VIETNAMESE - GERMAN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Programming Exercise 2

Course Project Report

Student 1: Vu Dinh Nam - 10422052

Student 2: Pham Bao Long - 10422044

Student 3: Pham Duc Manh - 10422047

Student 4: Tran Ngoc Anh Toan - 10422118

Student 5: Nguyen Thanh Bao Hung - 10422031

HỒ CHÍ MINH CITY, 26/06/2025

Content

1 Introduction

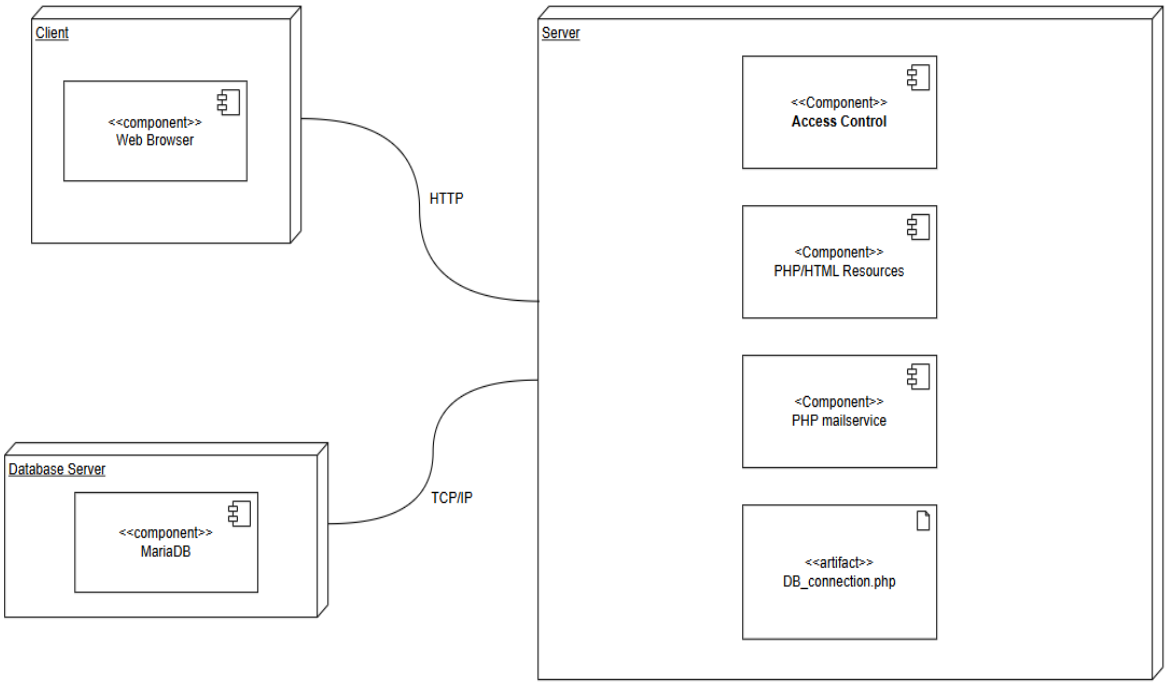
As digital solutions are being made to satisfy the needs of everyday life, we have found a gap in the market for our university students: academic assistance. Unlike education at lower levels, where materials are freely and widely available and tutoring can be done by a much larger group of academic providers, courses in university are far more specific, unique and exclusive to those who have enrolled. Therefore, if a student needs extra help with their classes, they have no choice but to consult their own professors or fellow students with whom they've already made acquaintance, who can be more difficult to reach and unable to cope with any moderate to large demand. This solution is also inaccessible to those who have a hard time socializing.

With VGtUtor, we aim to bring assistance to anyone that needs it, whilst providing an opportunity to make extra income to those who excel academically. Following the business models of some of the world's most prominent digital solutions pioneers like Uber and Amazon, we are creating VGtUtor as an intermediary platform connecting students within VGU: those who have demand for tutoring and fellow students who can provide such a service.

This project incorporates elements of web design and programming, database management and administrative work. Such a service is perhaps best suited for the mobile platform, although this will not be covered in the scope of this project. This report will follow the development process, from the analysis and design phase to programming and testing, as well as deployment. Simple steps can be followed to start hosting VGtUtor on your own server.

2 System Design

2.1 Deployment Diagram



Hình 2.1: Deployment Diagram

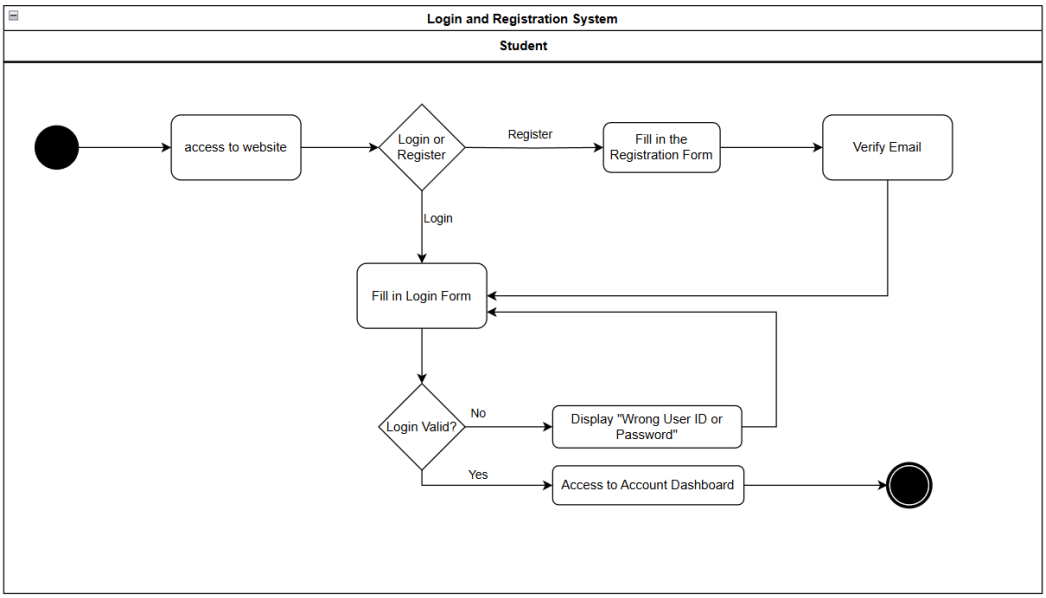
The deployment diagram for our Web Application is relatively simple, as all clients can access the system through their Web Browser. The server with all of its services will serve clients with PHP and HTML resources through standard HTTP/HTTPS protocols. Each client will need to satisfy the access control conditions in order to retrieve a particular resource. The server will also employ a mail service, sending automated emails for verifications and other features like password resets. This mail service will also provide users with a way to contact our team.

We have also made the database independent from the Web Server itself, where the connection logic can be configured through our connection artifact. This way, the database itself can be stored elsewhere, have backups, and can be configured separately from the rest of the system.

2.2 Activity Diagram

This section will discuss many of the processes that occur when users interact with the system in specific ways. Each can be represented with its own activity diagram, describing clearly every action that is taken based on certain conditions.

2.2.1 Login and Registration System

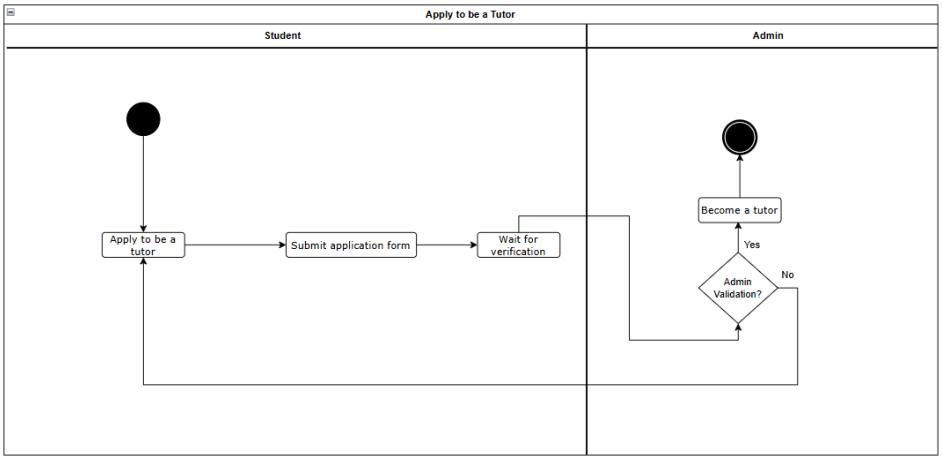


Hình 2.2.1: Login and Registration System

We first tackle the simple process of signing into the website. For new users, they will need to sign up for a student account. They start by filling out a personal information form, at which point the mail service will send them an account verification email. Clicking on the provided link will then activate their account. They can then sign in.

With their user ID and password, the user can log into the system. Wrong credentials will send them back to the log in screen with an error message. A successful log in will take them to your account. Administrators go through the same process, except one cannot sign up to be an admin, but to be assigned admin credentials by system managers.

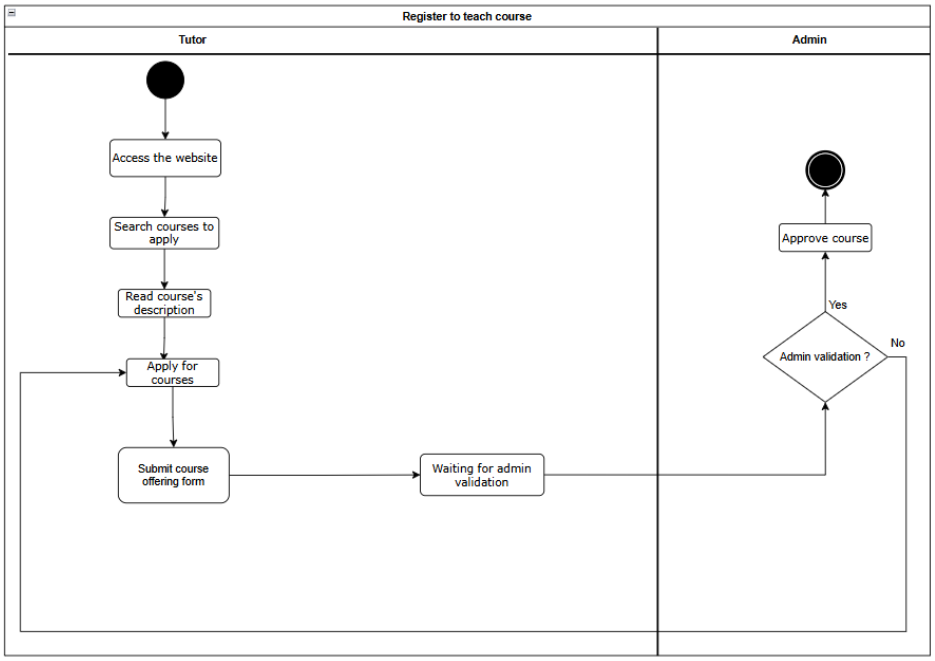
2.2.2 Apply to be a tutor



Hình 2.2.2: Apply to be a tutor

When students are signing up to become tutors, they will fill a registration form with any necessary information. They will then have to wait until an admin approves their request. They can then become a tutor and inherit tutor privileges. If rejected, nothing further will happen.

2.2.3 Register to teach course

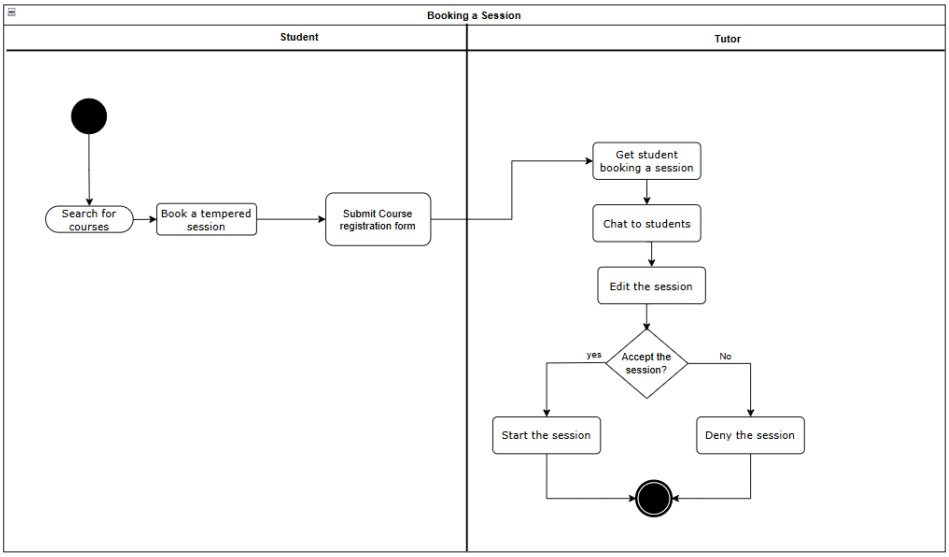


Hình 2.2.3: Register to teach course

A tutor can then, of course, register to teach a specific course. They can look through the website to choose the course they want to offer. Reading the course's description will make sure they are aware of the requirements for application. They can then fill and submit an application form,

and similarly to registering to become a tutor, they will then wait for an admin response. Once accepted, their offering will be valid, whilst nothing further will happen upon a rejection.

2.2.4 Book the Session between tutor and student

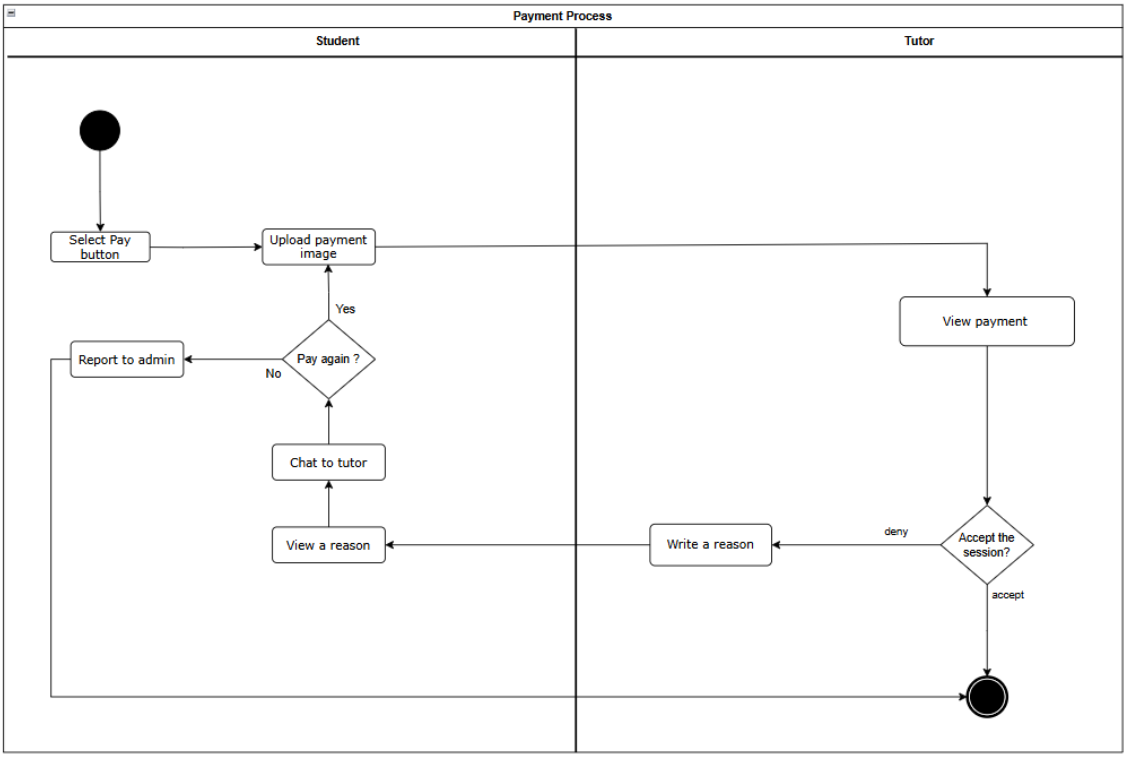


Hình 2.2.4: Booking Sessions between tutor and student

As a student looking for tutoring, you'd first look for the specific course that you'd like tutoring in, from which you can pick between all the offerings for said course. Once satisfied with your choice, you'd go ahead by filling and submitting a registration form. This would consist of information such as place and time for the session.

The corresponding tutor would be notified of the registration, upon which they would start a chat with the student. Modifications on the session can be made, then they can decide whether or not to accept said session. The process ends with this decision. The tutor can then go ahead and provide their services as per agreed terms. They are not forced to talk to the students, but decisions and interactions will be made through the chat box regardless.

2.2.5 Payment Process

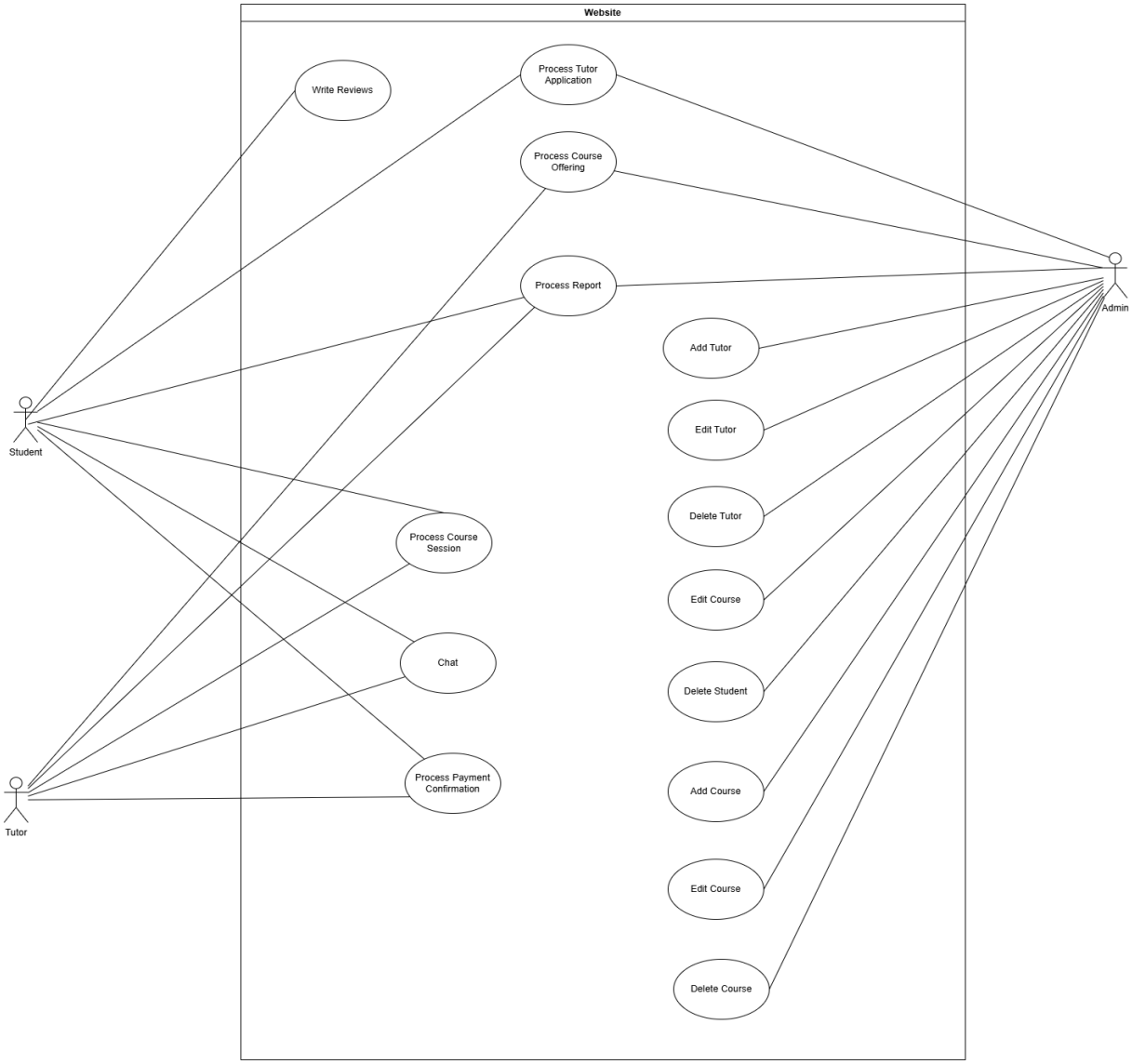


Hình 2.2.4: Payment Process

After successfully booking a session, the only thing left to do is to pay. As the tutor's bank information is already saved on the system database, the student can simply pay and upload their payment confirmation image for the tutor to see. The tutor can then view this proof of payment and either confirm or deny. Accepting would end the transaction and the payment process is complete.

If some issue occurred, like the tutor not receiving their payment, they can deny the confirmation and write back a reason for the student to resolve. The student can then attempt to pay and provide proof again, or contact administrators for further help. Financial clarity in these transactions is a sensitive matter, but in most cases that we are concerned with, this process would be run thoroughly and smoothly.

2.3 Use Case Diagram



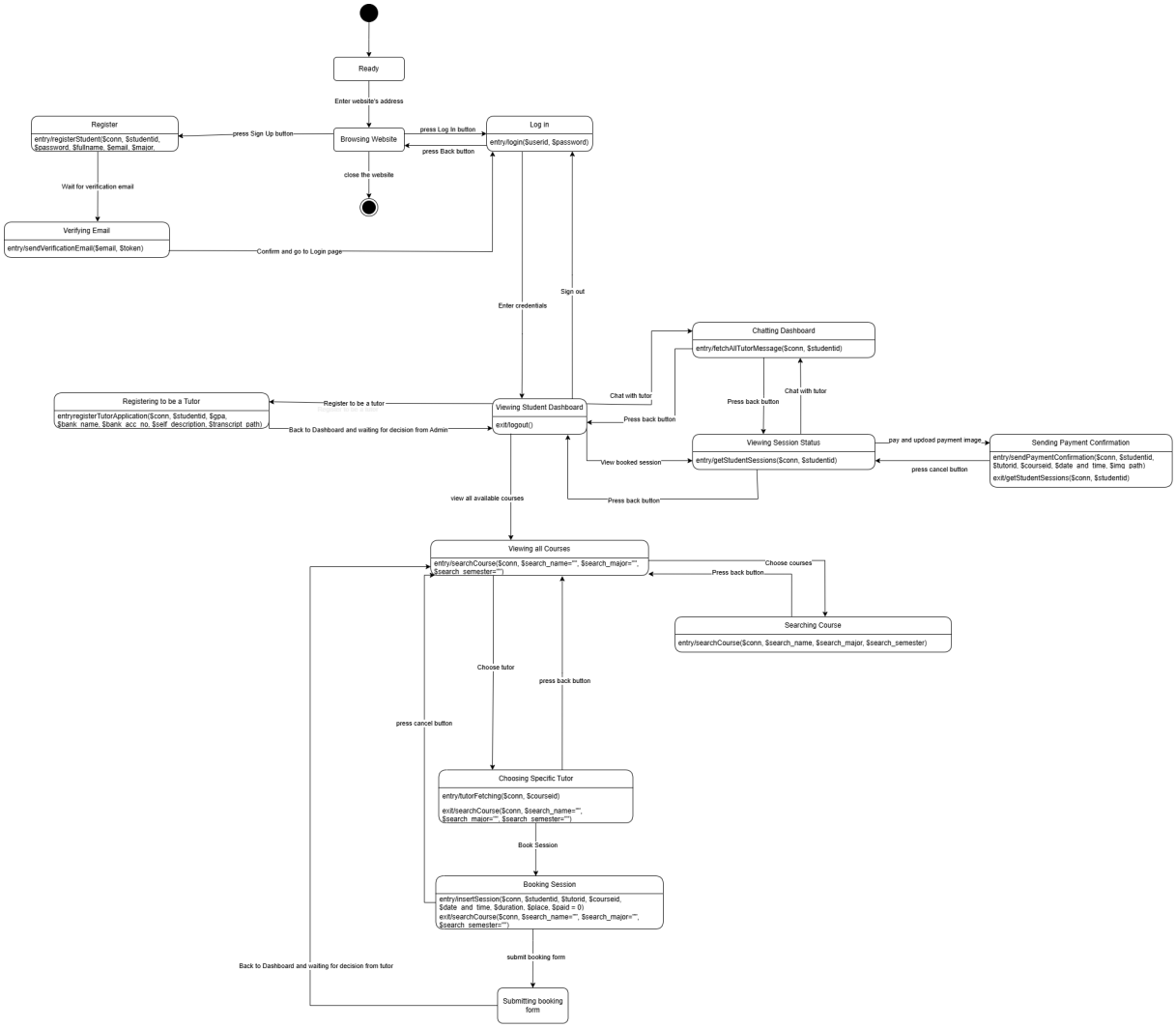
Hình 2.4: Website Use Case Diagram

As we can see from the above use case diagram, the student has the least access to the website resources. Their role is a participatory one, where they can interact with the website only in terms of purchasing the service. The tutor has a few more access points, where they can interact with their own offerings and any reports concerning their service.

This is in stark contrast with the administrator, who can manipulate and maintain the database in any way necessary for the management of the platform. They are also in charge of processing registrations as well as editing and deleting entries from the database. Their job is a necessary and important one for the correct functioning of the system and thus should only be relegated to either the university staff or a select few student workers.

2.4 State Chart Diagram

2.4.1 State Chart Diagram in Student Side

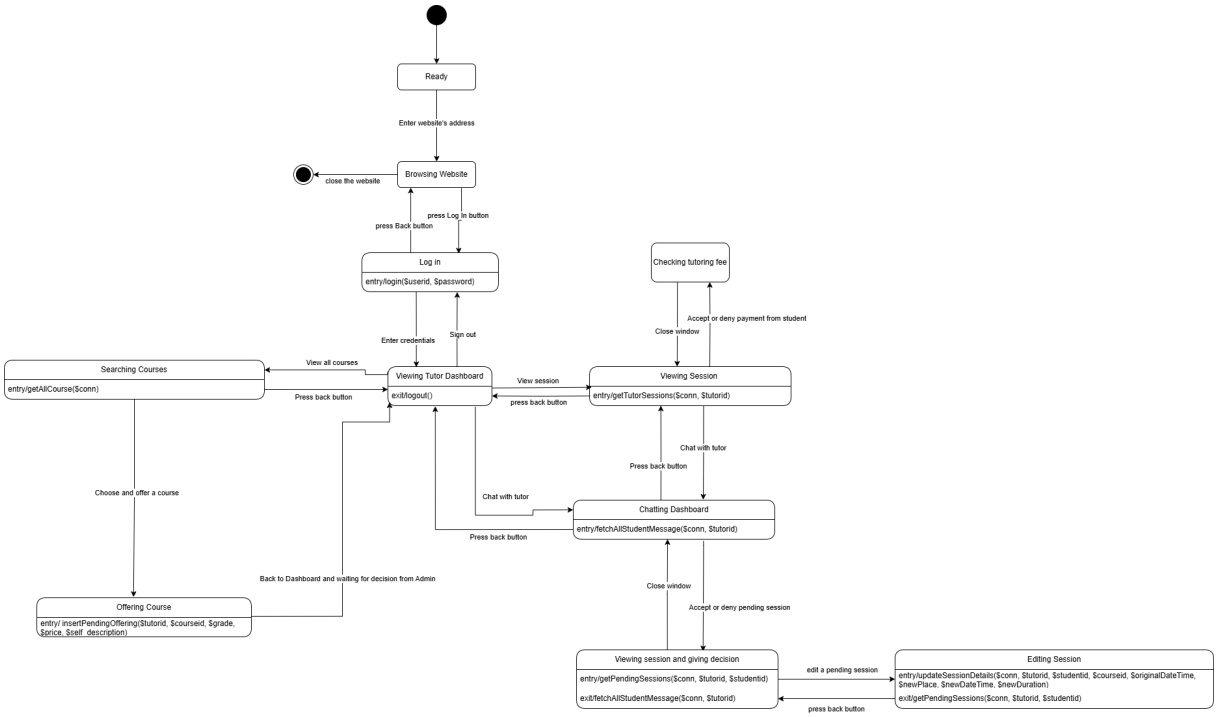


Hình 2.4.1: State Chart Diagram in Student Side

- The state diagram captures the essential lifecycle of a student's interaction with the VGtUtor platform. It starts with the user entering the website and choosing to either register or log in. Registration involves submitting personal information and awaiting email verification before gaining access. Upon successful login, the student is directed to their dashboard, which serves as the central hub for navigating the system's features.

- From the dashboard, students can search for available courses, view tutor profiles, and book sessions by filling out a booking form that includes specific details such as time, location, and price. They can also check the status of their booked sessions and communicate with tutors via a dedicated chat interface. Additionally, students who wish to become tutors themselves can submit an application for approval. Overall, the diagram outlines a streamlined and role-sensitive interaction model that guides students through learning, communication, and tutoring opportunities efficiently.

2.4.2 State Chart Diagram in Tutor Side

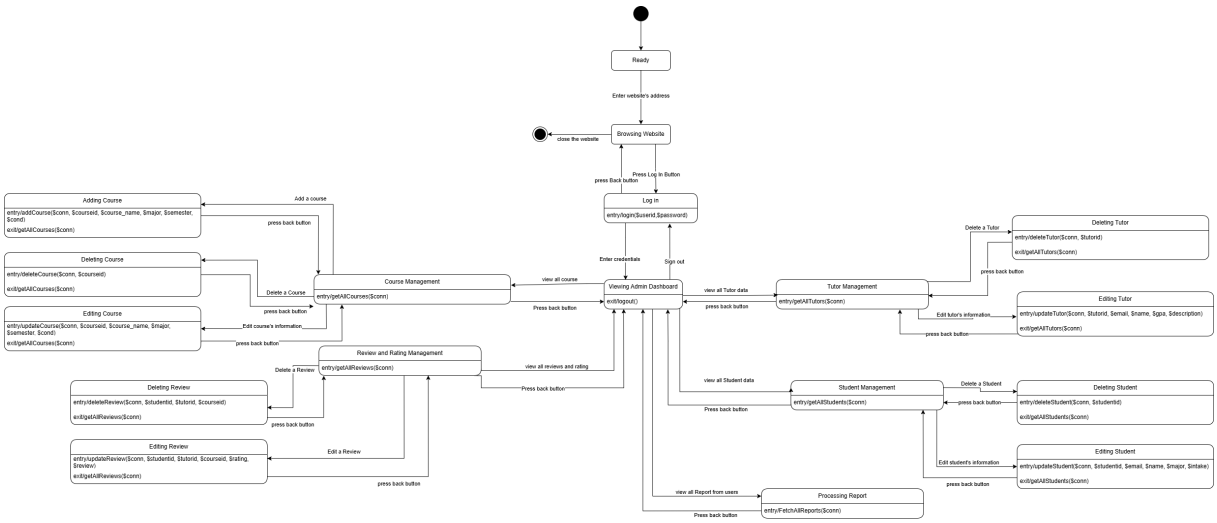


Hình 2.4.2: State Chart Diagram in Tutor Side

- The state diagram for the tutor illustrates the key workflow that guides a tutor through the VGtUtor platform. Beginning with the login process, tutors access their dashboard where they can manage their core responsibilities, such as offering courses, communicating with students, and handling session requests. The interface enables tutors to browse available courses and submit offerings, which are then subject to administrative review. This structured flow ensures that course creation and offerings are kept organized and monitored, maintaining the integrity of the platform.

- Furthermore, tutors can monitor session bookings made by students and make decisions on whether to accept or reject these requests. They are provided with comprehensive details of each session and have the ability to edit session attributes such as time, location, and duration. The integrated messaging system allows for direct communication with students, enhancing coordination and responsiveness. Additionally, tutors are tasked with verifying payments and session statuses, further highlighting their role in maintaining session logistics. This diagram succinctly captures the systematic approach that tutors follow, emphasizing control, communication, and collaboration within the application.

2.4.3 State Chart Diagram in Admin Side

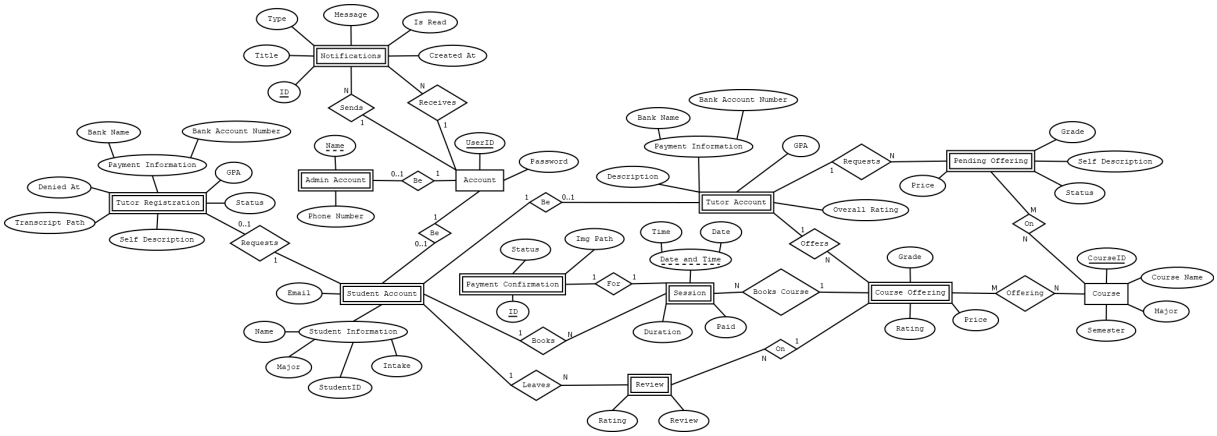


Hình 2.5.1: State Chart Diagram in Admin Side

- The state diagram for the admin role outlines a comprehensive control flow within the VGtUtor platform, reflecting the administrator's central responsibility for managing users, courses, and system integrity. The admin begins by logging in and is directed to the Admin Dashboard, from where all management activities are initiated. Key responsibilities include managing courses (adding, editing, or deleting them), reviewing and moderating user-submitted reviews, and accessing reports from users. These functionalities ensure academic content remains up to date and community feedback is monitored to maintain service quality.

- Additionally, the admin oversees user management for both tutors and students. Within the dashboard, the admin can retrieve, update, or delete tutor and student records, ensuring only verified individuals participate on the platform. This oversight extends to handling any user-generated reports, enabling the administrator to resolve disputes or system misuse efficiently. Through this diagram, the critical role of the administrator is made evident—they function as both gatekeeper and overseer of platform operations, ensuring all data and interactions adhere to the platform's standards and expectations.

2.5 Database Schema



Hình 2.4: Database ER Schema

Our system is dependent on 2 core entities: The Account and Course. Based on these 2 tables, all entries from all other tables will reference. For a start, all types of accounts will inherit from the account entity. This includes the admin account, the tutor account, and the student account. With just your log-in credentials, the database will check for your account's entry in either the admin table, the tutor table, or the student table. This will reflect your role in the system. Ideally, an enhanced entity relationship diagram (EER) could better illustrate this, however, for readability, we have elected to use the standard ER schema.

2.5.1 Accounts and Account Types

In detail, we should preface with the fact that the 2 main types of accounts are admin accounts and student accounts. Since all tutors have entries in the student account as well, this reflects our intention of making sure the service provider (i.e. tutor) also can purchase the service as a customer (i.e. student). This is relatively similar to many popular e-commerce apps like eBay and Shopee. As tutors, extra pieces of information are required from you to provide to advertise your service (tutoring). Student's personal information are also stored, but only those that are necessary for the verification of a student's identity within the university.

Student and therefore tutor accounts will be uniquely identified by their user IDs matching their student ID (matriculation number). All passwords are cryptographically hashed when first set, then the hash value will be stored in the database. This ensures that even if the database is compromised, we can protect users and their information from being accessed by malicious actors. Given that this is designed to be an internal system managed by a university, this risk might not be as large, but this simple implementation is still an effective method to provide a good level of security.

Some financial information about tutors are also stored in the database, but since these pieces of information are public and are only used to transfer money into their accounts, they cannot be used to compromise the tutors themselves. This is a good and secure way to provide students and tutors with a method of service payment. For practical implementation in the future, an automatic payment method will be put in place for students to pay for their tutoring through the system, from which the system will pay the tutors, instead of the customer and provider exchanging currency directly. This would be a much more secure and safe way to operate this application, with potential for the service to produce revenue as well.

2.5.2 Courses and Course Offerings

Another crucial element of our design is the Course table and the Course Offerings. The offerings themselves are indeed the core service that is being provided, and they will be based on the pre-existing courses defined by the system. Each entry in the Course table will correspond to a real module on offer at our university. This means administrators have complete control of what subjects are being covered and thus their contents. Aside from basic information like what major and semester the course belongs to, tutors also need to meet each course's prerequisite if they wish to teach. This will most likely be just a simple minimum grade condition.

Each tutor can have their own offering of any given course, given that they qualify to do so. A course can be offered by any given number of tutors, and the idea is for them to compete with each other for students, improving their chances with good reviews, a competitive price or higher expertise. If a tutor is capable, they can offer as many courses as they want. Each of

these offerings are unique and can be seen, looked up, and booked by students.

2.5.3 Tutor and Offering Registrations

Every student can start tutoring. Uploading the necessary extra information that all tutors need will send a request form to our administrators. They will then decide whether or not to approve such requests. Personal interviews or any further vetting process can be carried out as well, since this is an internally managed system for the university.

Each tutor can then start offering courses by filling a form, sending their grades and self descriptions for the administrators to vet through. Once approved, their offering will be available to be booked. This is the safest and most effective way to manage this process for a system of this scale and scope. If demand were to rise and a call for automation is necessary, such an algorithm can then be considered and implemented.

2.5.4 Sessions, Reviews and Notifications

Once a student has booked a course offering, a session can be created, with specified times and locations. A student can then pay for the session at any time. The database will keep track of this, as payments need to be confirmed for each session by the tutor. It should be noted that these course offerings aren't meant to be a repeated and scheduled events, like normal classes, but rather independent bookings, which we felt better fit the needs of our target demographic.

After a session, a student can write reviews on the course offering. This will include a rating out of 5 and some comment. Administrators will of course need to monitor and moderate these comments. Future implementations might be able to automate this process. These reviews are public and can help students to choose the tutor they need. These act as a motivator for our service providers to improve their work.

The notification system is used to let users know whenever changes have happened concerning them. This can be anything from their accounts being successfully verified to a student being notified that their application to be a tutor has been accepted. Bookings and requests are also very important for user experience, and those can be seen as notifications sent between one user to another, which explains the way we designed our database schema above.

3 User Interface

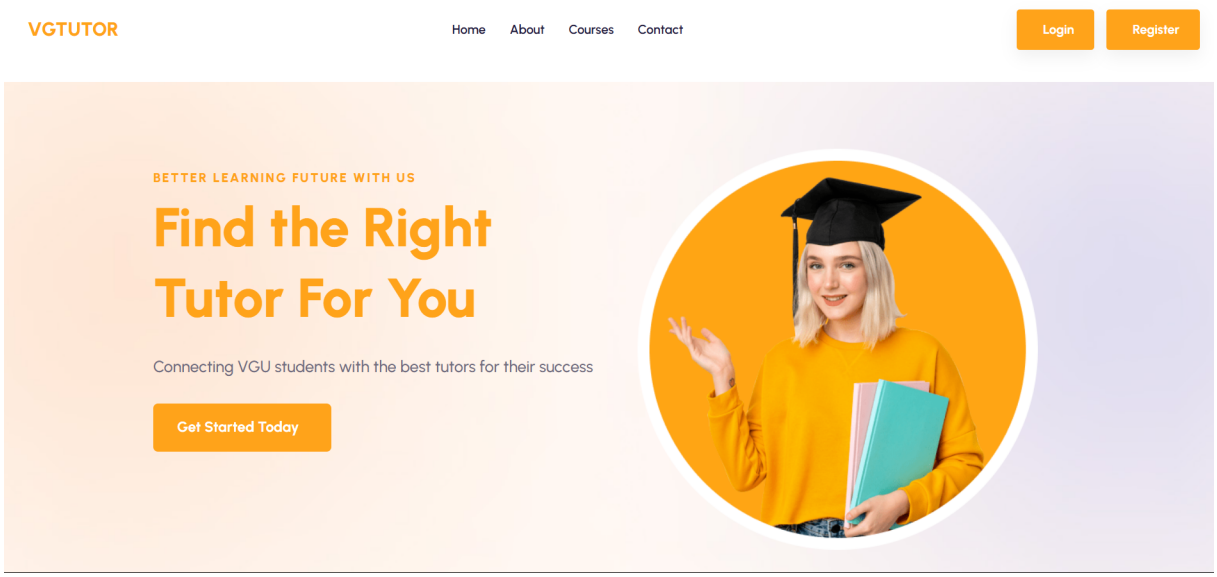
3.1 Introduction

The VGtUtor user interface (UI) is designed to provide an intuitive and streamlined experience for students seeking academic support and tutors offering their services. The interface is structured to focus on the student's needs, allowing them to easily access courses, interact with tutors, and manage their profiles, all in a user-friendly environment.
There are multiple dashboards tailored to different user roles within the system, each serving a unique purpose. In this document, I will only demonstrate a selection of these dashboards to highlight key functionalities and user flows.

3.2 Key Components

3.2.1 Home Page

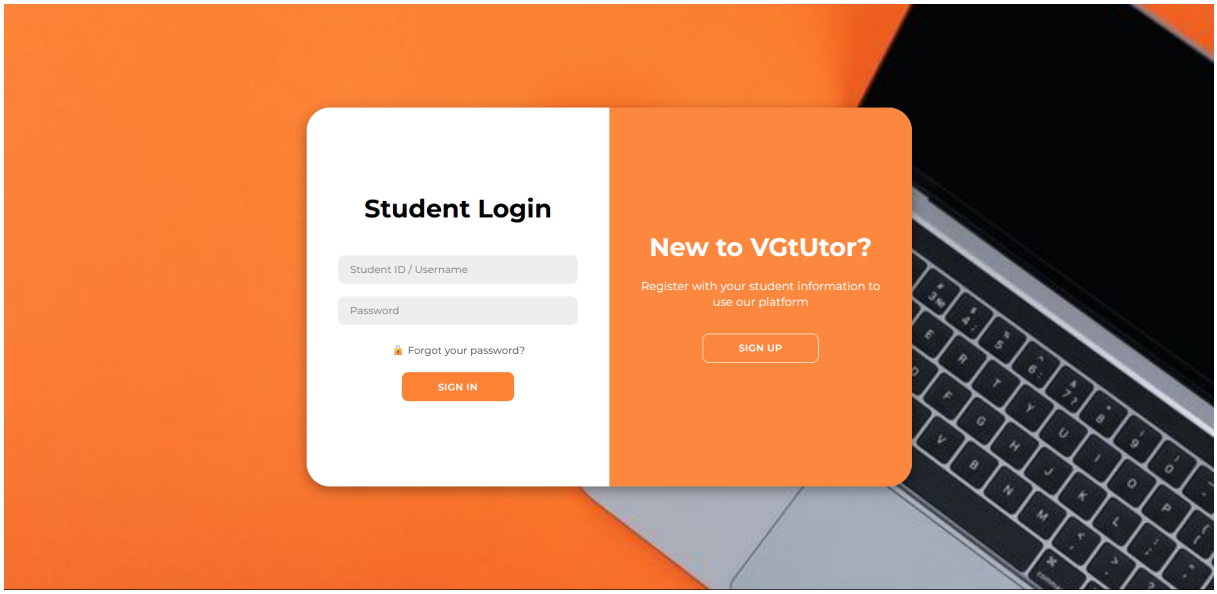
- We start by designing the home page for VGtUtor. With the goal of keeping design elements simple and easily navigable, this page will mainly serve to direct users to logging in and using our services. We follow traditional home page designs, with clearly visible options put into intuitive arrangements.
- A small "About Us" section provides introductory information for new users and a "Contact Us" form allows anyone to submit a message directly to our administrators, whether for enquiries, requests or even complaints.



Hình 3.2.1: Home Page

3.2.2 Log In

- Clicking the "Login" option from the home page will take you to our log in portal. Here, only your credentials are necessary for logging in. As each account has a unique user ID, the system can log you in automatically based on whether you are one of our admins, a tutor, or just a student. We have elected to not implement a register functionality as of the current version of the application, as we feel this shall be one of the university's internal systems, and accounts shall be created upon request, for security reasons.

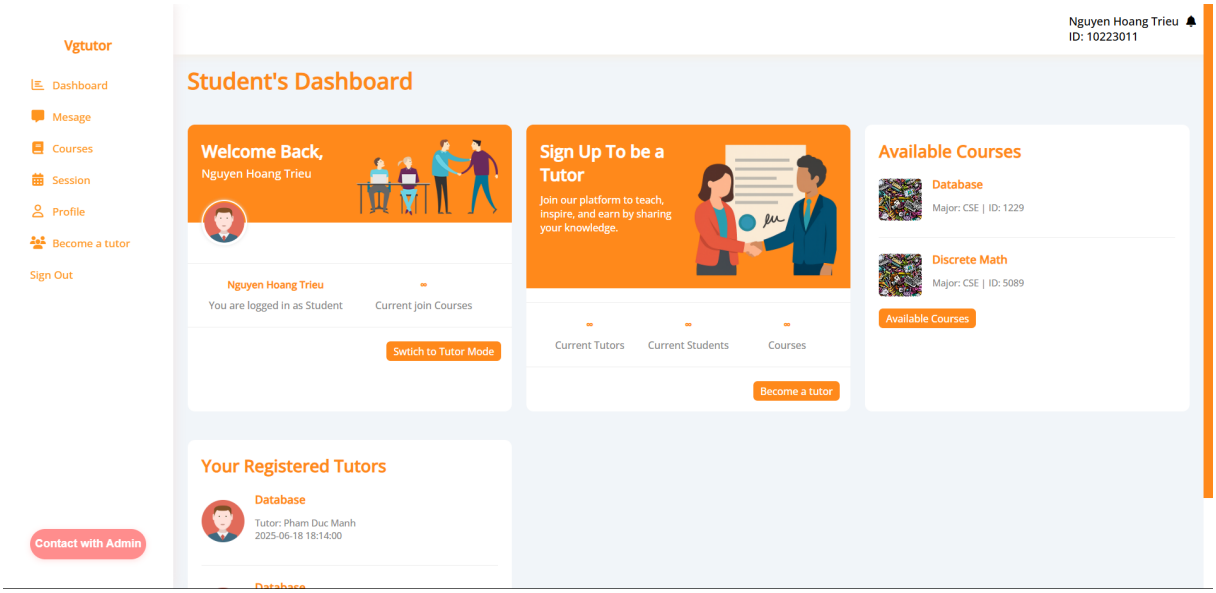


Hình 3.2.2: Log In

3.2.3 Student's Dashboard

- After successfully logging in, users are directed to the main dashboard. This central hub presents a clean and organized interface where students can quickly access key features and information relevant to their learning journey.
- At the top, a welcome message greets the student by name, helping personalize the experience. Basic user details such as their role and current courses are shown in this section. A button also allows students to switch to tutor mode, in case they are registered as both.
- On the right, a "Sign Up To be a Tutor" panel encourages students who are interested in teaching to join the tutor community. This section provides a brief explanation of the benefits and a shortcut to begin the tutor registration process.
- The "Available Courses" section displays courses the student can join. Each course includes the subject name, major, and course ID, with a button for accessing the full list of offerings.
- Below, the dashboard includes a summary of "Your Registered Tutors," allowing students to easily track which tutor they are working with and view session details such as time and tutor name.

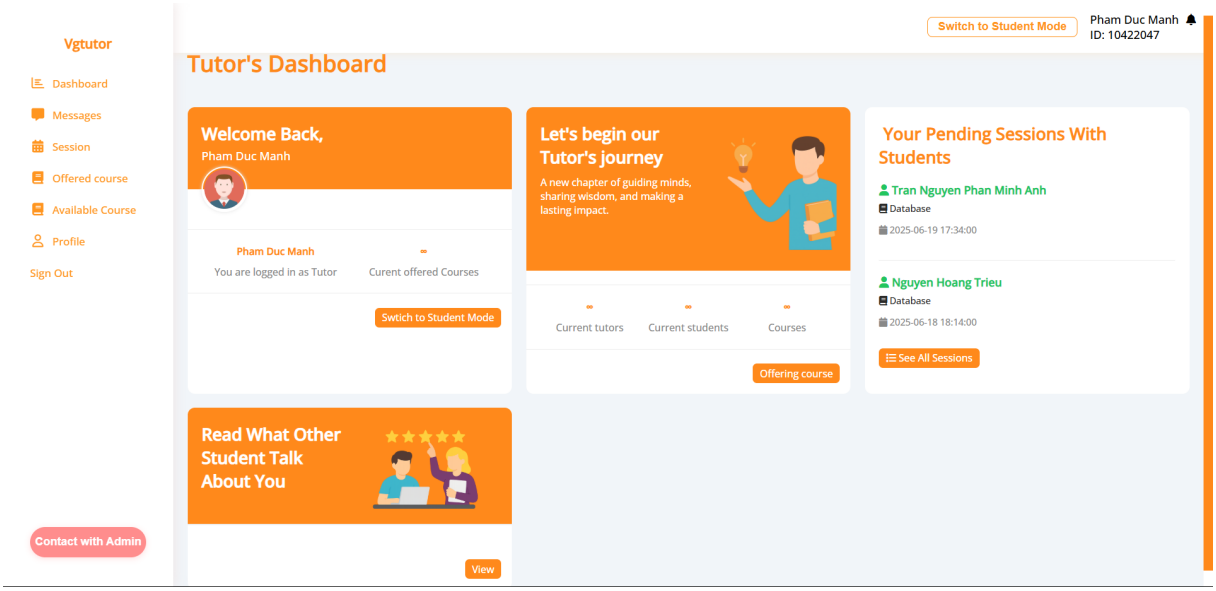
- The sidebar on the left provides quick access to other sections like Messages, Courses, Sessions, Profile, and the option to become a tutor or log out. There's also a dedicated button to directly contact the admin for help or support.



Hình 3.2.3: Student's Dashboard

3.2.4 Tutor's Dashboard

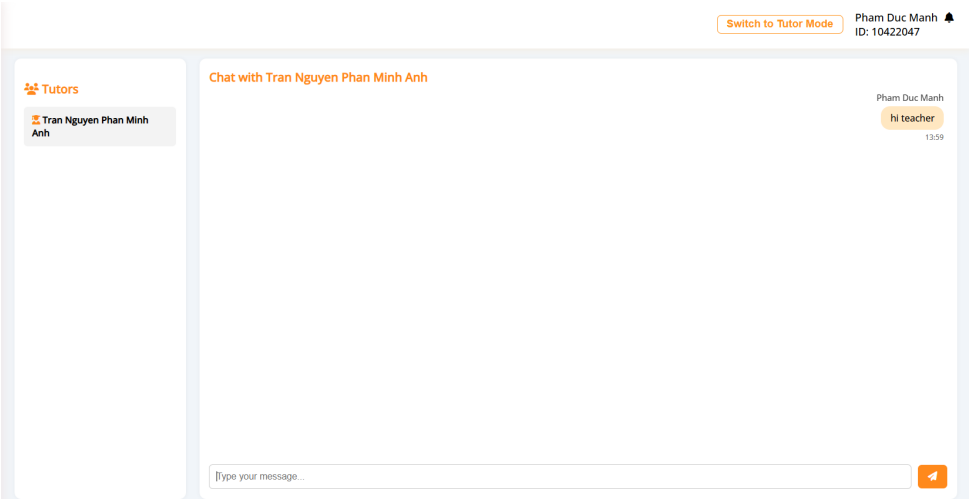
- The tutor dashboard is designed to support tutors in managing their teaching activities and connecting effectively with students.
- The "Let's begin our Tutor's journey" section serves as both an inspirational prompt and a shortcut to start offering new courses through the "Offering course" button.
- A panel titled "Your Pending Sessions With Students" summarizes upcoming tutoring sessions. It shows the student's name, course name, and scheduled time—making it easy for tutors to track their appointments at a glance. A "See All Sessions" button is also provided to explore the full session history.
- At the bottom, tutors can access feedback from their students through a dedicated section labeled "Read What Other Student Talk About You."



Hình 3.2.4: Tutor's Dashboard

3.2.5 Message Page

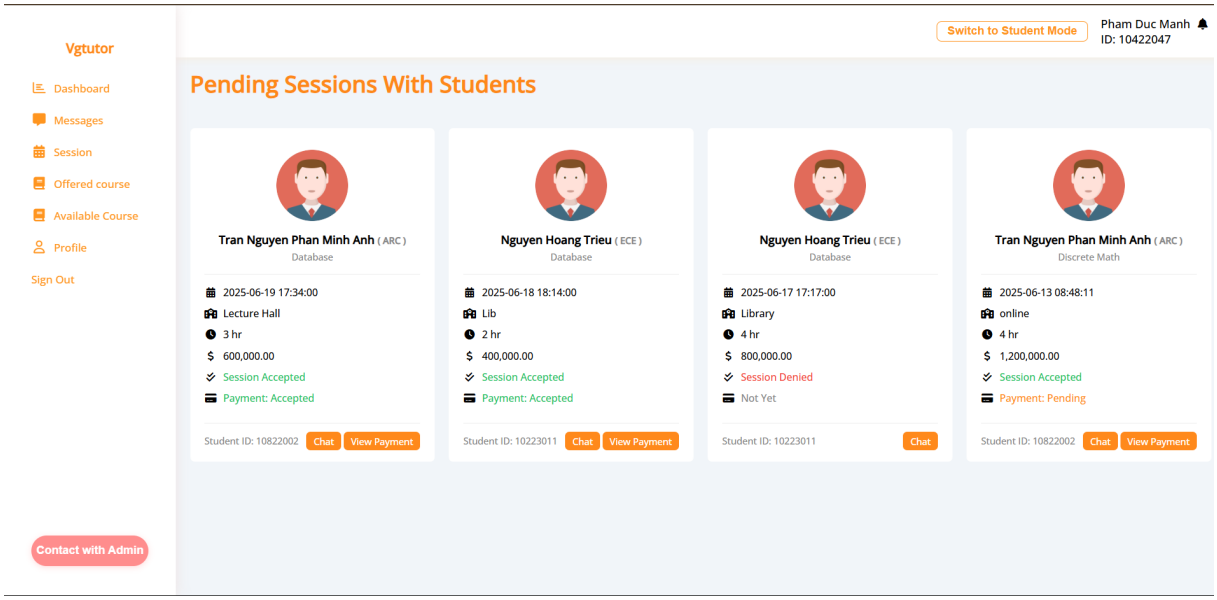
- Both students and tutors have access to this chat interface, which is designed to support seamless communication between users before and during their learning sessions.
- On the left-hand side, users can view a list of people they are currently connected with. In this example, a student is chatting with their tutor, Tran Nguyen Phan Minh Anh. This layout allows for quick switching between multiple conversations if needed.
- The main chat area shows the conversation in real time. Messages are displayed in a clean, bubble-style format, with timestamps and clear sender identification to avoid confusion.



Hình 3.2.5: Chat Panel

3.2.6 Pending Sessions(Tutor View)

- This page allows tutors to manage all upcoming and past session requests from students. Each session card provides a compact summary of all the essential details, helping tutors stay on top of their teaching schedule without missing any critical information.
- For every session, the system displays the student's name, course title, session date and time, location (whether physical or online), duration, and payment amount. This gives tutors a clear overview at a glance.
- At the bottom of each session card, the tutor can find the student's ID and two quick-access buttons:
  - **Chat**: to open the messaging interface for that student.
  - **View Payment**: to inspect payment details or confirm if the transaction has been completed.



Hình 3.2.6: Tutor's Pending Sessions with Students

The features and dashboards presented in this section highlight the core functionalities of VGtUtor's user interface. From intuitive student and tutor dashboards to real-time messaging and session management, the platform is designed to enhance user experience and streamline academic support interactions. While only a selection of screens has been demonstrated here, they represent the most essential components of the system.

4 System Implementation and Development

We have implemented and developed our Web App with PHP and SQL, as these are very popular and highly supported tools for this particular type of project. This section will concern different elements of VGtUtor, and some logical functionality shall be explained as necessary.

4.1 Access Control

Our access control system determines whether or not a user can reach a resource based on their sign in session role. Standard PHP access control practice specifies that at the beginning of the file. For example, when a file is only accessible for a tutor account, the beginning of the file will look as follows:

```
1 <?php
2 session_start();
3 if (isset($_SESSION['tutorid']) && isset($_SESSION['role'])) {
4   if ($_SESSION['role'] == 'Tutor') {
5     include "../DB_connection.php";
6     include "data/session.php";
7
8 ?>
```

For files that don't include HTML but instead is just business logic, like actions from certain buttons, we can also clearly define the accessibility at the beginning of said files as below. In this example, this is a specific action only allowed for administrators:

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['adminid']) || $_SESSION['role'] !== 'Admin') {
4     header("Location: ../../login.php?error=Unauthorized access");
5     exit;
6 }
```

4.2 SQL Handling

We separate SQL commands from the rest of our PHP code, defining them as callable functions, and calling them from other files. These commands include data fetching, inserting, updating, and deleting. The sample code below is for gathering all students and their information from the student account table:

```
1 <?php
2 // Fetch all student info from the database
3 global $conn;
```

```
function fetchStudentInfo($conn, $studentid) {
    $stmt = $conn->prepare("SELECT * FROM student_account WHERE accountid = ?");
    $stmt->execute([$studentid]);
    return $stmt->fetch(PDO::FETCH_ASSOC);
}
?>
```

This function requires the global connection variable, which is defined from our database connection file. This is the connection to our database and is essential for the functioning of these SQL commands.

4.3 Database Connection

For connecting to the database, we specify the connection credentials with server name, username, password and database name. For demonstration purposes, we have them set only to "localhost" as our database is indeed hosted locally on a given machine. For deployment, these values will be configured differently and scalability as well as database server backups can be achieved.

```
<?php
// Show all errors
error_reporting(E_ALL);
ini_set('display_errors', 1);

$sName = "localhost";
$uName = "root";
$pass = "";
$db_name = "vgtutor";

try {
    $conn = new PDO("mysql:host=$sName;dbname=$db_name", $uName, $pass);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Temporary output for test
} catch (PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
    exit;
}
?>
```

We feel that these code snippets are the most important and representative of the core functionalities of our Web Application.

5 Conclusion

Completing the VGtUtor web application project has been an enlightening and transformative experience for our entire team. Throughout this journey, we have not only applied our theoretical knowledge from university courses but also developed a deeper understanding of real-world software development practices. From designing system architecture and managing databases to implementing secure login procedures and ensuring user interaction flows are intuitive, each component of the project has challenged us to grow both technically and collaboratively. This project allowed us to understand the value of creating user-centered platforms that serve genuine needs within a community—in this case, the academic and tutoring needs of VGU students.
Beyond the technical aspects, this project has taught us the importance of teamwork, communication, and adaptability. We experienced firsthand how ideas evolve from abstract concepts into functional software, and how critical testing, iteration, and feedback are to the development process. Emotionally, it was rewarding to see our vision take shape and serve a purpose: empowering students to support one another academically. Although this marks the conclusion of our development timeline, we see VGtUtor as just the beginning of many impactful solutions we hope to contribute to. There is still room for expansion—particularly in areas like mobile accessibility and advanced recommendation algorithms—but this version stands as a meaningful and iconic milestone in our academic journey.

Our project repository: https://github.com/manhneee/vgtutor.git