**Class name: AccountMaintenance**

**Methods to be tested**:

1.isIssued

2.updateIssuedBooks

3.updateReturnedBooks

4.booksHistory

5.updateLoginActivity

6.updateLogoutActivity

7.displayLoginActivity

8.recommendBook

**Method 1**: isIssued

**File name**: BookTransactionData.csv

**Input**: Name of the book and the email of the user

**Output**: If the book is issued to the user, it returns Boolean true else it returns boolean false

**Method 2**: updateIssuedBooks

**Input:** The name of the book that is to be updated to the books issued data and the email of the user

**Output**: Name of the bookissued and the date of issueare added corresponding to the user email in the file

**Method 3:** updateReturnedBooks

**File name:** BookTransactionData.csv

**Input:**The name of the book that is to be updated to the books returned data and the email of the user

**Output:** Name of the book returned and the date of returnare added corresponding to the user email in the file

**Method 4:** booksHistory

**File name:** BookTransactionData.csv

**Input:** email of the user

**Output:**

1. (If books are issued)

Books issued are (list of books)

Book name Issued Date

1. (if no books are issued)

No books are issued

1. (if books are returned)

Books returned are

Book name Issued date

1. (if no books are returned)

No books are returned

**Method 5:** updateLoginActivity

**File name:**LoginActivity.csv

**Input**: email of the user

**Output:**Date and time of login are updated in the LoginActivity.csv file

**Method 6:** updateLogoutActivity

**File name:** LoginActivity.csv

**Input:** email of the user

**Output**: Date and time of logoutare updated in the LoginActivity.csv file

**Method 7**: displayLoginActivity

**File name**: LoginActivity.csv

**Input:** email of the user

**Output:**

Login :(date and time in dd-MM-yyyyHH.mm.ss format)

Logout: (date and time in dd-MM-yyyyHH.mm.ss format)

**Method 8:** recommendBook

**File:** RecommendedBooks.csv

**Input:** email of the user

**Output:**

Name of the book, Author name, ISBN number, Publisher Name of the books recommended are displayed.

--------------------------------------------------------------------------------------------------------------------------------------

**Class name: LibraryCaseStudy**

**Methods to be tested**:

1.run

2.search

**Method 1**: run

**Test case1**:(for user login)

**Input**: valid registered email and valid password

**Outpu**t: Login successful

**Test case 2**: (for user login)

**Input**: invalid email

**Output:** Enter a valid email

**Test case 3**:( for admin login)

**Input:** valid email and password

**Output**: login successful

**Test case 4**: (for admin login)

**Input:** invalid email

**Output:** enter a valid email

**Test case 5:** (for admin login)

**Input**: valid email and invalid password

**Output**: entered incorrect password

**Test case 6:** (For admin login)

**Input:** incorrect email

**Output:** you are not admin

**Test case 7**: (for register)

**Input:** valid user name, email id, year of study, branch, roll number, password

**Output**: user registration completed

**Test case 8:** (for register)

**Input**: already registered details

**Output**: user already registered

**Method 2**: search()

**Test case 1:**

I**nput**: title of book

**Output**: admin menu is displayed

**Test case 2:**

**Input:** ISBN of book

**Output:** admin menu is displayed

**Test case 3:**

**Input:** book author

**Output**: admin menu is displayed

**Test case 4:**

**Input**: book publisher name

**Output**: admin menu is displayed

**Class name: Menus**

**Methods to be tested:**

1.mainMenu

2.displaySearchMenu

3.displayAdminMenu

**Method 1**: mainMenu

**Input**: email of the user

**Output**:The main menu containing operations like search a book, reserve a book, return a book, helpand feedback, recommend a book are displayed.

**Method 2**: displaySearchMenu

**Input**: email of the user

**Output**: displays the search menu to the user. Based on the option selected by the user respective details are entered by the user.

**Method 3:** displayAdminMenu

**Input**: email and password of admin

**Output:** admin menu is displayed which is only visible to admin

--------------------------------------------------------------------------------------------------------------------------

**Class name: BookDatabase**

**Methods to be tested:**

1.searchByTitle

2.searchByAuthor

3.searchByISBN

4.searchByPublisher

5.reserveBook

6.returnABook

**Method 1**: searchByTitle

**File name:** booklist.csv

**Test case 1:**

**Input:**Title of the book which is in booklist.csv

**Output:**details of the book are printed

**Test case 2:**

**Input**: title of the book which is not present in booklist.csv

**Output:** Book is not available

**Method 2**: searchByAuthor

**File name**: booklist.csv

**Test case 1:**

**Input**: Author of the book which is in booklist.csv

**Output:** details of the book are printed

**Test case 2:**

**Input**: Author of the book which is not present in booklist.csv

**Output**: Book is not available

**Method 3**: searchByISBN

**File name:** booklist.csv

**Test case 1:**

**Input:**ISBN of the book which is in booklist.csv

**Output:** details of the book are printed

**Test case 2:**

**Input**: ISBN of the book which is not present in booklist.csv

**Output:** Book is not available

**Method 4:** searchByPublisher

**File name**: booklist.csv

**Test case 1:**

**Input:**Publisher of the book which is in booklist.csv

**Output:** details of the book are printed

**Test case 2:**

**Input**: Publisher of the book which is not present in booklist.csv

**Output**: Book is not available

**Method 5:** reserveBook

**File name:** booklist.csv

**Test case 1** : (book is available)

**Input**: Name of the book which the user wants to reserve

**Output:**Book reserved successfully

**Test case 2**: (book is not available)

**Input:**Name of the book which the user wants to reserve

**Output:** book is not available currently

**Method 6:** returnABook

**File name:**BookTransaction.csv

**Test case 1:**

**Input**: Name of the book(which is issued)

**Output**: Book returned successfully

**Test case 2:**

**Input:** Name of the book(which is not issued)

**Output:** Book has not been issued to you

**Class name: Database**

**Methods to be tested**:

1.userLogin

2.isRegistered

3.registerUser

**Method 1**: userLogin

**Input**: email and password of the user

**Output**: If both email and password are correct returns 1. If not returns -1

**Method 2**: isRegistered

**Input:**email of the user

**Output**: returns true if the user is registered. Otherwise, returns false

**Method 3**: registerUser

**Input**: name, roll number, branch, email, year, password

**Files**: Data.csv, detailsFile.csv

**Output:**

email and password are stored in Data.csv file in encrypted format

Remaining details are stored in detailsFile.csv

--------------------------------------------------------------------------------------------------------------------------

**Class name: Librarian**

**Methods to be tested**:

1.run

2.displayStudentDetails

3.addBook

4.displayRecommendedBooks

**Method 1**: run

**Input**: choice entered by librarian from the menus displayed

**Output:** Based on the choice entered,respective statements are displayed

**Method 2:** displayStudentDetails

**File name:** DetailsFile.csv

**Test case 1:**

**Input**: name of the student(present in file)

**Output**: Details of the student are displayed

**Test case 2:**

**Inpu**t: name of the student (not present in file)

**Output:** Student details not found

**Method 3:**addBook

**File name:** booklist.csv

**Input:** Name of the book, Author, ISBN, Publisher, available copies

**Output:** The details of the book will be added to the file

**Method 4**: displayRecommendedBooks

**File name:**RecommendedBooks.csv

**Input:**admin selects the option from the menu

**Output**: Details of the books recommended by the user along with the user email are displayed

--------------------------------------------------------------------------------------------------------------------------

**Class name: Validation**

**Methods to be tested:**

**1.**validateUserName

2.validateYear

3.validateEmail

4.branchValidation

**Method 1**:validateUserName

**Input**: username

**Outpu**t: returns true if the username is valid. Otherwise, returns false .

**Method 2**: validateYear

**Input:** year of study

**Output:** returns true if the year is valid else returns false

**Method 3**: validateEmail

**Input:** returns true if the email is valid else returns false

**Method 4:** branchValidation

**Input:** branch

**Output:** returns true if the branch is valid else returns false