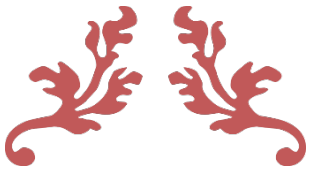

TeacherPiker

In search of knowledge

Project submitted in the context of the course
I3301 Software Engineering

Prepared by
Mohammad Ayoub 93274
Mariam Sayed 92044
Zahraa Al-Hek 107293

LEBANESE UNIVERSITY
FACULTY OF SCIENCES I
DEPARTMENT OF COMPUTER SCIENCES
2021-2022

Article I. Acknowledgments

We would like to express our thanks and gratitude to Dr. Abed Safadi, who, without his guidance and advice, would not have been able to complete this project successfully.

Thank you for your dedication and hard work.

Article II. Abstract

TeacherPicker is an app for anyone who wants to find the right teacher for their subject, as well as for any teacher looking for a job.

There will be space for posts that user can interact with, as an announcement that this person is ready to start a new job according to the conditions he determines if he is a teacher, or an announcement to request a teacher with the specifications he determines if he is a student or a private school. The teacher will have a profile with detailed information, which will make it easy to find on the search page using the filter options. There will be messaging, calls and meetings service to facilitate communication between users.

Through this application, we will contribute to alleviating the unemployment rate by finding jobs for these professors, in addition to facilitating the process of searching for knowledge for students.

Article III. Table of Contents

Article IV. Table of Figures

Article V. Table of Tables

Chapter I - Introduction

Recently, due to the quarantine of Covid-19, most schools have resorted to online education, which made the task of learning more difficult for the student and his need for a private tutor appeared more and parents have been struggling to find these teachers. On the other hand, there are many teachers who have graduated from universities and are still unemployed.

This chapter introduces the application being modeled and the underlying business domain.

I.1. The business domain

Technology has changed the possibilities within teaching and learning. Classes, which prior to the digital era were restricted to lectures, talks, and physical objects, no longer must be designed in that manner. Teachers and students now have a digital toolbox ranging from engaging devices to teach the students to online courses and digital textbooks. Digital education is the term used to refer to all online educational practices.

Projections show the e-learning market worldwide is forecast to surpass 243 billion dollars by 2022. In 2016, the self-paced e-learning product market amounted to 46.67 billion dollars and is projected to decrease to 33.5 billion U.S. dollars in 2021. A considerable share of faculty worldwide has shown willingness to support less traditional and digital education models. About 65 percent of faculty supports the use of open educational resources in teaching, and 63 percent showed support for the competency-based education system. Back Students have also shown willingness to embrace digital learning technologies (DLT) and practices. About 92 percent of students worldwide are interested in personalized support and information on degree progress. They also showed interest in other academic performance analytics.

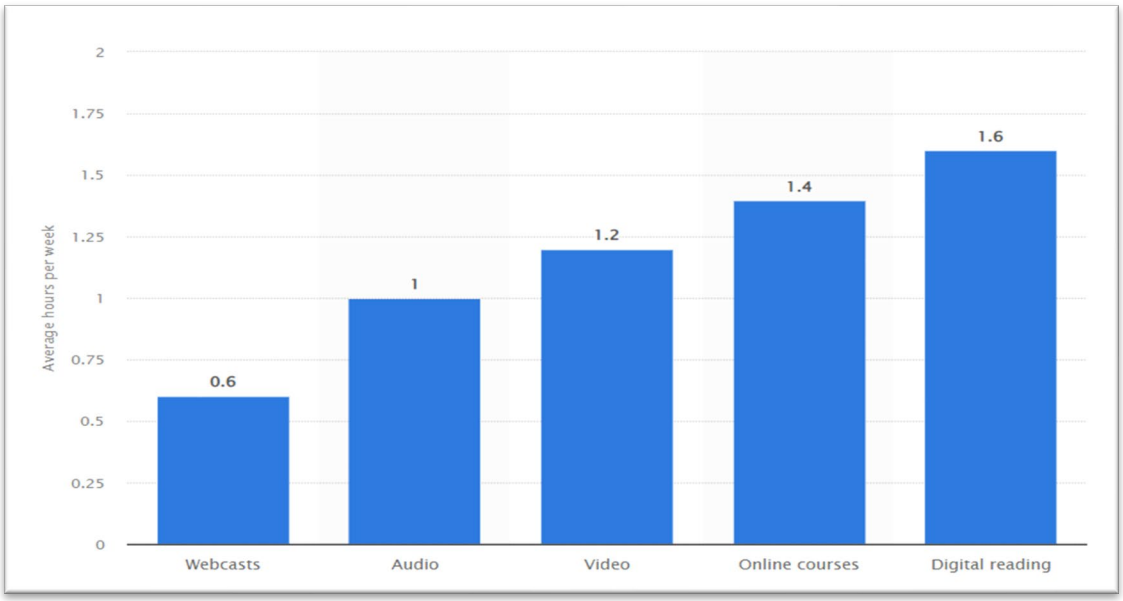

Figure 1. Online learning time per week by method worldwide 2018

Students are also interested on the use of different devices in the learning process. As of April 2015, about 56 percent of students worldwide wanted their instructors to enable them to use their laptop more in a learning context. Students already use this type of the device for school related activities. In the United States, about 56 percent of students stated using a laptop or desktop computer in the classroom on a weekly basis. As of 2016, the most common digital learning materials used in weekly in PreK-12 classrooms in the United States were online educational videos, educational apps, or software, as well as research websites.

Students are turning to online courses as well:

-In 2015, 49 percent of students stated that they had taken an online course in the last 12 months. This share is slightly higher than 2013 and 2014 figures, when 46 percent and 47 percent of the students stated taking this type of course.

-Despite this growth, 73 percent of the students in the world are not aware of massive open online courses (MOOC). As of April 2015, the share of students that took a MOOC in the past year was still low nine percent. From those global students who have interacted with MOOC and CBE, about 11 percent earned a competency-based badge, and about 19 percent said they would use competency-based digital badge on their resume.

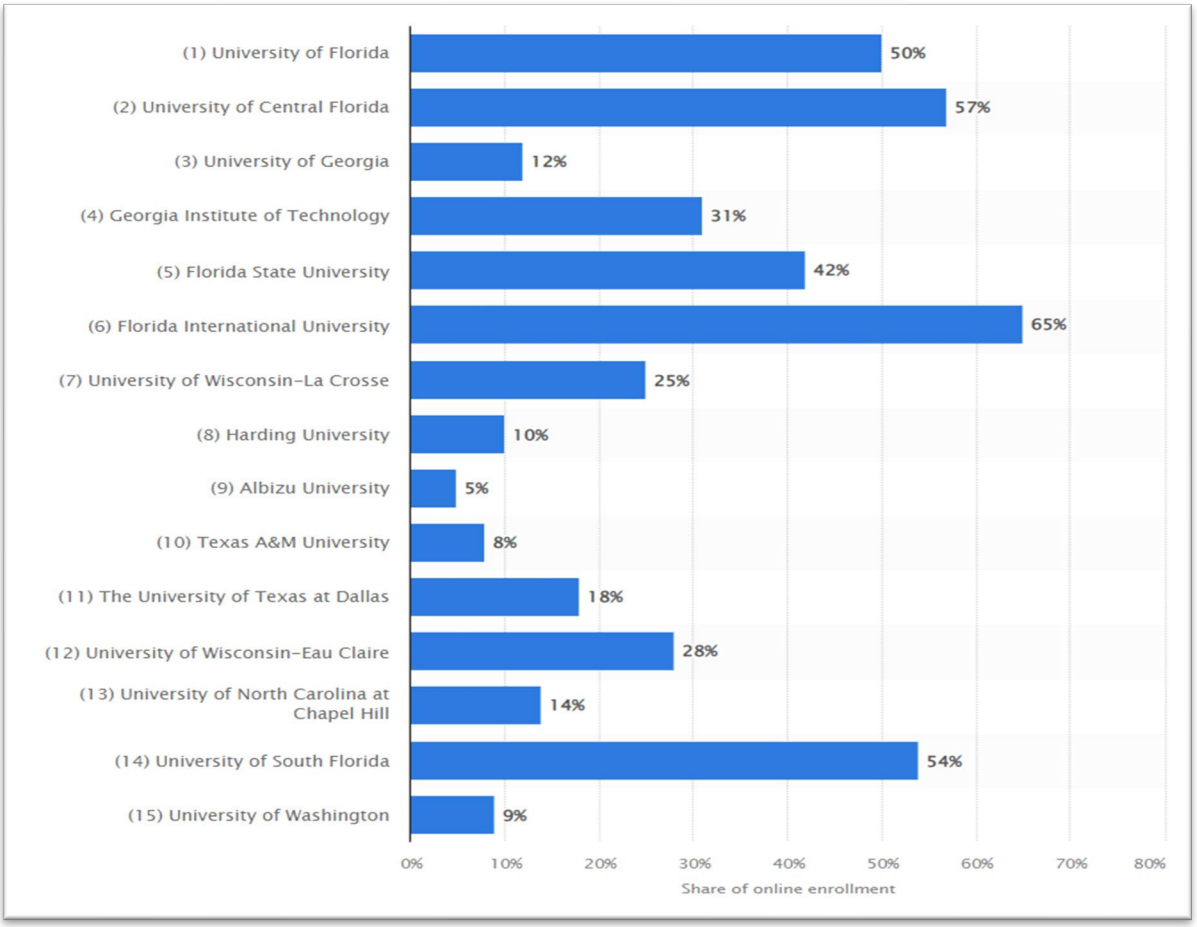

Figure 2. U.S. distance learning institutions, by share of online enrollment 2021

I.2. About the modeled of application

TeacherPicker application will be used by students to be able to find a teacher to help them overcome learning difficulties or by parents to find a teacher for their children.
It will also be used by teachers who have not had the opportunity to find a job or even teachers who want to work extra hours.
In addition, private schools can use this application to meet their teacher shortage needs by selecting the expert teachers they want.

Of course, the app is not only used for those already mentioned, but anyone who needs a instructor can use it.

2.1. Users Persona

Teacher Picker app is available to everyone from teachers, students, seniors and even the youngest user in the education sector.

Guests can find out all the information about the classes by checking the posts.

Simple user in addition to what guests do can follow anyone who has an account in this app and can send request to any teacher who can help them, create posts and stories as well as interact with other posts and use messenger to communicate with other users.

Teachers can apply everything they do in class, such as create a quiz or assignment, start a meeting with students, and they can even create a class post and upload their CV to the needs page.

Private schools share their information and requirements to receive and respond to requests from the teacher.

The administrator manages reports and accepts teachers and private schools' applications.

2.2. Analysis of the Existing Similar Apps

Some apps have features like ours such as:

- **Find Teacher:** provides an online platform for both the students and the teachers. Teachers can upload their profile to get Home Tution. Parents or students can easily locate teachers near to them.

Figure 3. Finder Teacher Profile Page

- **Amazing Talker:** learn a language and take a lesson anywhere you want using your mobile devices or computer.

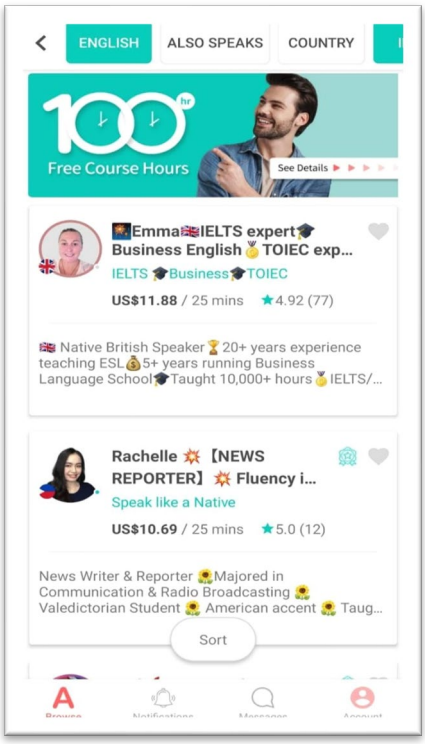

Figure 4. Amazing Talker Home Page

- **Eras Teachers:** mobile app offers a convenient way to explore the countless potential teaching jobs across a wide range of top institutions and apply instantly that best suits the profile.

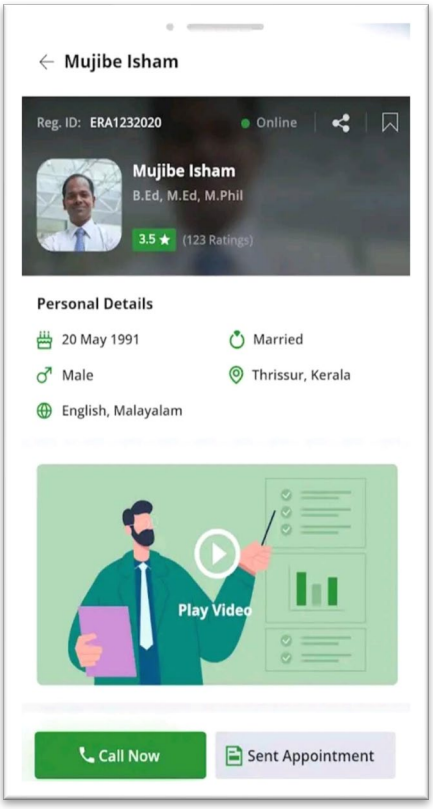

Figure 5. Eras Teachers Profile Page

The existing applications, presented above, can be synthesized in the following table:

	Support all subjects	used by private schools	Job interviews	Exams and lessons	Payment
Find Teacher	X				
Amazing Talker				X	X
Eras Teachers	X	X	X		

Table 1. Comparison of Existing Similar Applications

I.3. Plan of the document

In this document, we propose a specification, design, and project planning for the TeacherPiker application. We start first by showing gathered requirements from the client distributed between the functional and non-functional requirements.

To come in the next chapters, we will show the system design (interface, UMLs, and textual description), high level design and the technical design, which states the backend and the front end used to develop such software.

Chapter II - Requirement Analysis and Specification

II.1. Introduction

Our client is a young guy who wants to do something beneficial to our society, so he created the idea of TeacherPiker to become an initiative to help the education sector and increase educational awareness.

First, his idea was a little bit blur, and we have contributed together by intensive interviews to make it clear and bring it to light.

During interviews, some of our questions were:

1. Can anyone ask to meet?
2. What are the main criteria for obtaining a teacher or private school account?
3. Which payment facilities you want to add?
4. How do you make sure that the information a teacher puts in their profile (such as the number of years of experience) is correct?
5. Do you want the guest to reach the needs page?
6. Do you want to add filter options to the search page?
7. Who can upload CVs?
8. Do you want the same concept as Facebook's stories and posts?

II.2. Chosen life cycle

The software development life cycle that we have found to be most suitable for our project are Agile Model. Because Agile model is a combination of an incremental and iterative approach and is focused on fitting in well with flexible requirements. In addition, the Agile model does not require detailed planning like other predictive methods. If one needs to make a change, it can be done in the same sprint.

Some of its benefits:

- Quick development
- Quality and measurable results
- Business value can be delivered – demonstrated fast
- Requires minimum resources
- Highly adaptive to changing requirements

II.3. Prototype

Figure 6. Welcome Page

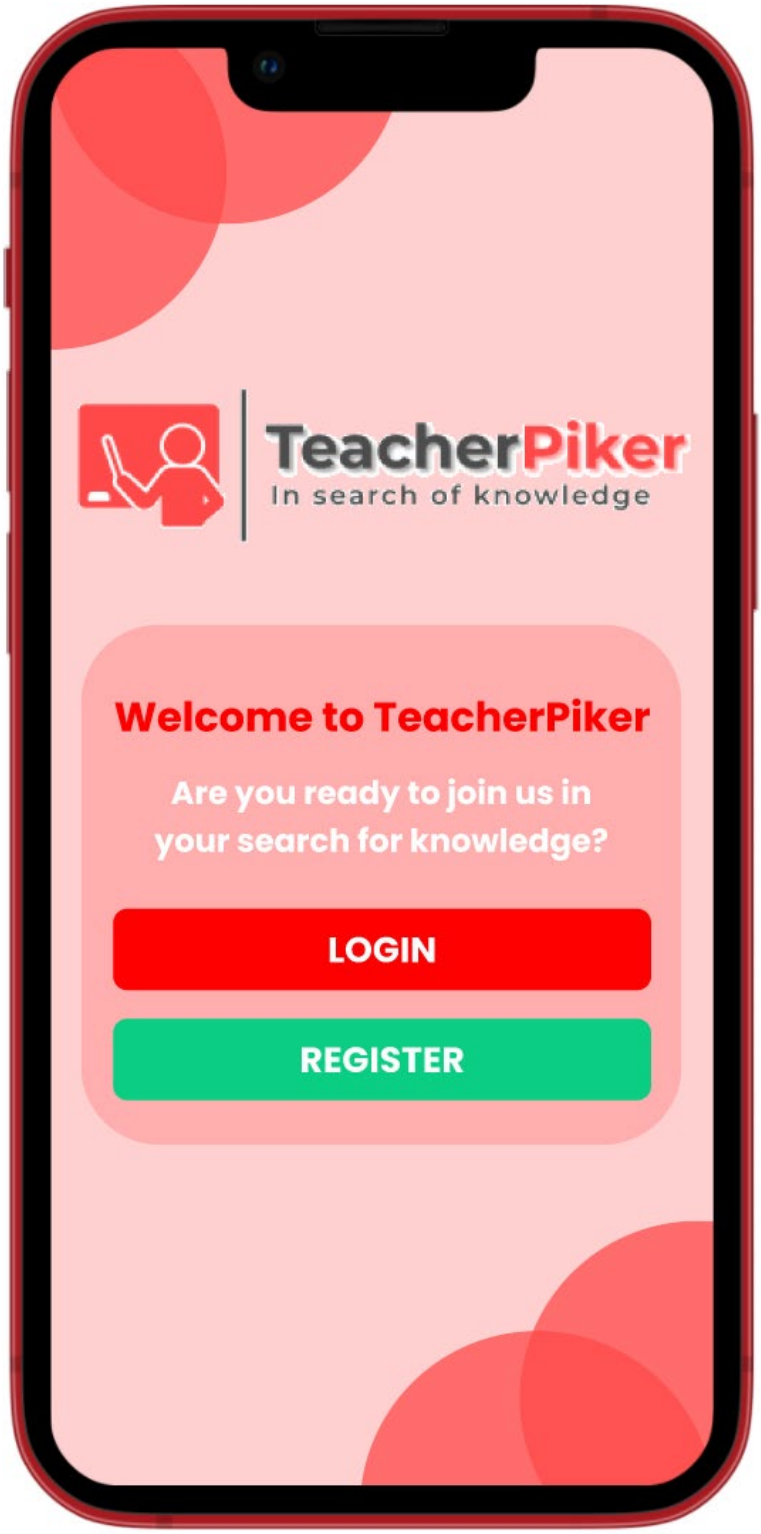

Figure 7. Login Method Page

Figure 8. Account Type Selection Page

Figure 9. Registration Page

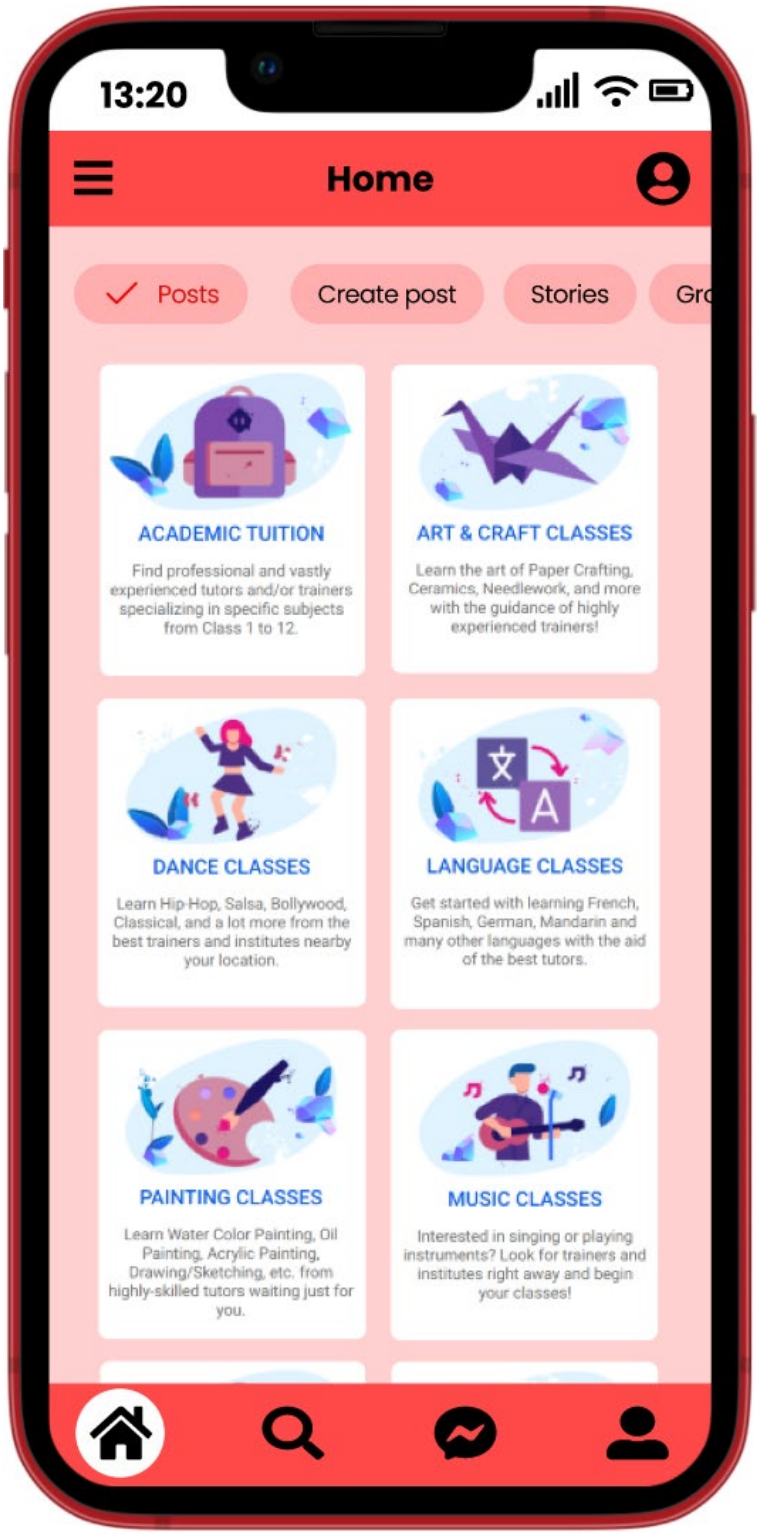

Figure 10. Home Page

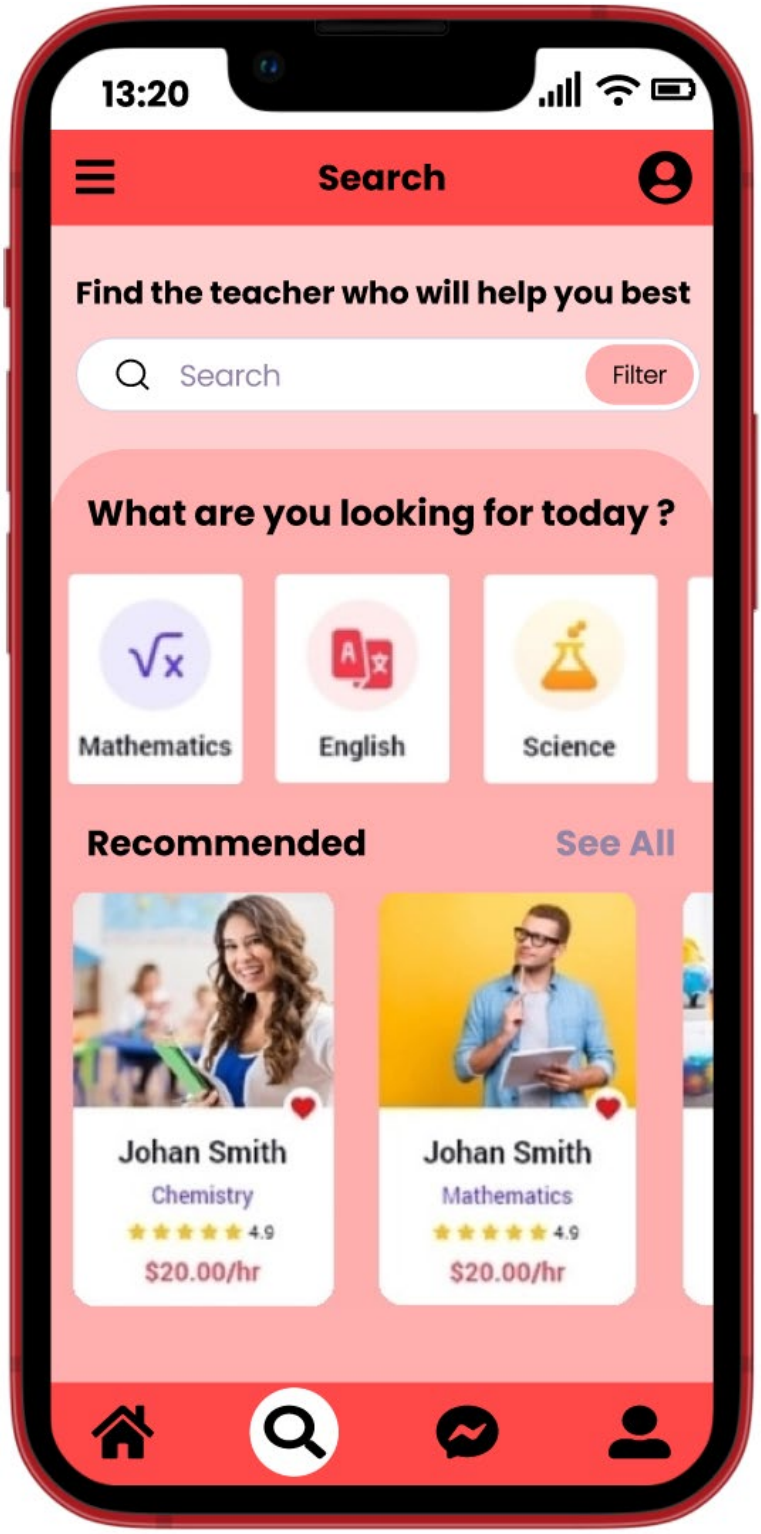

Figure 11. Search Page

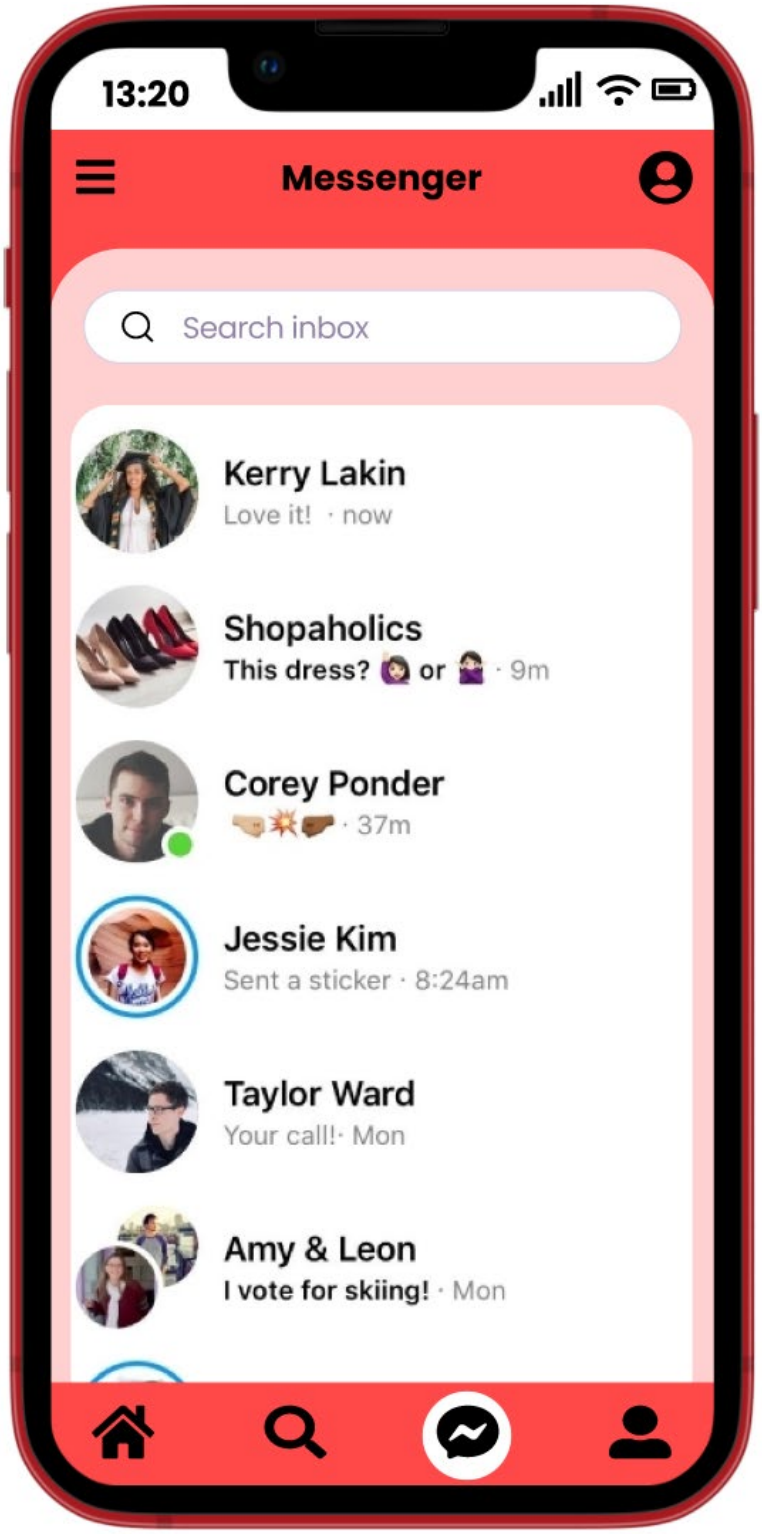

Figure 12. Messenger Page

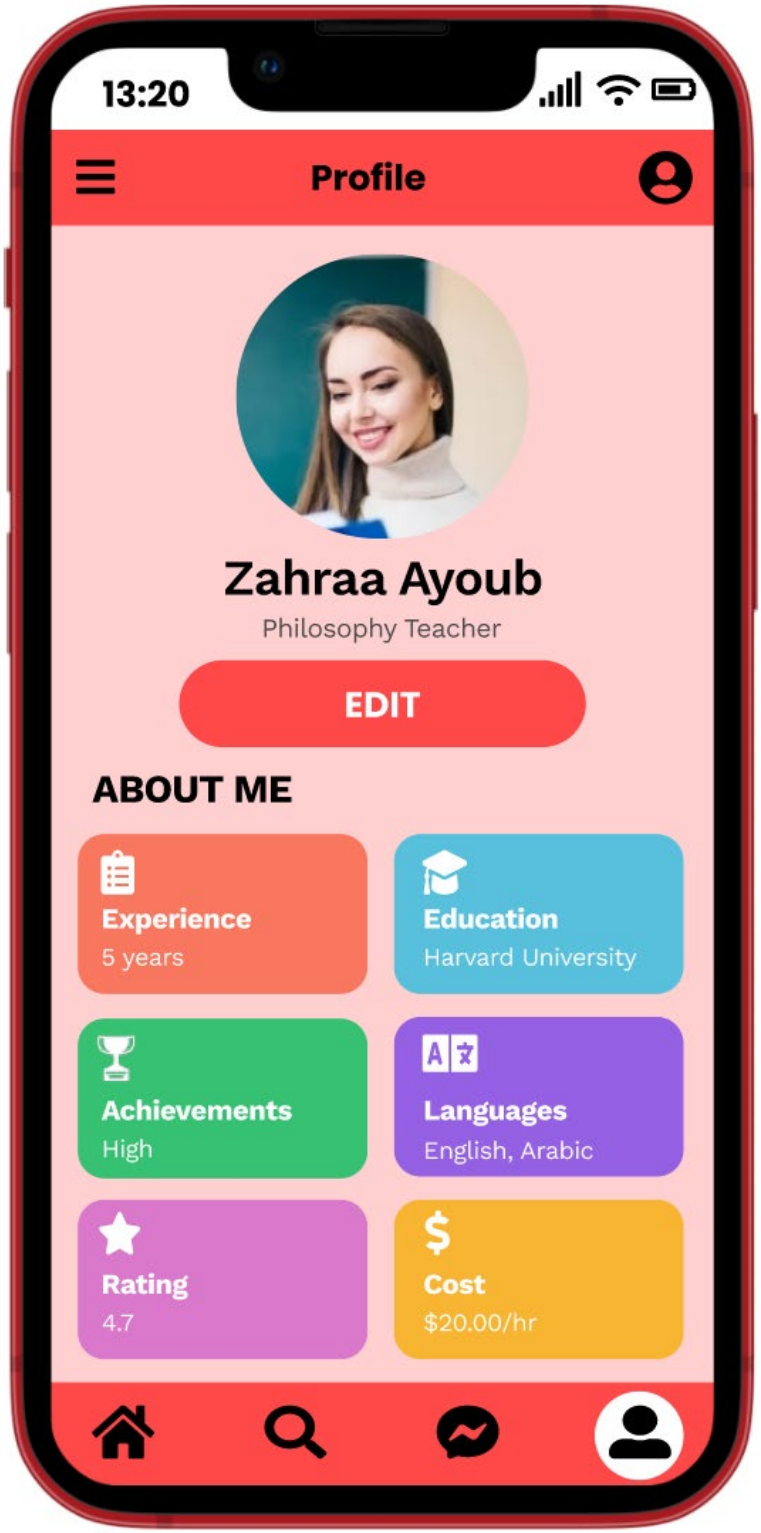

Figure 13. Profile Page

II.4. Requirements Analysis

4.1. Functional Requirements

1. The guest can select the language of the application (AR or EN).
2. The guest can view the posts and access the search page without registering.
3. The guest can select the location, subject, study online or not, maximum cost of hours and minimum years of experience in the search page by filtering option.
4. The guest can sign up as a user, teacher, or private school.
5. The guest can log in with his username and password whenever he wants.
6. To become a teacher or a private school, the user must prove it by filling out a special application.
7. The admin can accept or refuse the applications.
8. After logging in, the user can access the needs page and use the services of meeting request, calling, and chatting.
9. The user can edit their profile information.
10. The user must add his ID or something that proves his identity when he wants to request a meeting or call with someone.
11. The user can publish a photo, video, article, text, or any type of post he wants (Certificate, event announcement, ...) on the posts page.
12. The user can interact with the posts by liking, sharing, and commenting on them.
13. The user can share a 24-hour lasting story.
14. The user can report the content of another account.
15. The admin checks the reports and may delete the content or ban the user reported.
16. The user can submit a teacher's rate and put a comment on their profile.
17. The user can choose the method of payment by credit or by bank transfer.
18. The private school can make a list of all his requirements (subject, location, cost, ...) and put it on the posts page with the option of contact us.
19. The teacher can upload his CV to the posts of the needs page via the Easy Apply option.
20. The private school can check all CVs and return a response.
21. The teacher can create an assignment or quiz for the user with a deadline.
22. The user can upload the test response as text or image.
23. The teacher can return the test mark.

We are concerned about the idea of special application and IDs, but we can see that the customer is afraid of scams and false information. The environment must remain safe.

4.2. Non - Functional Requirements

- **Portability:** [Device independence] On-demand course lesson videos should be viewed by users from multiple operating systems including Microsoft Windows, macOS, and Android.
- **Maintainability:** The mean preventative maintenance time on applying routine plug-in updates TeacherPiker App should be less than 30 minutes every 2 weeks.
- **Accessibility:** [Accessible by people who are hard of hearing] All course lessons will provide a text alternative to audio content.
- **Confidentiality:** [No sensitive cardholder retention] TeacherPiker App will not retain users credit or debit card information entered during the Checkout payment processing.
- **Access Security:** [Forgotten password] Users may request a temporary password and should receive a link sent to their primary email address.
- **Efficiency:** [Video load time] All course lesson videos should load in 2 seconds or less.
- **Survivability:** [Update failure detected] When an update failure is detected all updates performed during the failed session shall be rolled back to restore the data to pre-session condition.
- **Usability:** [Downloads are easy to access] Users should have the option to download course materials when viewing a course lesson or the course overview.

II.5. Specification

5.1. Use Cases

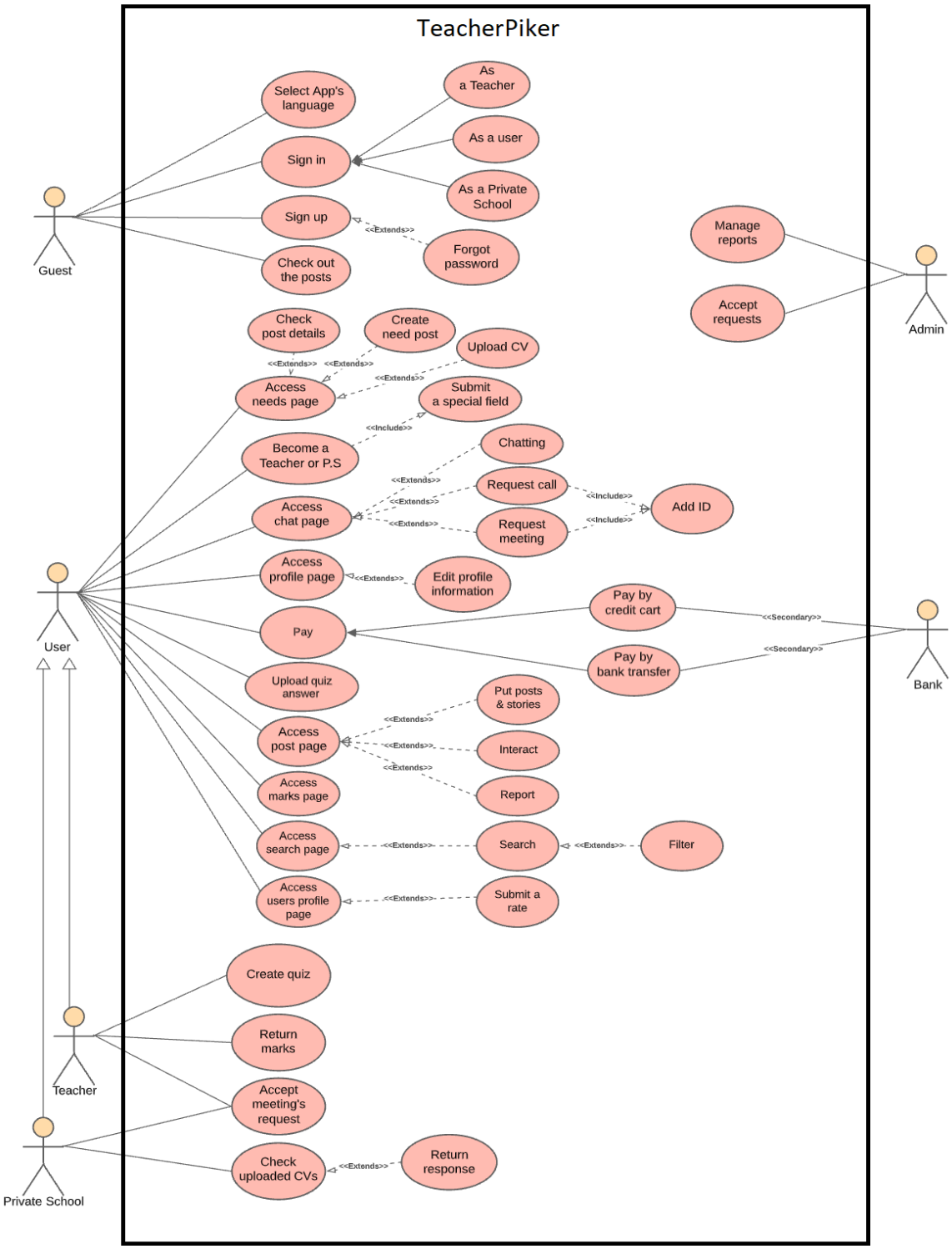

Figure 14. UML Use Cases Diagram

5.2. Use Cases Textual Description

Use Case ID	**F21**
Use Case Name	Create quiz
Actor	Teacher
Pre-Condition	None
Post-Condition	The quiz is uploaded.
Normal Flow	1- The system displays the Create Quiz Page with the Add Question button. 2- The teacher clicks the Add Question button, and the system will display a page with options, all of which are optional (except for 'Select Question Type'): • Select Question Type • Add Deadline Time • Add Photo or Video • Add Note 3- The teacher clicks on select question type and the system will present two options: • MCQ Question • Text Question 4- The teacher selects MCQ Question. 5- The system displays the question-and-answer field to be filled in. 6- The teacher selects the correct answer from among the written possibilities. 7- The teacher clicks on the End Question button and the system will return it to the Quiz Page containing the new question added. 8- The teacher clicks on the End Quiz button and the system will display a list of students. 9- The Teacher chooses which students are allowed to access the quiz and clicks the Upload button. 10- The system displays a successful upload message.
Alternative Flow	3.b The teacher selects Text Question => The system displays a field to type in the question.
Exceptional Flows	7' The teacher did not specify the question type => The system displays a message that you cannot add a question without specifying its type. 8' The teacher did not add any questions => The system displays a message that you cannot create a quiz without adding at least one question to it. 9' The teacher did not specify who is allowed to access the quiz => The system displays a message that you cannot upload a quiz without specifying which student has access to it.

Table 2. Create quiz DTD

Use Case ID	F6
Use Case Name	Become a teacher or private school
Actor	User
Pre-Condition	None
Post-Condition	The user account has been converted to a teacher or a private school account.
Normal Flow	1- The system displays a page with the following options: • Become a teacher • Become a private school 2- The user chooses Become a teacher. 3- The system shows a page with fields to be filled: • ID Number • Major • Graduation certificate copy 4- The user clicks the Submit button. 5- The system displays a message to the user that you must wait a few minutes to check your answers. 6- The system displays a successful account modification message and returns the user to the home page.
Alternative Flow(s)	3.b The user chooses Become a private school => The system shows a page with fields to be filled: • ID Number • School Name • Location
Exceptional Flow(s)	4’ The user did not fill out the form completely => The system displays a message that you cannot submit your answers without completely filling out the form. 6’ User answers are incorrect = > The system displays a message that the account modification failed.

Table 3. Become a teacher or private school DTD

II.6. Conclusion

In this chapter, we have presented:

- The sketches helped us in the specification.
- The User interface design prototype.
- The client requirements are categorized between functional and non-functional.
- The use case diagram contains all actors and functionalities.
- Some textual descriptions.

We noticed that defining requests was the basic process on which we built the rest of the operations. Although the process of determining orders is the most difficult because of the lack of clarity of customer ideas.

To clarify the customer's ideas, we proposed several programs that are similar in concept. We showed it to the customer and compared the services offered by these programs and the services that the customer requested.
This allowed the client to arrange their ideas and explain them to us further, and it also allowed us to visualize these ideas and requests.

Chapter III - Application Conception

III.1. Introduction

In the previous chapters, we have talked about the features that should be offered by our application. This reveals that the following entities are implied in the process. In this chapter we give, using UML class diagrams, the conceptual model that clarifies these entities, their underlying handled data, roles in the application, and associations with each other.

III.2. UML Class Diagram

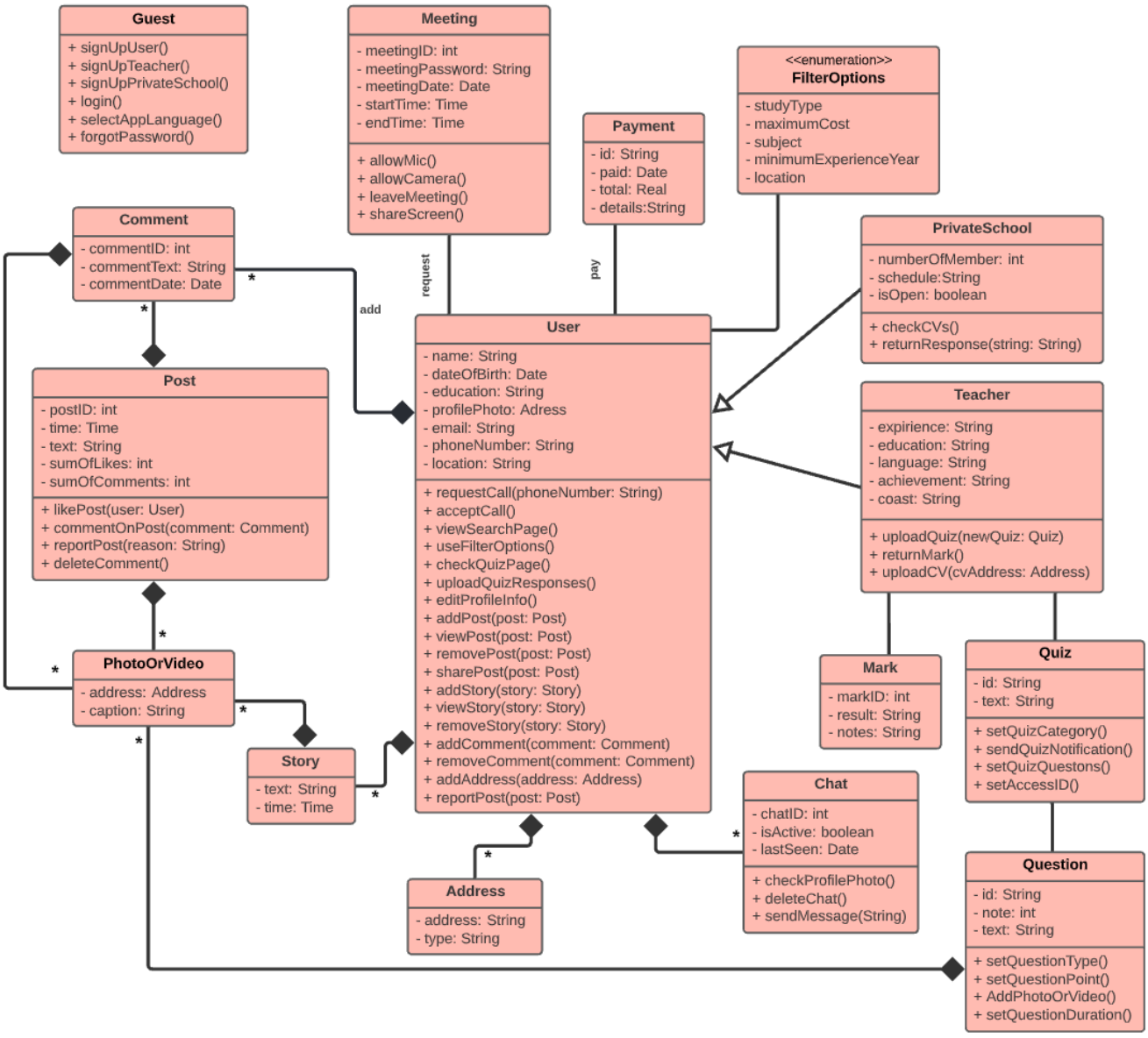

Figure 15. UML Class Diagram

III.3. Sequence Diagram

In this diagram we display the Sequence Diagram of the Upload Quiz scenario:

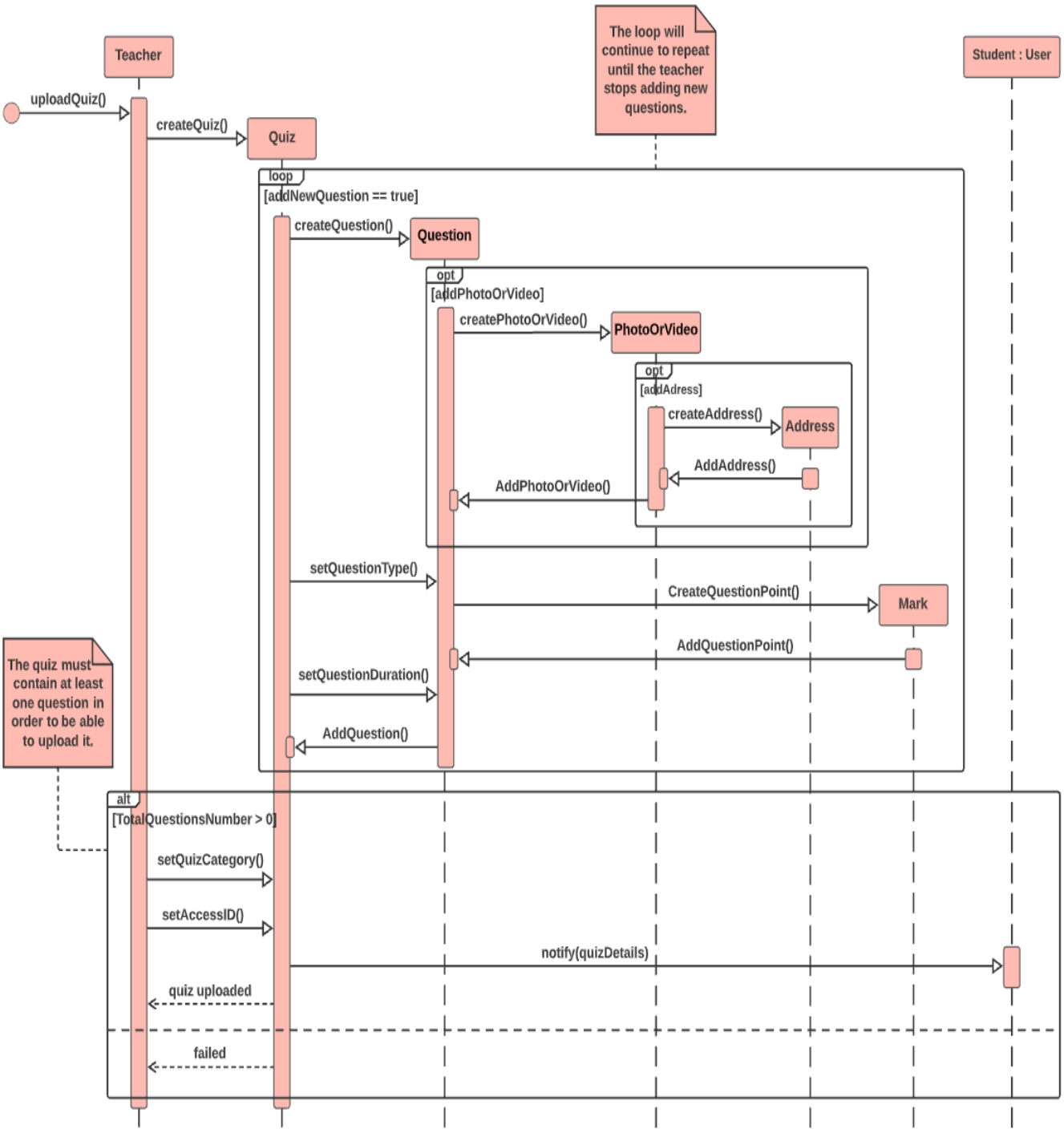

Figure 16. Upload Quiz UML Sequence Diagram

III.4. Statechart Diagram

In this diagram we display the Statechart Diagram of the Meeting:

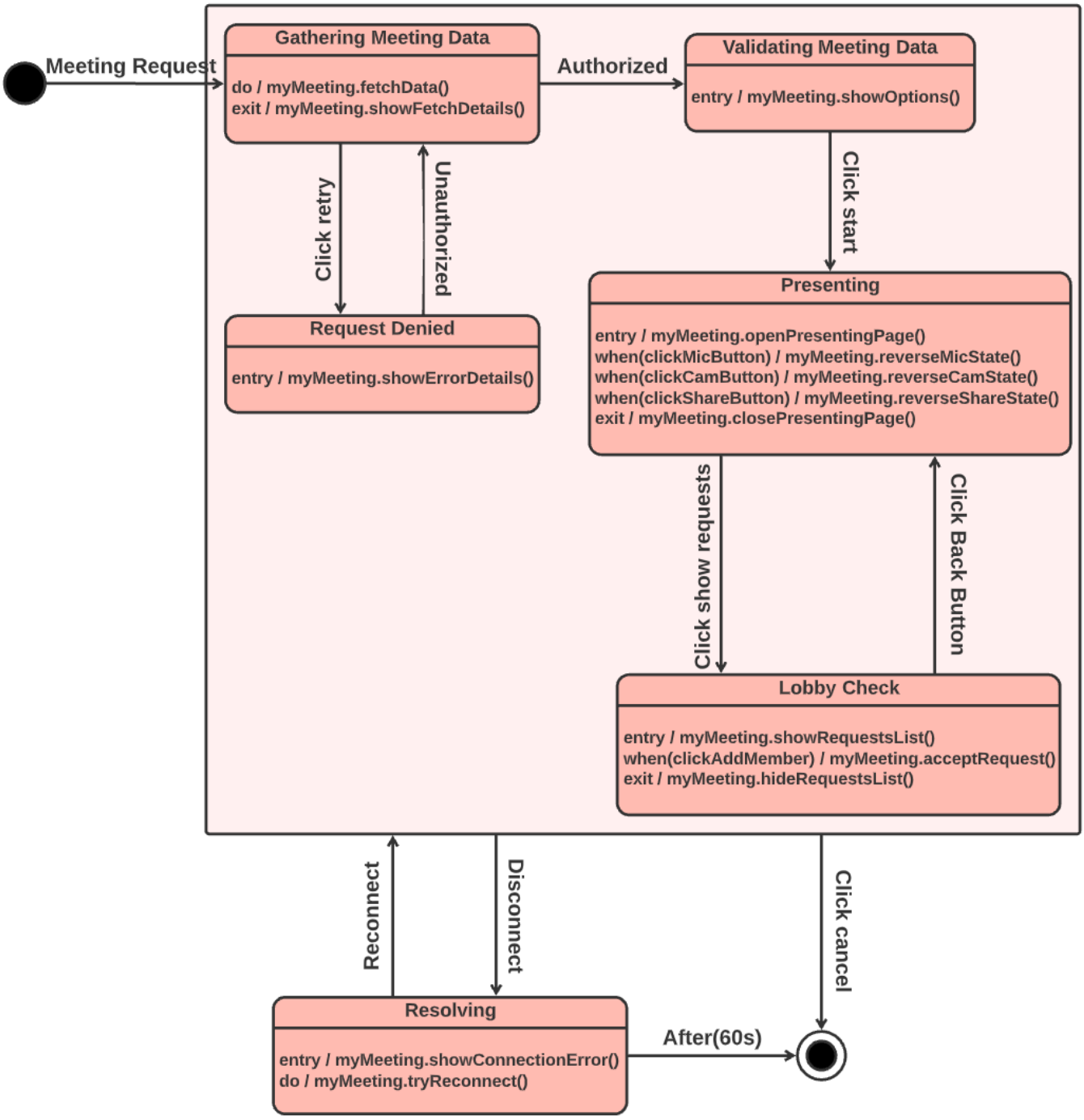

Figure 17. Meeting UML Statechart Diagram

III.5. Technical details

To build our app, we decided to use Xamarin which is an open-source platform to build modern and efficient apps for iOS, Android, and Windows with .NET. And for the backend, we want to use Azure which is the best option with Xamarin because Azure platform can be utilized for implementing many mobile features, and it can be used for creating Node.js and C#. Finally, we use MySql for the database which is an open source, multi-threaded and easy to use SQL database.

III.6. Conclusion

This chapter presented the design of our application. Firstly, we introduced our class diagram that explains the relationships between our different entities. Also, we continued with a sequence diagram. In addition, we have developed a state diagram that considers the various scenarios that the states of the object may go through. This helped us to understand the flow of events and identify scenarios accurately, which facilitates the programming process. Finally, we have outlined some technical details that depend on the languages used to build our app.

Chapter IV - Conclusion

After completing this project, we came to important conclusions. This project was an excellent way for us to understand and understand the software engineering course, prepare well for the career field abroad, and especially the teamwork environment. Many difficulties appeared while working on it, and these problems varied between understanding what the client needs and wants in his system and the features and aspects that the system should include. Doing many schematics seemed difficult at first, but in the end, it turned out that the programming process was greatly simplified by defining form, functions, objects and how they interact with each other before starting.
Besides, this project was an excellent application of what we took in the software engineering course, and to prepare for more complex systems in the future.

IV.1. Futures Plans

A lot of plans can be made for this program. We can include a Student Statistics section that shows the percentage of changes in grades per month and subject by subject and gives a warning if a fifteen percent drop is observed. This warning is directed to the teacher and student to take actions such as additional lessons. We can include a Student Frequent Errors section and this section saves the wrong answers in the solved tests and randomly collects them into separate tests that the student chooses to solve for review. For teacher and parents, we can add monthly student status report section.

Bibliography

[1] Grady Booch, *Object-Oriented Analysis and Design with Applications*, 2nd ed., Benjamin/Cummings, 1994.

[2] Grady Booch and James Rumbaugh and Ivar Jacobson, *Unified Modeling Language User Guide*, Addison-Wesley Object Technology Series, Summer 1998.