Distributed computer system for student project management

Software Design Document

Names:
Milcu Cristian-Mihai
Muga Bogdan-Marius
Ratoiu Cristina-Irina
Lab Section: SE
Workstation: Online

Date:21/03/2021

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

1.2 Scope

1.3 Overview

1.4 Reference Material

1.5 Definitions and Acronyms

2. SYSTEM OVERVIEW

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

3.2 Decomposition Description

3.3 Design Rationale

4. DATA DESIGN

4.1 Data Description

4.2 Data Dictionary

5. COMPONENT DESIGN

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

6.2 Screen Images

6.3 Screen Objects and Actions

7. REQUIREMENTS MATRIX

8. APPENDICES

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Distributed computer system for student project management” software. It will illustrate the purpose and complete a declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications.

1.2 Scope

The system will allow students to submit their projects online to teachers(upload facility) and teachers to evaluate them. The computer system comprises an interface with the user (web browser or dedicated application), a database server that will store information about students, students, grades, application server that manages students, projects, notes and a Web server. The user has access to the service only through the interface with the user. The system will provide facilities to students and teachers after they successfully authenticate [password and identifier] (the identifier and password are already considered generated):

- viewing account information
- password change
- uploading projects by the student and entering the required information
- the teacher’s visualization of the project files and the afferent information, date, student name
- giving grades and comments by teachers

Depending on the type of authentication(student, teacher) the user will be allowed certain functionalities.

1.3 Overview

A detailed system overview is found **on Chapter 2**.

1.4 Reference Material

Nr.Crt	Site
1.	Udemy-Angular8-the Complete Guide
2.	https://angular.io/guide/styleguide

3.	**https://www.c-sharpcorner.com/UploadFile/8a67c0/C-Sharp-coding-standards-and-naming-conventions/**
4.	**https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0**
5.	**https://stackoverflow.com/**

1.5 Definitions and Acronyms

NrCrt	Word	Meaning
1	User	someone who interacts with the application
2	UI	user interface
3	camel Case(camelCase)	naming conventions for variables such that the first letter of the first word is lowercase and the other words are upper cases(rest of the letters are lowercase)
4	Pascal Case(PascalCase)	naming conventions for variables such that the first letter of every word is uppercase(other letters are lowercase)
5	underscore Prefix(_underscore)	naming conventions for variables such that before the camelCase convention it is a _(_underscore) symbol
6	MVC	Model-view-controller
7	HTTP	Hypertext Transfer Protocol

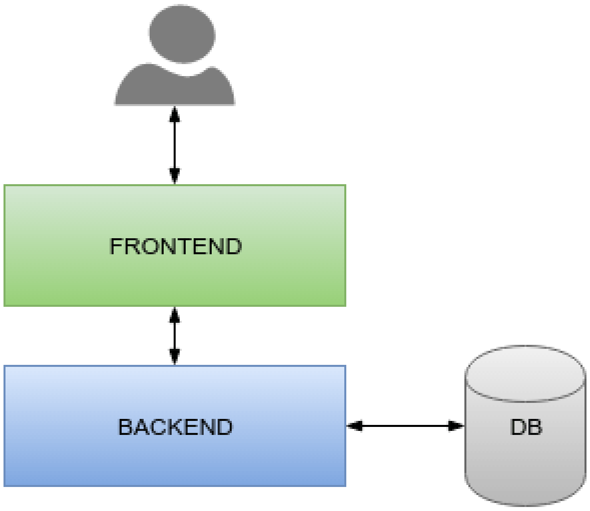
2. System Overview

With this application the users are able to connect to a platform which improves the study quality.
The main users will be students and teachers. They are going to access the platform using different credentials connected to the email address already generated by the faculty.
When a user connects as a teacher he is able to post assignments and evaluate students' work, he can post comments and upload documents related to the subject.
The student-user can see the assignments and upload a document as answer and address questions if something isn’t clear.
This application allows multiple classes to be created. When the purpose for which a class was created no longer exists, it can be deleted.

3. System Architecture

3.1 Architectural Design

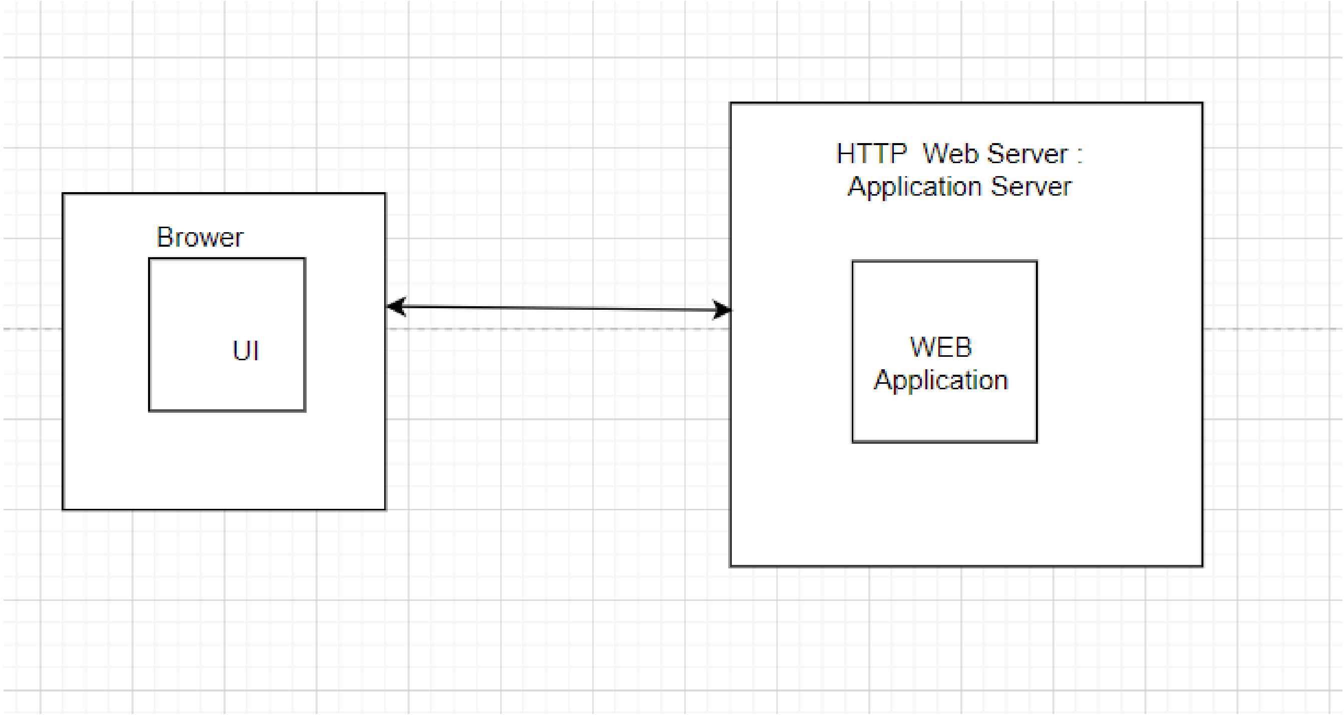
The application is based on three tier architecture.

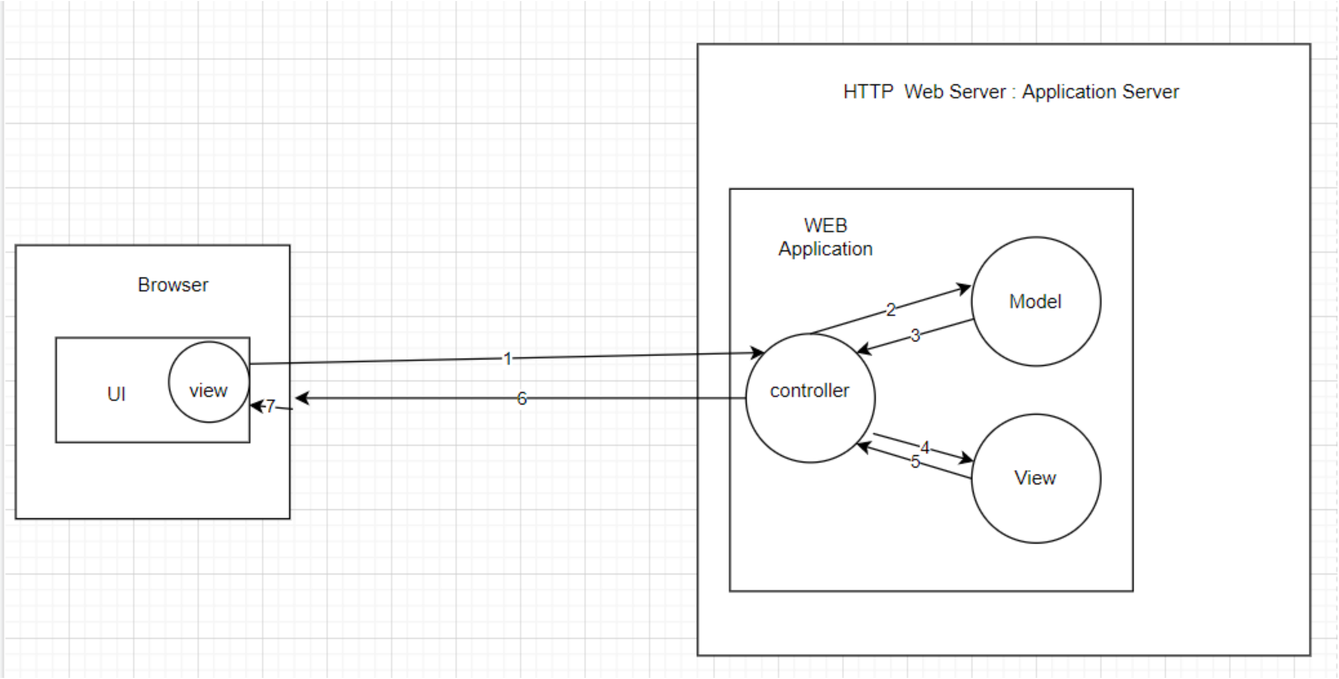


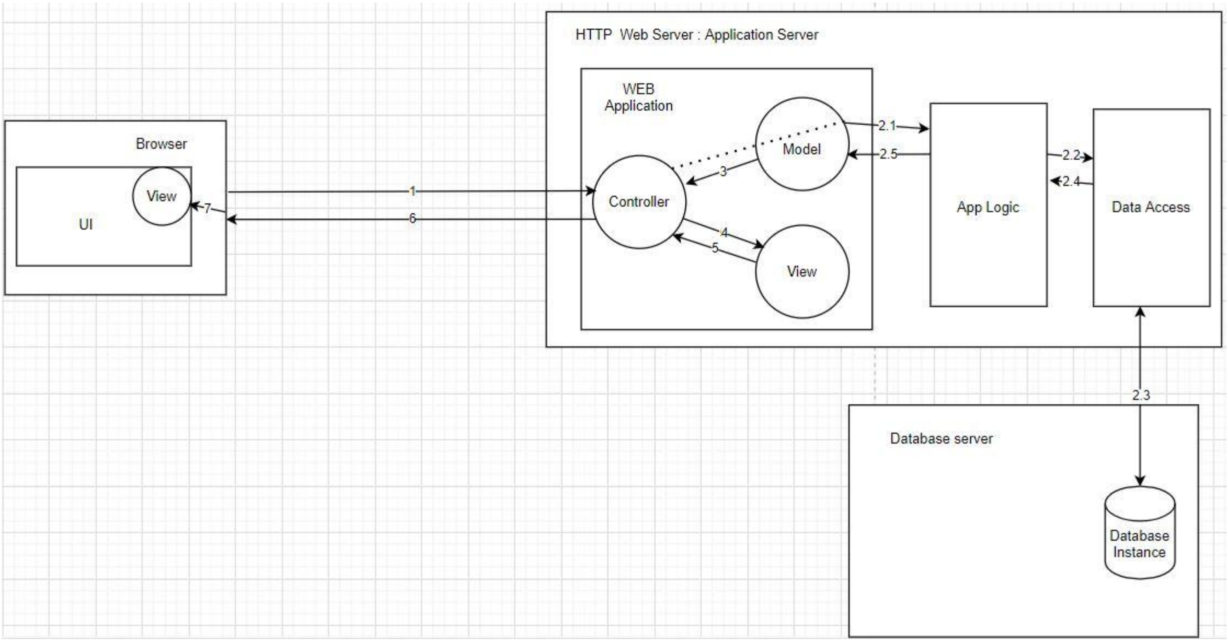
The front end (presentation tier) is the highest level of the application and manages the UI. Our application has login methods and upload documents, post comments. This information will be stored in the database together with the data about students and teachers through the logic tier.

The logic tier coordinates the application, the order processes, makes logical decisions and evaluations and makes calculations. The main role being to transfer and move information between the 2 surrounding layers.
The data tier stores and retrieves the information from a database or file system. Here is stored the information about users, notes, documents, comments. The data transfer is made through the logic tier back to the UI.

3.2 Decomposition Description

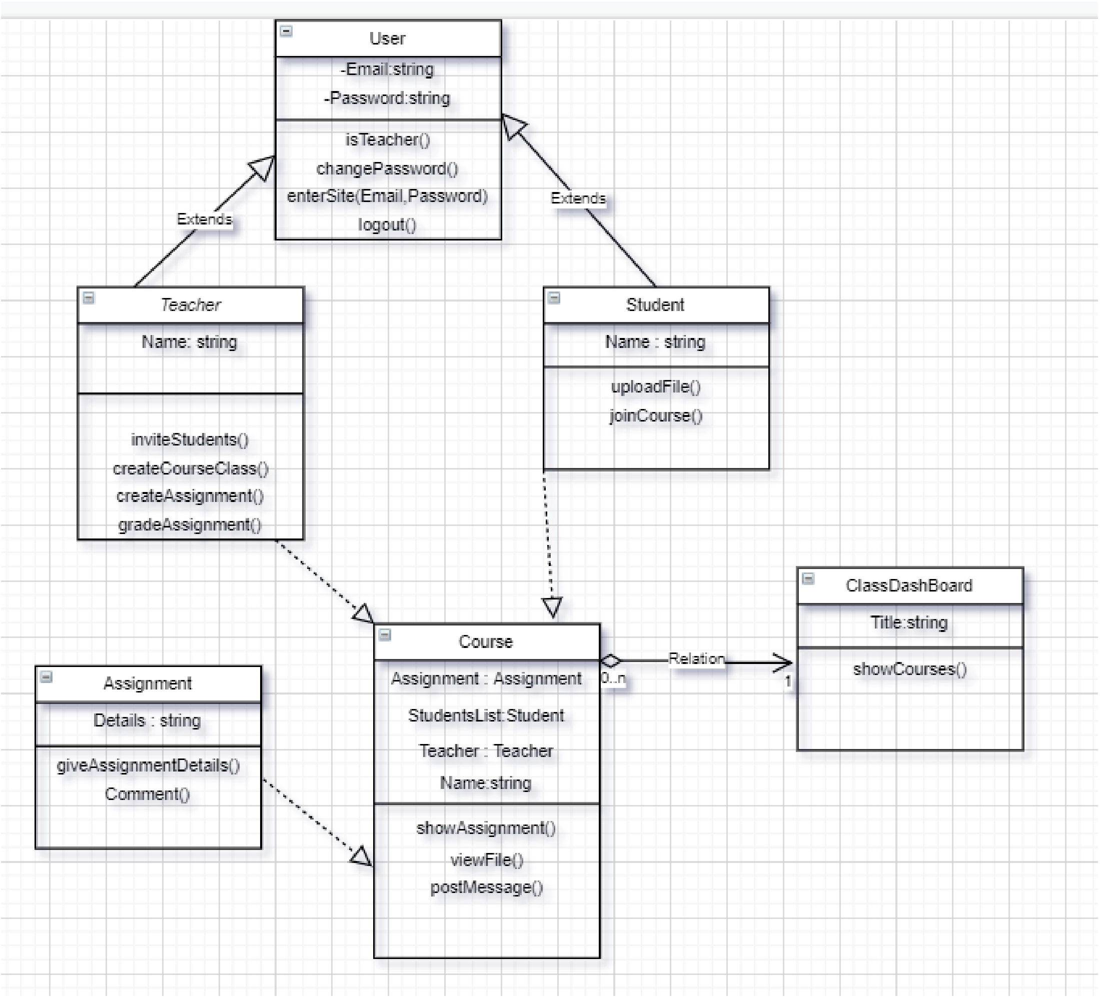






3.3 Design Rationale

At first glance, the three tiers may seem similar to the model-view-controller (MVC) concept; however, topologically they are different. A fundamental rule in a three tier architecture is the client tier never communicates directly with the data tier; in a three-tier model all communication must pass through the middle tier. Conceptually the three-tier architecture is linear. However, the MVC architecture is triangular: the view sends updates to the controller, the controller updates the model, and the view gets updated directly from the model.



Class diagram

4. Data Design

4.1 Data Description

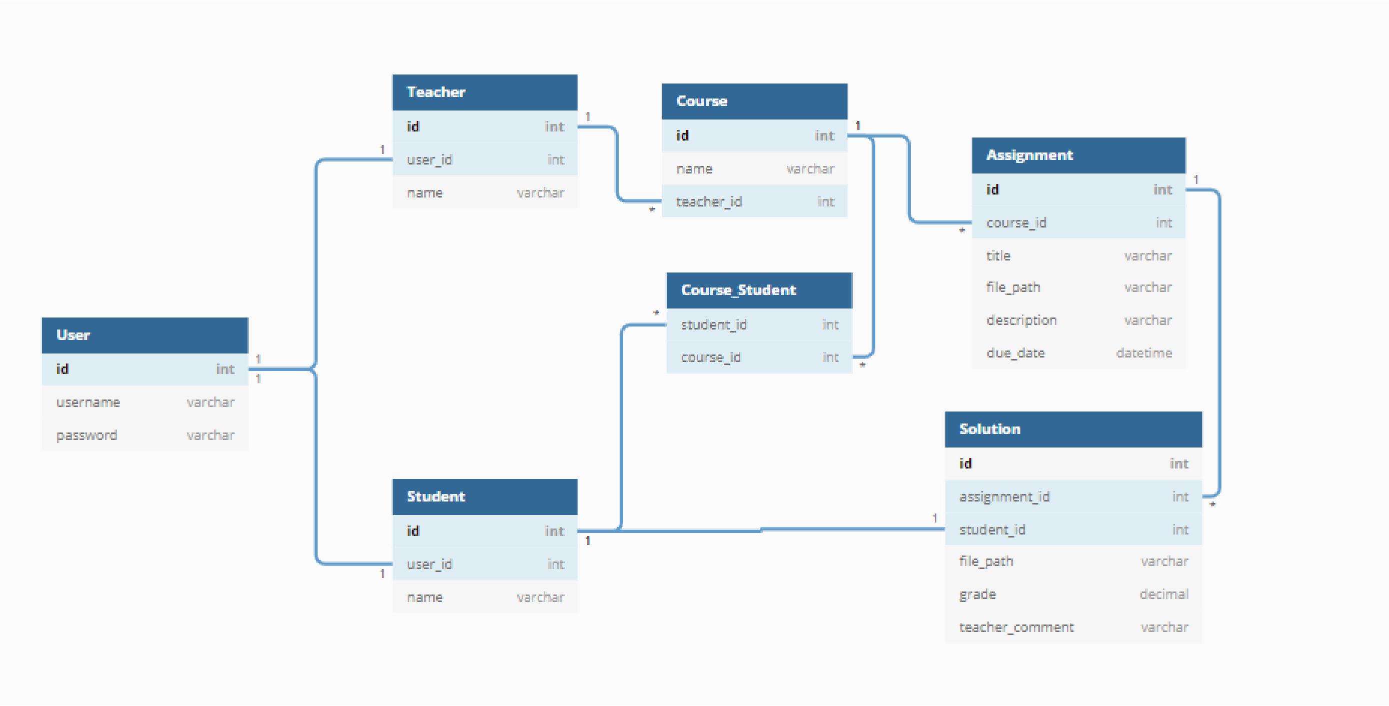
The domain models will be created by Entity Framework using code-first method. We will create our models as classes then using migration we will update our database.

The data and system entities are stored, processed and organized using the relational database management system SQL Server.

The uploaded files will be stored in a folder inside our project, and their paths will be stored as a string inside the database.

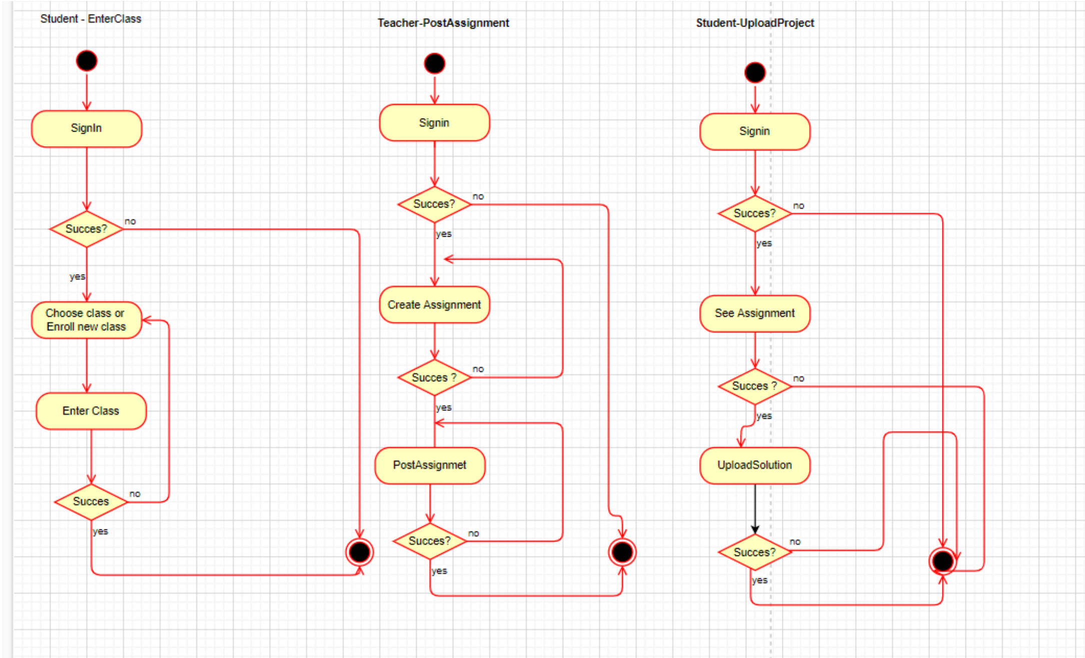
We will have two databases, one for application data management and one for authentication credentials.

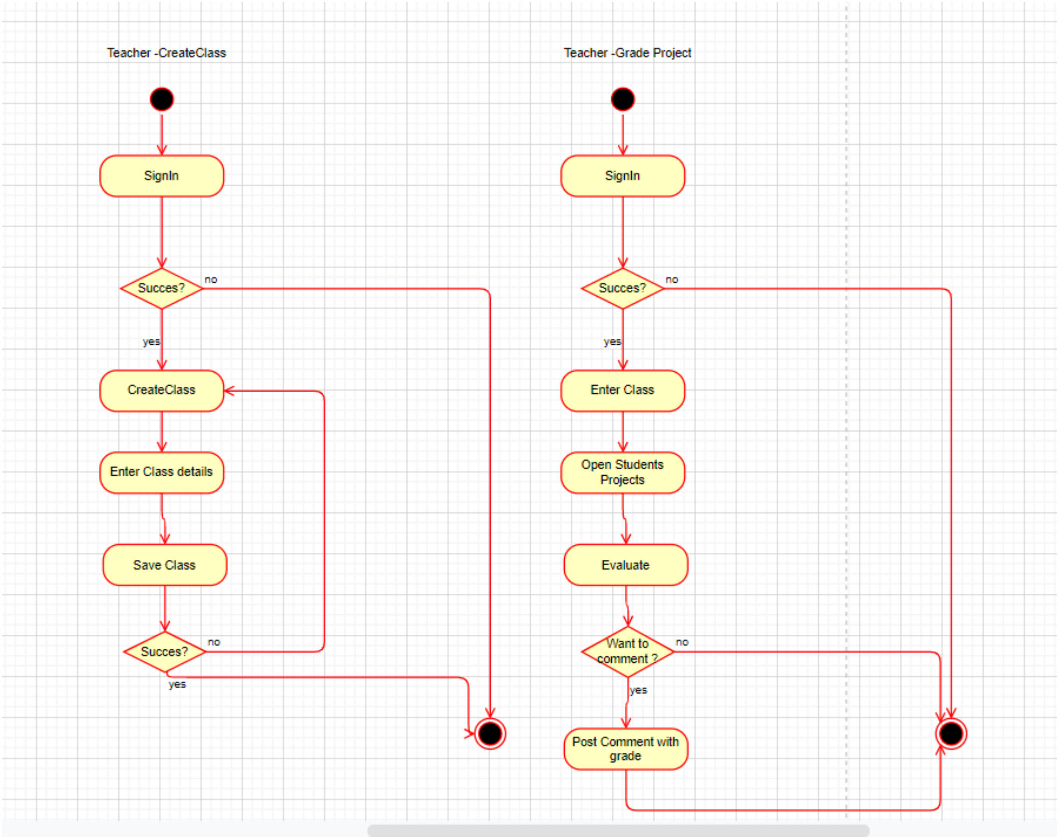
4.2 Data Dictionary

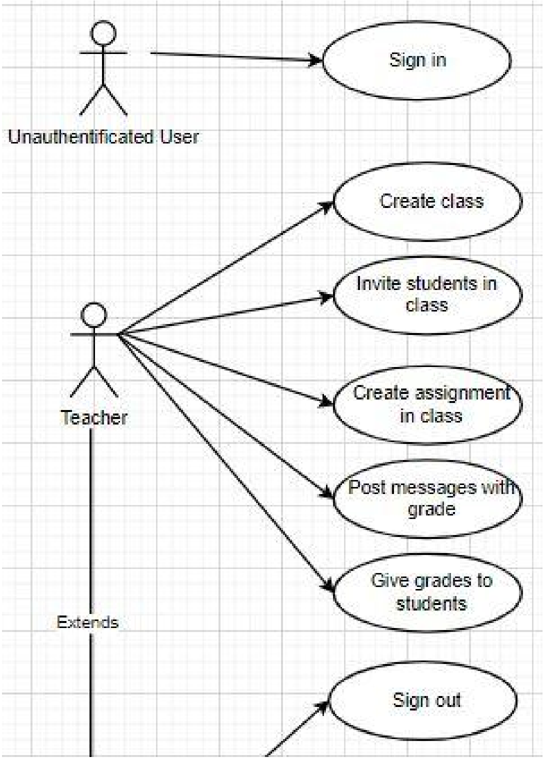


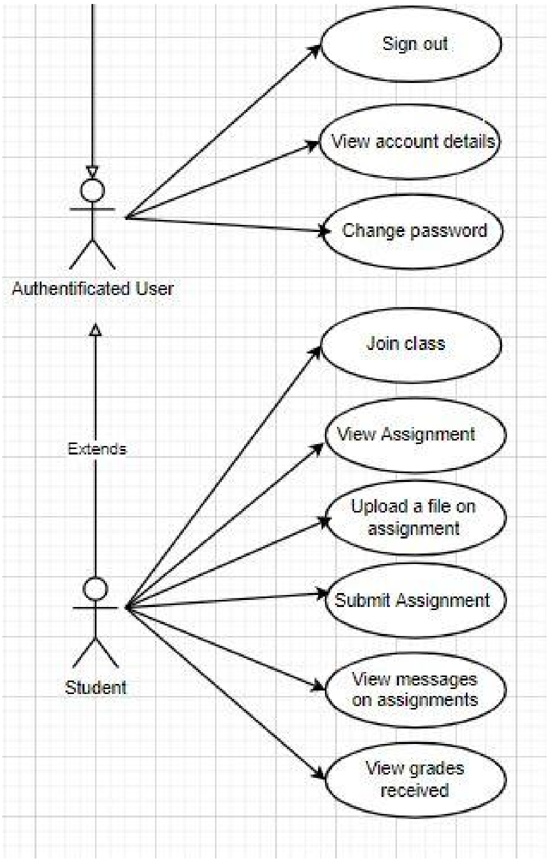
The domain entity map

5. Component Design




Teacher -CreateClass
SignIn
Succes?
no
yes
CreateClass
Enter Class details
Save Class
Succes?
no
yes
Teacher -Grade Project
SignIn
Succes?
no
yes
Enter Class
Open Students Projects
Evaluate
Want to comment ?
no
yes
Post Comment with grade


Unauthentificated User
Sign in
Teacher
Create class
Invite students in class
Create assignment in class
Post messages with grade
Give grades to students
Extends
Sign out

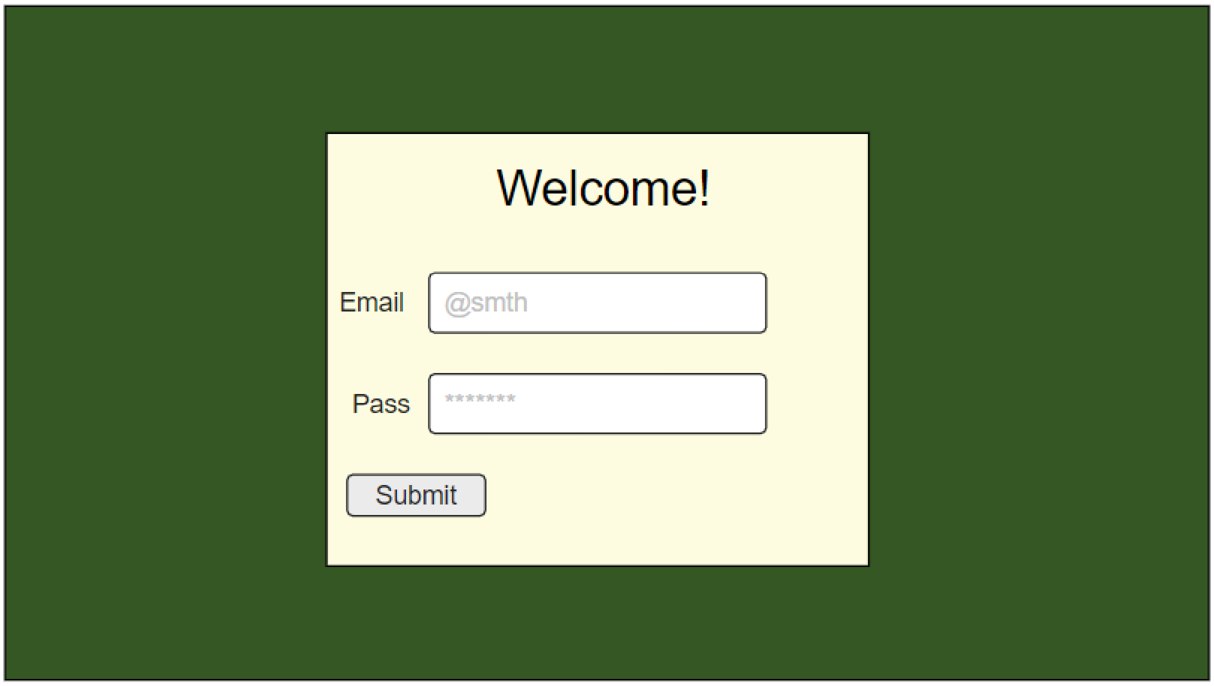


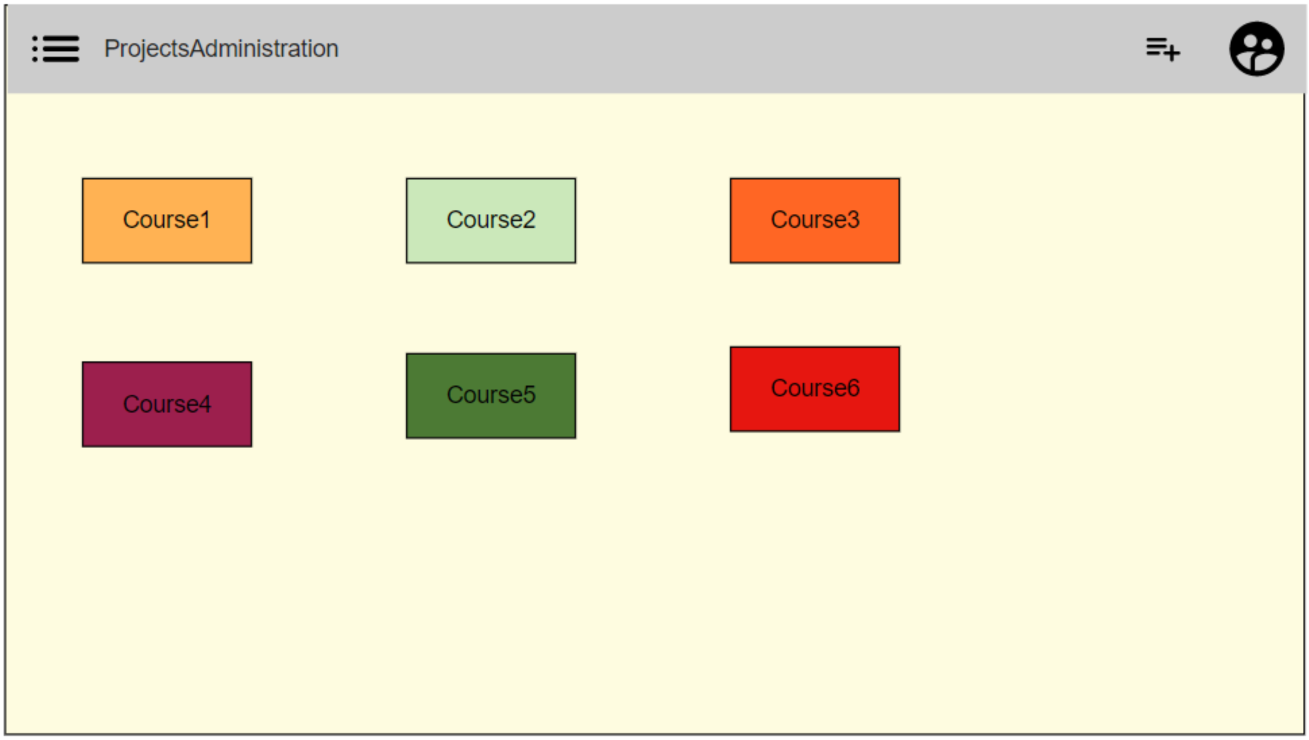
6. Human Interface Design

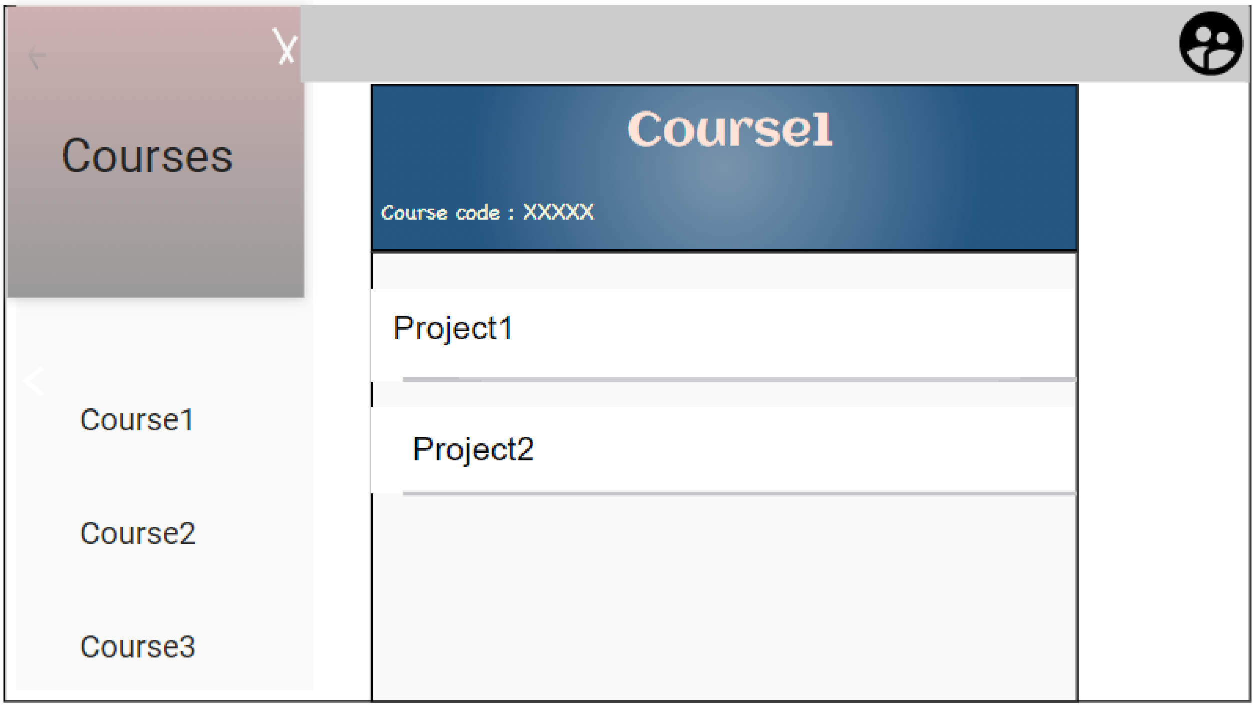
6.1 Overview of User Interface

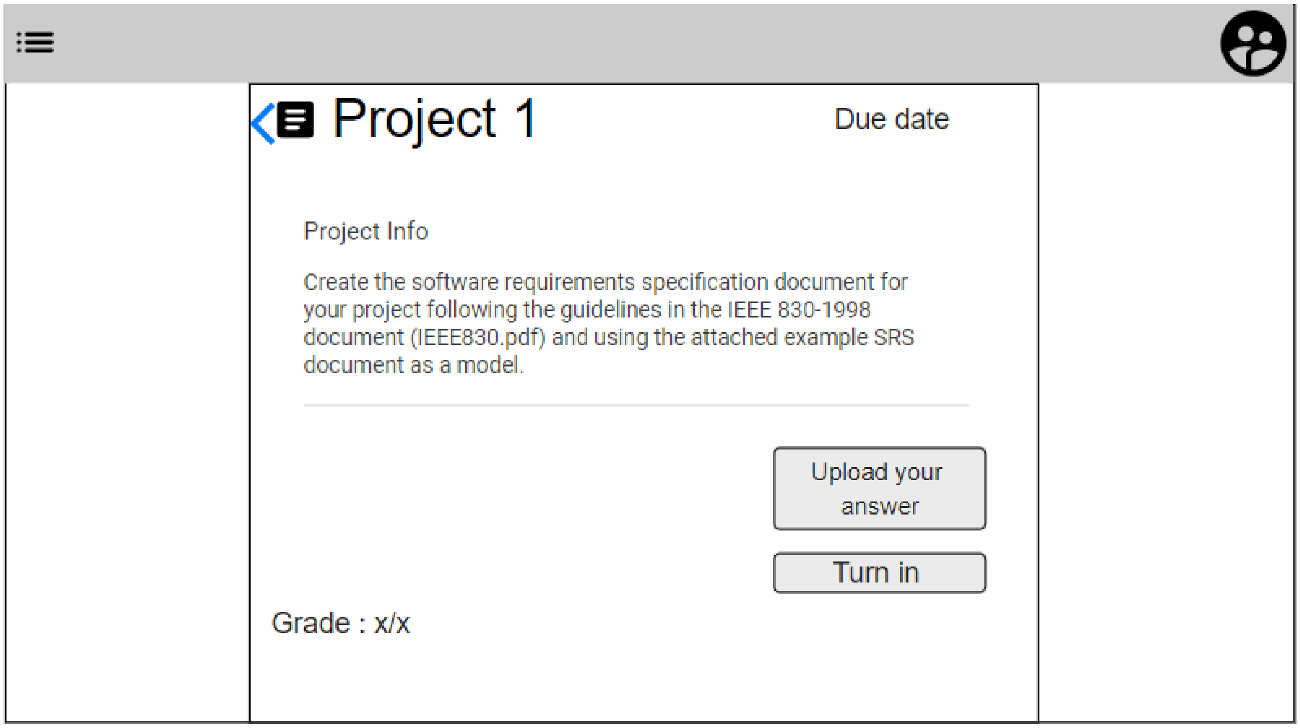
The users, as already described above, should be teachers and students. Often, due to the large number of students participating in a course, project / homework management is difficult.
That is why this application was created to facilitate the retention and management of student projects by their teachers. Thus the loss and damage of documents can be substantially reduced.

6.2 Screen Images



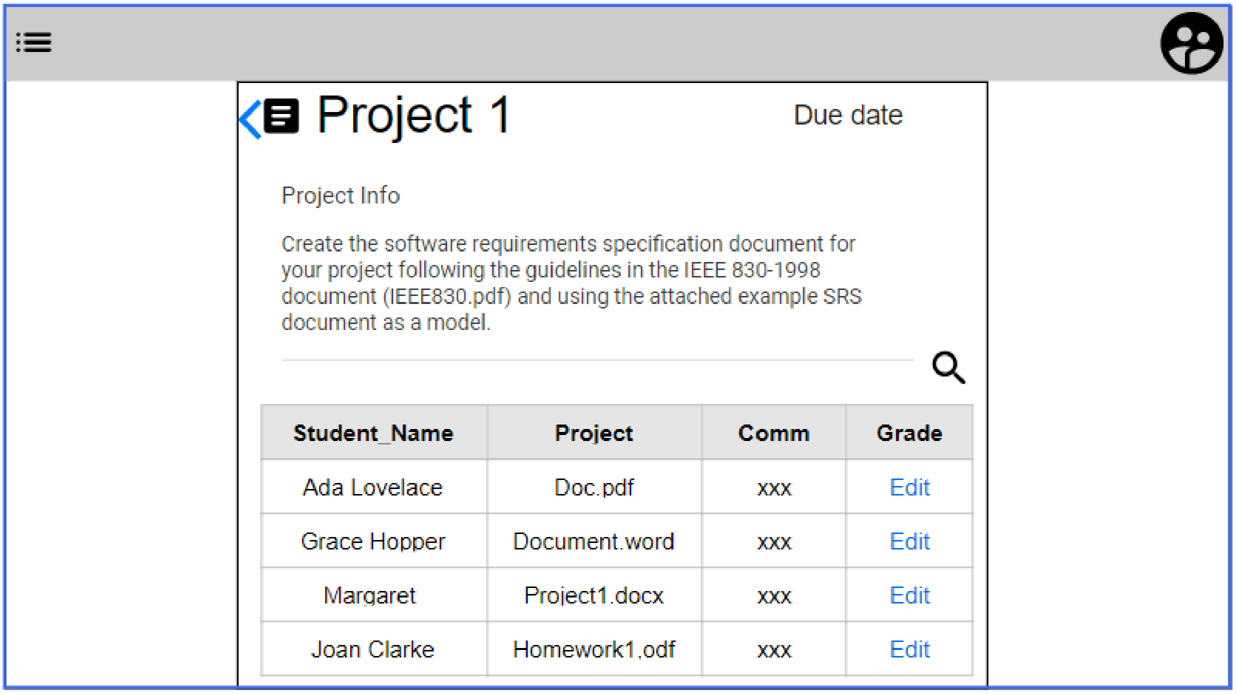






Create the software requirements specification document for your project following the guidelines in the IEEE 830-1998 document (IEEE830.pdf) and using the attached example SRS document as a model.

Grade : x/x



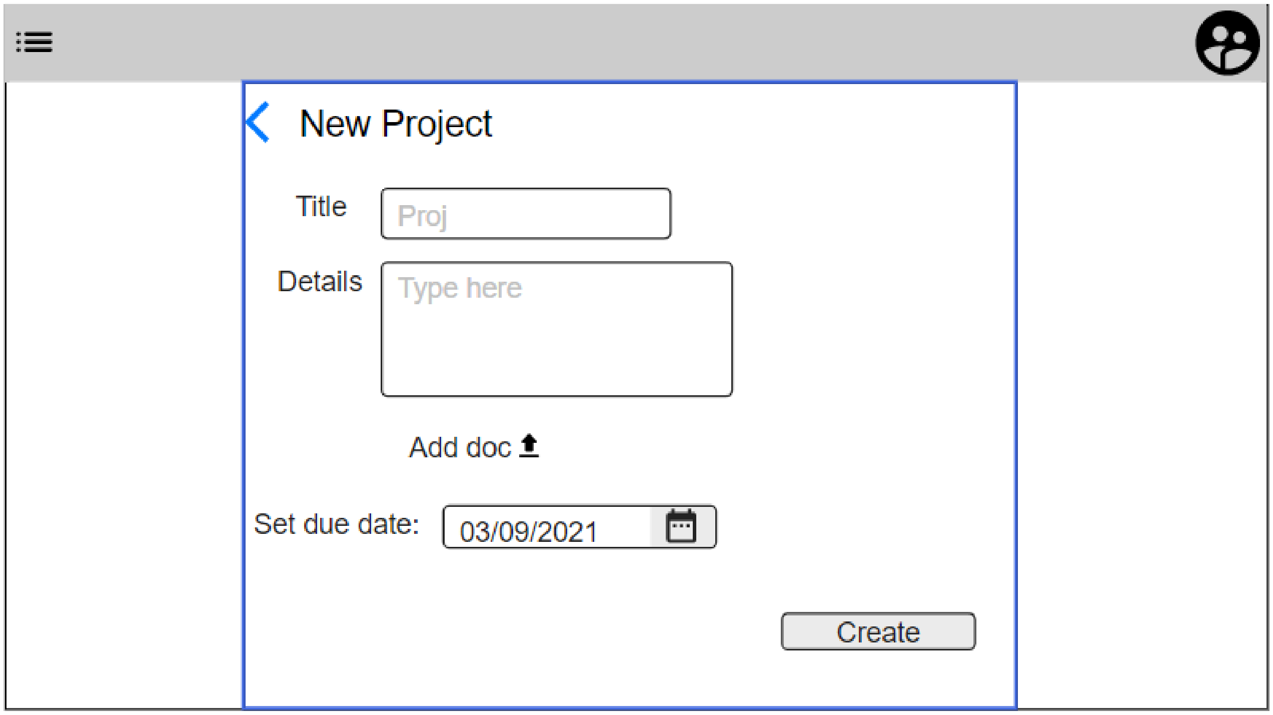
Project 1

Due date

Project Info

Create the software requirements specification document for your project following the guidelines in the IEEE 830-1998 document (IEEE830.pdf) and using the attached example SRS document as a model.

Student_Name	Project	Comm	Grade
Ada Lovelace	Doc.pdf	xxx	Edit
Grace Hopper	Document.word	xxx	Edit
Margaret	Project1.docx	xxx	Edit
Joan Clarke	Homework1.odf	xxx	Edit



New Project

Title: Proj

Details: Type here

Set due date: 03/09/2021

6.3 Screen Objects and Actions

Screen Object	Action
👥 User icon	Show user details
Add course/Class	For teacher: add a new class and generate class code For student : Enter a new class with a code from teacher
BurgerMenu	Shows the list of classes

7. Requirements Matrix

Nr. Crt	**SRS Code**	**Details**
1.	**4.1.3**	Login Screen
2.	**4.2.3**	Student Dashboard
3.	**4.3.3**	Teacher Dashboard
4.	**4.4.3**	Account information screen
5.	**4.5.3**	Student course detail screen
6.	**4.6.3**	Teacher course detail screen
7.	**4.7.3**	Student assignment detail screen
8.	**4.8.3**	Teacher assignment detail screen

8. APPENDICES