



Software Design Document

SEmester

Version:	1.0
Made By:	Chatpisut Chaiuan, Peerachada Limtrakul, Salinporn Rattanaprapaporn, and Thanida Paige Pholsukcharoen
Date of Document:	16/12/2024

Document History

Document Name	Version	Date	Reviewer	Responsible
SEmester_SDD_0.1.docx	0.1	16/12/2024		Chatpisut Chaiuan, Peerachada Limtrakul, Salinporn Rattanaprapaporn, and Thanida Paige Pholsukcharoen
SEmester_SDD_1.0.docx	1.0	18/12/2024		Chatpisut Chaiuan, Peerachada Limtrakul, Salinporn Rattanaprapaporn, and Thanida Paige Pholsukcharoen

Table of Content

1. Document Objectives	4
2. System Architecture	5
3. Detailed design	6
Class Diagram	6
Class Description	7
CD-01: Announcement	7
CD-02: Attendance	9
CD-03: StudentAttendance	11
CD-04: Assignment	13
CD-05: StudentAssignment	15
CD-06: Course	17
CD-07: StudentCourse	21
CD-08: TeacherCourse	23
CD-09: User	25
CD-10: Student	26
CD-11: Teacher	28
CD-12: Main	29
Sequence Diagram	36
SD-1: Teachers can create a course	36
SD-2: Students can enroll in courses	36
SD-3: Students can unenroll course	37
SD-4: Teachers can view assignments	37
SD-5: Teachers can create assignments	38
SD-6: Teachers can grade assignments	38
SD-7: Teachers can delete assignments	39
SD-8: Students can view assignments	39
SD-9: Students can submit assignments	40
SD-10: Students can unsubmit assignments	40
SD-11: Teachers can view announcement posts	41
SD-12: Teachers can create an announcement post	41
SD-13: Teacher can delete announcement post	42
SD-14: Students can view announcement posts	42
SD-15: Teachers can view attendance	43
SD-16: Teachers can create attendance	43
SD-17: Teachers can check attendance	44
SD-18: Students can view attendance	44
SD-19: Teachers can assign grades	45

SD-20: Teachers can edit grades	45
SD-21: Students can view grades for the assignments in a specific course	46
SD-22: Students can view grades for a specific assignment in a specific course	46
SD-23: Students can view all the grades of all the enrolled courses	47
SD-24: Teachers can view all the students' grades in a specific course	47
SD-25: Teachers can view all the students' grades in a specific assignment	48
SD-26: Teachers can view all the classes' grades for all the courses they are teaching	48
User Interface Design	49
Login Interface	49
Register Interface	49
Student's Courses List Interface	50
Student's Course Announcements Interface	51
Student's Course Assignments Interface	52
Student's Grade Interface	53
Student's Attendance Interface	54
Student's Profile Interface	54
Student's Course Details Interface	55
Teacher's Courses List Interface	55
Teacher's Course Announcement Interface	56
Teacher's Course Assignment Interface	57
Teacher's Course Attendance Record Interface	59
Teacher's Grade Interface	61
Teacher's Profile Interface	62
Teacher's Course Details Interface	63

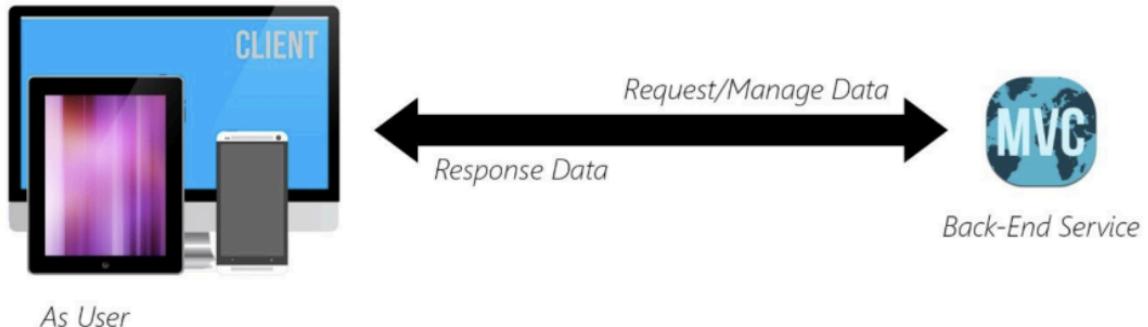
1.Document Objectives

Software Design Document is written for the development team of the software. To make sure that every member of the software development team will understand the design of the system correctly. It also will be used for the project development reference, both for the developing phase and for the other documents.

This version of the Software Design Document, will focus on the design that will be used in the First Progress of Software Development. This document consists of the following sections:

- Class Diagram including the descriptions
- Sequence Diagram including the descriptions
- Graphic User Interface

2. System Architecture



Firstly, the user registers into the web application site by visiting the server address. Then, if the registration information passes the server validation process, the server will store the registration information in the database and create the user of the web application with that registration information.

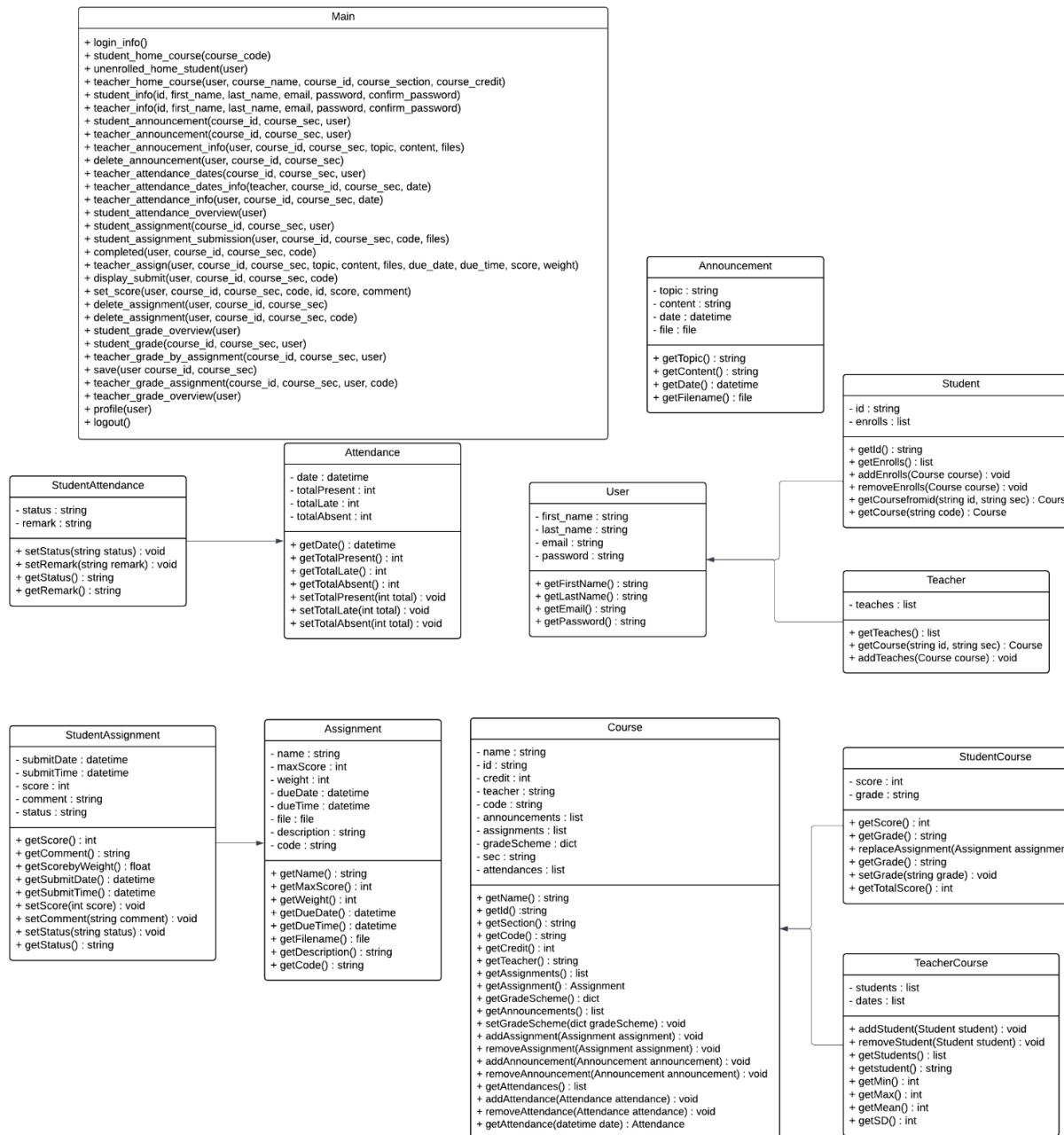
After that, the user is able to log in with the registered Email Address and Password in order to use the web application features from the server. The web application features consist of Teacher view and Student view.

For Teacher view, the user can create courses that they teach. After the user requests for course creation and the information passes the server validation process, the server will store the course information in the database and create the course of the web application with that course information for the user. From that point forward, the user can obtain the course code to further give to the students for them to enroll in the course. Once students are enrolled in the course, the user can give assignments, announce news and class materials, provide and receive details of students' attendance records, and announce students' grades. This information will be stored on the server after this information passes the server validation process, and the user request for each action has been sent to the Server.

For the Student view, the user can enroll in courses with the course code given by the teacher. After the student requests for enrollment in a course and the information passes the server validation process, the server will map the student to the course in the database and give the user access to the course. From this point onwards, the user can submit assignments, view announced news and class materials, check their attendance records, and see students' grades. This information will be stored on the server after this information passes the server validation process, and the user request for each action has been sent to the Server.

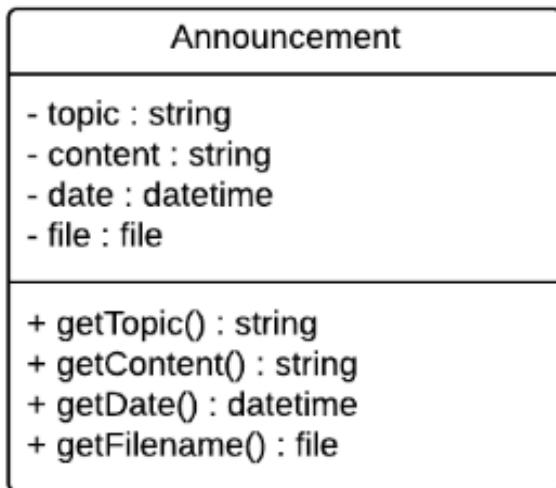
3. Detailed design

Class Diagram



Class Description

CD-01: Announcement



Entities

ID	Name	Description	Type
1	topic	Topic of Announcement	string
2	content	Content inside Announcement	string
3	date	Date of Announcement	datetime
4	file	File of Announcement	file

Method Detail

Method Name: getTopic

public string getTopic()

Description: Topic of the announcement

Returns: string

Method Name: getContent

public string getContent()

Description: Content of the announcement

Returns: string

Method Name: getDate

public datetime getDate()

Description: Date of the announcement

Returns: datetime

Method Name: getFilename

public file getFilename()

Description: File associated with the announcement

Returns: file

CD-02: Attendance

Attendance
- date : datetime - totalPresent : int - totalLate : int - totalAbsent : int
+ getDate() : datetime + getTotalPresent() : int + getTotalLate() : int + getTotalAbsent() : int + setTotalPresent(int total) : void + setTotalLate(int total) : void + setTotalAbsent(int total) : void

Entities

ID	Name	Description	Type
1	date	Date of Announcement	datetime
2	totalPresent	Number of present students	int
3	totalLate	Number of late students	int
4	totalAbsent	Number of absent students	int

Method Detail

Method Name: getDate

public datetime getDate()

Description: The method to get the due date of the attendance

Returns: datetime

Method Name: getTotalPresent

public int getTotalPresent()

Description: The method to get the number of present students of the attendance

Returns: int

Method Name: getTotalLate

public int getTotalLate()

Description: The method to get the number of late students of the attendance

Returns: int

Method Name: getTotalAbsent

public int getTotalAbsent()

Description: The method to get the number of absent students of the attendance

Returns: int

Method Name: setTotalPresent

public void setTotalPresent(int total)

Description: The method to set the number of the Present Student

Parameters: int total

Returns: void

Method Name: setTotalLate

public void setTotalLate(int total)

Description: The method to set the number of the Late Student

Parameters: int total

Returns: void

Method Name: setTotalAbsent

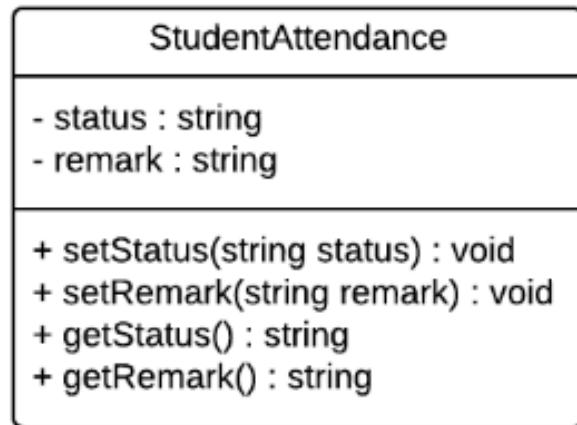
public void setTotalAbsent(int total)

Description: The method to set the number of the Absent Student

Parameters: int total

Returns: void

CD-03: StudentAttendance



Entities

ID	Name	Description	Type
1	status	Status of Student's Attendance	string
2	remark	Remark on Student's Attendance	string

Method Detail

Method Name: setStatus

public void setStatus(string status)

Description: The method to set the status of the Student Attendance

Parameters: string status

Returns: void

Method Name: setRemark

Public void setRemark(string remark)

Description: The method to set the remark of the Student Attendance

Parameters: string remark

Returns: void

Method Name: getStatus

public string getStatus ()

Description: The method to get the status of the Student Attendance

Returns: string

Method Name: getRemark

public string getRemark()

Description: The method to get the remark of the Student Attendance

Returns: string

CD-04: Assignment

Assignment	
- name : string	
- maxScore : int	
- weight : int	
- dueDate : datetime	
- dueTime : datetime	
- file : file	
- description : string	
- code : string	
+ getName() : string	
+ getMaxScore() : int	
+ getWeight() : int	
+ getDueDate() : datetime	
+ getDueTime() : datetime	
+ getFilename() : file	
+ getDescription() : string	
+ getCode() : string	

Entities

ID	Name	Description	Type
1	name	Name of Assignment	string
2	maxScore	Max score of Assignment	int
3	weight	Weight of Assignment	int
4	dueDate	Due date of Assignment	datetime
5	dueTime	Due time of Assignment	datetime
6	file	File of Assignment	file
7	description	Description of Assignment	string
8	code	Identification code of Assignment	string

Method Detail

Method Name: getName

public string getName()

Description: The method to get the name of the Assignment

Returns: string

Method Name: getMaxScore

public int getMaxScore()

Description: The method to get the max score of the Assignment

Returns: int

Method Name: getWeight

public int getWeight()

Description: The method to get the weight of the Assignment

Returns: int

Method Name: getDueDate

public datetime getDueDate()

Description: The method to get the due date of the Assignment

Returns: datetime

Method Name: getDueTime

public datetime getDueTime()

Description: The method to get the due time of the Assignment

Returns: datetime

Method Name: getFilename

public file getFilename()

Description: The method to get the file of the Assignment

Returns: file

Method Name: getDescription

public string getDescription()

Description: The method to get the description of the Assignment

Returns: string

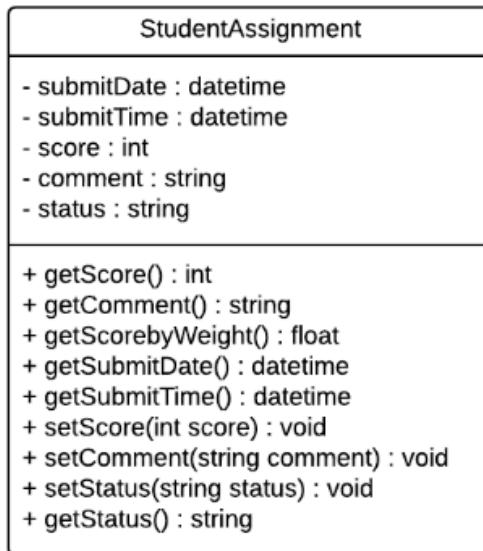
Method Name: getCode

public string getCode()

Description: The method to get the code of the Assignment

Returns: string

CD-05: StudentAssignment



Entities

ID	Name	Description	Type
1	submitDate	Date Assignment that Student submitted	datetime
2	submitTime	Time Assignment that Student submitted	datetime
3	score	Score of Student's Assignment	int
4	comment	Comment on Student's Assignment	string
5	status	Assignment's Status	string

Method Detail

Method Name: getScore

```
public int getScore()
```

Description: The method to get the score of the Assignment

Returns: int

Method Name: getComment

```
public string getComment()
```

Description: The method to get the comment of the Assignment

Returns: string

Method Name: getScorebyWeight

public float getScorebyWeight()

Description: The method to get the score by weight of the Assignment

Returns: float

Method Name: getSubmitDate

public datetime getSubmitDate()

Description: The method to get the submit date of the Assignment

Returns: datetime

Method Name: getSubmitTime

public datetime getSubmitTime()

Description: The method to get the submit time of the Assignment

Returns: datetime

Method Name: setScore

public int setScore(int score)

Description: The method to set the score of the Assignment

Parameters: int total

Returns: void

Method Name: setComment

public int setComment(string comment)

Description: The method to set the comment of the Assignment

Parameters: string comment

Returns: void

Method Name: setStatus

public int setStatus(string status)

Description: The method to set the status of the Assignment

Parameters: string status

Returns: void

Method Name: getStatus

public string getStatus()

Description: The method to get the status of the Assignment

Returns: string

CD-06: Course

Course	
- name : string	
- id : string	
- credit : int	
- teacher : string	
- code : string	
- announcements : list	
- assignments : list	
- gradeScheme : dict	
- sec : string	
- attendances : list	
+ getName() : string	
+ getId() : string	
+ getSection() : string	
+ getCode() : string	
+ getCredit() : int	
+ getTeacher() : string	
+ getAssignments() : list	
+ getAssignment() : Assignment	
+ getGradeScheme() : dict	
+ getAnnouncements() : list	
+ setGradeScheme(dict gradeScheme) : void	
+ addAssignment(Assignment assignment) : void	
+ removeAssignment(Assignment assignment) : void	
+ addAnnouncement(Announcement announcement) : void	
+ removeAnnouncement(Announcement announcement) : void	
+ getAttendances() : list	
+ addAttendance(Attendance attendance) : void	
+ removeAttendance(Attendance attendance) : void	
+ getAttendance(datetime date) : Attendance	

Entities

ID	Name	Description	Type
1	name		string
2	id		string
3	credit		int
4	teacher		string
5	code		string
6	announcement		list
7	assignments		list
8	gradeScheme		dict
9	sec		string
10	attendances		list

Method Detail

Method Name: getName

public string getName()

Description: The method to get the name of the Course

Returns: string

Method Name: getId

public string getId()

Description: The method to get the ID of the Course

Returns: string

Method Name: getSection

public string getSection()

Description: The method to get the section of the Course

Returns: string

Method Name: getCode

public string getCode()

Description: The method to get the code of the Course

Returns: string

Method Name: getCredit

public int getCode()

Description: The method to get the credit of the Course

Returns: int

Method Name: getTeacher

public string getTeacher()

Description: The method to get the teacher of the Course

Returns: string

Method Name: getAssignments

public list getAssignments()

Description: The method to get the assignments of the Course

Returns: list

Method Name: getAssignment

public Assignment getAssignment()

Description: The method to get the assignment of the Course

Returns: Assignment

Method Name: getGradeScheme

public dict getGradeScheme()

Description: The method to get the grade scheme of the Course

Returns: dict

Method Name: getAnnouncements

public list getAnnouncements()

Description: The method to get the announcements of the Course

Returns: list

Method Name: setGradeScheme

public void setGradeScheme(dict gradeScheme)

Description: The method to set the grade scheme of the Course

Parameters: dict gradeScheme

Returns: void

Method Name: addAssignment

public void addAssignment(Assignment assignment)

Description: The method to add the assignment to the Course

Parameters: Assignment assignment

Returns: void

Method Name: removeAssignment

public void removeAssignment(Assignment assignment)

Description: The method to remove the assignment from the Course

Parameters: Assignment assignment

Returns: void

Method Name: addAnnouncement

public void addAnnouncement(Announcement announcement)

Description: The method to add the announcement to the Course

Parameters: Announcement announcement

Returns: void

Method Name: removeAnnouncement

public void removeAnnouncement(Announcement announcement)

Description: The method to remove the announcement from the Course

Parameters: Announcement announcement

Returns: void

Method Name: getAttendances

public list getAttendances()

Description: The method to get the attendances of the Course

Returns: list

Method Name: addAttendance

public void addAttendance(Attendance attendance)

Description: The method to add the attendance to the Course

Parameters: Attendance attendance

Returns: void

Method Name: removeAttendance

public void removeAttendance(Attendance attendance)

Description: The method to remove the attendance from the Course

Parameters: Attendance attendance

Returns: void

Method Name: getAttendance

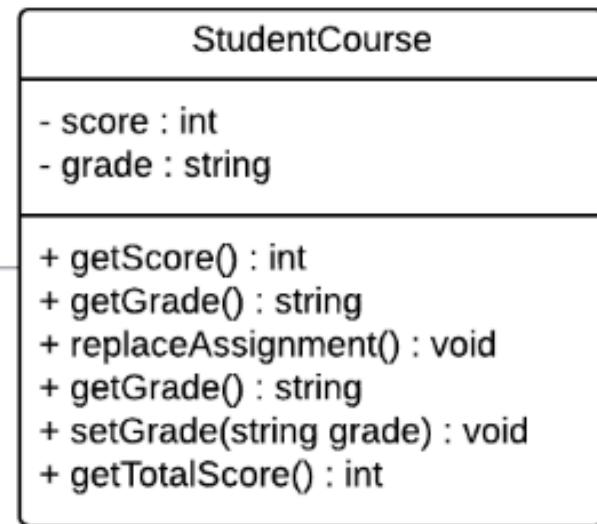
public Attendance getAttendance(datetime date)

Description: The method to get the attendance of the Course

Parameters: datetime date

Returns: Attendance

CD-07: StudentCourse



Entities

ID	Name	Description	Type
1	score	Student's score in Course	int
2	grade	Student's letter grade in Course	string

Method Detail

Method Name: getScore

public int getScore()

Description: The method to get the score on the Course from Student

Returns: int

Method Name: getGrade

public string getGrade()

Description: The method to get the calculated letter grade using the grading scheme inside Course

Returns: string

Method Name: replaceAssignment

public void replaceAssignment(Assignment assignment)

Description: The method to remove Assignment in assignments with the same code as the input Assignment's code and then append the input Assignment

Parameters: Assignment assignment

Returns: void

Method Name: getGrade

public string getGrade()

Description: The method to get the Student's grade in Course

Returns: string

Method Name: setGrade

public void setGrade()

Description: The method to set the grade of Student in Course

Returns: void

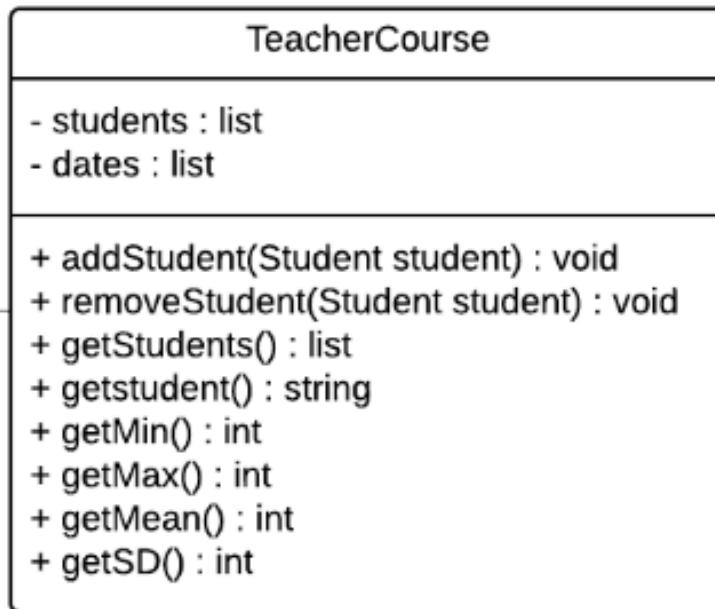
Method Name: getTotalScore

public int getTotalScore()

Description: The method to get the calculated total score of all the assignments of Student in Course

Returns: int

CD-08: TeacherCourse



Entities

ID	Name	Description	Type
1	students	List of students in Teacher's Course	list
2	dates	Dates of Teacher's Course	list

Method Name: addStudent

```
public void addStudent(Student student)
```

Description: The method to add the student to the Course

Parameters: Student student

Returns: void

Method Name: removeStudent

```
public void removeStudent(Student student)
```

Description: The method to remove the student from the Course

Parameters: Student student

Returns: void

Method Name: getStudents

```
public list getStudents()
```

Description: The method to get the students of the Course

Returns: list

Method Name: getStudent

public string getStudent()

Description: The method to get the student of the Course

Returns: string

Method Name: getMin

public int getMin()

Description: The method to get the min score of the Course

Returns: int

Method Name: getMax

public int getMax()

Description: The method to get the max score of the Course

Returns: int

Method Name: getMean

public int getMean()

Description: The method to get the mean score of the Course

Returns: int

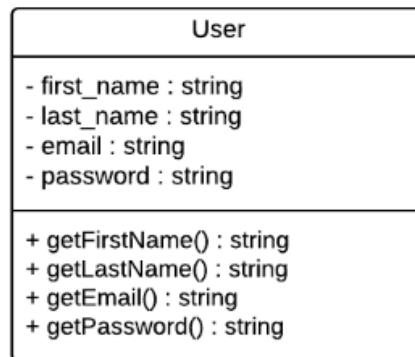
Method Name: getSD

public int getSD()

Description: The method to get the SD score of the Course

Returns: int

CD-09: User



Entities

ID	Name	Description	Type
1	first_name	User's first name	string
2	last_name	User's last name	string
3	email	User's email address	string
4	password	User's password	string

Method Detail

Method Name: getFirstName

public string getFirstName()

Description: The method to get the first name of User

Returns: string

Method Name: getLastname

public string getLastname()

Description: The method to get the last name of User

Returns: string

Method Name: getEmail

public string getEmail()

Description: The method to get the email address of User

Returns: string

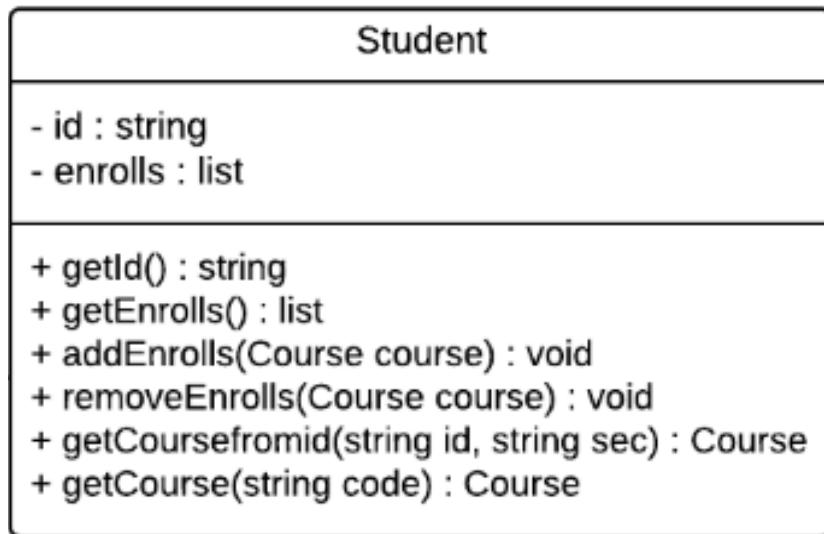
Method Name: getPassword

public string getPassword()

Description: The method to get the password of User

Returns: string

CD-10: Student



Entities

ID	Name	Description	Type
1	id	Student's identification number	string
2	enrolls	Student's enrollment list	list

Method Detail

Method Name: getId

```
public string getId()
```

Description: The method to get identification number of Student

Returns: string

Method Name: getEnrolls

```
public list getEnrolls()
```

Description: The method to get the courses Student enrolled in

Returns: list

Method Name: addEnrolls

```
public void addEnrolls(Course course)
```

Description: The method to add Course into the list of enrollments of Student

Returns: void

Method Name: removeEnrolls

public void removeEnrolls(Course course)

Description: The method to remove Course from the list of enrollments

Returns: void

Method Name: getCoursefromid

public Course getCoursefromid(string id, string sec)

Description: The method to get Course from the list of enrollments of Student using the identification number and section.

Returns: Course

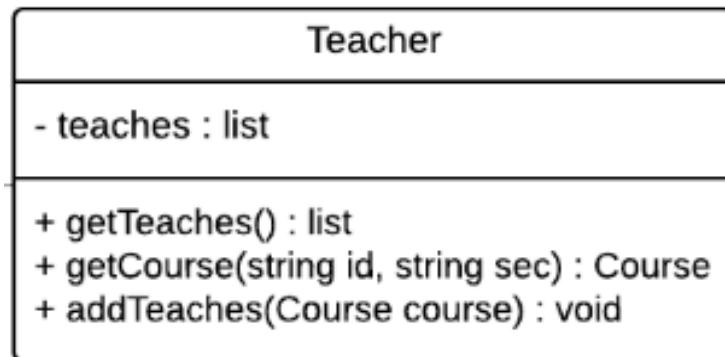
Method Name: getCourse

public Course getCourse(string code)

Description: The method to get Course from Student's list of enrollments using code

Returns: Course

CD-11: Teacher



Entities

ID	Name	Description	Type
1	teaches	List of courses Teacher taught	list

Method Name: getTeaches

public list getTeaches()

Description: The method to get all the courses that the teacher teach

Returns: list

Method Name: getCourse

public Course getCourse(string id, string sec)

Description: The method to get the course that the teacher teach

Parameters: string id, string sec

Returns: Course

Method Name: addTeaches

public void addTeaches(Course course)

Description: The method to add the course to the teacher

Parameters: Course course

Returns: void

CD-12: Main

Main
+ login_info() + student_home_course(course_code) + unenrolled_home_student(user) + teacher_home_course(user, course_name, course_id, course_section, course_credit) + student_info(id, first_name, last_name, email, password, confirm_password) + teacher_info(id, first_name, last_name, email, password, confirm_password) + student_announcement(course_id, course_sec, user) + teacher_announcement(course_id, course_sec, user) + teacher_announcement_info(user, course_id, course_sec, topic, content, files) + delete_announcement(user, course_id, course_sec) + teacher_attendance_dates(course_id, course_sec, user) + teacher_attendance_dates_info(teacher, course_id, course_sec, date) + teacher_attendance_info(user, course_id, course_sec, date) + student_attendance_overview(user) + student_assignment(course_id, course_sec, user) + student_assignment_submission(user, course_id, course_sec, code, files) + completed(user, course_id, course_sec, code) + teacher_assign(user, course_id, course_sec, topic, content, files, due_date, due_time, score, weight) + display_submit(user, course_id, course_sec, code) + set_score(user, course_id, course_sec, code, id, score, comment) + delete_assignment(user, course_id, course_sec) + delete_assignment(user, course_id, course_sec, code) + student_grade_overview(user) + student_grade(course_id, course_sec, user) + teacher_grade_by_assignment(course_id, course_sec, user) + save(user, course_id, course_sec) + teacher_grade_assignment(course_id, course_sec, user, code) + teacher_grade_overview(user) + profile(user) + logout()

Method Name: login_info

```
public void login_info()
```

Description: The method is for receiving a user's email address and password, validating the credentials and displaying error messages for invalid credentials and redirecting users with valid credentials to appropriate pages.

Returns: void

Method Name: student_home_course

```
public void student_home_course(course_code)
```

Description: The method is for student to enroll to a course

Parameters: course_code

Returns: void

Method Name: unenrolled_home_student

```
public void unenrolled_home_course(user)
```

Description: The method unenroll the student user from a course

Parameters: user

Returns: void

Method Name: teacher_home_course

```
public void teacher_home_course(user, course_name,  
course_id, course_section, course_credit)
```

Description: The method is for a teacher to create a new course using the inputted course's identification number, amount of credit, course's section, validating those fields and displaying error messages for invalid fields. For valid fields, the method uses input fields to find existing courses and reject attempts to create new courses. For unique fields, the method creates the new course and redirects users to the teacher's home page.

Parameters: user, course_name, course_id, course_section, course_credit

Returns: void

Method Name: student_info

```
public void student_info(id, first_name, last_name, email,  
password, confirm_password)
```

Description: The function processes student registration by validating input data, checking for errors, and saving valid information. If all validations pass, a new Student object is created.

Parameters: id, first_name, last_name, email, password, confirm_password

Returns: void

Method Name: teacher_info

```
public void teacher_info(id, first_name, last_name, email,  
password, confirm_password)
```

Description: The function validates and registers teacher data. If the email is already registered, it notifies the user. Otherwise, it creates a Teacher object and saves it to the database.

Parameters: id, first_name, last_name, email, password, confirm_password

Returns: void

Method Name: student_announcement

```
public void student_announcement(course_id, course_sec,  
user)
```

Description: This function display announcements for a specific course and section

Parameters: course_id, course_sec, user

Returns: void

Method Name: teacher_announcement

```
public void teacher_announcement(course_id, course_sec,  
user)
```

Description: This function allows a teacher to view or manage announcements for a specific course and section.

Parameters: user, course_id, course_sec, topic, content, files

Returns: void

Method Name: teacher_announcement_info
public void teacher_announcement_info(user, course_id,
course_sec, topic, content, files)

Description: This function allows teachers to create announcements for a specific course and section. A timestamp is generated for each announcement, which is then linked to the course and its students.

Parameters: user, course_id, course_sec, topic, content, files

Returns: void

Method Name: delete_announcement
public void delete_announcement(user, course_id,
course_sec)

Description: The function enables teachers to delete specific announcements for a course and section.

Parameters: user, course_id, course_sec

Returns: void

Method Name: teacher_attendance_dates
public void teacher_attendance_dates(course_id,
course_sec, user)

Description: This function lists all the class dates for a specific course and section, allowing a teacher to check attendance records.

Parameters: course_id, course_sec, user

Returns: void

Method Name: teacher_attendance_dates_info
public void teacher_attendance_dates_info(user,
course_id, course_sec, topic, content, files)

Description: The function creates an attendance date by retrieving the current date, time, using inputted course's identification number and section to get the teacher's course, and get the previously retrieved course's attendance then checking if this date and time attendance already exists, rejecting the attempt to create the attendance date. If the attendance date is successfully created, the function adds the attendance to each student's attendance list then returns the teacher's attendance page.

Parameters: user, course_id, course_sec, topic, content, files

Returns: void

Method Name: teacher_attendance_info

```
public void teacher_attendance_info(user, course_id,  
course_sec, date)
```

Description: function allows teachers to record attendance for students in a specific course and section on a given date. It updates each student's attendance status (present, absent, or late) and any remarks provided. The function also calculates and updates the total number of students present, absent, and late for that day.

Parameters: user, course_id, course_sec, date

Returns: void

Method Name: student_attendance_overview

```
public void student_attendance_overview(user)
```

Description: This function displays an attendance overview page for students.

Parameters: user

Returns: void

Method Name: student_assignment

```
public void student_assignment(course_id, course_sec,  
user)
```

Description: This function provides a student's view of assignments for a specific course and section.

Parameters: course_id, course_sec, user

Returns: void

Method Name: student_assignment_submission

```
public void student_assignment_submission(user,  
course_id, course_sec, code, files)
```

Description: This method allows students to submit the assignment in a specific course and section with a timestamp.

Parameters: user, course_id, course_sec, code, files

Returns: void

Method Name: completed

```
public void completed(user, course_id, course_sec, code)
```

Description: This function provides information about a specific submitted assignment for a student.

Parameters: user, course_id, course_sec, code

Returns: void

Method Name: teacher_assign

```
public void teacher_assign(user, course_id, course_sec,  
topic, content, files, due_date, due_time, score, weight)
```

Description: The function allows a teacher to create and assign a new assignment.

Parameters: user, course_id, course_sec, topic, content, files, due_date, due_time, score, weight

Returns: void

Method Name: display_submit

```
public void display_submit(user, course_id, course_sec,  
code)
```

Description: This function provides a page for teachers to check the status of student submissions for a specific assignment in a course.

Parameters: user, course_id, course_sec, code

Returns: void

Method Name: set_score

```
public void set_score(user, course_id, course_sec, code,  
id, score, comment)
```

Description: The function allows a teacher to set the score and optional comments for a student's assignment.

Parameters: user, course_id, course_sec, code, id, score, comment

Returns: void

Method Name: delete_assignment

```
public void delete_assignment(user, course_id,  
course_sec)
```

Description: The function allows the teacher to delete an assignment.

Parameters: user, course_id, course_sec

Returns: void

Method Name: delete_assignment

```
public void delete_assignment(user, course_id,  
course_sec, code)
```

Description: This function allows students to unsubmit their assignment.

Parameters: user, course_id, course_sec, code

Returns: void

Method Name: student_grade_overview

```
public void student_grade_overview(user)
```

Description: This function provides a grade overview page for students.

Parameters: user

Returns: void

Method Name: student_grade

```
public void student_grade(course_id, course_sec, user)
```

Description: This function provides score information for a student in a specific course. It calculates and displays statistics (mean, minimum, maximum) for each assignment, along with the student's score and the assignment's weight.

Parameters: course_id, course_sec, user

Returns: void

Method Name: teacher_grade_by_assignment

```
public void teacher_grade_by_assignment(course_id,  
course_sec, user)
```

Description: This function allows teacher to overview students' grades of each course and scores of each assignment

Parameters: course_id, course_sec, user

Returns: void

Method Name: save

```
public void save(user course_id, course_sec)
```

Description: The function allows a teacher to manually set and save grades for students in a specific course and section.

Parameters: u

Returns: void

Method Name: teacher_grade_assignment

```
public void teacher_grade_assignment(course_id,  
course_sec, user, code)
```

Description: This function allows teacher to check scores of students in the assignment

Parameters: course_id, course_sec, user, code

Returns: void

Method Name: teacher_grade_overview

```
public void teacher_grade_overview(user)
```

Description: This function displays an overview page for teachers to review the grading statistics of the courses they teach. The statistics include the minimum, maximum, and mean scores of the students in each course.

Parameters: user

Returns: void

Method Name: profile

public void profile(user)

Description: This function allows users to view their profile.

Parameters: user

Returns: void

Method Name: logout

public void logout()

Description: This function allows users to logout.

Returns: void

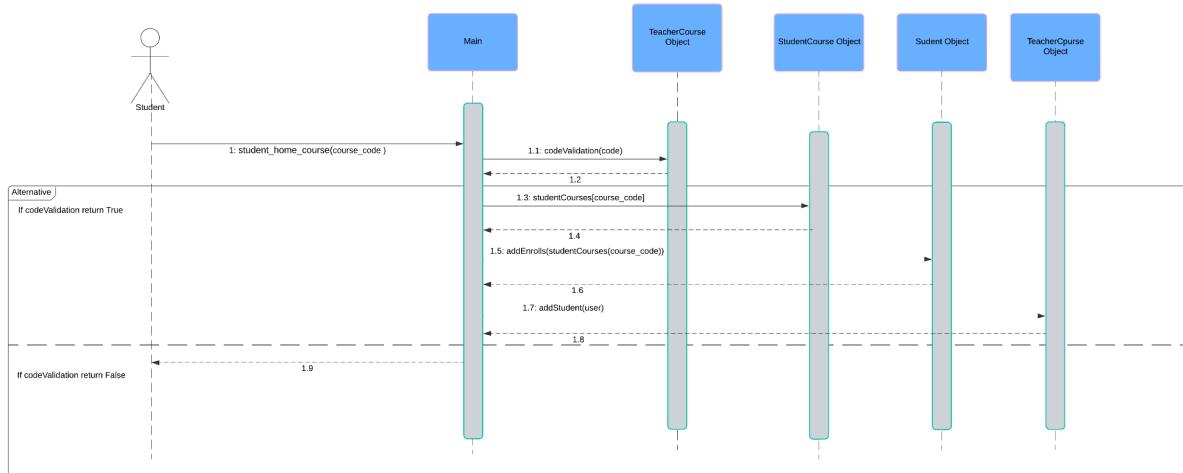
Sequence Diagram

SD-1: Teachers can create a course



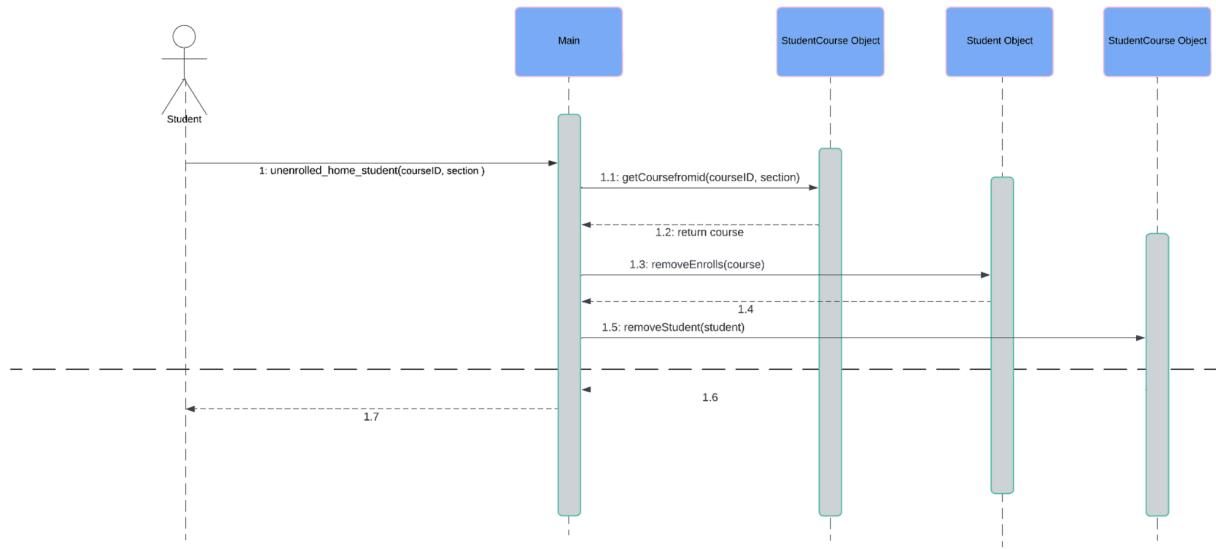
Teacher makes a POST method request to the Main. The main will validate the course information. Then, it will add the announcement into the TeacherCourse and Teacher object if the courseValidation returns true. If the courseValidation returns false, the Main will send the validation result to the user.

SD-2: Students can enroll in courses



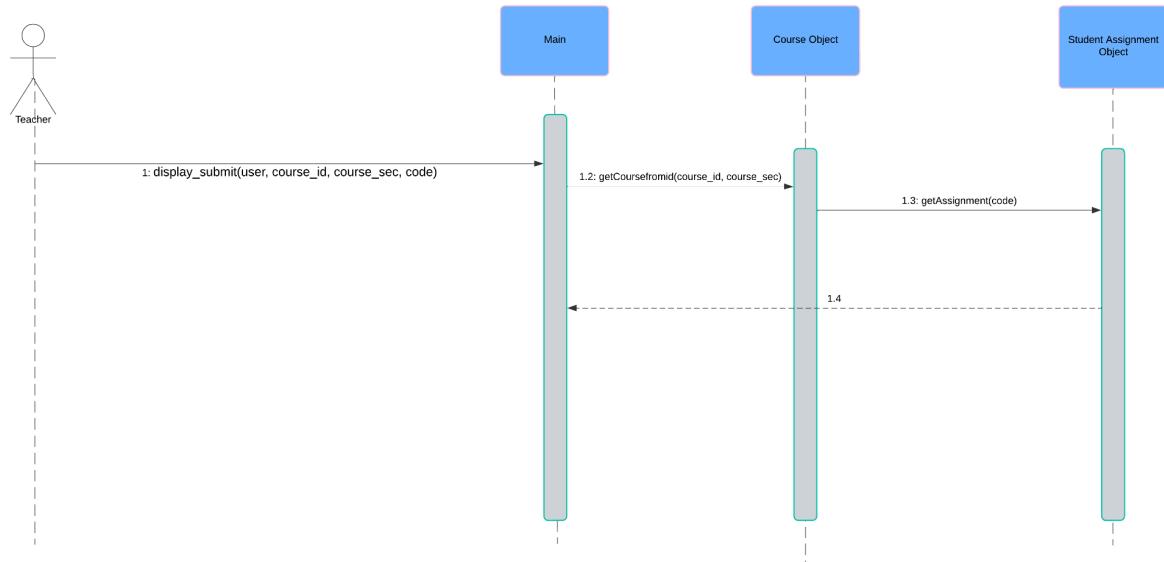
Student makes a POST method request to the Main. The main will validate the course information. If the codeValidation returns true, it will enroll the student into the StudentCourse, Student and TeacherCourse Object. If the codeValidation returns false, the Main will send the validation result to the user.

SD-3: Students can unenroll course



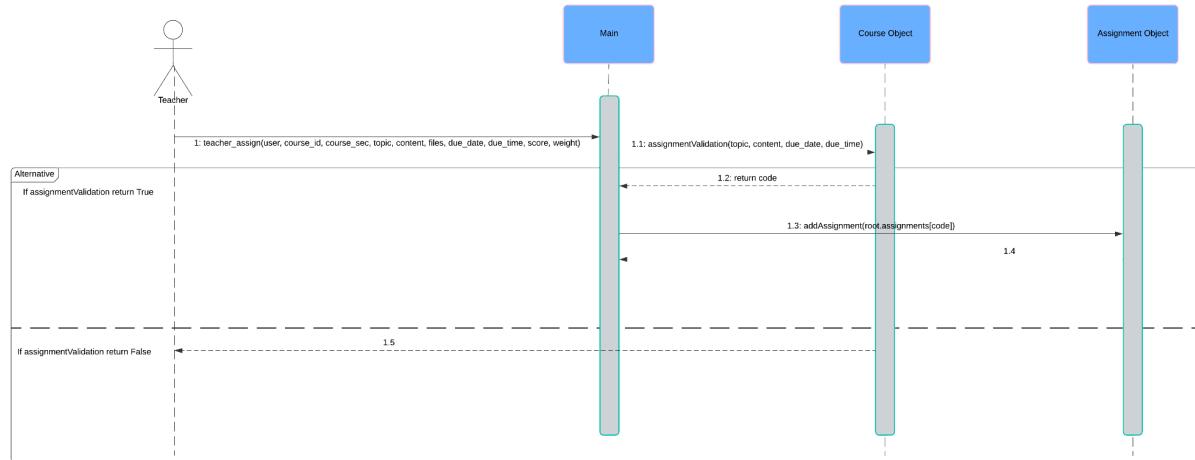
Student makes a POST method request to the Main. The main will get the course information and remove the student from Student and StudentCourse objects.

SD-4: Teachers can view assignments



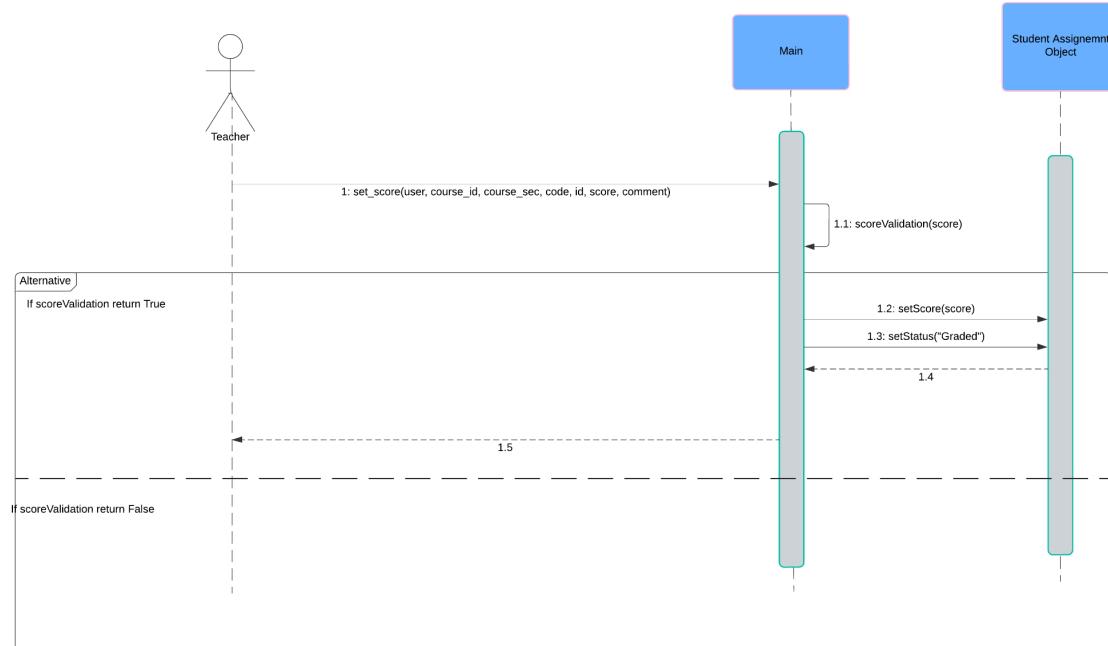
Teacher makes a GET method request to the Main. The main will get the course information and get the assignments from the StudentAssignment object.

SD-5: Teachers can create assignments



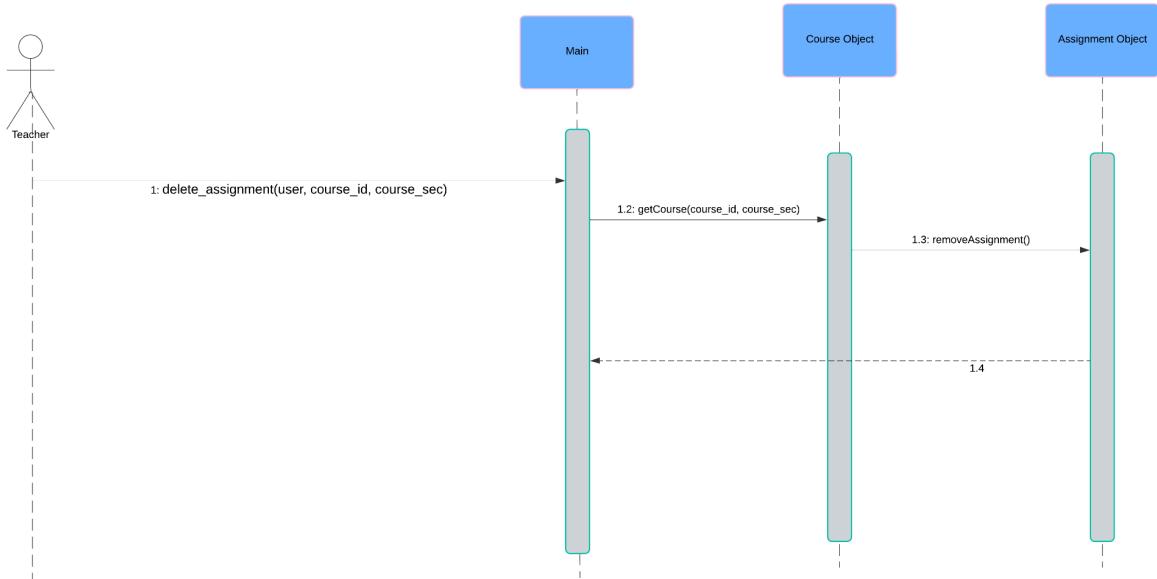
Teacher makes a POST method request to the Main. The main will validate the assignment information. Then, it will add the assignment into the Course and Assignment object if the assignmentValidation returns true. If the assignmentValidation returns false, the Main will send the validation result to the user.

SD-6: Teachers can grade assignments



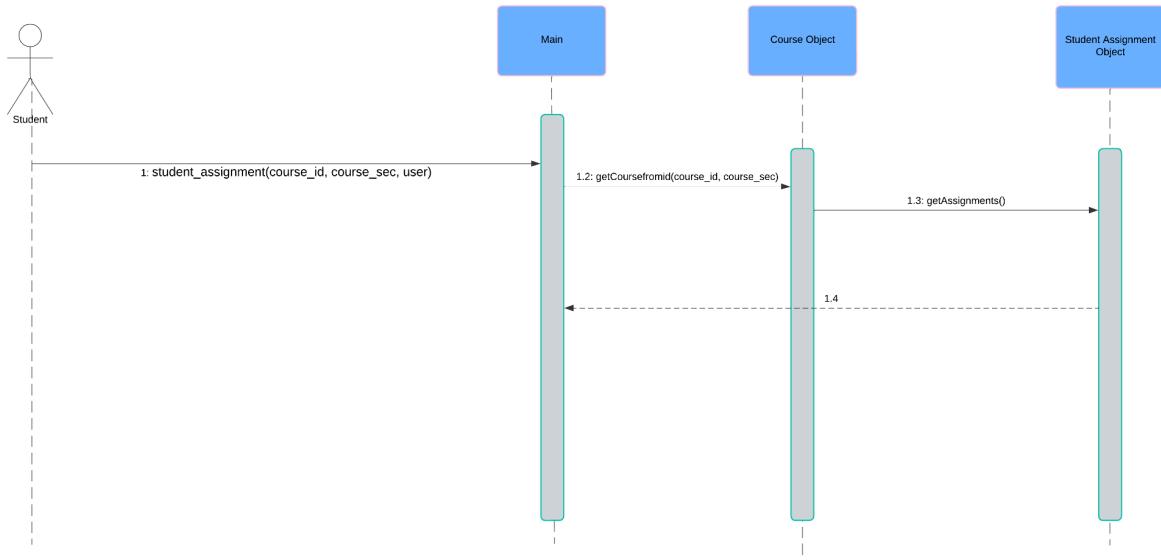
Teacher makes a POST method request to the Main. The main will validate the score information. Then, it will set the score and the status into the StudentAssignment object if the scoreValidation returns true. If the scoreValidation returns false, the Main will send the validation result to the user.

SD-7: Teachers can delete assignments



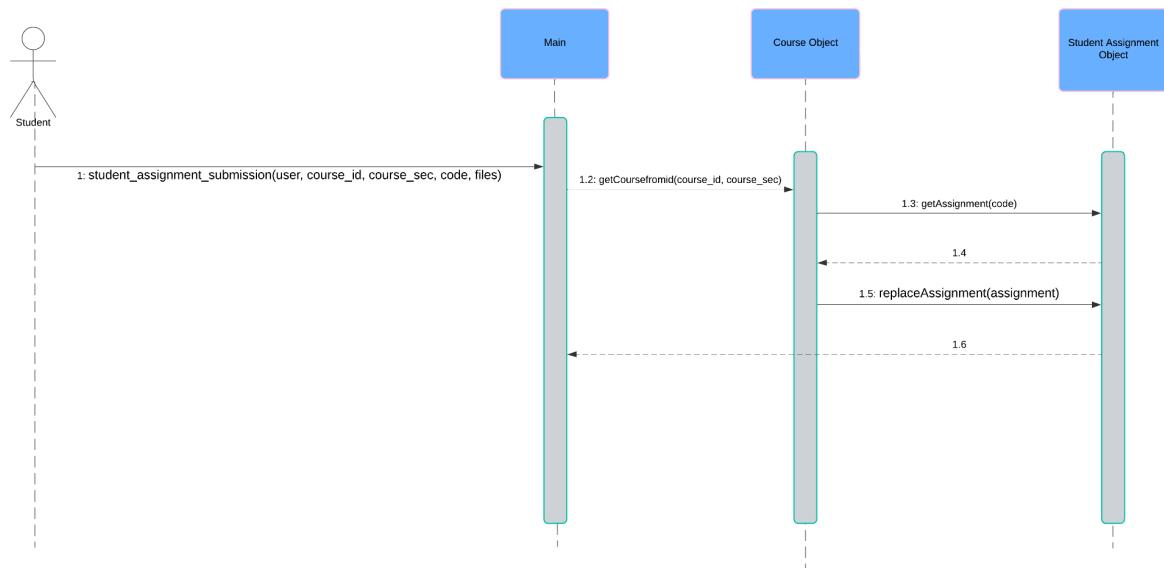
Teacher makes a POST method request to the Main. The main will get the course information and remove the assignment from the Assignment object.

SD-8: Students can view assignments



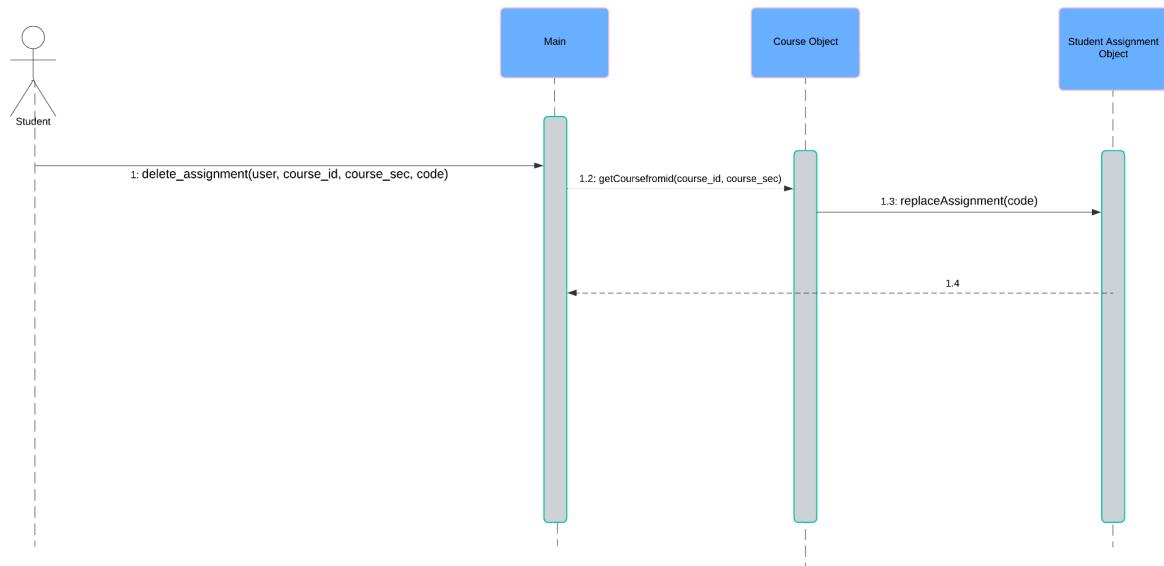
Student makes a GET method request to the Main. The main will get the course information and get the assignments from the StudentAssignment object.

SD-9: Students can submit assignments



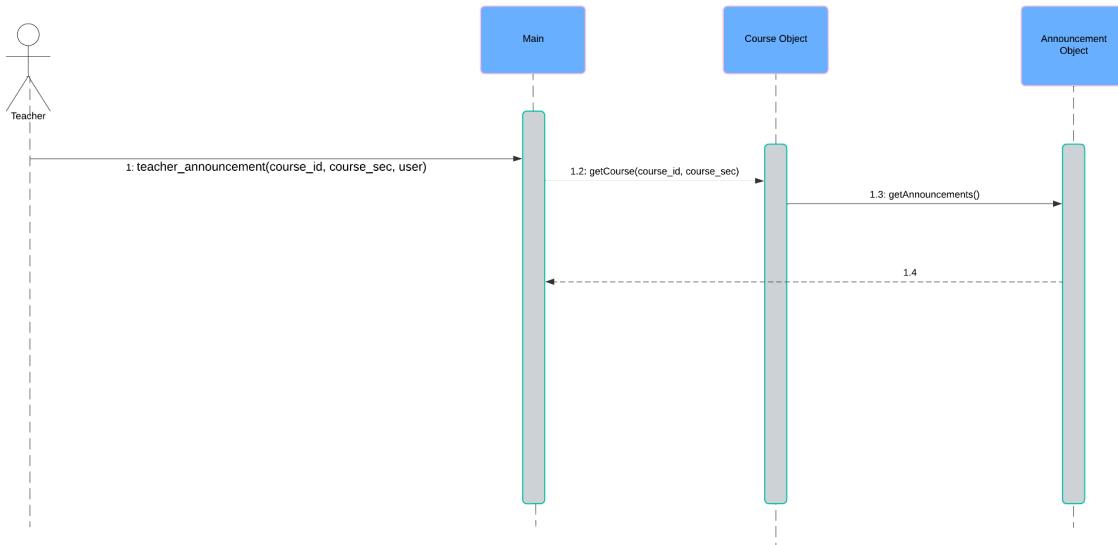
Student makes a POST method request to the Main. The main will get the course information and get the assignment code and submit the assignment to StudentAssignment object.

SD-10: Students can unsubmit assignments



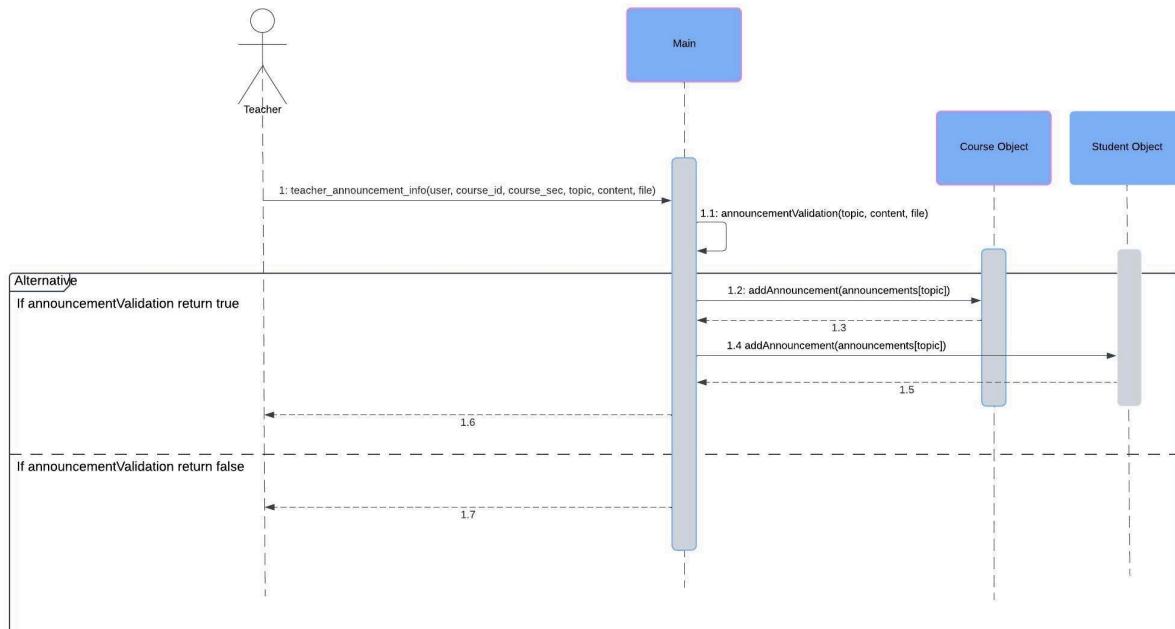
Student makes a POST method request to the Main. The main will get the course information and unsubmit the assignment from the StudentAssignment object.

SD-11: Teachers can view announcement posts



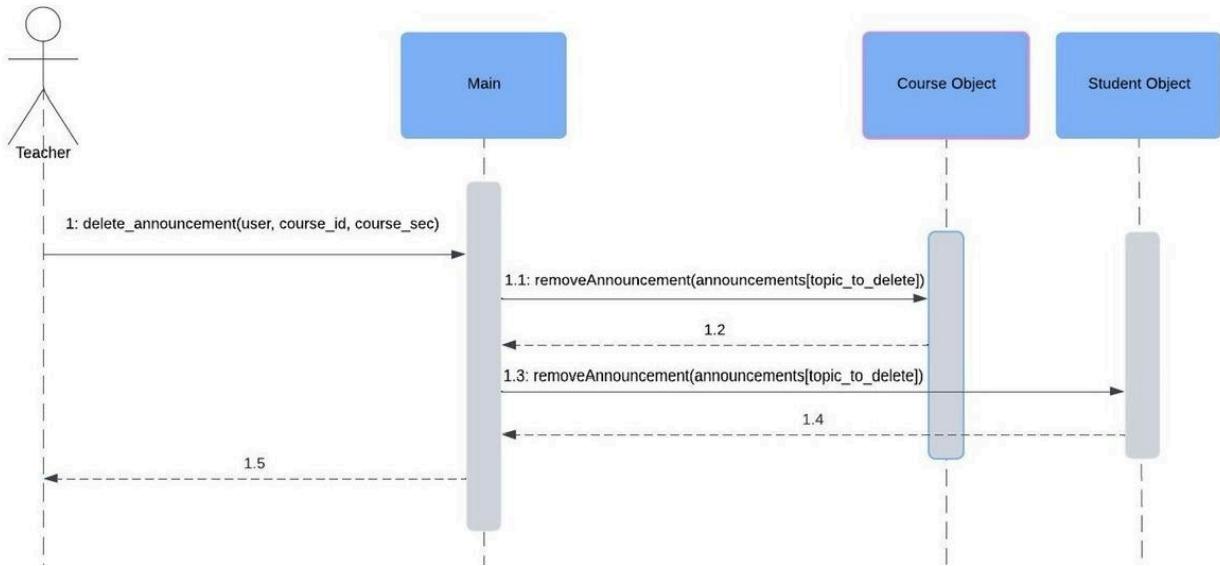
Teacher makes a GET method request to the Main. The main will get the course information and get the announcements from the Announcement object.

SD-12: Teachers can create an announcement post



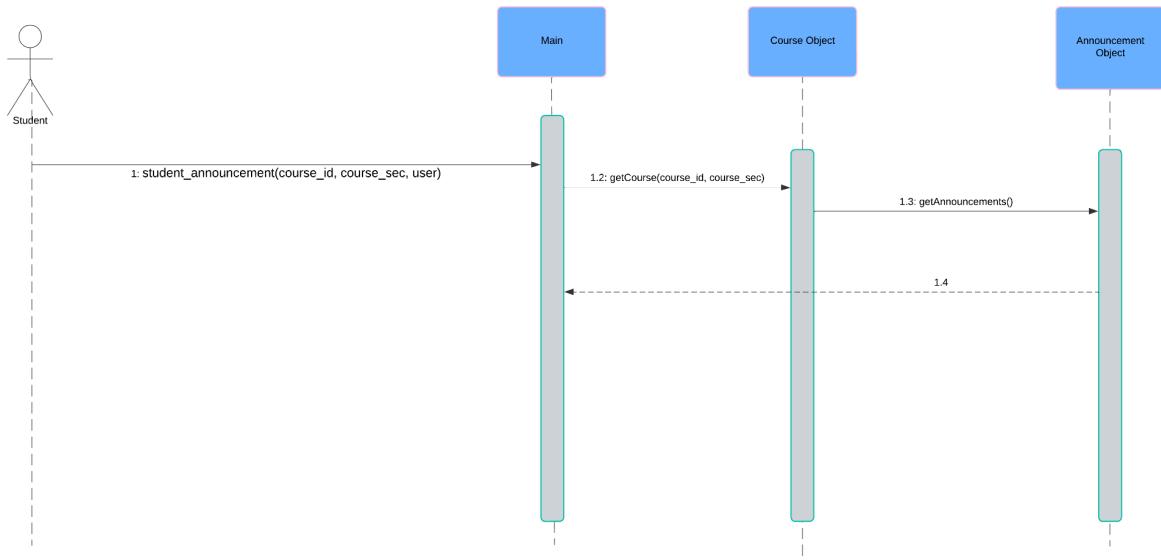
Teacher makes a POST method request to the Main. The Main will validate the announcement information. Then, it will add the announcement into the Course and Student object if the announcementValidation returns true. If the announcementValidation returns false, the Main will send the validation result to the user.

SD-13: Teacher can delete announcement post



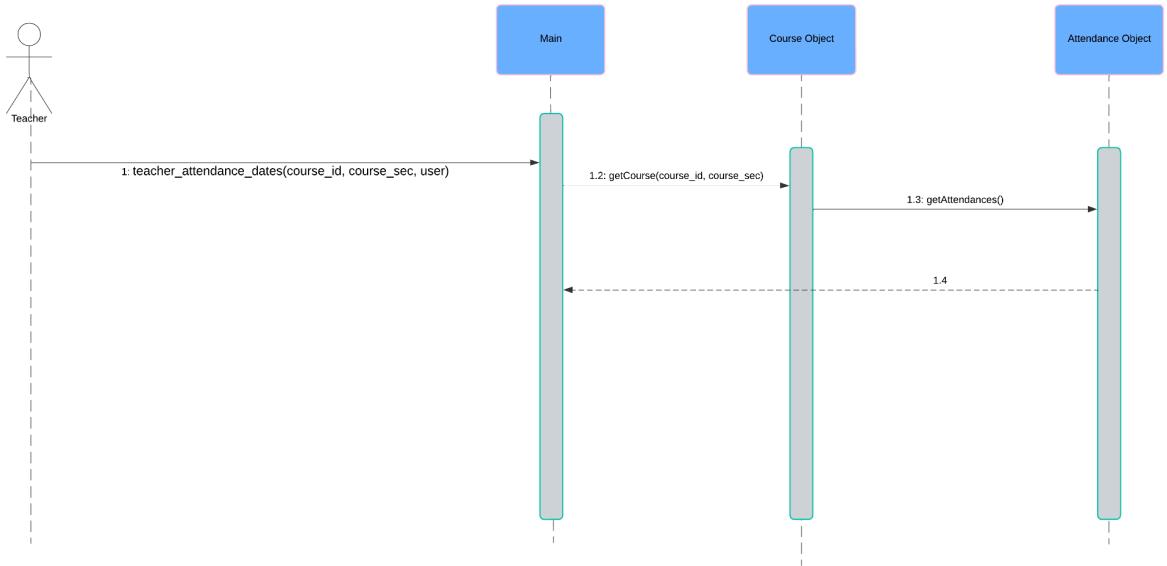
Teacher makes a POST method request that will be sent to the Main. After that, the Main will call removeAnnouncement to delete the announcement post from the Course and Student object.

SD-14: Students can view announcement posts



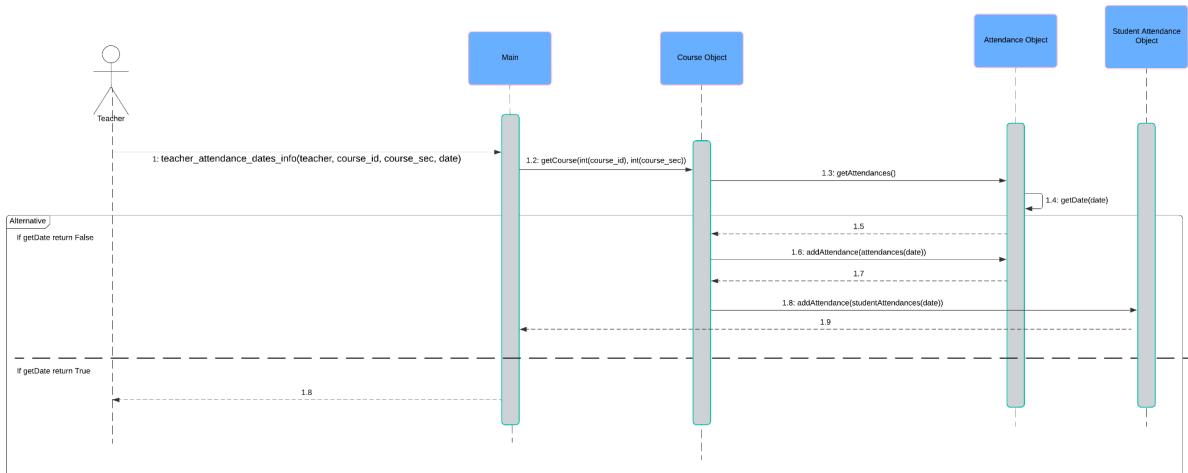
Student makes a GET method request to the Main. The main will get the course information and get the announcements from the Announcement object.

SD-15: Teachers can view attendance



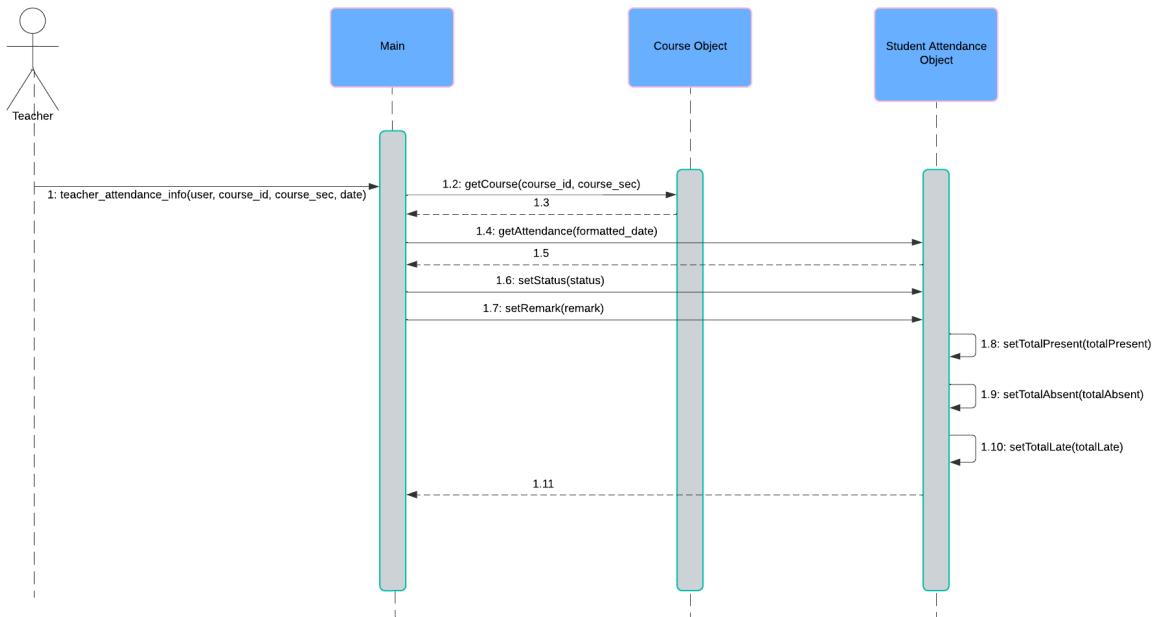
Teacher makes a GET method request to the Main. The main will get the course information and get the attendances from the Attendance object.

SD-16: Teachers can create attendance



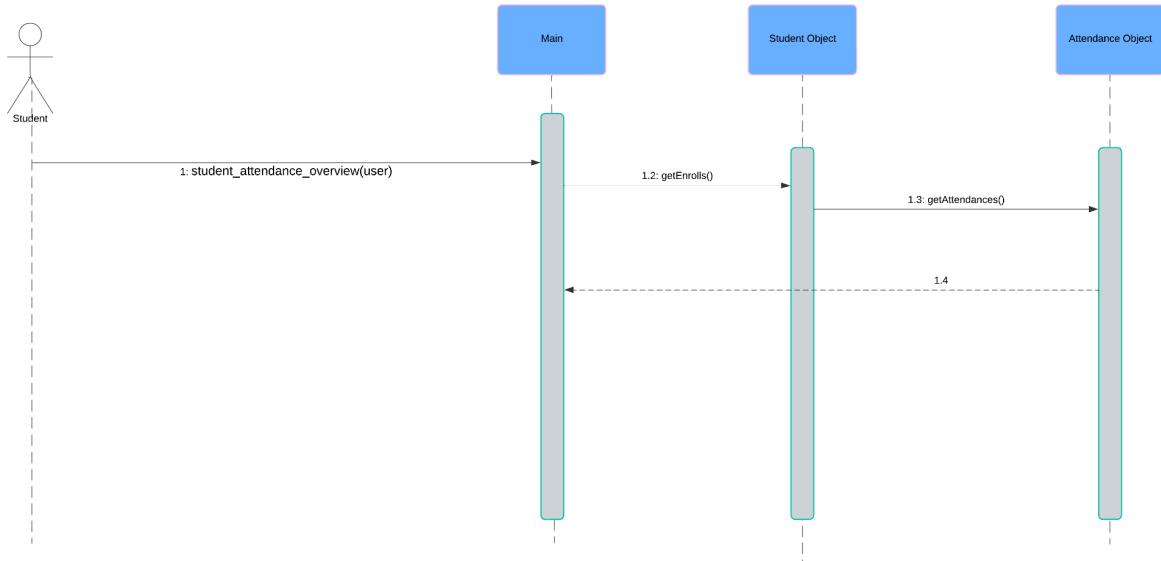
Teacher makes a POST method request to the Main. The Main will validate the attendance information. Then, it will add the attendance into the Attendance and StudentAttendance object if the attendanceValidation returns true. If the attendanceValidation returns false, the Main will send the validation result to the user.

SD-17: Teachers can check attendance



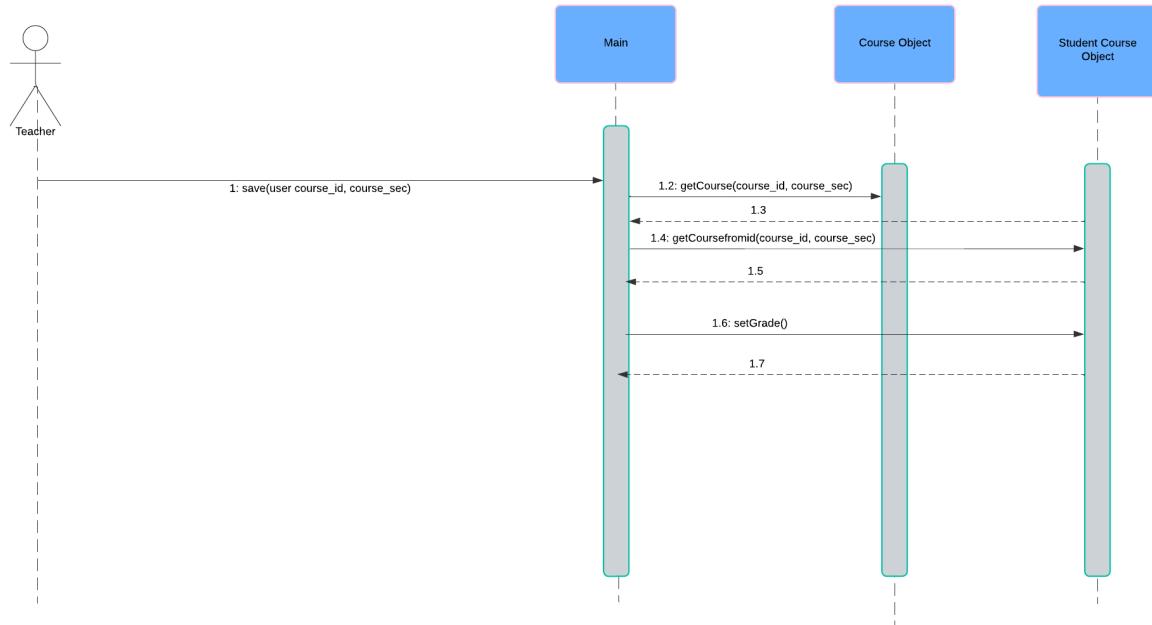
Teacher makes a POST method request to the Main. The Main will get course from Course object. Then, it will get the attendance and set the status and remark to the StudentAttendance object.

SD-18: Students can view attendance



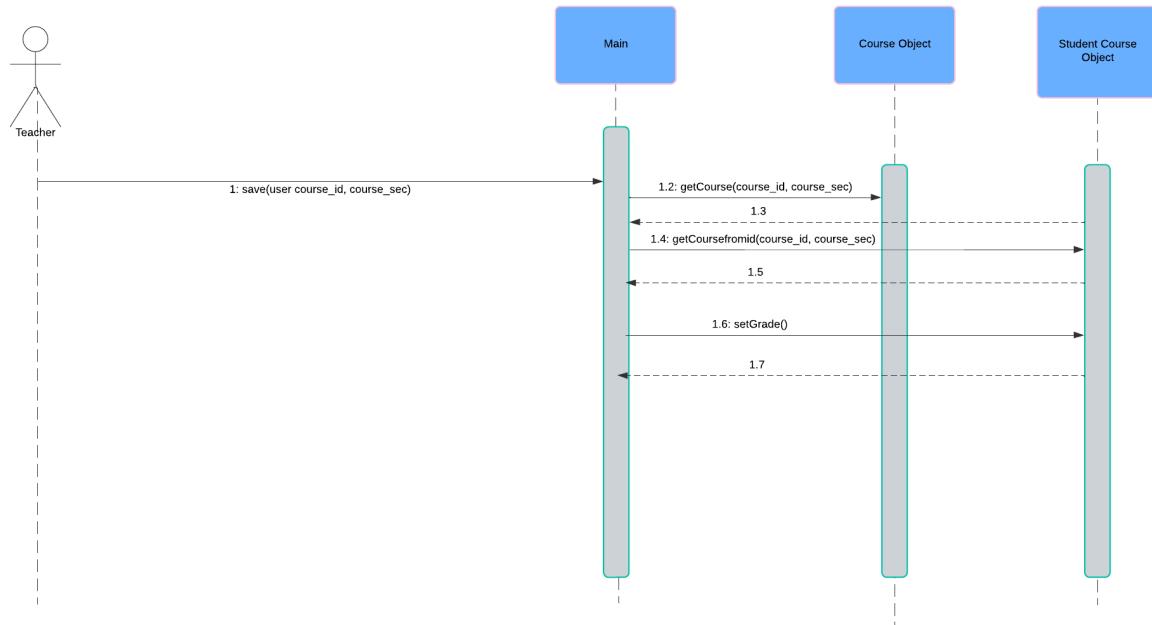
Student makes a GET method request to the Main. The Main will get the student enrollments. Then, it will get the attendance from the Attendance object.

SD-19: Teachers can assign grades



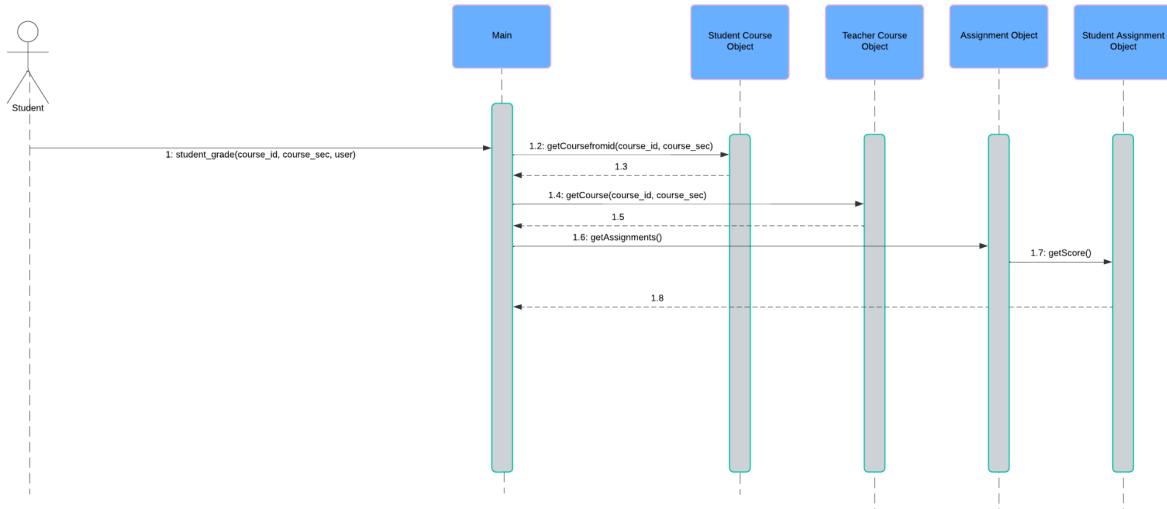
Teacher makes a POST method request to the Main. The Main will get courses from Course and StudentCourse objects. Then, it will set the grade for the student in the course.

SD-20: Teachers can edit grades



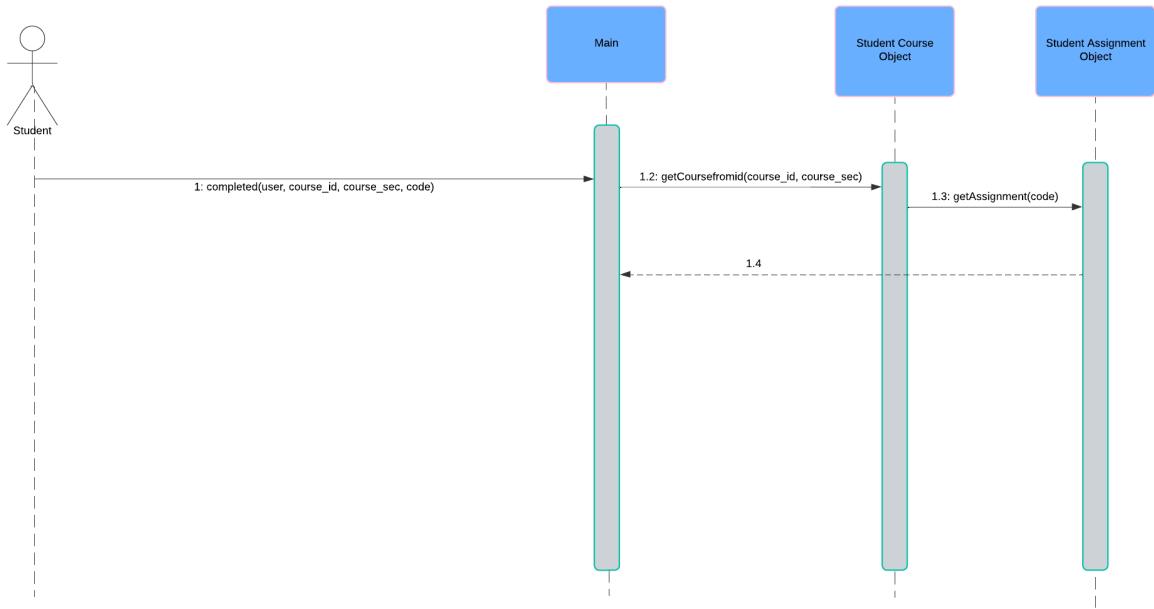
Teacher makes a POST method request to the Main. The Main will get courses from Course and StudentCourse objects. Then, it will set the edited grade for the student in the course.

SD-21: Students can view grades for the assignments in a specific course



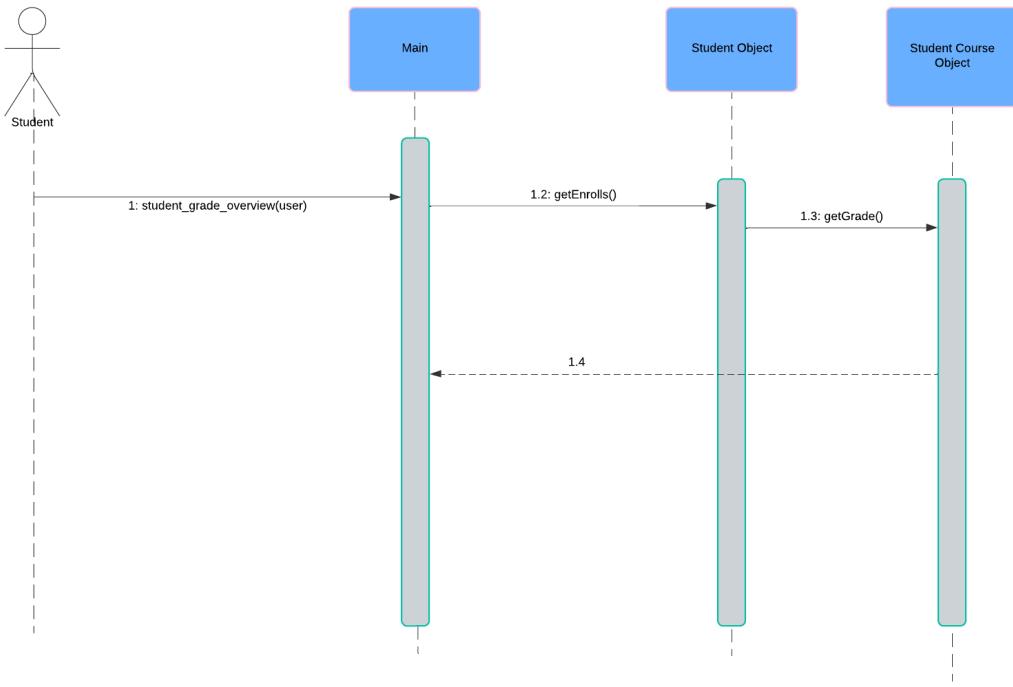
Student makes a GET method request to the Main. The Main will get courses from Course and StudentCourse objects. Then, it will get the assignments from the Assignment object. After that it will get the scores of the assignments from the StudentAssignment object.

SD-22: Students can view grades for a specific assignment in a specific course



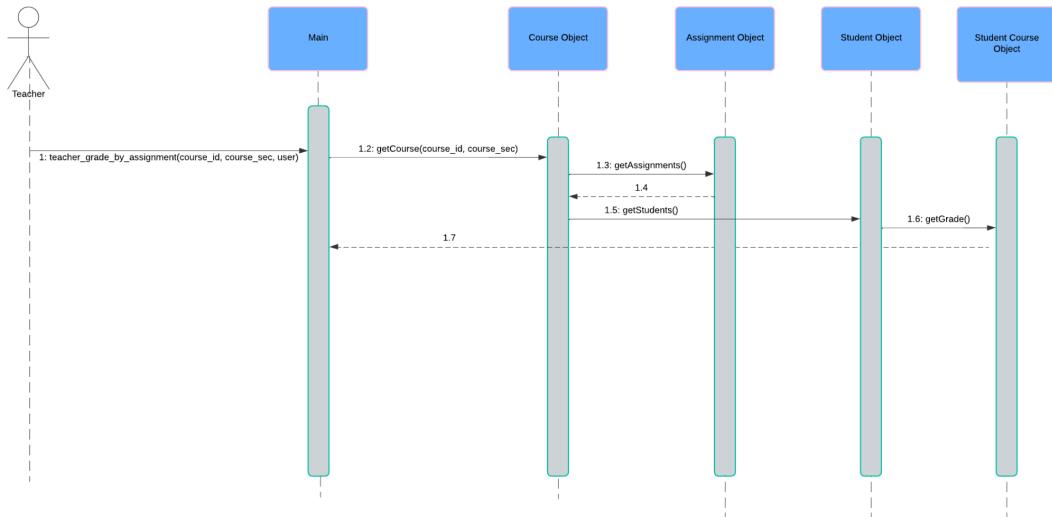
Student makes a GET method request to the Main. The Main will get a course from the StudentCourse object. Then, it will get the assignment details from the StudentAssignment object.

SD-23: Students can view all the grades of all the enrolled courses



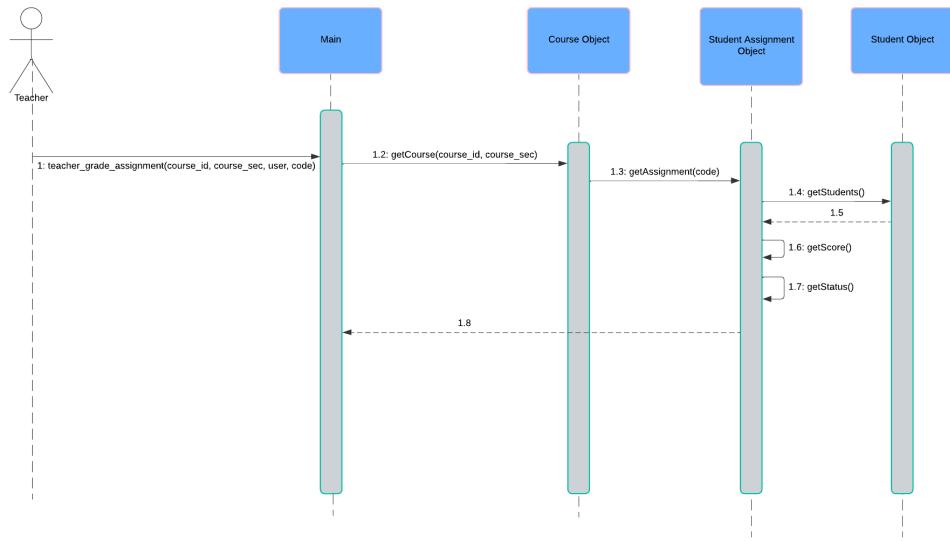
Student makes a GET method request to the Main. The Main will get the student enrollments. Then, it will get the grade details from the StudentCourse object.

SD-24: Teachers can view all the students' grades in a specific course



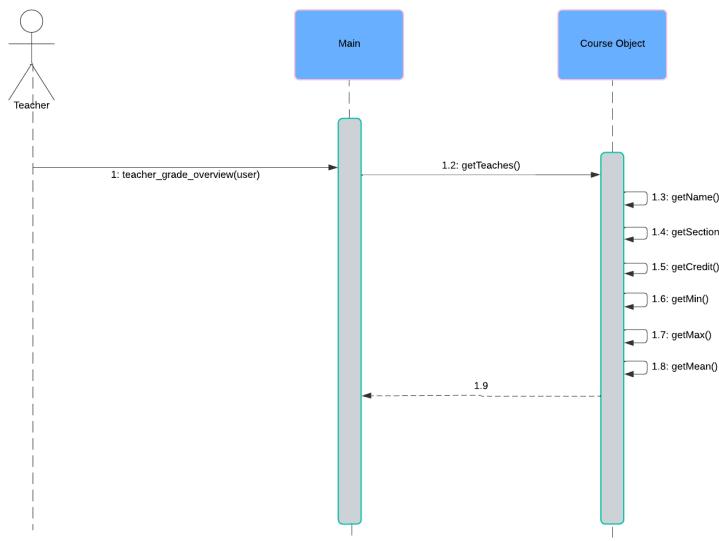
Teacher makes a GET method request to the Main. The Main will get the course from the Course object. Then, it will get the assignments and students from Assignment and Student objects. After that it will get the grade details from the StudentCourse object.

SD-25: Teachers can view all the students' grades in a specific assignment



Teacher makes a GET method request to the Main. The Main will get the course from the Course object. Then, it will get the assignment and students from StudentAssignment and Student objects. After that it will get the scores and status of each student from the StudentAssignment object.

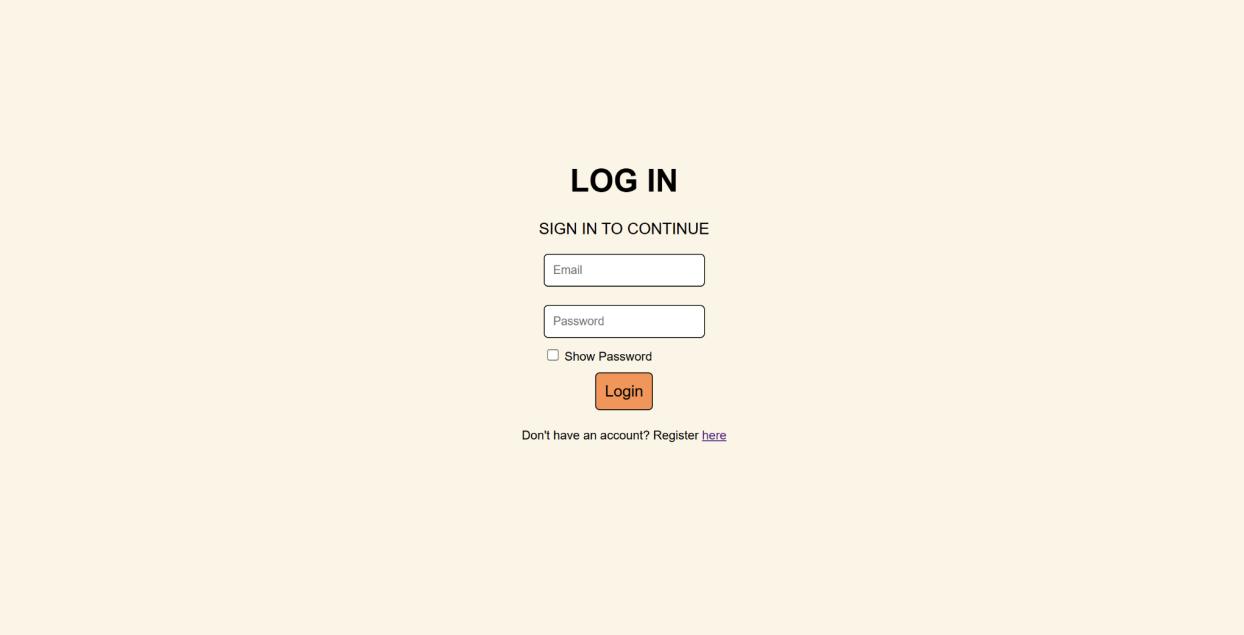
SD-26: Teachers can view all the classes' grades for all the courses they are teaching



Teacher makes a GET method request to the Main. The Main will get the details of the course that the teacher teaches from the Course object.

User Interface Design

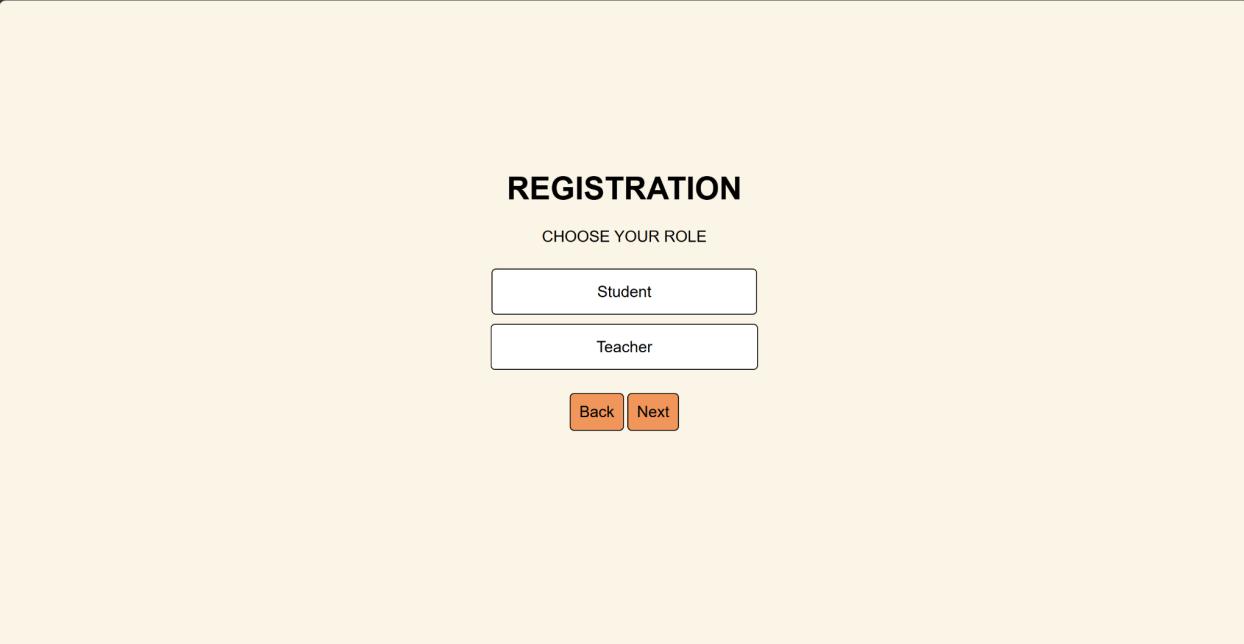
Login Interface



The login interface features a central "LOG IN" heading. Below it, a "SIGN IN TO CONTINUE" instruction is displayed. There are two input fields: one for "Email" and one for "Password". A "Show Password" checkbox is located next to the password field. A prominent orange "Login" button is centered below the fields. At the bottom, a link text "Don't have an account? Register [here](#)" is shown.

Teachers and students log in to their account

Register Interface



The registration interface has a central "REGISTRATION" heading. Below it, a "CHOOSE YOUR ROLE" instruction is displayed. Two rectangular buttons are shown: "Student" and "Teacher". At the bottom, there are two small orange buttons labeled "Back" and "Next".

User selects whether they're a student or a teacher

CREATE AN ACCOUNT

Firstname:

Lastname:

Student ID:

Email:

Password:

Show Password

Confirm Password:

Show Password

[Back](#) [Register](#)

User enters account information

Student's Courses List Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

[Join Course](#)

Linear Algebra

12234456 (Section:2)

[Unenroll](#)

GRADE ASSIGNMENT ANNOUNCEMENT

Web Programming

1286233 (Section:901)

[Unenroll](#)

Students view the courses they're enrolled in



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Linear Algebra
12234456 (Section:2)

Enter Course Code:
XXXX
Cancel Join

Join Course

GRADE ASSIGNMENT ANNOUNCEMENT

Web Programming
1286233 (Section:901)

Unenroll

Unenroll

Students enroll in a new course

Student's Course Announcements Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Announcement
Web Programming (1286233 Section: 901)

HTTP (16/11/2023 16:15:1)

Make sure to download the file
[http.pdf](#)

Lab resources (16/11/2023 16:16:34) +

Students view announcements of a course

Student's Course Assignments Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Assignment

Web Programming (1286233 Section: 901)

Forthcoming +

Past Due +

Graded -

HW1 (Due Date: 2023-11-23 23:59 Score: 10)
[View Assignment](#)

Completed -

Students view assignments of a course



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Assignment

Web Programming (1286233 Section: 901)

HW1 Score: 10 Due Date: 2023-11-23 23:59

Write a website by using HTML

Submitted Date: 16/11/2023
Submitted Time: 16:3
[transcript.html](#)

Score:
10

Comment:

[Unsubmit](#) [Code Preview](#)

Students view an assignment's details and submit an assignment

Student's Grade Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmitl.ac.th

Grade

(1286233 Section: 901)

Details	Weight(%)	Total Score	Mean	Min	Max	Score
HW1	5	10	8.75	5	10	10
mini project	10	10	5	0	10	0
HW3	5	10	0	10	10	0
HW2	5	10	4.5	0	9	0

Students view grade details of a specific course



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmitl.ac.th

Grade Overview

Course ID	Courses	Credits	Current Scores	Grades
1286233	Web Programming	4	5	B
11345287	Statistic and Probability	3	5	
11345289	Database System	3	0	
12234456	Linear Algebra	3	0	
12345670	Professional Communication and Presentation	3	0	
12345678	what	3	0	A

1

Grade range

Students view overall grades of all courses

Student's Attendance Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Attendance Overview

Course ID	Course Name	Present	Late	Absent
1286233	Web Programming (Section: 901)	3	0	1
11345287	Statistic and Probability (Section: 901)	1	0	0
11345289	Database System (Section: 122)	0	0	0
12234456	Linear Algebra (Section: 2)	0	0	0
12345670	Professional Communication and Presentation (Section: 901)	4	0	1
12345678	what (Section: 1223)	1	0	0

Students view overall attendance of all courses

Student's Profile Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW 65011587@kmit.ac.th

Profile

First Name: Thanida Paige
Last Name: Pholsukcharoen
ID: 65011587
Email: 65011587@kmit.ac.th

[Logout](#)

Student view their profile

Student's Course Details Interface



COURSES GRADE OVERVIEW ATTENDANCE OVERVIEW [65011587@kmitl.ac.th](#)

Web Programming

Course Info +

- Announcement
- Attendance
- Assignment
- Grade

Students view course details

Teacher's Courses List Interface



COURSES GRADE OVERVIEW [teacher12@kmitl.ac.th](#)

Create Course

what

12345678 (Section: 1223)

ATTENDANCE GRADE ASSIGNMENT ANNOUNCEMENT

Teachers view the courses they're responsible for



COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

what
12345678 (Section: 1223)

Create Course

Course Name:

Course ID:

Course Section:

Course Credit:

ATTENDANCE GRADE ASSIGNMENT ANNOUNCEMENT

Teachers create a new course

Teacher's Course Announcement Interface



COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

Announcement
what (12345678 Section: 1223)

test (16/12/2024 13:44:52)	-
test	
<input type="button" value="Delete"/>	

Teachers view announcements of a course



COURSES GRADE OVERVIEW teacher12@kmutt.ac.th

Add Announcement

Topic:

Description:

No file chosen

test (16/12/2024 13:44:52)

test

Teachers create a new announcement for a course

Teacher's Course Assignment Interface

SEmester

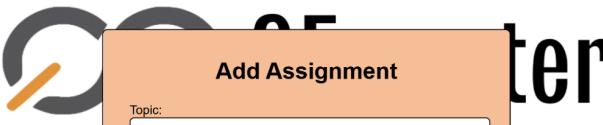
COURSES GRADE OVERVIEW teacher12@kmutt.ac.th

Assignment
what (12345678 Section: 1223)

test (Due Date: 2003-11-11 11:11 Score: 10)

test

Teachers view assignments of a course



COURSES GRADE OVERVIEW

teacher12@kmit.ac.th

test (Due Date: 2003-11-11 11:11 Score: 10)	-
test	
Delete Check	

Add Assignment

Topic:

Description:

Choose Files No file chosen

Due Date: mm / dd / yyyy [\[calender\]](#)

Due Time: -- : -- : -- [\[clock\]](#)

Score:

Weight(out of 100%):

[Post](#) [Cancel](#)

[Assign](#)

Teachers create a new assignment for a course



COURSES GRADE OVERVIEW

teacher12@kmit.ac.th

	Assignment: test what (12345678 Section: 1223)
Not Turned in	+
Turned in	+
Late	-
Thanida Paige Pholsukcharoen ID: 65011587 Check	
Graded	+

Teachers view list of student's submission status of an assignment

COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

Assignment: test
what (12345678 Section: 1223)

test

Name: Thanida Paige Pholsukcharoen
ID: 65011587
Submitted Date: 9/12/2024 13:56
Status: Late
[q1.py](#)

Score: 0 Out of 10
Comment:

Teachers view a student's submission and grade the assignment

Teacher's Course Attendance Record Interface

COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

Attendance
what (12345678 Section: 1223)

07/10/2004 [Create Attendance](#) [Remove](#)

Teachers view a list of dates for course attendance record



COURSES GRADE OVERVIEW teacher12@kmit.ac.th

Create Attendance

Select date:

Create Cancel

07/10/2004 Remove

Teachers create a new date for the attendance record



COURSES GRADE OVERVIEW teacher12@kmit.ac.th

⟳

Attendance

what (12345678 Section: 1223)

Student ID	Name	Status	Remarks
65011587	Thanida Paige Pholsukcharoen	present	<input type="text"/>

Summary

Total Present: 1
Total Late: 0
Total Absent: 0

Save

1 # of Students' Attendance

Teachers view a list of students for each attendance record and check their attendance status

Teacher's Grade Interface



COURSES GRADE OVERVIEW teacher12@kmit.ac.th

Grade

what (12345678 Section: 1223)

By Grade ▾

Student ID	Names	Total Score	Grades
65011587	Thanida Paige Pholsukcharoen	0	A

[Edit](#)

Score Summary

Mean: 0.00
Standard Deviation: 0.00
Minimum: 0.00
Maximum: 0.00

Teachers view students' overall grades in a course



COURSES GRADE OVERVIEW teacher12@kmit.ac.th

[\(K\)](#)

Grade

what (12345678 Section: 1223)

test

Total Score: 10 Weight: 10%

Student ID	Names	Scores
65011587	Thanida Paige Pholsukcharoen	Not graded

Summary

Min: inf
Max: 0
Mean: 0

Teachers view students' grades on a specific assignment in a course



COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

Grade Overview

Courses	Credits	Mean	Min	Max
what (Section: 1223)	3	0	0	0

Teachers view the overall class grade of each course

Teacher's Profile Interface



COURSES GRADE OVERVIEW teacher12@kmitl.ac.th

Profile

First Name: t
Last Name: t
Email: teacher12@kmitl.ac.th

[Logout](#)

Teachers view their profile

Teacher's Course Details Interface



COURSES GRADE OVERVIEW teacher12@kmit.ac.th

what

Course Info +

- Announcement
- Attendance
- Assignment
- Grade

Set Grade Scheme The scheme is set

This screenshot shows the teacher's course details interface. At the top, there are tabs for 'COURSES' and 'GRADE OVERVIEW'. The 'GRADE OVERVIEW' tab is active, showing the email 'teacher12@kmit.ac.th'. Below this, the word 'what' is displayed in a large, bold font. A 'Course Info' section is present with a '+' button. A sidebar contains links for 'Announcement', 'Attendance', 'Assignment', and 'Grade'. A prominent orange overlay box titled 'Set Grade' allows the teacher to define grade boundaries for various grades. The 'Set Grade' box contains fields for 'A min:' (with value '100'), 'B+ min:', 'B min:', 'C+ min:', 'C min:', 'D+ min:', 'D min:', and 'F min:' (with value '0'). Each field has a corresponding 'max:' input field. At the bottom of the overlay are 'Set' and 'Cancel' buttons. Below the overlay, a message says 'The scheme is set'. The URL '127.0.0.1:8000/Teacher/Announcement/12345678/1223' is visible at the bottom left.

Teachers view course details

COURSES GRADE OVERVIEW teacher12@kmit.ac.th

Set Grade

A min:	100
B+ min:	
B min:	
C+ min:	
C min:	
D+ min:	
D min:	
F min:	0

Set Cancel

Course Info +

- Announcement
- Attendance
- Assignment
- Grade

Set Grade Scheme The scheme is set

127.0.0.1:8000/Teacher/Announcement/12345678/1223

This screenshot shows the teacher's course details interface with a focus on the 'Set Grade Scheme' dialog box. The dialog box is titled 'Set Grade' and contains fields for defining grade boundaries. It includes fields for 'A min:' (100), 'B+ min:', 'B min:', 'C+ min:', 'C min:', 'D+ min:', 'D min:', and 'F min:' (0). Each field has a corresponding 'max:' input field. At the bottom of the dialog are 'Set' and 'Cancel' buttons. The background shows the course details interface with tabs for 'COURSES' and 'GRADE OVERVIEW', and a message 'The scheme is set'. The URL '127.0.0.1:8000/Teacher/Announcement/12345678/1223' is visible at the bottom left.

Teachers set the course's grading schema