DOKUZ EYLÜL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 2210

Object Oriented Analysis and Design

Students Dormitory System

by

YASSER EL HASAN

2019510006

CHAPTER ONE

INTRODUCTION

Mainly, this project is a system to perform the basic tasks of student dormitory starting from the room ending in the gym. It allows admins to register new students and employees. Also, there are internal parts within the dormitory as following:

- **Library:**

  The system has a library that has an employee (librarian) who is responsible for the operation of lending and returning the books within the determined time. It has many books in different majors, each book has its information like name, author, major, and date of the book.

- **Restaurant:**

  This part shows the available meals for students and some information about each kind of food.

  It has a chef who can control the operations that are included in this part.

- **Gym:**

  This part is for students who are interested in doing sports, they can register for the sports that they prefer

  It has an employee who controls the registration operation (Trainer), he can also access the students' information who are registered in the gym.

- **Absence List:**

  This part is especially for students to request permission from the dormitory's administration if they want to leave the dormitory for special cases.

- **Dashboards:**

  The system has two main dashboards, one for the admin who can use it to access all parts of the dormitory, and the second one for students to have an access to the parts specialized for them within the dormitory.

CHAPTER TWO

REQUIREMENTS

This System will be used by three kinds of users, admin, student, and employee.
There is a common login page to allow the users to access the system depending on the user's kind.

- **Admin:**

  After the admin login or do a new registration, will access the admin dashboard which allows him to do different action within the system. He can access the parts related to students and employees. He can add new students and employees, update their information or delete them.

  He can access the parts restaurant, library, and gym.

- **Student:**

  Students, after they log in or sign in, can access the student dashboard which allows them to deal with many parts like restaurant, library, and gym.

  He can add a new meal to his daily food list and can access the library to show him the available books in the library so he can borrow a new book or return a book to the library. Students can also register for a new sport in the gym and take training from the trainer of the gym. Also, the student can request from the admin permission to leave the dormitory.

- **Employee:**

  There are two kinds of employees who are **librarians** and **chefs**. they can are added by the admin, they can only log in not sign up for new registration.

  - The librarian can access the library part, can add or delete books from the library and he can see the student who borrowed aa books from the library.
  - The chef has the access to the restaurant, where he can add, update or delete food from the list. He can see what meals that student registered for.

Method	Type	Parameter	Return	Description
Authenticate()	Boolean	Email,password	True or false	Used for login
Add()	void	Object of person	-	Abstract method
Delete()	void	Object to delete	-	Abstract method
Update()	Boolean	Object to be updated	True or false	Abstract
Show()	List	-	List of Persons	To show persons
AddBook()	void	Object of Book	-	To add a new book
DeleteBook()	void	Object of Book	-	To delete a book
BorroweBook()	void	Object of Book	-	To borrow book
UpdateBook()	void	Object of Book	-	To Update a Book
ReturnBook()	void	Object of Book	-	To return a book
ShowBooks()	List	-	List of Books	To show books
AddMeal()	void	Object of Meal	-	To add a new meal
DeleteMeal()	void	Oject of Meal	-	To delete meal
ListMeals()	List	-	List of Meals	To show meals
UpdateMeal()	void	Object of Meal	-	To update meal
RegisterForGym()	void	Object Of Student	-	Register for gym
ShowMembers()	List	-	List of Members	To show Members
RequestAbsence()	void	Start and finish time	-	To request absence
BookRoom()	void	Student	-	To book room
ShowRooms()	List	-	List of rooms	To show rooms

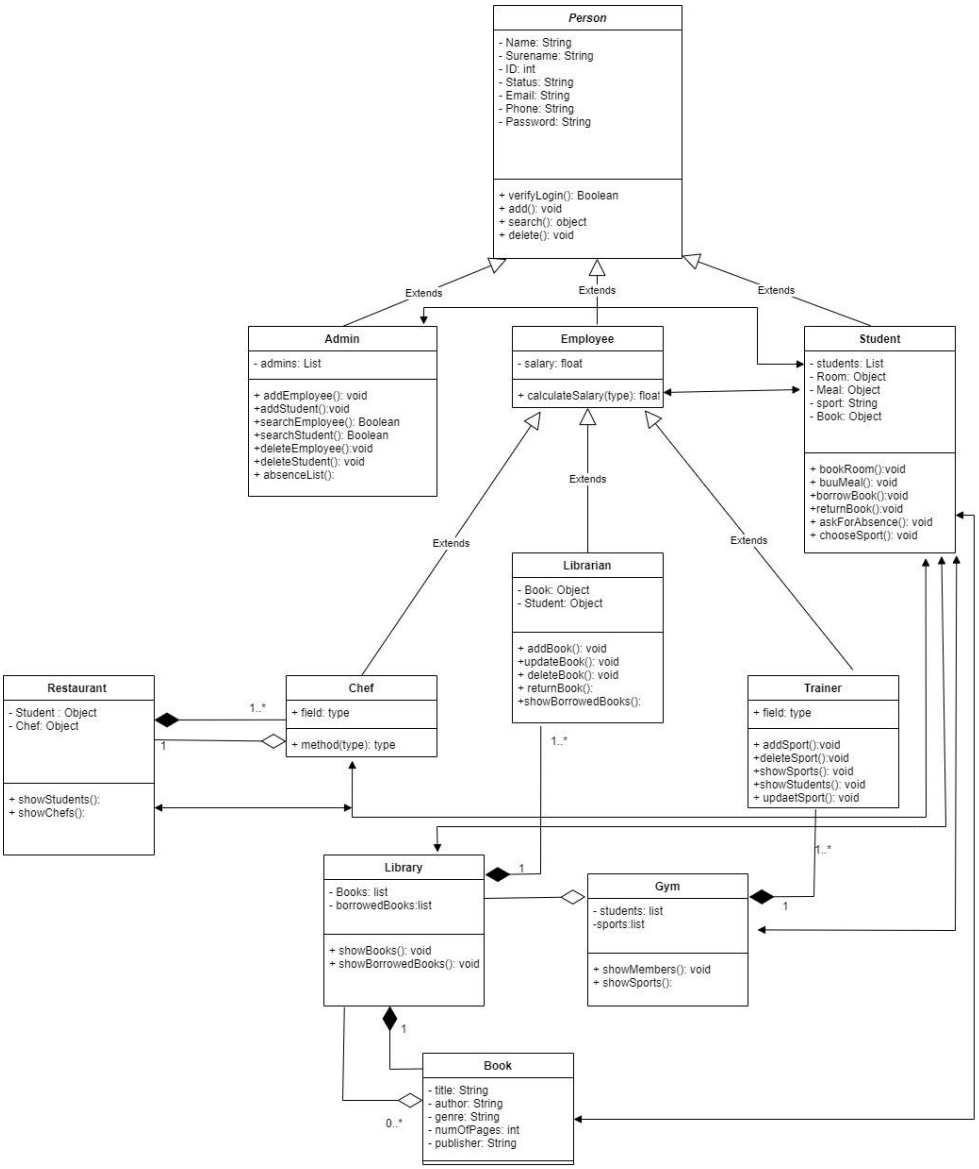
- **Note**: these are methods that I am to use until this moment, it can be more than these, or some of them can be deleted.

CHAPTER THREE

UML DIAGRAMS

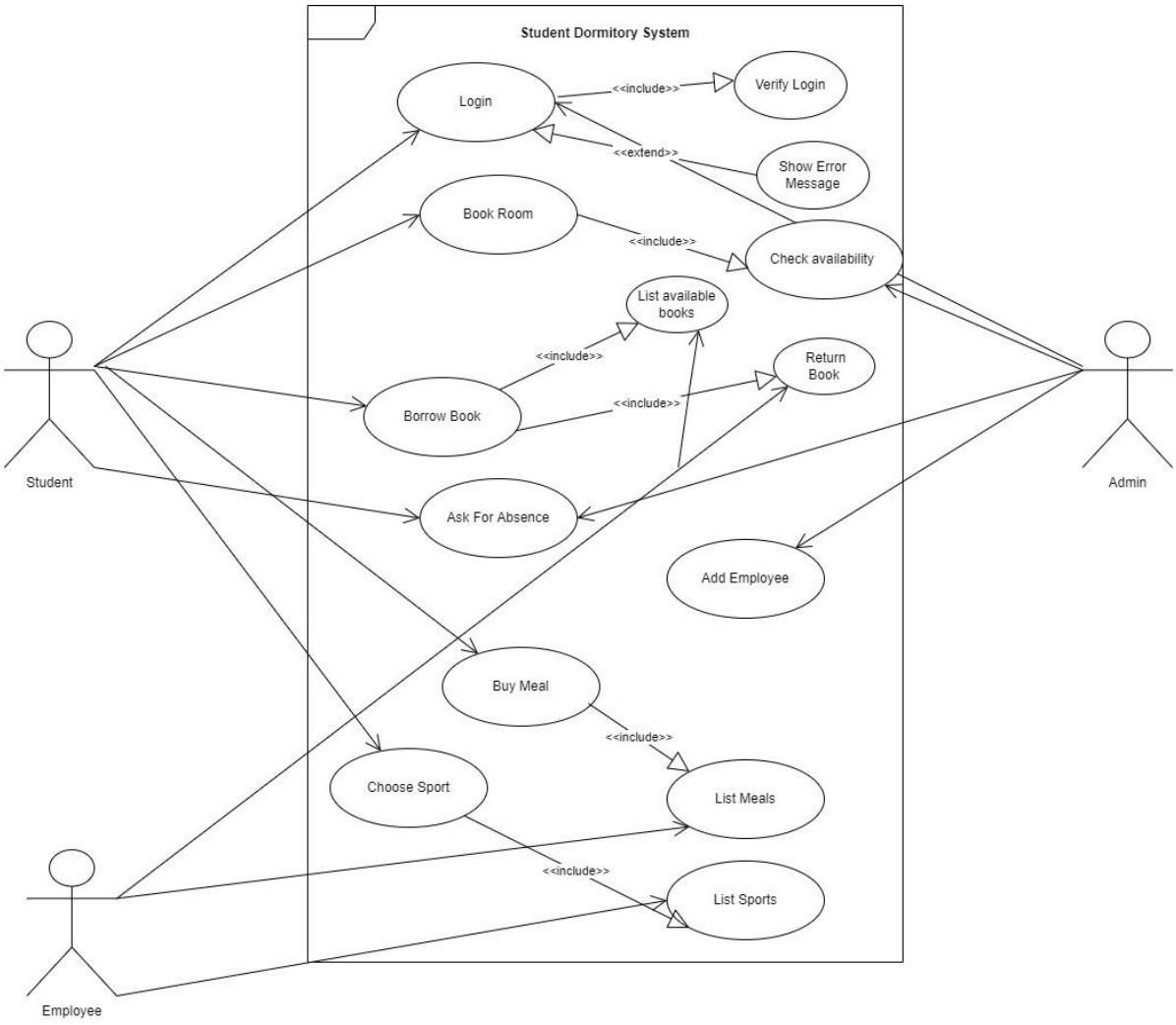
1. **Class Diagram:**

As we can see from the class diagram the system consists of many classes that have relations with each other. It has the Person class which is an abstract class that has the main attributes of persons who are included in the system like the Student, Admin, and Employee, where these classes extend from the Person class. There are also Chef, Trainer, and Librarian Classes which extend the Employee class. The system has Library, Restaurant, and Gym classes in addition to Book and Meal classes. We can notice the different relations between classes from the diagram below.



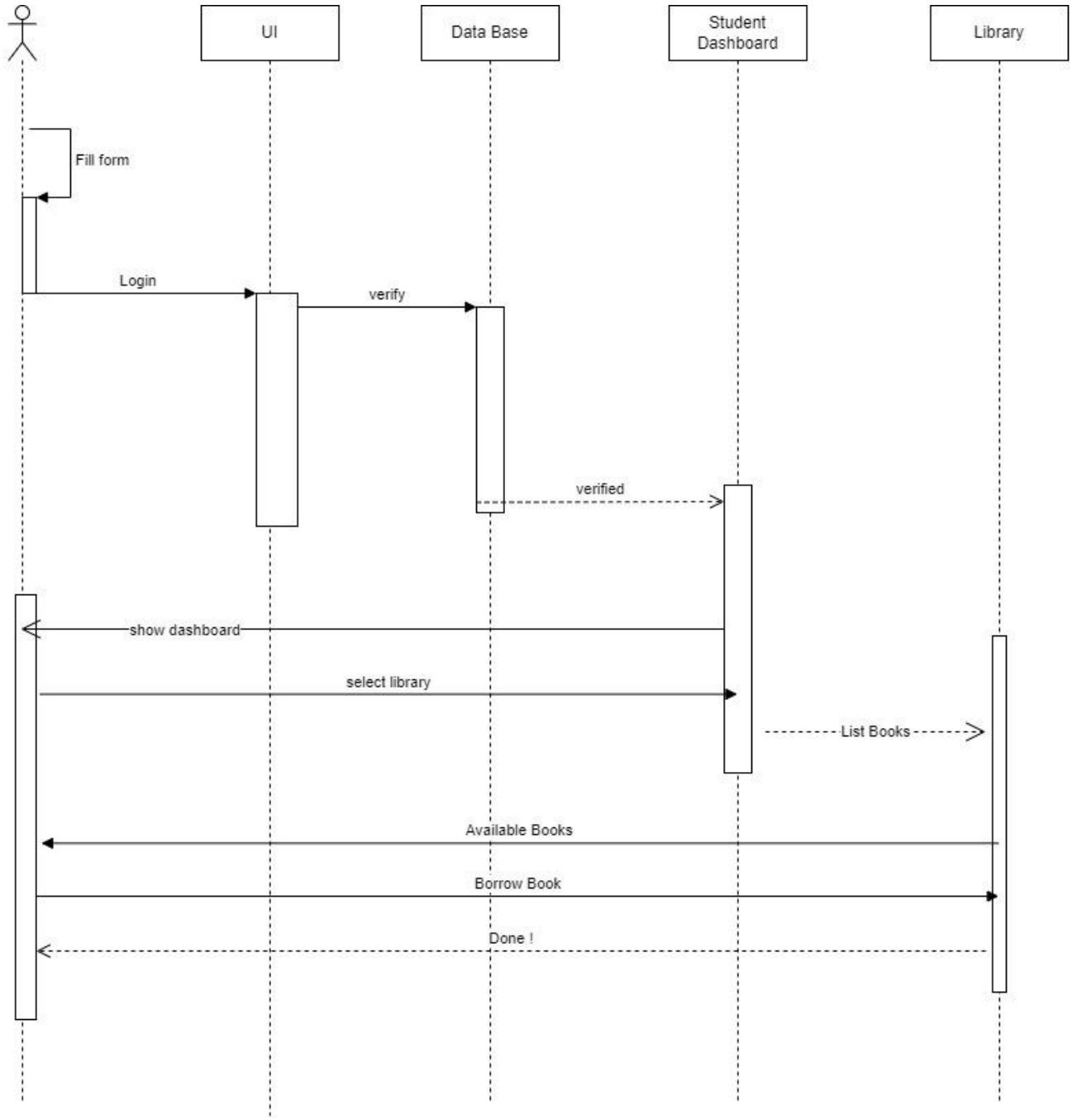
2. **Case Diagram:**

This diagram mainly shows the main operations that can be performed by the system, and it gives the user the way how to use the system. It has many actors like Student, Employee, and Admin. In the frame of the system, we have many basic operations like Login, Bookroom, Borrowe book, Asking for absence, Buying meal and choosing sport. Where these are related to the Student ,Employee and Admin actors. For the details, we can check the diagram below.



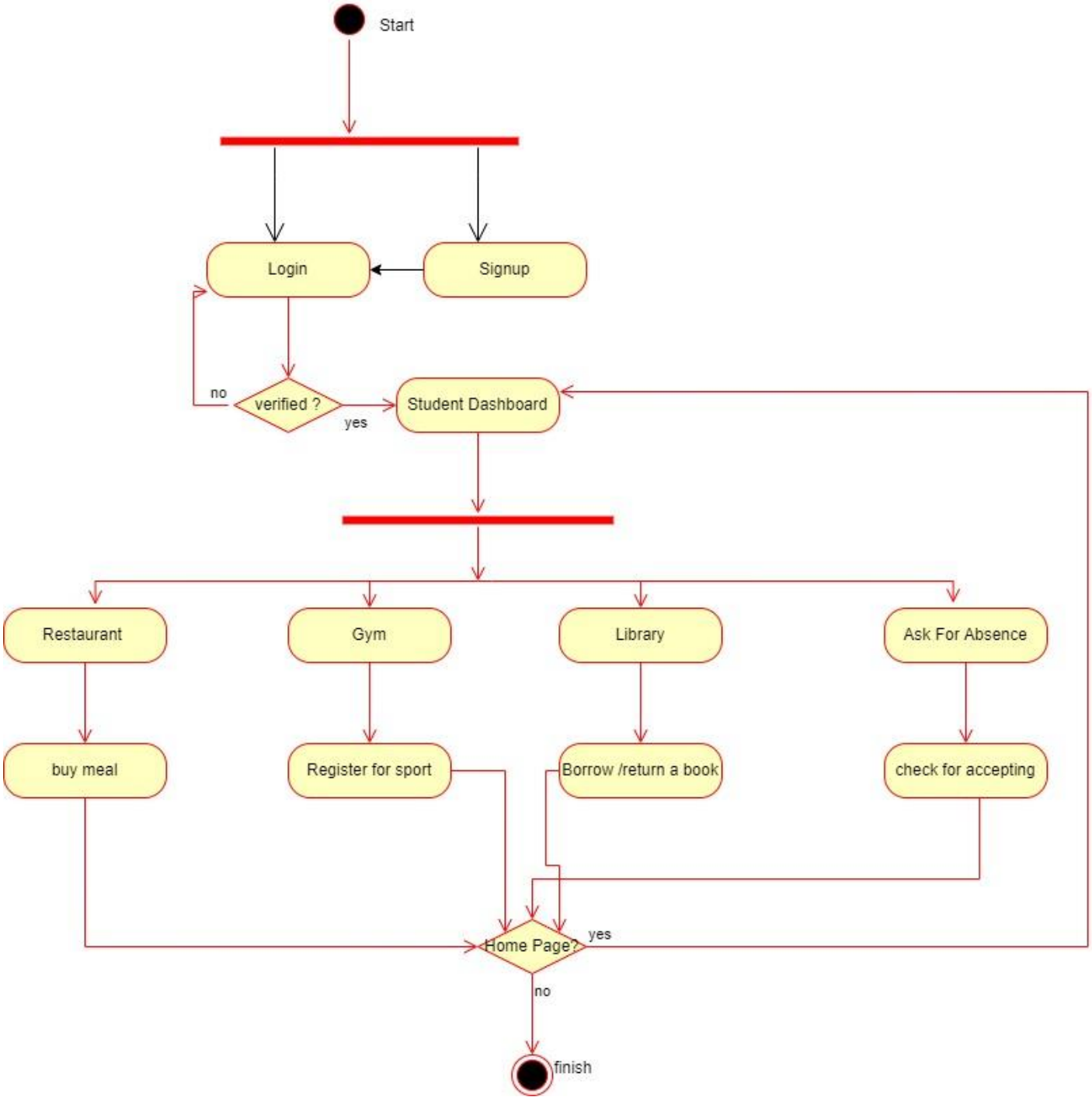
3. **Sequence Diagram:**

Basically, this diagram represents the sequence of operations within the system , in my case I am going to show only Borrowing and Returning operations. Where the diagram shows how the user starts by opening the UI screen and asking the user to fill the required information for login operation, after that the system checks the information by sending them to the database. After verifying user login, the system shows the student dashboard which has the Library part, then the student chooses Library and asks to list the available books in the library to borrow a book, or to return a book.



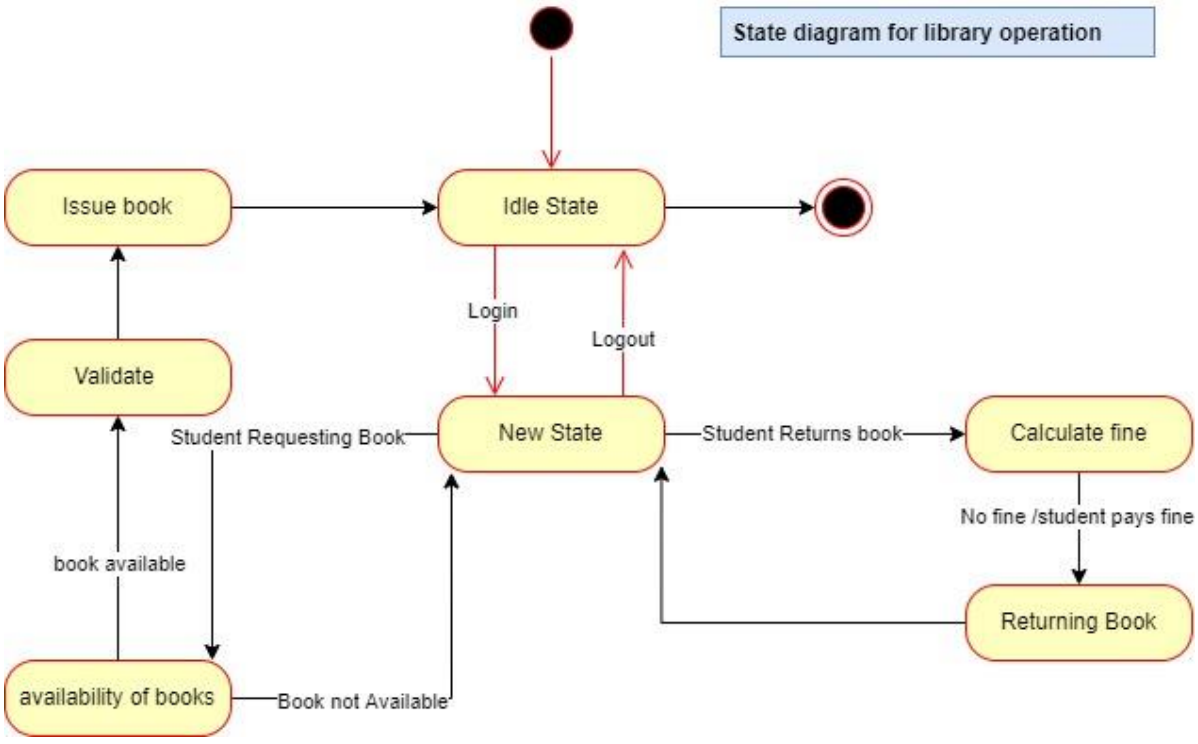
4. **Activity Diagram:**

Activity diagram shows the main activities of the system, depending on the logical sequence of the happening. In case the user is Student, I am showing how the system starts with the first activity which is Login or Signup. Then, we have parallel activities that can happen at this point, which are Asking for Absence, Restaurant, Library, Gym. Where a student can perform the different operations within them after that he chooses to go back homepage or exit from the system.



5. **State Diagram:**

State diagram is so similar to the last diagram but basically considers the operations within the system as a state where the user can move in between depending on the logical choices. In my case, I consider the user is a student and he wants to perform the library operations. The system starts with idle state which is before login or signup operations. Depending on the user information, the student can move to the second state which borrows book state where the system checks the availability of books and allows the user to choose, or to return a book where he moves to the state of paying fines. After these operations, the user moves to the last operation to exit from the system.



CHAPTER FOUR

IMPLEMENTATION

1. UI CLASSES:

The project has 16 windows that control the flow of the application to achieve the available features of the project.

- **AdminDahboard** has 8 buttons, each one has a click listener to direct the user to the required window of the application
- **StudentDashboard** has 6 buttons that allow the user to open the available parts of the project
- **AbsenceRequestList** this window has some components like Jtable shows the kist of the absence requests asked by students, and 2 buttons to react with requests (accept and reject)
- **Bedrooms** window has a Jtable that shows the rooms and their status
- **The EmploymentManagement** window which contains many text fields to allow the admin to add, update or delete employees, in addition to a Jtable shows the list of the employees. Additionally has 3 radio buttons to select the position of the employee.
- **Gym** window that has 2 Jtables , one to show the available sports in the gym, and the other one to show the registered students in the gym and their information. And contains TextLable that shows the photo of the selected sport from the table. And this window has two TextFields and buttons that allow the Trainer to add, update, delete or browse the files to upload a photo to the application.
- **Library** window that has 2 Jtables to show both, the available books in the library and the list of the students who borrowed books from the library. And there are Jlables to show the book information.

- **MyAbsences** window has 2 buttons to request an absence and delete a request.in addition, the Jtable shows the list of requests.
- **MyBooks** window is special to the students, it has 2 buttons to borrow and return books. in addition, it has 2 Jtables that show the lists of library books and borrowed books.
- **MyMeals** window has 2 buttons to add and remove meals. and Also has 2 tables that show the available meals and meals added by students. And has Jlables that show the photo of the meal.
- **MyRoom** window has a button to book a room and has Jlables to show the information of the booked book.
- **MySports** window has 2 buttons to add and delete a sport and 2 Jtabled that show the available sports in the gym and the sports added by the student.
- **Register** window that allows the new users to register to the system. it has many TextField to collect personal information and has 2 radio buttons to allow the user to select depending on his case.
- **Restaurant** window has 2 Jtables that show the available meals in the restaurant and a list of the students who took meals from the restaurant, in addition to buttons to add, update and delete a meal. And also the window has a Jlable to show the photo of the meal.
- **Students** window to allow admin to add, update and delete students using the buttons and the Jtable that shows the registered students.
- **WelcomePage** window has 2 Text fields to collect user information and has 2 buttons to login or register, and additionally has 3 radio buttons to select depending on the user case.

2. MAIN CLASSES:

- **Person Class** is the upper class of many classes that extend it like (Student, Employee, Admin). It has personal attributes like (name, surname, email, password, and phone) in addition to the getters and setters methods.
- **Employee Class** which also is the upper class of 3 classes(Chef, Librarian, and Trainer). It has many constructors.
- **Librarian Class** has many functions like **ReadBooks()** which read a database of books, **writeBook()** which always store the books list in the database, **CheckExistence()** which checks if a book exists in the library, and **AddBook()**is used to add a new book to the library, **ReadRegisteredStudents()** that returns the list of students who borrowed books from the library.
- **Chef Class** has many methods like **readMeals()** that read the data from meals from the database, **addMeal()** to add a new meal to the list, **deleteMeal()** delete a meal from the list, **isMealExists()** to check a meal if exists in meals list, **readRegisteredStudents()** to return the list of students who took meals from the restaurant.
- **Trainer Class** has methods like **readSports()** and **writeSportsData()** to read and write sports data from and to database, **addSport()** and **deleteSport()** to add and delete sports from list, **isSportExists()** to check the existence of a sport, **readRegisteredStudents()** to return the list of students who are registered to the gym.
- **Book Class** has the attributes of a book like name, author, number of pages, genre, and publisher. In addition to the getters and setters methods.
- **Meal Class** has the name of the meal and the path of photo attributes in addition to setters and getters methods.
- **Sport Class** has the name of the sport and the path of photo attributes in addition to setters and getters methods.
- **Room Class** has roomId, studentEmail, roomStatus, Date attributes in addition to **readData()** and **writeData()** to read and store data from and to the database. In addition to setters and getters methods.

- **Admin Class** has **authenticate()** method , readData() and writeData() methods to deal with database, **addEmployee()** to add a new employee, **getEmStatus()** to return the position of the employee, **checkExistence()** to check the existence of an employee within the list, **deleteEmployee()** to delte an employee , **add()** to add a new admin to the system.
- **Student Class** has the **readStiudentData()** and **writeStudnetData()** methods to deal with the database, **checkExistence()** to check the existence of a student within the system, **authenticate()** to check the login information of the student, **currentLogin()** and **currnetSession()** these methods are to get the information of the current student login within the system**, bookRoom()** to book a room in the dormitory, **emptyRoom()** to empty the room and change the status of the room after deleting the student, **updateRoom()** to update the information of a room **hasRoom()** to check if a student has room or not, **getRoomByEmail()** returns a room information depending on the student email, **borrowBook()** to borrow a new book from the library, **writeMyBooks()** to write the data of student books to the database, **readBorrowedBooks()** to get the information from database, **returnBooks()** to return a list of the books of a student, **deleteBook()** to delete a book from student books , **isBookExists()** check if student has a bbook or not, **haveMeal()** to get a meal from the restaurant, **writeMyMeal()** write the data of stuudents meals to the database, **readMyMeals()** to get data from database, **returnMeals()** to return a list of meals depending on the given email, **isMealExistence()** to check the existence of a meal, **deleteMeal()** o delete meal from studet meals list, **addStudent()** and **deleteStudent()** to add and delete students to the system, **getByEmailStudent()** to return a student depending on the given email, **returnSports()** gives the sports of a student**, isSportExists()** to check if a student has a sport, **practiceSport()** to add a new sport to student sports list, **readMySpors()** to get the list of student sports from the datatbase, **writeSports()** to write data to the database, **deleteSport()** to delete sport from student sports list, **deleteStudentFromGym()** to delete a student from the gym , **updateStudentInGym()** to update a student in the gym ,**deleteStudentFromLibrary()** to delete student from the library,

UpdateStudentInLibrary() to update student in the library,**deleteStudentFromGym()** to delete a student from the gym, **updateStudentInGym()** to uodate a student in the gym

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

This system was performed using the java programming language. Swing library was used to build the UI windows to interact with the user. The system helps the institution to manage the student and employees within a student dormitory. It has many parts within the system like the library, restaurant, and the gym, in addition to a part to deal with absence requests from students. All data are read and written from the database, so data can be saved safely.