



People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific  
Research  
University of Mohamed El Bachir El Ibrahimi -  
Borj Bou Arréridj  
Faculty of Mathematics and Computer Science  
Computer Science department



# Graduation Thesis

## Development of a Peer-Learning Mobile Application



*Presented for Graduation : Master's in  
Computer Science  
Specialty : Networks and Multimedia*

*Presented by :  
Douibi Aimane Malek  
Benlefki Abdessalem*

*In Front of the Jury composed of:      Date : 06 / 20 / 2024  
Supervisor: BENDIAF MESSAOUD  
President : MOSTEFAI MESSAOUD  
Examiner: BENMALEK MOUNIR*

**2023/2024**

## ***Appreciation***

*We thank God for giving us the courage and the will to carry out our project. We would like to thank our instructor Dr Bendiaf Messaoud for its availability, help, valuable advice, constructive criticism, explanations and relevant suggestions. We would like to express our gratitude to the members of the jury for agreeing to evaluate and judge our work. And finally we thank all those who have contributed far or near for the realization of our project.*

## ***Dedication***

*We dedicate this thesis to God, whose guidance and blessings have been our strength throughout this journey.*

*To our families, whose unwavering love, support, and encouragement have been our constant source of inspiration and motivation.*

*To our teachers, whose wisdom, knowledge, and guidance have been instrumental in our academic and personal growth.*

*Thank you all for being an integral part of this achievement.*

# Table Of Contents

List of abbreviations .....	ix
List of figures .....	ix
Context .....	6
2. Issue .....	
3. Description of available Solutions .....	
4. Objective And Contribution .....	
5. Thesis Plan .....	
CHAPTER 1: Basic Concepts .....	
1.1 Integration of E-Learning Principles : .....	7
1.1.1 Understanding E-Learning Principles : .....	
1.1.2 Content Delivery Methods in E-Learning : .....	7
1.2 Peer Learning Definition : .....	8
1.2.1 Peer-to-Peer Benefits : .....	8
1.2.2 Peer Learning Models : .....	8
1.3 State of the Art : .....	9
1.3.1 Introduction .....	9
1.3.2 Existing Models .....	9
1.4 Conclusion .....	13
CHAPTER 2: Analysis and UML Modeling .....	14
2.1 Introduction .....	15
2.2 Design Methodology : .....	15
2.2.1 Presentation of UML .....	15
2.3 Analysis and Design .....	15
2.3.1 Use Case Diagram : .....	15
2.3.1.3 Use Case Diagram : .....	17
2.3.2 Class diagram .....	18
2.3.3 Sequence diagramme .....	19
2.3.3.1 Authentication sequence diagram.....	19
2.4 Database SCHEMA .....	20

2.5 Conclusion.....	21
<b>CHAPTER 3 : Application Implementation</b>	<b>22</b>
3.1 Introduction :.....	23
3.2 Work Environment .....	23
3.2.1 Hardware Environment :.....	23
3.2.2 Software Environment : .....	24
3.2.3 PostgreSQL for Database Management :.....	26
3.3 Admin Panel .....	27
3.3.1 Tables in Supabase.....	28
3.3.2 Managing Users in Supabase .....	28
3.3.3 Check Request Log in Supabase.....	28
3.3.4 Managing Content in Supabase.....	30
3.4 Presentation of the Interfaces of Our Application .....	31
3.4.1 Splash Interface.....	31
3.4.2 Authentication Interfaces .....	32
3.4.3 Home Page Interfaces .....	34
3.4.4 Search Interface.....	37
3.4.5 Profile Interface.....	38
3.4.6 Complaint Interface.....	39
3.4.7 Chat Interfaces : .....	40
3.4.8 Subject selection : .....	42
3.4.9 Group creation interfaces .....	44
3.4.10 Become A Tutor .....	45
3.4.11 Rating Interface.....	46
3.5 Fahemni Website .....	47
3.5.1 Hostinger Web Hosting .....	48
3.5.2 Hostinger Dashboard .....	48
3.6 Conclusion .....	49
General coclusion .....	
Future work and perspectives .....	
SUMMERY .....	

# List of Abbreviations

**PL:** Peer Learning

**MVC:** Model View Controller.

**UML :** Unified Modeling Language.

**LMS :** learning management system.

**UI :** User Interface.

**DB :** Database.

**BBA:** Bordj Bou-Arredj.

# List Of Figures

1.1	E-Learning Platform .....	11
1.2	Stack-Overflow Website.....	12
1.3	Schoology Website.....	13
1.4	Canva Website.....	14
2.1	Use Case Diagram.....	19
2.2	Class Diagram.....	20
2.3	Sequence Diagram OF AUTHENTICATION.....	21
2.4	sequence Diagram of Post Problem .....	22
2.5	Database schema.....	23
3.1	MVC.....	28
3.2	Table Operations.....	31
3.3	Manage users interface.....	32
3.4	Logs.....	33
3.5	MANAGEMENT CONTENT interface.....	32
3.6	Statistics Interface.....	34
3.7	Splash Interface.....	35
3.8	Register interface.....	36
3.9	Login interface.....	37
3.10	New Section Interface.....	38
3.11	Specialties Interface .....	39
3.12	Problems Section Interface.....	40
3.13	Search Interface.....	41
3.14	Manage Profile Interface.....	42
3.15	Update Info Interface.....	43
3.16	File Complaint Interface.....	44

3.17 Chat Interface .....	45
3.18 specialties interface .....	46
3.19 Subjects Interface .....	47
3.20 Resource Interface .....	47
3.21 Subject Details Interface .....	48
3.22 List learner and Tutor .....	49
3.23 Create groupe Interface .....	49
3.24 Become Tutor Interface .....	50
3.25 Rating Interface .....	51
3.26 Web Home Interface .....	52
3.27 Web Problems Interface .....	52
3.28 Hosting Dashboard .....	53



# GENERAL INTRODUCTION

The virtual revolution has dramatically converted conventional instructional landscapes, introducing dynamic methodologies that redefine how understanding is accessed, shared, and created. Central to this change is the arrival of e-studying, which leverages virtual technology to dismantle boundaries and enlarge instructional possibilities past traditional school room settings. As we delve into the intricacies of this virtual shift, the emergence of peer studying structures represents a pivotal evolution, shaping the destiny of schooling through collaborative and interactive studying experiences. This document explores the improvement of an progressive cell utility designed to facilitate peer-to-peer studying, addressing the precise demanding situations and harnessing the capability of virtual schooling.

# 1. Contexte

E-learning has evolved significantly, particularly since the COVID-19 pandemic forced educational institutions to shift from in-person instruction to remote learning. This change accelerated the adoption and innovation of e-learning platforms, making education accessible globally.

E-learning democratizes education by enabling learners from various backgrounds to access high-quality resources. Accessibility ensures that learning materials are usable by all learners, including those with disabilities. Flexibility allows learners to access content at their own pace, balancing education with other responsibilities. Feedback and assessment provide critical insights into learner progress.

E-learning utilizes diverse delivery methods, including text, video, multimedia content, interactive modules, real-time lectures, discussion boards, and mobile learning. Peer learning components, such as peer-to-peer tutoring and collaborative problem-solving, enhance the learning experience by fostering community and knowledge sharing.

The pandemic highlighted the necessity and potential of e-learning to revolutionize education. By embracing effective e-learning principles and advanced technologies, educational institutions and learners can benefit from the flexibility, accessibility, and collaborative opportunities that digital learning offers.

## 2. ISSUE

In the sphere of e-learning, powerful control of getting to know and short to get right of entry to information are critical to make certain a hit getting to know experience. The absence of specialized mobile programs in e-learning to know poses the subsequent problems :

- **Limited Interaction** : Without those programs, Interactions among students regularly stays constrained to standard, thereby hindering real-time verbal exchange and collaboration.
- **Content Quality**: In the sphere of e-learning, content quality remains a significant concern. Often, online educational content can be static, outdated, or not aligned with the latest educational standards. Poorly designed materials can lead to disengagement, misunderstanding, and an overall ineffective learning experience. Unlike traditional face-to-face instruction, where educators can adapt and update content dynamically based on student feedback, e-learning platforms sometimes struggle to maintain the same level of adaptability and responsiveness.
- **Restricted right of entry to sources** : Educational sources along with textbooks and course materials can be scattered throughout one-of-a-kind systems, making it hard for beginners to get right of entry to them and resulting in a lack of time and performance in locating the necessary records.
- **Technological Glitches and User Interface**: Technological glitches and poor user interfaces are common stumbling blocks in e-learning. Technical issues such as slow loading times, broken links, and system crashes can disrupt the learning process and frustrate students. Additionally, a poorly designed user interface can make navigation difficult, leading to a disjointed learning experience. Unlike traditional classrooms where technical issues are minimal, the dependency on technology in e-learning makes it essential to ensure that platforms are user-friendly and reliable.
- **Difficulty Finding Help**: One significant issue in e-learning is the struggle students face when they do not understand something and need immediate help. In traditional classroom settings, students can raise their hands and get instant clarification from the teacher. However, in an online environment, finding timely assistance can be

challenging. Students may have to wait for responses from instructors or peers, and the lack of real-time support can hinder their learning progress. This delay can lead to frustration and gaps in understanding, making it difficult for students to keep up with their coursework.

### **3. Description of available solutions :**

- **Regular Content Updates:** Implement a system for regular updates and reviews of educational content to ensure it remains current and aligned with the latest educational standards.
- **Active Learning Techniques:** Incorporate active learning techniques such as interactive assignments, discussion forums, and group projects to enhance engagement and retention.
- **Interactive Elements:** Incorporate interactive elements such as quizzes, videos, and simulations to engage learners and enhance their understanding.
- **Feedback Mechanism:** Establish a robust feedback mechanism where students can report outdated or unclear content, prompting timely updates and improvements.
- **User-Centric Design:** Invest in user experience (UX) design to create intuitive and user-friendly interfaces that enhance navigation and usability.
- **Technical Support:** Provide dedicated technical support and troubleshooting resources to help users resolve issues quickly.
- **System Reliability:** Ensure robust infrastructure and regular maintenance to minimize downtime and technical glitches.
- **Peer Learning Communities:** Foster peer learning communities where students can collaborate, ask questions, and provide support to each other

## 4.Objective and Contribution :

Our objective is to build a peer learning platform targeting students. By matching up students who can help each other grow. By leveraging the power of collaborative learning, students are able to ask questions and find reliable answers from their peers in an organized environment, thus enhancing their conceptual clarity on subjects.

### The application must provide :

- **Collaborative Pairing System** : A System that matches learners with tutors based on their ratings and availability. This ensures learners receive high-quality instruction tailored to their schedules, while tutors can efficiently manage their time. The system dynamically updates to reflect changes in ratings and availability, continuously optimizing pairings for the best learning outcomes. By considering individual preferences and goals, the platform creates personalized learning paths, enhancing the educational experience for both learners and tutors.
- **Answer and Question Section** : Our application includes a dedicated Answer and Question section where students can post their academic questions and get help from other users. Similar to how programmers use forums to solve coding issues, students can leverage this section to address their educational challenges. When a question is posted, the system matches it with users who have the relevant knowledge and availability to provide a detailed response. This ensures that students receive timely and accurate answers, creating a collaborative and supportive learning community. This feature allows users to share insights and solutions, enhancing the overall educational experience.
- **Global Accessibility** : By leveraging technology, our platform transcends geographical barriers, allowing learners from around the country to connect and collaborate seamlessly. Whether students are in different time zones or remote locations, they can engage in meaningful learning experiences without constraints.
- **Messaging Platform Integration** : Implement a messaging platform in the app to facilitate seamless conversation among users. This characteristic allows college students to interact in immediate messaging, voice calls, video calls, textual content messaging, and media/document sharing. Users can have interaction with their matched friends in real-time, fostering collaborative getting to know studies and improving peer-to-peer tutoring sessions.

- **Supervised Learning Environment** : where tutoring sessions can take place in a supervised and structured manner. This includes providing tools for safe communication between tutors and learners, and offering resources for guidance and assistance during the learning process.

For that, our "Peer Learning Platform" application should consist of three parts :

- **The first part is for the learner** : It enables the learner to browse through available learning topics, view profiles of potential peers, and request to connect with them for collaborative learning sessions.
- **The second part is for the tutor** : It allows the tutor to showcase their expertise, availability, and teaching style, as well as to browse through requests from learners and accept or decline them based on their preferences.
- **The third part is for the administrator** : It provides administrative functionalities such as managing user accounts, monitoring platform activity, and resolving any disputes or issues that may arise during the learning sessions

## 5. Thesis Plan :

- In this first chapter, we will focus on E-learning principles, based on personalization, interactivity, accessibility, flexibility, and feedback and assessment. Additionally, the content is delivered through a range of ways from simple text and video material into multimedia content, interactive or passive modules, real-time lectures, discussion boards, or mobile learning. Peer learning can be integrated with the help of peer-to-peer tutoring and collaborative problem solving, ensuring the variety of collaborative experiences for the learners. Some best practices are presented on the market with examples of Moodle, Stack Overflow, Schoology, and Canvas.
- The second chapter will focus on the phase of application development analysis and modeling. In this stage, we introduce Unified Modeling Language (UML) and explain in as much detail as possible how to apply them to design the system architecture: class diagram, sequence diagram, use-case diagram, and will add a schema for the Database.

- The final chapter will detail the implementation phase of the application, highlighting the development process and key functionalities. We will discuss how the application works in practice, including user interactions, data flow,

# CHAPTER ONE:

## Basic Concepts:

### INTRODUCTION

1.1. Section One: Integration of E-Learning Principles

1.2. Section Two: Peer Learning Integration

1.3. Section Three: State of the Art

### CONCLUSION



## 1.1 Integration of E-Learning Principles :

### 1.1.1 Understanding E-Learning Principles :

E-learning principles encompass a variety of strategies and methodologies designed to optimize the online learning experience. This section explores key concepts such as :

- Personalizing : Tailoring academic content material to people getting to know patterns and preferences.
- Interactivity : Engaging learners via interactive activities, quizzes, and simulations.
- Accessibility : Ensuring that academic substances are reachable to all inexperienced persons, irrespective of their capabilities or limitation
- Flexibility : Providing learners with the liberty to research at their personal pace and convenience, regularly via asynchronous getting to know methods
- Feedback and Assessment : Offering well timed remarks and checks to gauge learner development and comprehension.

Understanding those concepts is important for designing powerful e-learning reviews that promote student engagement and know-how retention (T, 2008).

### 1.1.2 Content Delivery Methods in E-Learning :

Content delivery techniques in e-learning embody numerous processes for presenting educational materials to learners. This phase explores :

- Text-Based Materials : Delivering content material thru written text, along with articles, textbooks, and virtual documents.
- Multimedia Presentations: Utilizing multimedia factors together with videos, animations, and audio recordings to enhance engagement and comprehension.
- Interactive Modules : Incorporating interactive modules, simulations, and digital labs to inspire active learning and hands-on experience. Stay lectures and webinars to facilitate real

time interplay among teachers and learners.

- Discussion Forums and Collaborative Platforms : Providing dialogue boards and collaborative structures in which learners can have interaction with peers, ask questions, and share insights.
- Mobile Learning : Leveraging cell gadgets to supply bite-sized content material, and learning activities anytime, anywhere.

## 1.2 Peer Learning Definition:

Peer learning is a teaching method through which students will be able to consolidate their knowledge in a better manner by teaching it to one another. This ensures that in a controlled environment, one student teaching another will learn and retain better .

(Boud D. & Sampson J., n.d.)

### 1.2.1 Peer-to-Peer Benefits:

Features in peer-to-peer tutoring foster collaborative learning experiences among the students.

This section highlights the following:

- Students Can Learn A New Perspective: If a student is going to learn only from the teacher, they will have only learned one new perspective. If a student gets to learn from a peer,
- Studying Becomes Fun Due To Social Interaction: Human beings are social beings. They crave to get connected and be a part of the group. The additional element of social interaction inherent in peer learning can be exciting and empowering. Students hesitant to share with the instructor might do so freely with their classmates.
- Students Learn When They Teach: There is not a better way to build confidence in one's knowledge than to teach what one knows to someone else. As already mentioned, it can facilitate the education of the students who will grasp better what is being taught and be able to put it into practice; effective teaching needs some kind of in-depth understanding of the topic.

(Boud D. & Sampson J., n.d.)

### 1.2.2 Peer Learning Models :

Peer learning can be very effective using different models and strategies. See some of the tried-and-true ways to encourage peer learning:

- **Proctor Model:** In the proctor model, an older or more experienced student teaches a younger or less experienced peer. In an elementary school, this might mean that students from the higher grade level come and teach kindergartners. It could also be done with one of the cognitively skillful students helping the other peer in the class.
- **Discussion Seminars:** Discussion seminars are more common at the university level. They usually are held after students have learned the material via a lecture or weekly reading. These discussions allow students to carry their learning further and develop more insight.
- **Peer Support Groups:** Sometimes called self-study groups, peer support groups are learner-organized meetings that usually occur independently of the classroom setting and without the involvement of an instructor. These groups may convene to prepare for a test or work collaboratively on a project.
- **Peer Review Programs:** Peer review programs can be arranged in writing courses. For example, an AP English Language teacher could have students peer-review each other's essays.
- **Collaborative Projects:** Placing students in a team to work on any project involves them not only in terms of working together but also in preparation for the world of work and society. Such projects inculcate teamwork, combining various skills, and meeting deadlines
- **Mentoring:** A mentor is an experienced man in a specific area. They lead a student, guide them, and teach them the lesson they had to learn themselves in the first place. Peer tutoring lies under the shade of mentoring. Sometimes, a student who is expected to extend more help in arguments is provided with a personal peer mentor, and they work one-on-one with the student.

## 1.3 State of the Art :

### 1.3.1 Introduction

In discussing the "state of the art" for peer learning applications, we can look at examples of platforms that facilitate collaborative and peer-based learning. Here are two prominent platforms :

### 1.3.2 Existing Models

#### 1.3.2.1 Moodle (Elearning):

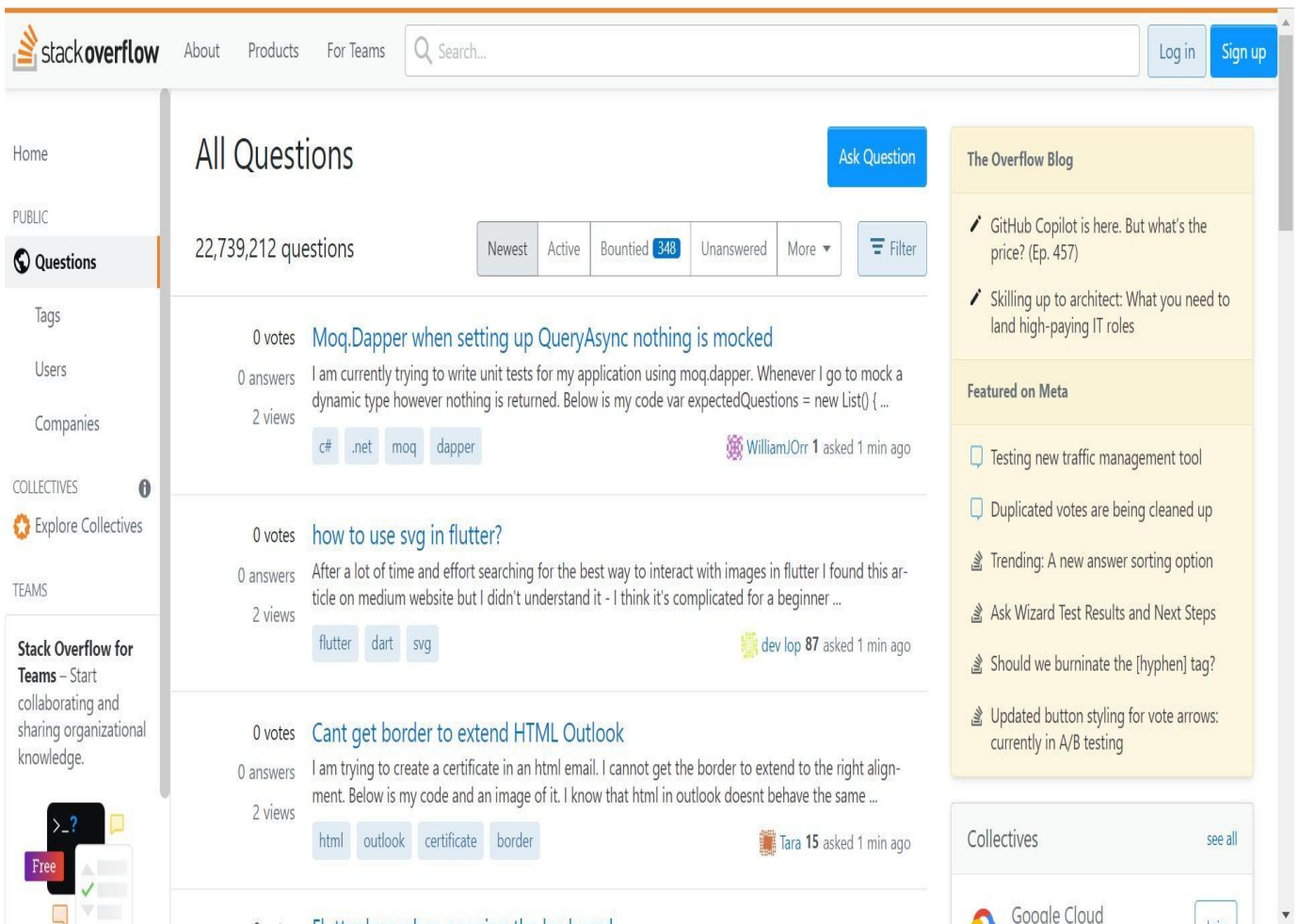


used in almost all Algerian Education Institutions such as the University of Bordj Bou Arreridj (example in Figure 1.1), Moodle is a widely-used open-source learning management system (LMS) designed to provide educators, administrators, and learners with a robust environment for personalized and collaborative learning. Moodle supports peer learning through its extensive range of features including forums, wikis, databases, and workshops, which allow learners to work together, share information, and assess each other's projects.

FIGURE 1.1 – E-learning Platform

### 1.3.2.2 Stack-Overflow : stackoverflow

Originally designed as a question and answer site for programmers, Stack Overflow has evolved into a vast learning community where users can ask questions, provide answers, and engage in discussions—effectively learning from their peers. Its reputation system rewards users for quality contributions, encouraging a collaborative and supportive learning environment, FIGURE 1.2 is the homepage of stackoverflow website. (*StackOverFlow*, 2008)



The screenshot displays the Stack Overflow homepage. At the top, there is a navigation bar with the Stack Overflow logo, links for 'About', 'Products', and 'For Teams', a search bar, and 'Log in' and 'Sign up' buttons. The main content area is titled 'All Questions' and shows a total of 22,739,212 questions. Below this, there are filter buttons for 'Newest', 'Active', 'Bountied 348', 'Unanswered', and 'More', along with a 'Filter' button. The main content area displays three question cards:

- Moq,Dapper when setting up QueryAsync nothing is mocked**: 0 votes, 0 answers, 2 views. Description: "I am currently trying to write unit tests for my application using moq,dapper. Whenever I go to mock a dynamic type however nothing is returned. Below is my code var expectedQuestions = new List() { ...". Tags: c#, .net, moq, dapper. Asked by WilliamOrr 1 min ago.
- how to use svg in flutter?**: 0 votes, 0 answers, 2 views. Description: "After a lot of time and effort searching for the best way to interact with images in flutter I found this article on medium website but I didn't understand it - I think it's complicated for a beginner ...". Tags: flutter, dart, svg. Asked by dev lop 87 1 min ago.
- Cant get border to extend HTML Outlook**: 0 votes, 0 answers, 2 views. Description: "I am trying to create a certificate in an html email. I cannot get the border to extend to the right alignment. Below is my code and an image of it. I know that html in outlook doesnt behave the same ...". Tags: html, outlook, certificate, border. Asked by Tara 15 1 min ago.

The left sidebar contains navigation options: Home, PUBLIC, Questions (selected), Tags, Users, Companies, COLLECTIVES, Explore Collectives, and TEAMS. The right sidebar features 'The Overflow Blog' with articles like 'GitHub Copilot is here. But what's the price?' and 'Skilling up to architect: What you need to land high-paying IT roles', 'Featured on Meta' with items like 'Testing new traffic management tool' and 'Duplicated votes are being cleaned up', and 'Collectives' with a link to 'see all'.

FIGURE 1.2 – stack-overflow website

### 1.3.2.3 Schoology schoology®

Schoology is an online course management system that allows teachers to create and manage academic courses for their students. Screenshot is provided in Figure 1.3. It provides teachers with a method of managing lessons, engaging students and sharing content. Schoology provides a secure and safe, easy-to-use, way for teachers, students, parents, and administrators to seamlessly communicate and collaborate to enhance students' learning and success.

Schoology is the digital extension of the classroom. Every teacher in WCASD uses Schoology in different ways to best support specific subjects and different students. However, both parents and students will view almost the exact same information when looking at a teacher's Schoology course. Below is a list of possible elements visible on Schoology : Teacher announcements and updates

- Calendar posts for important events and due dates
- Links to videos, articles, and other information to support student learning
- Descriptions of class assignments, projects, and tests/quizzes.
- Online discussion boards - Schoology parent accounts will not be able to see the names of other students on discussion posts. This is the only difference in what student and parent accounts can view on Schoology.
- Course grades and grade feedback from teachers

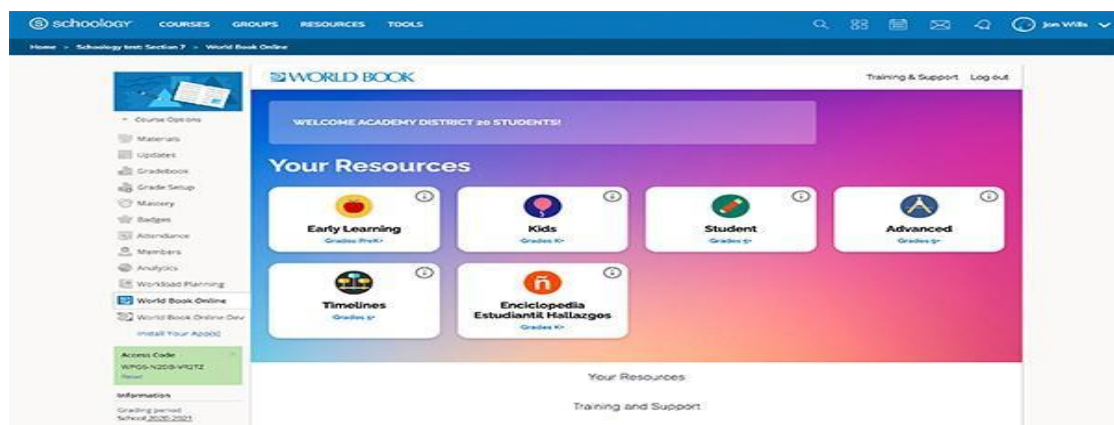


FIGURE 1.3 – Schoology website

### 1.3.2.4 Canvas : *Canva*

Canvas is a widely used learning management system by schools and universities. Canvas is an LMS that offers a convenient and easily adopted platform for teachers to run their courses, share content with students, and facilitate interactions between students and instructors regarding course activities. Canvas programs offer students quizzes, assignment submissions, access to their scores and progress, ability to participate in discussion forums, and interoperability with the other education software they use. Thanks to a reliable mobile app and an easy-to-use platform for teachers to take into account, Canvas is one of the most common options for schools that are moving towards online or hybrid learning a screenshot is provided in Figure 1.4. (Instructure inc, 2011)

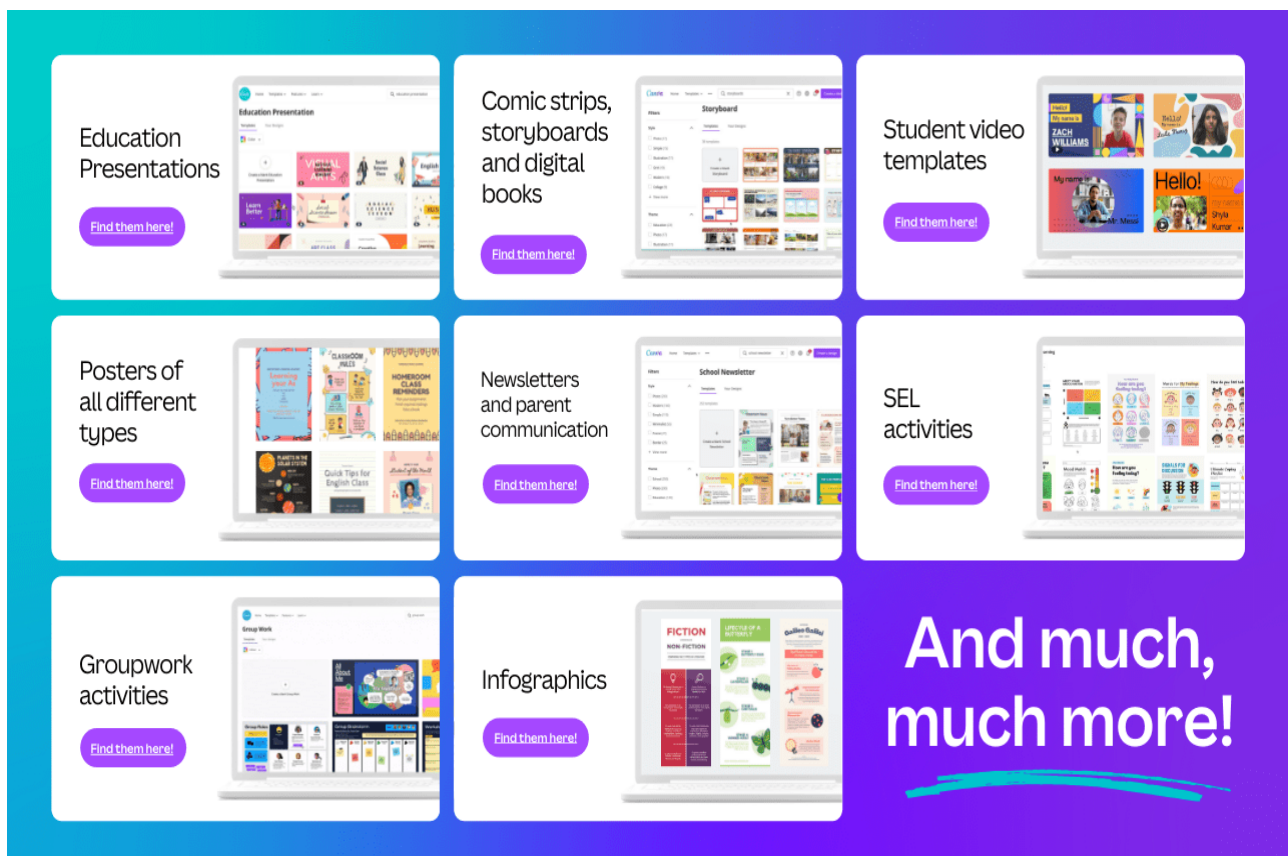


FIGURE 1.4 – canva website utilities

## 1.4 Conclusion

This chapter has discussed the amalgamation of the features of peer learning with the e-learning principles to point out their significance in effective changes in online education. The course design principles under the e-learning principles would entail personalizing, interactivity, accessibility, flexibility, and effective feedback. Instructors must understand the e-learning principles for course design so that, in light of personalizing, interactivity, accessibility, flexibility, and effective feedback, more engaging and practical courses can be designed. There are innumerable content-delivery modes nowadays, including text-based content, multimedia presentations, interactive modules, live lectures, and so on, making educational content presentations diverse to address different learning styles.

Support is fostered by creating an interactive learning environment among peer learning features, such as peer-to-peer tutoring and tools for collaborative problem-solving. The use of matching algorithms, instant messaging, voice and video, sharing files, scheduling of sessions, and feedback systems only strengthen the experience of collaborative learning. Examples of the current state-of-the-art in the development of personalized learning environments are Moodle, Stack Overflow, Schoology, and Canvas, which serve as one of the best sources of insight and study resources for teaching.



## CHAPTER TWO:

# Analysis and UML Modeling

### INTRODUCTION

2.2. Section One: Design Methodology

2.3. Section Two: Analysis and Design

2.4. Section Three: Database SCHEMA

### CONCLUSION

## 2.1 Introduction

The development of a system requires modeling that allows for anticipating, predicting, and studying the information related to that system. To accomplish this, we have chosen the UML language, which allows for the representation of graphical concepts and the modeling of applications. This UML modeling illustrates the different actors in the system as well as the roles they may hold. (Fowler M., 2004, )

## 2.2 Design Methodology :

In the following, we will present the UML language.

### 2.2.1 Presentation of UML

UML, or "Unified Modeling Language," is an object-oriented modeling language developed in response to a call for proposals issued by the OMG with the goal of defining a standard notation for modeling applications built using objects and also for software design. UML is a visual language consisting of a set of diagrams, called diagrams, each providing a different view of the project to be addressed. UML thus provides us with diagrams to represent the software to be developed : its operation, its implementation, the actions that may be performed by the software, etc. (Booch M., 2004)

## 2.3 Analysis and Design

### 2.3.1 Use Case Diagram :

The Use Case Diagram is used for modeling user needs. Use cases describe the behavior of the system from the user's perspective and the possibilities for functional interactions between the system and the actors. They help to define the boundaries and relationships between the system and its environment

#### 2.3.1.1 Role of the Use Case Diagram :

1. **Visual Representation of System Interactions :** Use case diagrams simplify the system's interactions with external entities, helping non-technical stakeholders

understand its functionalities. Clarification of System Boundaries : These diagrams clearly outline the system's scope, distinguishing between internal operations and external influences, crucial for accurate requirements gathering.

2. **Identification of User Roles** : They identify the different user types interacting with the system, aiding in understanding user needs and designing role-specific functionalities. Overview of System Functionalities : Use case diagrams list all possible actions available to users, providing a complete overview of what the system is designed to perform.
3. **Aid in System Design and Analysis** : Serving as a foundational tool for detailed design and analysis, these diagrams help translate user interactions into precise system functionalities.
4. **Support for User-Centered Development** : By focusing on user actions and interactions, use case diagrams ensure the development process remains aligned with user expectations and needs.

### 2.3.1.2 Components of a Use Case Diagram

The basic components of use case diagrams are the actor, the use case, and the association.

- **Actor** : An actor is a user who communicates and interacts with the system's use cases. This entity behaves like a person or another system and is involved in various interactions within the system.
- **Use Case** : A use case represents a functionality provided by the system, typically described in a verb-object format (for example, "register car," "delete user"). Use cases are depicted as ellipses containing their names, illustrating the actions that can be taken within the system.
- **Association** : Associations are used to link actors with use cases. They indicate that an actor participates in a use case in some form. These relationships are represented by lines connecting actors to their respective use cases.

### 2.3.1.3 Use Case Diagram :

The use case diagram for our application is structured as follows in FIGURE 2.1 :

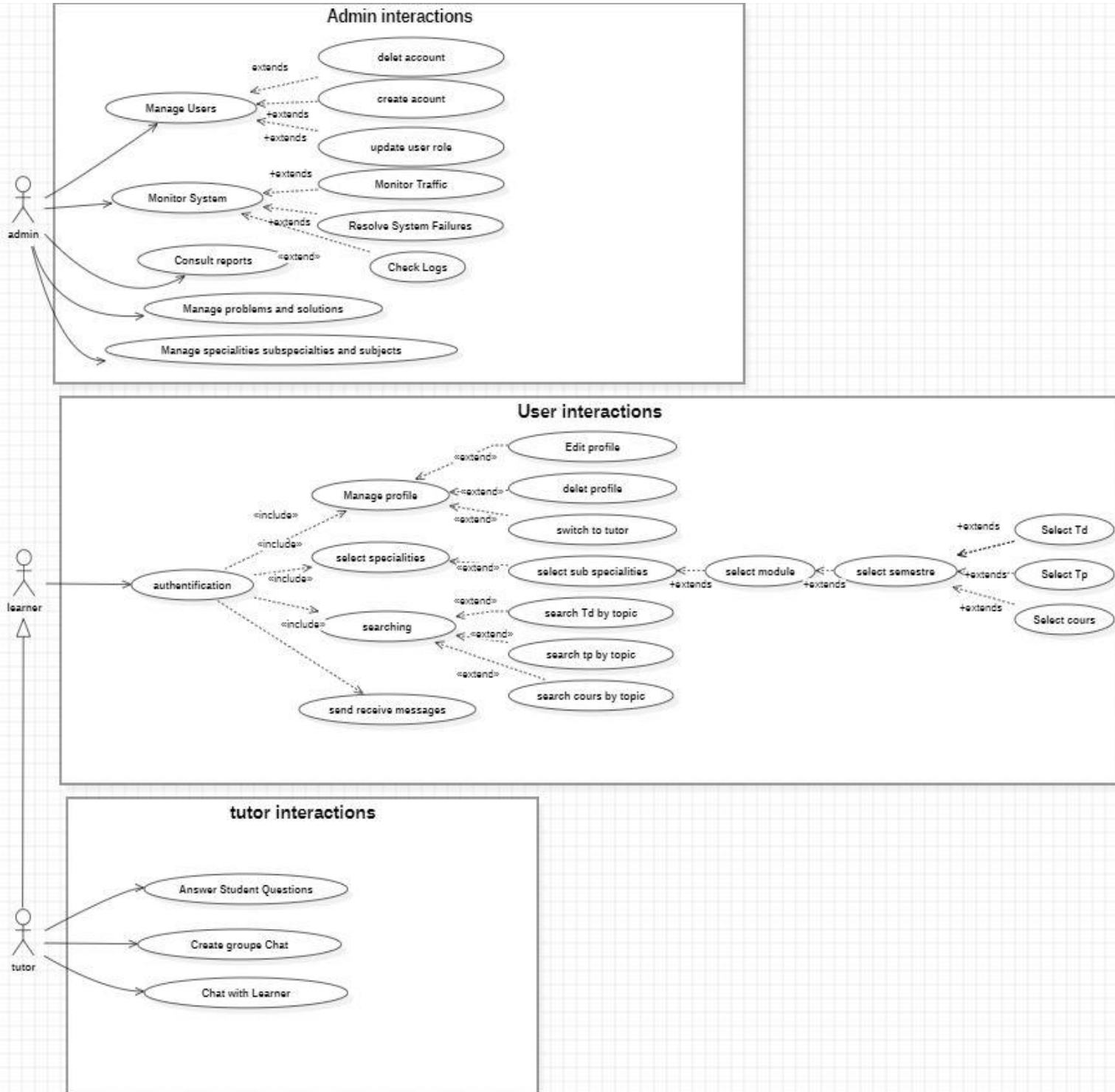


FIGURE 2.1 – use case diagram

### 2.3.2 class diagram

After the detailed study of the use cases, we deduced the overall class diagram of the system FIGURE 2.2, This diagram is considered the final phase of the theoretical design of our system and will be taken as the reference from which will take place the software development, and the writing of the source code of our application.

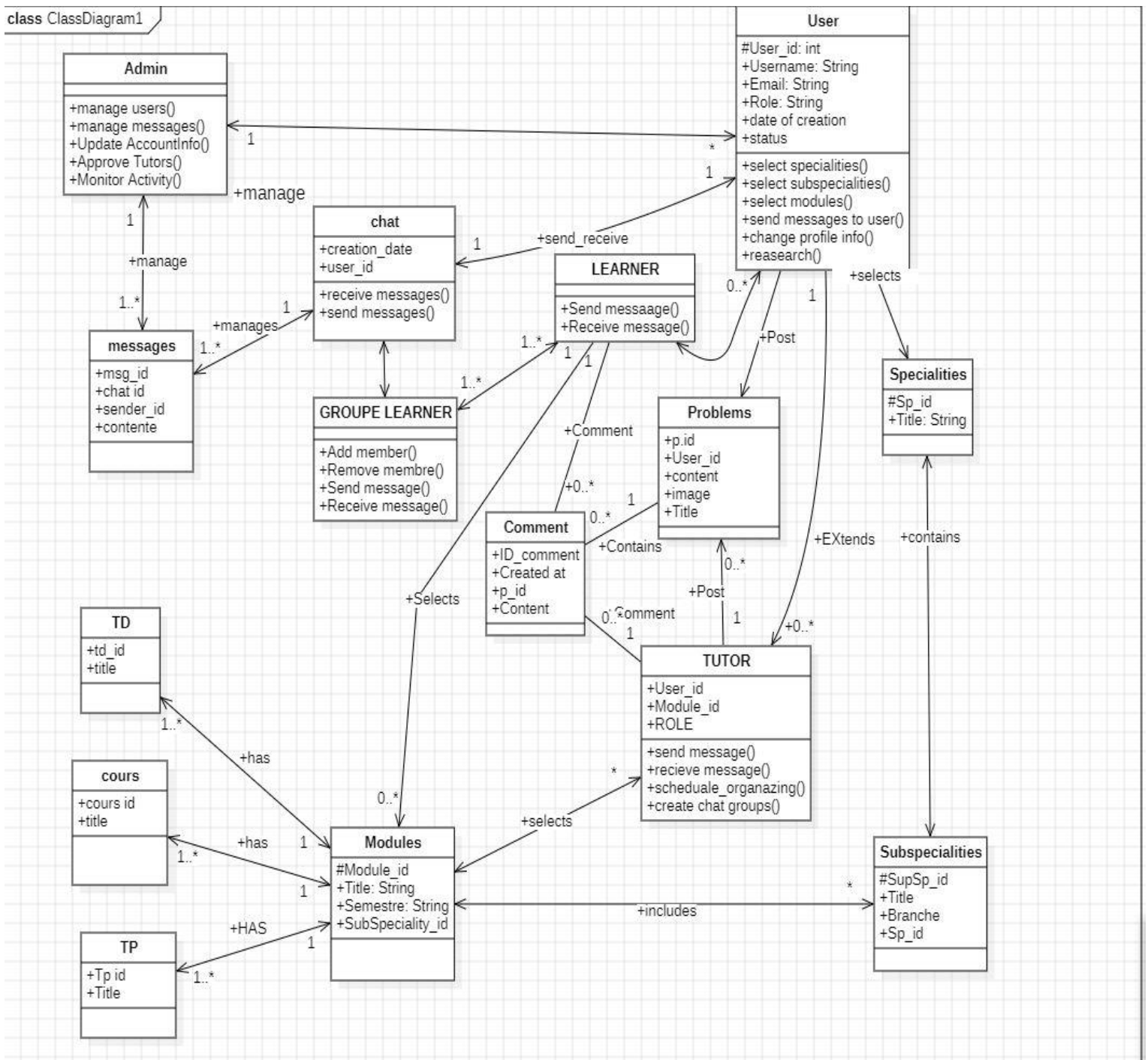


FIGURE 2.2 – class diagram

## 2.3.3 sequence diagramme

### 2.3.3.1 Authentication sequence diagram

Authentication consists in ensuring the confidentiality of the data, it is based on login and password verification. This information is pre-established in a database. When authenticating the user, two cases may occur : correct or incorrect information, which explains the use of the «alt» operator. If the information provided is correct, then the system grants access to the appropriate interface. On the other hand, if the user enters incorrect information, the system generates an error message and redisplay the authentication page where the «loop» operator is used as explained in FIGURE 2.3.

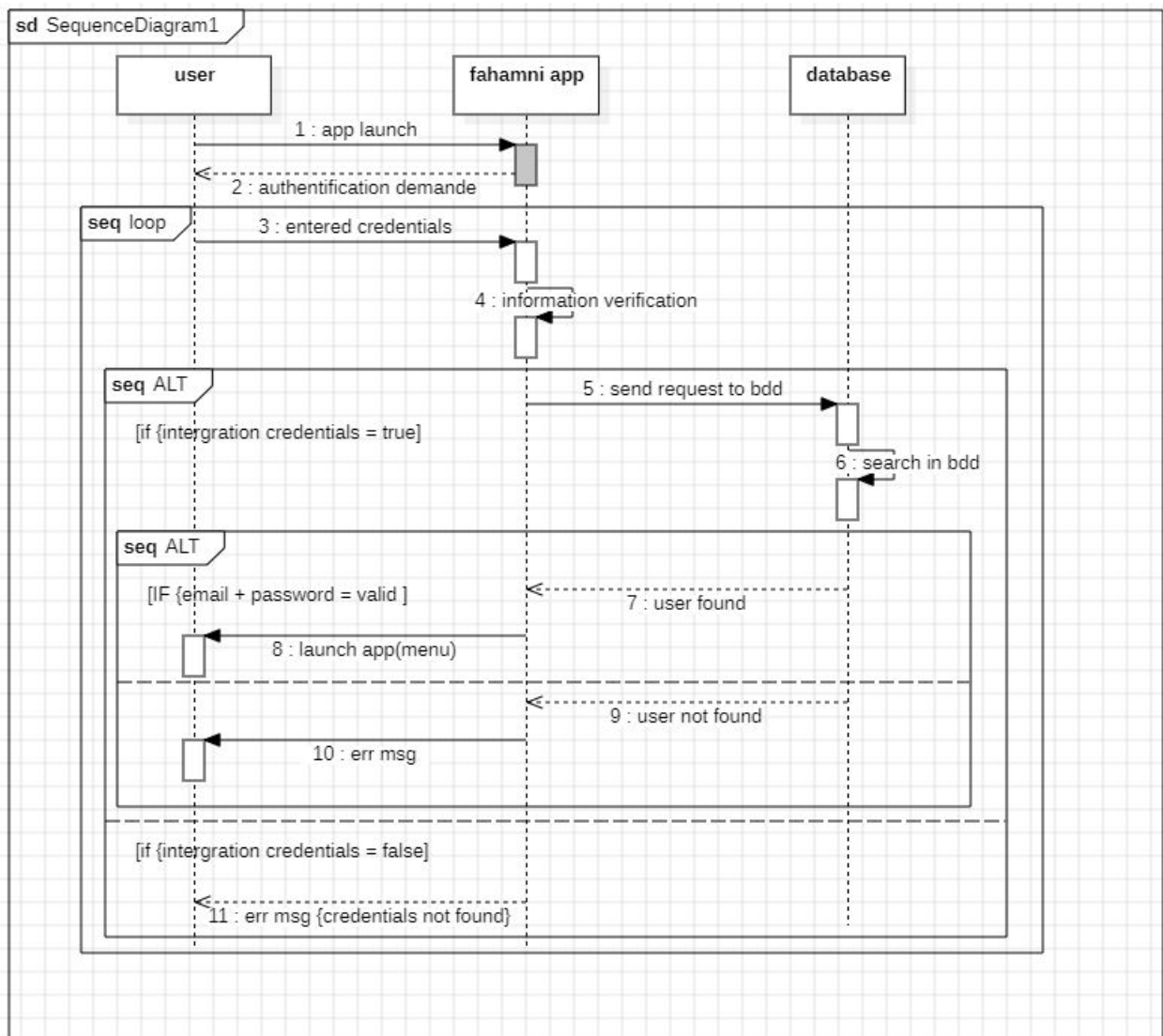


FIGURE 2.3 – sequence Diagram of AUTHENTICATION

### 2.3.3.2 Sequence Diagram of Posting Problem:

This sequence diagram now focuses solely on the interactions between the users, the Fahemni app, and the database for posting problems and submitting solutions.

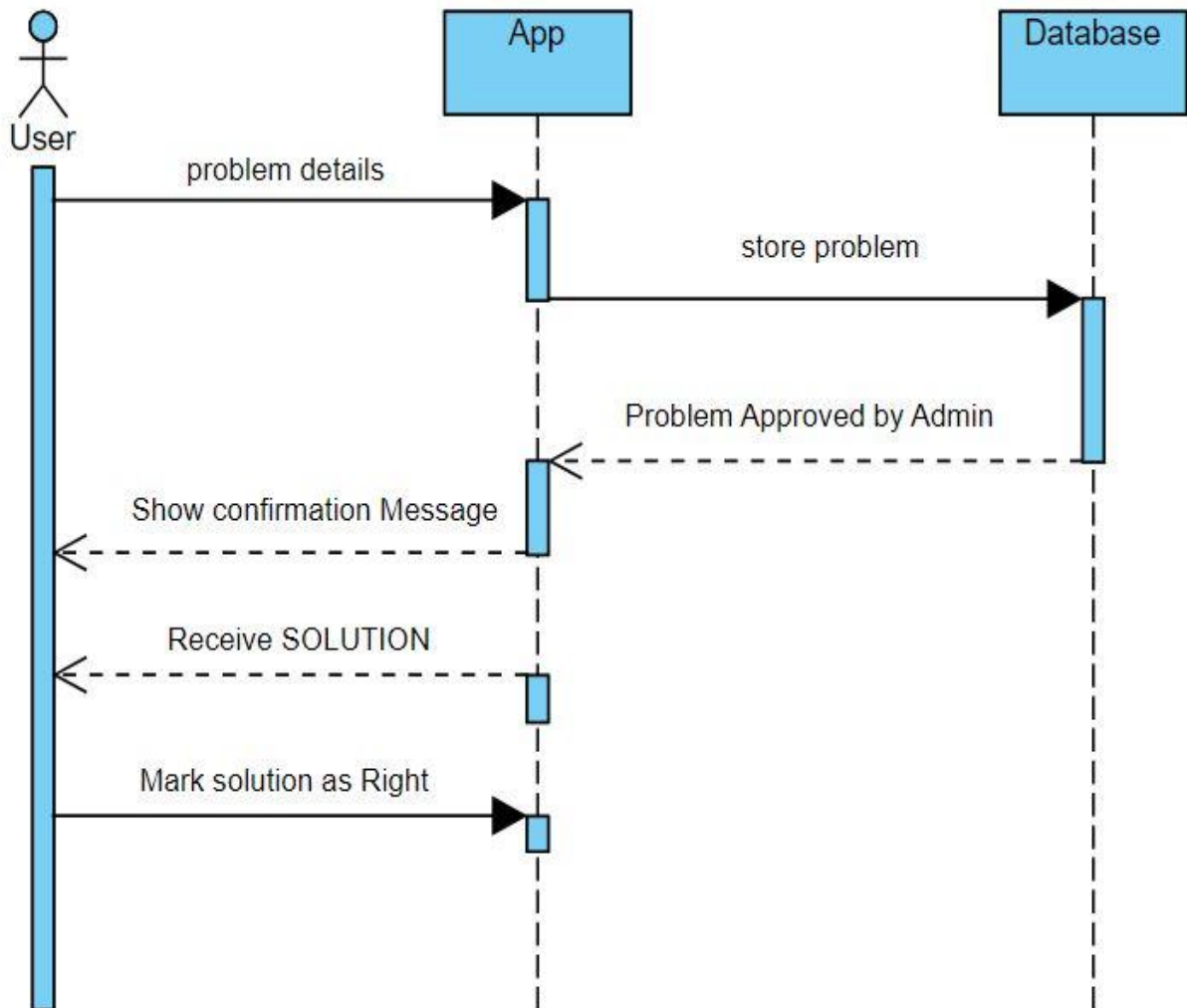


FIGURE 2.4 – sequence Diagram of Post Problem

## 2.4 Database SCHEMA

Open-source platform for a serverless backend as a service, Supabase, puts developers at the forefront with a full-featured toolkit to build applications. Central to the Supabase offer is a powerful and flexible database management system built on PostgreSQL—an assurance of healthy and scalable data storage, easing the process of integration with modern web and mobile application development. In the figure 2.4 we will see a general schema for our DB that shows branches between various tables.

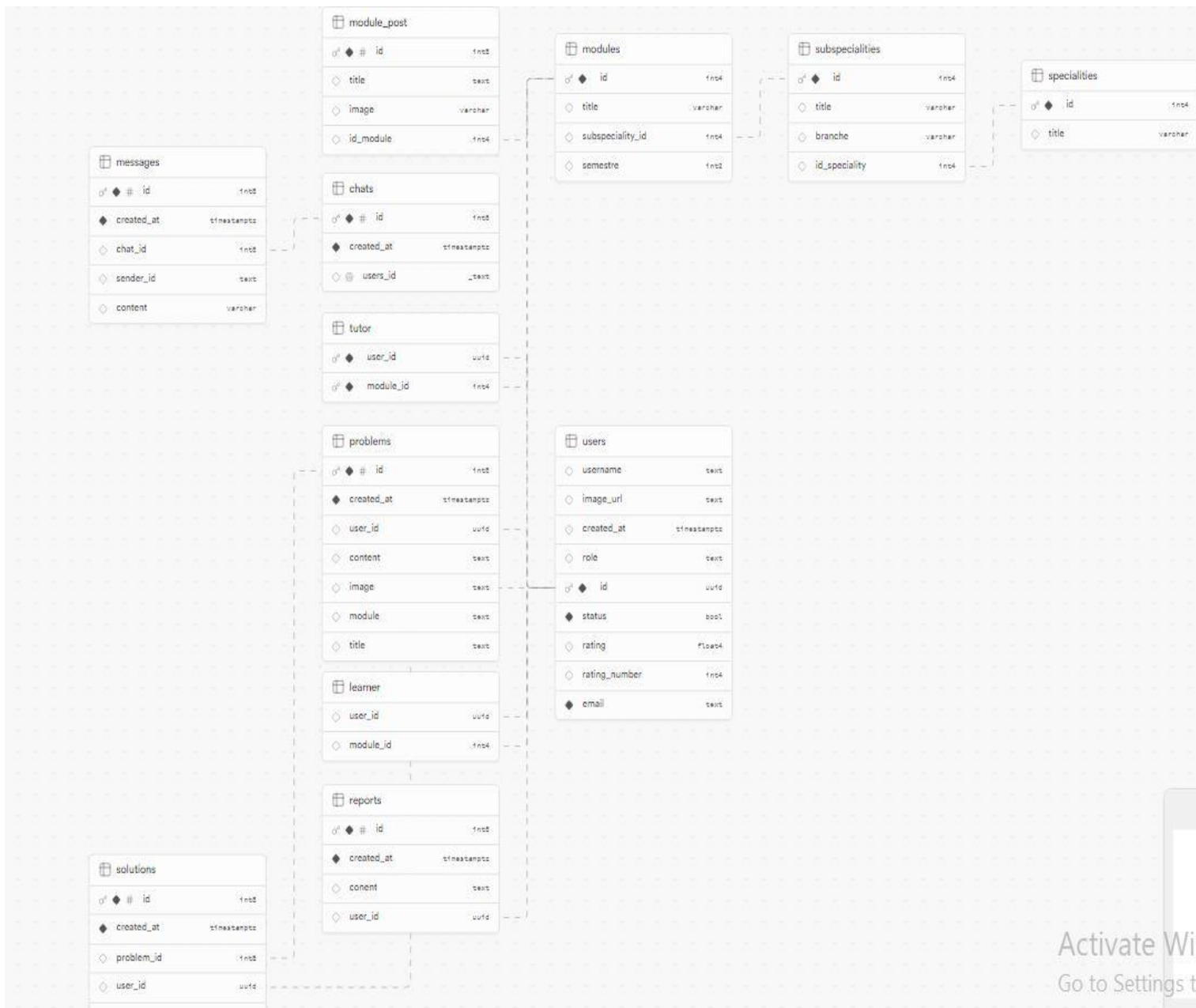


FIGURE 2.4 – Database Schema



## 2.5 Conclusion

In This chapter, we showcased the design methodology used for developing our system, focusing on the role of UML (Unified Modeling Language) in visualizing and modeling system components and interactions. We explored various UML diagrams, including use case diagrams, class diagrams, and sequence diagrams, to clearly depict the system's functionality, user interactions, and overall structure. This modeling approach is vital for accurately defining system requirements and ensuring a user-centered design.

Additionally, we highlighted the importance of a robust database schema, utilizing Supabase, to ensure scalable and efficient data management for modern web and mobile applications. These design and modeling efforts are fundamental in setting the stage for the next phases of software development and implementation, providing a solid foundation for building a functional and user-friendly system.

# CHAPTER THREE:

## Application Implementation

### INTRODUCTION

3.2. Section One: Work Environment

3.3. Section Two: Admin Panel

3.4. Section Three: Presentation of the Interfaces of Our  
Application

3.5 Section Four : Website

### CONCLUSION

## 3.1 Introduction :

This Chapter dives into the design of the user interface (UI) and the various functional components of the application. It thoroughly explains different interfaces, such as the user list page, subject selection page, group creation interfaces, and the tutor sign-up page. The chapter also discusses the rating system for tutors and how the website integrates with Hostinger for hosting services. These features are essential for creating a smooth and interactive user experience.

## 3.2 Work Environment :

### 3.2.1 Hardware Environment :

For the realization of our project, we used an ASUS computer characterized by :

- Operating system : Windows 10.
- Processor : IntelRCore™ (i7-7th Hq).
- RAM : 16GB.
- Hard drive : 750 GB SSD.

For the different stages of testing, installation and deployment of the application we needed a mobile terminal supporting the Android operating system whose characteristics are as follows :

- Device name : Samsung S10 5g.
- Operating system : Android.
- Connection : 4G, ADSL.
- RAM : 8GB.
- Hard drive : 256GB.

## 3.2.2 Software Environment :

### 3.2.2.1 Flutter Framework for Mobile App Development



Flutter is an open-source UI software development kit created by Google, primarily used for crafting natively compiled applications for mobile, web, and desktop from a single codebase. It provides a rich set of pre-designed widgets and tools that enable developers to build beautiful and responsive user interfaces. Flutter uses the Dart programming language and employs a reactive programming paradigm, allowing for fast development cycles and efficient performance. This section will provide an overview of Flutter's origins, its core principles, and its role in modern app development landscapes. (*The Official Flutter Website, 2022*)

### 3.2.2.2 MVC In Flutter :

MVC stands for model view controller, and its main work is to have a segregated code base. It strives to separate the software's code and area of responsibility. The core part of the model view controller is to separate the project's interface from the functionality and data embraced into the application (Abhishek, 2024).

As we know, in the case of all of us, tidy and arranged things are always more straightforward to use and maintain; the same way with the codebase, MVC makes the codebase into a neat and arranged form, and it does these through three of its components:

- **Model:** The model is a compromise of a data source; it may be from a DB, an API, a JSON, etc. In some cases, it may consist of some business logic.
- **View:** The view section represents the user interface, displaying data and taking input from the user.
- **Controller:** In this section, the business logic resides; it controls what data will be shown to the user and how user input will be handled.

### Communication Flow :

Differentiating the three primary elements in projects helps developers write clean and modular code, making the code easily reusable and helpful in parallel development.

Therefore, it becomes straightforward to work on the project because it does not affect others if something is changed in one part. So, keeping them separated needs a communication flow in which they will interact with each other and achieve the functioning of the project.

As we can see from the following flow diagram, the User interacts with the View part of the project, which helps the user see the Data and provide input to the Controller through the View. The active controller, acting as the system's brain, carries out the calculations and manipulations from the user data. It takes data from the model to operate and, subsequently, presents it to the view, which depicts the desired data to the user.

Here, we find that the View shows and receives the data but does not interact directly with the Model; the Controller acts as an intermediary between the view and model in the manipulation of data and the representation of data. The View and the Model are tightly bound to the Controller: it knows the View knows how to communicate with the Controller, the Controller knows how to communicate with the Model, and the Model and View work collectively without knowing about each other's existence. figure 3.1 showcases a typical lifecycle in an mvc app.

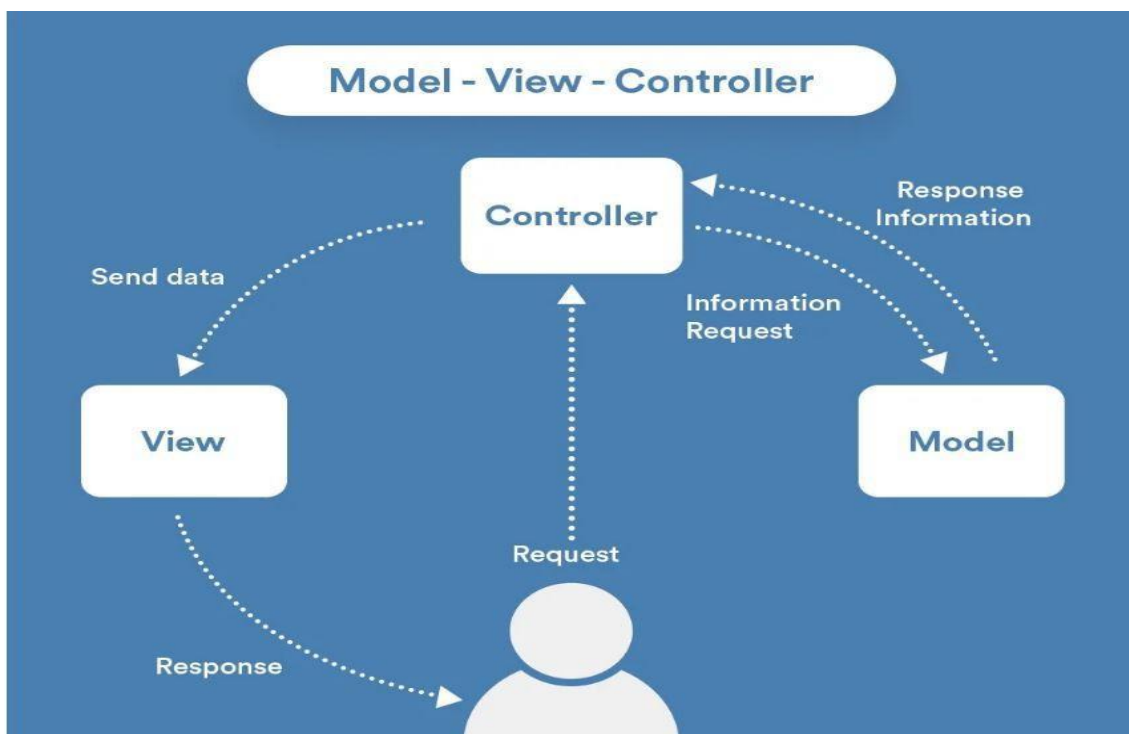


Figure 3.1- MVC

### 3.2.2.3 Advantages of Using Flutter :

Flutter offers several advantages that make it a compelling choice for mobile app development :

- **Cross-Platform Development** : Write code once and deploy it across multiple platforms, saving time and effort.
- **Hot Reload** : Instantly see changes made in the code reflected in the app, speeding up development iterations.
- **Rich Widgets** : Utilize a wide range of customized widgets to create visually appealing and interactive user interfaces.
- **High Performance** : Compile to native machine code for fast and fluid app performance on various devices

PostgreSQL is a robust open-source relational database management system renowned for its reliability and extensive feature set. It follows the relational model, ensuring efficient data organization. With ACID compliance, it guarantees data integrity. PostgreSQL's extensibility allows for customization through a vast array of extensions. Supported by a thriving community, PostgreSQL is a trusted choice for diverse applications.

### 3.2.2.4 Data Modeling and Schema Design :

Data modeling in PostgreSQL involves structuring data in a way that reflects real-world relationships and entities. Schema design defines the structure of tables, columns, and relationships within the database. This process ensures efficient data storage and retrieval, maintaining data integrity and facilitating query optimization. Effective data modeling and schema design are crucial for building scalable and maintainable database systems.

### 3.2.2.5 Supabase : supabase

The short and simple introduction to Supabase comes down to its core services : database, authentication, file storage, and auto-generated APIs. In other words, this platform promises

to offer everything you need to build a fully functional app

### Supabase vs Firebase :

The team behind Supabase strives to be just as good as, if not better than, Google's Firebase. However, Supabase is still in its infancy. While Firebase has received years of support and updates from Google, which is arguably the biggest tech giant globally, Supabase is open-source software that's been live for less than two years.

Recognizing that Supabase is still relatively new is important because it's still growing. Whereas Firebase has a large user base and lots of support, which in turn makes it trustworthy and easy to integrate, Supabase is still trying to find its place in the market.

It provides all the backend services you need to build a product. You can use it completely, or just the services you require.

The main difference is that Firebase provides NoSql databases which is would not be efficient for our case because of the number and relations between the tables we have,while Supabase provides a relational database Postgres which is perfect for our case



#### 3.2.2.6 Visual Studio Code :

#### Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, MacOS and Linux. It comes with built-in support for JavaScript, Typescript and Node.js and has a rich ecosystem of extensions for other languages and run-times such as( C++, 'C', Java, Python, PHP, Go, .NET) Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like Intelligence code completion and debugging. First and foremost, it is an editor that gets out of your way. The de- lightly friction-less edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas

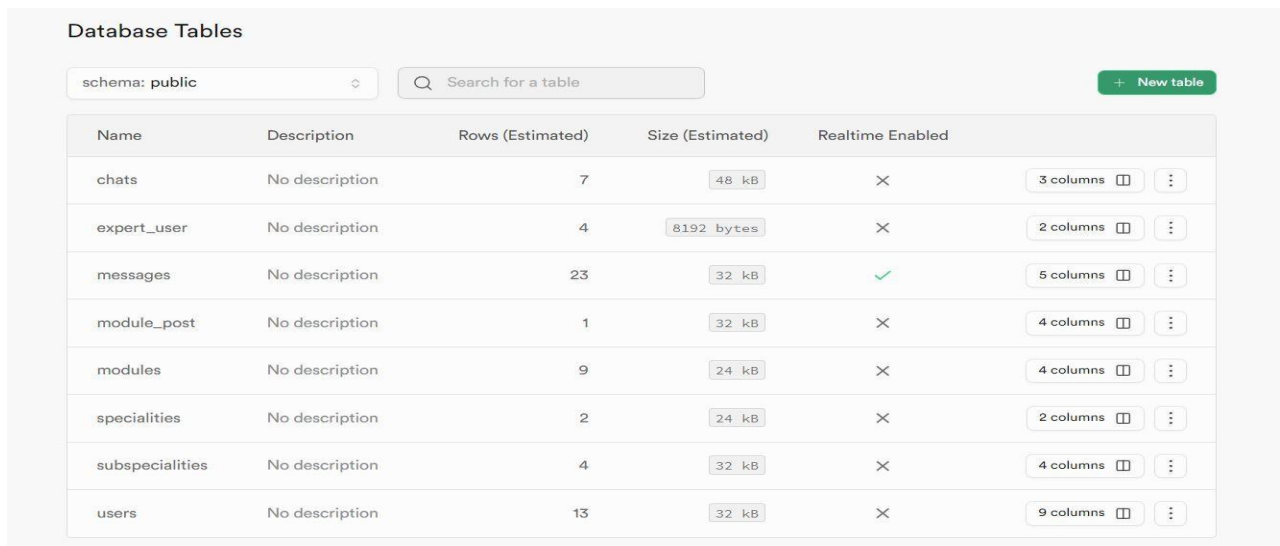
## 3.3 Admin Panel

The admin panel is a crucial component of our application, providing administrators with the tools to manage and oversee various aspects of the system. This section outlines the key functionalities of the admin panel, highlighting how administrators can add tables, manage

users, monitor activity, manage content, and handle specialties using Supabase. The following subsections delve into each of these functionalities in detail :

### 3.3.1 Tables in Supabase

Adding tables to Supabase is relatively very easy and does a good job of letting administrators structure a database in whatever way best fits the application. Admins can create new tables by specifying the table name, the columns, and the respective data types for those columns, all this using the Supabase interface. With the addition of new features in the future to the application, it ensures the possibility of further building a database schema and ensures strong grounds of storing and retrieving data examples in Figure 3.2.



Name	Description	Rows (Estimated)	Size (Estimated)	Realtime Enabled	
chats	No description	7	48 kB	×	3 columns [ ] [ ⋮ ]
expert_user	No description	4	8192 bytes	×	2 columns [ ] [ ⋮ ]
messages	No description	23	32 kB	✓	5 columns [ ] [ ⋮ ]
module_post	No description	1	32 kB	×	4 columns [ ] [ ⋮ ]
modules	No description	9	24 kB	×	4 columns [ ] [ ⋮ ]
specialities	No description	2	24 kB	×	2 columns [ ] [ ⋮ ]
subspecialities	No description	4	32 kB	×	4 columns [ ] [ ⋮ ]
users	No description	13	32 kB	×	9 columns [ ] [ ⋮ ]

FIGURE 3.2 – Table operations

### 3.3.2 Managing Users

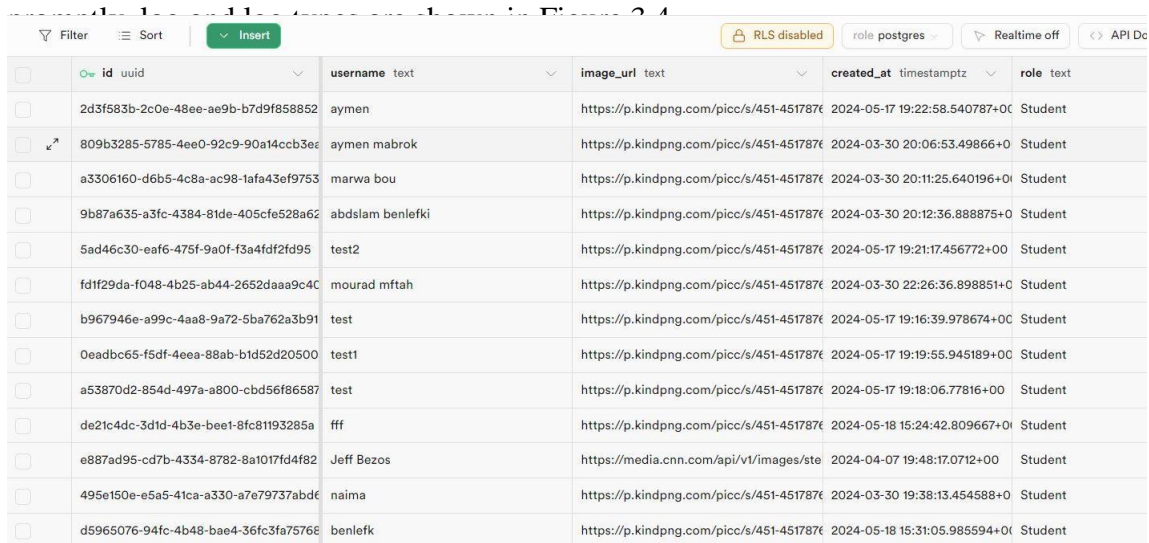
Managing users in Supabase is a crucial aspect of maintaining a well-functioning and secure application. Supabase provides a user-friendly interface and robust functionalities to help administrators effectively manage user accounts as shown in Figure 3.3.

### 3.3.3 Check Request Log

Monitoring request logs is crucial for understanding and troubleshooting application behavior. By checking the request log, administrators can gain insights into the API requests



made to the Supabase backend. By regularly checking the request log in Supabase, administrators can ensure the smooth operation of their applications and address any issues

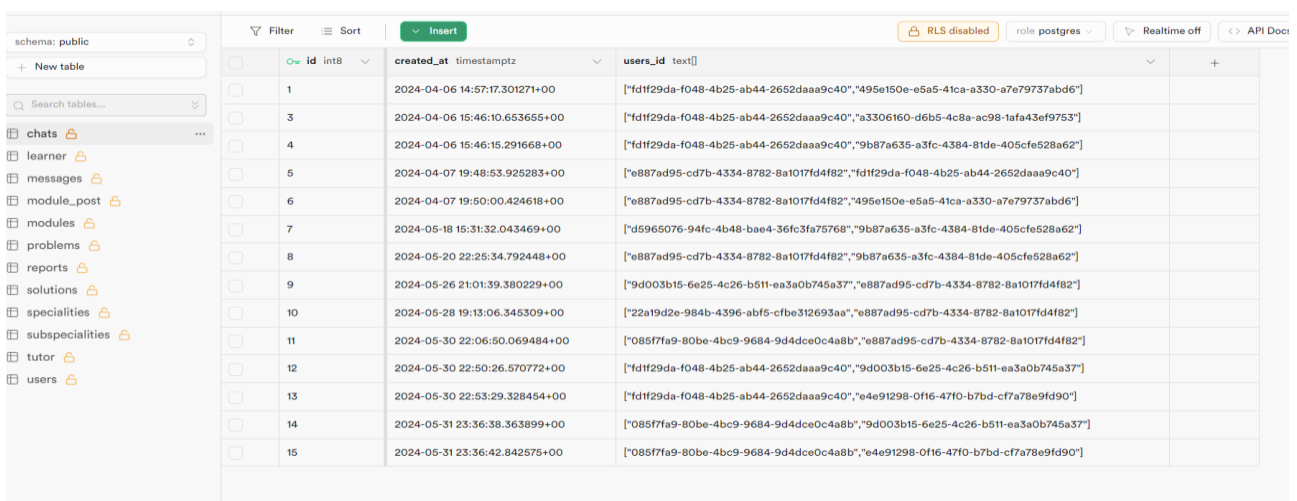


id	username	image_url	created_at	role
2d3f583b-2c0e-48ee-ae9b-b7d9f858852	aymen	https://p.kindpng.com/picc/s/451-451787	2024-05-17 19:22:58.540787+00	Student
809b3285-5785-4ee0-92c9-90a14ccb3e3	aymen mabrok	https://p.kindpng.com/picc/s/451-451787	2024-03-30 20:06:53.498660+00	Student
a3306160-d6b5-4c8a-ac98-1afa43ef9753	marwa bou	https://p.kindpng.com/picc/s/451-451787	2024-03-30 20:11:25.640196+00	Student
9b87a635-a3fc-4384-81de-405cfe528a62	abdslam benlefki	https://p.kindpng.com/picc/s/451-451787	2024-03-30 20:12:36.888875+00	Student
5ad46c30-eaf6-475f-9a0f-f3a4df2fd95	test2	https://p.kindpng.com/picc/s/451-451787	2024-05-17 19:21:17.456772+00	Student
fd1f29da-f048-4b25-ab44-2652daaa9c40	mourad mftah	https://p.kindpng.com/picc/s/451-451787	2024-03-30 22:26:36.898851+00	Student
b967946e-a99c-4aa8-9a72-5ba762a3b91	test	https://p.kindpng.com/picc/s/451-451787	2024-05-17 19:16:39.978674+00	Student
0eadbc65-f5df-4eea-88ab-b1d52d20500	test1	https://p.kindpng.com/picc/s/451-451787	2024-05-17 19:19:55.945189+00	Student
a53870d2-854d-497a-a800-cbd56f86587	test	https://p.kindpng.com/picc/s/451-451787	2024-05-17 19:18:06.77816+00	Student
de21c4dc-3d1d-4b3e-bee1-8fc81193285a	fff	https://p.kindpng.com/picc/s/451-451787	2024-05-18 15:24:42.809667+00	Student
e887ad95-cd7b-4334-8782-8a1017fd4f82	Jeff Bezos	https://media.cnn.com/api/v1/images/ste	2024-04-07 19:48:17.0712+00	Student
495e150e-e5a5-41ca-a330-a7e79737abd6	naima	https://p.kindpng.com/picc/s/451-451787	2024-03-30 19:38:13.454588+00	Student
d5965076-94fc-4b48-bae4-36fc3fa75768	benlefk	https://p.kindpng.com/picc/s/451-451787	2024-05-18 15:31:05.985594+00	Student

FIGURE 3.3 – Manage users interface

### 3.3.4 Managing Content

Supabase Content Management provides an easy-to-use, easy-to-navigate interface where administrators handle data in order. Apart from deleting records from the database, Supabase interface also provides tools for adding, reading, and updating the record. Role-based access control, real-time data updates, automated backups, and other features will have been put in place to assure integrity and safety in the data stored. Further, its interaction with various third-party services ensures supplication of a variety of functionalities, making it a versatile solution in fulfilling each type of application need.



id	created_at	users_id
1	2024-04-06 14:57:17.301271+00	["fd1f29da-f048-4b25-ab44-2652daaa9c40","495e150e-e5a5-41ca-a330-a7e79737abd6"]
3	2024-04-06 15:46:10.653655+00	["fd1f29da-f048-4b25-ab44-2652daaa9c40","a3306160-d6b5-4c8a-ac98-1afa43ef9753"]
4	2024-04-06 15:46:15.291668+00	["fd1f29da-f048-4b25-ab44-2652daaa9c40","9b87a635-a3fc-4384-81de-405cfe528a62"]
5	2024-04-07 19:48:53.925283+00	["e887ad95-cd7b-4334-8782-8a1017fd4f82","fd1f29da-f048-4b25-ab44-2652daaa9c40"]
6	2024-04-07 19:50:00.424618+00	["e887ad95-cd7b-4334-8782-8a1017fd4f82","495e150e-e5a5-41ca-a330-a7e79737abd6"]
7	2024-05-18 15:31:32.043469+00	["d5965076-94fc-4b48-bae4-36fc3fa75768","9b87a635-a3fc-4384-81de-405cfe528a62"]
8	2024-05-20 22:25:34.792448+00	["e887ad95-cd7b-4334-8782-8a1017fd4f82","9b87a635-a3fc-4384-81de-405cfe528a62"]
9	2024-05-26 21:01:39.380229+00	["9d003b15-6e25-4c26-b511-ea3a0b745a37","e887ad95-cd7b-4334-8782-8a1017fd4f82"]
10	2024-05-28 19:13:06.345309+00	["22a19d2e-984b-4396-abf5-cfbc312693aa","e887ad95-cd7b-4334-8782-8a1017fd4f82"]
11	2024-05-30 22:06:50.069484+00	["085f7fa9-80be-4bc9-9684-9d4dce0c4a8b","e887ad95-cd7b-4334-8782-8a1017fd4f82"]
12	2024-05-30 22:50:26.570772+00	["fd1f29da-f048-4b25-ab44-2652daaa9c40","9d003b15-6e25-4c26-b511-ea3a0b745a37"]
13	2024-05-30 22:53:29.328454+00	["fd1f29da-f048-4b25-ab44-2652daaa9c40","e4e91298-0f16-47f0-b7bd-cf7a78e9fd90"]
14	2024-05-31 23:36:38.363899+00	["085f7fa9-80be-4bc9-9684-9d4dce0c4a8b","9d003b15-6e25-4c26-b511-ea3a0b745a37"]
15	2024-05-31 23:36:42.842575+00	["085f7fa9-80be-4bc9-9684-9d4dce0c4a8b","e4e91298-0f16-47f0-b7bd-cf7a78e9fd90"]

FIGURE 3.5 – MANAGEMENT CONTENT interface

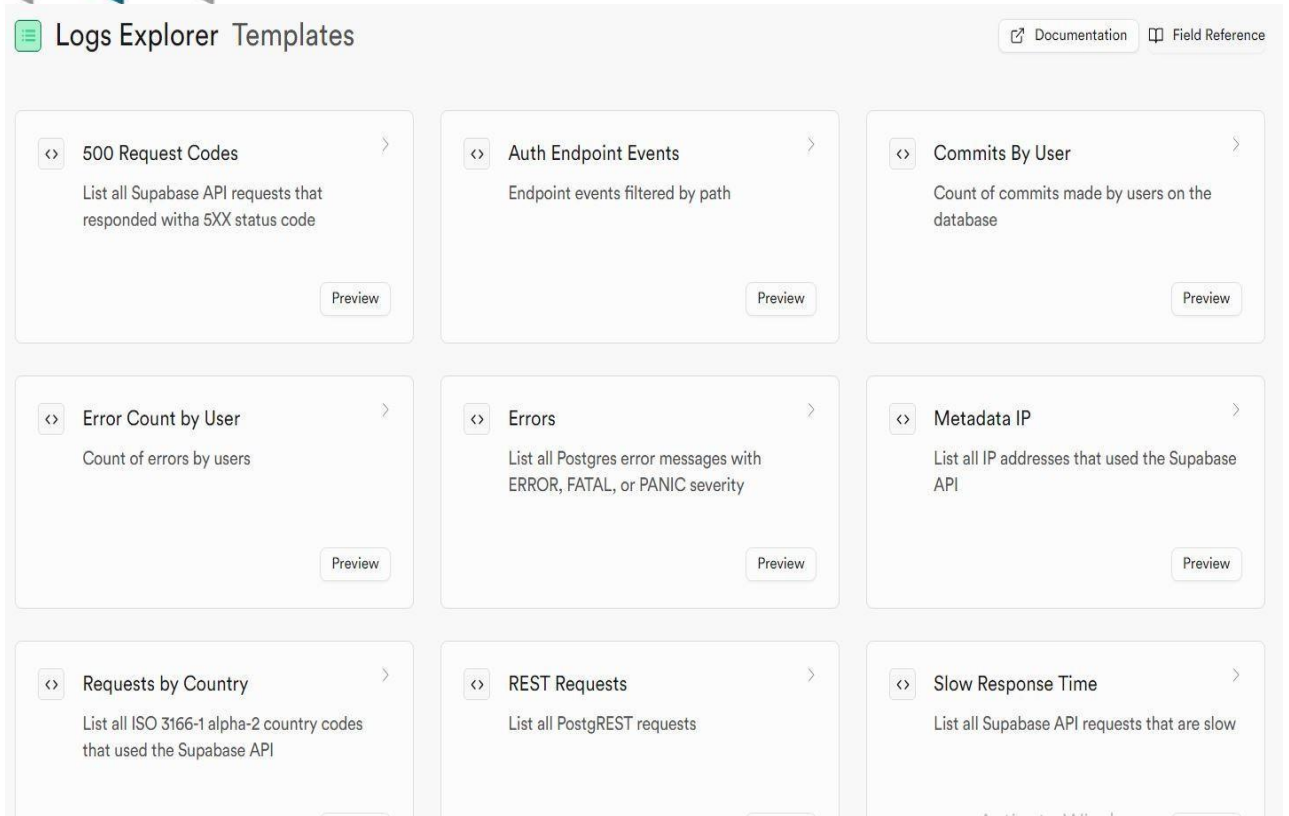


FIGURE 3.4 – checking Logs

### 3.3.5 Statistics

easily keep track of the app usage and requests performed in a specific period of time in the dashboard.

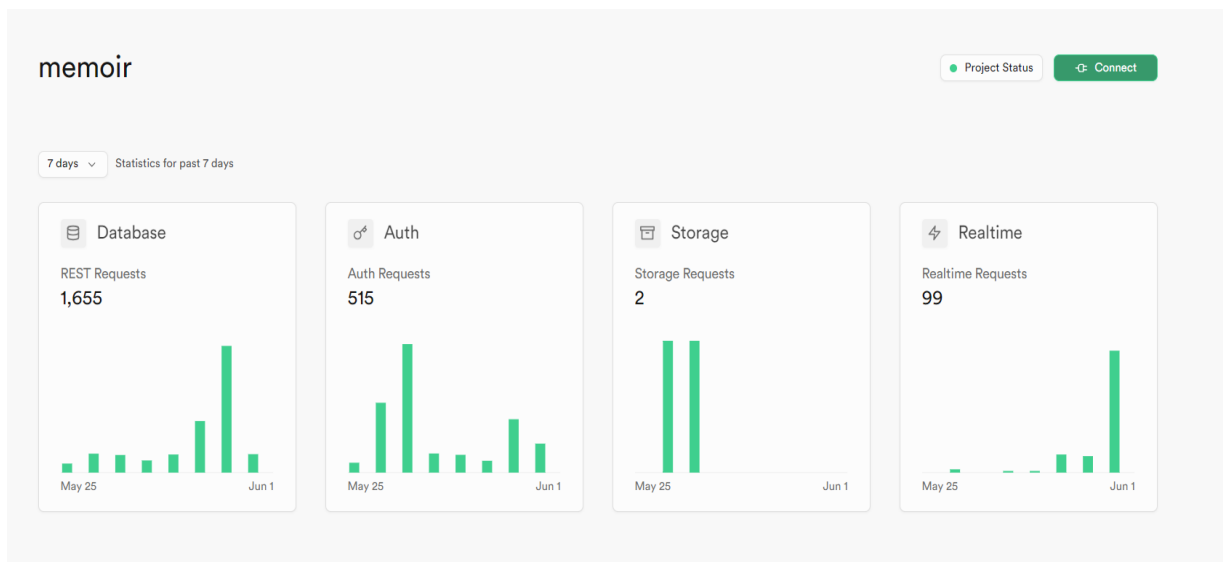


FIGURE 3.6 – Statistics interface

## 3.4 Presentation of the Interfaces of Our Application

### 3.4.1 Splash Interface

Splash interface (FIGURE 3.7) is the gateway of the application that checks if the user authenticates and retrieves most of the application data such as chats and profiles.

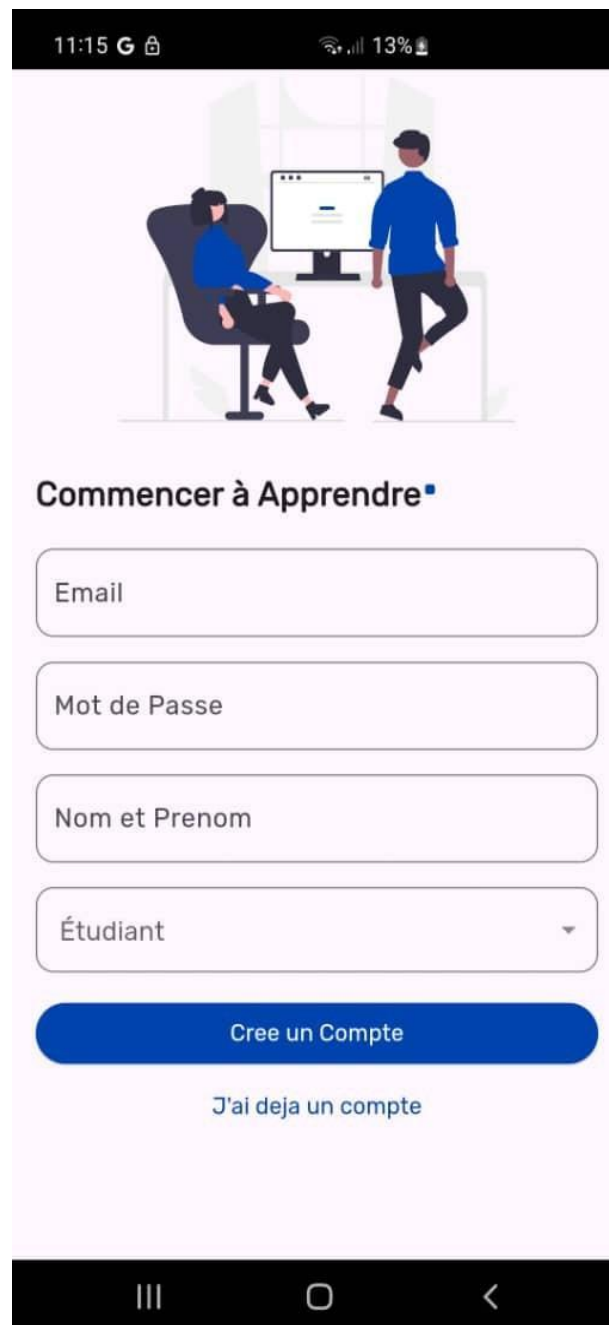


FIGURE 3.7 – splash Interface

## 3.4.2 Authentication Interfaces

### 3.4.2.1 Register Interface

This interface enables new users to create an account (FIGURE 3.8). Users need to provide necessary information such as their name, email, password, and any other required details to register.



11:15 G 🔒 📶 13% 🔋

Commencer à Apprendre

Email

Mot de Passe

Nom et Prenom

Étudiant ▼

Cree un Compte

J'ai deja un compte

FIGURE 3.8 – Register interface

### 3.4.2.2 Login Interface

This Interface allows existing users to log in using their credentials. Users need to enter their email/username as shown in Figure 3.8 and password to access their account.

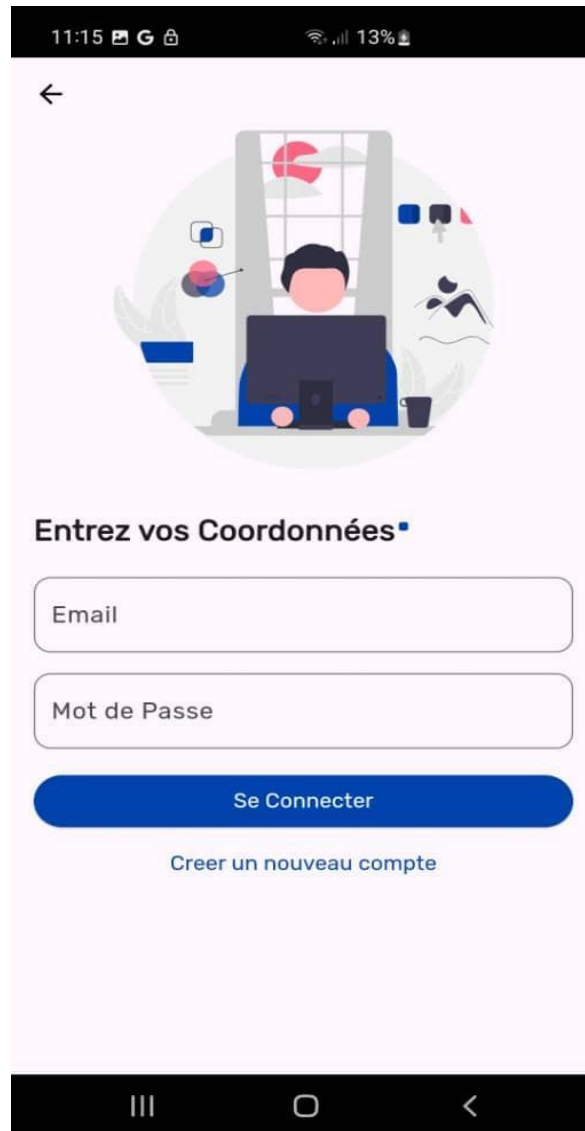


FIGURE 3.9 – Login interface

### 3.4.3 Home Interfaces

The main page of the application contains three sections:

#### 3.4.3.1 News Section

This interface provides information on events and seminars and all kinds of news (FIGURE 3.10).

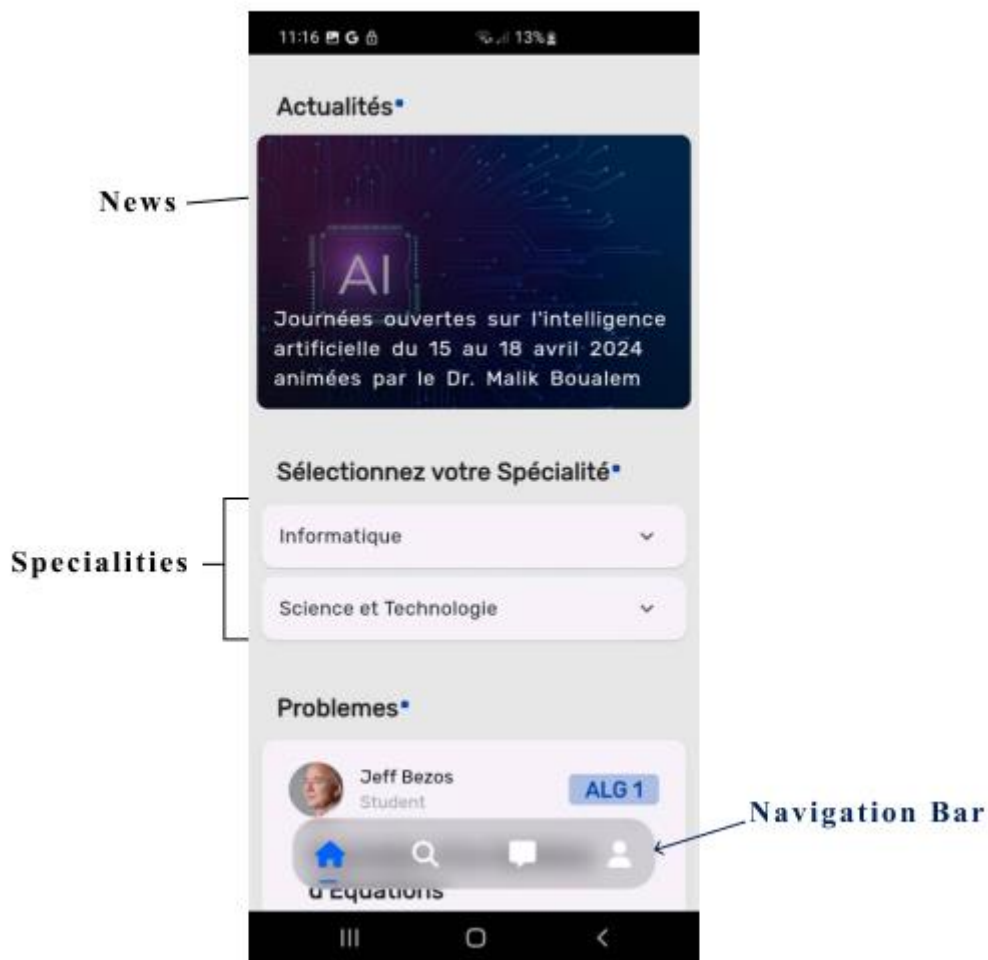


FIGURE 3.10 – News Section Interface

### 3.4.3.2 Specialties Section

This interface lists specialties and subspecialties, organized by their grade(FIGURE 3.11).

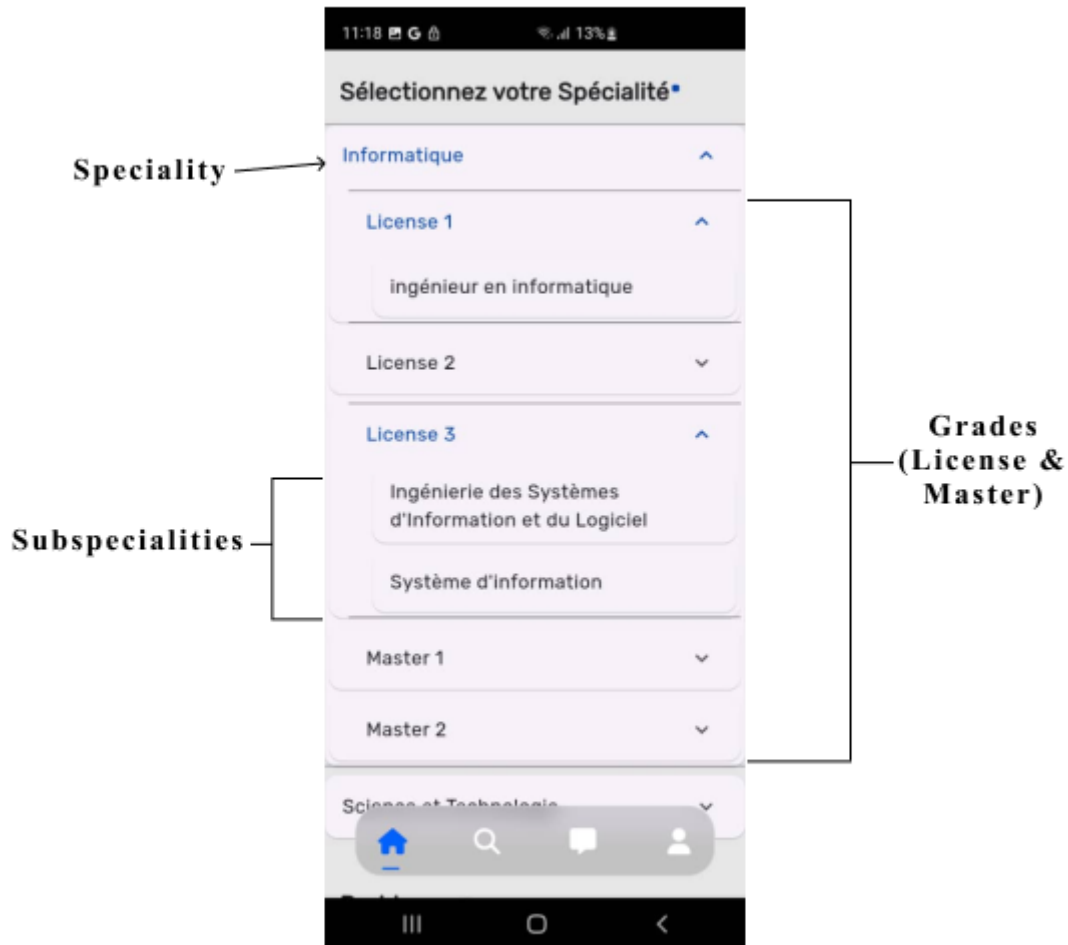


FIGURE 3.11 – Specialties Section

### 3.4.3.3 Problems Section

This section displays a list of posted problems, filtered by date and user's interest in the problem's subject followed by a list of possible answers posted by peers (FIGURE 3.12).

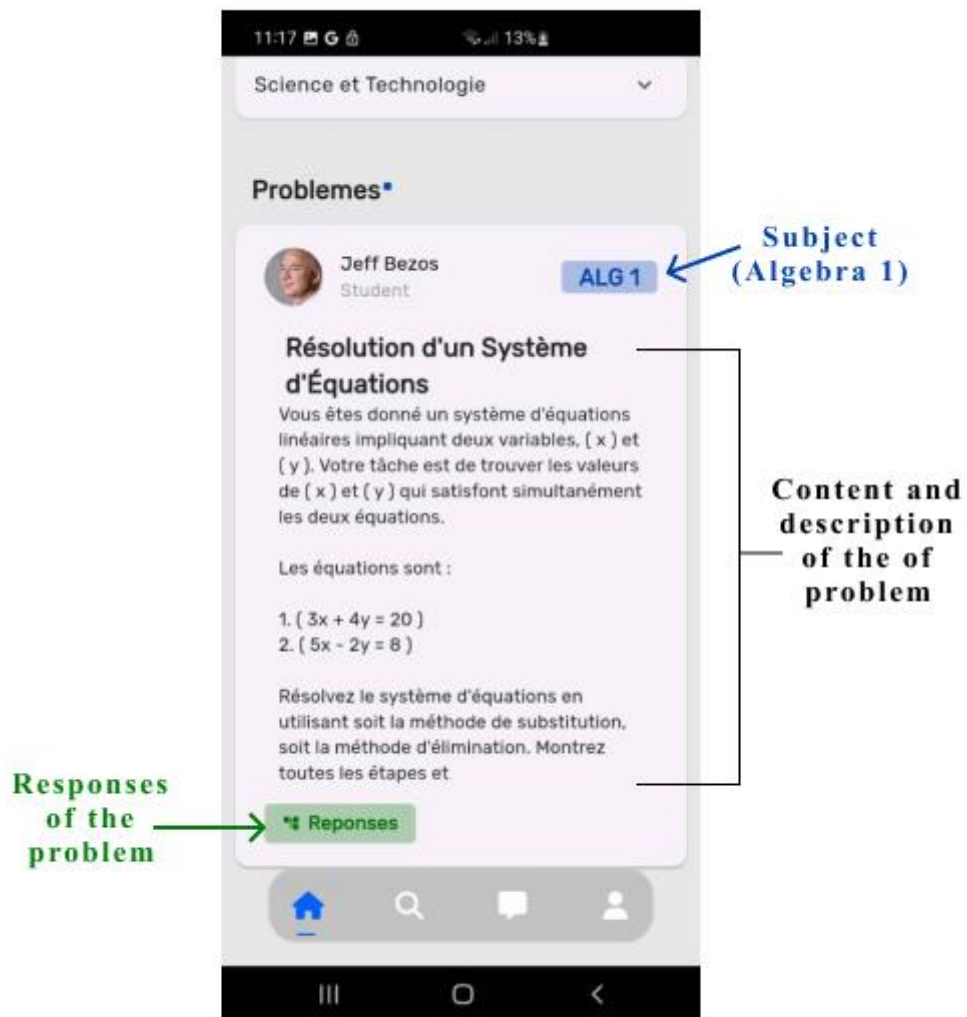


FIGURE 3.12 – Problems Section Interface



### 3.4.4 Search Interface

This interface allows users to search for subjects, specialties, or subspecialties.

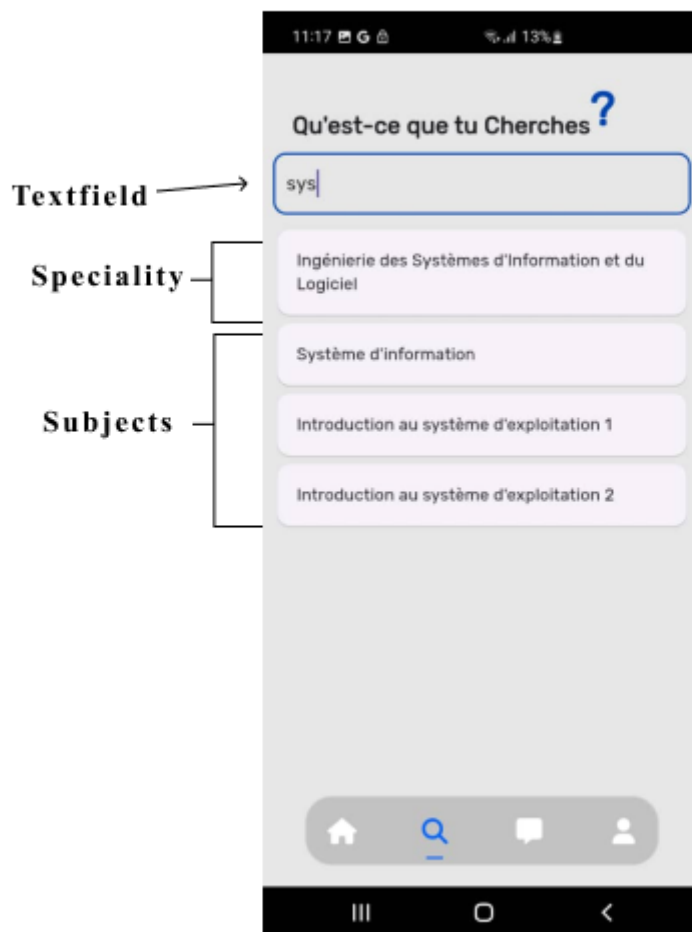


FIGURE 3.13 – Search interface

### 3.4.5 Profile Interface

This interface allows users to view and adjust their account settings (FIGURE 3.14) and also enables users to update their account information (FIGURE 3.15), such as their name, email, password, and other personal details.



FIGURE 3.14 – Manage profile Interface



FIGURE 3.15 – Update Informations Interface

### 3.4.6 File a Complaint Interface

This interface provides users with a way to file complaints or provide feedback. Users can submit issues or concerns related to their experience with the application(FIGURE 3.16).



FIGURE 3.16 – File Complaint Interface

### 3.4.7 Chat Interfaces :

The interface for user communication consists of two main parts :

#### 3.4.7.1 Chat Page :

This interface allows users to engage in conversations (FIGURE 3.17), Users can send and receive messages in real-time.



FIGURE 3.17 – Chat Interface

### 3.4.7.2 User List Page :

This interface displays the list of users with whom the user is having conversations. It shows details like usernames, profile pictures, and his status if he is online or not.

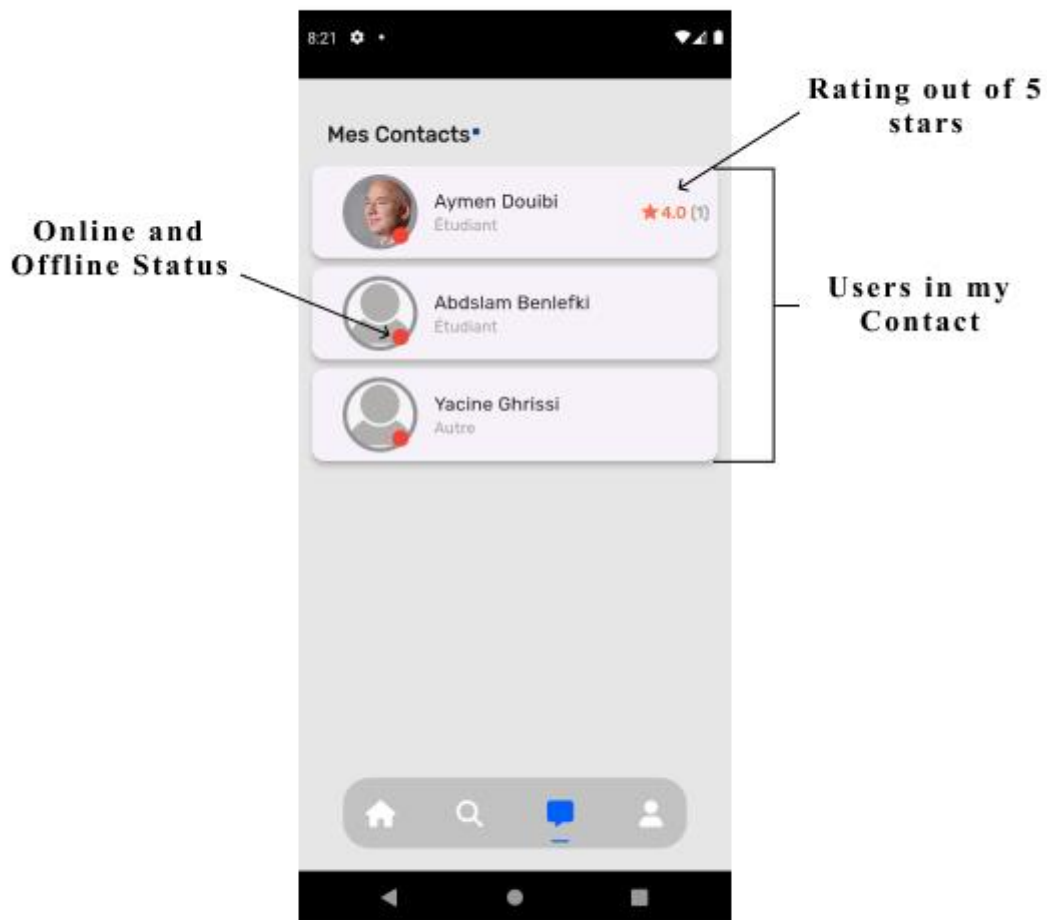


FIGURE 3.18 – specialties interface

### 3.4.8 Subject Interface:

The interface for user subject selection consists of three main section:

#### 3.4.8.1 List of Subjects Section

This interface displays a list of subjects (FIGURE 3.19) with their resources( FIGURE 3.20).

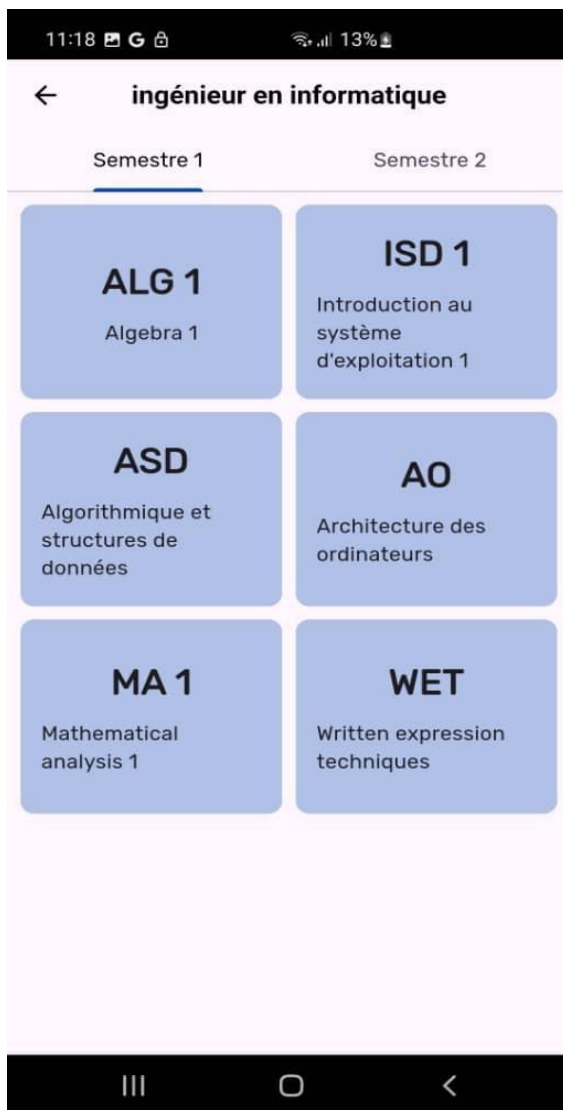


FIGURE 3.19 – Subjects Interface

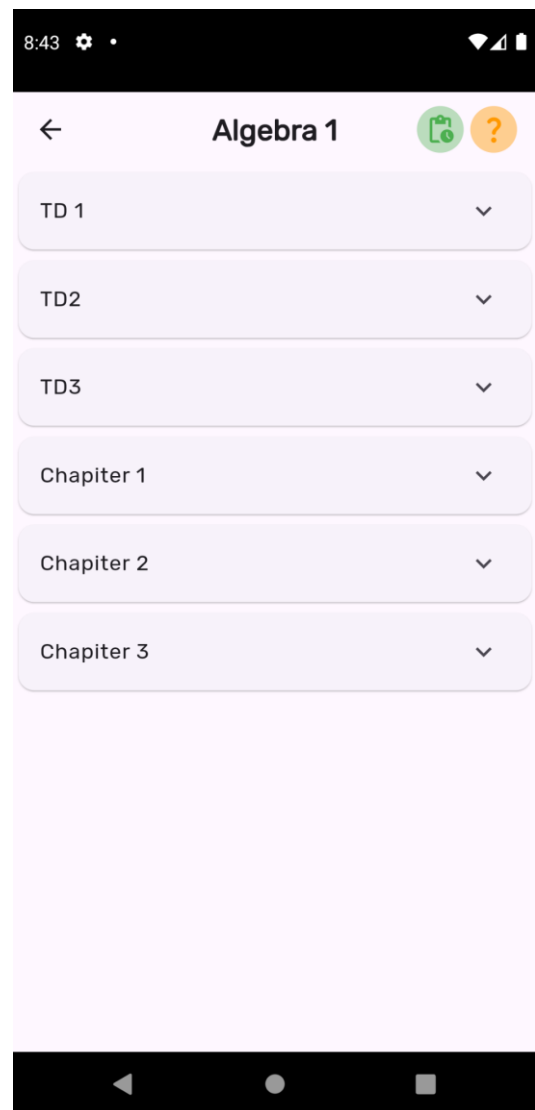


FIGURE 3.20 – Resource Interface

### 3.4.8.2 Subject Details

As shown in figure 3.21 this interface provides detailed information about a selected subject, including resources and a list of tutors that are available help, and list of learners who are seeking help.

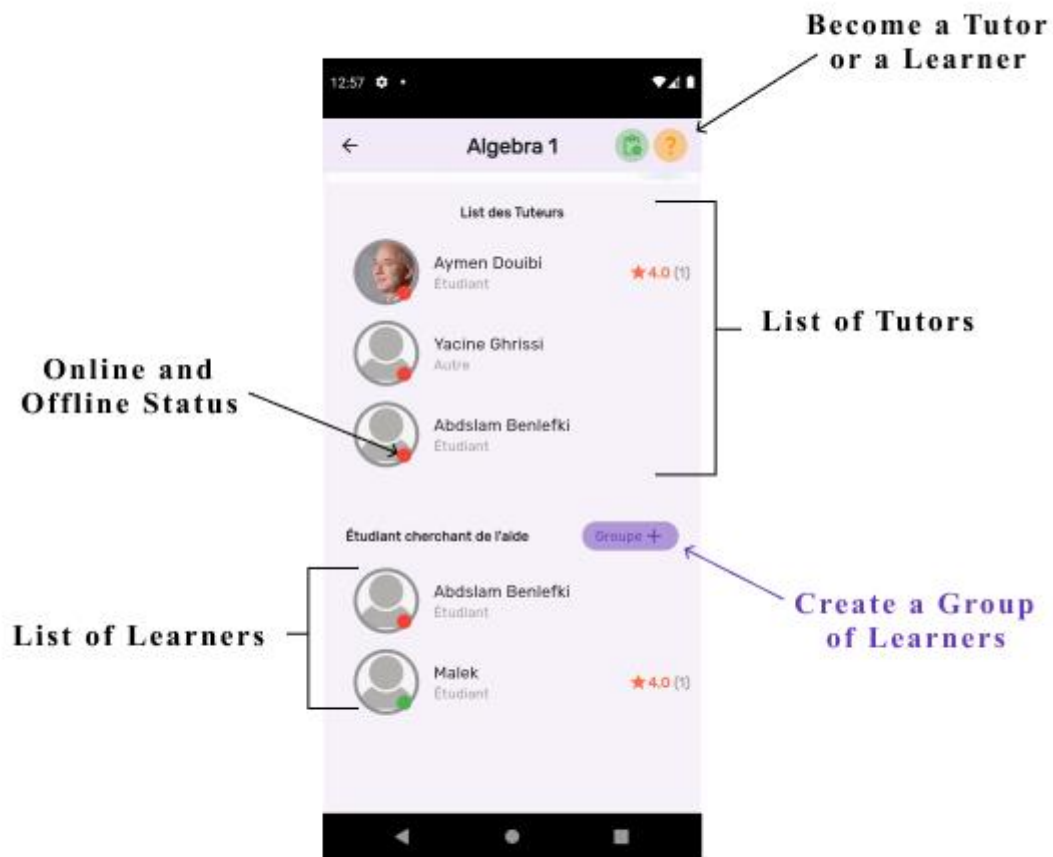


FIGURE 3.21 – Subject Details interface

### 3.4.9 Group Creation interfaces

in this interfaces a tutor can create a group chat for learners as shown in FIGURE 3.21 and FIGURE 3.22



FIGURE 3.22 –List of learners and Tutors



FIGURE 3.23 – Create Group interface



### 3.4.10 Become a Tutor

This interface (FIGURE 3.24) allows users to become a tutor for a specific subject. Users can provide their qualifications, availability, and other relevant details to offer their tutoring services.

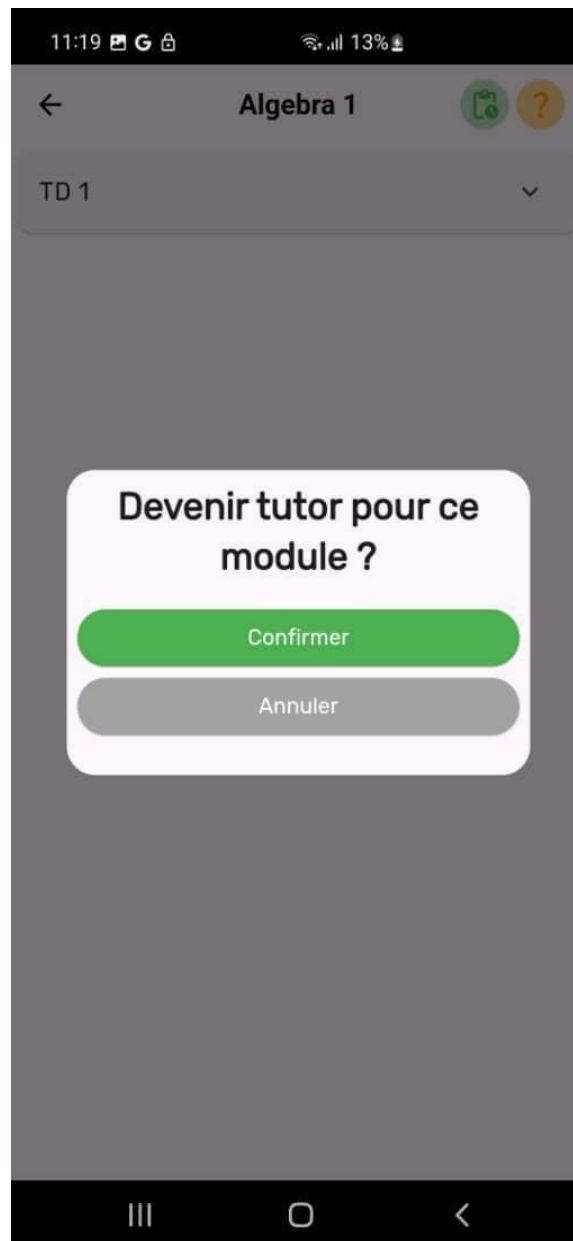


FIGURE 3.24 – Become tutor interface

### 3.4.11 Rating Interface

In this interface the student can give a rating of 1-5 stars for the tutor based on his performance (FIGURE 3.25).

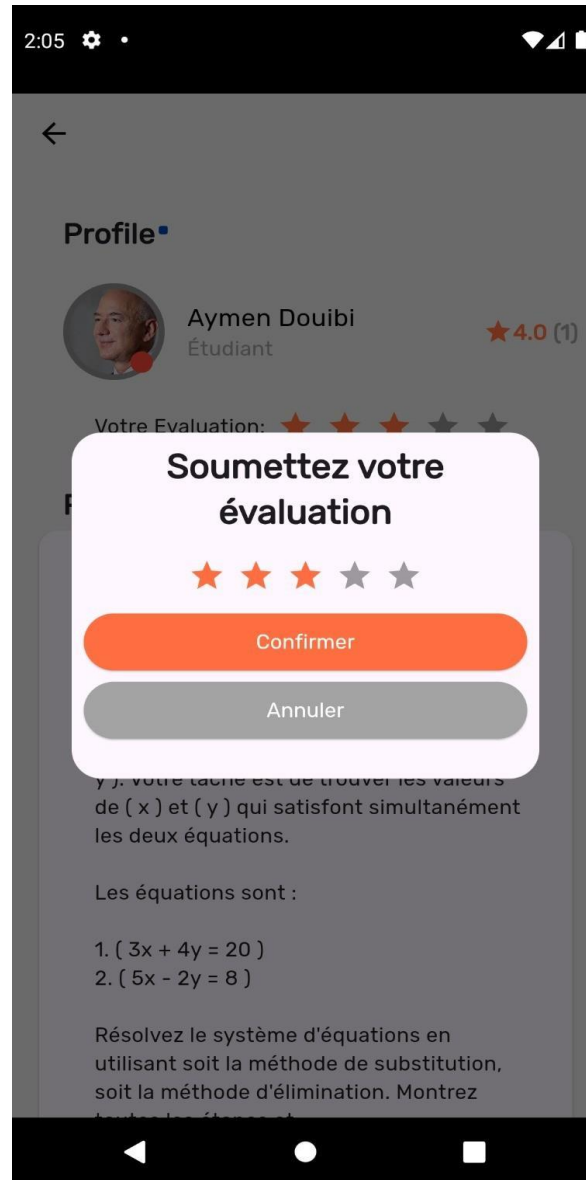


FIGURE 3.25 – rating interface

### 3.5 Fahemni Website

Since Flutter performs extremely well across different platforms, we decided to convert our app which initially was meant for mobile only, to a website and host it on the hostinger cpanel screenshots of the website are provided in FIGURE 3.26 and FIGURE 3.27.

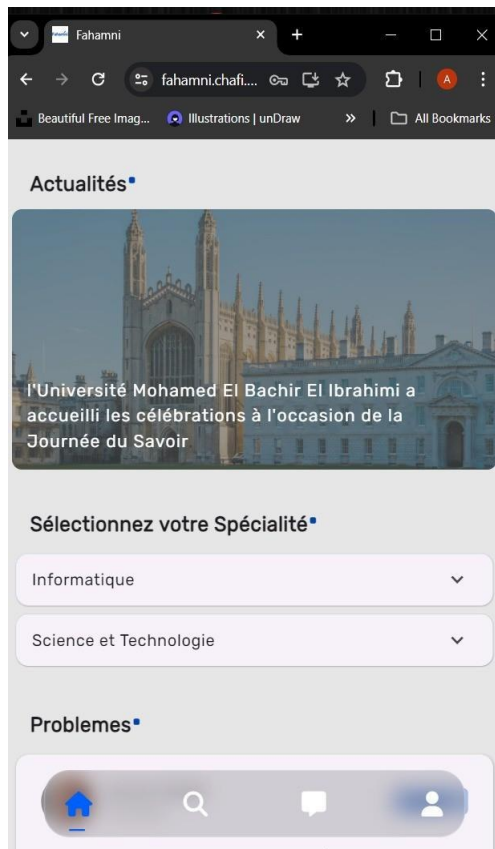


FIGURE 3.26 – Website Home Interface

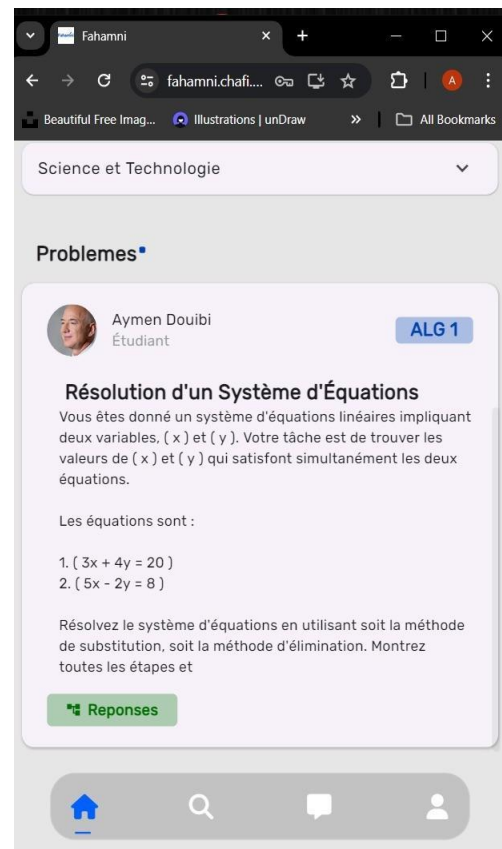


FIGURE 3.27 –Website Problems Interface

Link: <https://fahamni.chafi.net>

**Test Credentials:**

- Email: [user@fahemni.com](mailto:user@fahemni.com)

- password: 123456

### 3.5.1 Hostinger Web Hosting

Hostinger is a web hosting provider known for offering affordable and reliable hosting services, including shared hosting, VPS hosting, cloud hosting, and WordPress hosting. Founded in 2004, Hostinger provides a range of features such as a user-friendly control panel, free website builder, and 24/7 customer support, making it a popular choice for individuals and small to medium-sized businesses looking to establish an online presence.

### 3.5.2 Hostinger Dashboard

The Hostinger hPanel is a dedicated dashboard that is specifically designed to suit the use-case of web design freelancers and agencies that own or manage multiple websites for clients. It's an all-in-one dashboard.

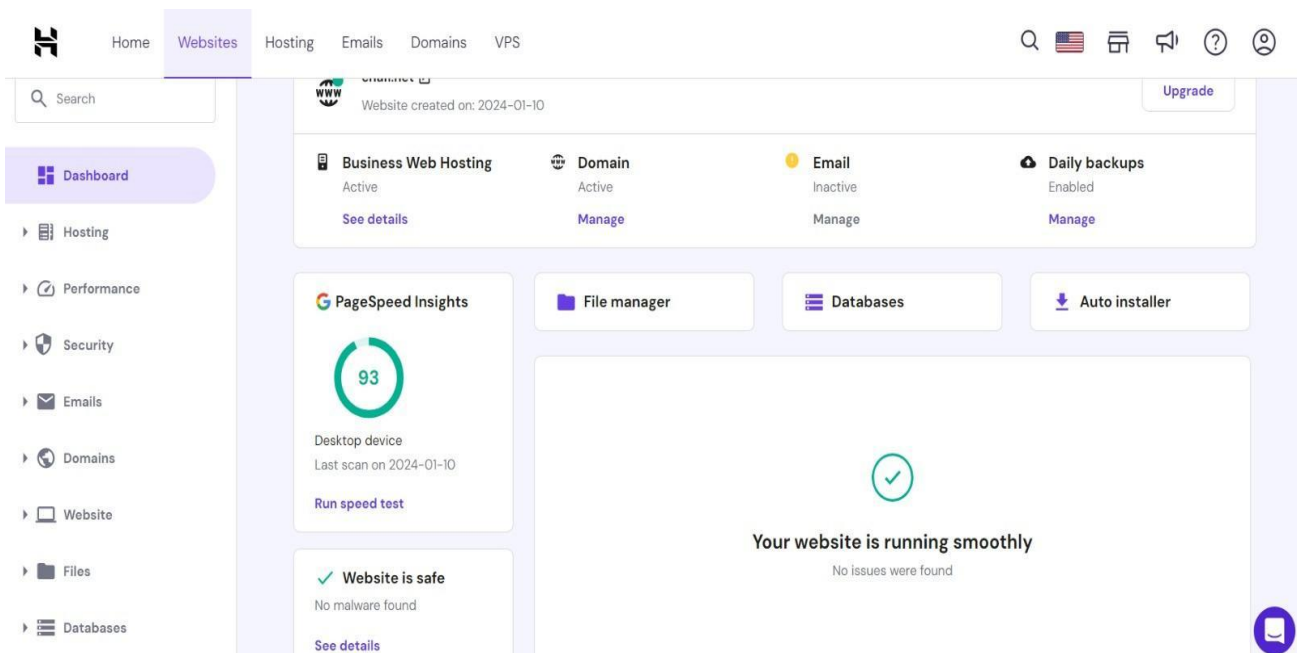


FIGURE 3.28 – Hostinger Dashboard

## 3.6 Conclusion

This Chapter gave us a closer look at the application's user interfaces and functionalities. We covered how users can manage conversations, select subjects, create groups, and sign up as tutors. We also explored the rating system for tutors and the seamless integration with Hostinger for reliable hosting and efficient website management. Together, these elements contribute to a well-rounded, user-friendly application designed to meet the needs of its users

## General Conclusion

This Thesis document outlines the development of a peer-learning platform. It begins with a presentation of the problem statement, followed by an introduction to the objectives and contributions of our model. We then provide an overview of the project plan.

In the second part of this Thesis, we review and explain various existing e-learning models, highlighting key points that informed the creation of our model. In the third part, we detail the design of our application using UML (Unified Modeling Language). We identify the actors who will interact with the platform and define their functionalities through use case and sequence diagrams. Finally, we extend these diagrams conceptually by constructing a class diagram.

In the final part, we present the implementation of our application. This includes a description of the technological choices made regarding the development environment and the system architecture, followed by an overview of the different interfaces of our peer-learning platform.

This Thesis has been both an interesting and enriching experience, allowing us to enhance our knowledge and skills in the development and design of complex systems.

## Future work and perspectives

In the future, several enhancements and expansions can be undertaken to improve and extend the capabilities of our peer-learning platform. One of the primary objectives for future development is to publish the app for iOS devices. Currently, the app is available on select platforms, and expanding its availability to iOS will make it accessible to a wider audience, ensuring that more learners and tutors can benefit from its features.

Another significant enhancement is the addition of a board sharing feature. This would allow tutors and learners to share a virtual whiteboard in real-time, facilitating better explanation of concepts, collaborative problem-solving, and more engaging tutorial sessions. This interactive tool can greatly enhance the learning experience by allowing for visual and collaborative learning techniques that are not easily replicated in text-based communication.

## Summary

During this modest work, we focused on developing a peer-learning platform. Our work was conducted in two parts: a theoretical part, where we studied various popular peer-to-peer platforms and their management, and a practical part, where we developed an application that facilitates and enhances collaborative learning among students.

**Keywords :** peer-learning platform, collaborative learning, mobile application

## Résumé

Durant ce modeste travail, nous nous sommes intéressés au développement d'une plateforme de peer-learning. Notre travail a été réalisé en deux parties : une partie théorique, dans laquelle nous avons étudié différentes plateformes de peer-to-peer populaires et leur gestion, et une partie pratique, dans laquelle nous avons développé une application qui facilite et améliore l'apprentissage collaboratif entre les étudiants.

**Mots-clés :** plateforme de peer-learning, apprentissage collaboratif, application mobile.

## ملخص

خلال هذا العمل المتواضع، ركزنا على تطوير منصة للتعلم النظير إلى النظير. تم إجراء عملنا في جزئين: الجزء النظري، حيث درسنا مختلف المنصات الشعبية للتعلم النظير إلى النظير وإدارتها، والجزء العملي، حيث قمنا بتطوير تطبيق يسهل ويعزز التعلم التعاوني بين الطلاب.

منصة للتعلم النظير إلى النظير، التعلم التعاوني، تطبيق محمول: الكلمات الرئيسية



## References

- Booch, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Pearson. Retrieved from <https://www.pearson.com/store/p/uml-distilled-a-brief-guide-to-the-standard-object-modeling-language/P100000723881>
- Boud, D., & Sampson, J. (n.d.). Peer Learning in Higher Education: Learning from & with Each Other. Routledge. Retrieved from <https://www.routledge.com/Peer-Learning-in-Higher-Education-Learning-from-and-with-Each-Other/Boud-Cohen-Sampson/p/book/9780749430095>
- Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Retrieved from <https://martinfowler.com/books/uml.html>
- Instructure Inc. (2011, February). Canvas Learning Management System. Canvas. Retrieved from <https://www.instructure.com/canvas>
- Stack Overflow. (2008, September 15). Retrieved from <https://stackoverflow.com/>
- Supabase. (n.d.). Supabase - Official Website. Retrieved from <https://supabase.io/>
- T, A. (2008). The Theory and Practice of Online Learning. Athabasca University Press. Retrieved from <http://www.aupress.ca/index.php/books/120146>
- Flutter. (2022). The Official Flutter Website. Retrieved from <https://flutter.dev/>
- Topping, K. J. (1996). The Effectiveness of Peer Tutoring in Further and Higher Education: A Typology and Review of Literature. Retrieved from <https://www.wgu.edu/blog/peer-learning2208.html>
- Visual Studio Code. (2015, April 29). Official Website. Retrieved from <https://code.visualstudio.com/>
- Abhishek. (2024, February 27). Cracking the Code: MVC Architecture Made Simple for Flutter Devs. Medium. Retrieved June 2, 2024, from <https://medium.com/@hpatilabhi10/cracking-the-code-mvc-architecture-made-simple-for-flutter-devs-1dcc2abf7b42>