Course assignment management website

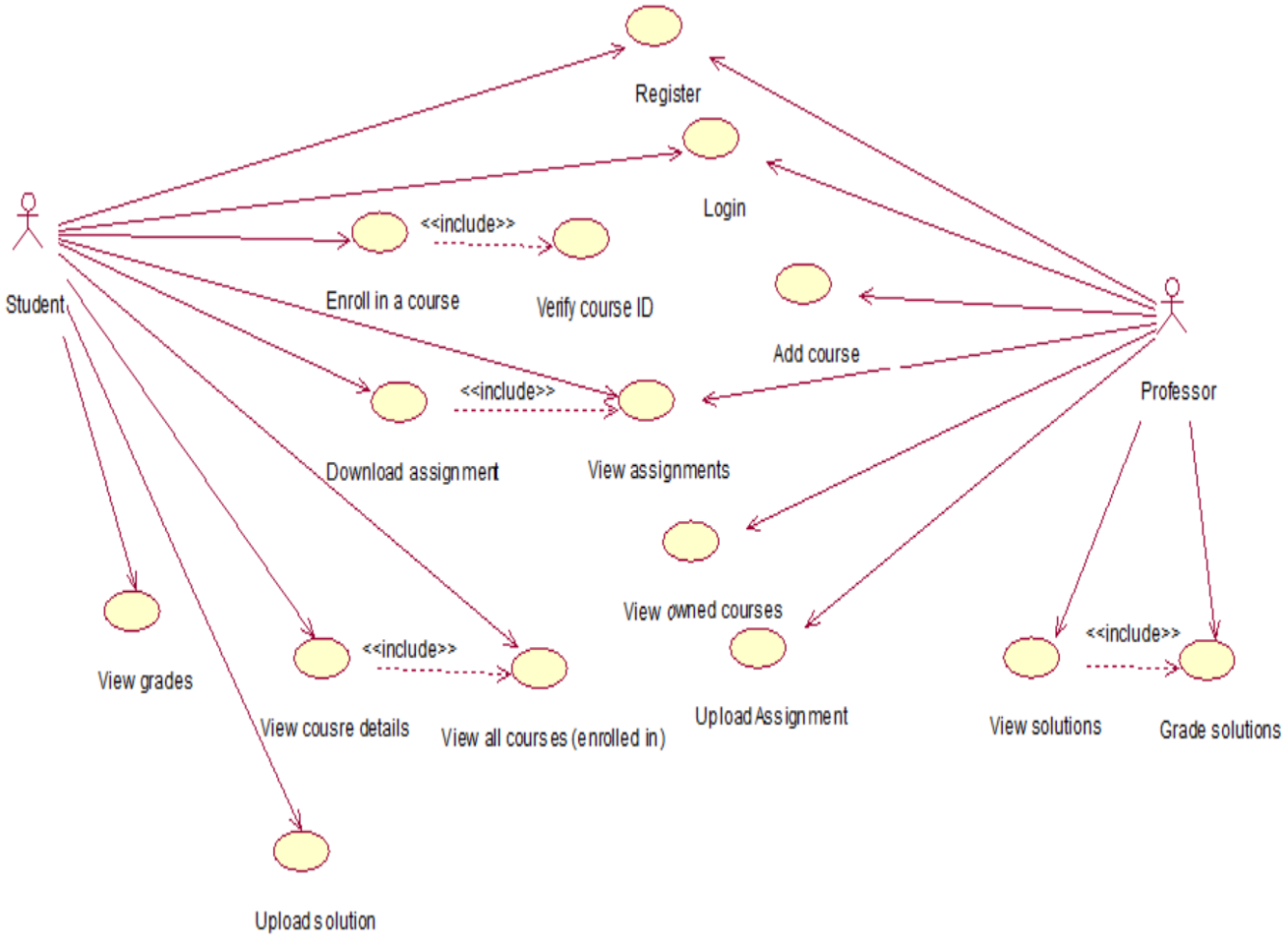
Abdelrhman Mahfouz, Fatma Jamal, Mohamed Hashish, Nosiba Fattouh, Zeina Fouad, Hadil El Rody
Faculty of computer and information science, Ain shams.
El-Kaeed, El-Qobba Bridge, Al Waili, Cairo Governorate
13/3/2019
Microsoft Office 365 Word

1. Introduction

- **1.1 Purpose of this document**
  This document describes in detail the characteristics of software and how it is used and implemented. This document is targeted for both the developers and to serve as a contract between the development party and the client party.
- **1.2 Scope of this document**
  The requirements gathered was only concerned with the intended user experience on the website; The internal structure of the system, the security of the data and the type and placement of the database and the servers was up to our time and budget to decide.
  The entire team contributes to the requirement elicitation and design process. Our names are mentioned right below the title. We are a 3rd year cs students and this documentation is part of our assignment.
  We only had one chance to interview the TA who acted like a regular client. It was our very first ever interview. We are still learning; this document is our second take on the requirements documentation process. We started with a very simple document that didn't follow the standards administered by the IEEE due to time limitations, however we reworked our document to this standard after we managed to set up enough time for the migration process.
  Obviously, we are not getting paid any money for this product. We will be paid with GPA in case we did what we had to do correctly.
- **1.3 Overview**
  The product is a website that automates the assignment delivery process. It acts as an interface between the students and the professor to easily facilitate the assignment delivery/grading process on any PC web browser.
- **1.4 Business Context**
  This is a student project for the System Analysis and Design course. We work solely on this project and the TAs only provide us guidelines and minor support throughout the development process. The goal is to learn the principles of software engineering by practicing it in a real project that should be implemented once this document is finalized.

2. General Description

- **2.1 Product Functions**



The student can

1. Login to the website with the faculty id and password of choice.
2. Enroll in a course when instructed to by the professor.
3. View all courses in which he/she enrolled.
4. For any course, view detailed information supplied by the professor responsible for that course.
5. View all required assignments.
6. Download any assignment (The format and the quality of the file is the responsibility of the professor; the website is not responsible for the quality of the format or any potential harmful software uploaded by the professor).
7. View all the grades of all submitted assignments in all courses
8. Ask a question for every assignment via the forums.

The professor can

1. Login to the website with the faculty id and password of choice.
2. Add courses to the website.
3. View owned courses.
4. View the assignments for any owned course.
5. Upload an assignment to a given course specifying the name, due date and all relevant information.
6. Answer questions submitted on any forum belonging to an assignment in an owned course.
7. View solutions submitted by students.
8. Grade solutions using a numerical grading system.

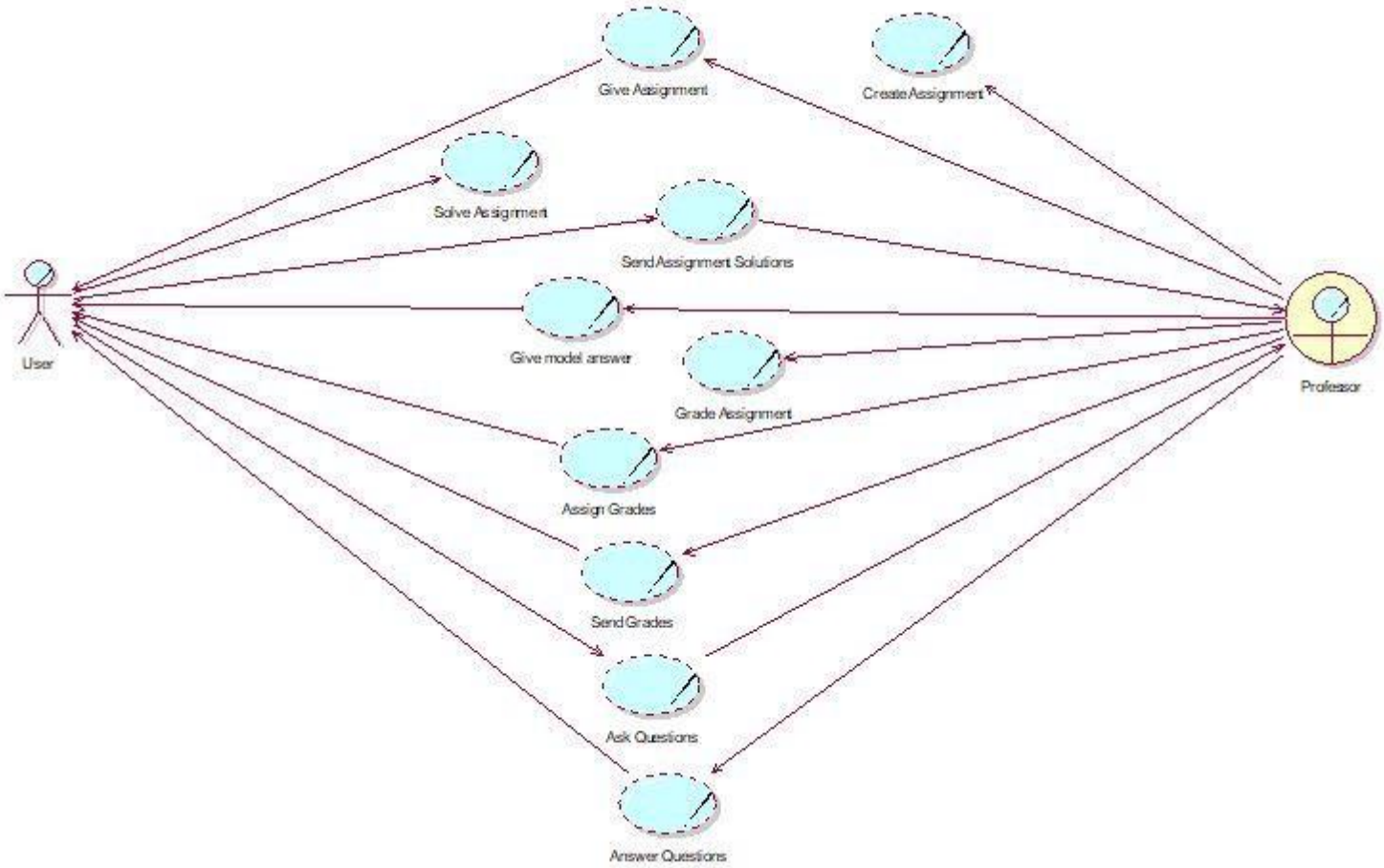
- **2.2 Similar System Information**

This is a standalone product; It doesn't interact with any other systems due to their absence.

- **2.3 User Characteristics**

Users should be familiar with general use of computers, using browser to browse websites, download and upload files. Users have limited to zero experience with any technical expertise regarding how browsers and/or websites work.

- **2.4 User Problem Statement**



The traditional assignment delivery process suffers from various problems such as spelling coffee on the assignment minutes before the lecture starts or forgetting it all together in home or sometimes silly issues stand in the way such as not finding proper papers. The grading system also has issues. As shown above, the grades must officially go through administration before it can be handed out to students. While some professors declare the students' grades in any of the lectures once they are graded, some students may not want everyone to know about their grades. The only other alternative is to go through the struggle of getting the grades from the administrations or at the office hours of the professor.

Sometimes students never know their assignment grades till the entire year work is released. Sometimes even the year work is not released till the finals are graded, by then the student would have lost all enthusiasm in finding about how did he do in the assignment.

Finally, the professor finds himself dealing with lots of not-so-well-made papers in 2019 were all modern institutes moved to technology. The professor may accidentally lose one of the assignments, leaving him/her with confusion about what student did that paper belong to and what grade should be given to that student.

- **2.5 User Objectives**

General objectives

- Automate the delivery process via an easy-to-use interface within the comfort of home.
- Professors don't want to worry about physical maintenance of the assignment.
- Students want to know their grades exclusively without going through long queues and ignorable emails.
- Students want to easily ask questions regarding a specific assignment in a specific course as they are solving it in their homes.
- Students would want to get back to the assignments and their previous solutions as well as the model answers before exams as a form of revision and to avoid past mistakes.

User/System requirements

1. User must be able to enroll in a course.
    a. The system must save in the database a relation between said user and the course
    b. The relation must be marked as pending till the professor approves it
2. User must be able to view all courses in which he enrolled.
    a. The system must store each course-student relation
3. User must be able to view assignments tied to a course.
    a. The system must store each assignment-course relation

4. Professor must be able to add a course to the website.
    a. The system must have a table for courses
    b. The system must only allow professors to add courses
5. Professor must be able to submit assignments for any of his/her owned courses.
    a. The system must store every assignment-course relation
    b. The system should allow the professor to specify the due date
    c. The system should accept anything uploaded by the professor
6. Professor must be able to download submitted solutions for assignments.
    a. The system must check for authenticity of the professor
7. Professor must be able to grade any and all students' submissions.

3. Functional Requirements

General Requirements

1. ***Users must be able to login***
    1. **Description**
       Users must be able to login to the website with their email and password of choice.
       User should be only aware that username and password are wrong. Details whether the username is correct should be obscure.
    2. **Criticality**
       Highly critical; System is unusable without proper authentication.
    3. **Dependencies with other requirements**
       No dependencies
2. ***Users must be able to view all courses that they are enrolled in***
    1. **Description**
       Users should be able to get a list of all courses. Each course in the list should contain the name and course ID contained within neat UI.
    2. **Criticality**
       Highly Critical; Critical navigation point.
    3. **Technical issues**
       Too many courses or no courses at all should have their special design Implementations.
    4. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
3. ***Users must able to view details of a specific course. Details include all associated assignments.***
    1. **Description**
       The user must be able to view the details of a course.
       The details are the course name, id, description, instructor name, instructor autobiography and all assignments associated with that course.
    2. **Criticality**
       Highly Critical; Critical navigation point.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
4. ***Users must be able to view assignment details.***
    1. **Description**
       Users must be able to see the details of an assignment. An assignment can be fetched from the back-end server via it's id.
       Assignment details should be: due date and a download link for the assignment itself.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (RESTful API for course details) & Requirement #2 (access_token should be present in cookies)

Student Specific Requirements

1. ***Student must be able to enroll to a course.***
    1. **Description**
       The professor will give his/her students the course ID. Students must be able to choose a course from a list of available courses. Verification for whether the student is eligible for that course is done manually by the professor.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in).
2. ***Student must be able to download any assignment for any course he/she enrolled in at any time.***
    1. **Description**
       The student must always be able to download assignments statement.
    2. **Criticality**
       Highly Critical; Core functionality.
    3. **Risks**
       the website is not responsible for any viruses planted within the assignment.
    4. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
3. ***Student must be able to submit solutions for any assignment he/she has access to before it's due date.***
    1. **Description**
       Student must be able to select any file format and upload it to the server for a specific course.
    2. **Criticality**
       Highly Critical; Core functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
4. ***Student must be able to exclusively view his/her grades for any of his/her submitted assignments.***
    1. **Description**
       Students must be able to see their grade for each assignment the submit in once the professor releases the grades.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)

Professor Specific Requirements

1. ***Professor/Instructor must be able to create a new course on the website.***
    1. **Description**
       According to the client, professors are responsible for adding courses to the system. Professors should be able to specify the name and code of their course.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
2. ***Professor/Instructor must be able to add assignments to a course that he/she owns.***
    1. **Description**
       Professors must be able to add assignments to any course specifying the name, total grades and due date of said assignment as well as uploading the assignment itself.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)
3. ***Professor/Instructor must be able to grade any submission to any of his/her assignments***
    1. **Description**
       Professors must be able to download and grade submissions of his students in his assignments
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)

4. ***Professor/Instructor must be able to accept enrollments in his/her course.***
    1. **Description**
       Professor must be able to view all students who want to enroll to his course and accept them individually.
    2. **Criticality**
       Highly Critical; Core Functionality.
    3. **Dependencies with other requirements**
       Requirement #1 (user must be logged-in)

4. Non-functional requirements

- ***General requirements***

    1. At worest cases and with stable connection from the end-user. System should not take more than 10 seconds to load any of its resources. The system should support up to 4000 concurrent users (such load may be possible when final grades are released for all students) without violating the constraint.
    2. Any and all crashes happening in the server must not be known by the end-user and handled silently by the server
    3. The UI must be clear and convenient. User should be able to navigate through all the website without human help in 10 minutes.
    4. Any functionality that a certain user doesn't have access to should be hidden. User must not be able to attempt to do anything he is not allowed to.
    5. Incase the user did manage to do something he has no access to, the back-end should check the access rights of the request before processing it. Client-side code should effectively handle such cases and navigate the user to the proper page (See activity diagrams for examples).

- ***Development requirements***

    1. The back-end should be ASP.NET MVC 5.
    2. The back-end database should be any code-first entity framework.
    3. The front-end should use bootstrap and jQueyr.
    4. The website should run as good on the top 3 browsers (Firefox, Chrome and Edge).
    5. The entire project will be hosted on Github.

5. Interview Guideline

Interview Outline	
Interviewee: Dr Esraa Gamal	Interviewer: Abdelrahman Mahfouz
Location/Medium: FCIS-ASU	Appointment Date: 28-Feb-2019 Start Time: 8:00 AM End Time: 8:15 AM
Objectives: Know how the data will be managed (external, our system) Learn about the intended overall experience	Reminders: Have strong experience in programming computer science and web design
Agenda: Introduction Background on Project Overview of Interview Topics to be covered Permission to tape record Scenario outline Summary of Major Points Questions from Interviewee Closing	Approximate time: 1 minute 2 minutes 1 minute 5 minutes 2 minutes 2 minutes 1 minutes
General Observations: Fairly generic and without any complications	
Unresolved Issues, Topics not covered :	
Interviewee:	Date: 28-Feb-2019
Questions:	Notes:
Question: 1 Is the data stored in another system?	Answer We will enter the data
Question: 2 How do you like the students to know about the assignments?	Answer Get notification from the site
Question: 3 How do you like the students to register?	Answer The student register using their ID number

Question: 4 How the students will ask questions related to the course	**Answer** Each course will have a forum in which they can write their questions
Question: 5 How do you want the grading system	**Answer** The grades will be numbers and the instructor will write the grades of each assignment
Question: 6 Give us a scenario for what you want *Student scenario* *Professor scenario*	**Answer** The student will sign up by the ID and enroll in courses available by the course ID given by the Professor, then download the assignments, upload the solutions before the deadline. If the student has a question, he writes it in the course's forum. The students can view his grades privately. The Professor is considered an administrator for the site; he can add new courses, add assignment for a course and assign a deadline for each assignment. He can also read and grade uploaded assignments.

6. **SOW**

FCIS-Ain Shams university **Prepared:03/03/2019**

Statement of Work

Project Name:	Course assignment management website
Customer:	Teacher Assistant
Project Sponsor:	Doctor
Project Start/End:	24/2/2019-9/5/2019

Project Description

Goal

The aim of this project is to the track and manage college assignments for each course, where professors can upload assignments and add the grades online, and students can download assignments, upload solutions and view their grades.

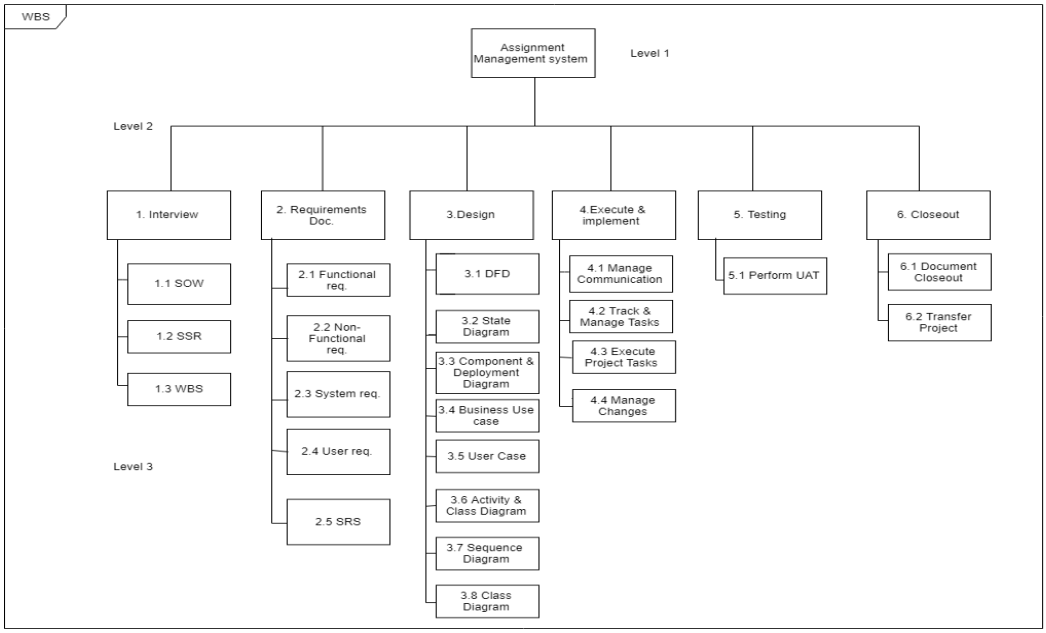
Objective

To facilitate the assignment operation by providing an easy to use interface between the students and the professors that is accessible from any web browser

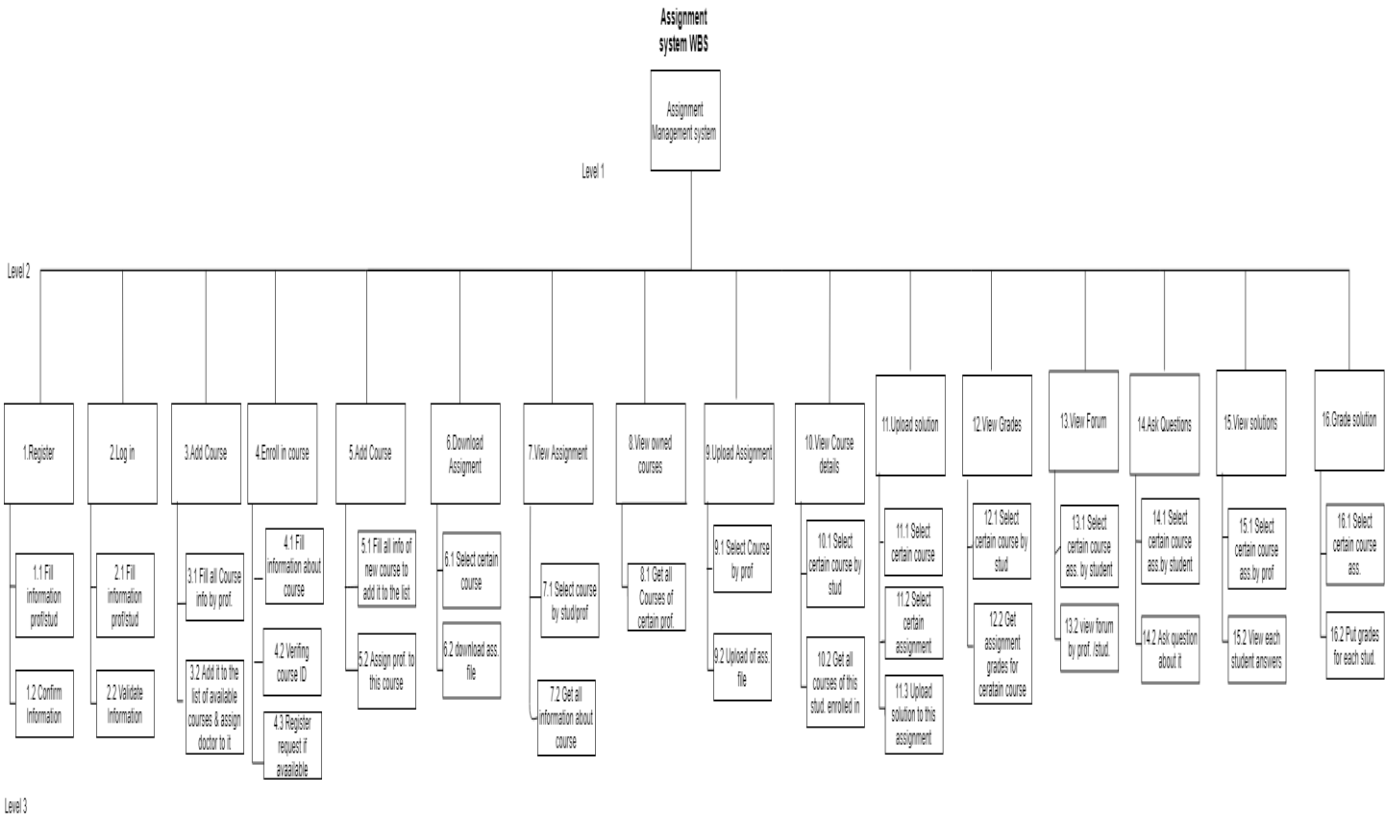
Phases of Work

1)Interview
2)SRS Document
3)Design
4)Execute & implement
5)testing
6)Close Out

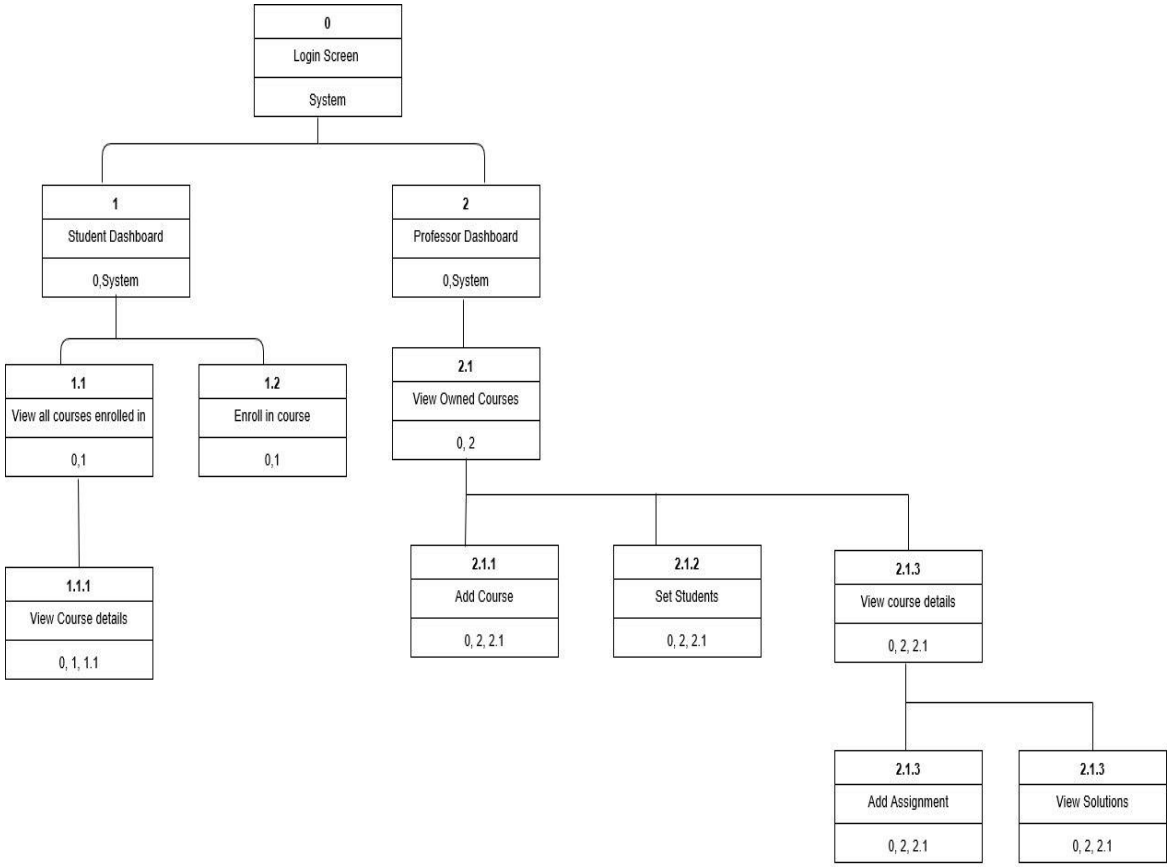
7. **WBS**



Assignment system WBS


Assignment system WBS
Assignment Management system
Level 1
Level 2
1.Register
2.Log in
3.Add Course
4.Enroll in course
5.Add Course
6.Download Assigment
7.View Assignment
8.View owned courses
9.Upload Assignment
10.View Course details
11.Upload solution
12.View Grades
13.View Forum
14.Ask Questions
15.View solutions
16.Grade solution
1.1 Fill information prof/stud
1.2 Confirm information
2.1 Fill information prof/stud
2.2 Validate information
3.1 Fill all Course info by prof.
3.2 Add it to the list of available courses & assign doctor to it
4.1 Fill information about course
4.2 Verifing course ID
4.3 Register request if available
5.1 Fill all info of new course to add it to the list
5.2 Assign prof. to this course
6.1 Select certain course
6.2 download ass. file
7.1 Select course by stud/prof
7.2 Get all information about course
8.1 Get all Courses of certain prof.
9.1 Select Course by prof
9.2 Upload of ass. file
10.1 Select certain course by stud
10.2 Get all courses of this stud. enrolled in
11.1 Select certain course
11.2 Select certain assignment
11.3 Upload solution to this assignment
12.1 Select certain course by stud
12.2 Get assignment grades for certain course
13.1 Select certain course ass. by student
13.2 view forum by prof. /stud.
14.1 Select certain course ass by student
14.2 Ask question about it
15.1 Select certain course ass by prof
15.2 View each student answers
16.1 Select certain course ass.
16.2 Put grades for each stud.
Level 3

8. **Interface**


0
Login Screen
System
1
Student Dashboard
0,System
2
Professor Dashboard
0,System
1.1
View all courses enrolled in
0,1
1.2
Enroll in course
0,1
1.1.1
View Course details
0, 1, 1.1
2.1
View Owned Courses
0, 2
2.1.1
Add Course
0, 2, 2.1
2.1.2
Set Students
0, 2, 2.1
2.1.3
View course details
0, 2, 2.1
2.1.3
Add Assignment
0, 2, 2.1
2.1.3
View Solutions
0, 2, 2.1

9. **SSR**

College Assignment dashboard

System Service Request

REQUESTED BY Dr Esraa Gamal DATE 25/2/2019

DEPARTMENT Software Engineering
LOCATION Faculty of Computer & Information Science, Ain Shams University
CONTACT (+**20**) 1150014224

TYPES OF REQUESTS

[✓] New System

[ ] System Enhancement

[ ] System Error Correction

URGENCY

[ ] Immediate – Operations are impaired or opportunity lost

[✓] problems exist, but can be worked around

[ ] Business losses can be tolerated until new system installed

PROBLEM STATEMENTs

Student management system has become important factor in modern education field. This system should help the university to streamline the instructure task and provide real-time access to the data. Building this system in web based interface will further help the ease of accessibility through any web browser.

SERVICE REQUEST

The main purpose of College assignment dashboard website is to the track and manage college assignments for each course, where professors are able to upload assignments and add the grades online, and students can download assignments, upload solutions and view their grades.

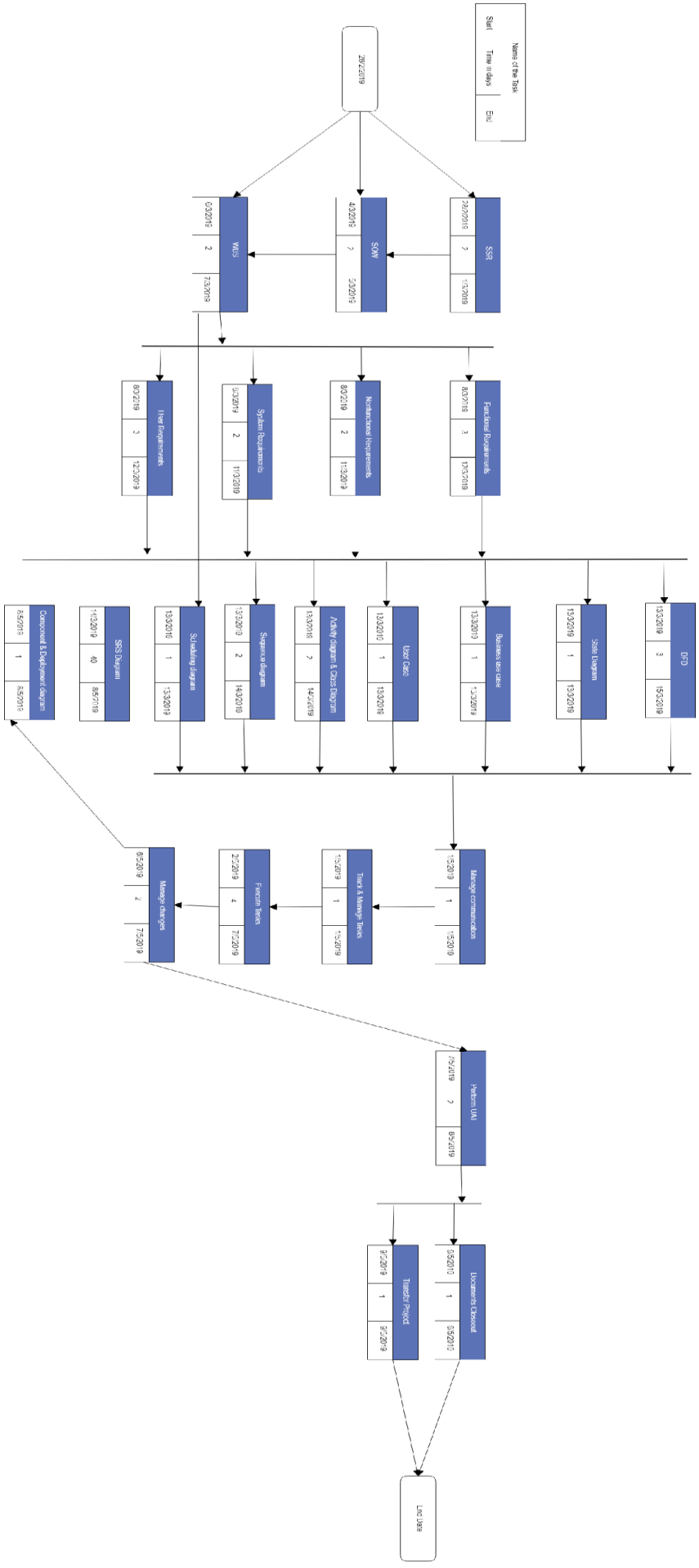
IS LIAISON _____
SPONSOR _____

---------- TO BE COMPLETED BY SYSTEMS PRIORITY BOARD ----------

[✓] Request approved Assigned to _____
Start date 04/03/2019

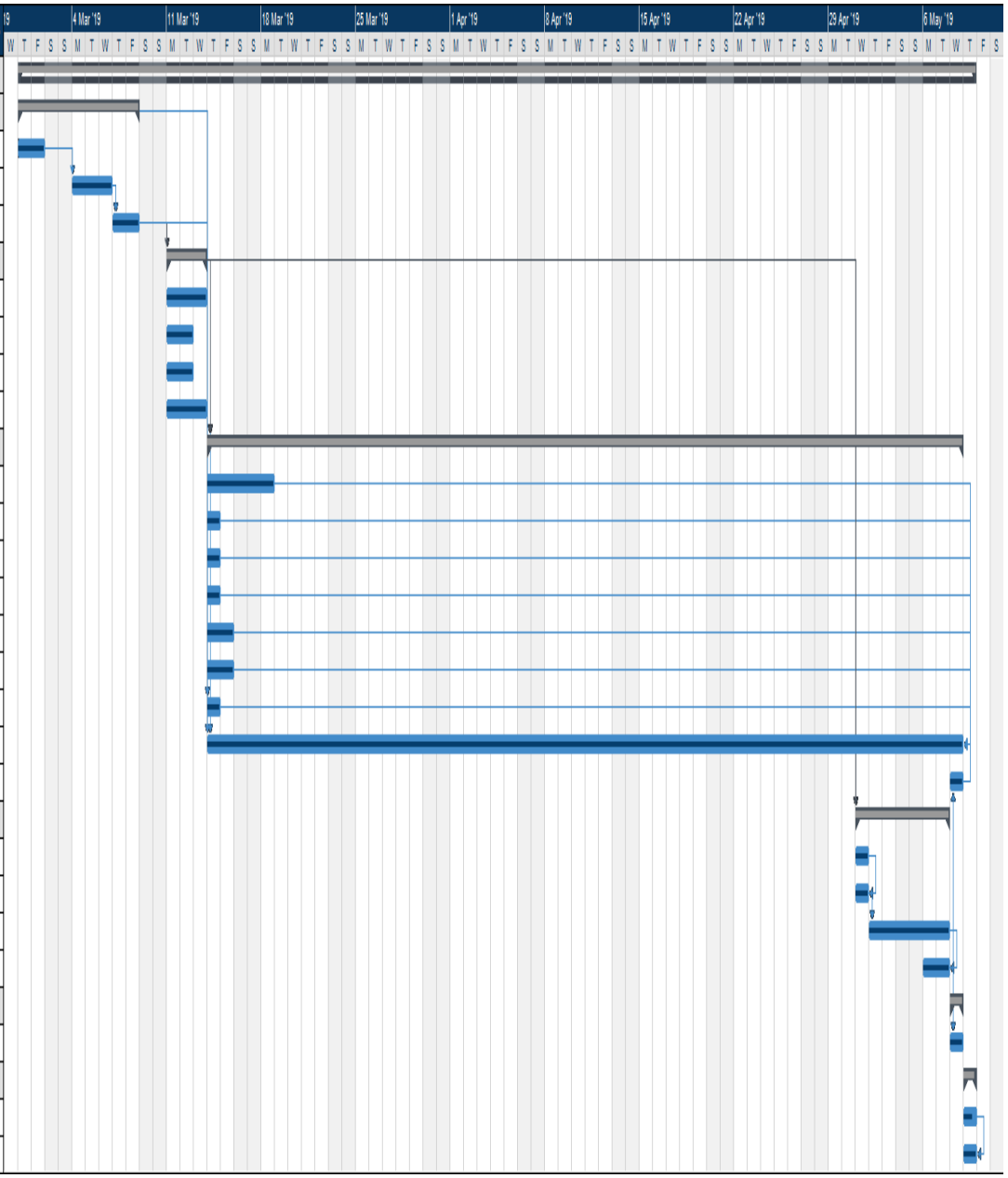
[ ] Recommended revision
[ ] Suggested user development
[ ] Rejected for reason _____

10. **Activity Network**



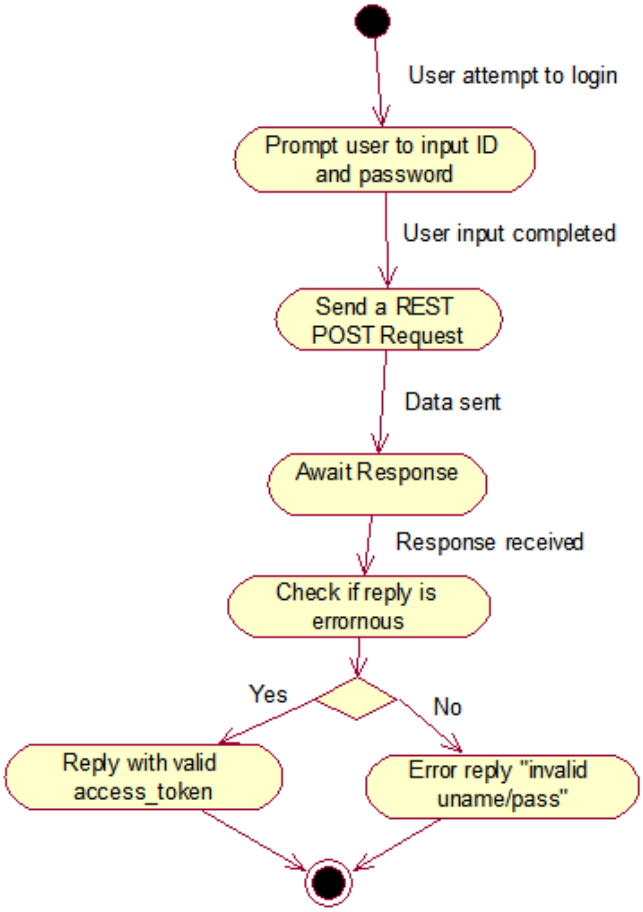
11. **Gantt Chart**

	Task Name	Duration	Start	End	Predecessors
1	Course assignment management website	51 days	28-Feb-19	09-May-19	
2	⊟ 1 Interview	7 days	28-Feb-19	08-Mar-19	
3	1.1 SSR	2 days	28-Feb-19	01-Mar-19	
4	1.2 SOW	3 days	04-Mar-19	06-Mar-19	3
5	1.3 WBS	2 days	07-Mar-19	08-Mar-19	4
6	⊟ 2 Requirements Docs	3 days	11-Mar-19	13-Mar-19	5
7	2.1 Functional Requirements	3 days	11-Mar-19	13-Mar-19	
8	2.2 Nonfunctional Requirements	2 days	11-Mar-19	12-Mar-19	
9	2.3 System Requirements	2 days	11-Mar-19	12-Mar-19	
10	2.4 User Requirements	3 days	11-Mar-19	13-Mar-19	
11	⊟ 3 Design	40 days	14-Mar-19	08-May-19	6
12	3.1 DFD	3 days	14-Mar-19	18-Mar-19	
13	3.2 State Diagram	1 day	14-Mar-19	14-Mar-19	
14	3.3 Business use case	1 day	14-Mar-19	14-Mar-19	
15	3.4 Use case	1 day	14-Mar-19	14-Mar-19	
16	3.5 Activity diagram & Class Diagram	2 days	14-Mar-19	15-Mar-19	
17	3.6 Sequence diagram	2 days	14-Mar-19	15-Mar-19	
18	3.7 Scheduling diagram	1 day	14-Mar-19	14-Mar-19	5
19	3.9 SRS Diagram	40 days	14-Mar-19	08-May-19	2;6;12FF;13FF;...
20	3.10 Component & Deployment diagram	1 day	08-May-19	08-May-19	25
21	⊟ 4 Execute & Implementation	5 days	01-May-19	07-May-19	6
22	4.1 Manage communication	1 day	01-May-19	01-May-19	
23	4.2 Track & Manage Tasks	1 day	01-May-19	01-May-19	22FF
24	4.3 Execute Tasks	4 days	02-May-19	07-May-19	23
25	4.4 Manage changes	2 days	06-May-19	07-May-19	24FF
26	⊟ 5 Testing	1 day	08-May-19	08-May-19	
27	5.1 Perform UAT	1 day	08-May-19	08-May-19	25
28	⊟ 6 Closeout	1 day	09-May-19	09-May-19	
29	6.1 Documents Closeout	1 day	09-May-19	09-May-19	
30	6.2 Transfer Project	1 day	09-May-19	09-May-19	29FF

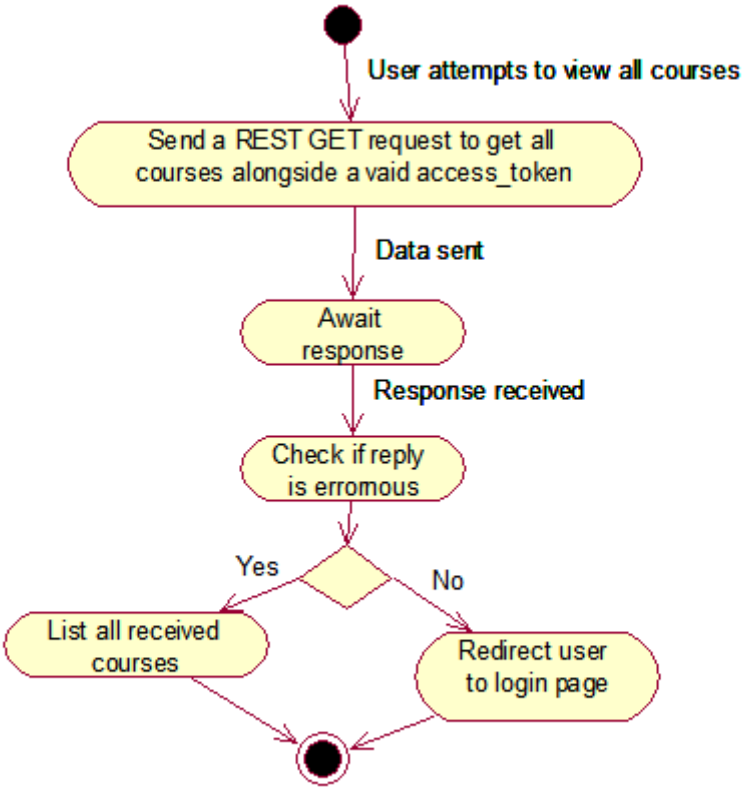


5. Activity Diagrams

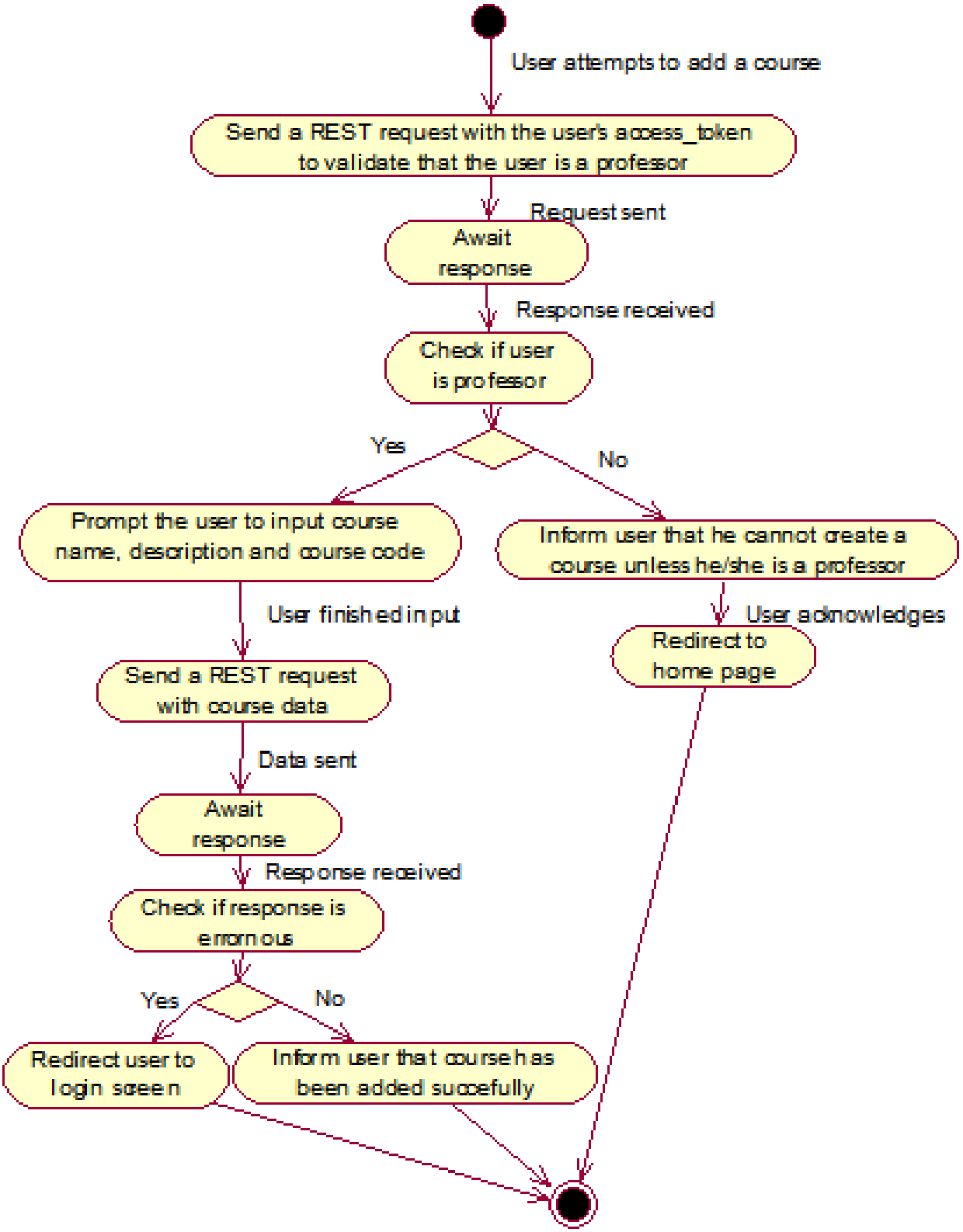
1. ***Login Activity***



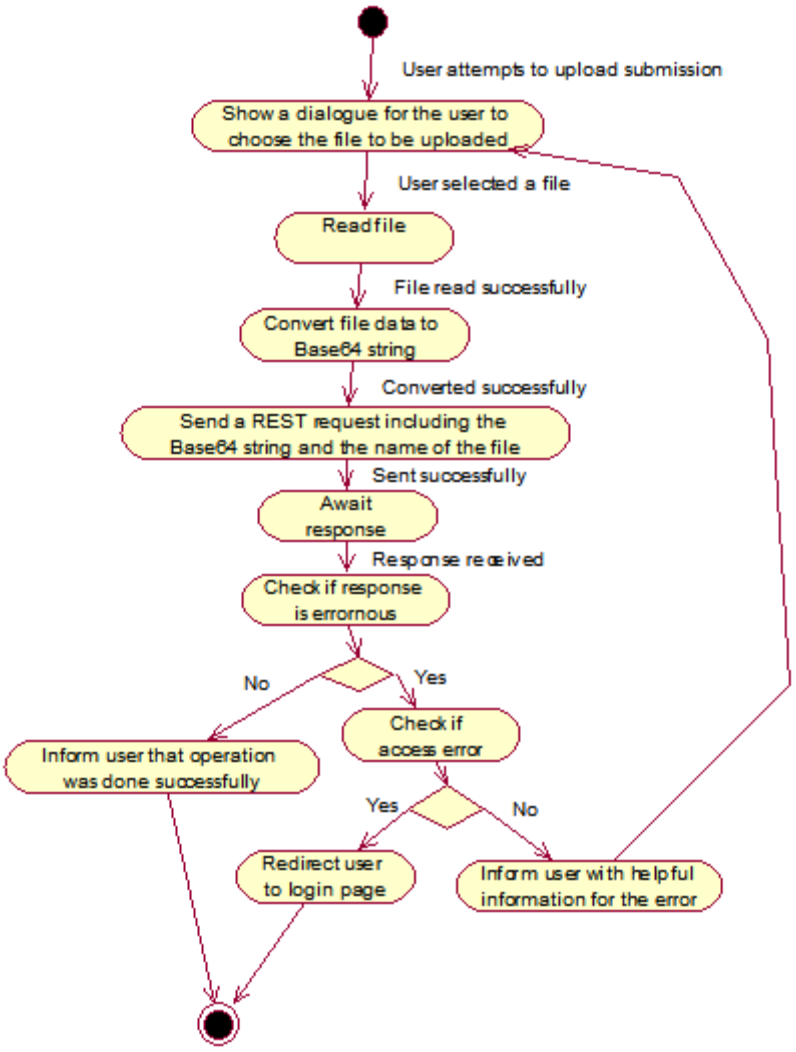
2. ***View all courses***



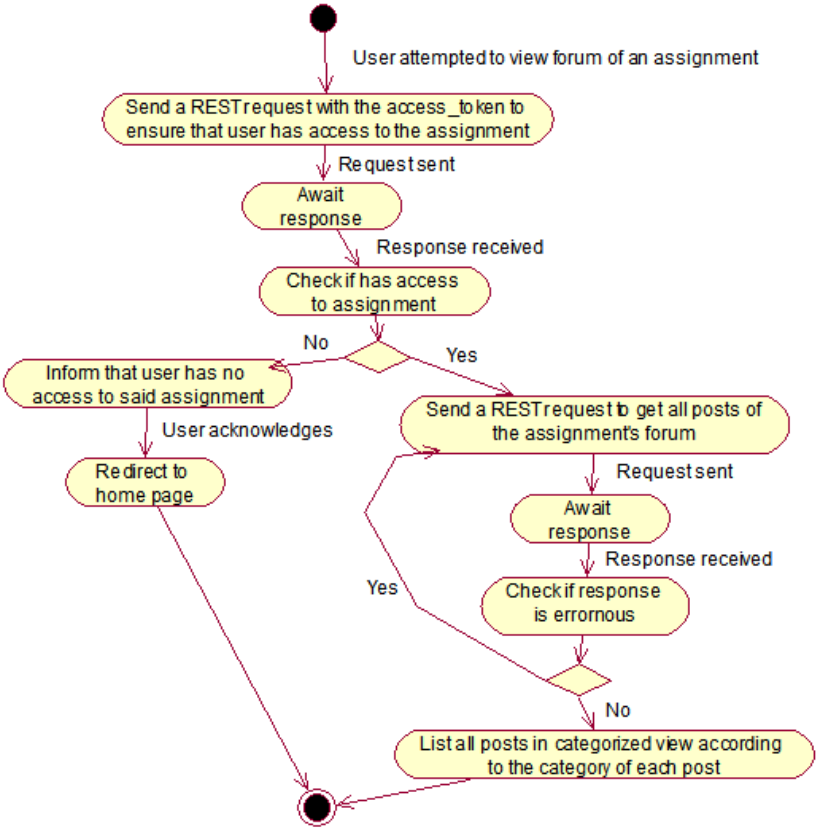
3. Add course


User attempts to add a course
Send a REST request with the user's access_token to validate that the user is a professor
Request sent
Await response
Response received
Check if user is professor
Yes
No
Prompt the user to input course name, description and course code
Inform user that he cannot create a course unless he/she is a professor
User finished input
User acknowledges
Redirect to home page
Send a REST request with course data
Data sent
Await response
Response received
Check if response is erroneous
Yes
No
Redirect user to login screen
Inform user that course has been added succefully

4. Upload assignment solution

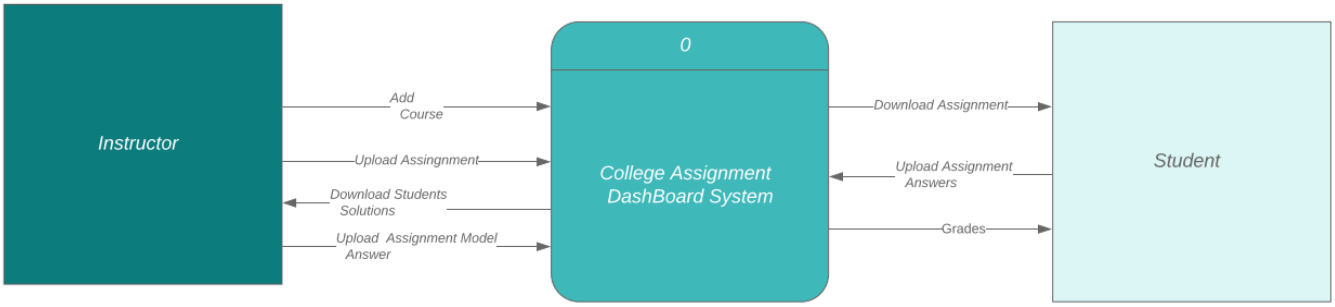


5. View forum of an assignment

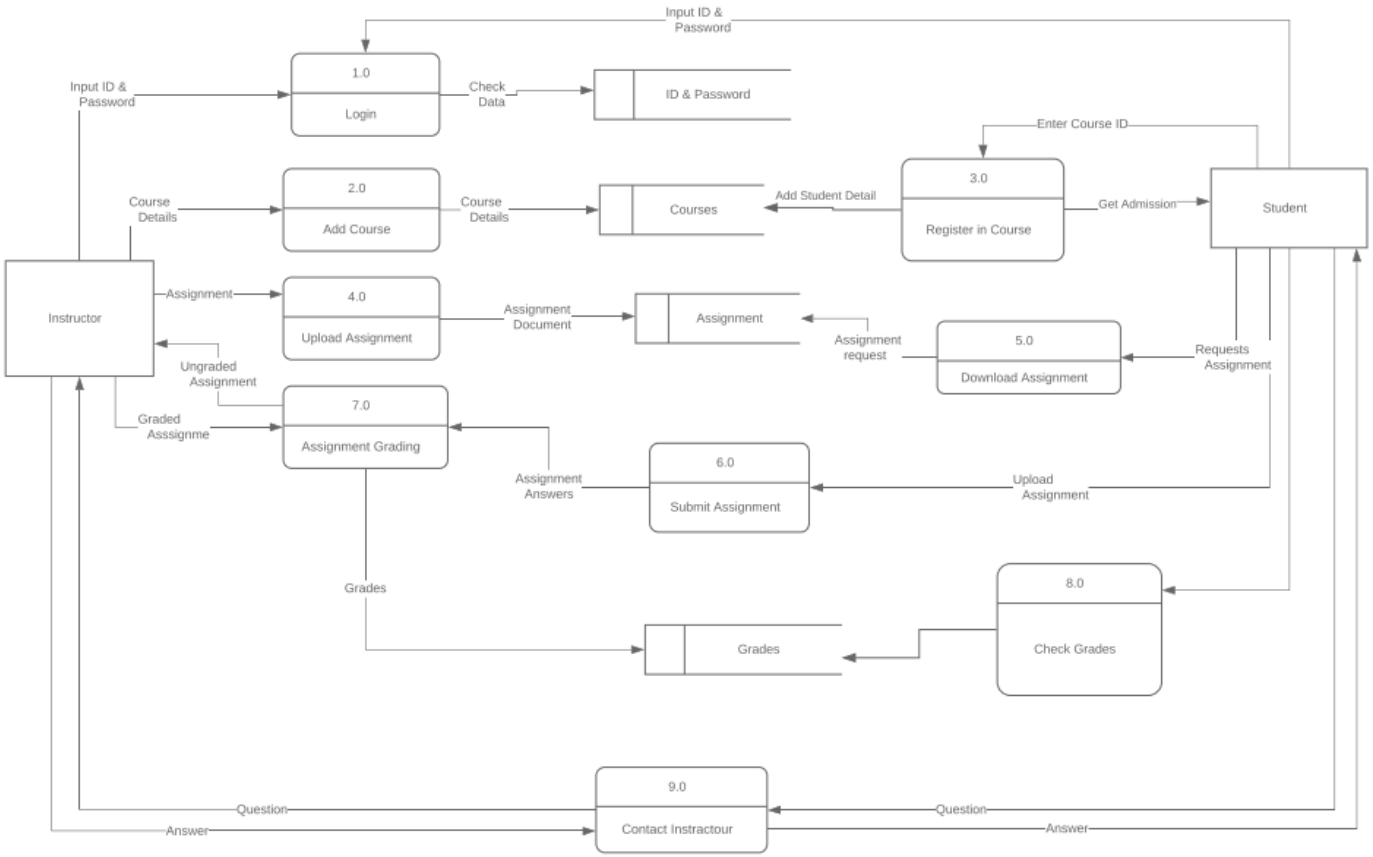


6. Data-flow Diagrams

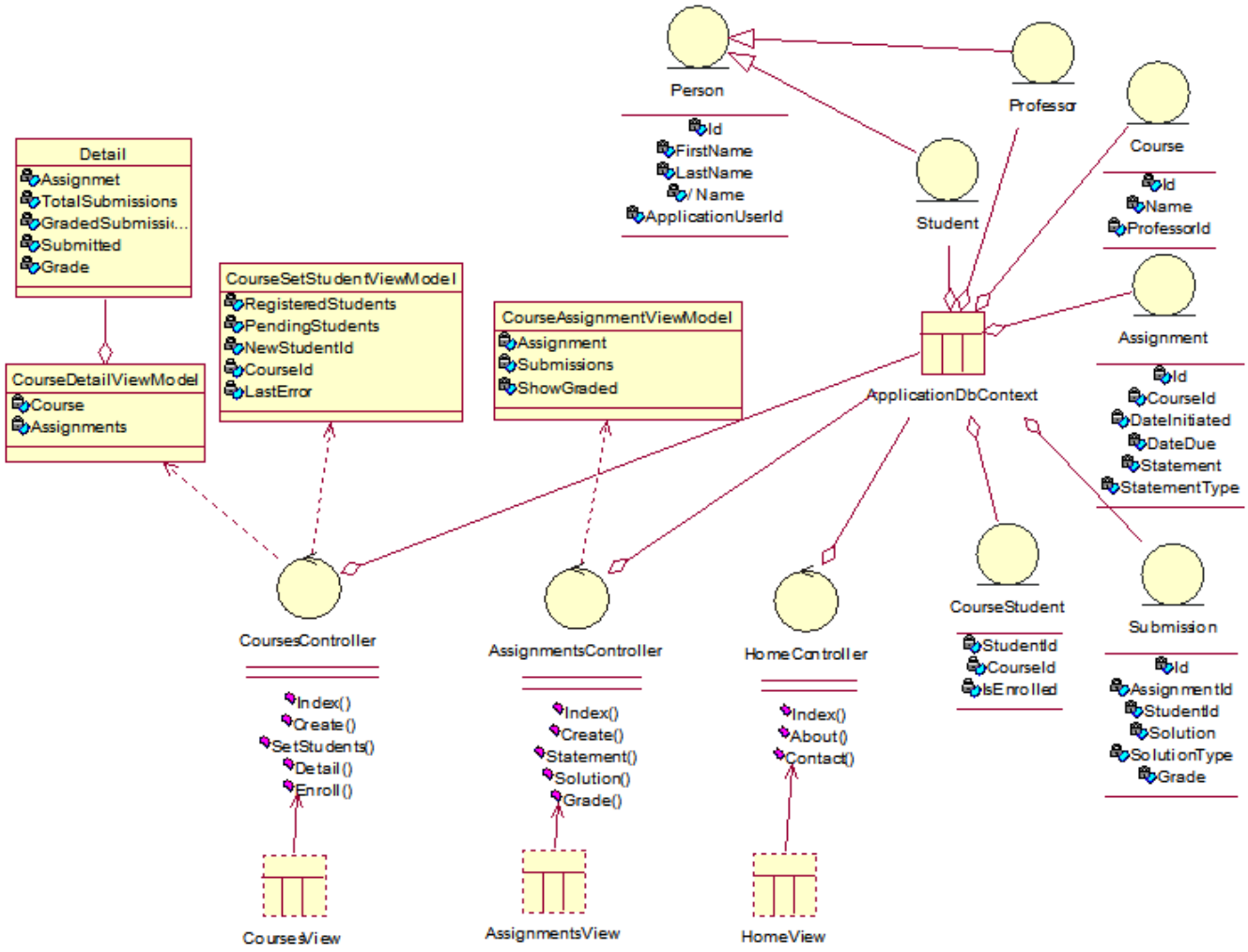
- ***Context Level***



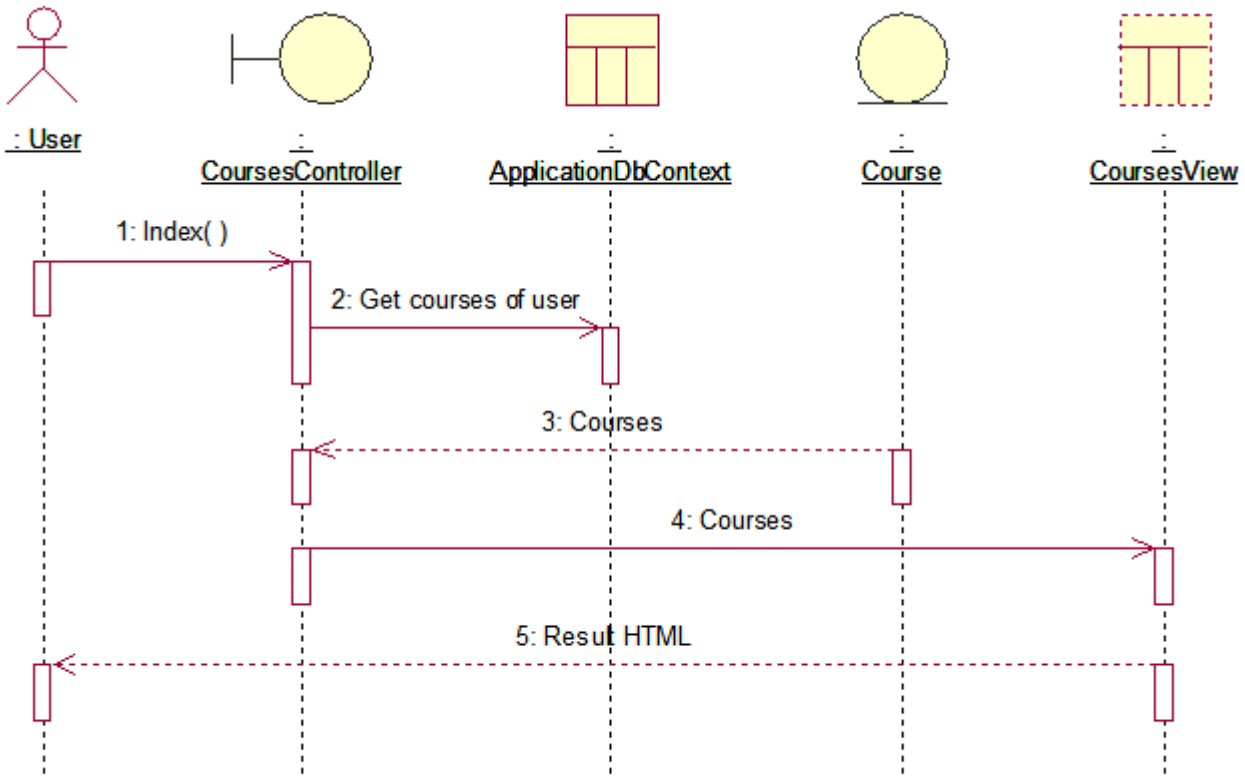
- ***Level 0***

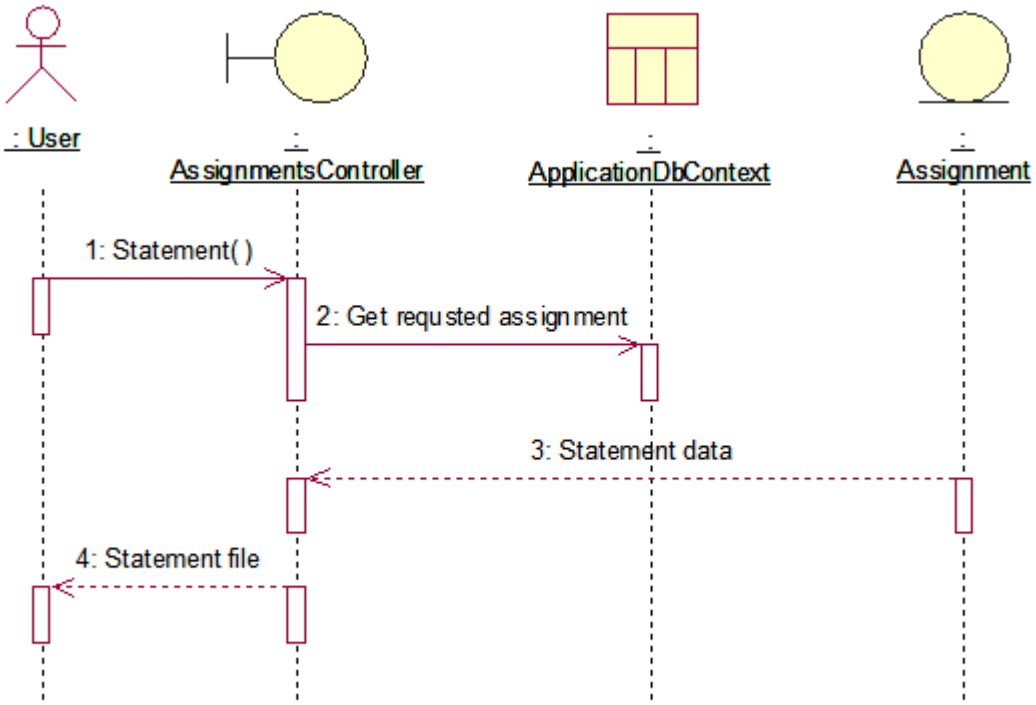


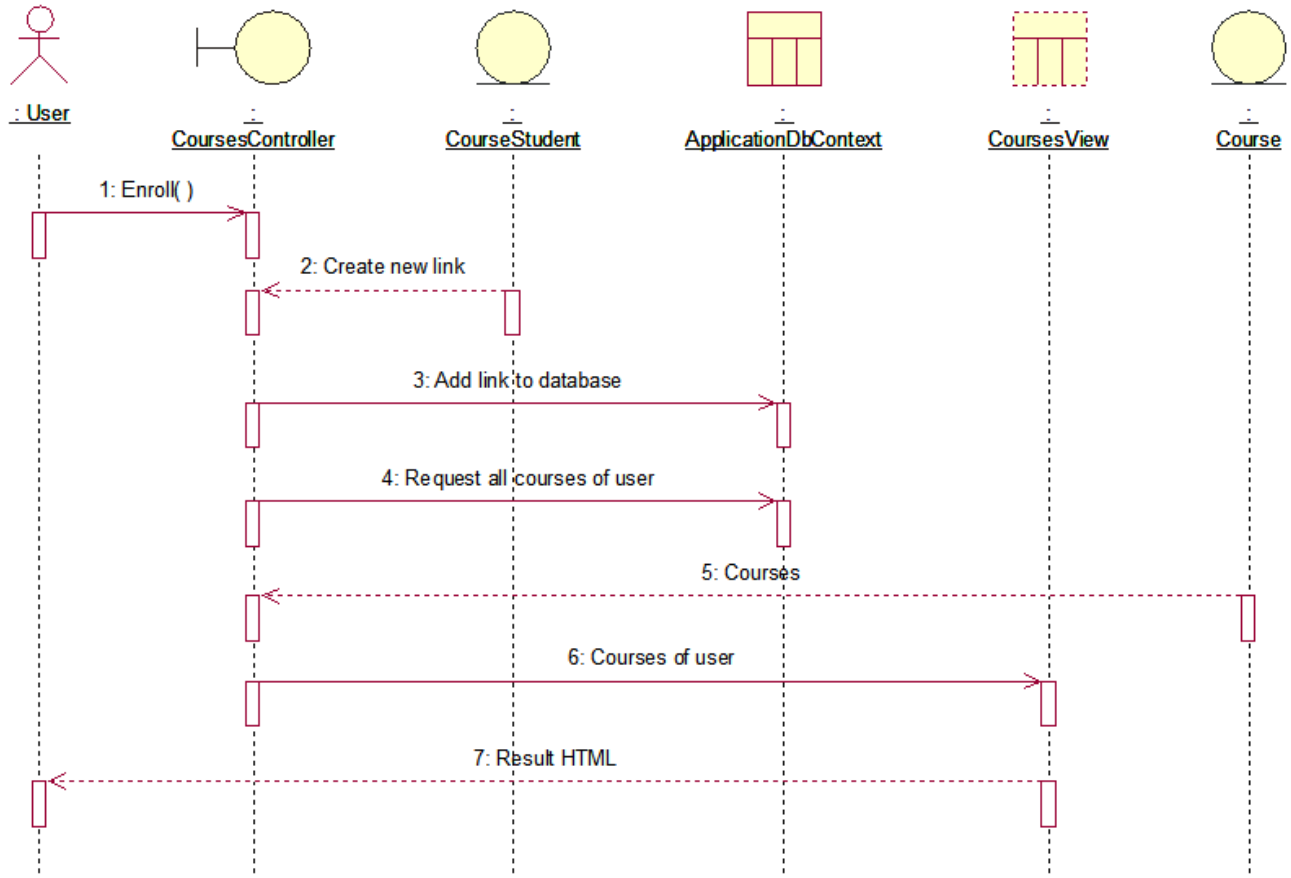
5. Class Diagram

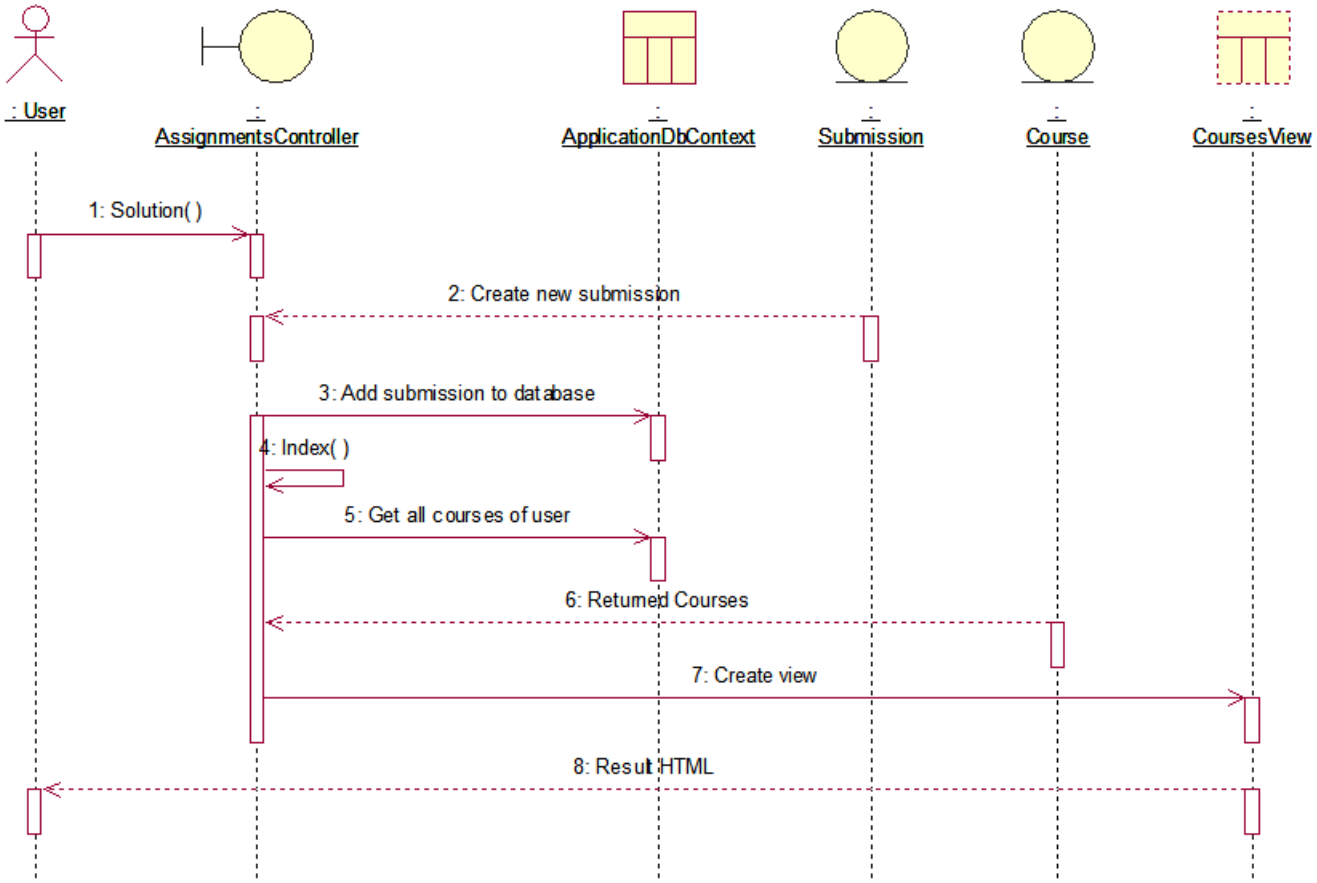


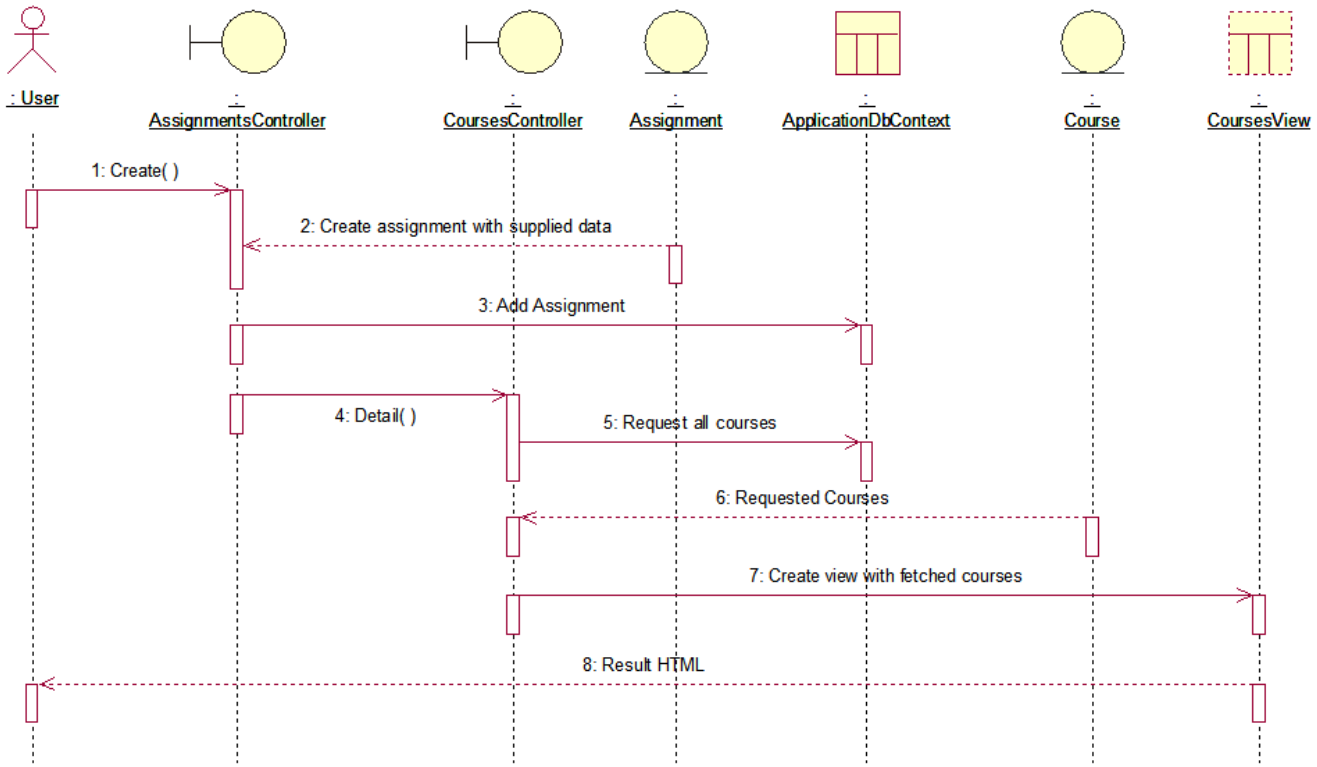
6. Sequence Diagram






: User
: CoursesController
: CourseStudent
: ApplicationDbContext
: CoursesView
: Course
1: Enroll( )
2: Create new link
3: Add link to database
4: Request all courses of user
5: Courses
6: Courses of user
7: Result HTML


: User
: AssignmentsController
: ApplicationDbContext
: Submission
: Course
: CoursesView
1: Solution( )
2: Create new submission
3: Add submission to database
4: Index( )
5: Get all courses of user
6: Returned Courses
7: Create view
8: Result HTML


: User
: AssignmentsController
: CoursesController
: Assignment
: ApplicationDbContext
: Course
: CoursesView
1: Create( )
2: Create assignment with supplied data
3: Add Assignment
4: Detail( )
5: Request all courses
6: Requested Courses
7: Create view with fetched courses
8: Result HTML