

Breakdown of individual contributions:

Contributions of: *Sarah Gamal El-Deen Mohamed*

- Bookandresources class
- SetBookInfo function
- Lending function
- Getcomputeravailability function
- SearchEbook function
- sendRecommendation function
- BookandResources , lending data conceptual model
- Created BooksandResources , lending relational ables
- GUI

Contributions of: *Rana Mohamed Hussein*

- Member and employee classes
- Setmemberdata function
- Setemployeedata function
- Employee and member data conceptual model
- Created Employee and member relational tables.
- Created database in my sql work bench.
- GUI design

Contributions of: *Rahma Mohamed Makram*

- Internal computers class
- setInternalcomputer function
- setsession function
- getcomputeravailability function
- freecomputer function
- Internalcomputer ,ManagePayment data conceptual model
- GUI
- Created internalComputer , ManagePayment relational tables
- created database in my sql workbench

Contributions of: *Solwan Shokry Ahmed Mohamed*

- Member and visitor classes
- Printmemberdata function
- Setvistordata function
- Login function
- Visitor conceptual model
- Created visitor, read relational tables
- GUI design
- Derby database and Database connection with netbeans

2. Data (Conceptual) Model:

a) Domain:

1. Books and Resources

Field	Data type	Constraints
BookNumber	INT	PRIMARY KEY
BookName	Varchar(30)	NOT NULL
AuthorName	Varchar(30)	NOT NULL
Price	INT	
Availability	INT	NOT NULL
Updated by	INT	Foreign key

2. Employee

Field	Data type	Constraints
ID	INT	PRIMARY KEY
FNAME	VARCHAR(20)	NOT NULL
LNAME	VARCHAR(20)	NOT NULL
SALARY	INT	
JOBDESCRIPTION	VARCHAR(20)	
WORKHOURS	INT	NOT NULL
PHONENUM	INT	

3. Members

Field	Data type	Constraints
M_ID	INT	FOREIGN KEY
MEMBERSHIPNUMBER	INT	PRIMARY KEY
INTRESETS	VARCHAR(35)	
MONTHLYPAYMENT	INT	
EMAIL	VARCHAR(20)	
E_ID	INT	FOREIGN KEY

4. Lending

Field	Datatype	Constraints
EmployeeID	INT	PRIMARY KEY
BookNumber	INT	PRIMARY KEY
MembershipNum	INT	PRIMARY KEY

5. Read

Field	Datatype	Constraints
VisitorID	INT	Primary key-Foreign key
Booknumber	INT	Primary key -Foreign key

6. Visitor

Field	Data type	Constraints
ID	INTEGER	PRIMARY KEY
NAME	VARCHAR(50)	NOT NULL
ADDRESS	VARCHAR(3700)	-
PHONENUMBER	INTEGER(10)	-
USERNAME	VARCHAR(20)	NOT NULL
PASSWORD	VARCHAR(20)	NOT NULL
E_ID	INT	FOREIGN KEY

7. Internal Computers

Field	Data type	constraints
State	Boolean	NOT NULL
SessionDuration	Int	NOT NULL
ipAddress	varchar	PRIMARY KEY
memberid	Int	Foreign key

8. ManagePayment

Field	Data type	Constraints
M_ID	INT	Foreign Key – primary key
MEMBERSHIP_NUMBER	INT	Foreign Key – primary key

Employee:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Tables

- booksandresources
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- employee
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- internalcomputers
- lending
- managepayment
- member
- read
- visitor

booksandresources - Table booksandresources employee - Table employee

Limit to 1000 rows

1 • SELECT * FROM cloms.employee;

Result Grid

ID	FNNAME	LNNAME	SALARY	JOBDESCRIPTION	WORKHOURS	PHONENUM
2000	Rahma	mukram	4500	librarian	8	1234556
2001	Sara	carial	4500	librarian	8	234567
2002	rana	mohamed	4500	librarian	9	345679
2003	salwan	shakry	5000	recreationist	10	123478
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Internalcomputers:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Tables

- booksandresources
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- employee
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- internalcomputers
- lending
- managepayment
- member
- read
- visitor

booksandresources - Table booksandresources employee - Table employee

Limit to 1000 rows

1 • SELECT * FROM cloms.internalcomputers;

Result Grid

STATE	SESSIONDURATION	IPADDRESS	USERID
1	1	128.0.0.2	3001
1	1	128.0.0.6	3001
NULL	NULL	NULL	NULL

Member :

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator:

SCHEMAS

Filter objects

- Columns
- Indexes
- Foreign Keys
- Triggers
- internalcomputers
- lending
- management
- member
- read
- visitor

Administration Schemas

Information:

Table: **lending**

Columns:

- EmployeeId** int PK
- BookNumber** int PK
- MembershipNum** int PK

employee - Table employee internalcomputers lending - Table lending member - Table member x

1 • SELECT * FROM cloms.member;

Result Grid

M_ID	MEMBERSHIPNUM	INTERESTS	MONTHLYPAYMENT	EMAIL	EE_ID
3003	200	Art	120	sa@hotmai.com	2001
3004	220	history	120	ab@hotmai.com	2003
3000	4000	football	100	we@fomai.com	2000
3001	4001	music	100	em@fomai.com	2001
3002	4002	tennis	100	ok@fomai.com	2002
NULL	NULL	NULL	NULL	NULL	NULL

Automatic toolbar to caret position

Visitor :

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator:

SCHEMAS

Filter objects

- Triggers
- internalcomputers
- lending
- Columns
- Indexes
- Foreign Keys
- Triggers
- management
- member
- read
- visitor

Administration Schemas

Information:

Table: **lending**

Columns:

- EmployeeId** int PK
- BookNumber** int PK
- MembershipNum** int PK

internalcomputers lending - Table lending member - Table member visitor - Table visitor x

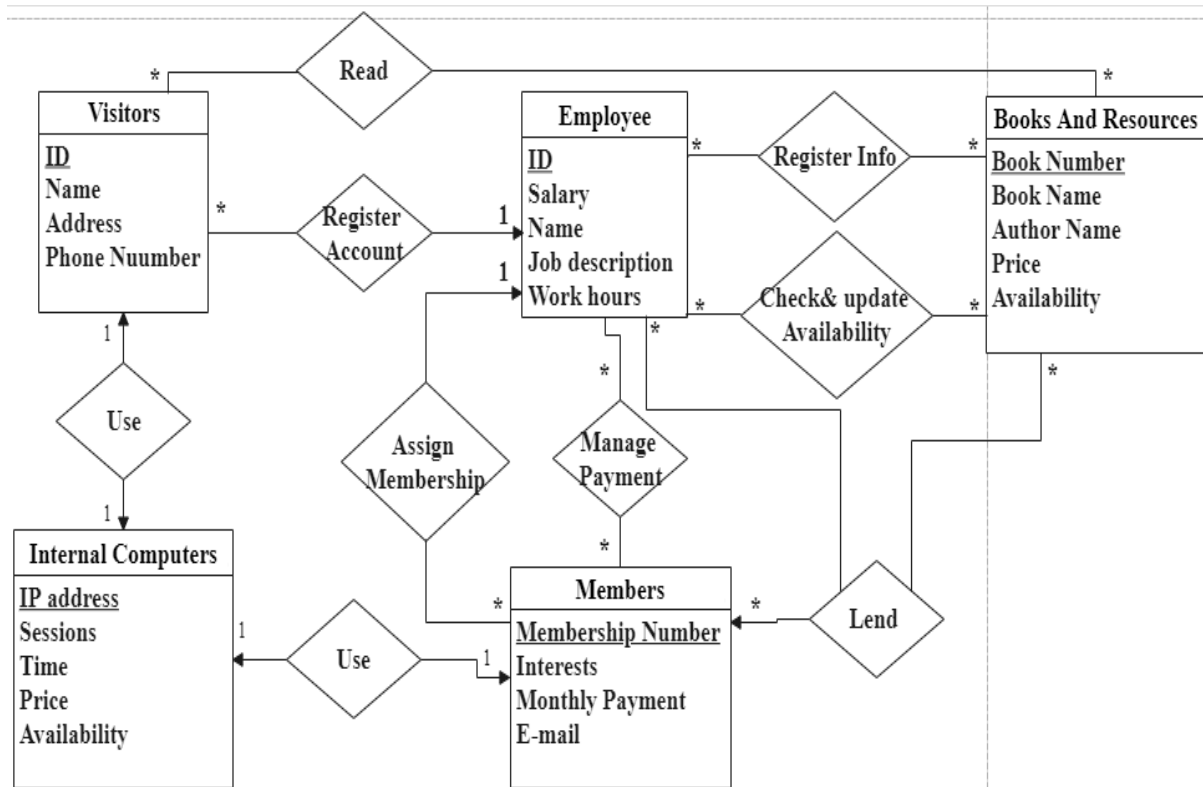
1 • SELECT * FROM cloms.visitor;

Result Grid

V_ID	NAME	ADDRESS	PHONENUMBER	USERNAME	PASSWORD	E_ID
3000	marv	22, Nelson	1027689	marv	3000	2001
3001	john	15, outst	1028793	john	3001	2000
3002	ahmed	nour city	1028756	ahmed	3002	2002
3003	mai	2, masad	1056782	mai	3003	2003
3004	marina	outst	13468	marina	3004	2002
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Automatic toolbar to caret position

b) Conceptual Data Model Diagram:



3. Logical and Physical Modelling:

Relational Scheme:

Employee

<u>ID</u>	FNAME	LNAME	SALARY	JOBDESCRIPTION	WORKHOURS	PHONENUM
-----------	-------	-------	--------	----------------	-----------	----------

Books and Resources

<u>BOOKNUMBER</u>	BOOKNAME	AUTHORNAME	PRICE	AVAILABILTY	UPDATEDBY
-------------------	----------	------------	-------	-------------	-----------

Visitors

<u>V_ID</u>	NAME	ADDRESS	PHONENUMBER	USERNAME	PASSWORD
-------------	------	---------	-------------	----------	----------

Members

MID	<u>MEMBERSHIPNUMBER</u>	INTERESTS	MONTHLYPAYMENT	EMAIL	E_ID
-----	-------------------------	-----------	----------------	-------	------

InternalComputers

State	SessionDuration	<u>ipAddress</u>	memberid
-------	-----------------	------------------	----------

ManagePayment

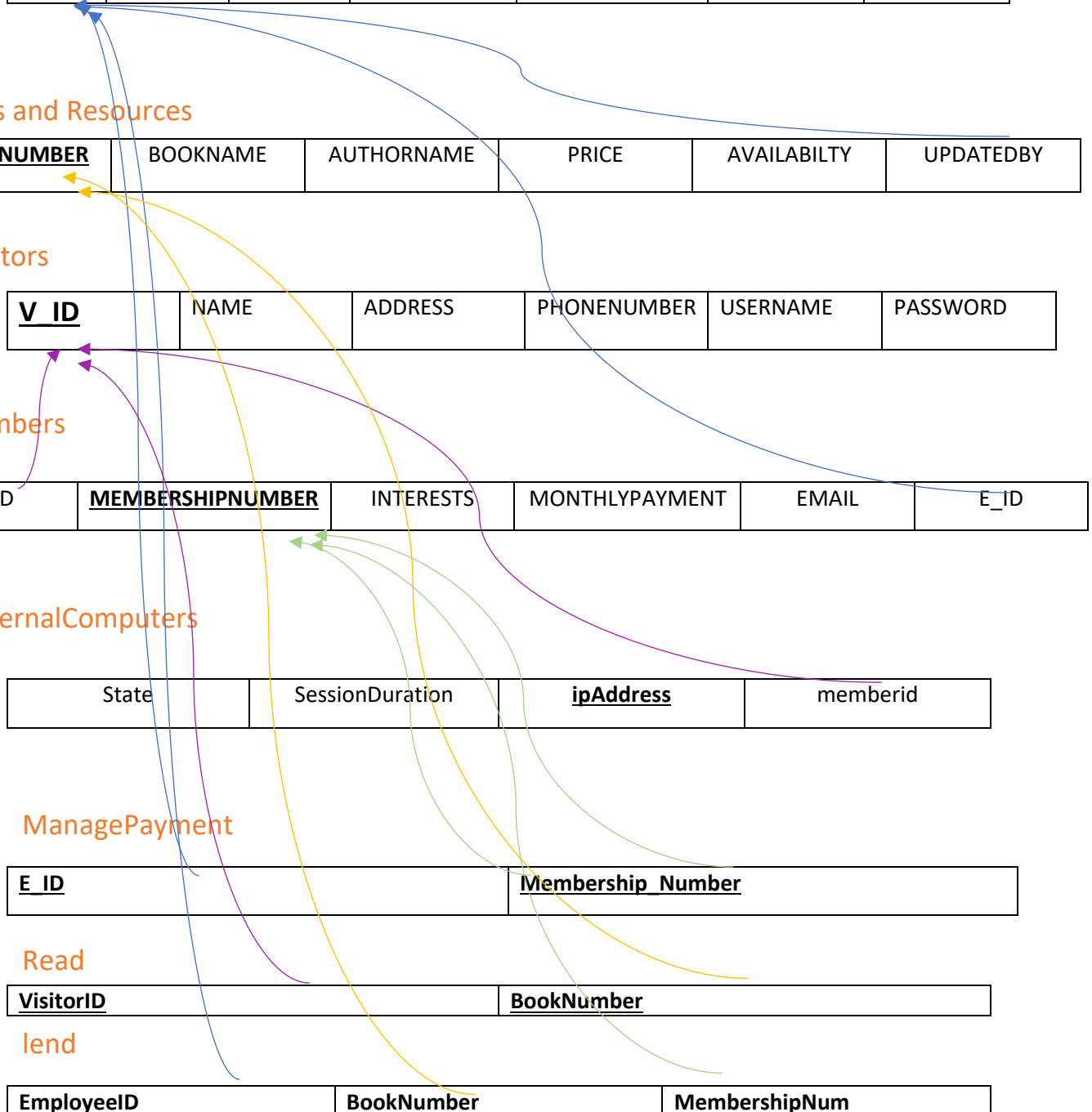
<u>E_ID</u>	<u>Membership_Number</u>
-------------	--------------------------

Read

<u>VisitorID</u>	<u>BookNumber</u>
------------------	-------------------

lend

<u>EmployeeeID</u>	<u>BookNumber</u>	<u>MembershipNum</u>
--------------------	-------------------	----------------------



4 Database Implementation:

SQL statements:

1. Books and Resources:

CREATE TABLE BooksAndResources

(BR_NUMBER int NOT NULL ,

BR_Name varchar(30) NOT NULL,

AuthorName varchar(30) NOT NULL,

Price int NOT NULL,

Availability int NOT NULL ,

updatedBy INT NOT NULL

);

ALTER TABLE BooksAndResources

ADD PRIMARY KEY (BR_NUMBER);

ALTER TABLE BooksAndResources

ADD FOREIGN KEY (updatedBy)REFERENCES EMPLOYEE(ID);

2. Lending:

CREATE TABLE LENDING

(EmployeeID INT NOT NULL,

BookNumber INT NOT NULL,

MembershipNum INT NOT NULL ,

);

ALTER TABLE LENDING

ADD PRIMARY KEY (EmployeeID,BookNumber,MembershipNum);

ALTER TABLE LENDING

ADD FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID);

ALTER TABLE LENDING

ADD FOREIGN KEY (MembershipNum) REFERENCES MEMBER
(MEMBERSHIPNUMBER);

ALTER TABLE LENDING

ADD FOREIGN KEY (BookNumber) REFERENCES BooksAndResources
(BR_NUMBER);

3. Read:

CREATE TABLE READ_BOOK (

VisitorID INT NOT NULL,

Booknumber INT NOT NULL

);

ALTER TABLE READ_BOOK

ADD PRIMARY KEY (VisitorID, Booknumber);

ALTER TABLE READ_BOOK

ADD FOREIGN KEY (VisitorID) REFERENCES VISITOR (V_ID);

```
ALTER TABLE READ_BOOK
```

```
ADD FOREIGN KEY ( Booknumber) REFERENCES BooksAndResources  
(BR_NUMBER);
```

4. Employee:

```
CREATE TABLE EMPLOYEE (  
ID INT NOT NULL,  
FNAME VARCHAR(20) NOT NULL,  
LNAME VARCHAR(20) NOT NULL,  
SALARY INT,  
JOBDESCRIPTION VARCHAR(20) NOT NULL,  
WORKHOURS INT NOT NULL,  
PHONENUM INT  
CHECK(SALARY>=500),  
CHECK(WORKHOURS>=8)  
);  
ALTER TABLE EMPLOYEE  
ADD PRIMARY KEY (ID) ;
```

5. Visitors:

```
CREATE TABLE VISITOR (  
V_ID INTEGER NOT NULL,  
V_NAME VARCHAR(50) NOT NULL,  
ADDRESS VARCHAR (3700),
```

```
PHONENUMBER INTEGER,  
USERNAME VARCHAR(20) NOT NULL,  
PASSWORD VARCHAR(20) NOT NULL.  
E_ID INT NOT NULL  
);  
  
ALTER TABLE VISITOR  
ADD PRIMARY KEY(V_ID);  
  
ALTER TABLE VISITOR  
ADD FOREIGN KEY(E_ID) REFERENCES EMPLOYEE(ID);
```

6. Members:

```
CREATE TABLE MEMBER (  
M_ID INT NOT NULL,  
MEMBERSHIPNUMBER INT NOT NULL ,  
INTERESTS VARCHAR(30),  
MONTHLYPAYMENT INT NOT NULL ,  
EMAIL VARCHAR(20),  
E_ID INT NOT NULL  
);  
  
ALTER TABLE MEMBER  
ADD PRIMARY KEY (MEMBERSHIPNUMBER) ,  
ADD FOREIGN KEY (M_ID) REFERENCES VISITOR (V_ID),
```

ADD FOREIGN KEY (E_ID) REFERENCES EMPLOYEE (ID);

7. InternalComputers

CREATE TABLE InternalComputers (

State Boolean NOT NULL,

SessionDuration INT,

IpAddress VARCHAR (30),

memberid INT NOT NULL

);

ALTER TABLE INTERNALCOMPUTERS

ADD PRIMARY KEY (ipAddress);

ALTER TABLE INTERNALCOMPUTERS

ADD FOREIGN KEY (memberid) REFERENCES VISITOR (V_ID);

8. ManagePayment

CREATE TABLE ManagePayment (

M_ID INT NOT NULL,

MEMBERSHIP_NUMBER INT NOT NULL

E_ID NOT NULL

);

ALTER TABLE ManagePayment

ADD PRIMARY KEY (E_ID,M_NUMBER)

ALTER TABLE ManagePayment

ADD FOREIGN KEY (E_ID) REFERENCES EMPLOYEE (ID),

ALTER TABLE ManagePayment

ADD FOREIGN KEY (M_NUMBER) REFERENCES MEMBER

(MEMBERSHIPNUMBER);

INSERT COMMANDS:

INSERT INTO employee VALUES (2000, Mary, Henry, 3000,
Receptionist, 8, 001120);

INSERT INTO internalcomputers VALUES (1, 1,"128.0.0.1,3000);

INSERT INTO lending VALUES(2000,1,1);

INSERT INTO managepayment VALUES(3000,1,2000);

INSERT INTO member VALUES

(3000,1,"politics",200,"name@mail.com" ,2000);


INSERT INTO visitor VALUES(3000,"Suzi Choi", "South Korea",
0011222333, "suzi", 3000, 200);

INSERT INTO read (3000,1);

INSERT INTO booksandresources VALUES (1,"", "", 80, 1,2000);

5. Application Implementation

1.SartUP Page:



Start Page

CENTRAL LIBRARY ORGANIZATION AND MANAGMENT SYSTEM

Please Choose Mode then enter Username and Password to Login:

Mode:

☒ Employee

☐ Members

UserName:

Password:

LOGIN

Inputs: in UserName text field : Employee.USERNAME="cloms"

in Password text field : Employee.Password= "1234"

CENTRAL LIBRARY ORGANIZATION AND MANAGMENT SYSTEM

Please Choose Mode then enter Username and Password to Login:

Mode:

☒ Employee

☐ Members

UserName: cloms

Password: ****

LOGIN

CENTRAL LIBRARY ORGANIZATION AND MANAGMENT SYSTEM

Please Choose Mode then enter Username and Password to Login:

Mode:

☐ Employee

☒ Members

UserName: rana

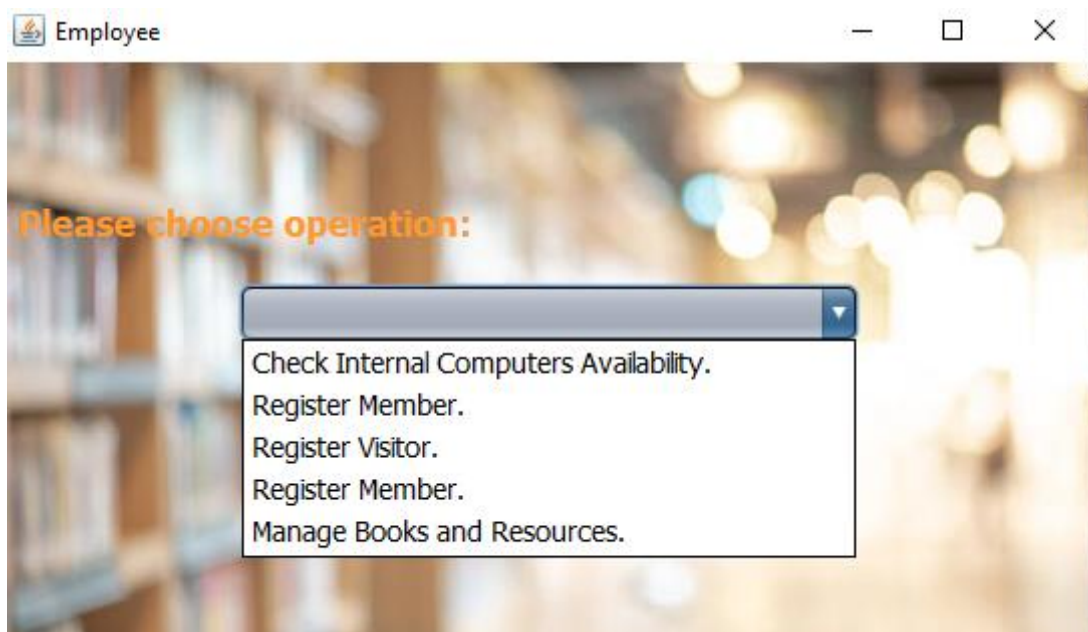
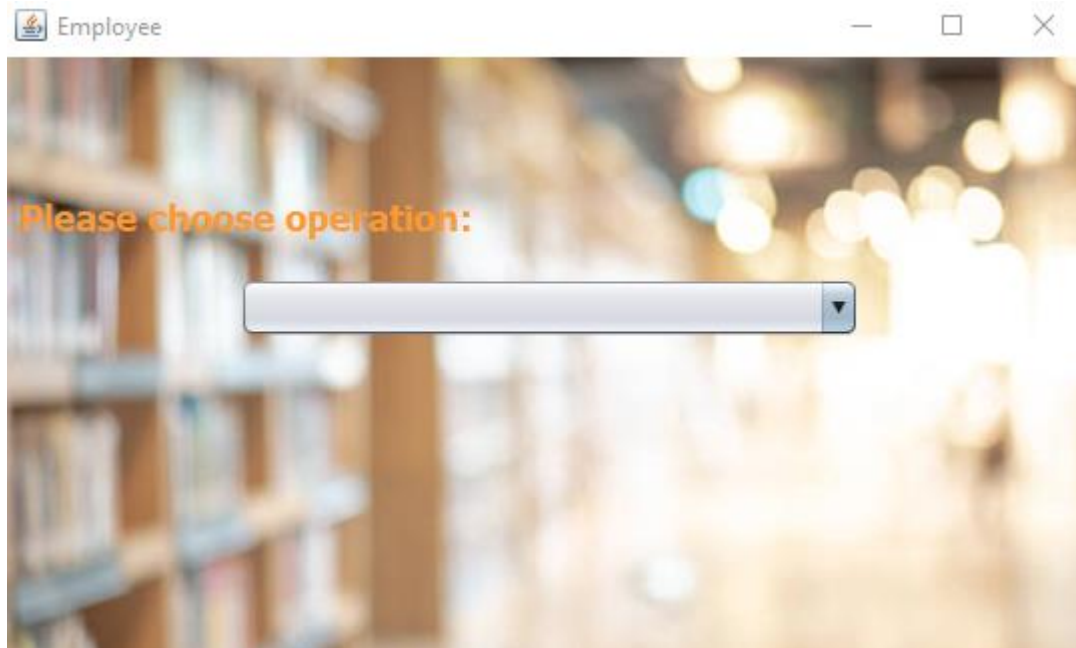
Password: ***

LOGIN

in UserName text field : Member.USERNAME="rana"

in Password text field : Member.Password= "306"

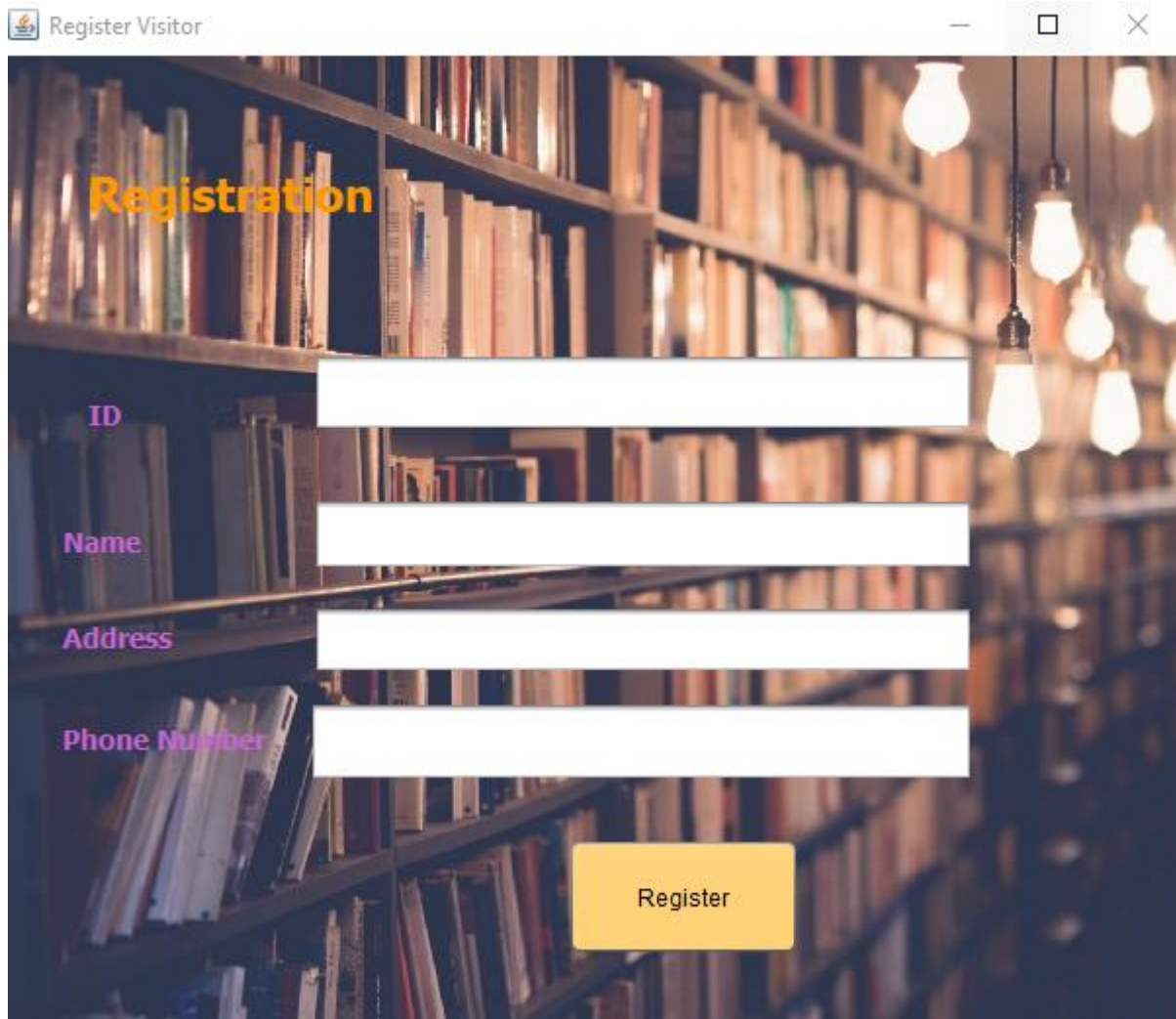
2.Employee Window:



Input:

Choose Register visitor window

3. Register Visitor Window



The screenshot shows a window titled "Register Visitor" with a background image of a library. The window contains a registration form with the following fields and labels:

- Registration** (Title)
- ID** (Label) with a text input field.
- Name** (Label) with a text input field.
- Address** (Label) with a text input field.
- Phone Number** (Label) with a text input field.
- Register** (Button)

Input

In ID text field: Visitor.V_ID= 3006

In Name text field Visitor.V_NAME="Mark"

In Address text field Visitor.ADRESS= "Lee"

In Phone Number text field Visitor.PHONENUMBER=0112223344

Registration

ID

3009

Name

Mark

Address

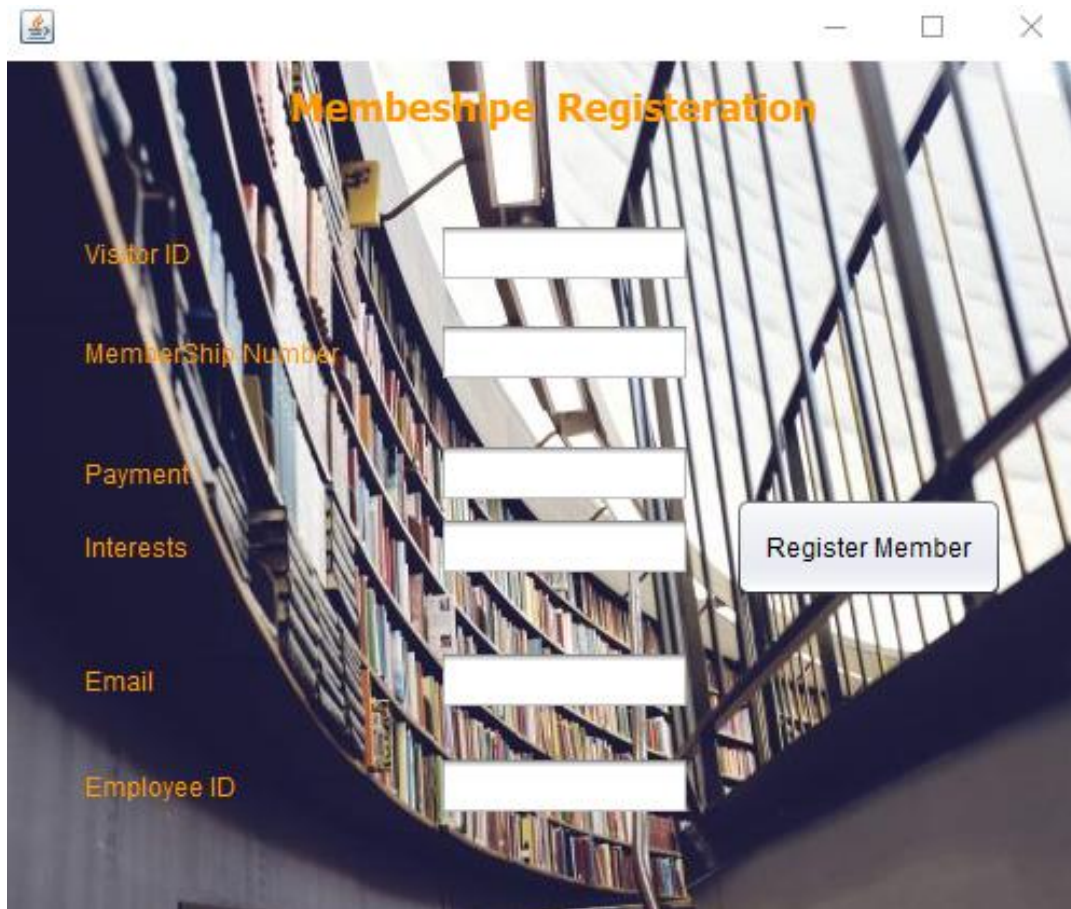
Lee

Phone Number

0112223344

Register

4.Member Registration



Membeshipe Registration

Visitor ID

MemberShip Number

Payment

Interests

Email

Employee ID

Register Member

Input:

In Visitor ID text field: Member.M_ID=3009

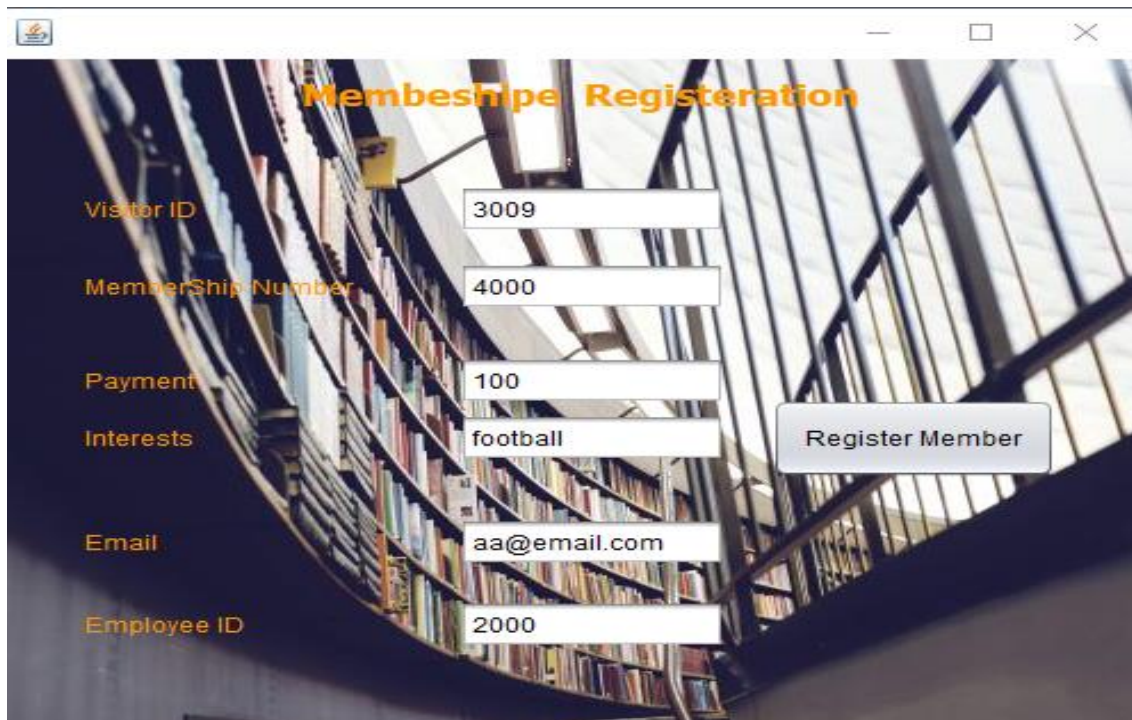
In MemberShip Number text field: Member.MEMBERSHIPNUMBER=4000

In Payment text field: Member.MONTHLYPAYMENT=100

In Interests text field: Member.INTERESTS=football

In Email text field: Member.EMAIL=aa@email.com

In Employee ID text field: Member.EE_ID=2000



Membership Registration

Visitor ID

Membership Number

Payment

Interests

Email

Employee ID

4. Check Computers Availability Window:

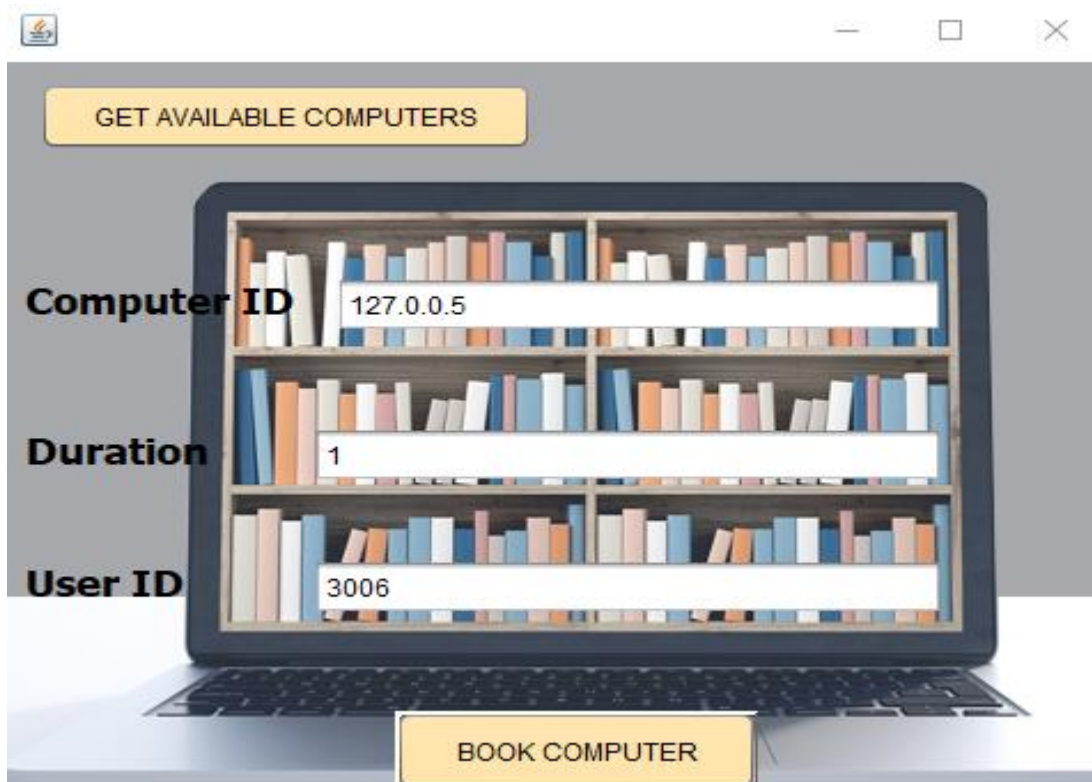


Input:

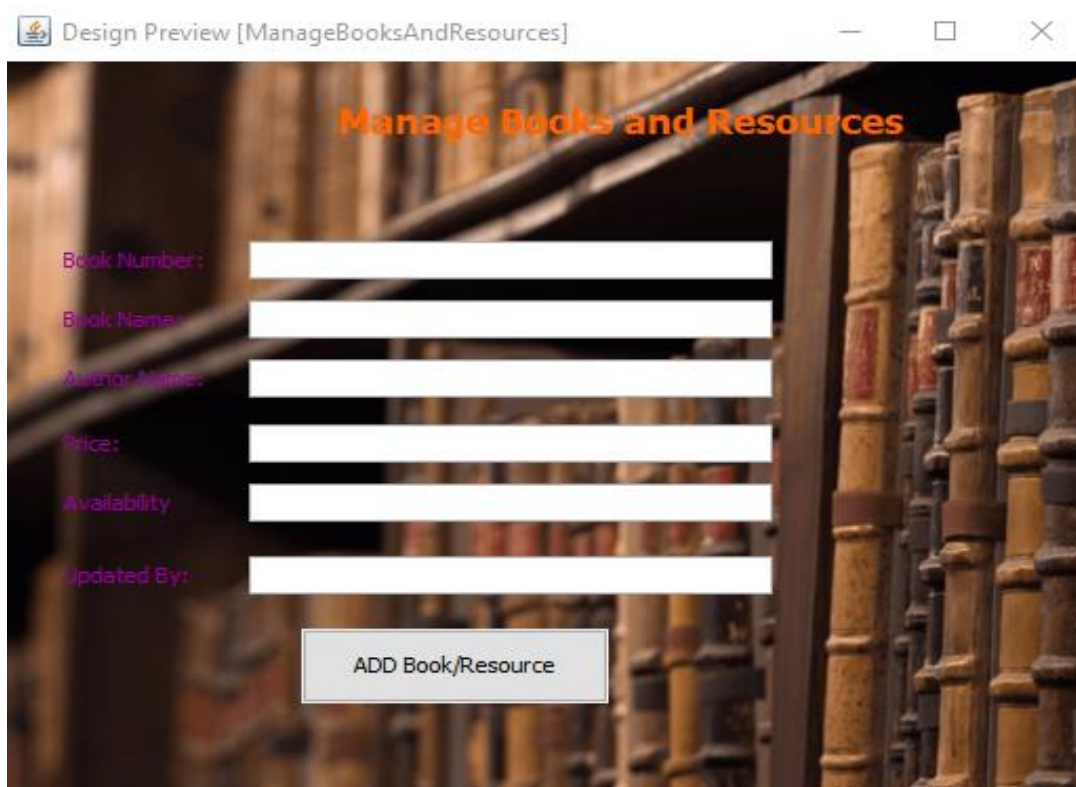
In Computer ID text field: InternalComputers.IPADDRESS=127.0.0.5

In Duration text field: InternalComputers.SESSIONDURATION=1

In user ID text field: InternalComputers.USERID=3006



4. Manage Books And Resources:



Input:

In Book Number text field: BooksAndResources.BR_NUM=300

In Book Name text field: BooksAndResources.BR_NAME=Weathering Heights

In Author Name text field: BooksAndResources.AuthorName=E.A

In Price text field: BooksAndResources.Price=200

In Availability text field: BooksAndResources. Availability =1

In Updated By text field: BooksAndResources. updatedBy =2000

Design Preview [ManageBooksAndResources]

Manage Books and Resources

Book Number:	300
Book Name:	Weathering Heights
Author Name:	E.A
Price:	200
Availability	1
Updated By:	2000

ADD Book/Resource

5. Members Window

WELCOME

Search For Books

Choose Event to see details:

Add Complain/ Suggestion

Sumbit