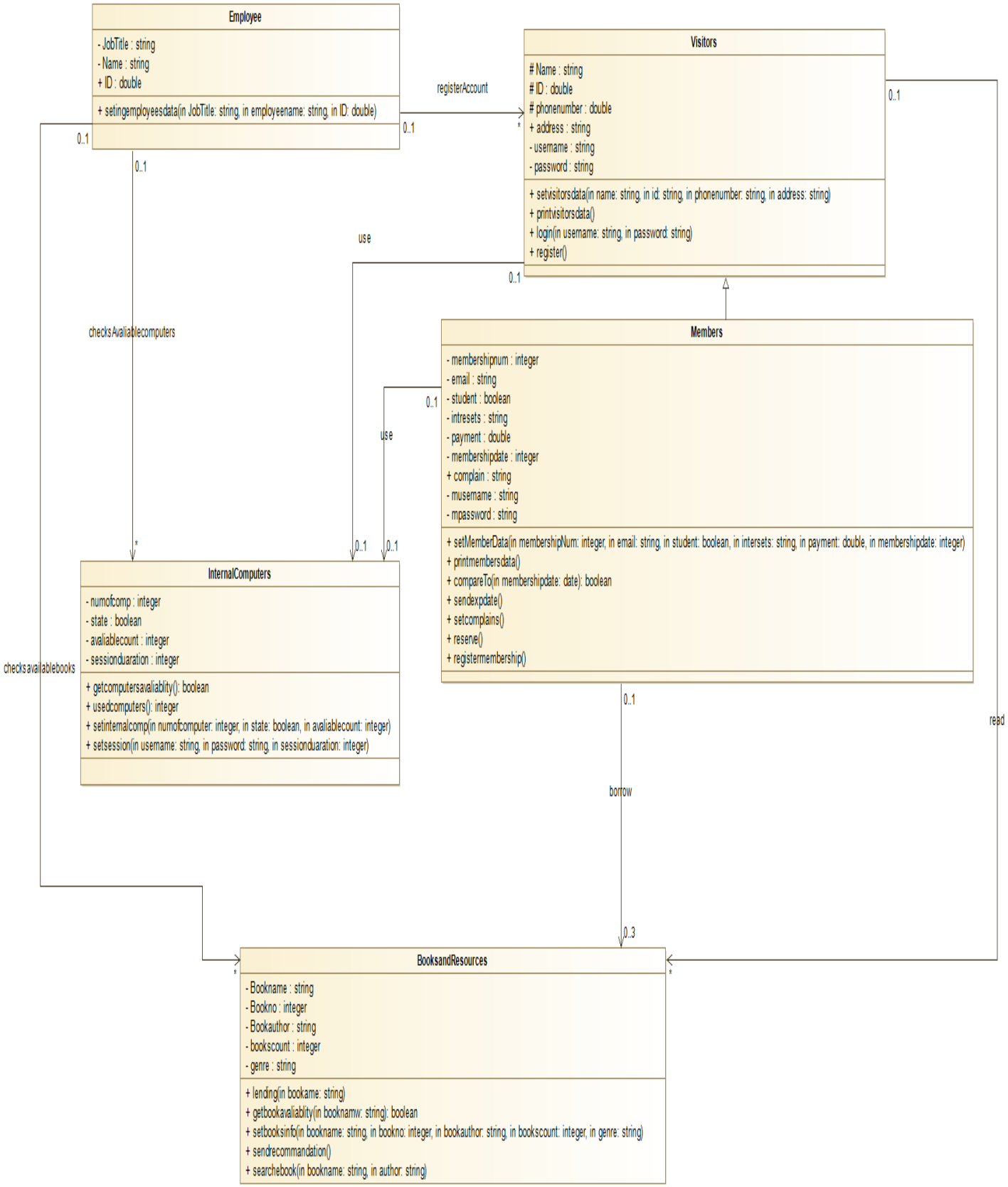


5- Sequence Diagram for Lending Books and Resources:



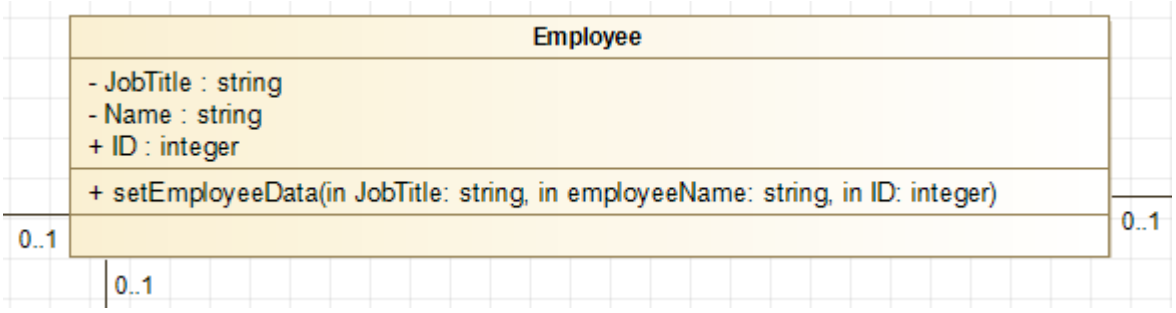
e. Class Diagram and Interface Specification




Employee
- JobTitle : string
- Name : string
+ ID : double
+ setingemployeesdata(in JobTitle: string, in employeename: string, in ID: double)
registerAccount
Visitors
Name : string
ID : double
phonenumber : double
+ address : string
- username : string
- password : string
+ setvisitorsdata(in name: string, in id: string, in phonenumber: string, in address: string)
+ printvisitorsdata()
+ login(in username: string, in password: string)
+ register()
use
checksAvaliablecomputers
Members
- membershipnum : integer
- email : string
- student : boolean
- intresets : string
- payment : double
- membershipdate : integer
+ complain : string
- musername : string
- mpassword : string
+ setMemberData(in membershipNum: integer, in email: string, in student: boolean, in intersets: string, in payment: double, in membershipdate: integer)
+ printmembersdata()
+ compareTo(in membershipdate: date): boolean
+ sendexpdate()
+ setcomplains()
+ reserve()
+ registermembership()
use
InternalComputers
- numofcomp : integer
- state : boolean
- avaliablecount : integer
- sessionduaration : integer
+ getcomputersavaliablity(): boolean
+ usedcomputers(): integer
+ setinternalcomp(in numofcomputer: integer, in state: boolean, in avaliablecount: integer)
+ setsession(in username: string, in password: string, in sessionduaration: integer)
checksavailablebooks
borrow
read
BooksandResources
- Bookname : string
- Bookno : integer
- Bookauthor : string
- bookscount : integer
- genre : string
+ lending(in bookame: string)
+ getbookavaliablity(in booknamw: string): boolean
+ setbooksinfo(in bookname: string, in bookno: integer, in bookauthor: string, in bookscount: integer, in genre: string)
+ sendrecommandation()
+ searchebook(in bookname: string, in author: string)

Class diagram description:

a) Employee Class:



Attributes in Employee Class:

1. **JobTitle:** its data type is String, and its access Modifier is private.
2. **Name:** its data type String, and its access Modifier is private.
3. **ID:** its data type is integer, and its access Modifier is public.

Operations in Employee Class:

1. **setEmployeeData:**

- It is a public function that does not return any parameter, it takes 3 parameters as an input JobTitle, Employee name and ID. This operation is used to set the attributes of class employee.

Relationships with other Classes:

1. With Visitors Class:

Association relation: The employee **registerAccount** for the visitor.

Multiplicity: 0..1 employees can register more than one visitor.

2. With BooksAndResources Class:

Association relation: The employee **checksAvailableBooks** of the books and resources.

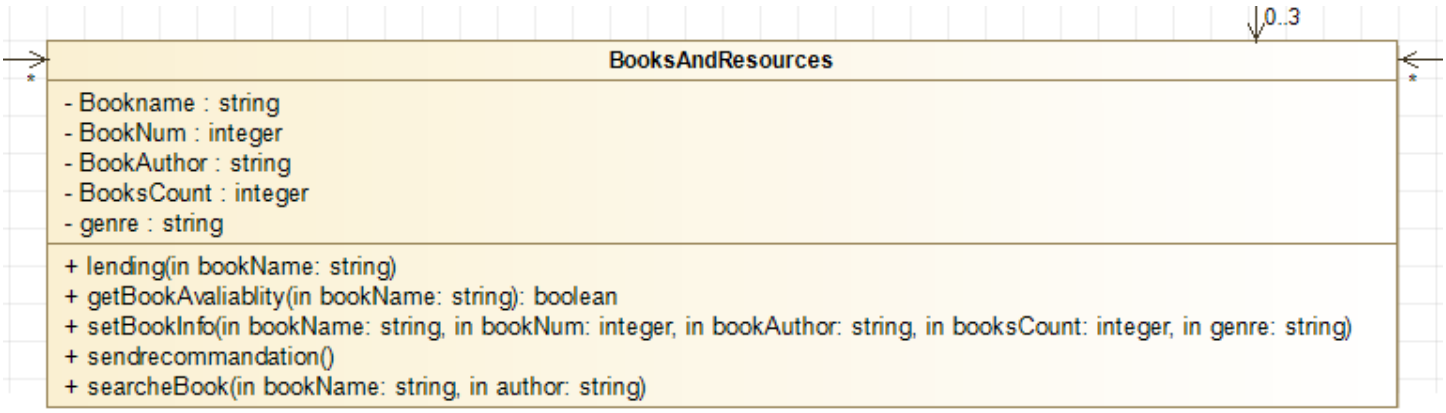
Multiplicity: 0..1 employees can check more than one book and resource.

3. With InternalComputers Class:

Association relation: The employee **checksAvailableComputers** for the books and resources.

Multiplicity: 0..1 employees can check more than one book and resource.

b) BooksAndResources Class:



Attributes in BooksAndResources Class:

1. **Bookname:** its data type is string, and its access modifier is private.
2. **BookNum:** its data type is integer, and its access modifier is private.
3. **BookAuthor:** its data type is string, and its access modifier is private.
4. **BooksCount:** its data type is integer, and its access modifier is private.
5. **genre:** its data type and string and its access modifier is private

Operations in class books and resources:

1. **getBookAvailability:** a public function that takes a Bookname as a parameter and its return type is Boolean, it checks the availability of book by checking the number of books (BooksCount), if they are more than 0, then book is available (returns true) if not available it returns false.
2. **lending:** its public function that check getBookAvailability function if it returns true, BooksCount attribute decreases by 1, the function return BooksCount as integer.
3. **setBookInfo:** it takes Bookname, BookNum, BookAuthor, BooksCount and genre as parameters, and it has no return type, it used to set the attributes of class BooksAndResources.
4. **SearcheBook: i**t takes the Bookname & BookAuthor as parameters. This function is used to search for the eBooks on the internal computers or on the internet.

Relationships with other Classes:

a. With Visitors Class:

Association relation: The visitor **can read** a book/resource.

Multiplicity: a book may be read by 0..1 visitor.

b. With Members Class:

Association relation: The member **can borrow** a book/resource.

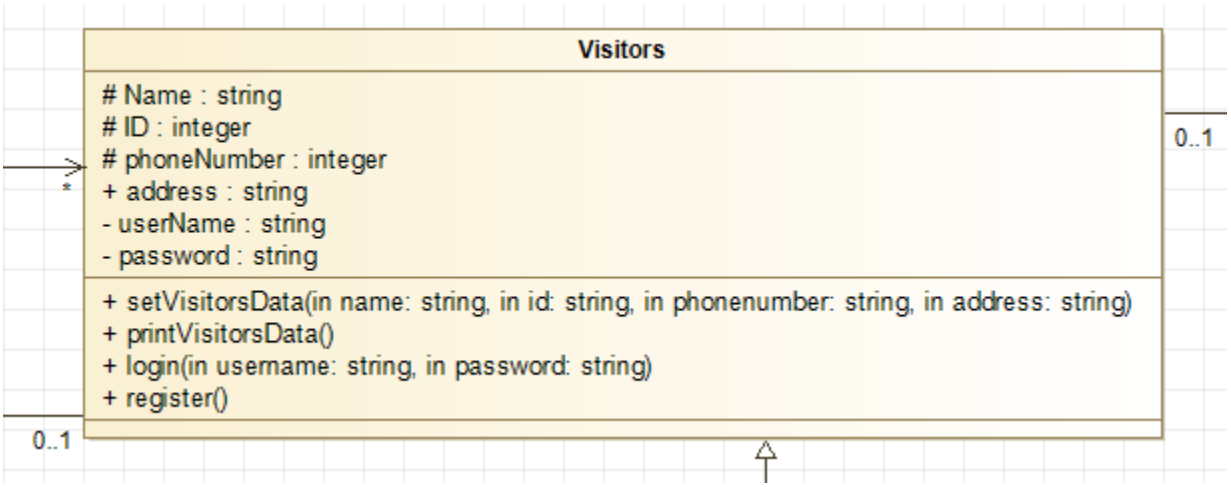
Multiplicity: a book may be borrowed by 0..1 member.

c. With Employee Class:

Association relation: A book will be checked for availability by the employee.

Multiplicity: a book can be checked by 1 or more employees.

c) Visitors Class:



Attributes of Visitors Class:

4. **Name:** its data type is string and its access modifier is protected.
5. **ID:** its data type is integer and its access modifier is protected.
6. **phoneNumber:** its data type is integer and its access modifier is protected.
7. **address:** its data type is string and its access modifier is private.

8. **userName:** its data type and string and its access modifier is private.
9. **password:** its data type and string and its access modifier is private.

Operations of Visitors Class:

1. **setVisitorsData:** it is a public function that has no return .it takes the name, ID, phoneNumber, and address as parameters, this function is used to set visitors data.
2. **printVisitorsData**: it is a public function that takes no parameters and has no return, it prints the visitor's data.
3. **login:** it is a public function that takes username and password of visitor as parameters and has no returns, this function asks the visitor to add username and password to give him access to internal computers, read eBooks and access to the internet.
4. **register**: it is a public function that has no parameters and has no return, it makes a register account for each visitor.

Relationships with other Classes:

a. With Employee Class:

   Association relation: the visitor **gets registered** by employee.

   Multiplicity: a visitor may be registered by 0..1 employee.

b. With InternalComputers Class:

   Association relation: the visitor **can use** an internal computer.

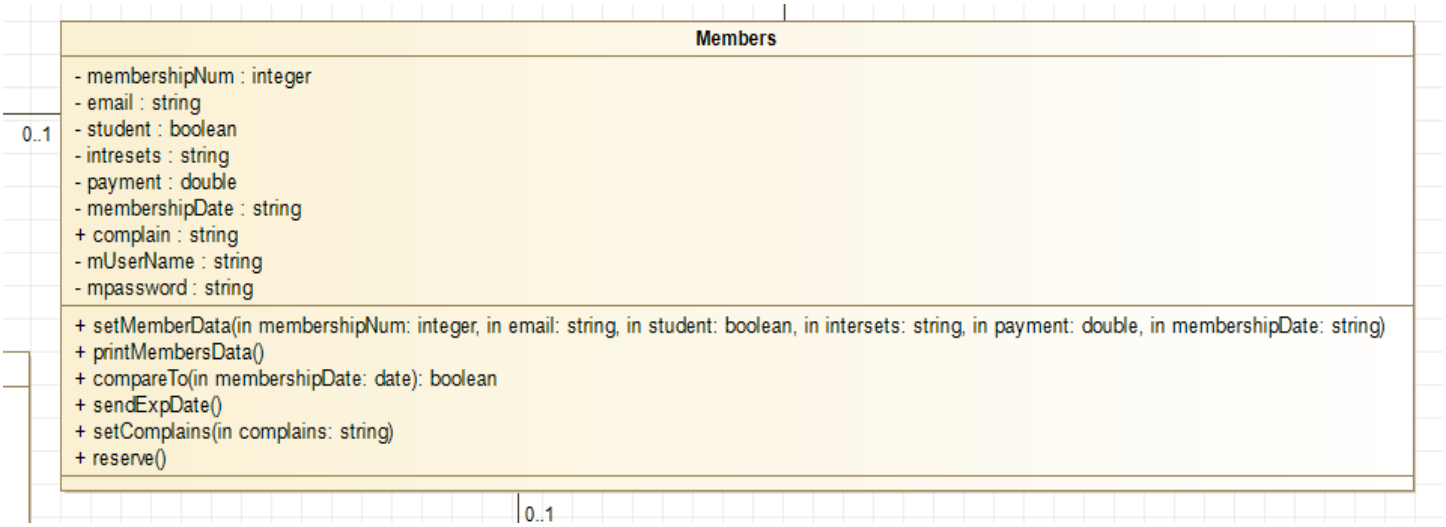
   Multiplicity: a visitor may use by 0..1 computers.

c. With BooksAndResources Class:

   Association relation: a visitor can **read** books.

   Multiplicity: a visitor can read 1or more books.

d) Members Class:



Attributes of Members Class:

1. **membershipNum:** its data type is integer and its access modifier is private.
2. **email:** its data type is string and its access modifier is private.
3. **student:** its data type is boolean, and its access modifier is private.
4. **interests:** its data type is string and its access modifier is private.
5. **payment:** its data type is double and its access modifier is private.
6. **membershipDate:** its data type is string and its access modifier is private.
7. **complain:** its data type is string and its access modifier is public.
8. **mUserName:** its data type is string and its access modifier is private.
9. **mpassword:** its data type is string and its access modifier is private.

Operations of Members Class:

1. **setMemberData:** it's a public function that takes membershipNum, email, student, interests, payment and membershipDate as parameter, it has no return, it used to set member data.
2. **printMembersData:** it's a public function that has no return and takes no parameters, it used to print member data.

3. **compareTo:** it's a public function that takes the membership date as a parameter and compare it with current date, it returns Boolean (true) when it completes one year.
4. **sendExpDate :** it's a public function that takes no parameters , it sends a message to the member two weeks before expiration date to update his membership.
5. **setComplains:** it's a public function that takes no parameters and no return, the function ask the member to set his complains
6. **reserve:** it's a public function that takes no parameters and no returns, this function for reservation of upcoming events

Relationships with other Classes:

With Visitors Class:

a. Generalization relation: Members Class inherits Visitors Class.
b. With Employee Class:

   Association relation: the member **getsupgraded** by employee.

   Multiplicity: a member may be registered by 0..1 employee.
c. With InternalComputers Class:

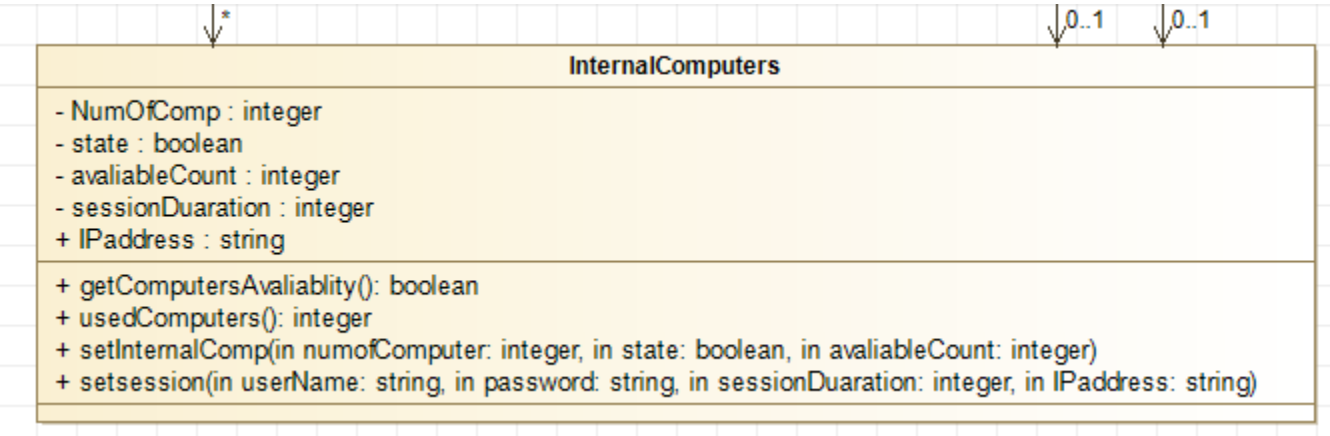
   Association relation: the member **can use** an internal computer.

   Multiplicity: a member may use by 0..1 computer.
d. With BooksAndResources Class:

   Association relation: a member can **borrow** books.

   Multiplicity: a member can borrow 1 or more books.

e) InternalComputers Class:



Attributes of InternalComputers Class

1. **NumOfComp:** its data type is integer, and its access modifier is private.
2. **state:** its data type is boolean, and its access modifier is private.
3. **avaliableCount:** its data type is integer, and its access modifier is private.
4. **sessionDuration:** its data type is integer, and its access modifier is private.
5. **IPaddress:** its data type is string, and its access modifier is public.

Operations of InternalComputers Class:

1- **getComputersAvailability**: it is a public function that receives no parameters, it is used to inform the employee if there are internal computer available or not. It has a return variable from the data type: boolean that is true if the NumOfComp is more than 0.

2- **usedComputers**: it is a public function that receives no parameters, it is used to keep count on the number of the used internal computers, to then update the value of the availableCount. It has a return variable from the data type integer that holds the number of used computers.

3- **setInternalComp**: it is a public mutator function that sets the data of the internal computers inside the library's system; as it receives parameters, like NumOfComp, State, avaliableCount, and IPaddress. It has no return variables.

4- **setsession**: it is a public function that receives userName, password, sessionDuration, and IPaddress as inputs. It is used to reserve internal computers sessions by the staff for a specific member or visitor of the library.

Relationships with other Classes:

a. With Employee Class:

Association relation: The employee checksAvailableComputers.

Multiplicity: One or more internal computer may be checked by 0..1 employee.

b. With Visitors Class:

Association relation: The visitor **can use** an internal computer.

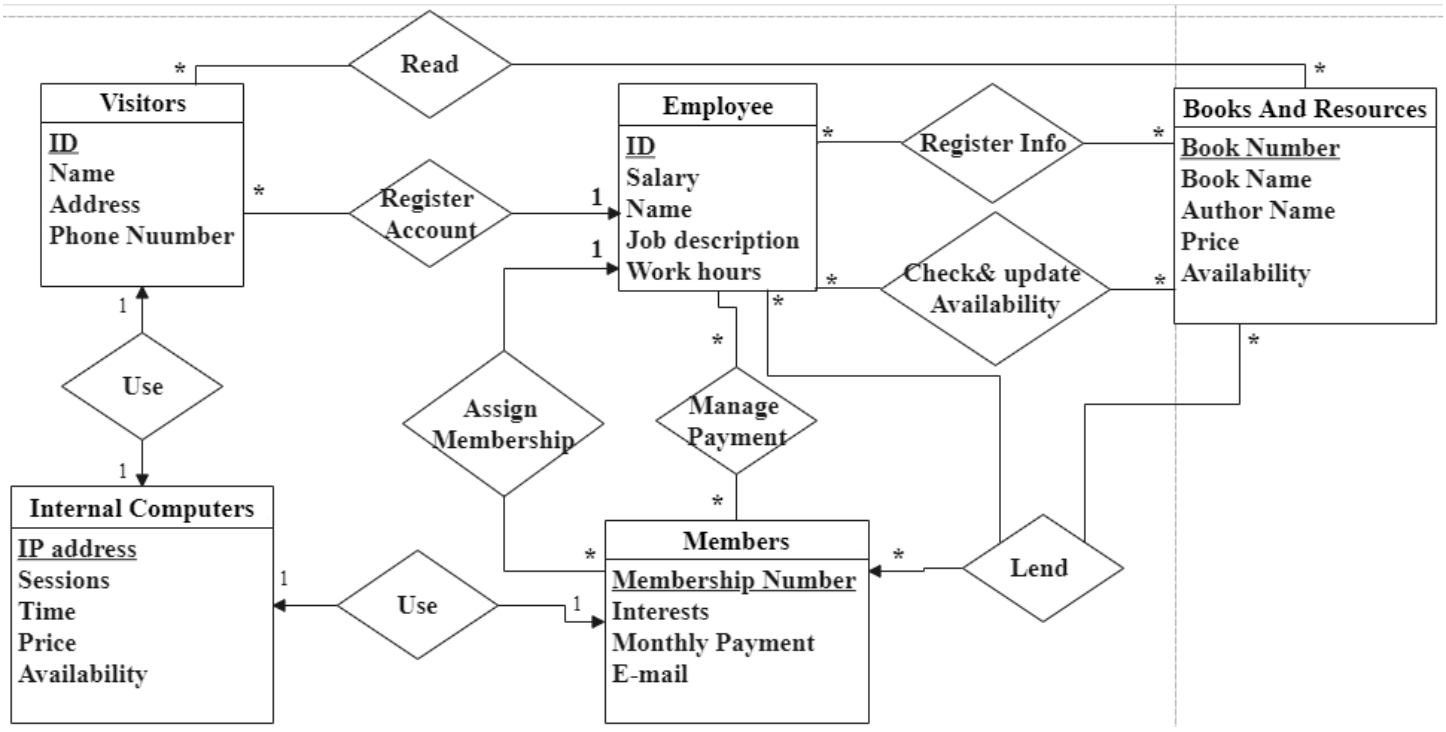
Multiplicity: 0..1 Visitor may use one internal computer.

c. With Members Class:

Association relation: a member **can use** an internal computer.

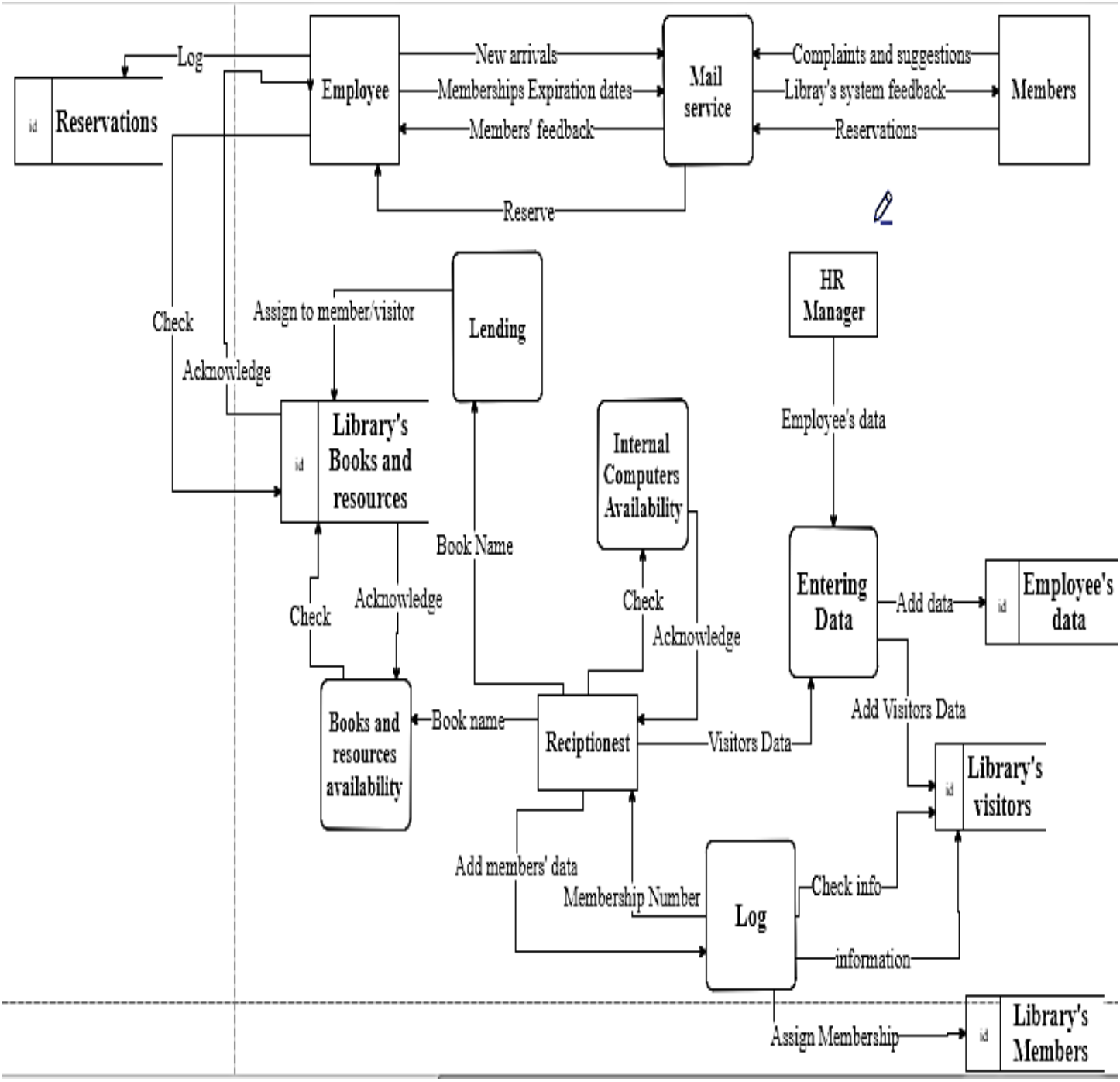
Multiplicity: 0..1 Member may use one internal computer.

Entity Relationship Diagram (ERD):

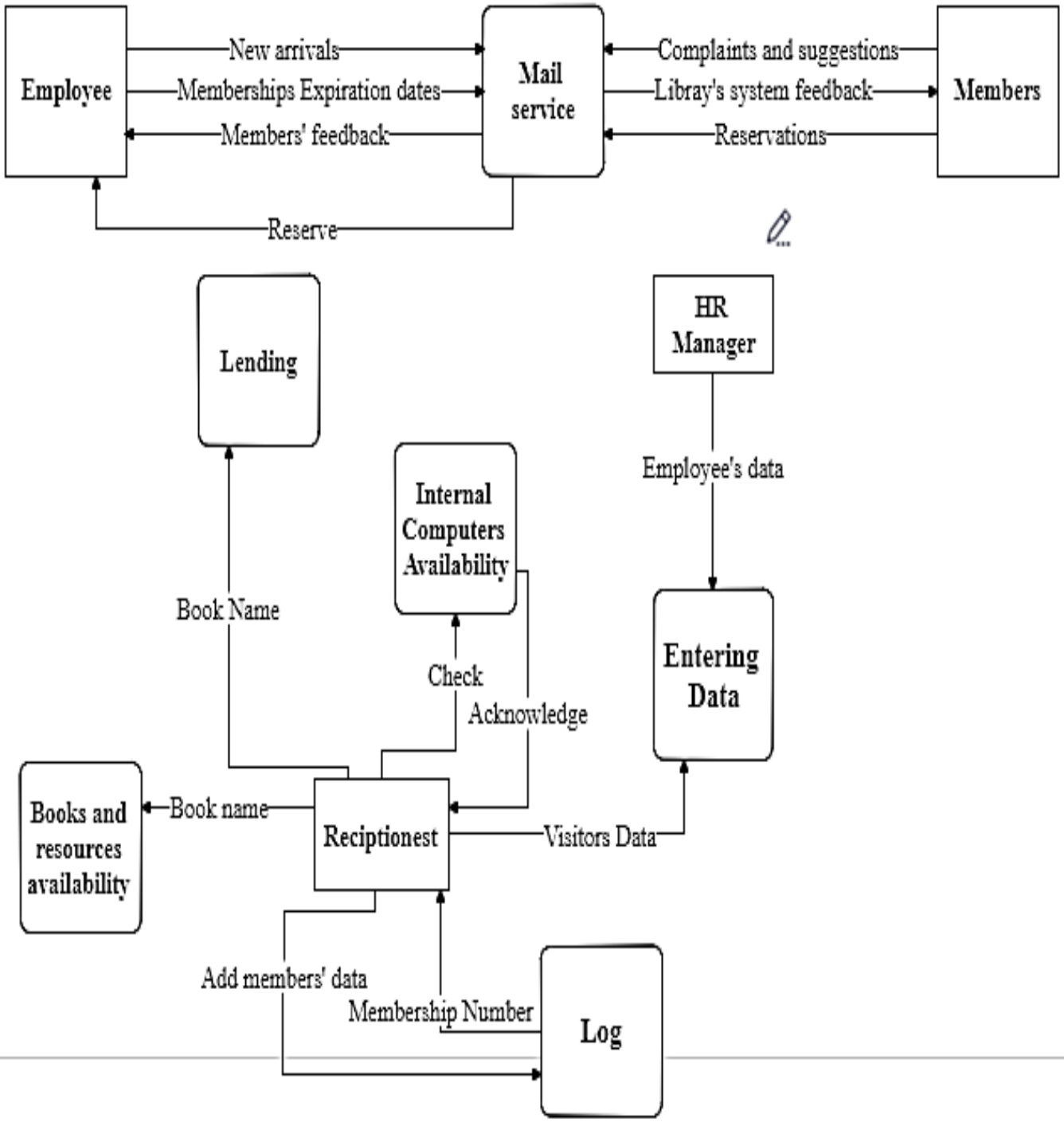


Data Flow Diagram (DFD):

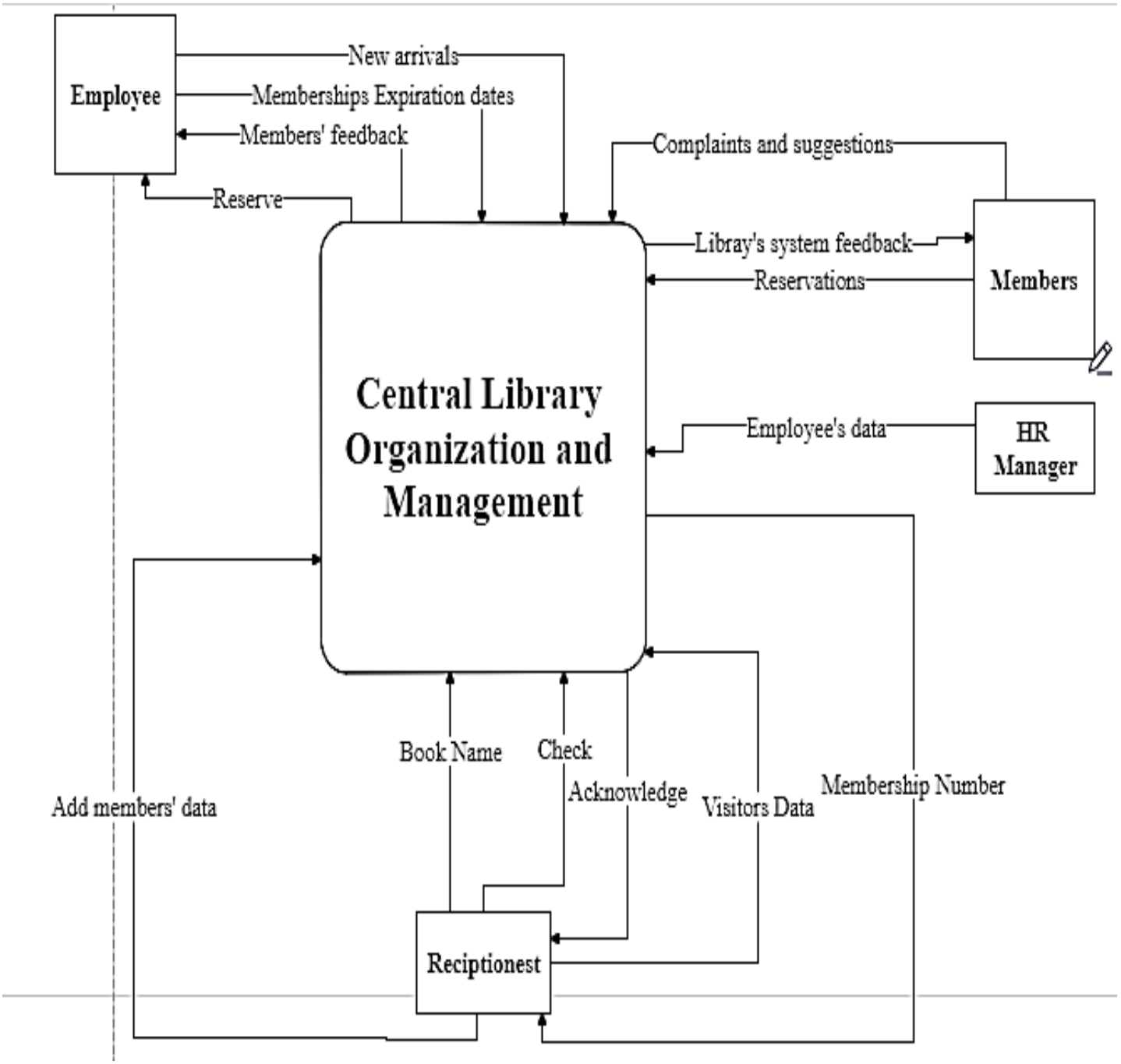
1-Level 1



1-Level 0


Employee
New arrivals
Memberships Expiration dates
Members' feedback
Mail service
Complaints and suggestions
Libray's system feedback
Reservations
Members
Reserve
Lending
HR Manager
Employee's data
Internal Computers Availability
Book Name
Check
Acknowledge
Entering Data
Books and resources availability
Book name
Reciptionest
Visitors Data
Add members' data
Membership Number
Log

3-Context level



6.4 Requirements Testing

1- Employee Class:

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	• JobTitle=”Receptionist” • EmpFName=”Mark” • EmpLName=”Tom” • Id=2001 • phoneNum=1122334 • salary=500 • work_hours=8	TRUE	TRUE	SUCCESSFUL
2	• JobTitle=”Receptionist” • EmpFName=”Lily” • EmpLName=”Dawn” • Id=2001 • phoneNum=1122334 • salary=500 • work_hours=8	TRUE	FALSE	UNSUCCESSFUL (Uniqueness Constraint)
3	• JobTitle = "Staff"; • EmpFName = null; • EmpLName ="Dawn"; • id = 2007; • phone_Num = 112277; • salary = 3000; • work_hours = 8;	FALSE	FALSE	SUCCESSFUL
4	• JobTitle = "Librarian"; • EmpFName = "Lisa"; • EmpLName = "Choi"; • id = 2008; • phone_Num = 112277; • salary = 5000; • work_hours = 6;	FALSE	FALSE	SUCCESSFUL

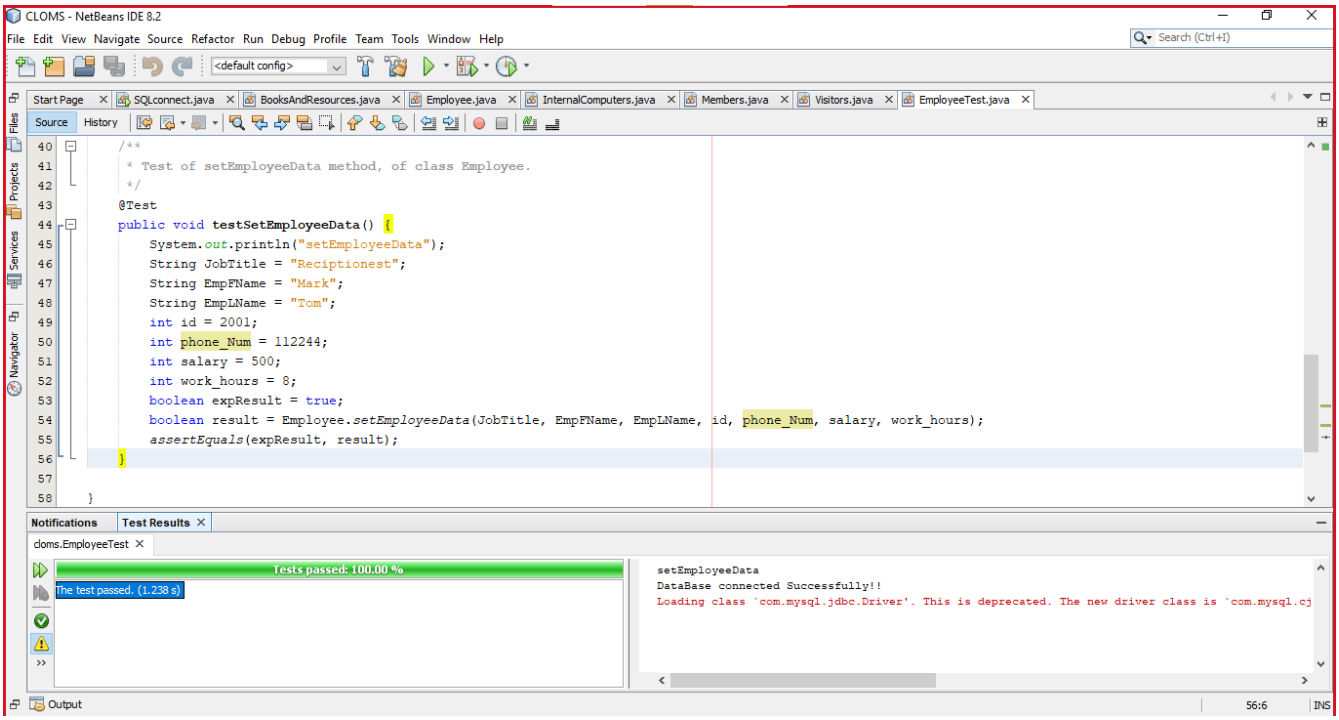
2- Visitor Class:

USECASE 1: SetVisitorsData

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	• name = "Adam karam" • ID = 3000 • phoneNumber =012345 • address = "22UNIVERSITY HELWAN" • userName = "adam" • password = "3000" • empID = 2000	TRUE	TRUE	SUCCESSFUL
2	• name = "Sam claflin"; • ID = 3000 • phoneNumber = 011222333; • address = "UNIVERSITY HELWAN" • userName = "sam" • password = "3000" • empID = 2001	FALSE	FALSE	SUCCESSFUL
3	• name = "Arial Walter"; • ID = 3001 • phoneNumber = 02233344; • address = "UNIVERSITY HELWAN" • userName = "arial" • password = "3001" • empID = 2005	FALSE	FALSE	SUCCESSFUL
4	• name = "Omar"; • ID = 3001	FALSE	FALSE	SUCCESSFUL

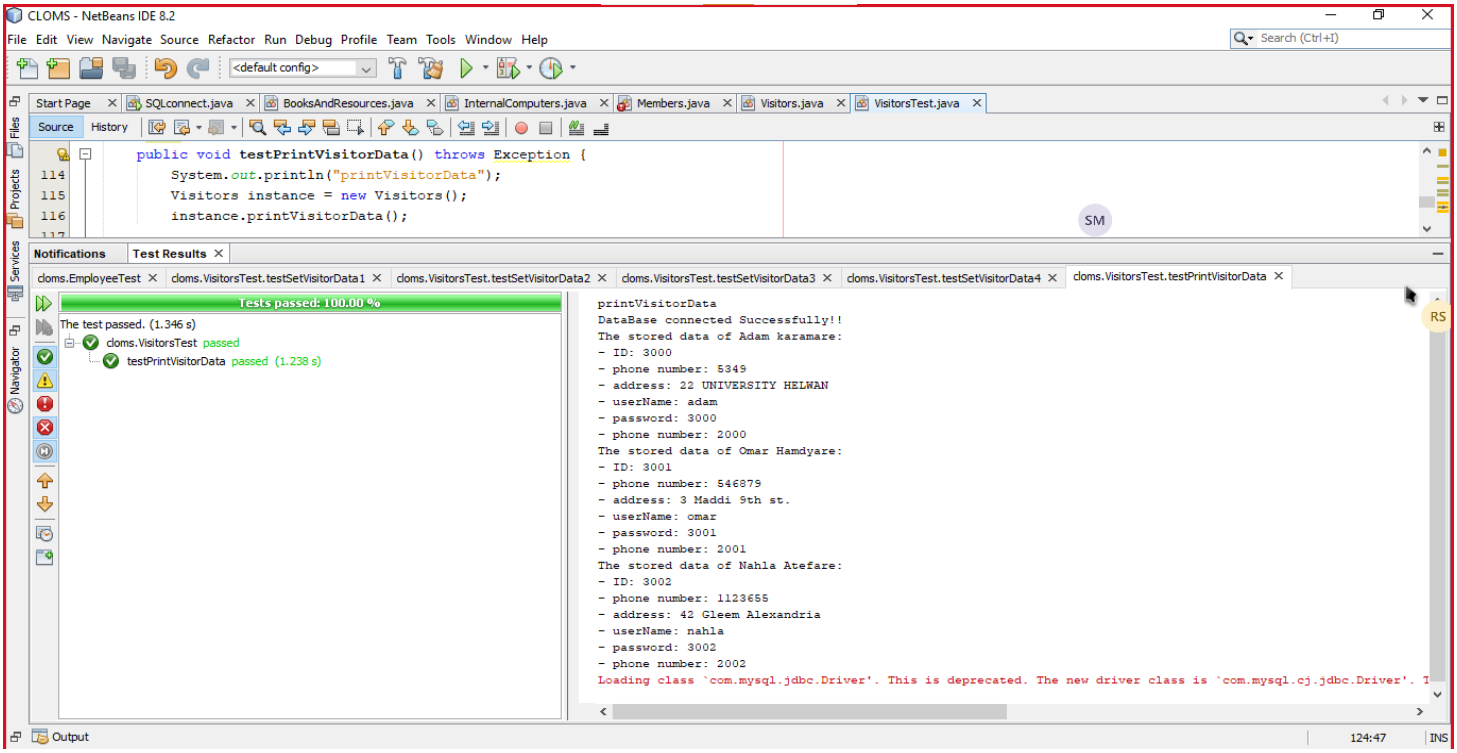
	• phoneNumber = 02233344; • address = "UNIVERSITY HELWAN" • userName = null • password = "3001" • empID = 2002			

Test Case 1:



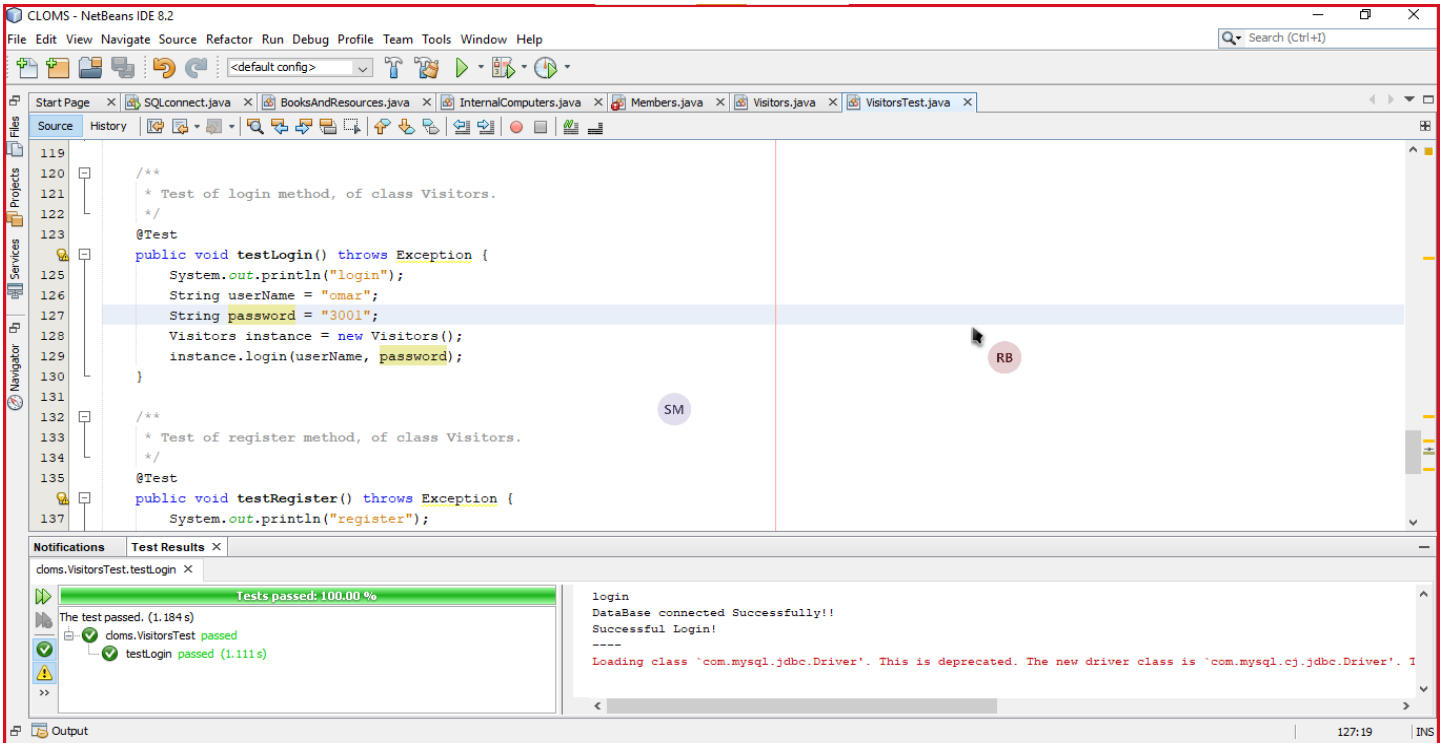
USECASE 2 : PrintVisitorsData

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	-	Prints all Tuples in Visitor Table	Prints all Tuples in Visitor Table	SUCCESSFUL



USECASE 3: Login

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	• Username=”Omar” • Password=”3001”	Successful Login	Successful Login	Successful Login
	•			

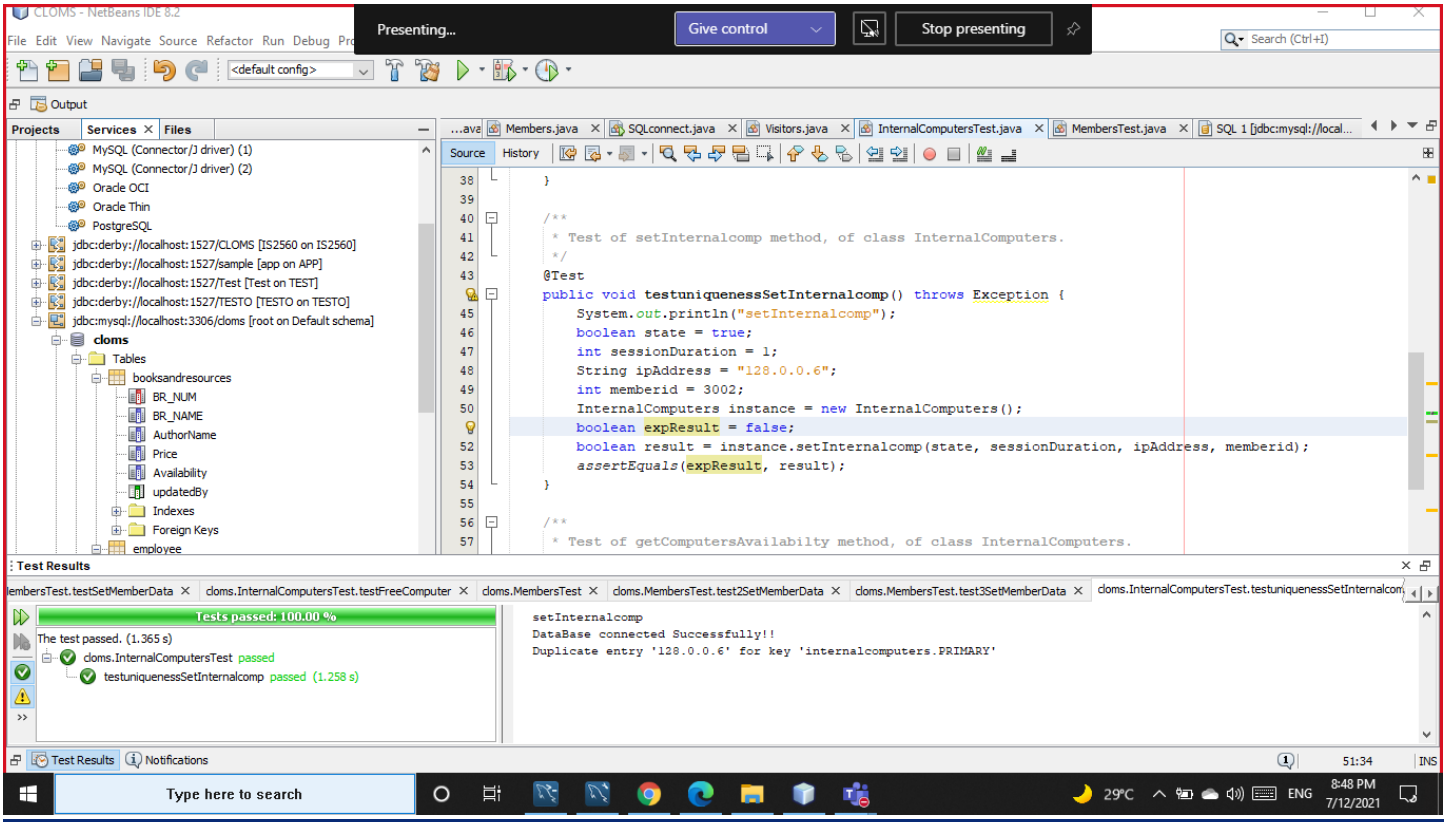


3- INTERNAL Computer:

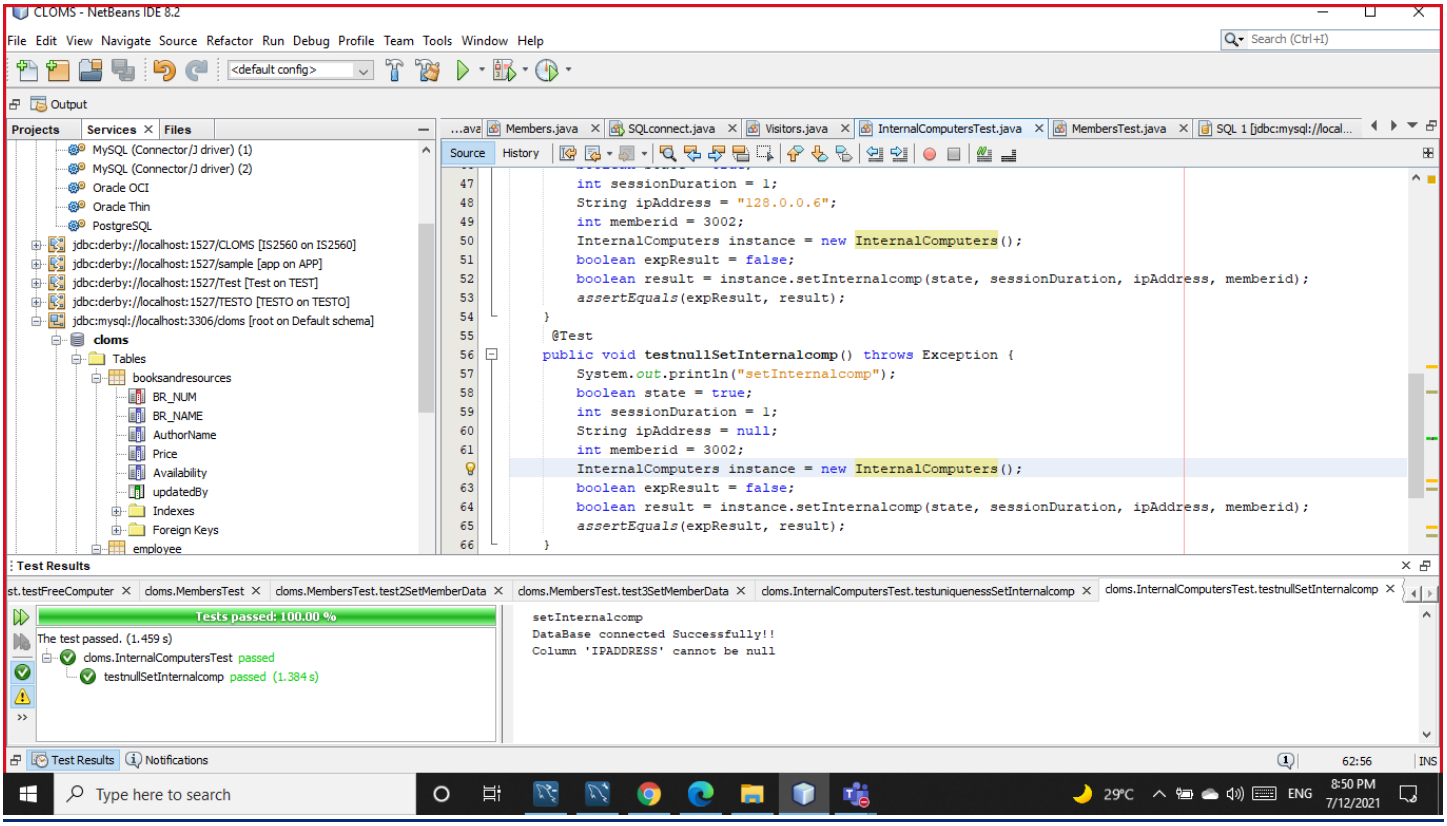
USE CASE 1: SetInternalComp

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	• State= true • Sessionduration=1 • ipAddress=”128.0.0.1 • memberid=3000	TRUE	TRUE	SUCCESSFUL
2	• State= true • Sessionduration=1 • ipAddress=”128.0.0.1 • memberid=3000	FALSE	FALSE	SUCCESSFUL
3	• State= true • Sessionduration=1 • ipAddress=null • memberid=3000	FALSE	FALSE	SUCCESSFUL

Test case 2:

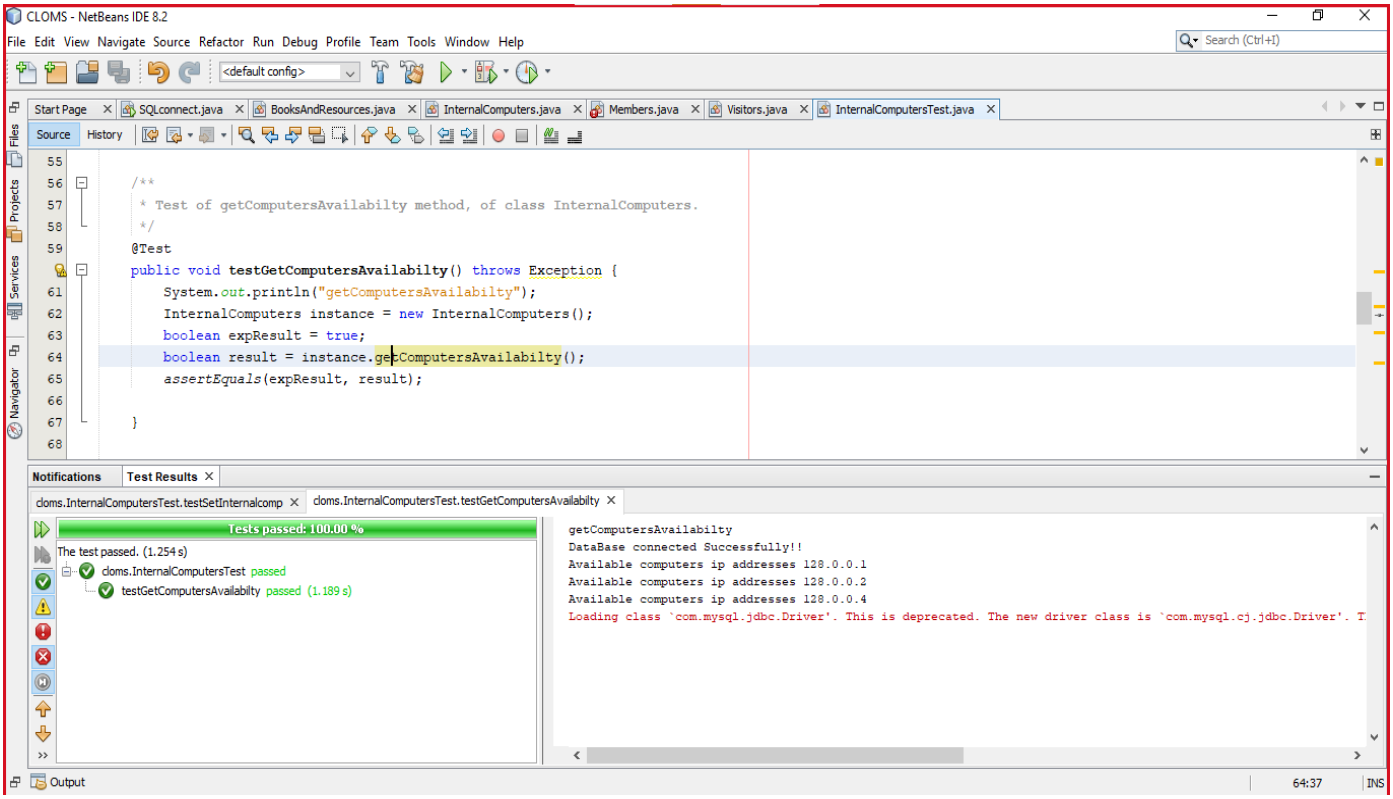


Test case 3:


CLOMS - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output
Projects
Services
Files
MySQL (Connector/J driver) (1)
MySQL (Connector/J driver) (2)
Oracle OCI
Oracle Thin
PostgreSQL
jdbc:derby://localhost:1527/CLOMS [IS2560 on IS2560]
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:derby://localhost:1527/Test [Test on TEST]
jdbc:derby://localhost:1527/TESTO [TESTO on TESTO]
jdbc:mysql://localhost:3306/cloms [root on Default schema]
cloms
Tables
booksandresources
BR_NUM
BR_NAME
AuthorName
Price
Availability
updatedBy
Indexes
Foreign Keys
employee
Members.java
SQLconnect.java
Visitors.java
InternalComputersTest.java
MembersTest.java
SQL 1 [jdbc:mysql://local...
Source
History
47 int sessionDuration = 1;
48 String ipAddress = "128.0.0.6";
49 int memberid = 3002;
50 InternalComputers instance = new InternalComputers();
51 boolean expResult = false;
52 boolean result = instance.setInternalcomp(state, sessionDuration, ipAddress, memberid);
53 assertEquals(expResult, result);
54 }
55 @Test
56 public void testnullSetInternalcomp() throws Exception {
57 System.out.println("setInternalcomp");
58 boolean state = true;
59 int sessionDuration = 1;
60 String ipAddress = null;
61 int memberid = 3002;
63 InternalComputers instance = new InternalComputers();
64 boolean expResult = false;
65 boolean result = instance.setInternalcomp(state, sessionDuration, ipAddress, memberid);
66 assertEquals(expResult, result);
66 }
Test Results
st.testFreeComputer
cloms.MembersTest
cloms.MembersTest.test2SetMemberData
cloms.MembersTest.test3SetMemberData
cloms.InternalComputersTest.testuniquenessSetInternalcomp
cloms.InternalComputersTest.testnullSetInternalcomp
Tests passed: 100.00 %
The test passed. (1.459 s)
cloms.InternalComputersTest passed
testnullSetInternalcomp passed (1.384 s)
setInternalcomp
DataBase connected Successfully!!
Column 'IPADDRESS' cannot be null
Test Results
Notifications
62:56
INS
Type here to search
29°C
ENG
8:50 PM
7/12/2021

USE CASE 2:GetComputerAvailability:

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	-	TRUE	TRUE	PRINTS AVAILABLE COMPUTER IP ADDRESS

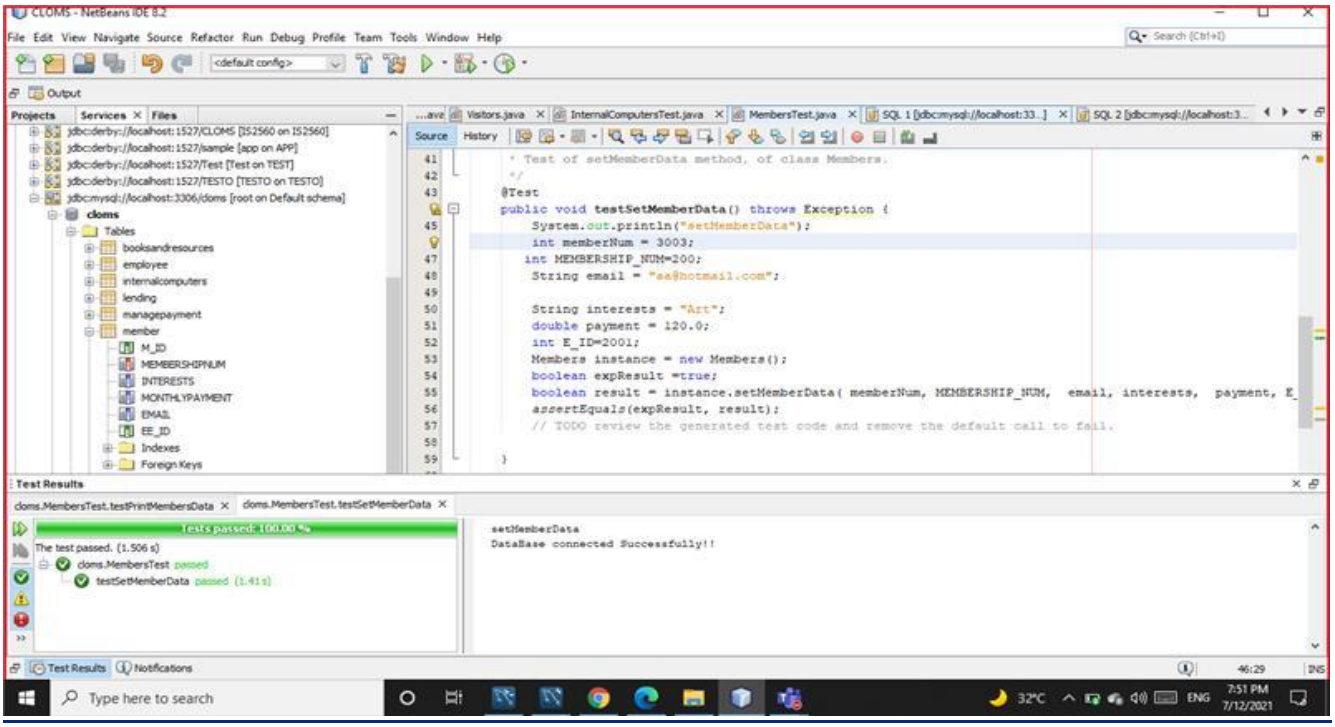


4- Member

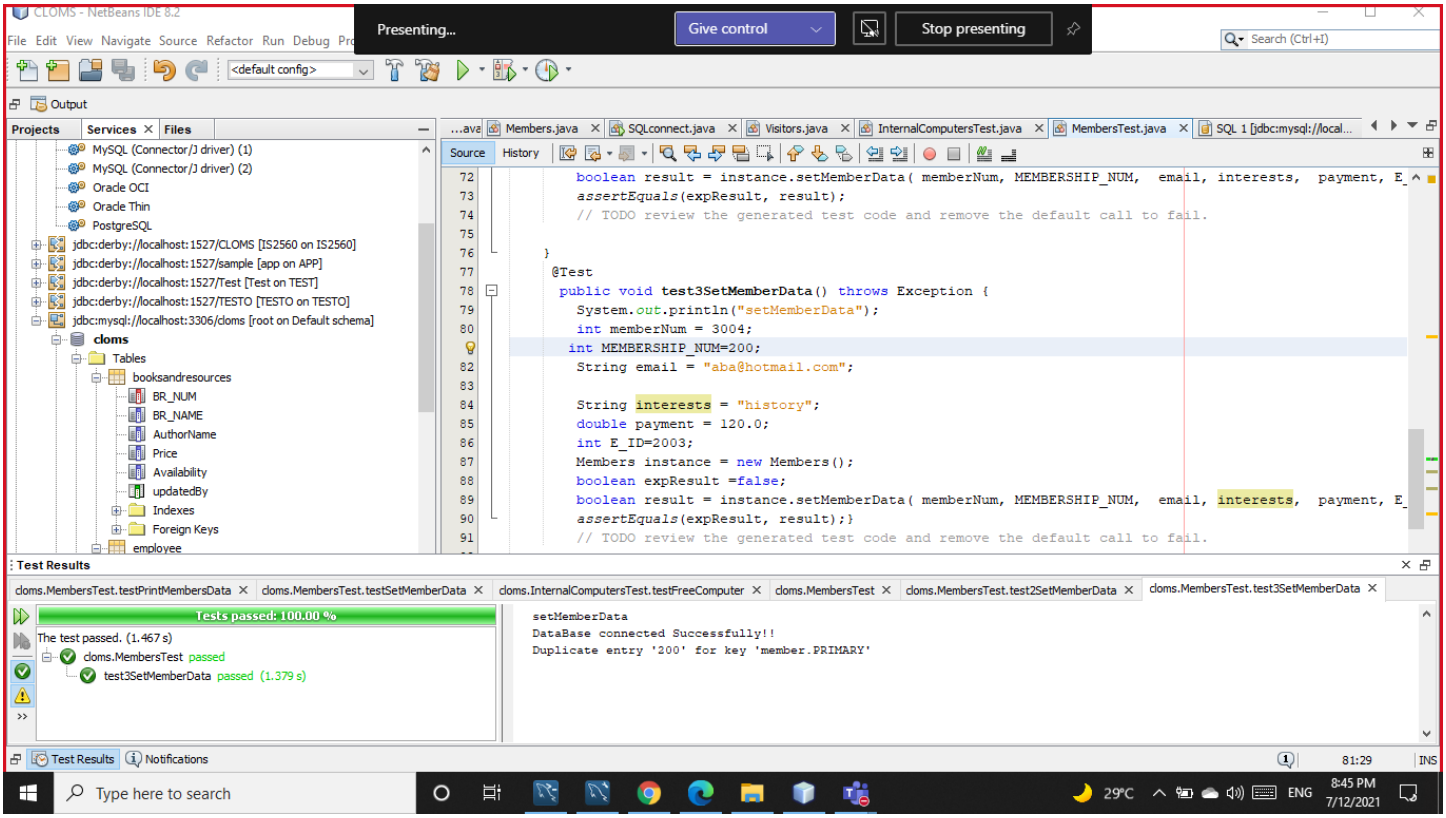
USECASE1: SetMemberData:

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	• memberNum=2003 • MEMBERSHPNUM=200 • Payment=100 • Email="aa@gmail.com" • Interests="football" • E_id=2000	TRUE	TRUE	SUCCESSFUL
2	• memberNum=2004 • MEMBERSHPNUM=200 • Payment=100 • Email="oo@gmail.com" • Interests="gaming" • E_id=2000 •	FALSE	FALSE	SUCCESSFUL

Test case1:



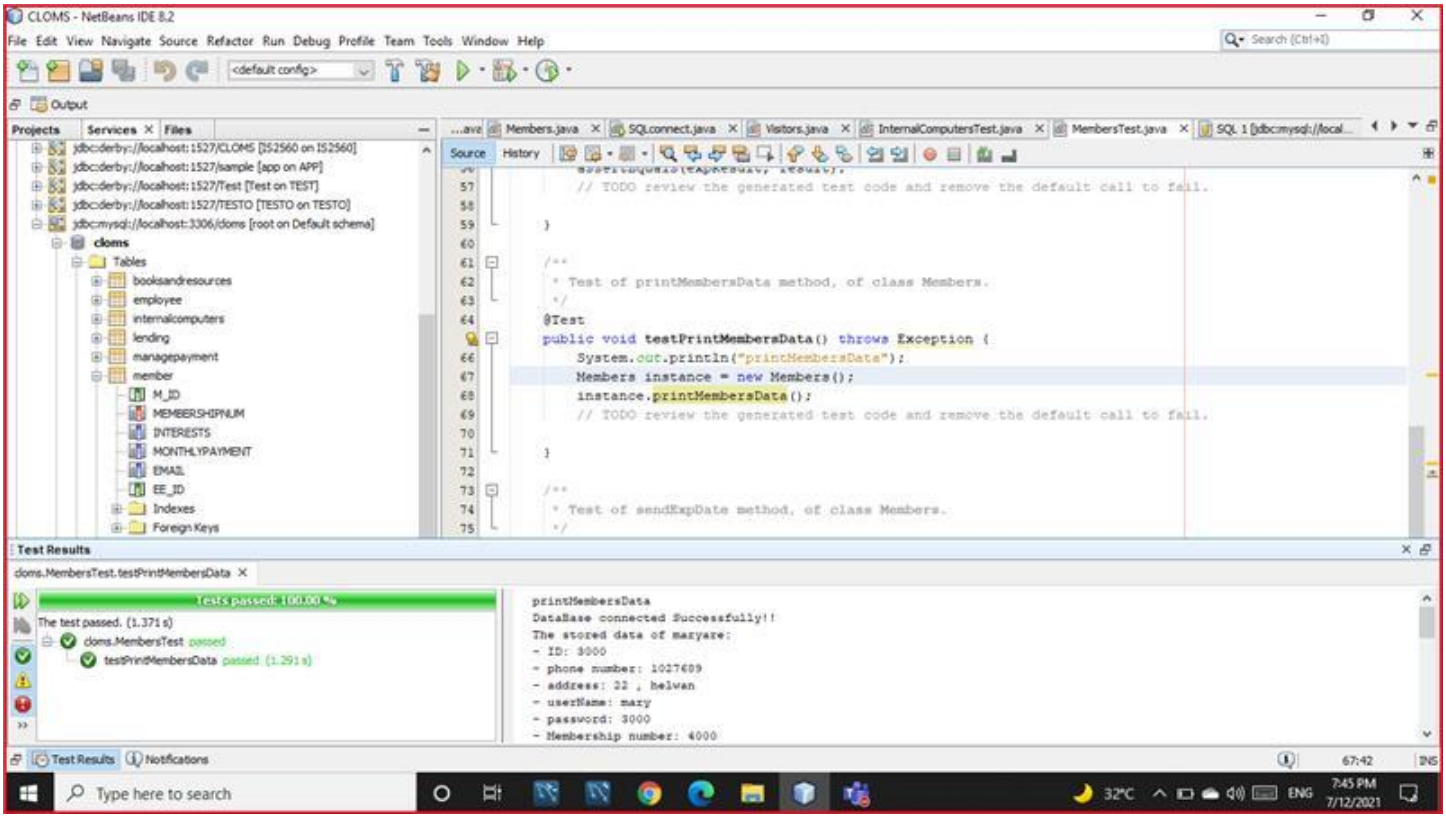
Test case 2:



USE CASE2: PrintMembersData:

Test Case ID	Input	Expected Output	Actual Output	System reaction
1	-	Prints Members Data	Prints Members data	Prints Members Data from both visitor and Member tables (joined)

Test case1:


CLOMS - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Services
Files
jdbc:derby://localhost:1527/CLOMS [IS2560 on IS2560]
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:derby://localhost:1527/Test [Test on TEST]
jdbc:derby://localhost:1527/TESTO [TESTO on TESTO]
jdbc:mysql://localhost:3306/cloms [root on Default schema]
cloms
Tables
booksandresources
employee
internalcomputers
lending
managepayment
member
M_ID
MEMBERSHIPNUM
INTERESTS
MONTHLYPAYMENT
EMAIL
EE_ID
Indexes
Foreign Keys
Members.java
SQLconnect.java
Visitors.java
InternalComputersTest.java
MembersTest.java
Source
History
// TODO review the generated test code and remove the default call to fail.
/**
* Test of printMembersData method, of class Members.
*/
@Test
public void testPrintMembersData() throws Exception {
System.out.println("printMembersData");
Members instance = new Members();
instance.printMembersData();
// TODO review the generated test code and remove the default call to fail.
/**
* Test of sendExpDate method, of class Members.
*/
Test Results
cloms.MembersTest.testPrintMembersData
Tests passed: 100.00 %
The test passed. (1.371 s)
cloms.MembersTest passed
testPrintMembersData passed (1.291 s)
printMembersData
DataBase connected Successfully!!
The stored data of maryare:
- ID: 3000
- phone number: 1027609
- address: 22 , helwan
- userName: mary
- password: 3000
- Membership number: 4000
Test Results
Notifications
67:42
INS
Type here to search
32°C
ENG
7:45 PM
7/12/2021