

Democratic Republic of Hellenic  
University of Piraeus  
Department of Informatics

People's Democratic Republic of Algeria  
Abou Bakr Belkaid - Tlemcen University  
Faculty of Sciences  
Department of Computer Science

Thesis submitted for the degree of Master in Computer  
Science

*Option: Software Engineering*

*Theme*

**Online Examination Management System**

**Realized by:**

- Miss. Alaa TAHRAOUI.

**Presented on 03 July 2022 in front of the jury composed of:**

- Mr.Chouiti Sidi Mohammed. (*President*)
- Mr. Mahfoud Houari. (*Examiner*)
- Mrs. Seladji Yassamine (*Supervisor*)
- Mr. Emmanouil Skondras. (*Co-supervisor*)

academic year: 2021-2022

# Acknowledgements

*First of all, we thank Almighty Allah, who has granted us by this grace courage, patience and help during all these years and also to accomplish this work and reach this level.*

*By way of appreciation, I would like to express my thanks to my professor and supervisor, Mrs. Yassamine SELADJI, for guiding, helping and advising us in the realization of this project and writing this dissertation.*

*I would like to thank Mr. Emmanouil Skondras, professor at the University of Piraeus for his help in carrying out this work.*

*I would also like to thank the members of the Jury Mr. CHOUI Sidi Mohammed and Mr. MAHFOUD Houari for having accepted to be part of this jury and who honored us with their presence and the evaluation of our work.*

*and finally a very warm thanks to our parents and their great support, their patience and their contributions, also to all the professors in our department, as well as to our friends and family.*

## *Dedications*

*I dedicate my graduation project with deep gratitude and sincere words to:*

*My very dear parents who have always been here for me, and who helped me in all aspects and especially morale, I want to tell them that no words can express gratitude, love, respect that I have for you. May Allah grant you good health and long life and happiness in your lives.*

*To all my sisters “Nesrine”, “Nihel” and “Manel” and her little family, and to my only brother who is far from us “iheb”, Words are hardly enough to express the love and affection I have for you.*

*To the most important people to whom I owe great respect and who have supported me throughout this project, “Zakaria” I really appreciate your efforts and your help.*

*And finally to all my family, my friends, and anyone who directly or indirectly contributed in the realization of this thesis.*

# Table of Contents

List of Figures .....	6
List of Tables .....	7
General Introduction .....	8
1. Overall context.....	8
2. Problem.....	8
3. Objective.....	9
4. Thesis organization .....	9
Chapter I: Examination Systems.....	10
1. Introduction.....	11
2. Traditional examination definition .....	11
3. Online examination definition .....	11
4. Steps of online exams conduction .....	11
5. Traditional VS Online examination .....	12
6. Methods of cheating in online exams .....	13
7. Existing Systems.....	13
8. UnivExam VS Existing systems .....	14
9. Conclusion .....	16
Chapter II: Analysis and Conception.....	17
1. Introduction.....	18
2. Development Process.....	18
2.1. Waterfall Model.....	18
2.2. Project management.....	18
3. Requirements specification.....	21
3.1. Functional requirements .....	21
3.1.1. Student.....	21
3.1.2. Professor .....	21
3.1.3. Administrator.....	21
3.2. Non-functional requirements .....	21
4. Architectural Patterns .....	22
4.1. MVC pattern .....	22
5. Design Patterns .....	24
5.1. Observer pattern.....	24
5.2. Strategy pattern.....	24
6. Conception and modeling .....	24
6.1. Use cases diagrams .....	24
6.1.1. Student use case.....	24
6.1.2. Professor use case.....	26
6.1.3. Admin use case.....	28
6.2. Sequences diagrams.....	30
6.2.1. Authentication sequence diagram .....	30
6.2.2. Student Authentication sequence diagram .....	31
6.2.3. Consult taken exam sequence diagram .....	32
6.2.4. Take exam sequence diagram .....	32

6.2.5. Add exam sequence diagram .....	34
6.3. Class diagram.....	34
7. Conclusion .....	35
Chapter III: Realization and Development .....	36
1. Introduction.....	37
2. UnivExam architecture .....	37
3. UnivExam project structure .....	37
3.1. Web applications .....	38
3.1.1. Frontend web application .....	38
3.1.2. Backend web application.....	38
3.1.3. Shared application .....	39
3.2. Database.....	39
4. Development environment.....	39
4.1. Softwares and platforms used.....	39
4.2. Technologies used .....	40
5. Production environment.....	41
6. UnivExam Graphical interfaces.....	42
6.1. Front-office interfaces .....	42
6.1.1. Authentication interface .....	42
6.1.2. Take exam interface .....	43
6.2. Back-office interfaces .....	44
6.2.1. Professor interfaces .....	44
6.2.1.1. Create exam interface.....	45
6.2.2. Admin interfaces .....	46
6.2.2.1. Create student interface.....	46
6.2.2.2. Specialties list interface .....	47
7. System Security .....	47
8. Quality Features .....	48
9. Conclusion .....	48
Conclusion and Perspectives .....	49
1. Conclusion .....	49
2. Perspectives.....	49
References.....	50
Abstract.....	52
Αφηρημένα .....	52
Résumé.....	52
ملخص.....	52

## List of Figures

Figure 1: Difference between Traditional and Online Examination.....	12
Figure 2: Mercer and Mettl's test list.....	14
Figure 3: Eklavya's exam list .....	14
Figure 4: Waterfall cycle .....	18
Figure 5: Trello Board .....	19
Figure 6: Work environment preparation task .....	20
Figure 7: Fine tuning card.....	20
Figure 8: MVC pattern in angular.....	23
Figure 9: MVC pattern elements .....	23
Figure 10: Student Use Case Diagram.....	25
Figure 11: Professor Use Case Diagram.....	27
Figure 12: Administrator Use Case Diagram .....	28
Figure 13: Authentication sequence diagram .....	30
Figure 14: Student authentication sequence diagram .....	31
Figure 15: Consult taken exam sequence diagram .....	32
Figure 16: Take exam sequence diagram .....	33
Figure 17: Add exam sequence diagram.....	34
Figure 18: Class diagram .....	35
Figure 19: UnivExam application.....	37
Figure 20: Login with QR code interface .....	42
Figure 21: Invalid QR code interface .....	43
Figure 22: Authentication with credentials interface.....	43
Figure 23: Take exam interface 1 .....	44
Figure 24: Take exam interface 2 .....	44
Figure 25: Create exam interface.....	45
Figure 26 : Create answers interface.....	46
Figure 27 : Create student interface .....	46
Figure 28 : Specialties list interface.....	47

## List of Tables

Table 1: UnivExam VS existing systems .....	15
Table 2: Descriptions of Student Use Cases .....	25
Table 3: Descriptions of Professor Use Cases .....	27
Table 4: Descriptions of Administrator Use Cases .....	29

# General Introduction

## 1. Overall context

The examinations are one of the important tools to be relied upon in measuring and evaluating students' abilities, and knowing the extent of their achievement level. It also helps in knowing the extent of how the required educational outcomes are achieved, and the educational activities that the professor performs, and helps raise the achievement levels of students.

These exams are held at all educational levels, primary, middle and high schools and finally universities. If we take the university exams as an example, there can be divided into three or more written examination types: Weekly Tests, Continuous Assessments and the Final Exams that takes place at the end of each semester, and every semester the universities go through a series of procedures for the preparations of the exams to be highly efficient in the process of measurement and evaluation.

In the last three years, universities have begun to face difficulties in the organization and the preparation of the exams, as well as in the possibility of conducting them. These difficulties are caused by the appearance of the “COVID-19” virus, which has changed the life mode and affected the study routine. As the World Health Organization indicated, as a result of the Corona pandemic, the necessity of avoiding closed or crowded spaces that involve close contact, this is what applies to the educational institutions, which necessitated their closure.

In conjunction with technology and development in the field of informatics in which we live now, it became easy to find solutions to achieve a balance between following what was issued by the health organization to maintain the safety of citizens on the one hand and the success of the education process, specifically maintaining the credibility of conducting exams on the other hand, this is what we will reach in this study.

## 2. Problem

The closure of educational institutions in general and universities in particular led to a disturbance in the educational system, as it led delaying the onset of the next academic year due to the cancellation and postponement of the exams indefinitely. This prompted the relevant authorities to reconsider the method of education and resort to the use of technological solutions for integration and to cope with these crises that may be faced in the future, as well as in order to facilitate examination preparations.

In addition, the organization and the conduct of exams in the traditional system requires a series of steps to follow and more time to spend as well as great efforts for both students and universities.

It is very difficult for students to come to the test center due to the distance from their home and therefore they will waste a lot of time to reach the center. In addition, they have to wait more than 15 days for the score of the exam they took,



they also need to go back to the university again in case of they want to view their exams (comparing their answers to the professor's answers).

On the other side the universities staff both the professors and the administration members have a bigger part in the preparation of the large number of paper sheets for the questions and the answers of the exams as well as the monitoring the students during the exams by the professors which is time and resources consuming thing to do, on top of all of that the human error that can occur while correcting and the calculation of the grades.

Our project consists on the realization of an application that contributes to the success of the examination process in light of crises and facilitates the long and exhausting procedures in the preparation of the exams and their conduction.

### 3. Objective

The aim of this study is to design and implement a web application that will allow universities to manage and conduct the entire examination procedures online, as it will be cost-effective, time and effort efficient.

This system helps students to take their exams anywhere without seeking to reach the examination center and there will be no obligation for the attendance of the professors during the exam, which reduces the risk of virus infection in case a pandemic like COVID.

Our solution generates results for students to review after completing the exam without having to go back to the university, and also saves professors valuable time from checking students' answer paper sheets, thus reducing error incidents.

Finally, this web application will record videos of students while taking their exams, so that the professors can consult them later if any suspicion of cheating were to be discovered.

### 4. Thesis organization

We will detail the project in this thesis through three chapters:

1. **Chapter 01:** in this chapter we will introduce the examination types and highlight the points of differences between them, then we go over the process of taking examinations online and how cheating can be done, and finally we will present a comparative study of some existing systems with our solution.
2. **Chapter 03:** this is dedicated to the design and the conception of our system, the process of the implementation and requirements specification, as well as the modeling using Unified Modeling Language.
3. **Chapter 04:** this chapter represents the architecture of the system to be developed and the various tools used for the realization, as well as the deployment of the project. Finally, an overview of the interfaces performed.

and we will conclude this thesis with a general conclusion that summarizes our work with future perspectives.

# Chapter I

## Examination Systems

## **1. Introduction**

Examination plays an important role in students' academic career, where the ability to pass it is a great asset to them, and the examination are used for the final assessment of a student's performance in the course training, where they are conducted during the official exams period.

## **2. Traditional examination definition**

The traditional examination is a method followed since the beginning of the educational system in the world, as it is based on paper and pen that are held at the exam halls under the supervision of the examiner, the student needs to come to the exam center with his student identification before they allow him to take his exams.

This manual examination system typically begins with preparing the preparation of question paper by the lecturers, creating the marking scheme, conducting the examination for the students and finally the examiner collects the answer sheets at the end of the exam to be marked with hand by the lecturer.

## **3. Online examination definition**

The online exam is the latest technique of taking the exams that has emerged in recent years to enhance the variability in the education system and reduce the chance of losing the integrity of the exam pattern, as it allows students to take their exams from a distance. This new system can be considered as an e-assessment where is conducted through the internet or in the intranet, which will be remotely and from anywhere.

The online examination system, with its stand, constitutes the main lifeline of the examination process in many educational nowadays. In addition, it is also one of the easiest ways to prepare for exams as it does not contain many steps that consume time and resources and drain huge energy.

## **4. Steps of online exams conduction**

Online Examination is a web-based platform. It allows students to conduct online exams and get their evaluation, and this is done through the following steps:

### **Generate the exam**

This is the first step in which the exams are created and added through an online platform, by adding a set of questions with its answers, as well as scheduling the date of the exam and the start time and its duration.

### **Invite students to exam**

The administrators can use the online exam system to create students accounts with their information and after the exam creation, the invitations are sent automatically to the students.

### Authorize and proctor exam

The authentication in the system plays the first step to access to the students' account. Also, the whole exam is recorded for professors to check later if necessary.

### Real-time Results

The online examination system can provide real-time results and answers, in a way the moment the student submits its answers the system can auto correct his answers.

## 5. Traditional VS Online examination

Many universities around the world have started to abandon the old method and adopt the new method for conducting exams, this is due to the fact that traditional methods are time consuming and involve manual invigilation and grading. However, the exams can be simply generated on a digital platform, allowing students to take them from anywhere in the world. Additionally, online monitoring and automatic grading reduce costs and time, if we take as an example, Monash University in Australia will save nearly AU\$7 million annually by moving 80% of its MCQ exams online [1].

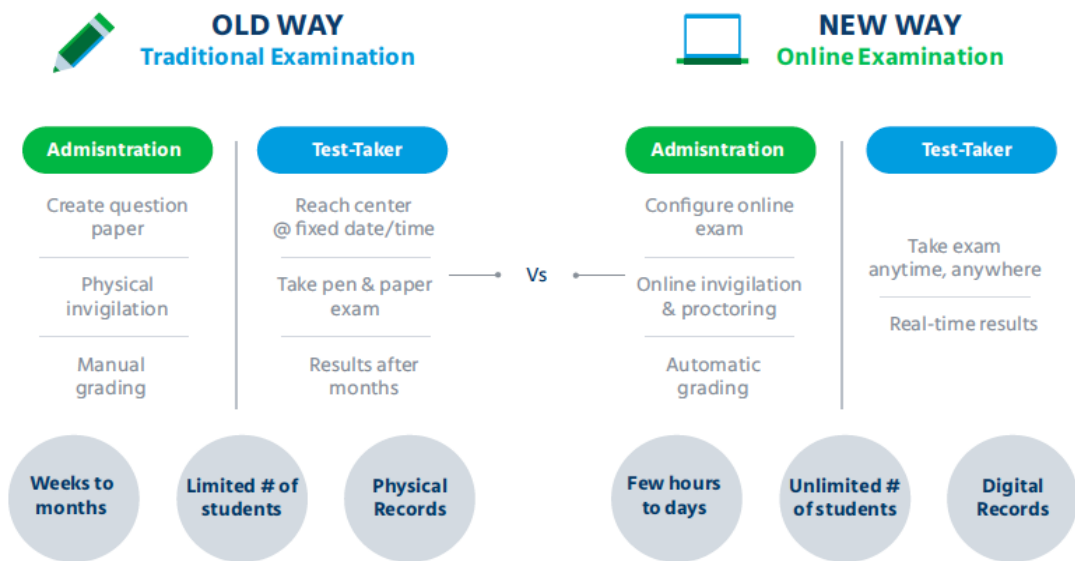


Figure1: Difference between Traditional and Online Examination. [2]

## 6. Methods of cheating in online exams

Cheating is the easiest way for students to get better marks on tests unethically. The cheating could be in traditional exams or on online exams, therefore the colleges devote a lot of resources to combat cheating. In this section we will mention a few possible ways that could the student use to cheat in the online examination.

### **Identity theft**

In some exams, students may be able to find other people to take the tests on their behalf, as they forge the students' identity in order to enter, and this is called identity theft.

### **External help**

Many students' own smartphones, smartwatches, laptops and tablets, therefore they can use them behind the webcam to get the correct answers and solutions of the questions and also, they can get help from their friends or from their family because the exam is considered as remote.

### **Easy navigation**

Since the platform that students use required the internet as a primary factor, the student take advantage of that on their devices to search for answers by using the browser and navigating between tabs, as well as communicating with colleagues through social networking sites or applications, in addition to their abilities to share the screen and publish answers with others.

## 7. Existing Systems

The study of the existing systems is a critical step. It is characterized by the search for applications that are comparable to our project, with the goal of processing and upgrading the existing features and looking for ones that are not currently incorporated. In this section we will introduce two similar platforms to our system:

### **Mercer | Mettl**

Mercer Mettl is an online assessment technology solutions provider, established in 2010 in India [3], where it is used for creating and administering exams and evaluating the candidates.

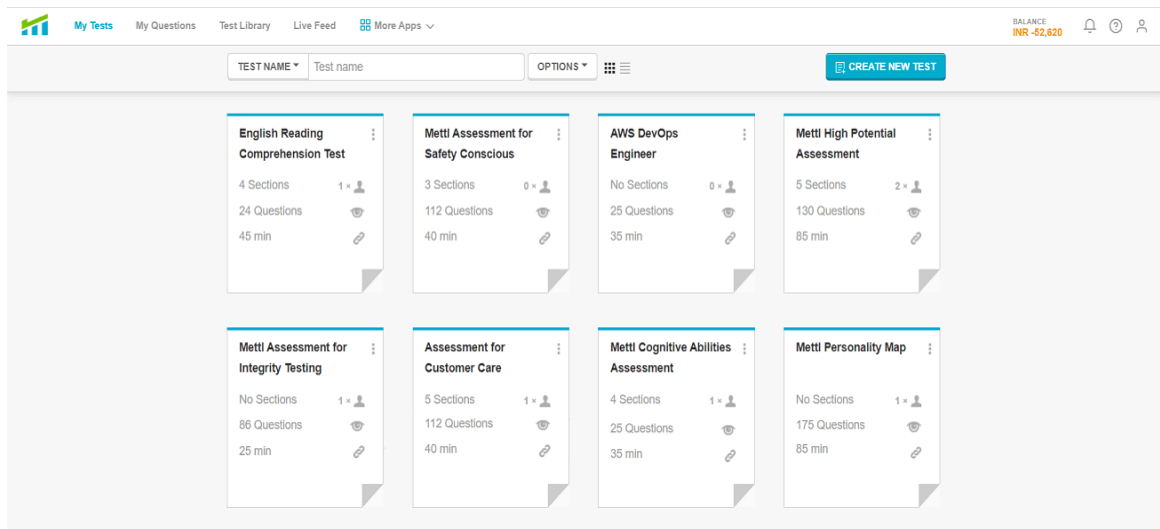


Figure 2: Mercer and Mettl test list. [4]

### Eklavya

Eklavya is a cloud-based online test management system designed for schools, training institutions, and individual educators. The program may be used for test-related tasks such as creating exam forms with questions [5].

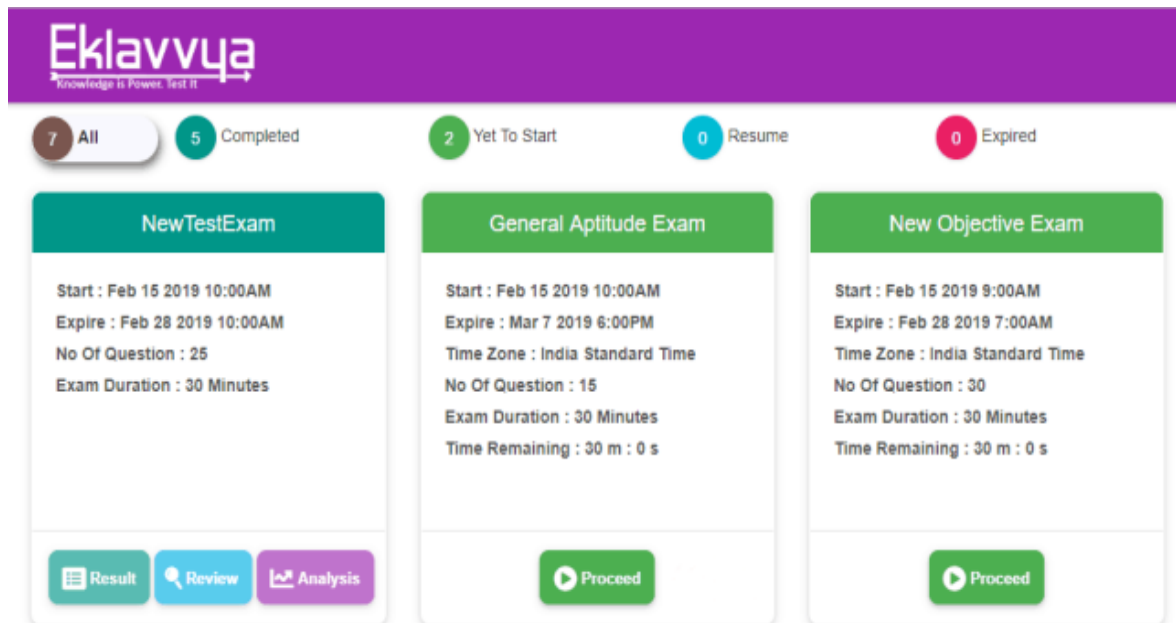


Figure 3: Eklavya Exam list. [6]

## 8. UnivExam VS Existing systems

In the table 1, we will represent the different and common points between the two existing systems introduced above and our solution UnivExam:

	<b>UnivExam</b>	<b>Eklavya</b>	<b>Mercer   Mettl</b>
Login the exam-takers into the system	<ul style="list-style-type: none"> <li>Email and password.</li> <li>QR Code.</li> </ul>	None	None
Exam Proctoring	Video, audio	Video, audio, image, AI	Video, audio, image, AI, Human
Exam Evaluation	Auto	Manually	Auto / Manually
Scheduled the Exam	Exist	Exist	Exist
Send Invitation to exam-takers	In the application	None	Via email
Exam-Takers review their answers and results	Exist	None	None
Check the Scheduled Exams List by Exam-takers	Exist	None	None
Languages	English	7+ Languages	20+ Languages

Table 1: UnivExam VS existing systems.

Table 1 shows a comparison of the existing functionality between our system, and the other online examination management systems Eklavya and Mercer | Mettl. As we can see the Eklavya and Mercer | Mettl systems does not offer an account for the student so there's no login, but UnivExam system offer for the student the possibility to login with two different ways as well as the possibilities to check the scheduled exams list and review their answer and results after taking the exam directly.

In addition, the three systems have the possibility of scheduling the exams and inviting the exam-takers but in different way except the system Eklavya does not support this functionality. We also notice that all of these applications have the option of exam proctoring where it allows the system to detect the cheating cases, but each system has more methods than the other, where we can see that the system Eklavya and Mercer | Mettl has 2-3 techniques more than the UnivExam system.

Also, we observe that Mercer | Mettl system offers two different methods of exams evaluation, where UnivExam has only auto evaluation and Eklavya has only manually evaluation. Finally, both of Eklavya and Mercer | Mettl supports more than 7 languages and for UnivExam supports only one language.

## **9. Conclusion**

In this chapter, we have introduced two ways in which students can take their exams and the stages used in order to prepare for the exam in each of them, we also compared them in terms of consumption of time, resources, cost and effort, as well as we discussed how the student can get high grades using cheating in the online examination. Finally, we studied existing systems that are similar to our system and highlighted the common and different points between them.



# **Chapter II**

## Analysis and Conception

## 1. Introduction

In order to have a good project management and to better structure our work, we made a design based on diagrams for better understanding our system, as well as to facilitate the development and have an efficient work.

## 2. Development Process

The software development process includes all phases required to bring a product from concept to production.

### 2.1. Waterfall Model

The traditional waterfall model splits the life cycle into a series of phases, each one can start after the previous phase is completed and the output of one stage is considered as an input of the next stage as shown in figure 4. Therefore, the development process can be viewed as a sequential flow in waterfall [7].

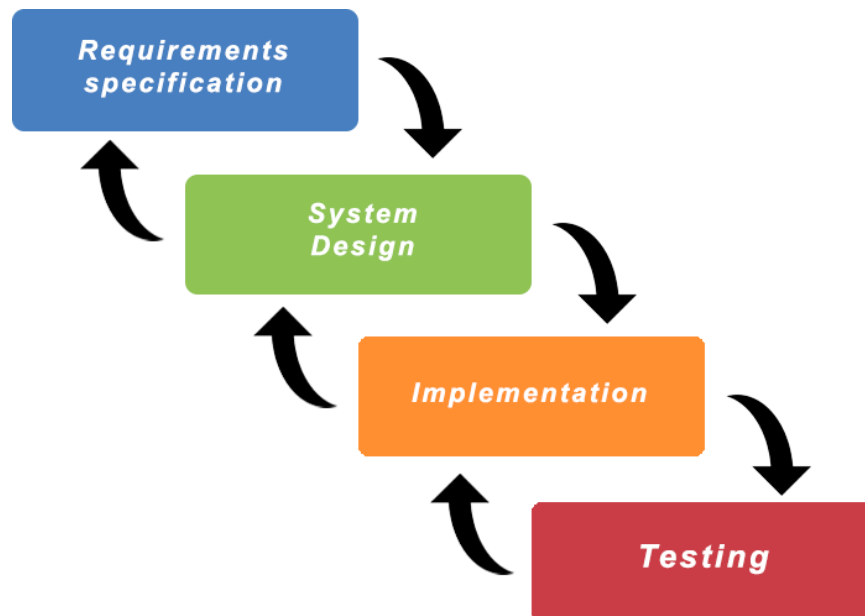


Figure 4: Waterfall cycle.

### 2.2. Project management

Trello is a project management tool that divides your projects into lists. Trello tells you what's being worked on, who's working on what, and where something is in the process all at one glance [8], as well as it indicates the importance of each task so you know where to begin and that is possible by using the labels.

In our project, we used 4 lists (figure 5) and each of them is represented by:

**Backlog list:** which has all the cards that exist in the project, in another way, it has all the tasks that we need to implement.

**To Do list:** the tasks will be moved for the first time to this list, and they are cards that we are going to do.

**In Progress list:** it has the tasks that we are working on now.

**Done list:** which contains the tasks that we finished them.

In addition to the lists, we used 3 colors (figure 5) for indicating the importance of the tasks which are:

**Red color:** the tasks that has a red label means this is the most important task to do and we should to start with.

**Orange color:** which is the second level of the importance and we should work on them after finishing the tasks with high importance.

**Yellow color:** is the less level of the importance, and the tasks are the last one to do them.

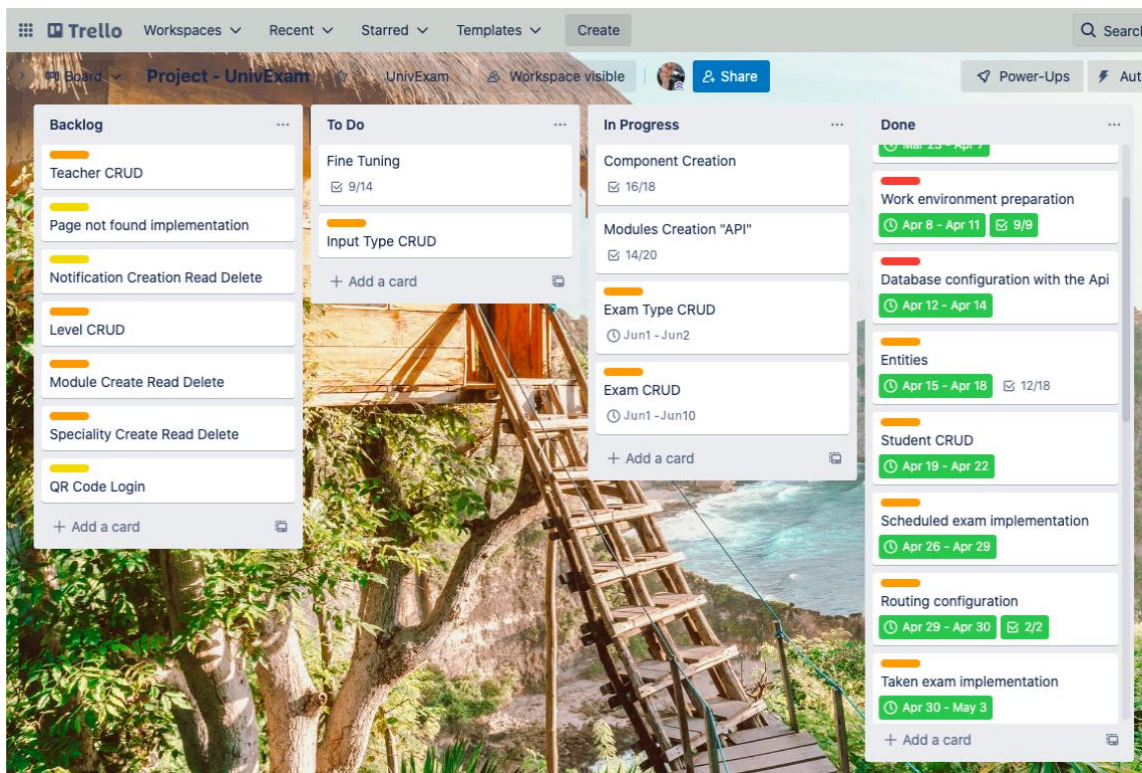


Figure 5: Trello Board

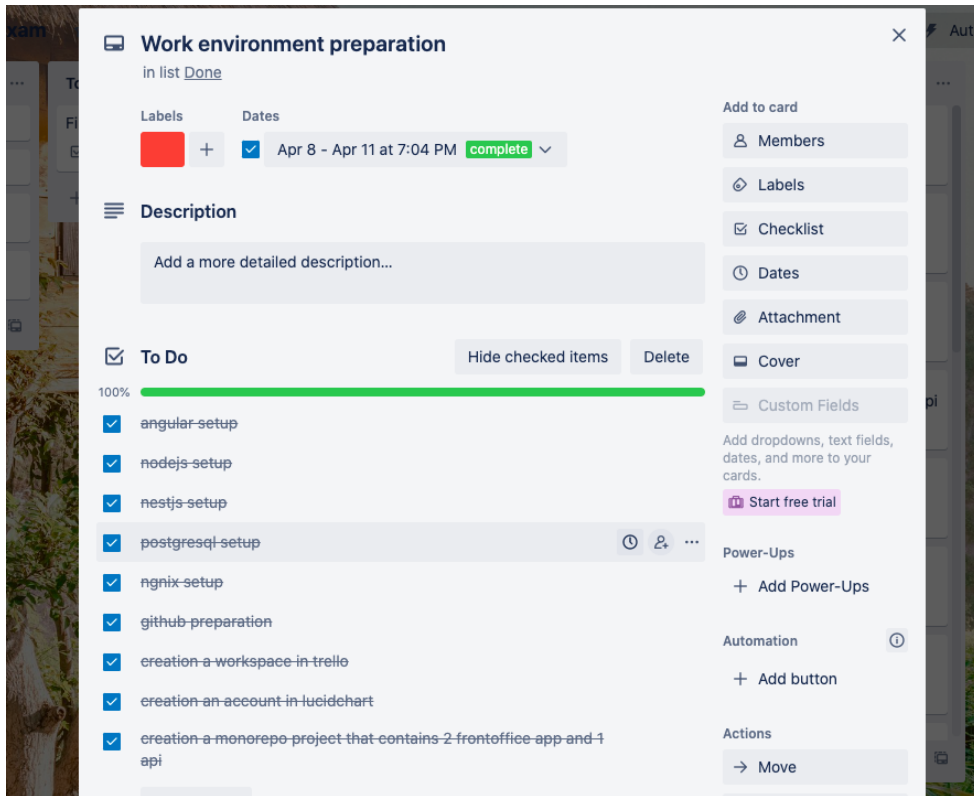


Figure 6: Work environment preparation task

The To Do list in figure 5 contains a card called “Fine Tuning” (figure 7) that does not belong to the project tasks. This card contains a small and simple tasks that need to be added or to be fixed, and we discover them during the implementation of the project tasks.

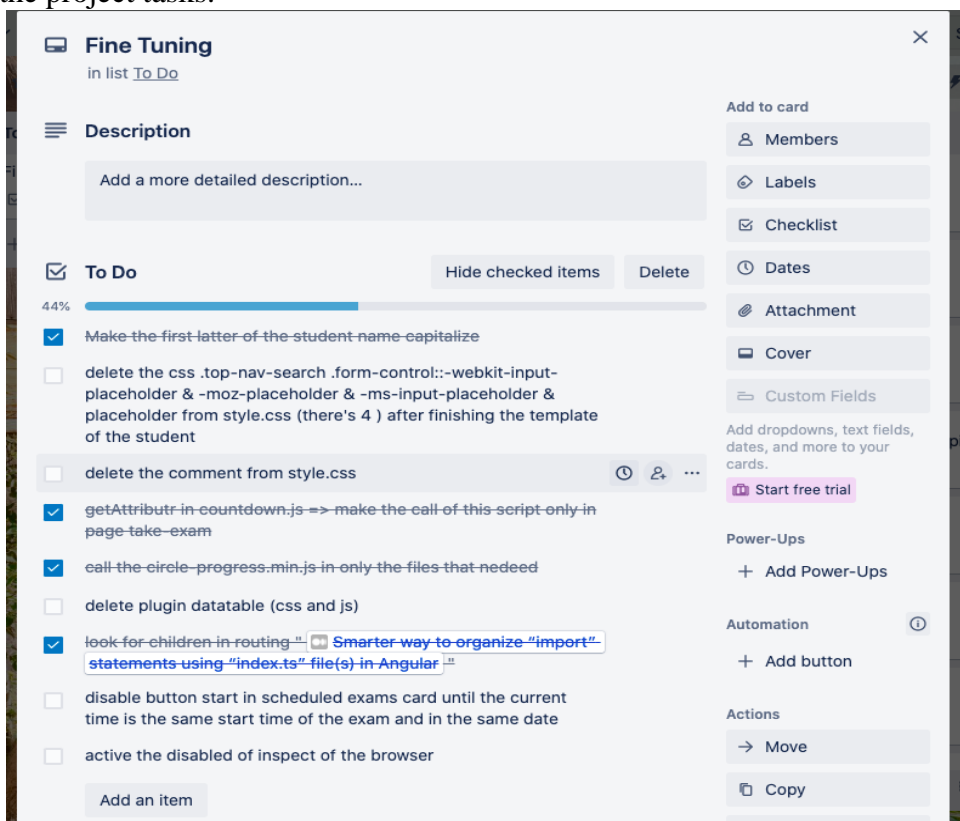


Figure 7: Fine tuning card

### **3. Requirements specification**

The objective of requirements specification is to clearly describe the creation of the application by specifying its functional and non-functional requirements. This specification defines the processes and functions that the system will perform.

#### **3.1. Functional requirements**

Functional requirements describe the services and functions that a software system must offer, they product features that focus on user needs [9]. The actors for the proposed platform have been identified and described as follows:

##### **3.1.1. Student**

- The student has the possibility to take the exams.
- The student has the possibility to consult the result of the exams that he takes.
- The student can consult the notification of the publication of the exams.
- The student can search about the exams.
- The student has two possibilities to login, the first by email and password and the second by scanning the QR code that the he can get it from his profile.

##### **3.1.2. Professor**

- The professor has the possibility to manage the exams.
- The professor can check the students' answers.
- The professor can consult the students.
- The professor can login only with email and password.

##### **3.1.3. Administrator**

- The admin has the possibility to manage the users.
- The admin has the possibility to manage the exam' types and the input's types.
- The admin can manage the modules, levels and specialties.
- The Administrator can login only with email and password.

In addition, the system will respond to these requirements:

- The system needs to generate the QR code for the student.
- The system needs to record the student during the exams.
- The system needs to generate the result of the student answers automatically.
- The system needs to stop the access to take the exam when the date and the time expired.
- The system needs to stop the student from starting the exam if he turned off the webcam or the microphone.
- The system offers two answers' types, checkbox and select.
- The system should send notification to students, that tells that the exam is published.

#### **3.2. Non-functional requirements**

Non-functional requirements support the functional requirements and determine how the software must perform [10], also they act indirectly on the result and on the performance of the user.

### **Usability**

- The system must have an efficient, intuitive and easy-to-use interface with clear and simple design for both beginners and users with experience of similar systems.

### **Portability**

- The system must to work on any operating system.
- The system is designed to be independent by running on different RDBMS: PostgreSQL, MySQL.

### **Performance**

- The performance of the system should be very fast when the students are using it. Furthermore, the system should never run too slowly under any circumstance.

### **Accessibility**

- The application should be accessible by several students at any given time, with no downtime on the system.
- The system should be available 24/7, so that students and professors may utilize it whenever it is convenient for them.

### **Maintainability**

- The system must be adaptable in terms of adding new features and detecting and correcting problems as soon as they occur.

### **Security**

- The system must be secure to preserve data and assure the confidentiality of transactions.
- The system must have secure and flexible management to handle the hashing and storage of user passwords.
- Disconnection after a certain period of inactivity (24h).
- The system must manage all user roles: Administrator, Student, Professor.

## **4. Architectural Patterns**

When designing a system with a user interface, it is important to structure the code in a way that allows easy extension and maintenance [11], that's why we used the MVC models.

### **4.1. MVC pattern**

The View Controller Model indicates that the application will be separated into at least three elements that interact with each other in a homogeneous way. The three parts are as follow:

**Model:** it represents the business end of the application, it manages access and modification of data.

**View:** it used to present the model content as a user interface.

**Controller:** it handles the incoming requests, process actions on data given by Models, and choose Views to display to the user [12].  
 The figure 8 represent the 3 elements of MVC pattern in Angular framework, where the model is a type that can store a value.

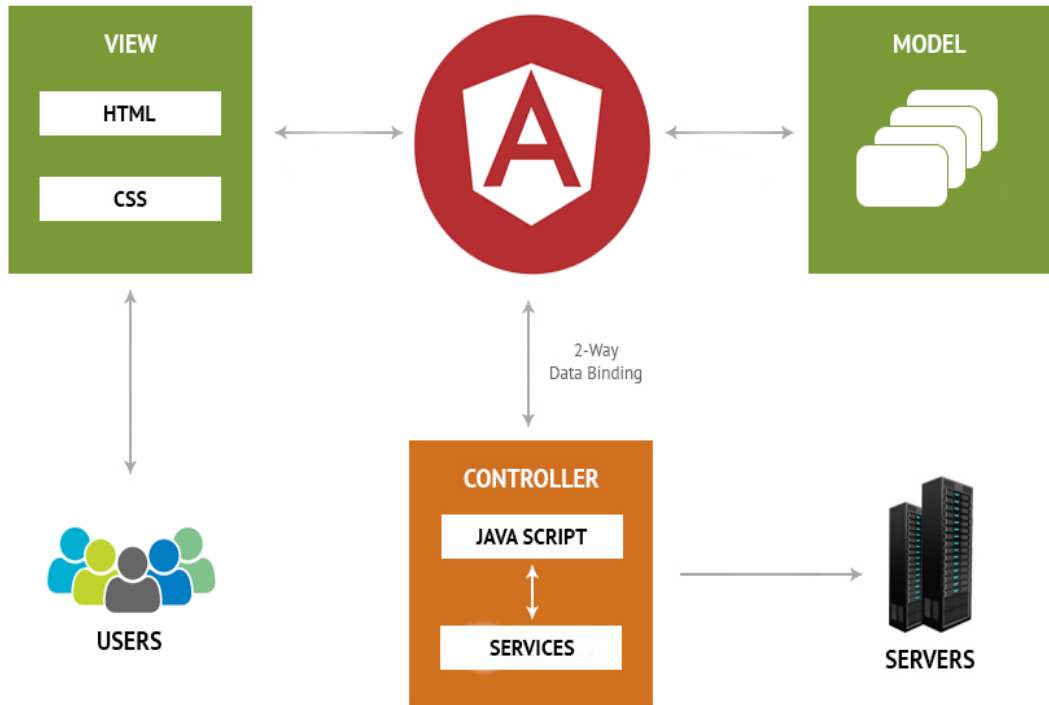


Figure 8: MVC pattern in angular [13]

Figure 9 summarizes our web application as an MVC pattern:

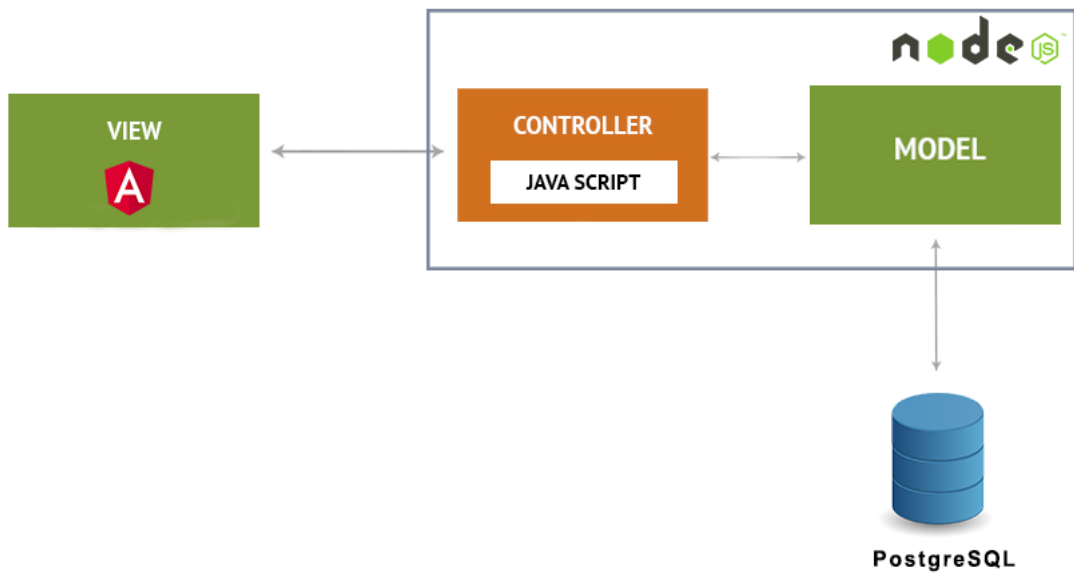


Figure 9: MVC pattern elements

## **5. Design Patterns**

Design pattern is a generic repeatable solution to a typically occurring problem in software design.

### **5.1.Observer pattern**

The observer pattern is a type of behavioral design pattern, it keeps the coherence between classes cooperating with each other while maintaining their independence. It used in one-to-many relationship, when changing the state of an object the objects that are dependent on it update automatically [14]. With the MVC pattern, the views are considered as observers of the model.

### **5.2.Strategy pattern**

Strategy is a behavioral design pattern that allows you to construct a family of algorithms, classify them, and make their objects interchangeable [15]. In MVC architecture the controllers are considered as listeners of the events generated on the views by users (click, change events).

## **6. Conception and modeling**

We used UML modeling language to model our system, it helps in presenting our system objectives in a clear and simplified way. This modeling language contains a set of diagrams to represent the goals, where we used: use case diagram, sequence diagram, class diagram.

### **6.1.Use cases diagrams**

The use case diagram describes the users of a system, and what they are supposed to do. Each action that actors do is represented by a use case.

The following diagrams show the use cases of the different actors of the UnivExam system:

#### **6.1.1. Student use case diagram**

The student is the important user, figure 10 represents the student use case diagram:



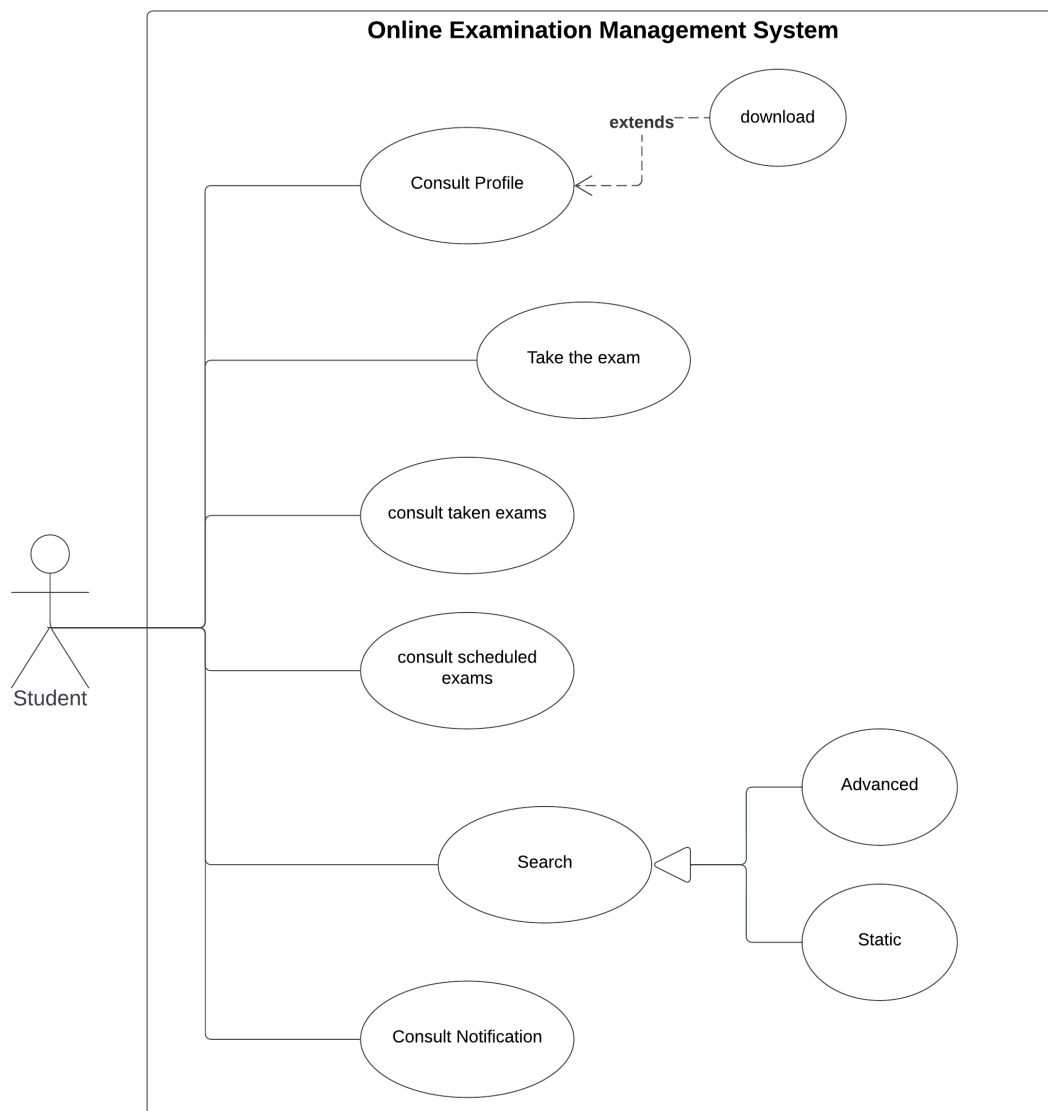


Figure 10: Student Use Case Diagram

The table 2 describes each use case of student and its importance as an implementation task:

Use Cases	Description	Importance
Take the exam	The student can take different types of exams: Tests, Continuous Assessment and final exam. The student can choose only one or multiple answers for each question.	High

Consult taken exams	The student can check his answers of the exam and compare them with the answers that the professor provided, as well as he can check the correct and wrong answer of each question with grade.	High
Consult scheduled exams	The student have the possibility to check the exams that he needs to take.	High
Search	The student can search about the scheduled and taken exams of his specialty and his level by the exam name or exam type.	Low
Consult Notification	The student will receive notification when the professor publishes the exams.	High
Consult Profile	The student can check his information, also he can find and the QR code that helps him to login the system.	Medium
Download	The student can download his the QR code.	Low

Table 2: Descriptions of Student Use Cases

### 6.1.2. Professor use case diagram

The figure 11 represents the professor use case diagram:

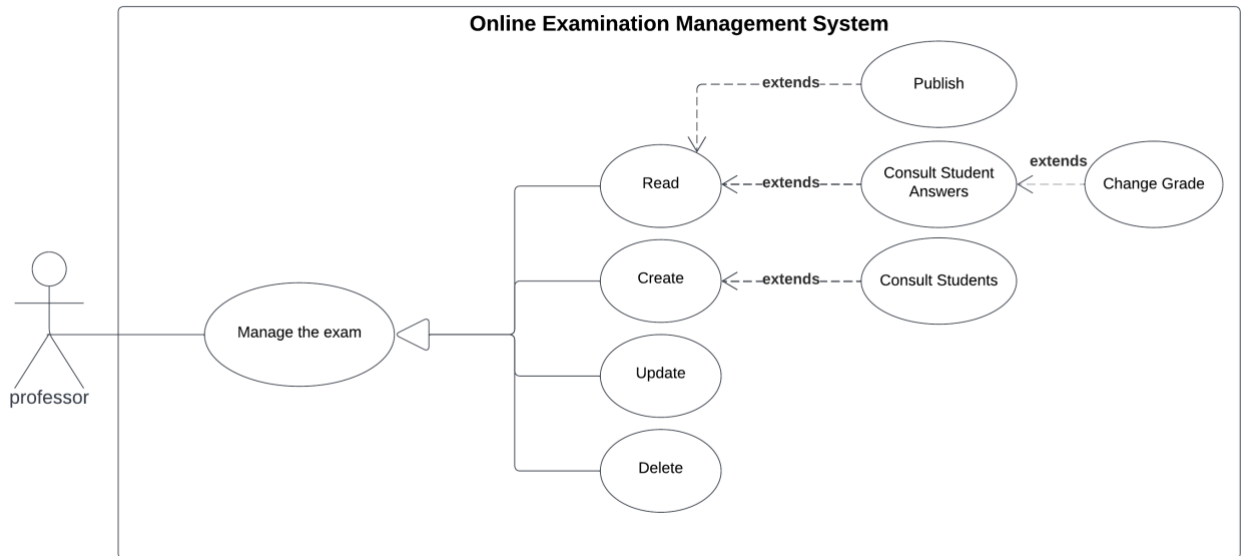


Figure 11: Professor Use Case Diagram

The table 3 describes each use case of professor and its importance as an implementation task:

Use Cases	Description	Importance
Manage the exam	The professor can create, update, delete and consult the exams, where each exam has multiple questions with multiple answers and belongs to a specific specialty, level and module.	High
Consult students	The professor can check the list of students that will take the exam.	Medium
Consult student answers	The professor has the possibility to consult the students answers with their grade as well as the recorded video of each student.	Medium

Publish	The professor can publish the exam after its creation.	High
Change grade	The professor has the possibility to change the student's grade if he discovers unacceptable behavior from the student after watching the video.	Medium

Table 3: Descriptions of Professor Use Cases

### 6.1.3. Admin use case diagram

The figure 12 represents the administrator use case diagram:

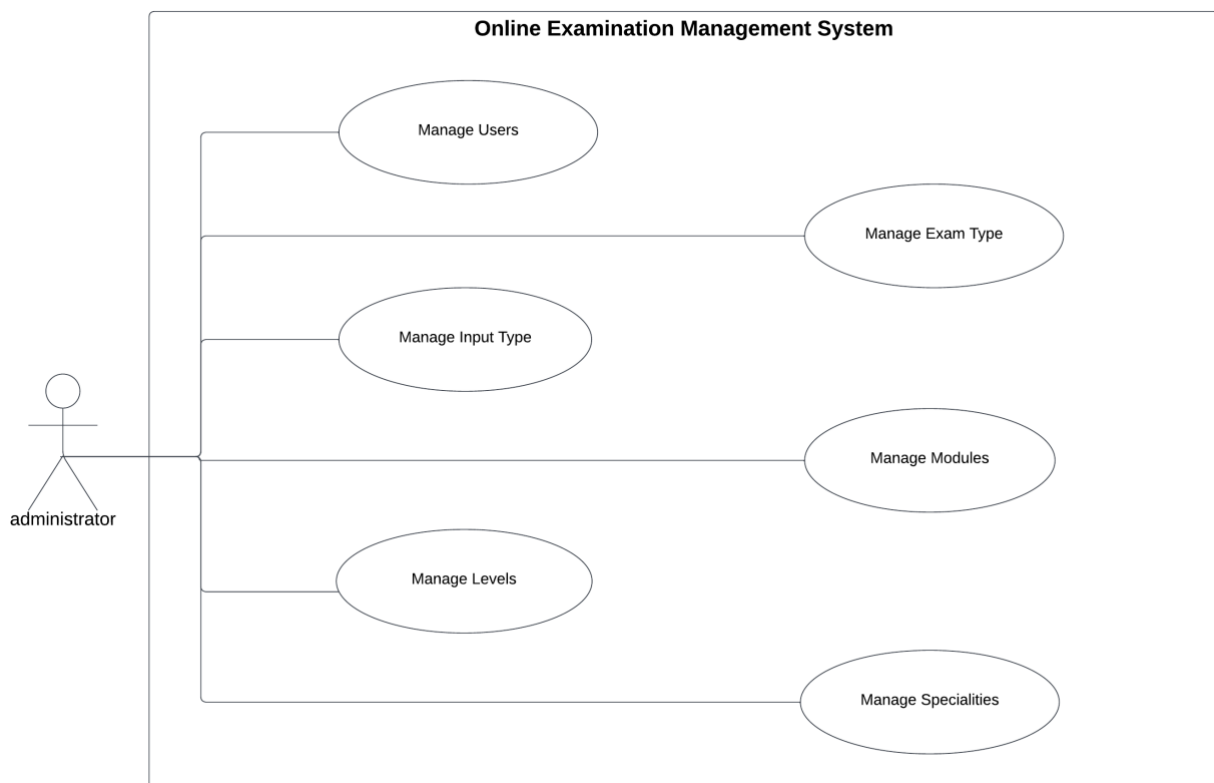


Figure 12: Administrator Use Case Diagram

The table 4 describes each use case of administrator and its importance as an implementation task:

<b>Use Cases</b>	<b>Description</b>	<b>Importance</b>
Manage users	The admin can create, update, delete and consult the users which are the students, professors and admins.	High
Manage Modules	The admin has the possibility to create, delete and consult the modules where each exam belongs to module.	High
Manage Levels	The admin can manage the levels by adding, updating, deleting and consulting them where each level has many modules.	High
Manage Specialties	The admin has the possibility to create, delete and consult the specialties where each specialty has many levels.	High
Manage Exam Types	The admin can create, update, delete and consult the exam type.	High
Manage Input Types	The admin has the possibility to create, update, delete and consult the input type.	High

Table 4: Descriptions of Administrator Use Cases

**NOTE**

The actors of the project must be authenticated to be able to use the application.

**6.2. Sequences diagrams**

The use cases give us with an almost real view of the system. This richness provides us with a global perspective of the application, but in order to observe the interactions between the actors and the system, we need another model that

defines the operation sequencing, this model is the sequences diagram. We will present in this section some diagrams of our application:

### 6.2.1. Authentication sequence diagram

The figure 13, we represent the steps to login using the email and the password, where the student will fill in the credentials needed which is email and password after the login form appear and after the submission the system will check if exist a user with same email first and then check the password second. The system will inform the student in case of the email or password are incorrect.

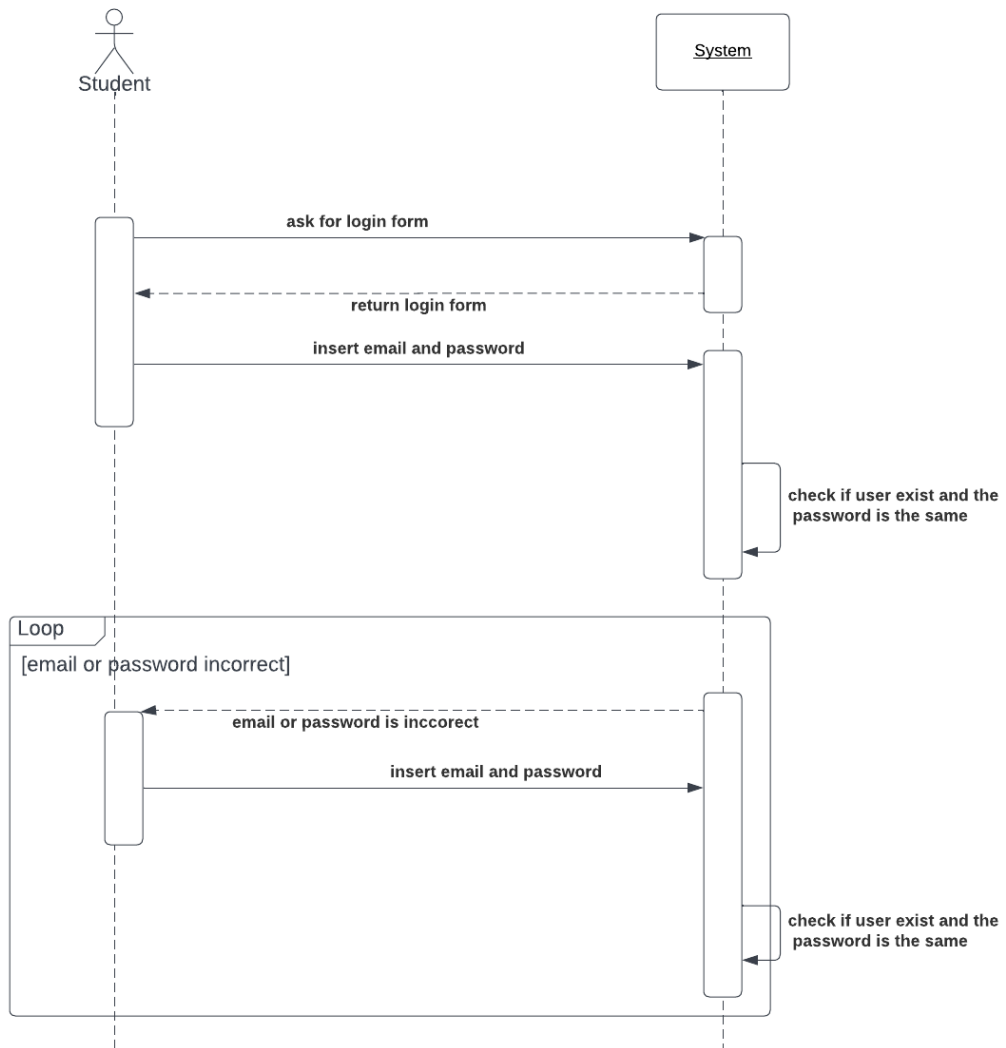


Figure 13: Authentication sequence diagram

## 6.2.2. Student authentication sequence diagram

The figure 14 represent the two login possibilities for the student, where he can login with the email and password as described in figure 13 or with the code QR that he got from this application after the login to his account with the first method of login. For the second possibility of login, the system will check the webcam permission and shows the student an alert in case it couldn't access to the webcam, then the student will present his QR code in front of the camera and the system will check the code and shows him an error in case of the code is invalid, else the student will be connected and the redirected to the scheduled exams.

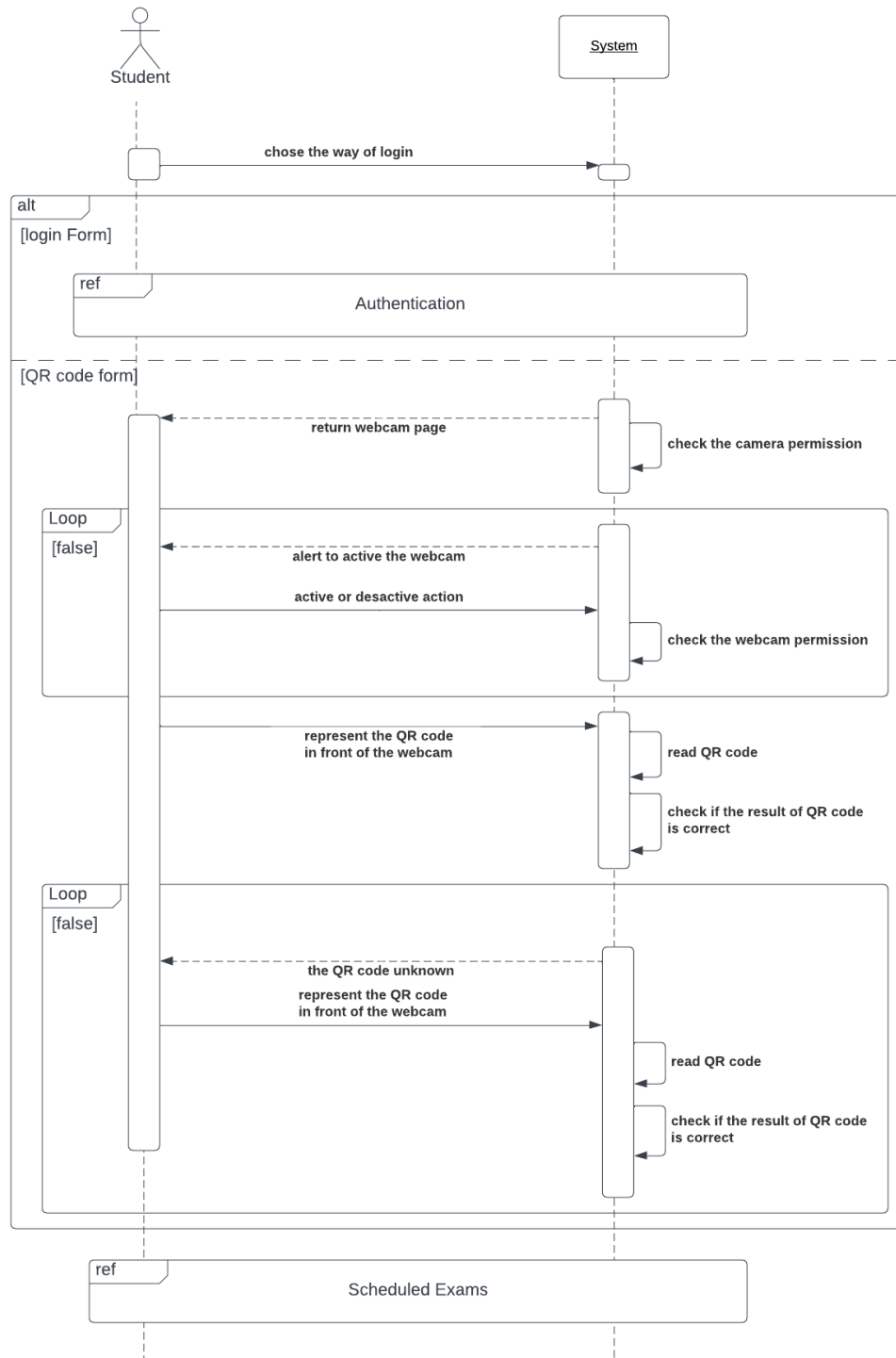


Figure 14: Student authentication sequence diagram

### 6.2.3. Consult taken exam sequence diagram

The figure 15 describe the steps to consult a taken exam, where the student needs to be authenticated to be able to access the taken exams list, then he needs to select a specific exam to consult, and the system will show him the details of the exam he took with his answers as well as the exam correction.

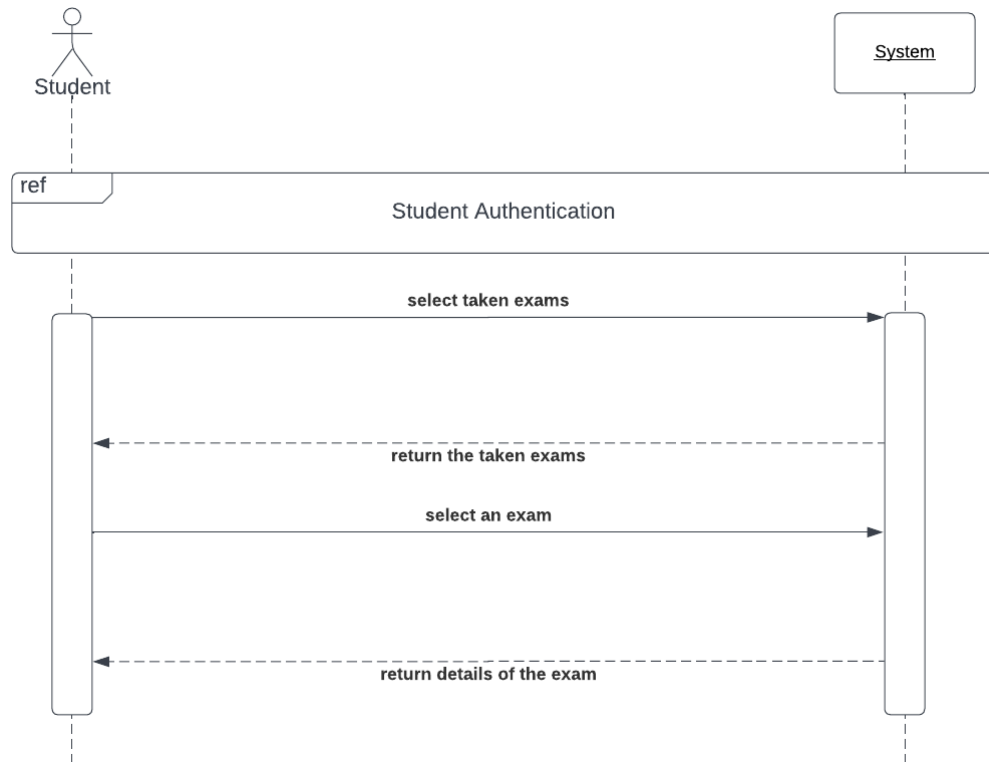


Figure 15: Consult taken exam sequence diagram

### 6.2.4. Take exam sequence diagram

The figure 16 represent the interactions between the actor student and the system and the database, where the student is required to be authenticated to start taking the exam. The student first needs to select the exam to take from scheduled exams list, then the system will return some information that the student needs to know before taking the exam as well as the preparation of the webcam, where the system will check the permission and shows an alert in case it couldn't access to the webcam. After these steps, the student can start taking the exam and the system will start recording the student, also the countdown will start. Finally, when the exam time expires or the student finishes, his answers will be submitted and saved in the database, and he will be redirected to the taken exams list as shown in figure 15.



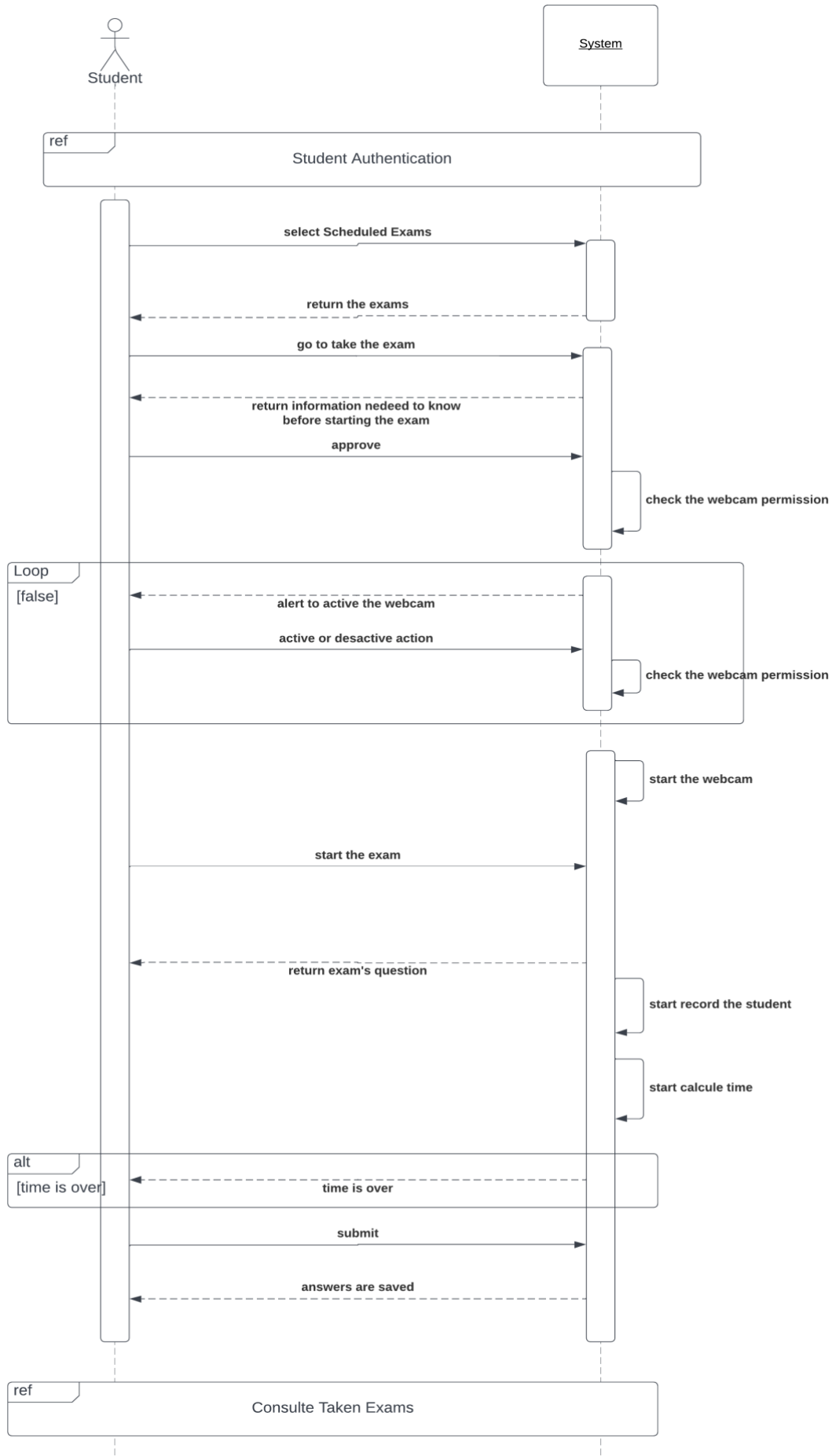


Figure 16: Take exam sequence diagram

### 6.2.5. Add exam sequence diagram

The figure 17 describe how can the professor create an exam in our application. The first step in this the authentication, the professor needs to be authenticated to be able to add an exam, then the professor will fill in the exam information such as the name, type, date, specialty, questions and its answers, etc, after getting the exam form. Also he can check the students list that they will take the exam, and as the last step, the professor will submit the data to be saved in database.

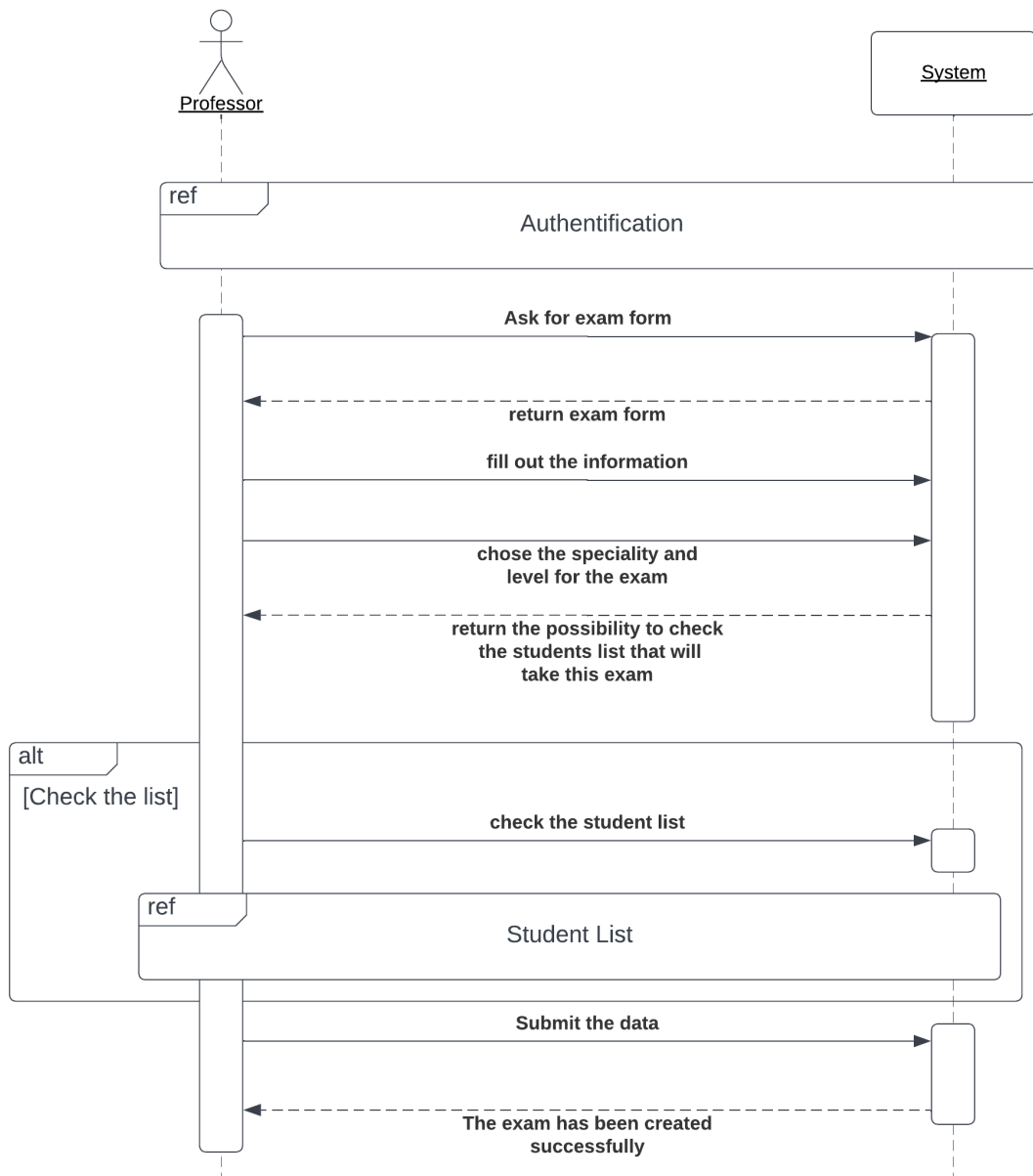


Figure 17: Add exam sequence diagram

### 6.3. Class diagram

The class diagram represents the classes contributed to the application with its attributes and the relationships between them, figure 18 represents a data model of our system:

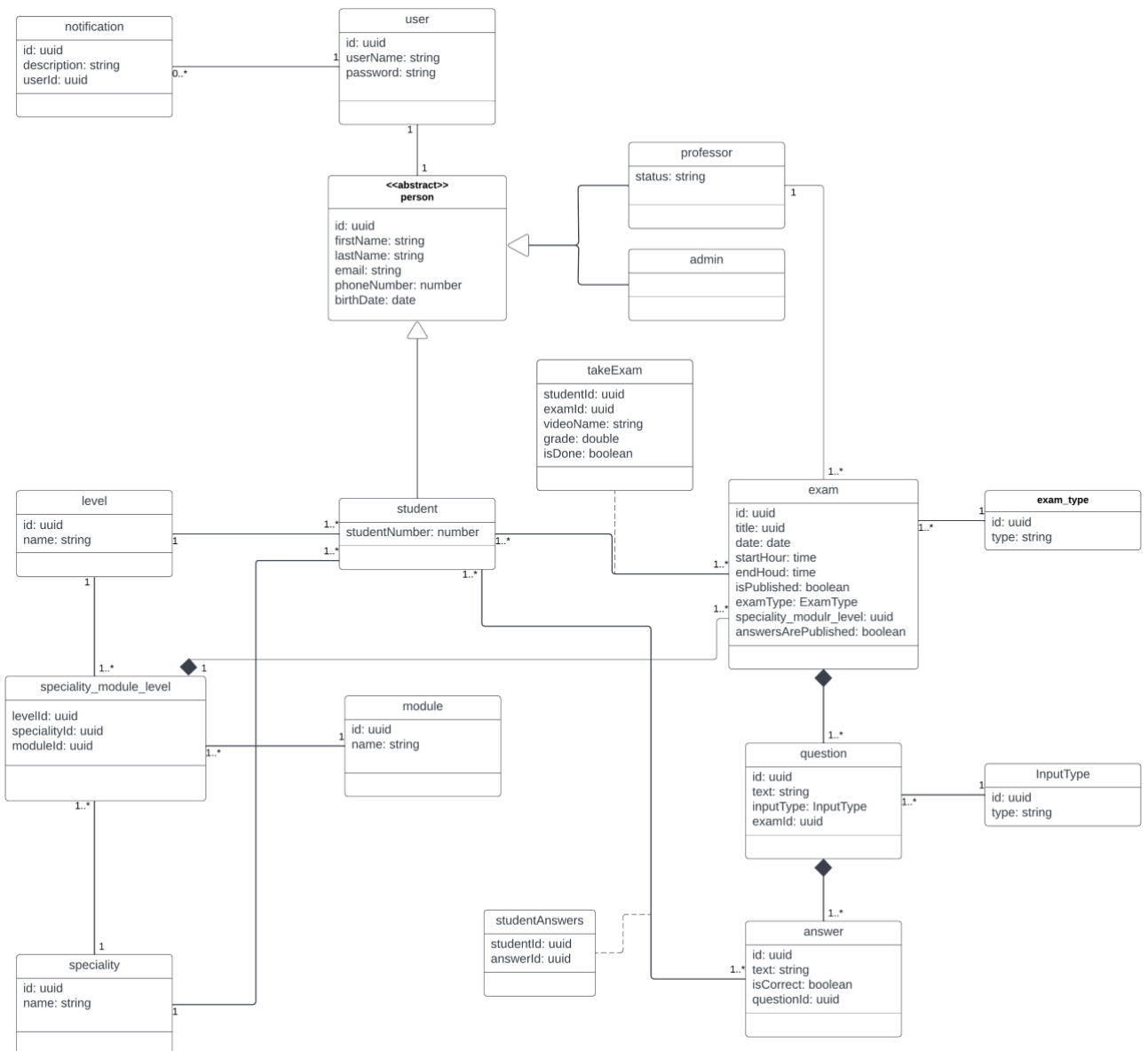


Figure 18: Class diagram

In the diagram above, we have the classes student, admin, professor that represent our 3 users, where they are considered as a person and each one of them has its own functionality to do. Also we have the exam class that is the important class in our system which is characterized with title, date of the exam, start and end time, type of the exam, to which module, level and speciality it belongs, etc. Each exam is composed by multiple questions and each question is composed by several answers.

## **7. Conclusion**

The analysis and design stage is the most significant phase in the development processes, because the outcome guides us in the implementation phase. In the following chapter, we will demonstrate how we used the results of this chapter to realize our UnivExam application.

# **Chapter III**

## Realization and Development

## 1. Introduction

This chapter focuses on the development of our web application, which is based on the analysis and modeling described in the previous chapter. We will present the architecture and the structure our application, the list of technologies used for the implementation. We Finish with some user interfaces and a description of the security of our system.

## 2. UnivExam architecture

Our system's architecture consists of three subsystems that are built as web applications. These subsystems describe the overall idea of creating and taking the exams, by certain principal actors such as professors, students and the administrators for the supervision of the system. All of our web applications are build using JavaScript for reasons of simplicity and its strength as a language in the coding market. One application serves as the REST API that handles requests from and to the frontend applications and with the database. Besides to two frontend applications, one for the students to access and the other one for the administrators and the teachers.

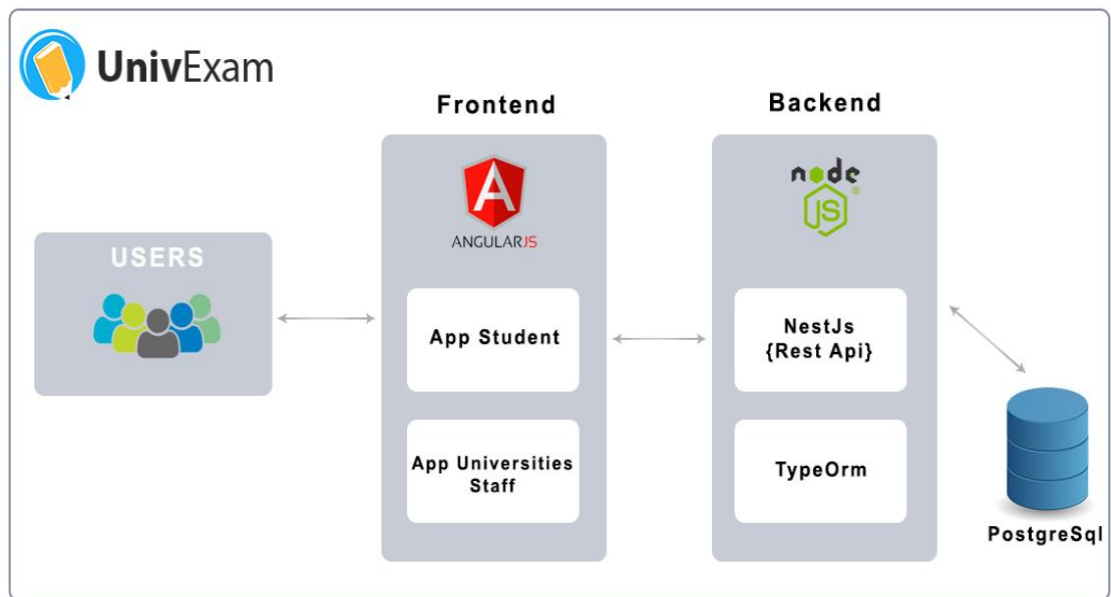


Figure 19: UnivExam application

## 3. UnivExam project structure

Most web-based projects subsystems are developed separately as in each sub web application is developed in a separate repository, but this is an old way of doing things which encouraged us to develop our project's web applications inside one repository using a technology called mono-repo and in our project, we used a well-known tool called Nrwl.

## **Nrwl**

Nrwl as itself is a monorepo tool and is a version-controlled code repository that holds many projects. While these projects may be related, they are often logically independent and run by different teams.

The opposite of the monorepo is multirepo, where each project is held on an entirely separate, version-controlled repository. Multirepos come naturally it's what most of developers do when starting a new project, going from multi to monorepo is a matter of moving all your projects into a single repository and this is why we decided to structure our project in a monorepo manner now instead of developing each of our system's web application separately in its own repository we are going develop all of them in a single repository.

As a plus in monorepo projects such as nrwl allows us to create a shared application that contains code that can be a duplicate code across all the project's applications and share it with all the web applications inside the project.

### **3.1. Web applications**

Our Project's applications are composed of three categories of web applications, two frontend applications and one backend application and one shared library application.

#### **3.1.1. Frontend web applications**

We decided to develop our frontend applications using Angular which is a single page application JavaScript based framework. In a Single Page Application (SPA), all of the application's functions exist in a single HTML page. As users access your application's features, the browser needs to render only the parts that matter to the user, instead of loading a new page, this pattern can significantly improve our application's user experience and this was the biggest criteria that pushed us to choose an SPA framework as Angular instead of classical web application based on vanilla JavaScript and html and CSS on each page of the application. The two frontend applications are:

##### **Front-office web application**

This application is for the students to access their dashboard to view and take their exams.

##### **Back-office web application**

This application is for the administrators of our system and the professors, in which they can manage the users and the exams.

#### **3.1.2. Backend web application (Rest API)**

This is the core web application in which it plays the main role of handling the requests of the users through the front-end applications, and manipulating the data that sits in the database. We decided to work with a well know ORM library "TypeORM" that allows us to connect with any type of database as ms-sql, oracle, mysql, sqlite or postgresql. Finally, we developed our system's api using Nestjs framework that is based on expressjs.

### 3.1.3. Shared Application

This is a shared library application between all the mentioned web applications above, it contains all the duplicate code that can occur between them, as a simple example in all of our project's applications there can be duplicated DTO (Data transfer Object) classes that each application needs, instead of defining the same DTO's in each application we can have them all shared in this application, as well as the interfaces, enums, pipes.

As a side note this type of application is present only before the build of the web applications that depends on them, in another sense after the build, it will not be present as a separate application.

## 3.2. Database

Databases are the essential part of any web application system, it is the system that holds all of the system's data in a structured way depending on what kind of database such as mysql , mongodb , postgresql ...etc, our database had to be a relational sql database to be precise postgresql instead of a nosql database because of our system's data is well structured and has relational aspect to it unlike nosql is better for unstructured data like documents or JSON which is not our case.

## 4. Development environment

While developing the UnivExam web application, we used the following platforms, software, and technologies:

### 4.1. Softwares and platforms used

**Visual studio Code:** is a code editor that supports many programming languages thanks to the power of its extension system. It supports development operations such as debugging and executing tasks. It also works on windows, macOS and linux [16].



**PgAdmin:** is a database administration tool for PostgreSQL and related relational databases like EnterpriseDB EDB Advanced Server. It is available as an online or desktop application [17].



**Photoshop cc:** is software that helps to create, edit and process images with excellent quality [18].



**Trello:** is an online project management tool that divides your projects into lists, and assists the team in managing any form of project or process, as well as tracking duties [19].



**GitHub:** is manage web software development and hosting service, it uses Git version control software. It is mainly used for computer code, and it also makes it easy to combine teamwork together [20].



**Lucidchart:** is a web-based collaborative diagramming and flowchart program with a smart diagram editor that connects teams together. It has multiple diagrams as UML diagrams, network diagrams and others [21].





## 4.2. Technologies used

**AngularJs:** Is a client-side JavaScript framework much powerful to develop a complex application, can be added to HTML to declare dynamic pages, it follows the MVC architecture model [22].



**NodeJs:** is a server-side interpreted language written in JavaScript, it provides JavaScript libraries as needed through the NPM package manager [23].



**NestJs:** is a framework for creating scalable and efficient Node.js server-side applications, built with TypeScript. It uses powerful HTTP Server frameworks such as Express or Fastify. Nest adds a layer of abstraction on top of the standard Node.js frameworks [24].



**PostgreSQL:** is a robust, open source object-relational database system that uses and extends the SQL language, as well as several capabilities that allow it to reliably handle the most complex data workloads [25].



**Sql:** is the acronym for Structured Query Language. It is a standardized programming language for managing relational databases and performing different operations on their contents [26].



**Swagger:** is a suite of open-source tools for designing, building, documenting REST APIs, and describing the structure of the API [27].



**TypeScript:** is a client-side controller and programming language that has added an extra layer to JavaScript such as object-oriented and many other features that make the development of web applications or other projects more elegant [28].



**Bootstrap:** is a frontend framework and a collection of tools that helps to create web applications easily, it contains a set of HTML / CSS and JavaScript [29].



**TypeOrm:** is an Object Relational Mapper framework built in TypeScript that runs in node.js and it supports a variety of databases, including MySQL, PostgreSQL, MariaDB and others. Also it is a simple ORM for building new apps that link to databases [30].



**ngx-qrcode:** is an Angular 6+ library to generate QR Codes. This library has two principle parameters: Value, which is the data to generate code with and ErrorCorrectionLevel that indicate the qr correction level [31].

**zxing-ngx-scanner:** is an Angular library to read QR codes. This library offers 6 events, the most important are: ScanSuccess, issued when a scan is successful performed, PermissionResponse, issued when the user change the webcam permission and it will return true if the webcam is activated else return false [32].

## 5. Production environment

After all the steps mentioned in this study such as the conception and the realization of the project, we needed to put our system out of the development environment and publish it in a real production environment, and for that we needed to prepare some of essentials key components:

### Domain Name

Every web application has to be linked with a Domain name, for the purposes of our system we decided to use an already existing domain name of ours that we already acquired quite a time ago: alaathr.com, after that we need a Remote Server with a static IP address so that our domain name can point to it.

### Remote Server

We chose to work with GCP “google cloud platform”, we create a Linux virtual machine with a static IP address, after that we set a ssh port so that we can have a shell access into the server and execute Linux commands remotely.

Our system is composed of two frontend applications and we wanted to serve each in a separate subdomain so we created two subdomains for each:

univ-exam.alaathr.com: The Student front office applications will be served through this subdomain.

admin.univ-exam.alaathr.com: The backoffice application will be served through this subdomain.

Both of the above subdomains must point to the same create Server’s static IP so that we list http requests via nginx and serve each web application according the requested URL.

### Nginx

In terms of webservers there are two dominant technologies in the market, Apache and Nginx. We decided to work with Nginx because it’s the most used one for its simplicity and ease of use, and it has a big community behind it as well as it is far superior than Apache in terms of speed by a factor of almost 2.5. Besides, it consumes less memory which makes it resource friendly and this makes it a good choice for our project’s application in the long run if it gets larger.

Now that all the essential key components for the deployment of our project live in the internet are met, all we have to do now is to clone our project from GitHub into the remote server, then build all the applications. Also, with the necessary configuration of nginx, it will be ready to serve each build application separately with its respective subdomain.

## 6. UnivExam Graphical interfaces

In this section we will present the main interfaces of our web application, where each front-end application has its own design and schema that is easy-to-use for all type of users:

### 6.1. Front-office interfaces

#### 6.1.1. Authentication interface

The student has two ways to access his account, which gives him the right to perform his functionalities.

##### QR Code interface

The figure 20 represent the authentication with QR code, it is displayed when the student wants to access his account. He needs to present the QR code that he got from our application in front of the webcam, and the picture will appear in the left side. As well as he can switch to login with his credentials. In case of the code is invalid, the system will inform the student as shown in figure 21.

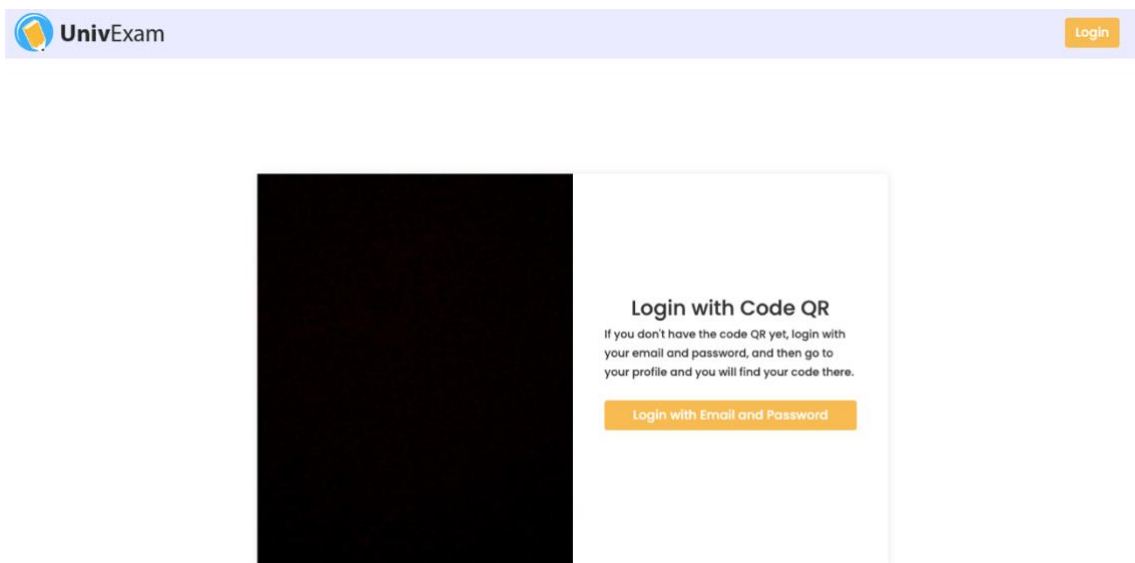


Figure 20: Login with QR code interface

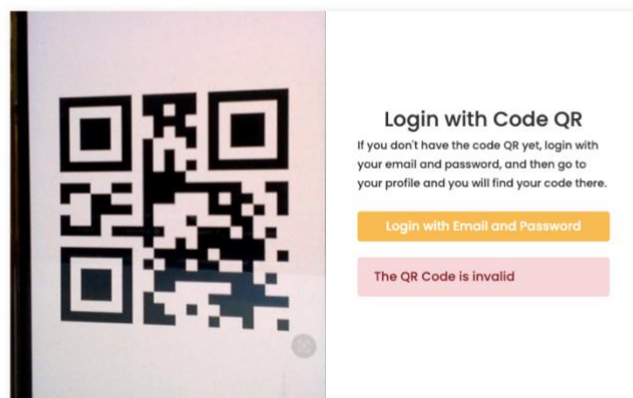


Figure 21: Invalid QR code interface

### Credentials interface

The interface 22 represent the authentication with credentials, does mean login with email and password, it is displayed as well when the student wants to connect to access his account. The student needs to fill in his email and password and then login. In addition, he has the possibility to switch to login with the QR code.

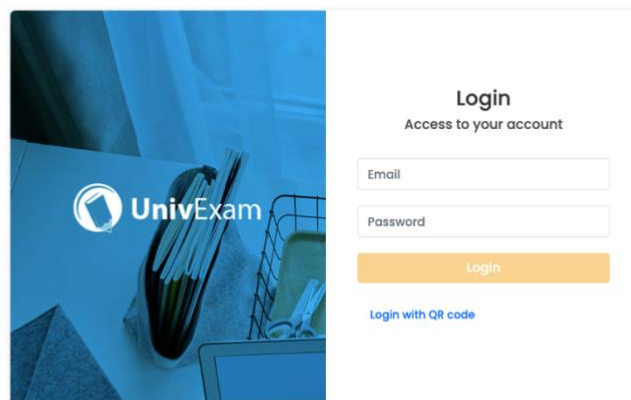


Figure 22: Authentication with credentials interface

### 6.1.2. Take exam interface

The figures 23 and 24 represent take exam interface, where it will appear to the student when he starts the exam. The student will be in a fullscreen mode as the countdown begins, and he will find a set of questions with the point of each one of them and also its proposed answers to choose. After finishing all the steps, the student submits his answers with his recorded video.

<b>Full Name:</b> AMEL MAHI <b>Student Number:</b> 227580186875	<b>3</b> TOTAL QUESTIONS	<b>00:12:52</b> TIME LEFT
--	-----------------------------	------------------------------

**Questions**

1 question: 6.5 pts

2 question: 6.5 pts

3 question: 7 pts

**#1** Which of the following functions are NOT everywhere continuous ?

$f(x) = (x + 3)^4$

$f(x) = (x^2 - 4)/(x + 2)$

$f(x) = 1066$

$f(x) = mx + b$

[Next](#)

Figure 23: Take exam interface 1

<b>Full Name:</b> AMEL MAHI <b>Student Number:</b> 227580186875	<b>3</b> TOTAL QUESTIONS	<b>00:16:41</b> TIME LEFT
--	-----------------------------	------------------------------

**Questions**

1 question: 6.5 pts

2 question: 6.5 pts

3 question: 7 pts

**#3** Find  $\lim_{x \rightarrow 2} f(x)$ :  $f(x) = 1776$  ?

is:  $+\infty$

is: 1770

is:  $-\infty$

is: Does not exist!

is: None of the above

[Back](#)
[Submit](#)

Figure 24: Take exam interface 2

## 6.2. Back-office interfaces

### 6.2.1. Professor interfaces

In this section we will represent and describe the professor's interfaces as below:

### 6.2.1.1. Create exam interface

The figure 25, represent and shows the steps of creating the exam. The professor needs to fill in the exam information and questions with its details as the input type, the score and answers as shown in figure 26, as well as he has the possibility to delete, update both of questions and answers. Besides to that, the professor can check the list of students who will take the exam after choosing the specialty and the level, also the possibility to publish the exam or not.

**Add Exam**

Exam Information

Exam Title: Data analysis I

Exam Type: Exam

Speciality: Quantitative Methods Of Decision Making Support

Level: B3

Module: Data Analysis

Exam Date: 30/06/2022

Start Time: 08:00

End Time: 10:00

Publish the exam?

[Check student List](#)

**Add Questions**

Questions Information

[Add Questions +](#)

Question Title: Which of the following is true about hypothesis testing

Input Type: Checkbox

Question Score: 0,25

Add Answers: [Add Answers +](#)

Question	Input Type	Question Score	Answers	Action
Data analysis is a process of	radio	4	<a href="#">view answers</a>	<a href="#">edit</a> <a href="#">delete</a>
Data analysis is defined by the statication	checkbox	2.5	<a href="#">view answers</a>	<a href="#">edit</a> <a href="#">delete</a>

[Add Exam](#)

Figure 25: Create exam interface

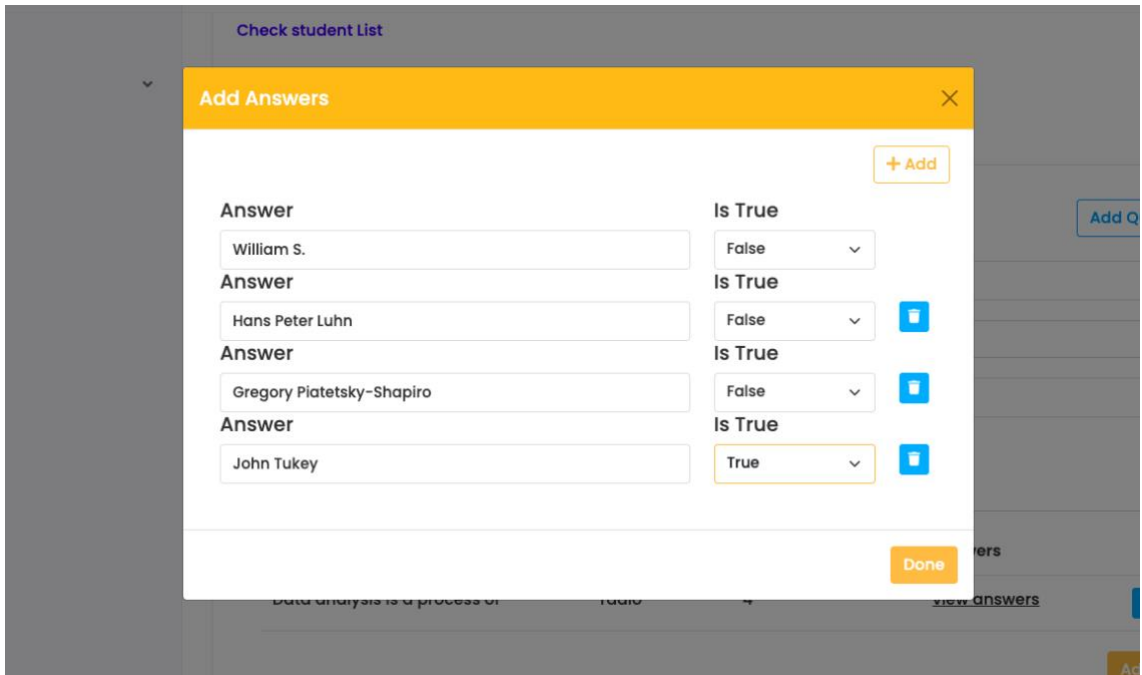


Figure 26: Create answers interface

## 6.2.2. Admin interfaces

In this section we will represent and describe the administrator's interfaces:

### 6.2.2.1. Create student interface

To create a student, the administrator needs to fill in all the necessary information of the student including the email, password, first name, last name, phone number, student number and finally the speciality and the level as presented in the figure 27.

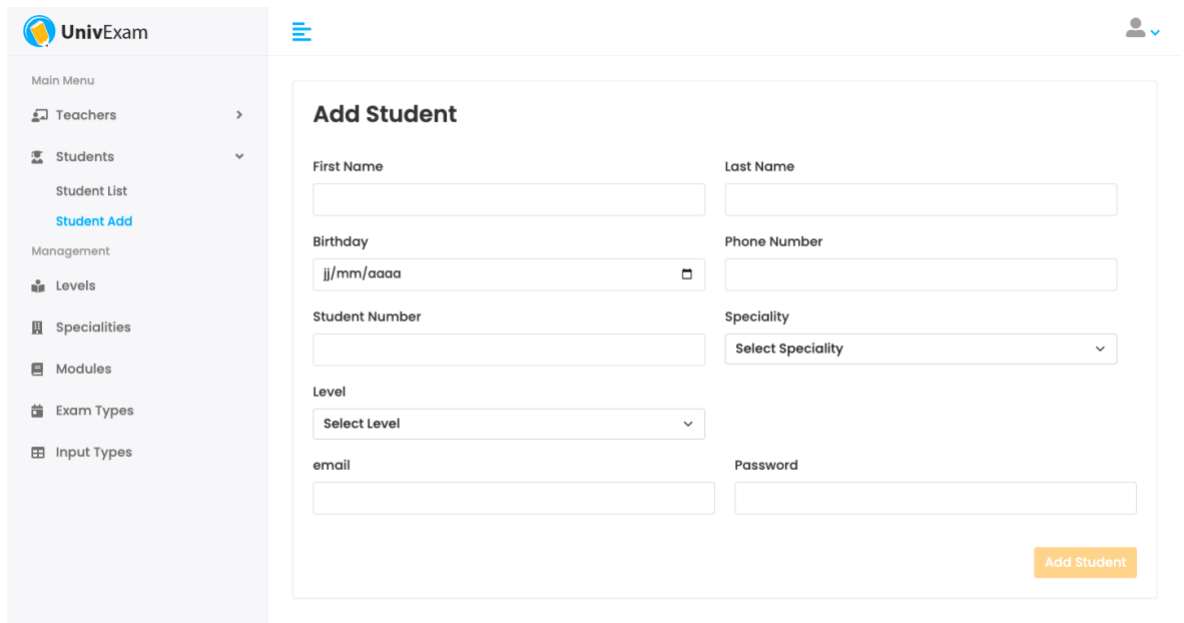


Figure 27 : Create student interface

### 6.2.2.2. Specialties list interface

The figure 28 represent the list of specialties with their details and for each specialty there is a button that helps to delete it. In addition to an add button that helps in creating a new specialty.

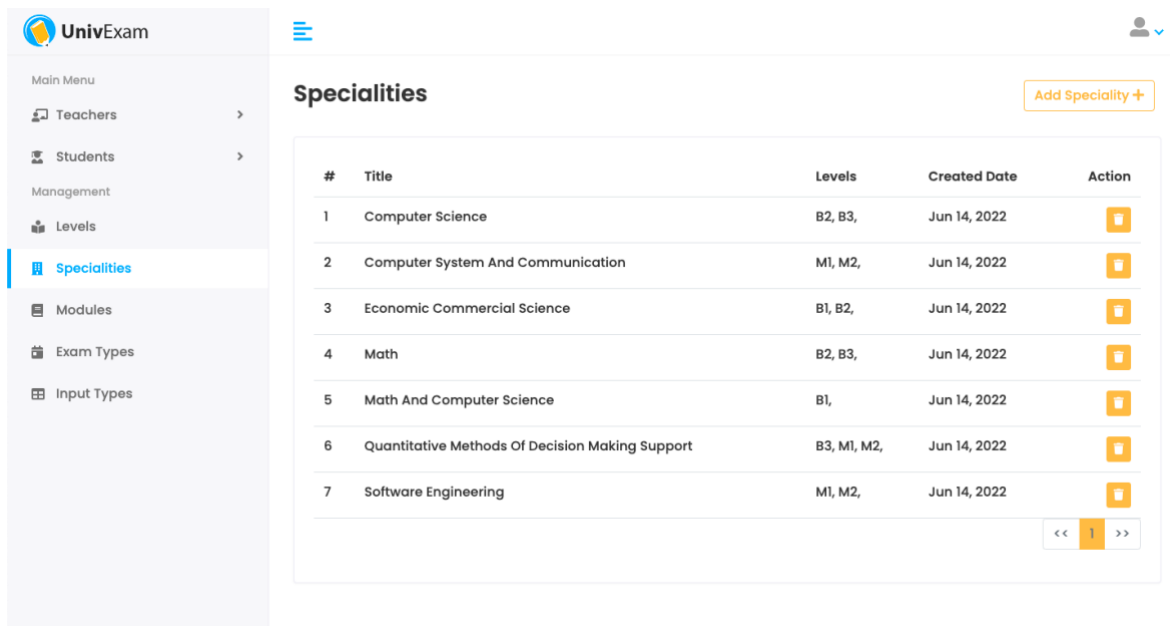


Figure 28: Specialty list interface

## 7. System Security

The security plays the most important role in systems implementation, this stage is based on the protection of the system against any violation, attacks or data theft that could happen during the utilization of the system in the production. For that we decided in our system to adopt a rigorous security policy and use certain security mechanisms to make our application more secure and make sure that our users are safe, and we can mention them as follow:

**Password strength:** The framework nestJs validates password through call-validator library to boost strength of passwords.

**Database password storage:** In our system we don't allow passwords to be stored as plain text, but we store them encrypted using BcryptJs library where it encrypts the password using a salt that is generated by the library itself.

**SQL injections:** NestJs is integrated with an ORM that helps protect the system against this risk.

**User Guards:** Our application has 3 types of users, the student, the professor and the admin and each one of them has his specific functionalities to do and his specific pages or url to navigate. For that we used guards in our system to limit each user's access to responsibilities other than his own, and protect each user.



## 8. Quality Features

In this section represent how our UnivExam application can fight and prevent cheating methods by students during the exams and ensuring the credibility of exams and conducting them in a safe manner. Our system will prevent cheating as follow:

**Exam proctoring:** our web application has the possibility to record the students by video and audio during the exam, that will allow the professor to check the student identity and if there was any external help.

**Fullscreen mode:** The UnivExam application transfer the students to fullscreen mode as a first step of taking the exam, also the system will detect when the students exit from fullscreen mode. That will avoid the easy navigation cheating methods.

## 9. Conclusion

In this final chapter, we have given an overview of all the tools used for the development and production of our UnivExam web application, we have also described the security mechanisms used to protect our system from any attacks, and as well as we have presented certain main interfaces that match the requirements mentioned in the specification phase.

# General Conclusion

## 1. Conclusion

The work presented in this thesis is organized around the design and the implementation of an e-assessment system in the form of a web application, with the goal of establishing a virtual exam center on this web application.

This project allowed us to extremely contribute to the amelioration of our knowledge in an interesting topic, as well as explore new technologies and learn new powerful and popular programming languages that will help us in the professional world.

We started our study by understanding the general context of our application and conducting research on existing similar applications in order to identify different requirements and add certain functionalities to make the application richer and to ensure the smooth operation of the various managements.

Our next step was devoted to the application's development. This process covers the analysis and the design of the functions that our application has, which includes database modeling and application development.

We finished our work by the implementation and the production of our application. In this part we implemented web application using the best development tools to optimize the response time and we used a smooth architecture that allows other developers to continue developing more.

## 2. Perspectives

As a perspective of our work, we intend to add other functionalities, as:

- Integration of the artificial intelligence to increase the unacceptable behavior detection that indicates cheating by the student as use of phone, tablets, constantly looking at areas other than the computer screen and the presence of persons other than the student.
- Integration of video streaming so that the professor or other administrative person can monitor students live.
- Demo exams, that allows student to prepare to their exams as well as to be familiar with the platform.
- Providing all exam types of the last years.
- Providing other Question types such as long answers type, upload files, fill in the blank, coding type, etc.
- Integration of applications that helps the student to develop his answers such as conception tools (modelio), Excel, etc.
- Adding students to the system using an excel file.
- Adding chat that allows student to communicate with professors in case of problem or misunderstanding in the exam.
- Statistics for both admin and professor.

## References

- [1] Aimee Chanthadavong, Senior Journalist. Digital exams to help Monash University save AU\$7 million. ZDNet, Aug 21, 2019: consulted: 06-06-2022: <https://www.zdnet.com/article/digital-exams-to-help-monash-university-save-au7-million/>
- [2] Bhuvi Kathpalia. Online Examination System: A Guide for Universities to Shift Online, Mettl and Mercer.
- [3] Piyush Ray. Mercer Mettl Assessments Overview, G2: consulted: 17-06-2022: <https://www.g2.com/products/mercer-mettl-assessments/reviews>
- [4] Mercer Mettl Online Assessment, SoftwareSuggest: consulted: 17-06-2022: <https://www.softwaresuggest.com/mettl-online-assessment>
- [5] Splashgain. Eklavya Software Overview, Techjockey: consulted: 17-06-2022: <https://www.techjockey.com/detail/eklavya>
- [6] How to Read Instructions Related to Online Exam, Eklavya: consulted: 17-06-2022: <https://www.eklavya.com/blog/read-instructions-related-online-exam/>
- [7] Sayan Kumar Pal. Software Engineering | Classical Waterfall Model, GeeksforGeeks, Mar 02, 2022: consulted: 07-06-2022: <https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/>
- [8] Trello: consulted: 07-06-2022: <https://www.indexventures.com/companies/trello/>
- [9] Ladan Jafari. What Are Functional Requirements? Types and Examples, Wina Talent, May 12, 2020: consulted: 11-06-2022: <https://winatalent.com/blog/2020/05/what-are-functional-requirements-types-and-examples/>
- [10] Ladan Jafari. What Are Non-Functional Requirements? Types and Examples, Wina Talent, May 31, 2020: consulted: 11-06-2022: <https://winatalent.com/blog/2020/05/what-are-non-functional-requirements-types-and-examples/>
- [11] Alligator.io. Angular MVC - A Primer, DigitalOcean, May 29, 2019: consulted: 22-06-2022: <https://www.digitalocean.com/community/tutorials/angular-angular-mvc-primer>
- [12] Yogi Hosting. Controllers in ASP.NET Core, April 8, 2022: consulted: 22-06-2022: <https://www.yogihosting.com/aspnet-core-controllers/>
- [13] Vivek Kumar. 5 Key Features of AngularJs that make it the best for web development, Matridtech: consulted : 22-06-2022 : <https://www.matridtech.net/5-key-features-of-angularjs-that-make-it-the-best-for-web-development/>
- [14] Yassamine Seladji. Design Patterns 2, university of Tlemcen, January 11, 2021. P-20.

- [15] Refactoring. Strategy: consulted: 22-06-2022: <https://refactoring.guru/design-patterns/strategy>
- [16] Visual studio code : consulted: 20-06-2022: <https://code.visualstudio.com/>
- [17] PgAdmin : consulted: 20-06-2022: <https://www.pgadmin.org/faq/#1>
- [18] Katie Terrell Hanna. Photoshop, Techtarget : consulted: 20-06-2022 : <https://www.techtarget.com/whatis/definition/Photoshop>
- [19] Trello: consulted: 20-06-2022: <https://trello.com/>
- [20] GitHub: consulted: 20-06-2022: <https://docs.github.com/en/get-started/quickstart/hello-world>
- [21] Lucidchart: consulted: 20-06-2022: <https://www.lucidchart.com>
- [22] AngularJs: consulted: 20-06-2022: <https://angularjs.org/>
- [23] NodeJs: consulted: 20-06-2022: <https://nodejs.dev/learn>
- [24] Selleo Developers. Why Choose NestJS As Your Backend Framework?, Selleo, oct 31, 2020 : consulted : 20-06-2022: <https://selleo.com/blog/why-choose-nest-js-as-your-backend-framework>
- [25] PostgreSQL: consulted: 20-06-2022: <https://www.postgresql.org/about/>
- [26] Peter Loshin. Structured Query Language (SQL), TechTarget, February, 2022: consulted: 20-06-2022: <https://www.techtarget.com/searchdatamanagement/definition/SQL>
- [27] Swagger: consulted: 20-06-2022: <https://swagger.io/docs/specification/about/>
- [28] Remo H. Jansen, Vilic Vane, Ivo Gabe de Wolff. TypeScript:Modern JavaScript Development , Publishing Ltd, 2016.
- [29] Tomislav Bacinger. What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How, Toptal: consulted: 20-06-2022: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>
- [30] TypeORM - Quick Guide: Consulted: 20-06-2022: [https://www.tutorialspoint.com/typeorm/typeorm\\_quick\\_guide.htm#:~:text=TypeORM%20framework%20is%20an%20Object,SQL%2C%20PostgreSQL%2C%20etc.](https://www.tutorialspoint.com/typeorm/typeorm_quick_guide.htm#:~:text=TypeORM%20framework%20is%20an%20Object,SQL%2C%20PostgreSQL%2C%20etc.)
- [31] Ngx-qrcode: consulted: 20-06-2022: <https://www.npmjs.com/package/@techiediaris/ngx-qrcode>
- [32] Ngx-scanner : consulted : 20-06-2022 : <https://github.com/zxing-js/ngx-scanner/wiki/Component-API#api>

## Abstract

Nowadays, online examination systems play a significant role in every higher education institution in all the world due to the several issues related with the manual procedure of conducting tests for students and it became more difficult with the emergence of pandemics like COVID 19.

Our thesis is dedicated to build a virtual exam centers platform called "UnivExam", in order to facilitate the procedures for taking and managing exams through an online environment, which is much faster and easier than a manual examination system and also to ensure the health and the safety of students and university staff.

**Keywords:** e-assessment, exams, university, pandemics, web application.

## Résumé

De nos jours, les systèmes d'examen en ligne jouent un rôle important dans tous les établissements d'enseignement supérieur du monde entier en raison des nombreux problèmes liés à la procédure manuelle de réalisation des tests pour les étudiants et cela est devenu plus difficile avec l'émergence de pandémies comme COVID 19.

Notre thèse est dédiée à la construction d'une plate-forme de centres d'examens virtuels appelée "UnivExam", afin de faciliter les procédures de passage et de gestion des examens via un environnement en ligne, beaucoup plus rapide et plus simple qu'un système d'examen manuel et également pour assurer la santé et la sécurité des étudiants et du personnel universitaire.

**Mots clés:** e-examen, examens, université, pandémies, application web.

## Αφηρημένη

Στις μέρες μας, τα διαδικτυακά εξεταστικά συστήματα διαδραματίζουν σημαντικό ρόλο σε κάθε ίδρυμα τριτοβάθμιας εκπαίδευσης σε όλο τον κόσμο λόγω των πολλών θεμάτων που σχετίζονται με τη χειρωνακτική διαδικασία διεξαγωγής τεστ για φοιτητές και έγινε πιο δύσκολη με την εμφάνιση πανδημιών όπως το COVID 19.

Η διατριβή μας είναι αφιερωμένη στη δημιουργία μιας πλατφόρμας εικονικών εξεταστικών κέντρων που ονομάζεται "UnivExam", προκειμένου να διευκολυνθούν οι διαδικασίες για τη διεξαγωγή και τη διαχείριση των εξετάσεων μέσω ενός διαδικτυακού περιβάλλοντος, το οποίο είναι πολύ πιο γρήγορο και ευκολότερο από ένα σύστημα χειροκίνητων εξετάσεων, καθώς και για τη διασφάλιση της υγείας και την ασφάλεια των φοιτητών και του προσωπικού του πανεπιστημίου.

**Λέξεις κλειδιά:** ηλεκτρονική εξέταση, εξετάσεις, πανεπιστήμιο, πανδημίες, διαδικτυακή εφαρμογή.

## ملخص

في الوقت الحاضر، تلعب أنظمة اجراء الامتحان عبر الإنترنت دورًا مهمًا في كل مؤسسة للتعليم العالي في جميع أنحاء العالم نظرًا للعديد من المشكلات المتعلقة بالإجراءات اليدوية لإجراء الاختبارات للطلاب، وأصبح الأمر أكثر صعوبة مع ظهور أوبئة مثل COVID 19.

رسالتنا مخصصة لبناء منصة مراكز امتحانات افتراضية تسمى "UnivExam"، من أجل تسهيل إجراءات إجراء الاختبارات وإدارتها من خلال بيئة عبر الإنترنت، وهي أسرع وأسهل بكثير من نظام الفحص اليدوي وأيضًا لضمان الصحة و سلامة الطلاب وموظفي الجامعة. الكلمات المفتاحية: امتحان إلكتروني، اختبارات، جامعة، أوبئة، تطبيق ويب.