

**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.**

## Instructions [To be removed]

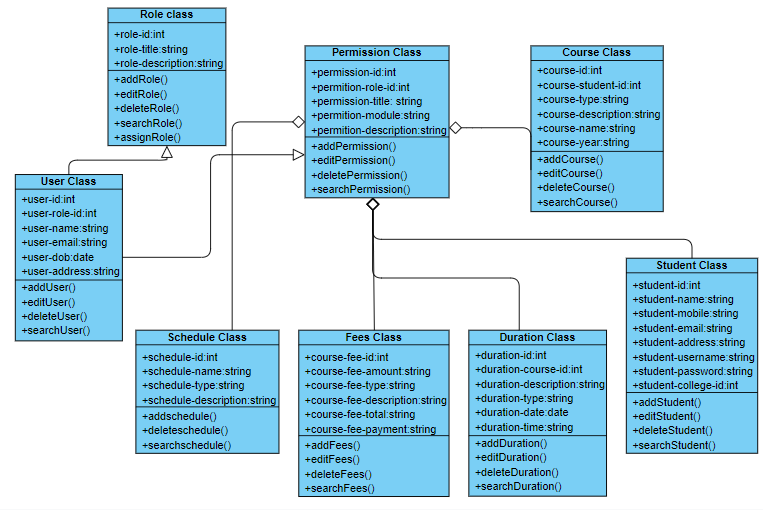
## Team

|  |  |  |  |
| --- | --- | --- | --- |
| **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

## System Models

### Class diagrams



### Sequence diagrams

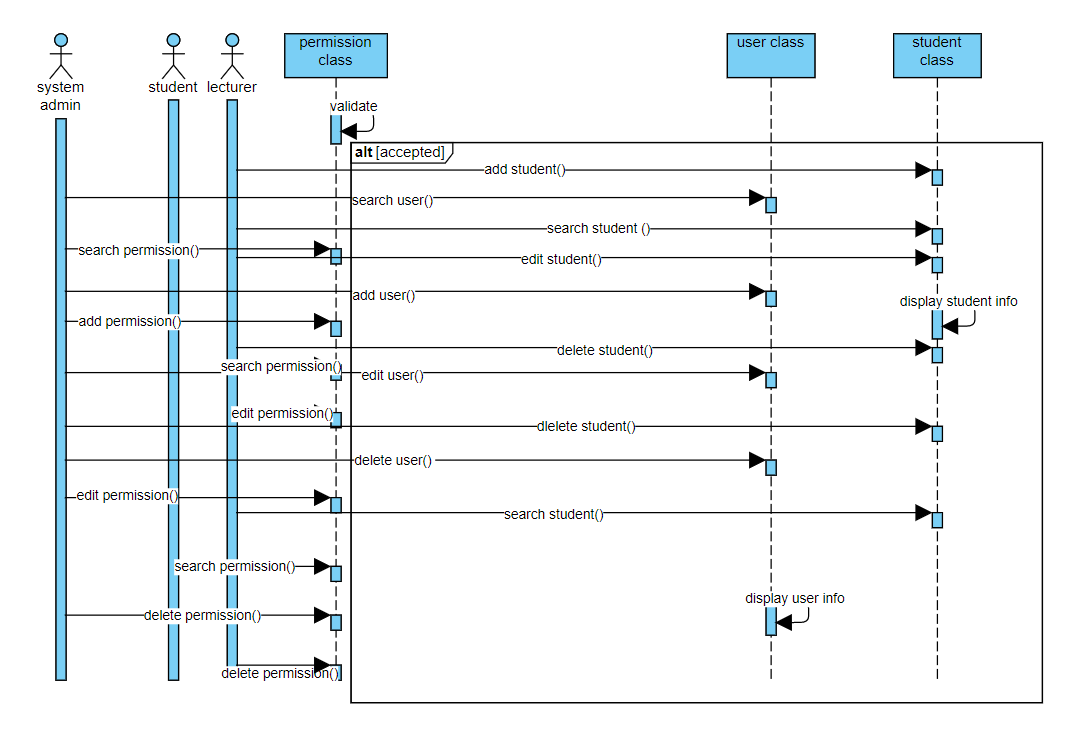
(1)



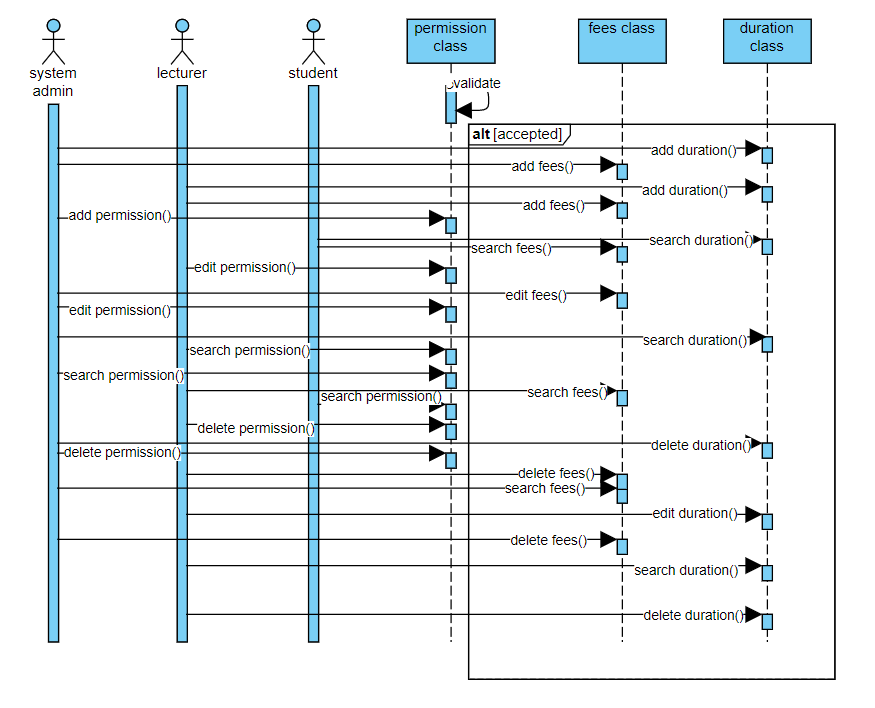
**(2)**



(3)



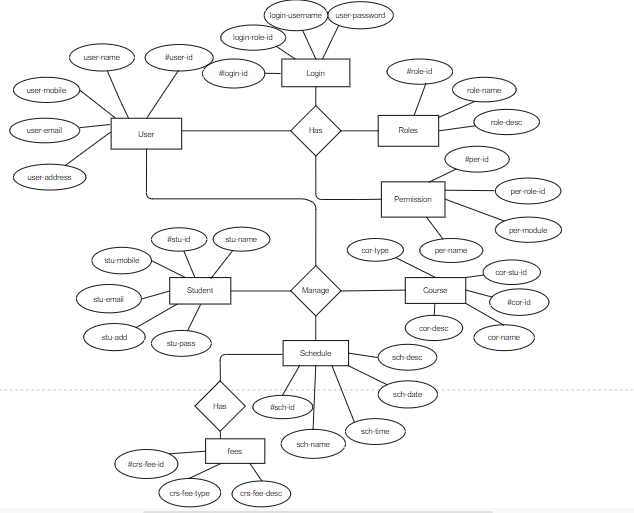
(4)



Class - Sequence Usage Table

|  |  |  |
| --- | --- | --- |
| **Class name** | **Sequence diagram** | **Overall used methods** |
| User class | 3 | Add user ()  Edit user ()  Search user ()  Delete user () |
| Student class | 3 | Add student ()  Edit student ()  Search student ()  Delete student () |
| Role class | 1 | Add role ()  Edit 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 ()  Edit 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 () |

### Physical Entity-Relationship Diagram

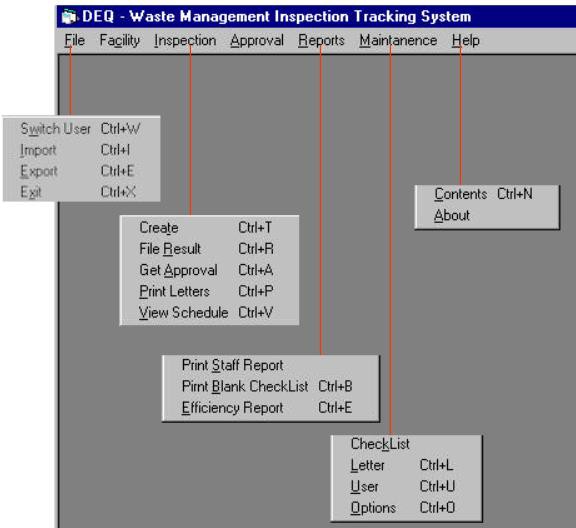


### User Interface Design

* + **Use a prototyping tool like** [**https://app.moqups.com**,](https://app.moqups.com/) [**http://Infragistics.com**](http://infragistics.com/) **or NinjaMock or using a GUI builder (like the one in NetBeans) to build your interface.**
  + **Develop a prototype for each screen / page that your application will have and relate them to each other showing which one leads to which one.**
  + **For each screen specify the buttons, menus, etc. that will be on it and their functions.**
  + **An example is shown below.**
  + **Screen 1 – Login Screen (example)**



* + **Screen 2 – Main Interface (example)**



* + **Navigation tree:**

Login Screen

|

Main Screen

### Dataflow diagram (DFD)

* + **Provide the DFD**

### Algorithms and Data Structures

* + **Specify what algorithms you need in order to build the application. If it is an existing one, just refer to it. If it is one you will develop, then write in detail in mathematical notation, pseudo code, or as a flowchart. Example of such algorithms:**
    - **The steps for calculating if there is winner in a two-player game.**
    - **The steps for calculating the salary in a payroll program.**
    - **The algorithm for deciding which posts to show first in a social network application.**
  + **Specify which data structures (DS) you will use to store which data in memory, other than regular arrays and array lists. Justify your choice and explain the reasons behind it.**
  + **In the rare occasion that no existing data structures supports your requirements and you need to create a new one or implement a non-implemented one, include the design of this new DS.**

## Ownership Report

* + **Remove the following notes and any red notes**
  + **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100.%**
  + **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |

Policy Regarding Plagiarism**:**

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة

2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.

3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.

4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.

5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.

6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المرر.