

**Course code**

**Course Name**

Phase 2: Software Design Document Team Name YM team

Full Name  **ميار السيد عبد الفتاح \يارا زكريا محمد** ID

203149

201462

# Dec 2020

December & 2020

**Contents**

[Instructions [To be removed] 3](#_bookmark0)

[Team 3](#_bookmark1)

[Document Purpose and Audience 3](#_bookmark2)

[System Models 4](#_bookmark3)

1. System Decomposition Error! Bookmark not defined.
2. [Class diagrams 4](#_bookmark4)
3. [Sequence diagrams 5](#_bookmark5)

[Class - Sequence Usage Table 7](#_bookmark6)

1. [Physical Entity-Relationship Diagram 8](#_bookmark7)
2. [User Interface Design 8](#_bookmark8)
3. [Algorithms and Data Structures 10](#_bookmark9)

[Ownership Report 10](#_bookmark10)

[Policy Regarding Plagiarism 10](#_bookmark11)

References **Error! Bookmark not defined.**

Authors **Error! Bookmark not defined.**

## Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 203149 | Mayar el-sayed abd al-phatah | Sim.mayarelsayed3149@alexu.edu.eg | 01208024293 |
| 203162 | Yara Zakaria Mohamed | Sim.yarazakaria3162@alexu.edu.eg | 01064469304 |

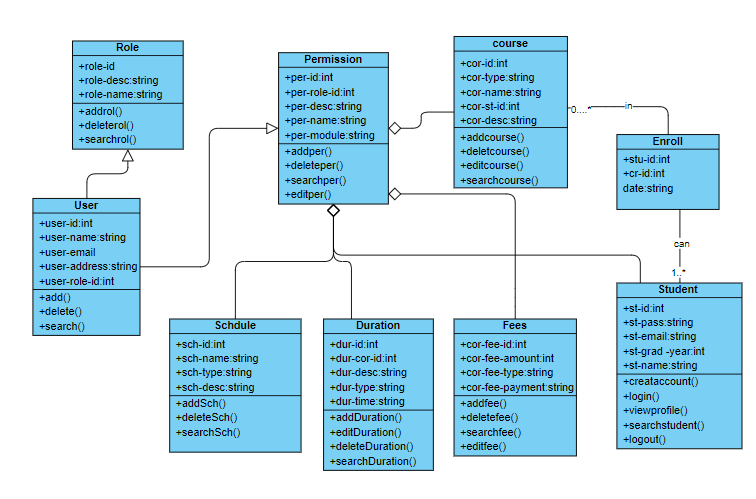
## Document Purpose and Audience

-E-learning management system make learning process more easy for individuals at any-time and any-place. E-learning has huge database which carries lots of student records, course records, course materials and so on.

-the student and the lecturer.

## System Models

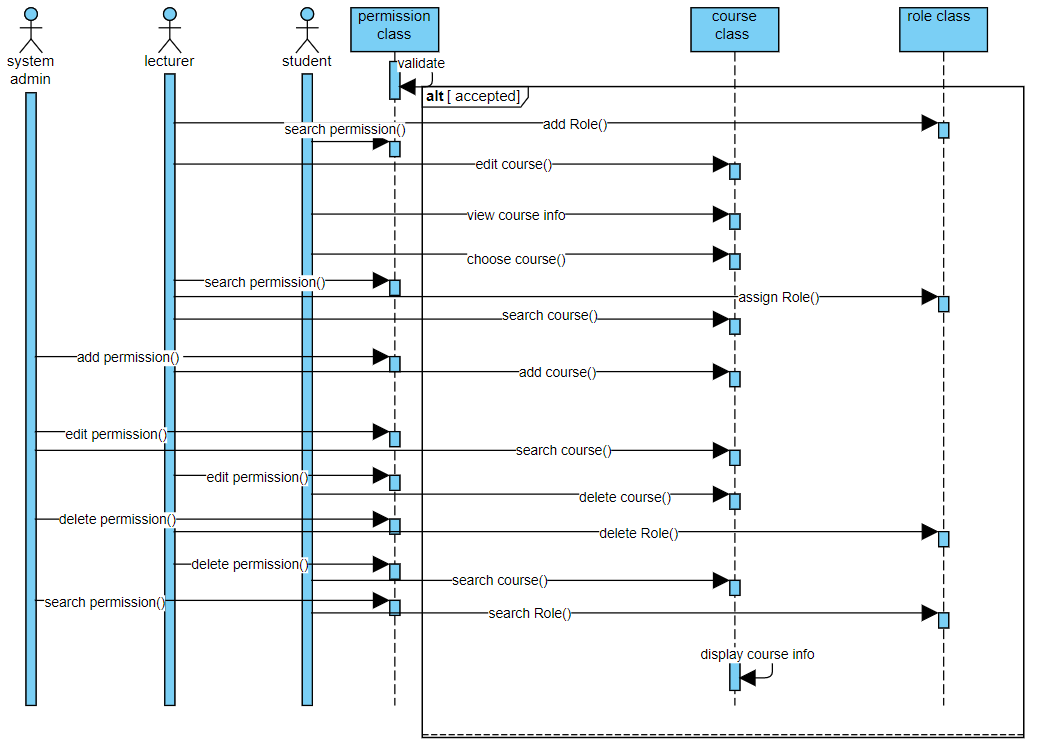
### Class diagrams



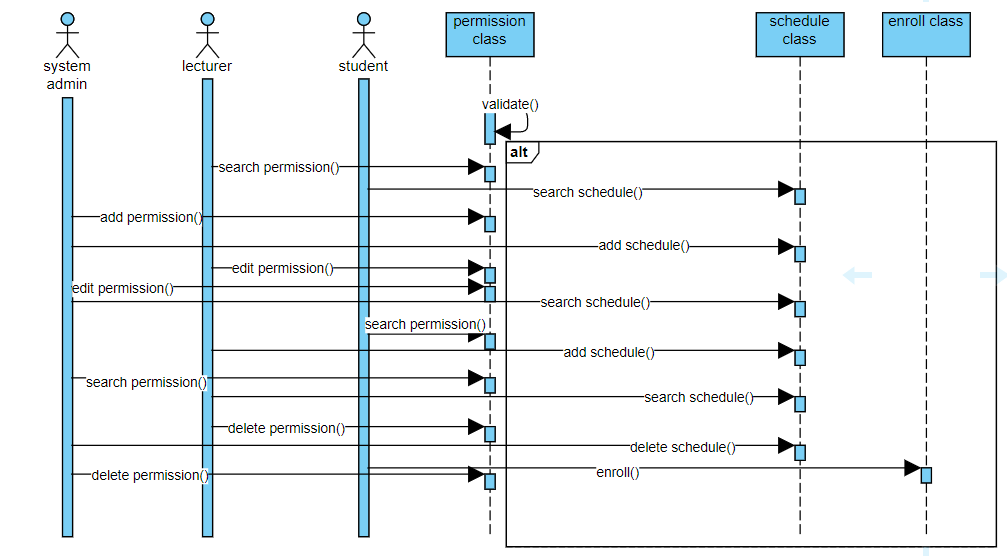
|  |  |  |  |
| --- | --- | --- | --- |
| **Class ID** | **Class Name** | **Subsystem ID** | **Description & Responsibility** |
| 00005 | Role Class | 1 | Each user has some roles.  This class contains all the operations that can be done to roles like: searching for a user role. |
| 00008 | User Class | 1 | To add or delete or search or edit information of user in system. |
| 00011 | Schedule Class | 1 | To allow lecturer to make schedule for his course and edit it. |
| 00077 | Fees Class | 1 | -Lecturer add fees of his course and edit it.  -Allow students to search about fees for each course. |
| 00066 | Enroll | 1 | To allow student enroll to any course he wants. |
| 00045 | Student Class | 1 | Each student can login to our system by using his username and password and view his profile and logout when he want |
| 00111 | Course Class | 1 | To allow each lecturer to add or edit or delete his courses and allow student to search about courses that he wants to take it. |
| 11022 | Permission Class | 1 | -To add or delete or edit permissions of this system.  -Allow users to search about this permissions. |
| 44005 | Duration | 1 | To allow lecturer to determine duration for his course and edit it if he wants. |

### Sequence diagrams

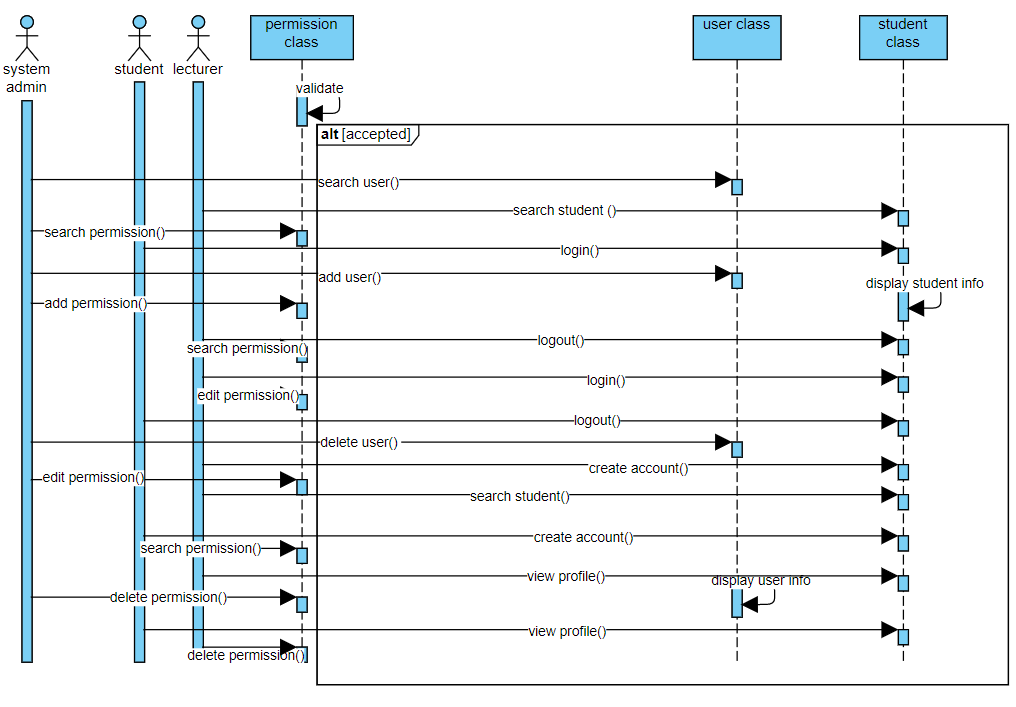
(1)



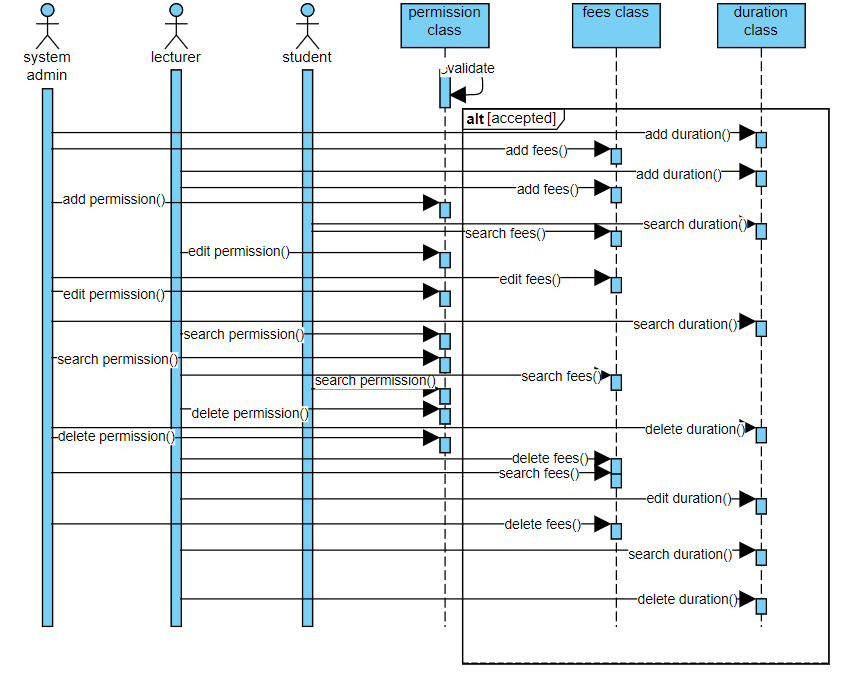
**(2)**

****

(3)



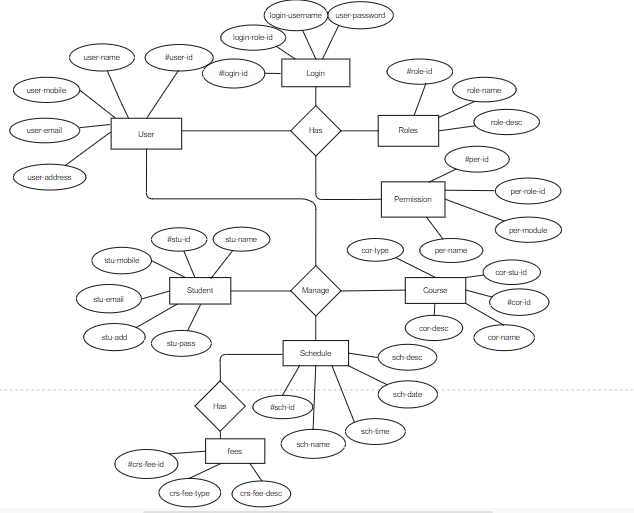
(4)



Class - Sequence Usage Table

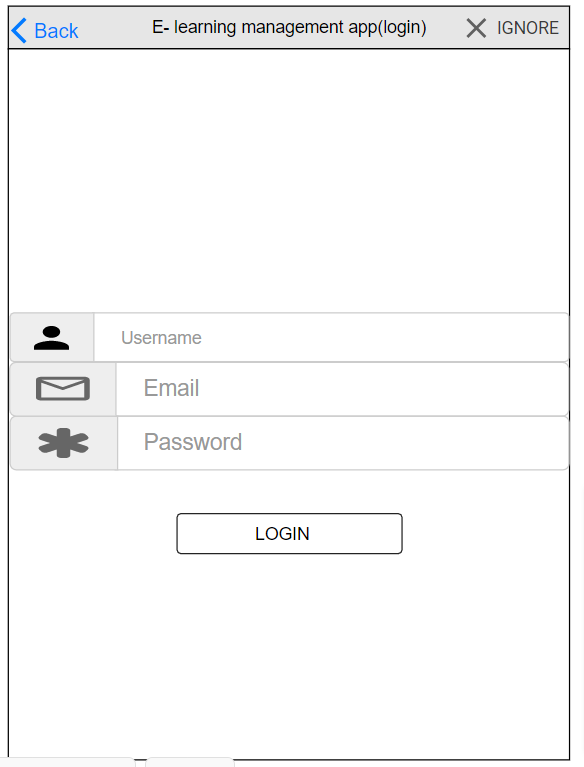
|  |  |  |
| --- | --- | --- |
| **Class name** | **Sequence diagram** | **Overall used methods** |
| User class | 3 | Add user ()  Search user ()  Delete user () |
| Student class | 3 | Add student ()  Login ()  Create account ()  View profile ()  Search student ()  Logout () |
| Role class | 1 | Add role ()  Search role ()  Delete role () |
| Permission class | 1,2,3,4 | Add permission ()  Edit permission ()  Search permission ()  Delete permission () |
| Course class | 1 | Add course ()  Edit course ()  Search course ()  Delete course () |
| Schedule class | 2 | Add schedule ()  Search schedule ()  Delete schedule () |
| Fees class | 4 | Add fees ()  Edit fees ()  Search fees ()  Delete fees () |
| Duration class | 4 | Add duration ()  Edit duration ()  Search duration ()  Delete duration () |
| Enroll class | 2 | Enroll () |

### Physical Entity-Relationship Diagram

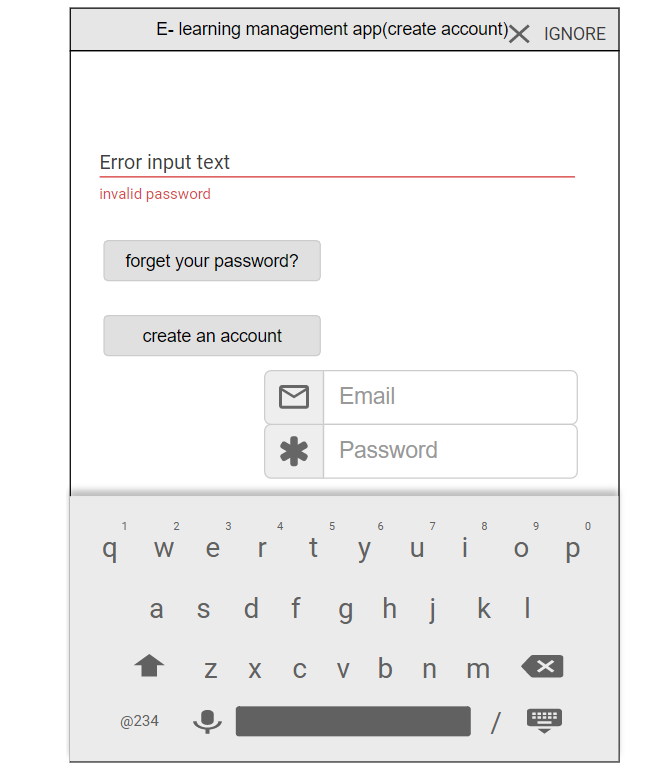


### User Interface Design

Screen 1 -login interface



* + **Screen 2- create account interface**



* + **Navigation tree:**

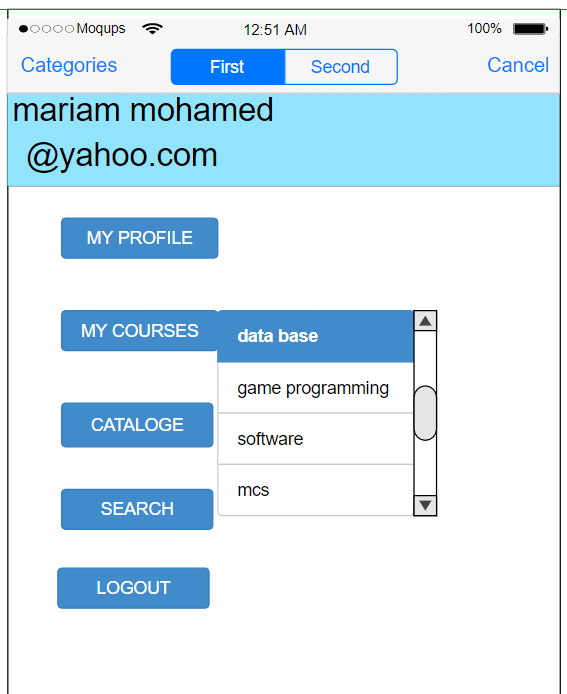
Login Screen

|

Create account

Screen

Screen 3-main interface



Navigation tree:

|  |
| --- |
| Login screen |

|

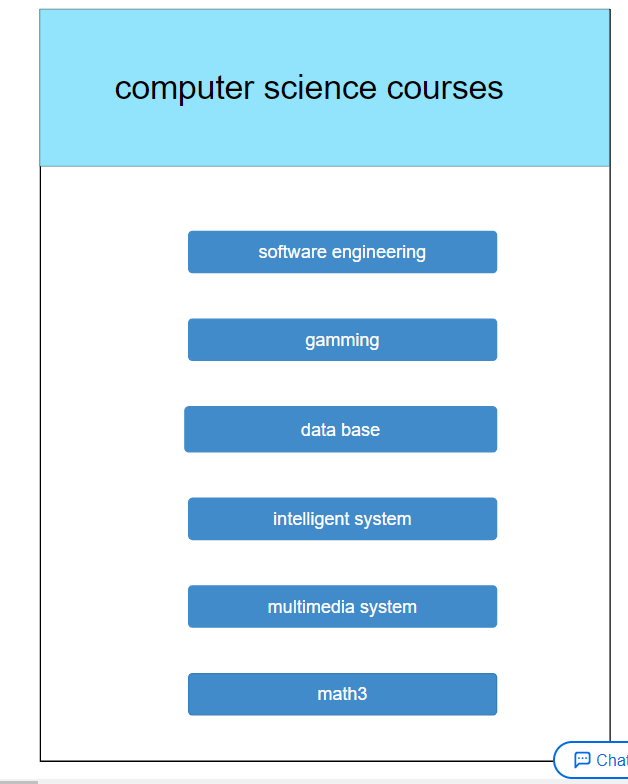
Create account

Screen

|

|  |
| --- |
| Main interface screen |

Screen 4- view courses interface



Navigation tree:

|  |
| --- |
| Login interface screen |

|

|  |
| --- |
| Create account interface screen |

|

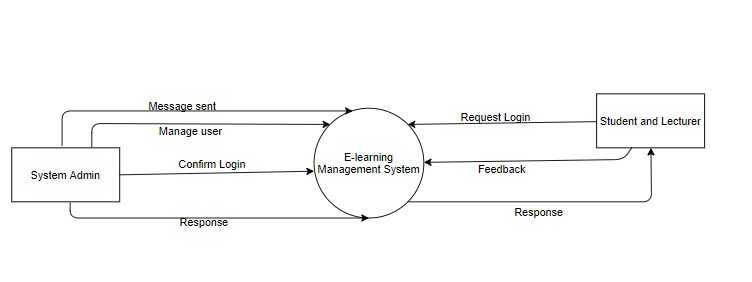
|  |
| --- |
| Main interface screen |

|

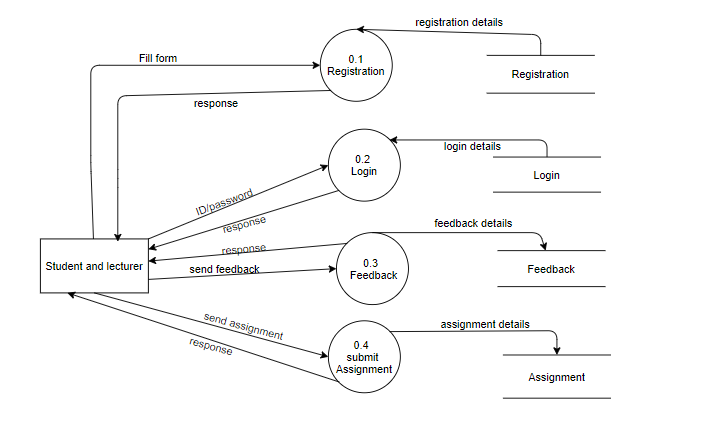
|  |
| --- |
| View courses interface screen |

### Dataflow diagram (DFD)

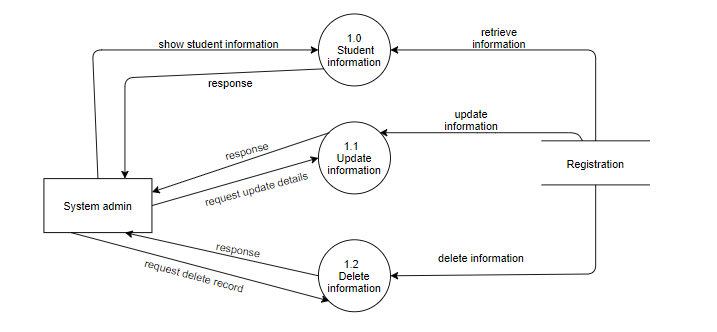
Context diagram (zero level):



Level (1):



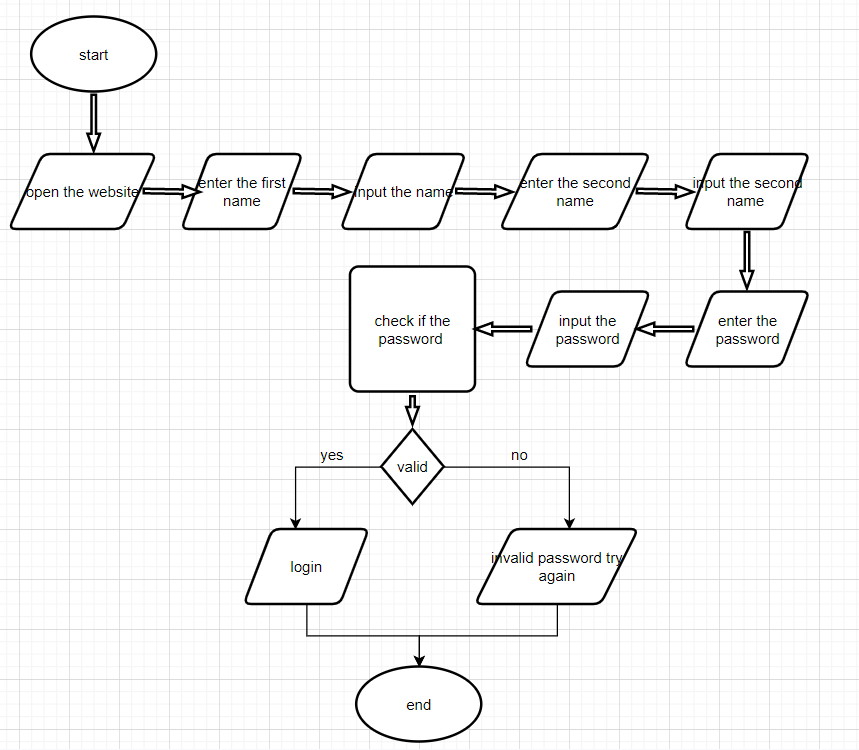
Second level:



### Algorithms and Data Structures

The system performs 4 main functionalities. The functionalities are Create, Retrieve/Search, Update, and Delete. They are applied to both the user and the courses. There are different types of users such as Professors, and Students but they are treated the same from the functionality point of view.

1. Create: after filling the information of the user/course we will save it in an Array of objects.
2. Retrieve/Search: to make the search faster we will sort the users/courses at first so we can use binary search to get specific user or course using its id.
3. Update: At first we need to perform search to get the required user or course using the technique in 2 and then update the information.
4. Delete: same as update but instead of update we will delete it and then shits the array elements to occupy the empty place.
5. Create

Algorithm

Step1: start

Step2: open “the website”.

Step3: output “enter your first name”.

Step4: input “the first name”

Step4: output “enter your second name”.

Step5: output “enter your password”.

Step6: input “input password”.

Step7: check if (the password is valid or not).

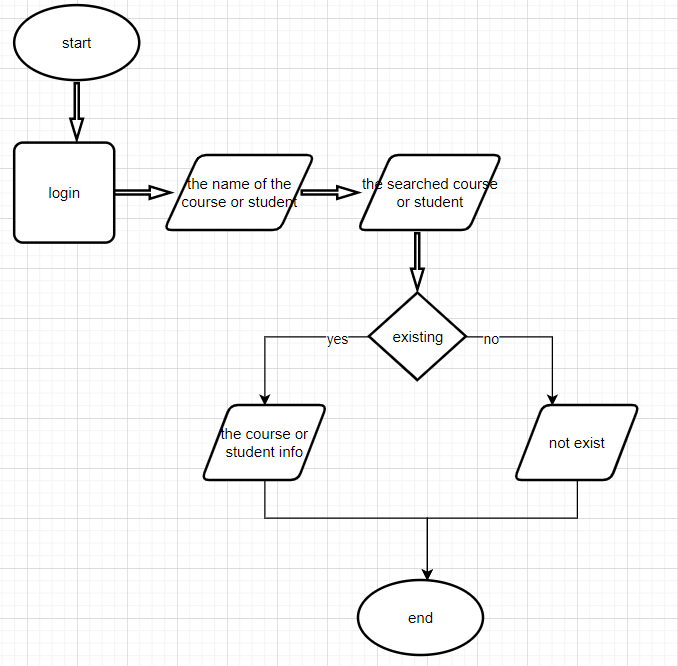
Step8: output “login”.

Step9: else.

Step10: output “invalid password please try again”.

Step11: end.

1. Search



Algorithm

Step1: start.

Step2: login.

Step3: output “enter the name of the course or the name of the student”.

Step4: input “input the name”.

Step5: if (the course or student are exist).

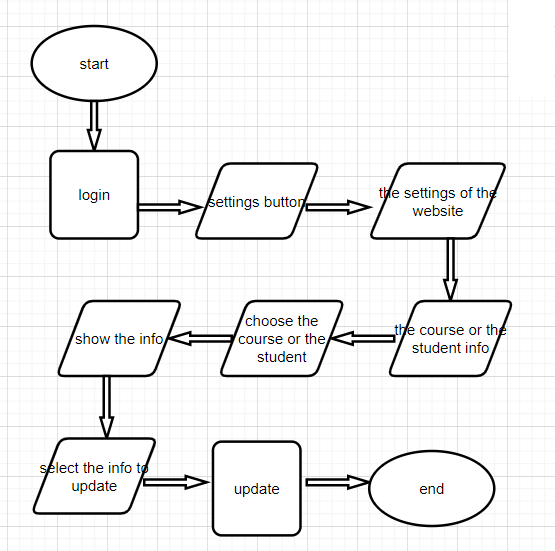
Step6: output “the course or student information”.

Step7: else.

Step8: output “the course or the student is not exist”.

Step9: end.

1. Update



Algorithm

Step1: start.

Step2: login.

Step3: output “setting”.

Step4: input “view the setting of the website”.

Step5: output “the course or the student information”.

Step6: input “choose the course or student information”.

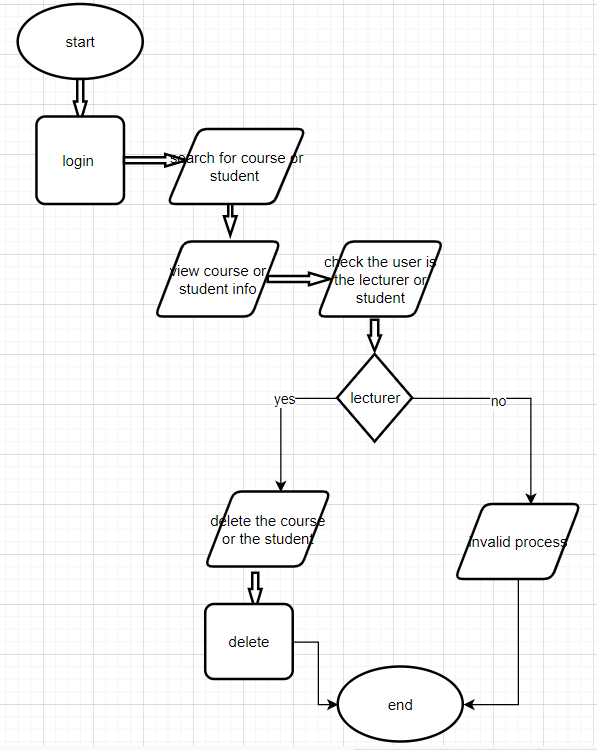
Step7: output “show the information”.

Step8: input “select the information you want to update”.

Step9: update.

Step10: end.

1. Delete



Algorithm

Step1: start.

Step2:login.

Step3: output “search for course or student you want to delete”.

Step4: input “view course or student information”.

Step5: if (the user is the lecturer).

Step6: output “delete the course or the student”.

Step7: delete.

Step8: else.

Step9: output “invalid process”.

Step10: end.

Array is the data structure that we will use.

## Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| **Team** | Mayar and yara |
| **Document Purpose and Audience** | mayar |
| **Class diagram** | Yara and Mayar |
| **Sequence diagrams** | Mayar and yara |
| **Physical Entity-Relationship diagram** | yara |
| **User interface diagram** | mayar |
| **Dataflow diagram** | Yara and mayar |
| **Algorithms and Data structure** | mayar and yara |
| **Ownership Report** | Yara |

<https://github.com/mayar6832/software1>