جامعة الأهرام الكندية
AHRAM CANADIAN UNIVERSITY

Faculty of Computer Science and Information Technology

E-Learning

Phase 1: Software Engineering Project

Submitted To

Dr. Reham Adel

TA. Mai Mohamed

Submitted By

Student ID	Student Name	Section Number
42110274	Sama Ahmed	B1
42110167	Shimaa Naser	B1
42110196	Sohila Emam	B1
42110343	Rahma Mahmoud	B1
42110227	Yara Asuity	B1
42110122	Waleed Rashad	B1

Contact Email: 42110274.sama@acu.edu.eg

1. Introduction

- Introducing our innovative e-learning Android application, designed exclusively for students and doctors. Our app offers a seamless learning experience, with features such as chat for communication, attendance code creation, grade updates, quiz solving, and material uploads and downloads. With a user-friendly interface, students and doctors can collaborate effectively and enhance their educational journey. The chat feature enables real-time communication, fostering interaction and knowledge sharing. Attendance tracking is made effortless, as students can listen and respond to unique attendance codes generated by doctors. Keeping track of academic progress is simplified, as doctors can easily update and manage student grades. Engaging quizzes and assessments promote active learning and knowledge retention. Accessing study materials and resources is convenient with our app's upload and download capabilities.

1.1 Purpose

- E-Learning system helps doctors and students to interact with each other with no face to face needed, it handles some operations between them, on doctors side, they can create attendance code, upload material, chat with students, create a quiz and see students grades, And on the side of students, they can solve a quiz, attend a lecture, download material, chat with the doctor of the course.

1.2 Scope

1) User registration and profiles: -

Students and doctors can create their own accounts and set up their relevant information.

2) User roles and access: -

Differentiate between student and doctor roles to provide appropriate access and features.

3) Chat and communication: -

Enable real-time chat between students and doctors.

4) Attendance management: -

implement attendance tracking, allowing doctors to generate unique attendance codes for each session and students to mark their attendance using those codes.

5) Grades management: -

Provide a system for doctors to update and manage student grades, allowing students to view their grades and academic progress.

6) Quizzes and Assessments: -

Design a quiz module that allows doctors to create quizzes and assessments, and students to attempt and submit their answers.

7) Material management: -

Enable doctors to upload study materials, lecture notes, resources, and other relevant documents for students to access and download.

8) Data Management: -

Establish a database system (firebase) to store user profiles, grades, attendance records, quiz results, and other relevant data securely.

1.3 Definitions, acronyms, and abbreviations

- In a table, list all needed ones. Consider the audience doesn't know
  - Think as following: Document has abbreviation ATM..IFF audience doesn't know it, let's clarify it.

2. Requirements

2.1 Functional Requirements

Login

Code	Requirements statement	Must/should	Comments
FR001	To login on the system, the user must have an email and password saved on database of the system to provide any updates on the user's data.	Must	Each doctor has a unique code to represent the user of the system.

FR002	The user can have the option to change the password while logging in if it's forgotten by clicking on forgot my password label and entering the new email.	Must	None
FR003	The login page should be the first page that appears after opening the system.	Must	None
FR004	If either of the Email or Password labels or both is left empty the system should report the user by a message.	Must	None
FR005	If the user entered any of the email or the password wrong the user should be informed by a message.	Must	None

Register

Code	Requirements statement	Must/should	Comments
FR006	The user must enter his/her name, email, password and which doctor or student.	Must	Must not be empty.
FR007	The data will be saved on the database of the system.	Must	None

FR008	The password of each user on the system should has at least 2 numbers and 1 special character.	Must	Totally the password should include at least 8 digits.
FR009	The Register Process is only allowed if the user doesn't have account on the system.	Must	None
FR0010	If the user tried to register on the system and it already has an account, the system will throw a warning that the user has account.	Must	None

Add new course

Code	Requirements statement	Must/should	Comments
FR001	The doctor is the only one allowed to add a course on the system.	Must	None
FR002	The doctor must set a name for the course.	Must	None
FR003	While creating the course the doctor must determine the course grades.	Must	None

FR004	The course grades are divided into quizzes, attendance, projects.	Must	None
FR005	The grades are divided according to what the doctor want.	Must	None

Enroll to courses

Code	Requirements statement	Must/should	Comments
FR001	The students are allowed to enroll to an existing course.	Must	None
FR002	They can register to the course using the course id.	Must	None
FR003	The course id is given automatically to the doctor when the course is created.	Must	None
FR004	The doctor gives the students the course code any way out of the system.	Must	None
FR005	The student must enter the code correctly, otherwise, the system will tell the student that the course doesn’t exist.	Must	None

Upload materials

Code	Requirements statement	Must/should	Comments
FR001	The path of the material will be saved on the database of the system.	Must	None
FR002	The material which will be uploaded must be only pdfs.	Must	None
FR003	The material must be given a name while uploading.	Must	None

Create attendance codes

Code	Requirements statement	Must/should	Comments
FR001	The attendance code will be generated randomly.	Must	None
FR002	The attendance code consists of 4 decimal numbers from 0 to 9.	Must	None
FR003	The attendance code will be given to students out of the scope of the system.	Must	None

Create Quiz

Code	Requirements statement	Must/should	Comments
FR001	The doctor is the only one who can create a quiz.	Must	None
FR002	The quiz will appear to the student once it's created.	Must	None
FR003	The doctor will create the quiz consisting of a question and 4 answers.	Must	None
FR004	The doctor must determine the right answer.	Must	None
FR005	The questions, the answers and the right answer will be saved in the database to retrieved.	Must	None

Solve Quiz

Code	Requirements statement	Must/should	Comments
FR001	Once the quiz is created, the student is allowed to submit answers to it.	Must	None
FR002	The student is allowed to choose only one answer for each question.	Must	None

FR003	When the student reaches the last question, the button changes from next question to finish.	Must	None

Chat on doctor’s side

Code	Requirements statement	Must/should	Comments
FR001	The doctor will chat with all students at once (group).	Must	None
FR002	Any received message on the doctor’s side, must contain the name of the sender.	Must	None

Chat on student’s side

Code	Requirements statement	Must/should	Comments
FR001	The student will chat with the doctor in group with other students.	Must	None
FR002	The student will chat only with the doctor of the course and the members students of the course.	Must	None

Attending Lectures

Code	Requirements statement	Must/should	Comments
FR001	The student must enter the attendance code correctly.	Must	None
FR002	The student must enter the attendance code correctly.	Must	None

Download Materials

Code	Requirements statement	Must/should	Comments
FR001	The student is allowed to download the material using its path.	Must	None
FR002	The student is allowed to download many PDFs, and one PDF many times.	Must	None

Show grades to students

Code	Requirements statement	Must/should	Comments
FR001	The grades will appear at first with its total.	Must	None
FR002	All quizzes grades will appear after summation of all of them.	Should	None

Show grades to doctors

Code	Requirements statement	Must/should	Comments
FR001	The grades appear to the doctor with the id of the student.	Must	None
FR002	As long as the project grade is not assigned, it won’t appear.	Must	None
FR003	If the attendance and quizzes grades equal to zero it will appear as 0.	Must	None

2.2 Non-Functional Requirements

Reliability

Code	Requirements statement	Must/should	Comments
NFR001	System opens in a few seconds.	Must	None

Usability

Code	Requirements statement	Must/should	Comments
NFR001	The system should be easy to use and not complex so that a	Must	None

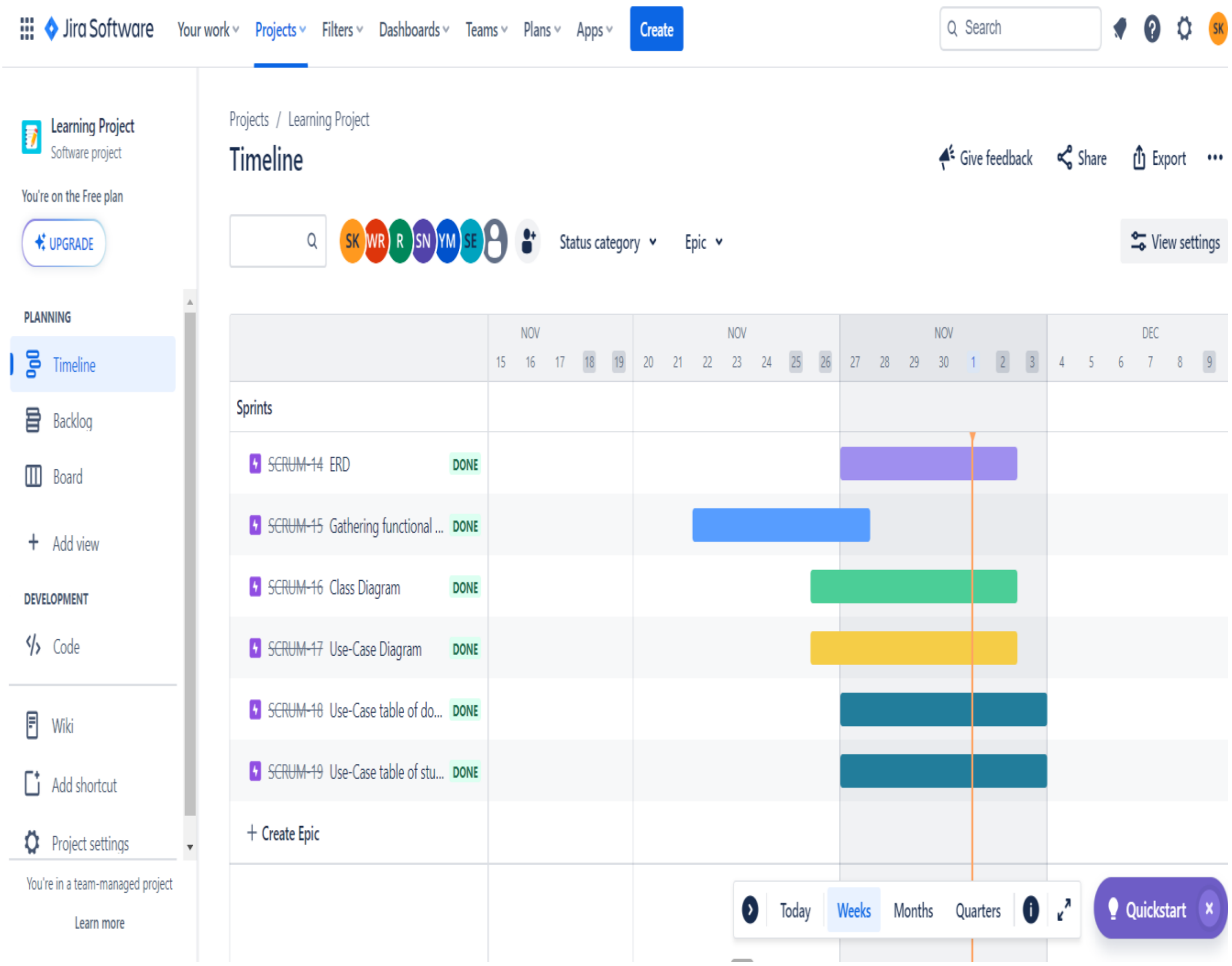
	user with basics in computer skills will use it perfectly.		
NFR002	At most 2 seconds to measure system response times like clicking a button.	Must	None
NFR003	At most 5 seconds for a user to complete a basic task like retrieving.	Must	None

Scalability

Code	Requirements statement	Must/should	Comments
NFR001	The system should be able to handle a large number of concurrent users, allowing multiple students and instructors to access the system simultaneously without performance degradation.	should	

3. Used Technologies

- **Jira**: Is a website to make a board, put all tasks on it and assign each task to each member. We used it to manage our project and tasks upon the team members by organizing the tasks, creating sprints, and keeping track of our progress.



~~SCRUM-14~~ ERD DONE

~~SCRUM-15~~ Gathering functional ... DONE

~~SCRUM-16~~ Class Diagram DONE

~~SCRUM-17~~ Use-Case Diagram DONE

~~SCRUM-18~~ Use-Case table of do... DONE

~~SCRUM-19~~ Use-Case table of stu... DONE

- **GitHub repository link**: This repository is made to make it easy for our team to communicate with each other. Every member will upload his copy of code with his modification to the code, so every member will always know the updates. This repository will help avoiding and reducing the reasons of errors.

  Provide your repository public link

  https://github.com/samaAhmedKabil/E-Learning-App

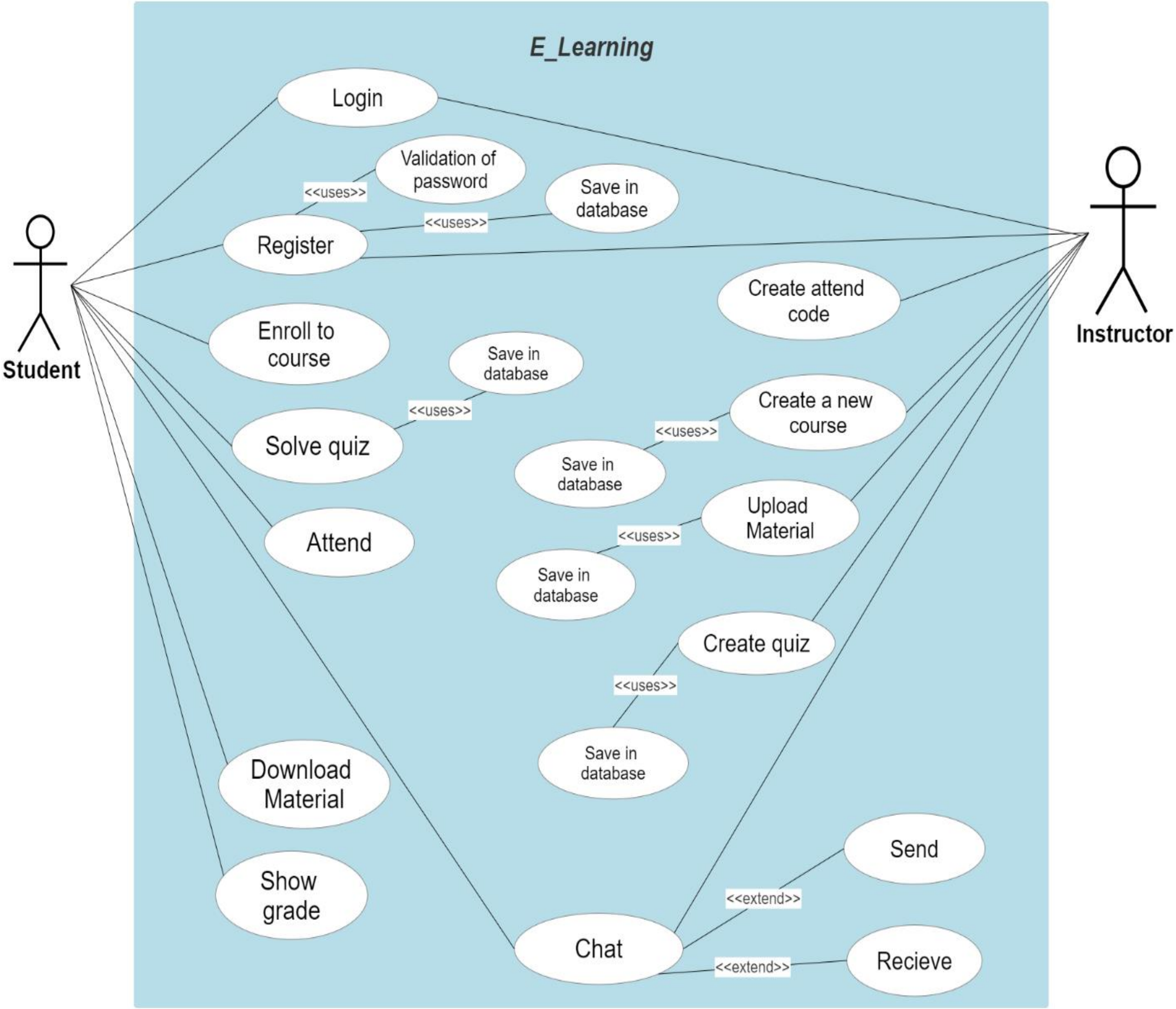
- **Frontend technology:** Android technology using XML for front-end design for multiple activities.
- **Backend technology:** Android technology using Java supported with firebase Realtime, storage, and authentication.

4. System Models

Use tools to draw diagrams

4.1 Use Case Model

- Draw the use case model expressing the systems, actors, use cases, and relationships



- **Use Case Description Tables**
  - Using below table template, for each requirement write a use case table
  - If one requirement is so big, you could divide it to more than table

Use Case Number:	E learning. diagram#. Use-case#1	
Use Case Name:	Login	
Actors:	Doctor and student	
Overview:	It ensures that only registered and authorized users can access the system.	
Related use cases:		
Event (Stimulus):	**User Action**	**System Action**
	1-Doctor and student enter e-mail and password.	1-The login function verifies the identity of the user by matching the provided username and password with the stored credentials in the system. if email and password true can access system if wrong have not account
Exceptions:	If either of the Email or Password labels or both is left empty the system should report to the user by message. If the user entered any of the email or the password wrong the user should be informed by a message.	
Comments:		

Use Case Number:	002	
Use Case Name:	Register	
Actors:	Student	
Overview:	Student will use this case to create an account.	
Related use cases:	Database updated; user account created.	
Event (Stimulus):	**User Action**	**System Action**
	1- Student and doctors enters name, password e-mail and choose student.	1- Database updated, student or doctors account is created.
Exceptions:	If the student or the doctor tries to create account with the existing account, system will display a message that this account is already existed.	
Comments:	Students and doctors must not leave any field empty, and password should be 8 digits at least. This case is allowed only if the student or the doctor don't have an account.	

Use Case Number:	E learning. diagram#. Use-case#3	
Use Case Name:	create attendance code	
Actors:	Doctor	
Overview:	allow user to attend the lecture.	
Related use cases:	save code in database.	
Event (Stimulus):	**User Action**	**System Action**
	1-allow system generate system code.	1-The attendance code will be generated randomly.
		2-save code database.
Exceptions:		
Comments:	The attendance code consists of 4 decimal numbers from 0 to 9. The attendance code will be given to students out of the scope of the system.	

Use Case Number:	E learning. diagram#. Use-case#3	
Use Case Name:	create attendance code	
Actors:	Doctor	
Overview:	allow user to attend the lecture.	
Related use cases:	save code in database.	
Event (Stimulus):	**User Action**	**System Action**
	1-allow system generate system code.	1-The attendance code will be generated randomly.
		2-save code database.
Exceptions:		
Comments:	The attendance code consists of 4 decimal numbers from 0 to 9. The attendance code will be given to students out of the scope of the system.	

Use Case Number:	E learning. diagram#. Use-case#4	
Use Case Name:	create a new course	
Actors:	Doctor	
Overview:	course creation with its details.	
Related use cases:	The data will be saved on the database of the system.	
Event (Stimulus):	**User Action**	**System Action**
	1-The doctor must set a name for the course.	1-The data will be saved on the database of the system.
	2-While creating the course the doctor must determine the course grades.	
	3-The course grades are divided into quizzes, attendance, projects.	
Exceptions:		
Comments:		

Use Case Number:	E learning. diagram#. Use-case#5	
Use Case Name:	upload material	
Actors:	Doctor	
Overview:	The material which will be uploaded must be only pdfs. The path of the material will be saved on the database of the system.	
Related use cases:	The path of the material will be saved on the database of the system.	
Event (Stimulus):	**User Action**	**System Action**
	1-The material must be given a name and a path while uploading.	1-save path and name in database.
Exceptions:		
Comments:		

Use Case Number:	E learning. diagram#. Use-case#6	
Use Case Name:	create quiz	
Actors:	Doctor	
Overview:	The student test and update total grade quizzes.	
Related use cases:	The questions, the answers and the right answer will be saved in the database to retrieved.	
Event (Stimulus):	**User Action**	**System Action**
	1-The doctor will create the quiz consisting of a question and 4 answers.	1-The questions, the answers and the right answer will be saved in the database to retrieved.
	2-The doctor must determine the right answer.	
Exceptions:		
Comments:		

Use Case Number:	E learning. diagram#. Use-case#7	
Use Case Name:	Chat	
Actors:	Doctor and student	
Overview:	Doctor can communicate with students.	
Related use cases:		
Event (Stimulus):	**User Action**	**System Action**
	1-The doctor will chat with all students at once (group).	1-System allows the real-time delivery of messages
	2-Student sends message to doctor or another student.	2-System allows the real-time delivery of messages.
Exceptions:		
Comments:		

Use Case Number:	008	
Use Case Name:	Enroll to course	
Actors:	Student	
Overview:	Students enrolls to a course to be able to study.	
Related use cases:	Login and adding new course.	
Event(Stimulus):	**User Action**	**System Action**
	1- Student enrolls the course by typing course ID.	1- Database updated, student is enrolled into the course.
Exceptions:	If student enters wrong course ID, system will display a message that ID isn't correct.	
Comments:	Students are allowed to enroll to only an existing course.	

Use Case Number:	009	
Use Case Name:	Solve a quiz	
Actors:	Student	
Overview:	The quiz is available to students.	
Related use cases:	Login and enroll to course.	
Event (Stimulus):	**User Action**	**System Action**
	1- Student enters the quiz and starts solving questions.	1- Database updated, system starts to save the chosen answers.
	2- Student submits his answers and finished the quiz.	2- Database updated, score is recorded.
Exceptions:		
Comments:	Student is allowed to choose only one answer.	

Use Case Number:	0010	
Use Case Name:	Attending lectures	
Actors:	Student	
Overview:	Student sets the given attendance code to assure attending lectures.	
Related use cases:	Login, enroll to course and create attend codes.	
Event (Stimulus):	**User Action**	**System Action**
	1- Student enters attendance code.	1- Database updated, attendance mark is marked.
	2- Student enters wrong attendance code.	2- System informs student with a message that code is wrong.
Exceptions:	If student enters wrong attendance code, system will display a message that code is wrong.	
Comments:		

Use Case Number:	0011	
Use Case Name:	Download a material	
Actors:	Student	
Overview:	Student can download the material of his/her course.	
Related use cases:	Login, enroll to course and upload material.	
Event (Stimulus):	**User Action**	**System Action**
	1- Student clicks on the material to be downloaded.	1- System will deliver user to browser to complete the download process.
Exceptions:		
Comments:	Student is allowed to download the same material many times.	

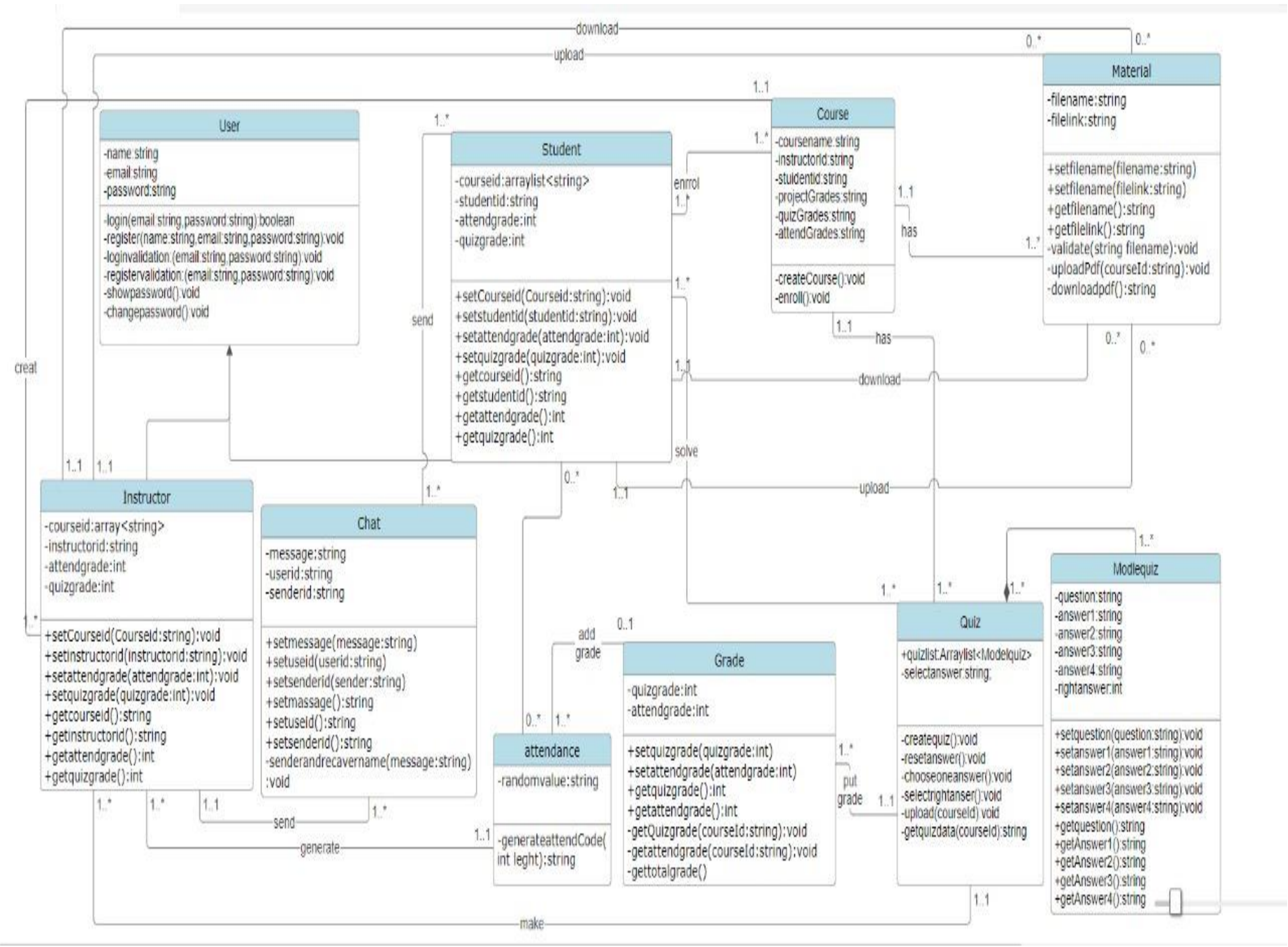
Use Case Number:	0012	
Use Case Name:	Show a grade	
Actors:	Student	
Overview:	Student can show his/her finished quizzes' grades.	
Related use cases:	Login, enroll to course and solve a quiz.	
Event (Stimulus):	**User Action**	**System Action**
	1- Student show his/her grade.	1- Grades will appear with its current total.
Exceptions:		
Comments:	All quizzes grades will appear after summation of all of them.	

Use Case Number:	0013	
Use Case Name:	Show a grade	
Actors:	Doctor	
Overview:	doctor can show their students grades.	
Related use cases:	Login, create course and create a quiz and attendance code.	
Event (Stimulus):	**User Action**	**System Action**
	1- Doctors see grades of students with their id.	1- Grades will appear with its current total.
Exceptions:		
Comments:	All quizzes grades will appear after summation of all of them.	

4.2 Class diagrams

- You should provide your class diagram. In case on diagram is so complex, divide it to several ones of reasonable size.
- Put Relationships between classes and the types of relationships.
- Put multiplicity.
- Put relationship name (example, faculty "offer" course).
- Put attributes in the classes.

- Put functions & Put parameters



• Use Classes Description Tables

Class ID	Class Name	Description
1	User	This class manages user information such as name, email, and password, and handles login, registration and Allows the user to change and show their password.
2	Instructor	This class managing instructor information such as their ID, grades of students which related to attendance and quizzes, and the course IDs they are associated with, and handle setting and retrieving this student-related information.
3	Student	This class managing student information such as their ID, grades related to attendance and quizzes, and the course IDs they are associated with, and handle setting and retrieving this student-related information.
4	Chat	This class use to manage chat-related information, including the message content, user IDs involved in the conversation, and methods to set and retrieve these details, and retrieve the name of sender and receiver of any message in chat.
5	Attendance	This class use to handle the generation of attendance codes, for students to input and confirm their attendance.
6	Course	This class use to handle course-related information such as the course name, instructor and student IDs, and various types of grades (project, quiz, attendance) and use to create a new course within the system and enroll students into the course.
7	Grade	This class use to manage and provide functionalities related to grading, including setting and getting grades for quizzes and attendance, Retrieves the quizzes and attendance grade associated with a specific course ID., calculating the total grade by combining individual grades.
8	ModelQuiz	This class manage the details of model of one question, allowing setting and getting question content, and identifying the correct answer.

Class ID	Class Name	Description
9	Quiz	This class use handle functionalities related to managing quizzes, creating new quizzes, resets the selected answer, select the right answer for a question, that used during evaluation, Allows the user to select or choose one answer from four choices in a quiz question, uploading quiz data, return quiz data for a given course ID
10	Material	This class use to handle the storage of file-related information, setting and getting filenames and file links, validating if putting the filenames during upload file, and performing operations like uploading and downloading PDF.

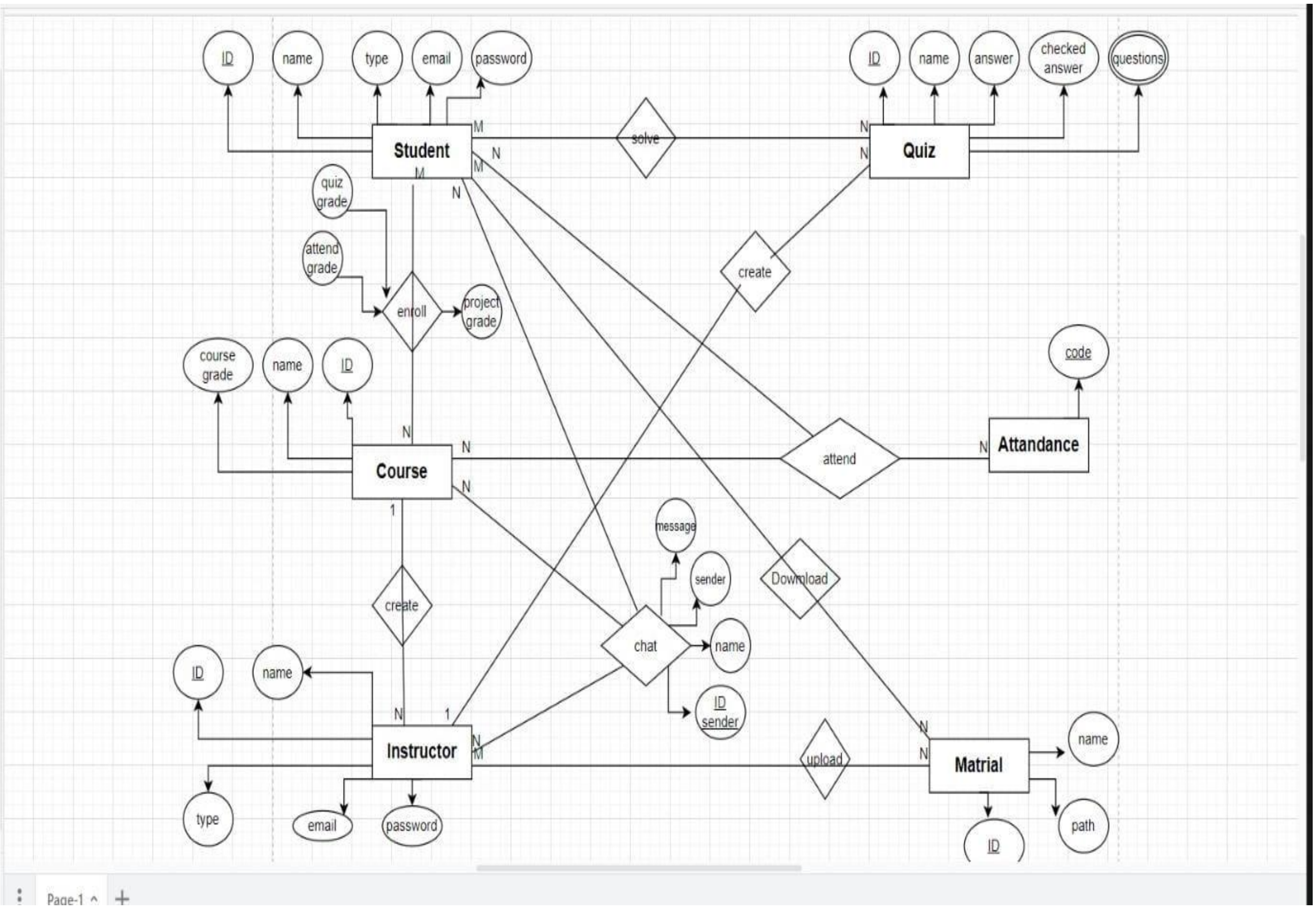
4.3 Sequence diagrams

- List Sequence diagrams for all requirements. Provide each Sequence an ID.
- Not necessary diagram per requirement. May be better to merge more than a flow. Typically, 1 sequence diagram per your use case table.
- Make sure that each object in the sequence diagram has a corresponding class in the class description table above. If not, it will be REJECTED.
- Put actual function calls with proper parameters and return types corresponding to class diagrams.
  - Please always specify the parameters in the call, matching the class diagram.

Insert Sequence diagrams Images Here

4.4 Physical Entity-Relationship Diagram

- Provide the ERD Diagram
- Convert all entity objects of the class diagram and their relationships into ERD
- Put Relationships between entities.
- Put relationship name (example, faculty "offer" course).
- Put attributes of the entities (primary key, Foreign key…).
- Put cardinality (1:1, 1:m, m: n,..).



جامعة الأهرام الكندية
AHRAM CANADIAN UNIVERSITY

Faculty of Computer Science and Information Technology

Ownership Report

Student ID	Student Name	What the student done
42110274	*Sama Ahmed*	*Functions and non-functions requirements.*
42110196	*Sohila Emam*	*ERD.*
42110167	*Shimaa Naser*	*Use-case diagram.*
42110343	*Rahma Mahmoud*	*Use-case table of instructor/doctor.*
42110227	*Yara Asuity*	*Class diagram.*
42110122	*Waleed Rashad*	*Use-case table of students.*