

# Graduation Projects System

TA: Dalia Elalfy

Team ID: 3

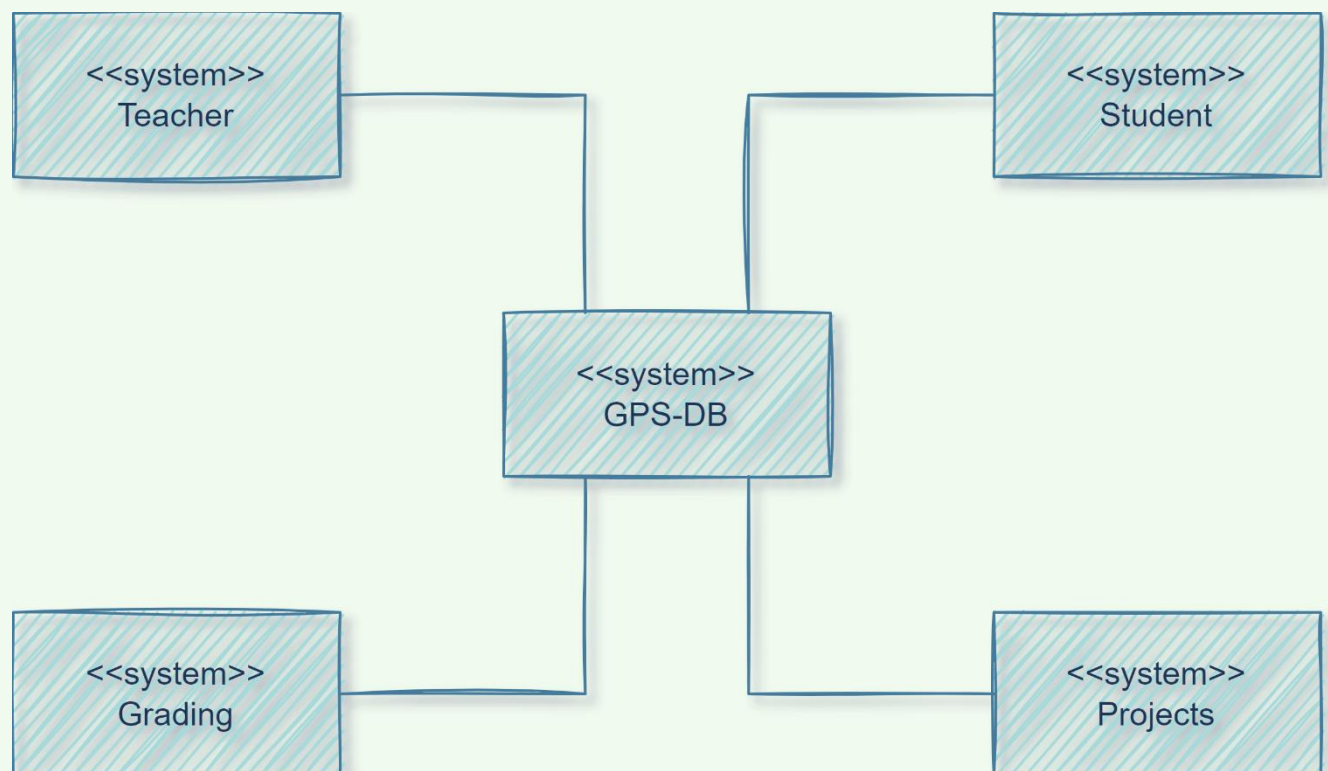
Team Members:

Department	ID	Name
AI	20201701725	علي احمد حامد شاكر محمد حجاب
AI	20201701703	احمد مدحت محفوظ محفوظ
AI	20201701733	محمد رضا أبو بكر عثمان
AI	20201701732	محمد إبراهيم محمد مزروع
AI	20201701723	عبد الفتاح تامر محمد اسماعيل
SWE	20201701671	عمر احمد حماده احمد شريف
AI	20201701741	مينا جورج رؤوف اسكندر

## Introduction

The graduation projects system benefits students and teachers in grading the graduation projects done by the students, it helps the teachers to set each project's requirements and grading it after the student's submission, also it helps the students to easily upload their work to be graded and know their scores.

## Context Diagram



## User Requirements

- 1- Each student and teacher need to create an account on the system at first then accessing it easily whenever they need.
- 2- Teachers need to have the ability to set the minimum requirements for the graduation projects to be done by the students.
- 3- Students must be able to upload their work after finishing it to be graded.
- 4- Teachers must have the ability to grade each project and submitting this grade so the students can see their grades later on.

## Functional Requirements

### **Log in:**

- Allows the user to log in and access his information and related tasks
- Inputs: Username, Password
- Source: User (Student, Teacher)
- Pre-condition: Username, Password should match the data stored
- Post-condition: It will compare the data entered to the data stored
- Output: Project working space and account page will be displayed

### **Submitting projects:**

- Allows the user to upload the project
- Input: Files (pdf, pictures, code)
- Source: User (Student), Storage
- Pre-condition: It must have a file to upload and does not exceed 20 mb
- Post-condition: Project will be uploaded
- Output: User will get a message that the file has been submitted

### **Project Validation:**

- Data uploaded should be checked in a short term
- Input: Files uploaded by the user (Student)
- Source: Database
- Pre-condition: The user must be a teacher
- Post-condition: User (Teacher) will be allowed to validate the project
- Output: Projects submitted will be shown

**Grades:**

- Each student will get his grade after being uploaded by the teacher
- Input: Student ID, Grade
- Source: User (Teacher)
- Pre-condition: Student data is found in the database
- Post-condition: It will merge the grade with the student data
- Output: Student data and grades will be shown

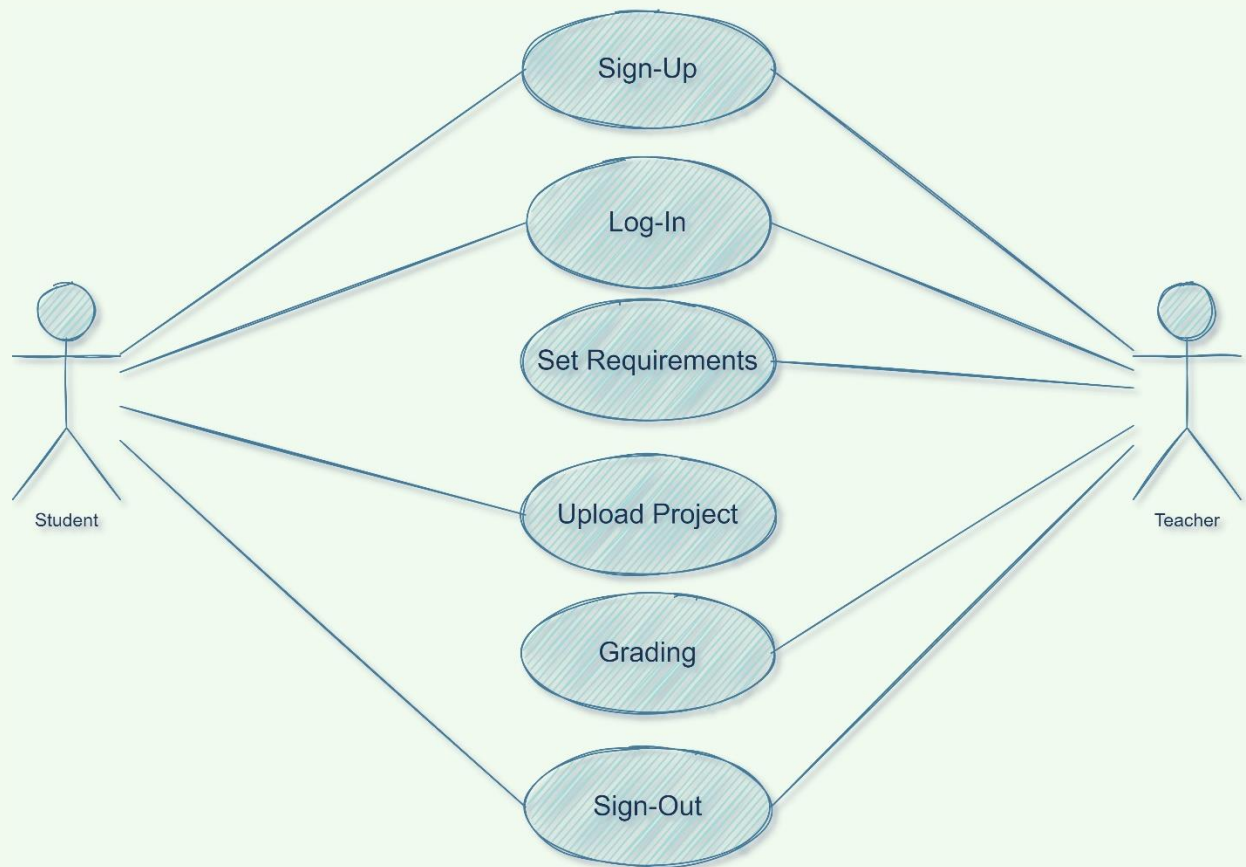
**Project Requirements:**

- Teacher will put requirements for each project
- Input: Requirements
- Source: Teacher
- Pre-condition: The user must be a teacher
- Post-condition: Access to set requirements
- Output: Projects Requirements will be posted

## Non-Functional Requirements

1. Reliability:
  - i. Downtime failure of the system shouldn't exceed 5 ms per day
  - ii. The account update process shall roll back all related updates when any update fails to commit
2. Efficiency:
  - i. Product size shouldn't exceed 25 MB
  - ii. The screen refresh time shouldn't exceed 0.025 seconds
  - iii. Reports generation process shouldn't exceed 2 seconds
3. Performance: The system load time should not be more than one second
4. Security: The system should ensure that only authorized users can access to the project information (project developers)
5. Usability: The website's interface has to be user-friendly and easy to use
6. Portability: System should be available for any platform and any browser

# Use Case Diagram



# Sequence Diagram

