HO CHI MINH CITY UNIVERSITY OF TRANSPORT

----o0o----

SOFTWARE ENGINEERING

TOPIC

STUDENT ATTENDANCE SYSTEM

Group : 14

Members:

- **Trần Thanh Thuận - 086206000354**
- **Phạm Xuân Thường - 067206006450**
- **Nguyễn Phương Trâm - 082306009692**
- **Phan Thị Thuỳ Quyên - 086304000733**
- **Phan Nhật Tài - 075206023010**

Contents

Chapter 1 : Requirement Specification

I. Functional Requirements

The school is required to develop an online student attendance system that provides the following core functionalities:

1.1 User Authentication (Login & Password Recovery)

- The system shall support three types of users: Admin, Teacher, and Student.
- Each user shall be provided with a unique account by the Admin, including a username and password. Username can be an email address.
- The system shall authenticate users based on the entered username and password.
- Users who access via link or have not logged-in will be called Guest
- Only authenticated users shall be allowed to access the system.
- The system shall provide a Forgot Password function:
+ Users enter their registered email address.
+ The system sends a password reset link or a new password via email.

1.2 View and Search Attendance Reports

1.2.1 For Students

- Students shall be able to view the attendance history of all classes they are joined by the admin.
- The attendance list shall be sortable in ascending or descending order by:
+ Date and time
+ Class code
- Students shall be able to search their attendance history by:
+ Class code
+ Attendance date
- The system shall display attendance records filtered by class and date.

1.2.2 For Teachers

- Teachers shall be able to view the attendance history of all classes they teach.
- The attendance list shall be sortable in ascending or descending order by:
+ Date and time
+ Class code
- Teachers shall be able to search attendance sessions by:

+ Class code
+ Teaching date
- The system shall display attendance sessions grouped by class and date.

1.3 Attendance Session Management (Teacher)

- The Teacher creates an attendance code for a class attendance
- Each attendance session includes the following information: class code, start time, end time, and session status (Open / Closed).
- Each attendance session is applied to only one class meeting. A class may have multiple attendance sessions during a semester.
- The system stores attendance session information and records the start time and end time in the database.
- The system automatically closes the attendance session when the defined time expires.
- The attendance session information includes:
+ Class code
+ Start time
+ End time
+ Session status (Open / Closed)
+ Attendance method (QR / Code + Password/Token / Auto-login / Manual)
+ QR scanning time (QR window, configurable)
+ Late attendance time (late window, configurable)

1.4 Attendance Methods

1.4.1 Attendance via code:

- The Teacher generates a code with a password or token
- The code or token is valid for a minimum duration of 2 minutes and will be automatically closed after the time expires.

1.4.2 Attendance via QR code:

- The Teacher generates QR code and share it for student
- Each QR code is valid for only 30 seconds. After that, a new QR code is generated for students to scan. The number of QR codes is customized by the Teacher.

1.4.3 Manual attendance:

- The Teacher can take attendance manually by calling each student’s name and updating the attendance list by selecting one of the following statuses:

+ Present
+ Late
+ Absent
- Manual attendance is used in cases where students arrive late or have valid reasons that prevent them from submitting attendance online.

1.5 Submit Attendance (Student)

- Each class shall have a specific attendance submission time window.
- Students shall only be allowed to submit attendance within the permitted time period using one of the following three methods: QR code scanning, access code with password/token, or manual attendance
- Attendance submissions outside the allowed time window shall be automatically marked with the corresponding status, such as:
+ Late
+ Absent

1.6. Manage User Accounts

- Administrators shall be able to:
+ Create new user accounts
+ Edit user account information
+ Delete user accounts
- Administrators shall be able to assign roles to users:
+ Administrator
+ Teacher
+ Student

1.7. Manage Classes

- Administrators shall be able to create, update, and delete classes.
- Each class shall be associated with:
+ A class code
+ A teacher in charge
+ A list of enrolled students

1.8. View Dashboard

The system shall provide a role-based dashboard interface for each authenticated user.

+ Studentsa

The system shall provide students with a dashboard displaying their enrolled classes and current attendance status.

+ Teacher

The system shall provide teachers with a dashboard displaying their assigned classes and scheduled sessions.

+ Admin

The system shall provide administrators with a dedicated dashboard for system-level management.

II. Non-Functional Requirements

Usability:
The user interface shall be responsive and adaptable to different screen sizes. The system shall be able operate functions under 3 steps, it shall be <9 buttons per screen.

Performance:
The system shall support approximately 2,000 concurrent users and under 3 seconds responding. The system shall be capable of storing attendance data for 10,000+ students and 50+ courses.

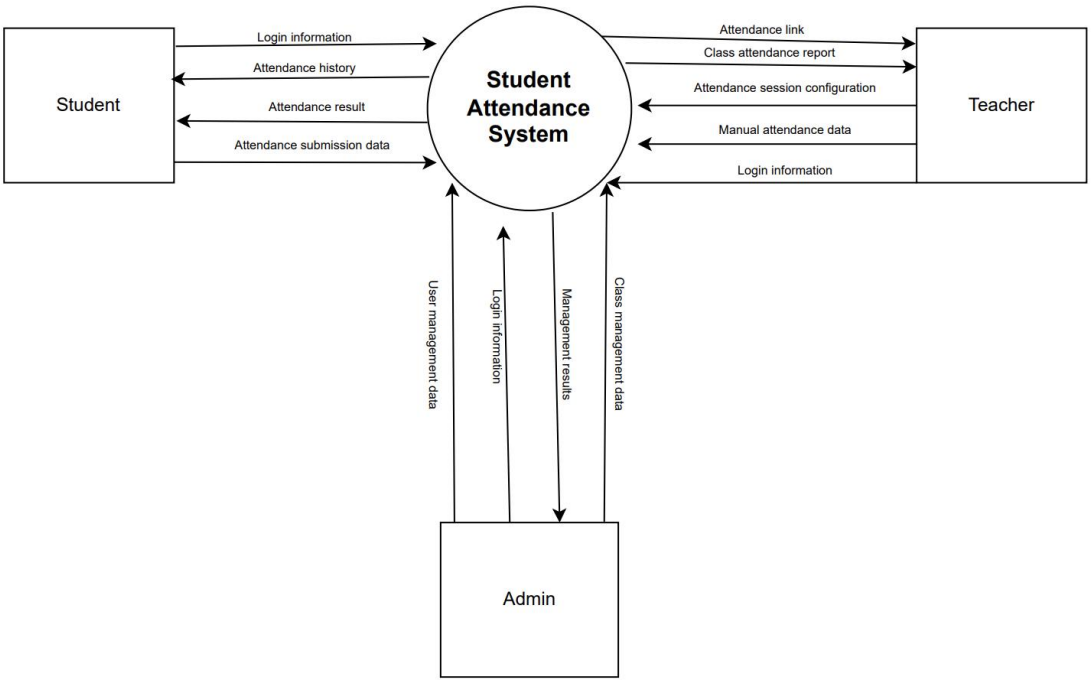
Security:
The system shall ensure that students can only access their own attendance data. Security measures shall be applied to prevent common vulnerabilities such as SQL Injection and Cross-Site Scripting (XSS). Automated activities shall be restricted to prevent fraud or mass data submission.

Reliability:
The system shall operate continuously 24/7 with a minimum reliability of 99.5% uptime under normal operating conditions.

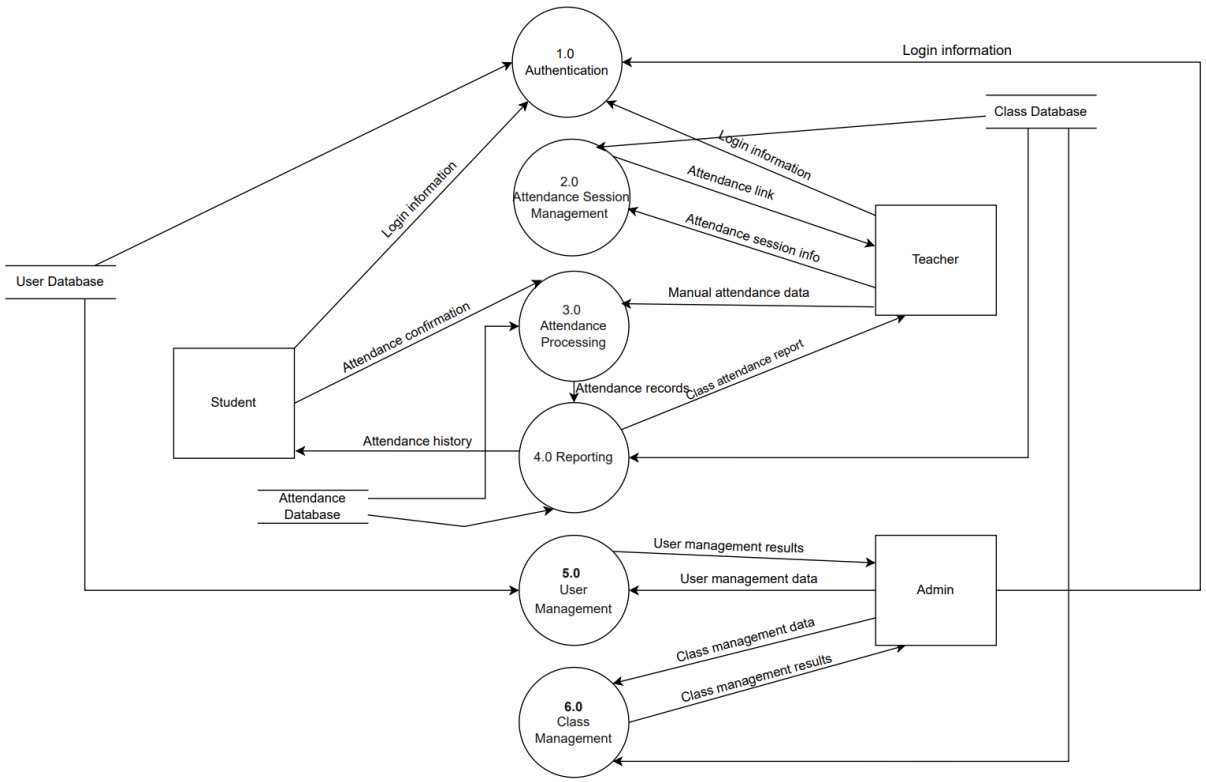
Design Constraints:
The system shall be developed using **Python, SQLite3, and Custom Tkinter** for the graphical user interface.

III. Data Flow Diagram

0-Level DFD



1-Level DFD


1.0
Authentication
Login information
Class Database
Login information
Attendance link
Attendance session info
2.0
Attendance Session
Management
Teacher
Login information
User Database
Manual attendance data
3.0
Attendance
Processing
Attendance confirmation
Student
Attendance records
Class attendance report
Attendance history
4.0 Reporting
Attendance
Database
User management results
User management data
5.0
User
Management
Admin
Class management data
Class management results
6.0
Class
Management

IV. Use case Diagram

1. Overview diagram

1.2. List of Actors

No.	Actor	Description
1	Guest	Guest is user that only access to system by link but they haven't logged in yet
2	Admin	Manages the system, users, classes, and system-wide reports
3	Teacher	Manages assigned classes, creates attendance, views and edit attendance Class
4	Student	Submits attendance and views personal attendance history

1.3. List of Use Cases

No.	Use-case	Description
1	Login	Log in to the system
2	Reset Password	Recover user password
3	Edit Profile	Update personal information
4	View Attendance History	View personal attendance history
5	Submit Attendance	Submit attendance for a class
6	Manage Attendance Sessions	Custom a new attendance session for a class
7	View System Status	View current system status and activity
8	Manage System	Manage system configuration and settings
9	Manage Users	Allow admin to view and edit users infor.
10	View Dashboard	Allow users to view attendance statistics and summary reports in charts or tables

2. Use-case Login

2.1. Summary

This use case describes how a user (Student, Teacher, or Administrator) logs in to the Student Attendance System in order to access system functions.

2.2. Event Flow

2.2.1. Main Flow

1. The system displays the login interface and requests the user enters their **assigned username (or Student/Teacher ID)** and **password**.
2. The user enters login information and submits the login request.
3. The system verifies the entered credentials.
4. If the credentials are valid, the system allows the user to log in and redirects the user to the main interface based on their role.

2.2.2. Alternative Flows

2.2.2.1. Invalid Username or Password

If the credentials are incorrect, the system displays an error message (e.g., "Invalid username or password").

The user may re-enter the information or cancel the login process.

2.3. Special Requirements

Passwords must be hidden during entry for security.

2.4. Pre-condition

The user account has been pre-created and activated by the Administrator

The user has received their login credentials.

The user is not logged into the system.

2.5. Post-condition

If the use-case is successful, the user is logged into the system. Otherwise, the system state remains unchanged.

2.6. Extension Points

Use-case Reset Password: The user selects this option if they forget their password.

3. Use-case Reset Password

3.1. Summary

This use case describes how a user resets their password when it is forgotten.

3.2. Event Flow

3.2.1. Main Flow

This use-case begins when the user selects the Reset password function.

1. The system requires the user to enter email information.
2. The system checks whether email exists in the system. If the email exists, the system will send a new password to the user's email and notify the user of the password recovery results.

3.2.2. Alternative Flows

If the email does not exist in the system, the system displays an error message.

3.3. Special Requirements

None.

3.4. Pre-condition

None.

3.5. Post-condition

If successful, the user’s password is updated. Otherwise, the system state and password remain unchanged.

3.6. Extension Points

None.

4. Use-case Edit Profile

4.1. Summary

This use-case allows users to update their personal information after logging into the system.

4.2. Event Flow

4.2.1. Main Flow

1. The student selects the Edit Profile function.
2. The system displays the current personal information.
3. The student modifies the desired information.
4. The student confirms the update.
5. The system saves the updated information.

4.2.2. Alternative Flows

The student cancels the update, and no data is changed.

4.3. Special Requirements

None.

4.4. Pre-condition

The student is logged into the system.

4.5. Post-condition

If successful, the personal information is updated. Otherwise, the system state remains unchanged.

4.6. Extension Points

None.

5. Use-case View Attendance History

5.1. Summary

This use-case allows **Students** to view their own attendance history and **Teachers** to view the attendance history of students under their responsibility.

5.2. Flow of Events

5.2.1. Main Flow – Student

The use-case begins when the Student selects the View Attendance History function.

1. The system retrieves the Student's attendance records.
2. The system displays a list of attendance records.
3. The system provides filter options such as:
- Date range
- Attendance status (present, absent, late)
4. The Student selects filter criteria.
5. The system displays the filtered attendance history.
6. The Student views the attendance details.

5.2.2. Main Flow – Teacher

The use-case begins when the Teacher selects the View Attendance History function.

1. The system retrieves attendance records of classes or students managed by the Teacher.
2. The system displays a list of attendance records.
3. The system provides filter options such as:
- Student
- Class
- Date range
- Attendance status (present, absent, late)
4. The Teacher selects filter criteria.
5. The system displays the filtered attendance history.
6. The Teacher views the attendance details.

5.2.3. Alternative Flows

If no attendance records match the filter criteria, the system displays a message.

5.4. Special Requirements

None.

5.5. Pre-condition

The Student or Teacher is logged into the system.

5.6. Post-condition

The system state remains unchanged.

5.7. Extension Points

None.

6. Use-case Submit Attendance

6.1. Summary

This use-case allows a student to submit attendance for a class session.

6.2. Flow of Events

6.2.1. Main Flow

1. The student selects the class session to attend.
2. The system displays attendance information, including:
- Attendance status
- Available attendance methods (QR code / Attendance code).
3. The student performs one of the following actions:
- Scan the QR code provided by the teacher, **or** enter the attendance code to submit attendance.
4. The system validates the QR code or token.
5. The student confirms attendance.
6. The system records the attendance result.
7. The system notifies the student of successful attendance submission.

6.2.2. Alternative Flows

A1. Attendance session is closed or expired
If the attendance session has expired, the system displays a message and does not allow submission.

A2. Invalid or expired QR code / token
If the QR code or token is invalid or expired, the system displays an error message and rejects the attendance submission.

A3. Student already submitted attendance

If the student has already submitted attendance for the session, the system informs the student and prevents duplicate submission.

6.3. Special Requirements

Each QR code or attendance code must contain a **unique token**.

Tokens must be **time-limited** and **single-use per student**.

The system must prevent attendance submission from unauthorized users.

Attendance data must be recorded accurately and securely.

6.4. Pre-condition

The student is logged into the system.

The attendance session is active.

6.5. Post-condition

If successful, the attendance record is updated. Otherwise, the system state remains unchanged.

6.6. Extension Points

None

7. Use-case Manage Attendance Session

7.1. Summary

This use-case allows the **teacher** to manage attendance activities in the system, including **viewing, creating, and editing** attendance information.

7.2. Flow of Events

7.2.1. Main Flow

This use-case begins when the teacher selects the **Manage Attendance Sessions** function.

1. The system displays a list of class’s information (View Attendance) and two methods for taking attendance(QR and code).
2. If the teacher selects "Create Attendance" and chooses a method, the "Create Attendance" process will be executed.
3. If the teacher selects the class for which they need to edit attendance, the "Edit Attendance" process will be executed.

7.2.1.1. View Attendance

1. The system shall display a list of classes after the user selects the attendance management function.
2. Each class entry shall include the class name, session date, and the number of attendance status indicators.
3. The system shall allow users to export the attendance list to csv files.

7.2.1.2. Create Attendance

1. The system requests the teacher to enter attendance session information (class, time, status).
2. The teacher enters the required information and confirms the creation.
3. The system validates the entered information.
4. The system creates a new attendance session and attendance via two methods (QR code, attendance code).
5. The system displays the result of the creation process by a message.

7.2.1.3. Edit Attendance

1. The system displays the attendance information of the selected class or session.
2. The teacher edits the attendance information as required.
3. The teacher confirms the update.
4. The system validates the updated information.
5. The system updates the attendance data in the system.

6. The system displays the result of the update process.

7.2.2. Alternative Flows

A1. Attendance session is closed or expired

If the attendance session has expired. The system prevents the teacher from reopening the session unless permitted by system policy and allows the teacher to view attendance records in read-only mode.

A2. Invalid or expired QR code / attendance code

If the QR code or attendance code generated for the session has expired. The system will send a message that the QR/code has expired. The system shall automatically generate a new QR code or attendance code if the session is still open and update the displayed QR/code in real time.

7.3. Special Requirements

None.

7.4. Pre-condition

The teacher is logged into the system.

7.5. Post-condition

If the use-case is successful, attendance information is created or updated in the system. Otherwise, the system state remains unchanged.

7.6. Extension Points

None.

8. Use Case View System Status

8.1. Summary

This use case allows the administrator to **view, analyze, and download system reports and analytical insights** related to attendance, academic performance, user activity, and system security.

8.2. Event Flow

8.2.1. Main Flow

1. This use case begins when the administrator selects the **System Status** function.

2. The system retrieves analytical data and available report information.

3. The system displays summary indicators, including:

    - Departmental attendance rate

    - Number of active students

    - Number of active modules

4. The system displays a list of available reports, including:

    - Monthly Attendance Summary

    - Faculty Performance Review

    - Student Retention Analysis

    - Security and Access Audit

5. The administrator selects a report to view or download.

6. The system generates the selected report in the requested format (e.g., PDF, XLSX, CSV).

7. The administrator reviews or downloads the generated report.

8. The use case ends.

8.2.2. Alternative Flows

8.2.2.1. Report data unavailable

1. If the system is unable to retrieve report data:

2. The system displays an error message indicating that the report data is temporarily unavailable.

3. The administrator may choose to retry or exit the Reports & Insights function.

8.2.2.2. Report generation fails

1. If an error occurs during report generation:

2. The system displays a notification indicating that the report could not be generated.

3. The administrator may retry generating the report or cancel the operation.

8.3. Special Requirements

The administrator is logged into the system.

8.4. Pre-conditions

- The administrator has successfully logged into the system

8.5. Post-conditions

- If the use case is successful, the selected report is displayed or downloaded by the administrator.

- The system data remains unchanged after viewing or downloading reports.

8.6. Extension Points

None.

9. Use-case Manage System

9.1. Summary

This use-case allows the administrator to manage and configure system settings of the Student Attendance System, including attendance rules and system parameters.

9.2. Flow of Events

9.2.1. Main Flow

This use-case begins when the administrator selects the **Manage System** function.

The system displays the current system configuration settings.

The administrator selects a system setting to manage.

9.2.2. Alternative Flows

9.2.2.1. Invalid Configuration Data

If the administrator enters invalid configuration values:

The system displays an error message.

The administrator may correct the data and retry or cancel the operation.

9.3. Special Requirements

None.

9.4. Pre-condition

The administrator is logged into the system.

9.5. Post-condition

If the use-case is successful, system configuration settings are updated in the system. Otherwise, the system state remains unchanged.

9.6. Extension Points

None.

10. Use-case Manage Users

10.1. Summary

This use-case allows the **administrator** to manage system users, including **viewing user lists and editing user information**.

10.2. Flow of Events

10.2.1. Main Flow

This use-case begins when the administrator selects the **Manage User** function.

1. The system displays a list of users in the system.
2. The administrator selects a user to manage.
3. If the administrator selects **Edit User**, the *Edit User* flow is executed.

10.2.1.1. Edit User

1. The system displays detailed information of the selected user.
2. The administrator edits user information (name, role, account status).
3. The administrator confirms the update.
4. The system validates the updated information.
5. The system updates the user information in the system.
6. The system displays the result of the update process.

10.2.2. Alternative Flows

None.

10.3. Special Requirements

None.

10.4. Pre-condition

The administrator is logged into the system.

10.5. Post-condition

If the use-case is successful, user information is updated in the system. Otherwise, the system state remains unchanged.

10.6. Extension Points

None.

11. Use-case View Dashboard

11.1. Summary

This use-case allows **users (Student, Teacher or Admin)** to view attendance statistics and summary information through a dashboard, depending on their role.

11.2. Flow of Events

11.2.1. Main Flow

1. This use-case begins when the user logs into the system.
2. The user selects the **View Attendance Dashboard** function.
3. The system identifies the user's role (**Student, Teacher or admin**).
4. The system retrieves attendance data based on the user's role.
5. The system displays the appropriate dashboard.
6. The use-case ends.

11.2.2. Role-based Flows

11.2.2.1. Student Dashboard Flow

If the user is a **Student**:

1. The system retrieves the student's personal attendance data.
2. The system displays the student dashboard, including:

- Attendance percentage
- Total lab time
- Number of absences
- Scholar rank

- Academic schedule
- Attendance verification log

3. The student reviews the displayed information.

11.2.2.2. Teacher Dashboard Flow

If the user is a **Teacher**:

1. The system retrieves attendance data of the teacher's assigned classes.
2. The system displays the teacher dashboard, including:

- Assigned subjects
- Total number of students
- Average attendance rate
- Attendance load chart
- Class schedule (today and upcoming)

3. The teacher reviews the displayed information.

11.2.2.3. Admin Dashboard Flow

If the user is a Admin:

1. The system retrieves attendance data of the teacher's assigned classes.
2. The system displays the teacher dashboard, including:

- Overall system health status
- System performance metrics (e.g., QR scanning latency, database connections)
- Attendance activity statistics
- Authentication and access control information
- System alerts and security-related events

3. The teacher reviews the displayed information.

11.2.3. Alternative Flows

11.2.3.1. No Attendance Data Available

If no attendance data is available:

- The system displays a message.
- The user may exit the dashboard.

11.3. Special Requirements

None.

11.4. Pre-condition

The user is logged into the system.

11.5. Post-condition

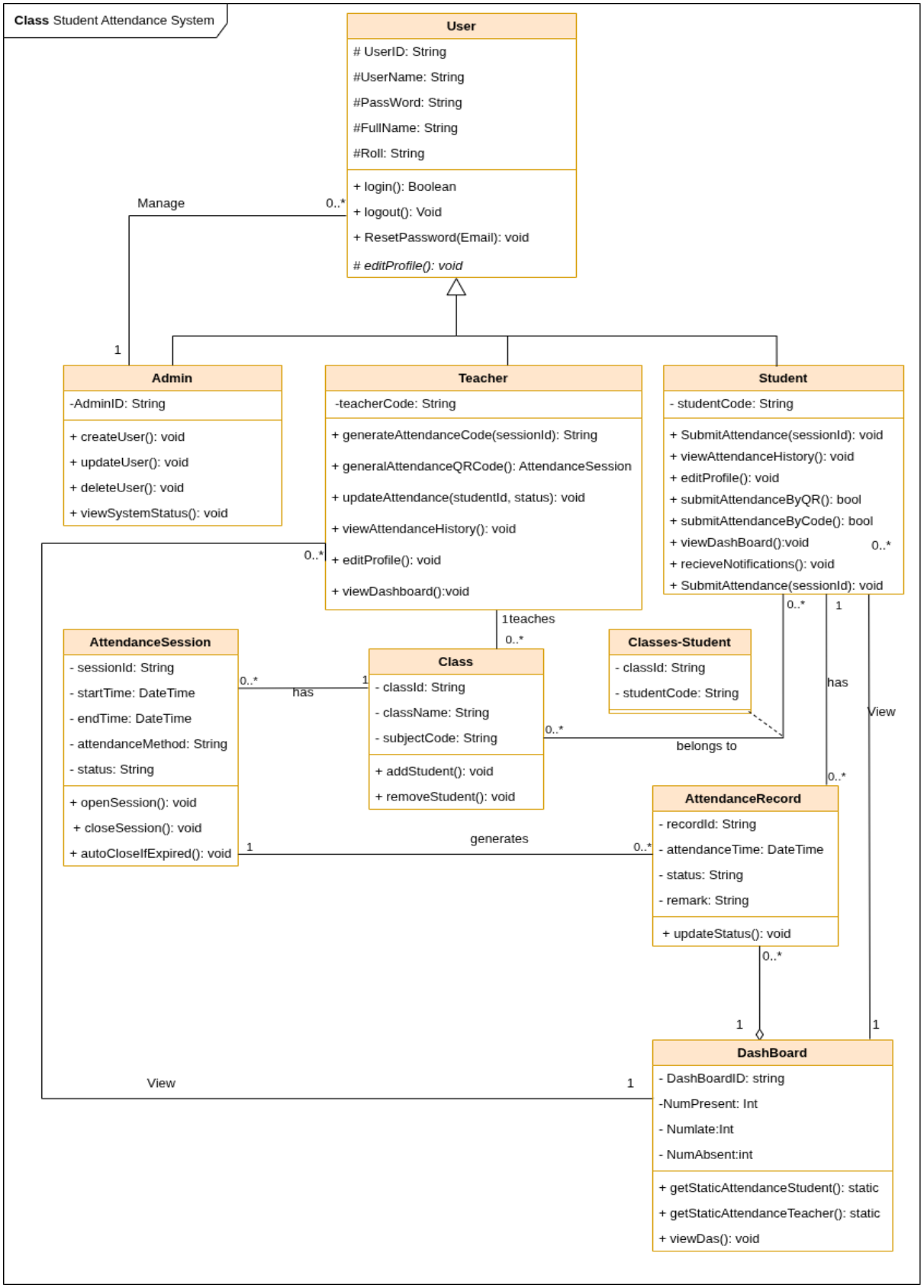
Attendance dashboard information is displayed according to the user’s role.

The system state remains unchanged.

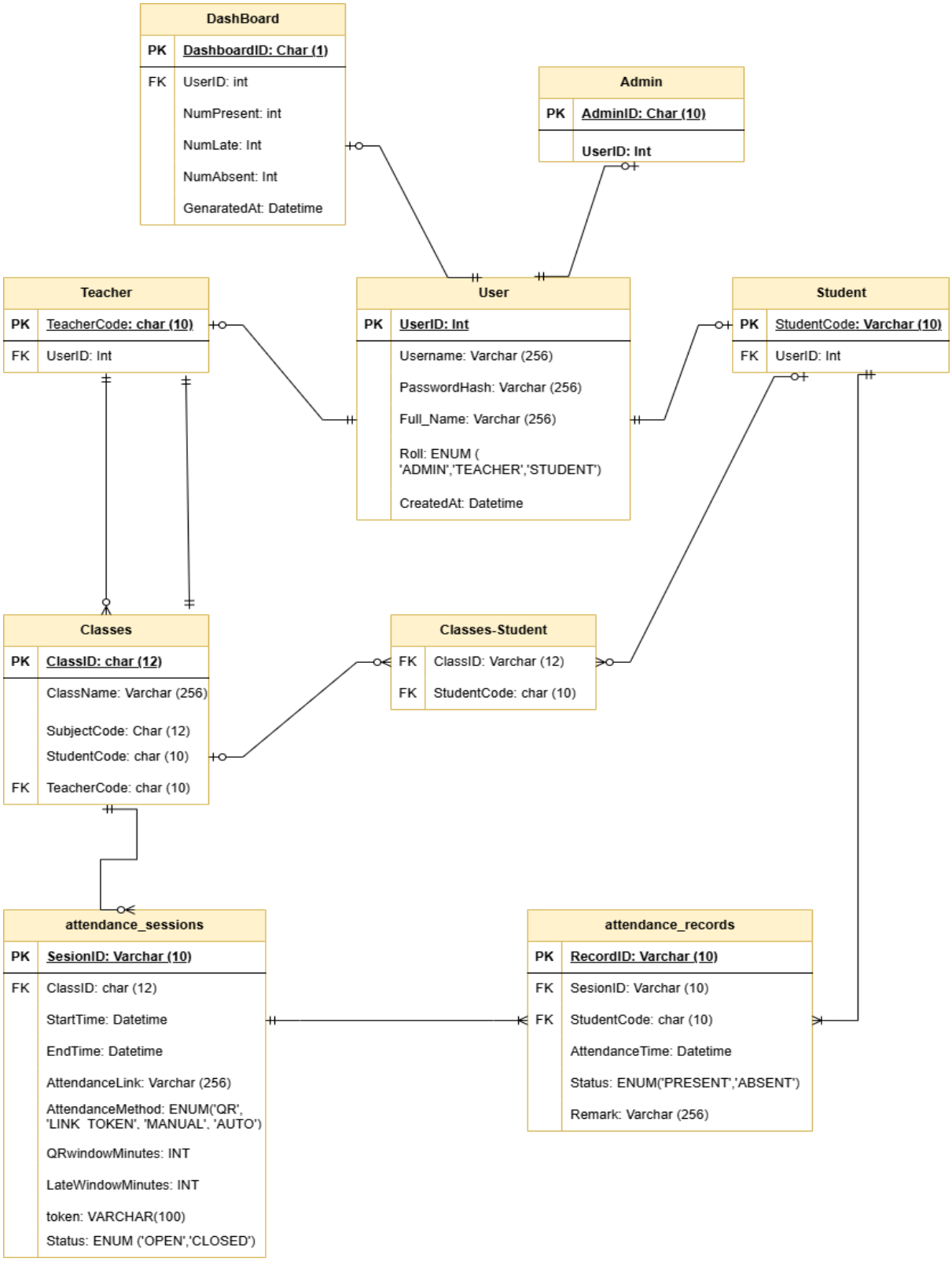
11.6. Extension Points

None.

V. Class Diagram



VI. Data Model

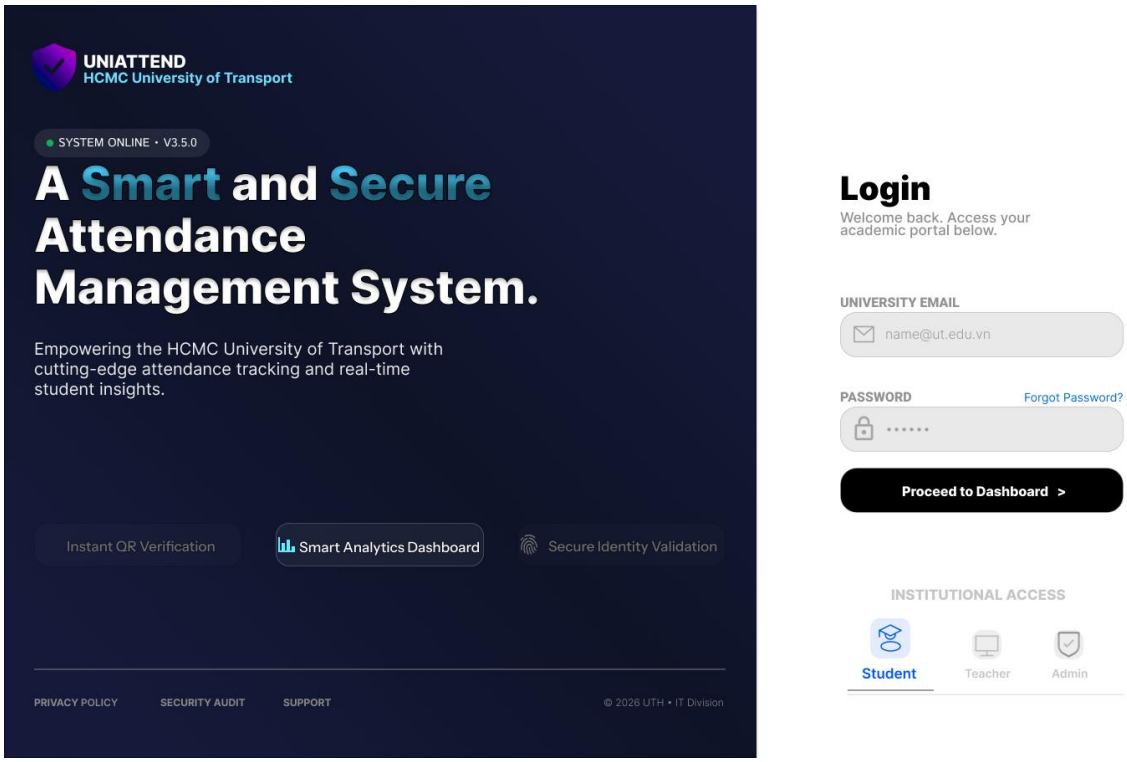


Chapter 2 : Design

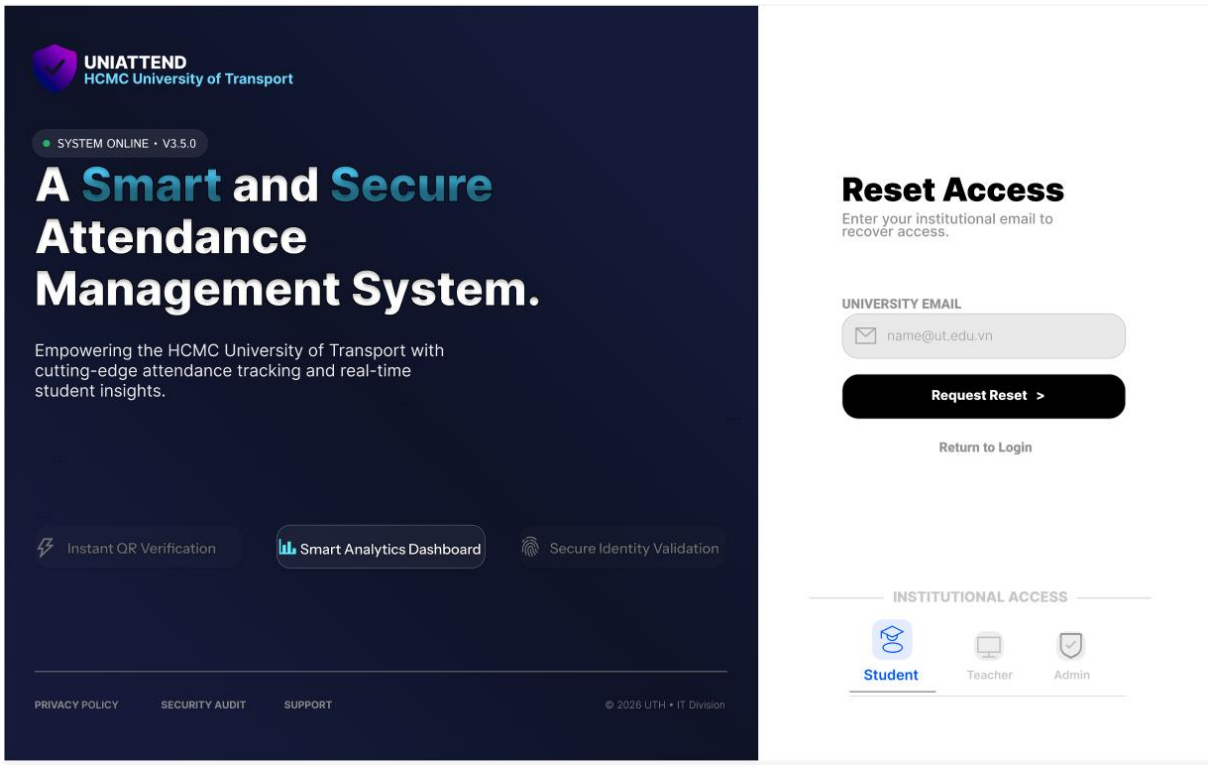
Interface Design Description.

1. **Guest.**

- Login:

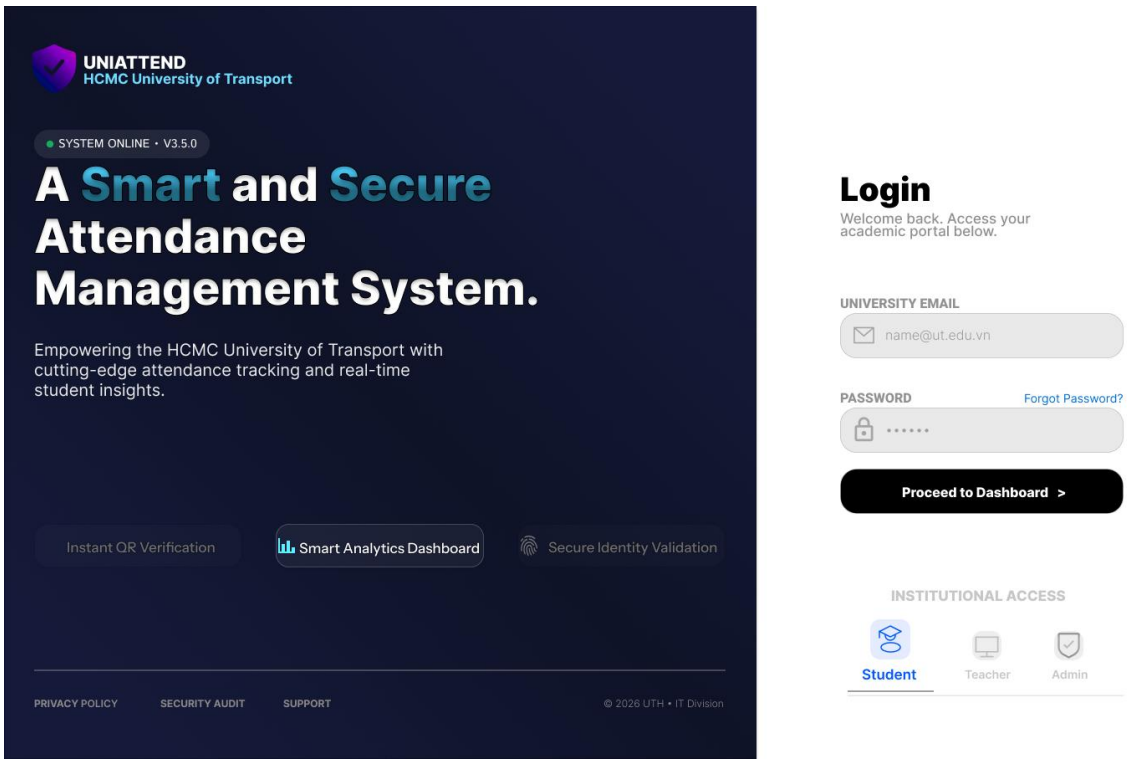


- Reset password:

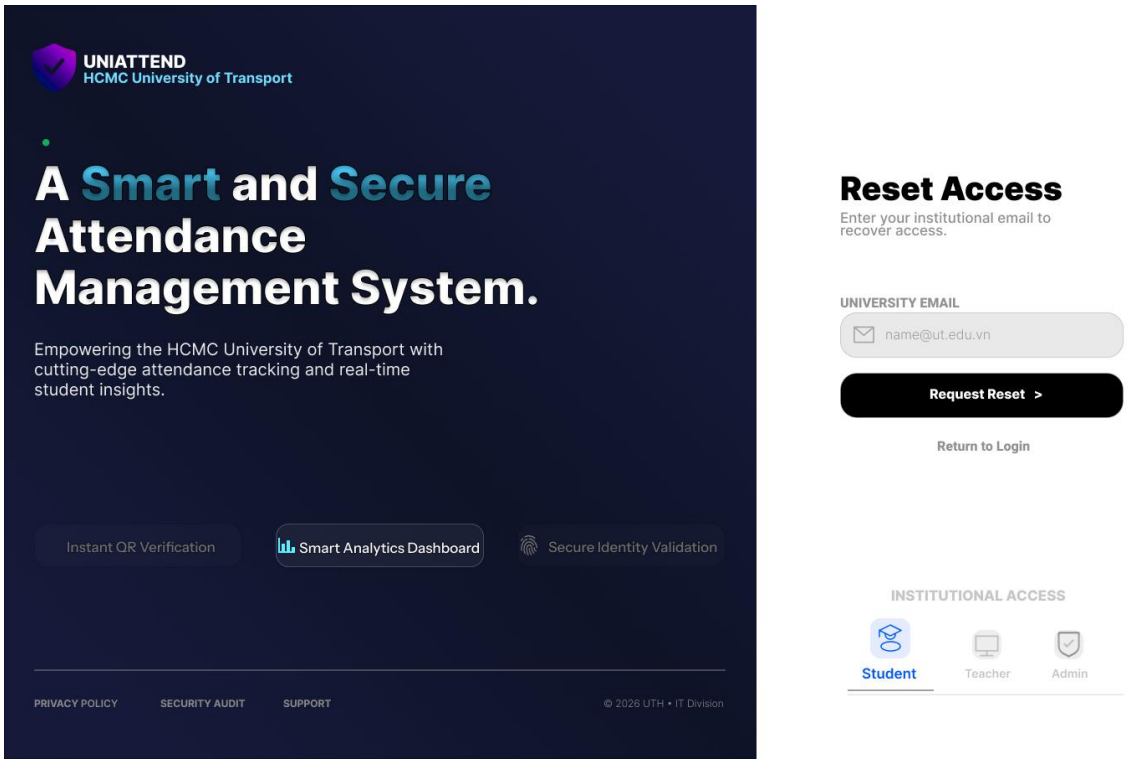


2. Student.

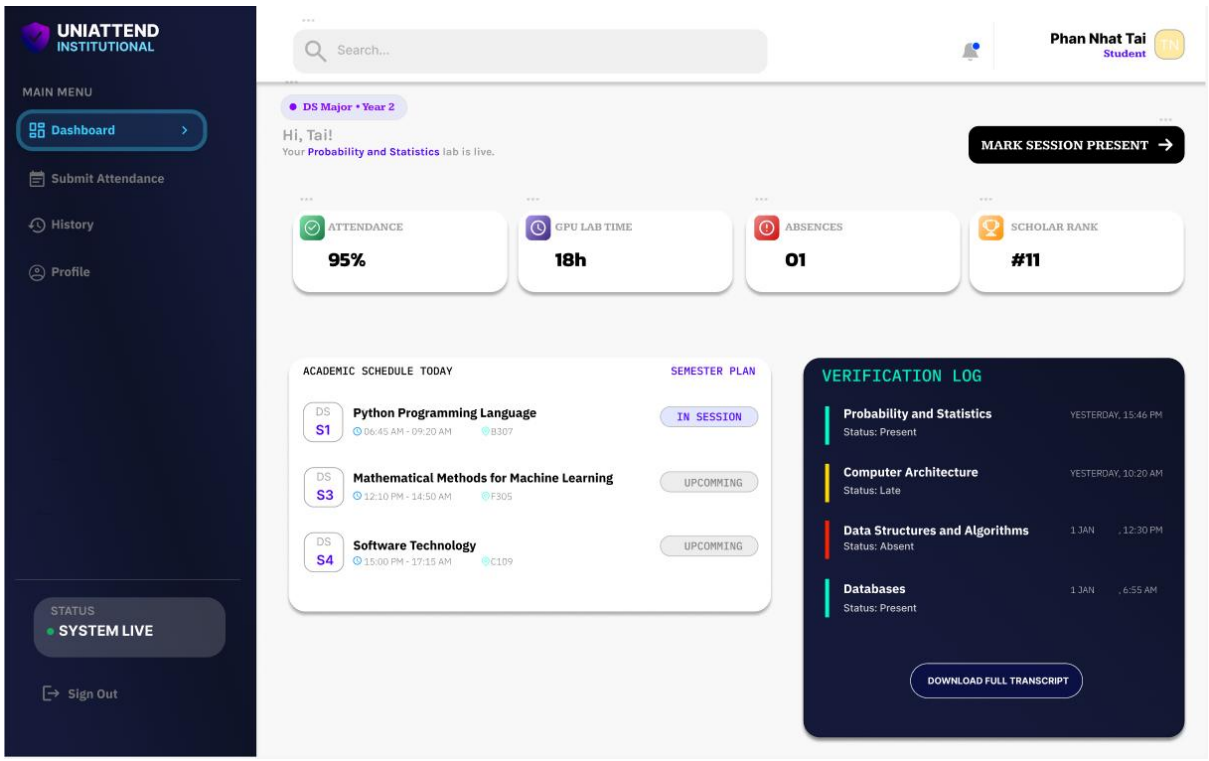
- Login:



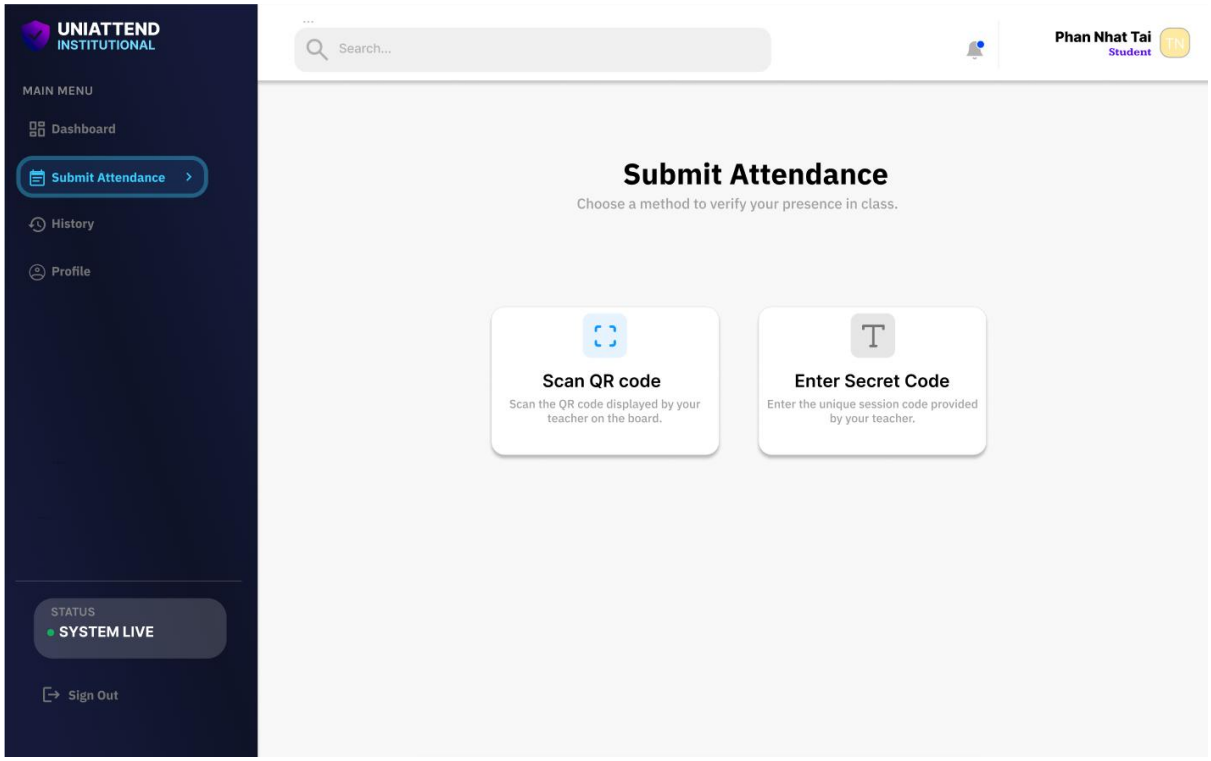
- Reset password:



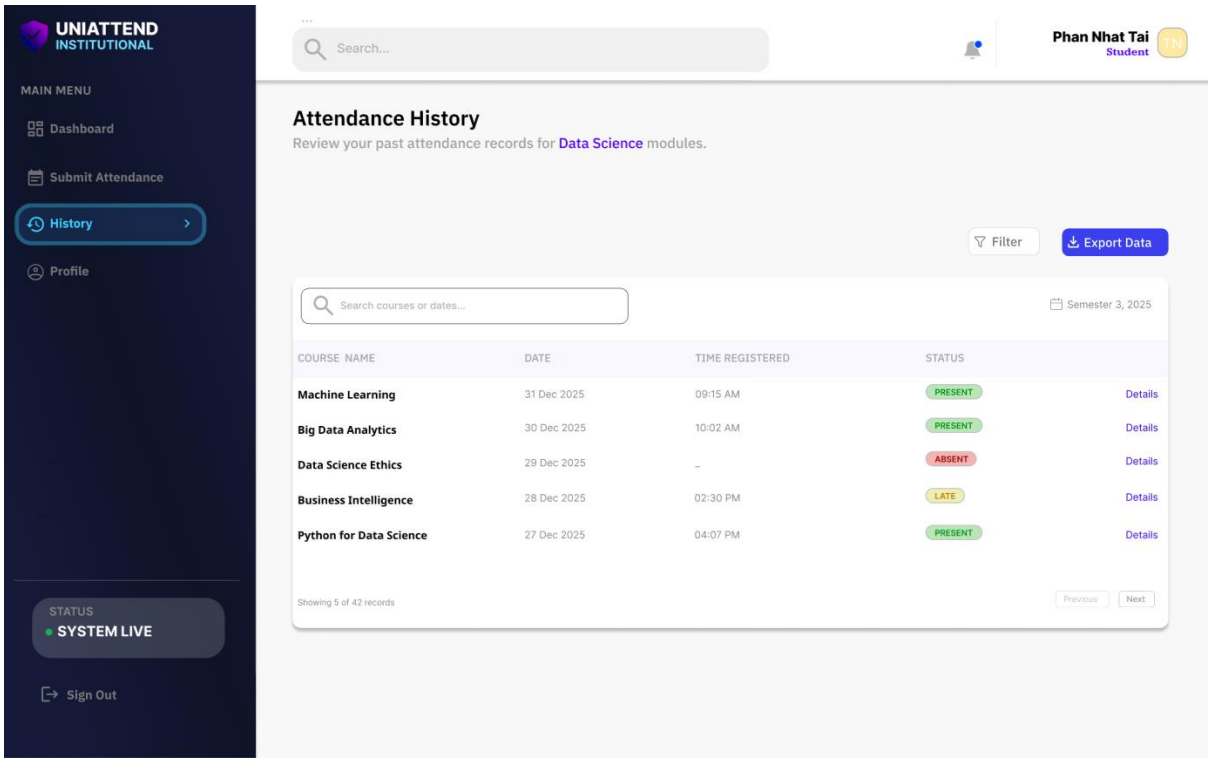
- View Dashboard:



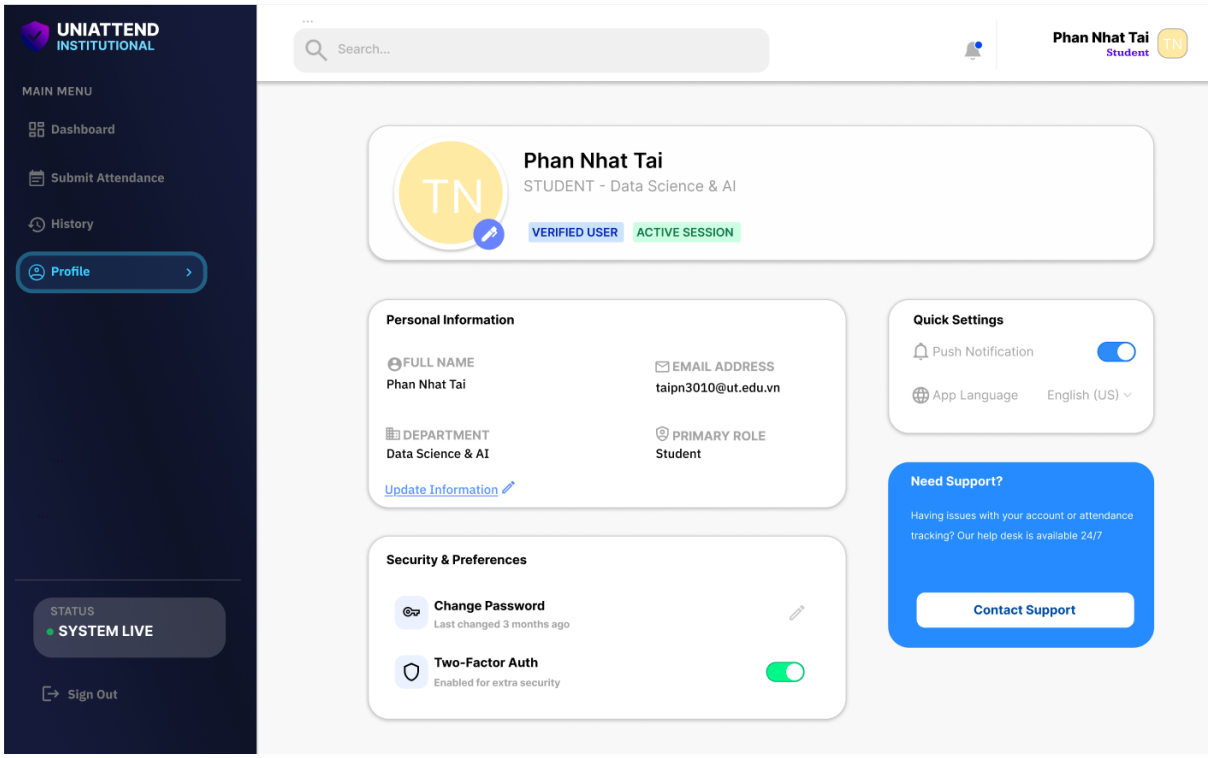
- Submit Attendance:



- View Attendance History:

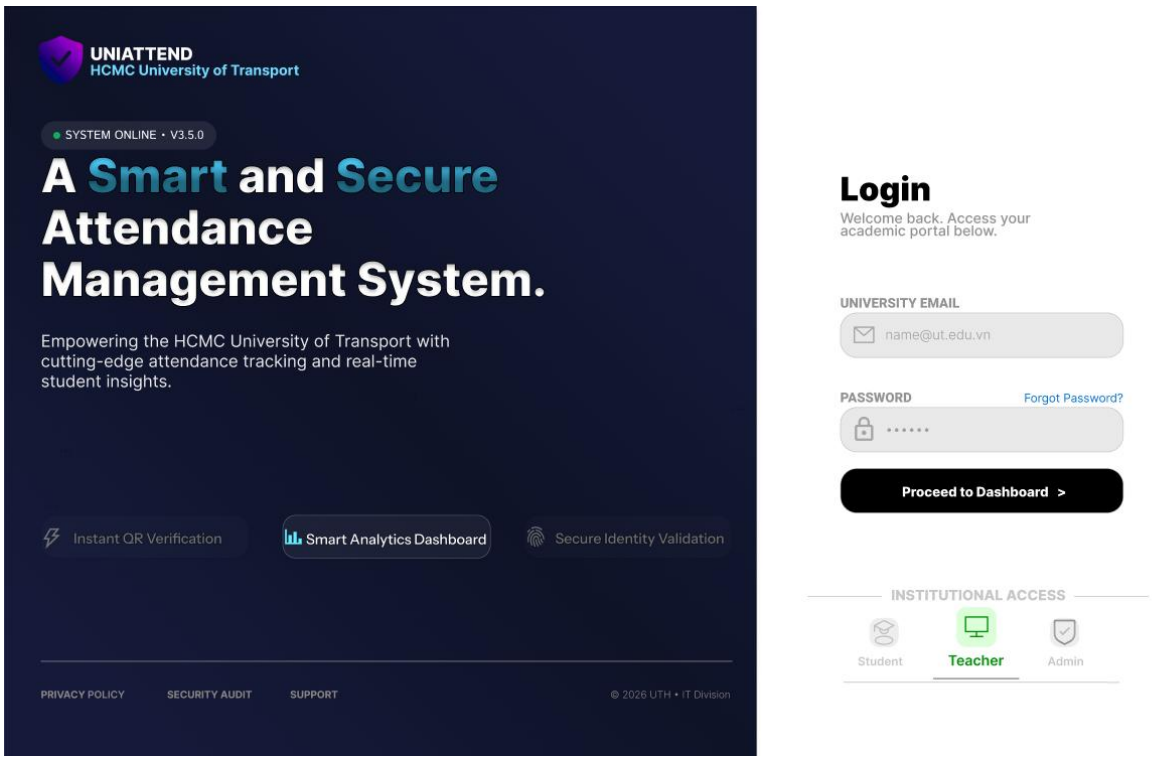


- Edit Profile:

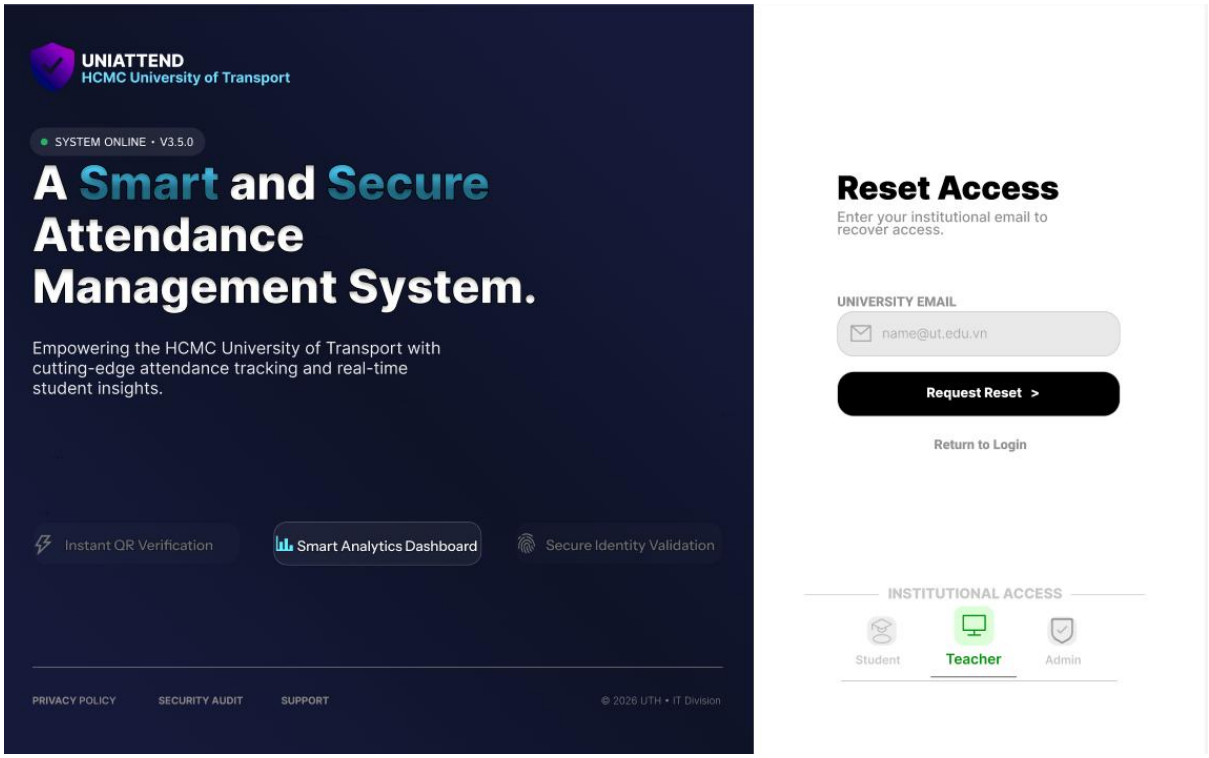


3. **Teacher.**

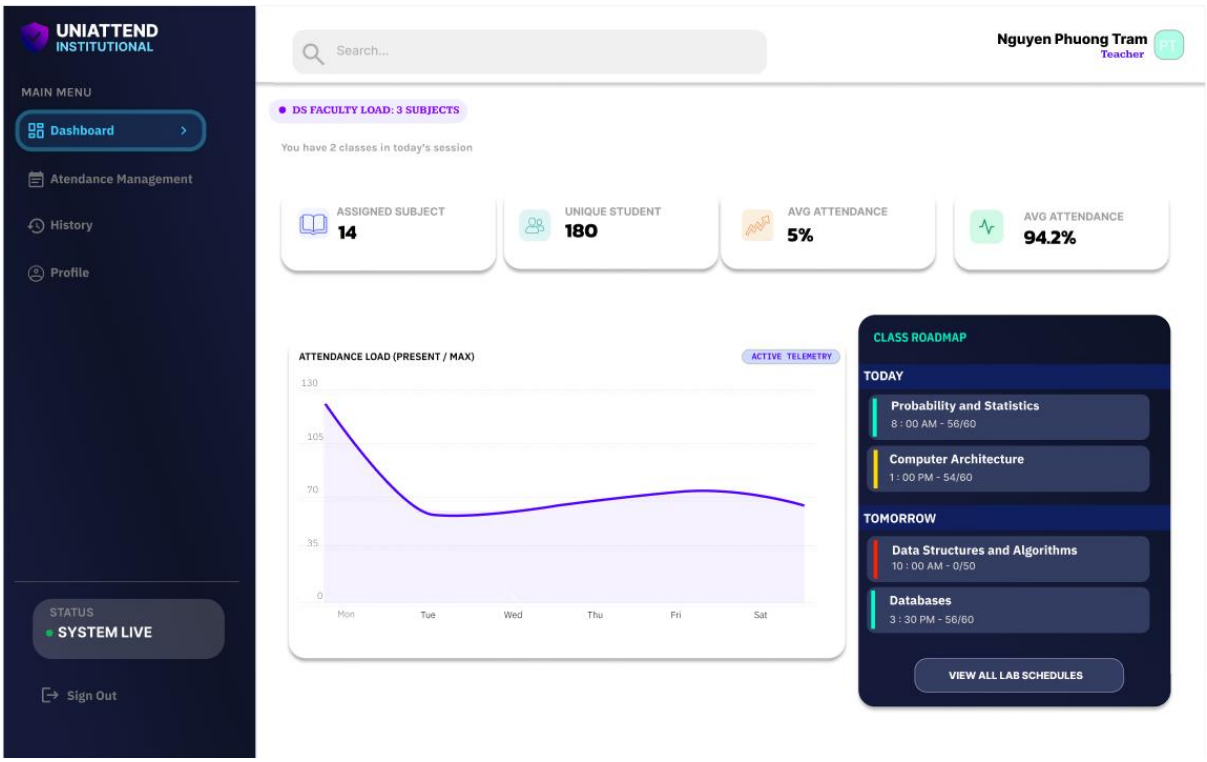
- Login:



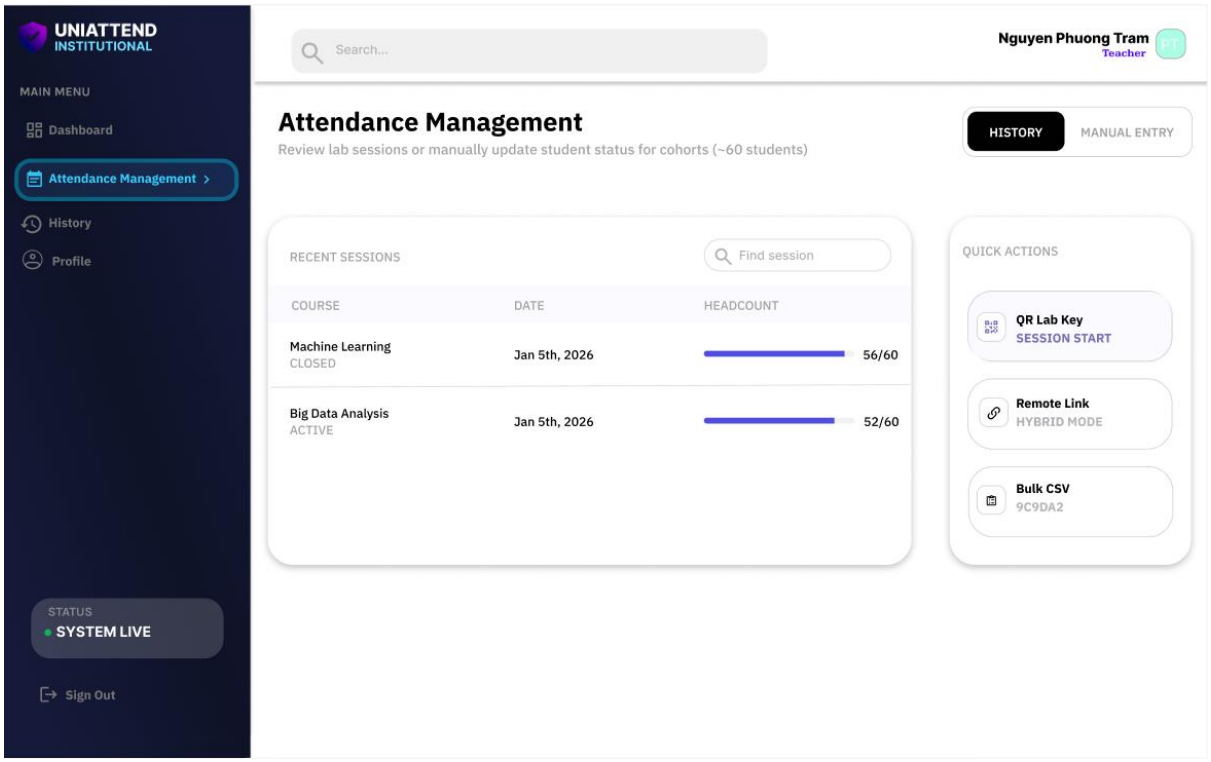
- Reset password:



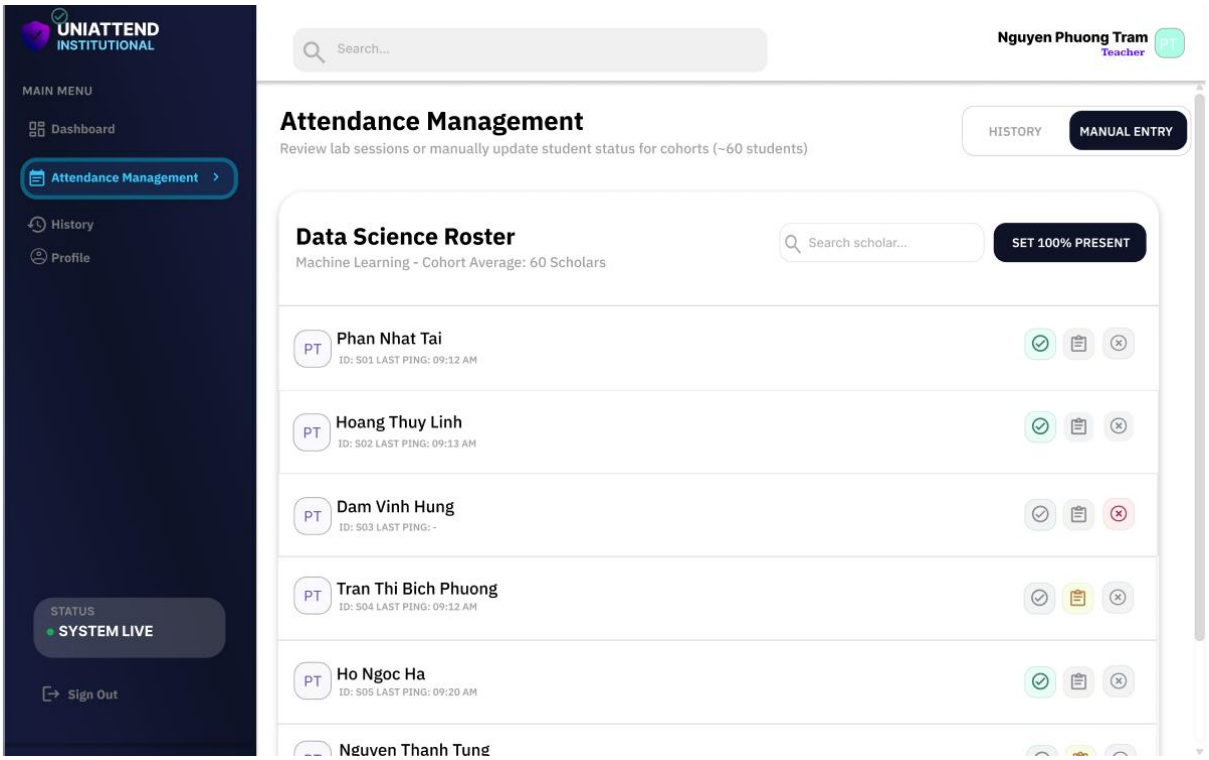
- View Dashboard:



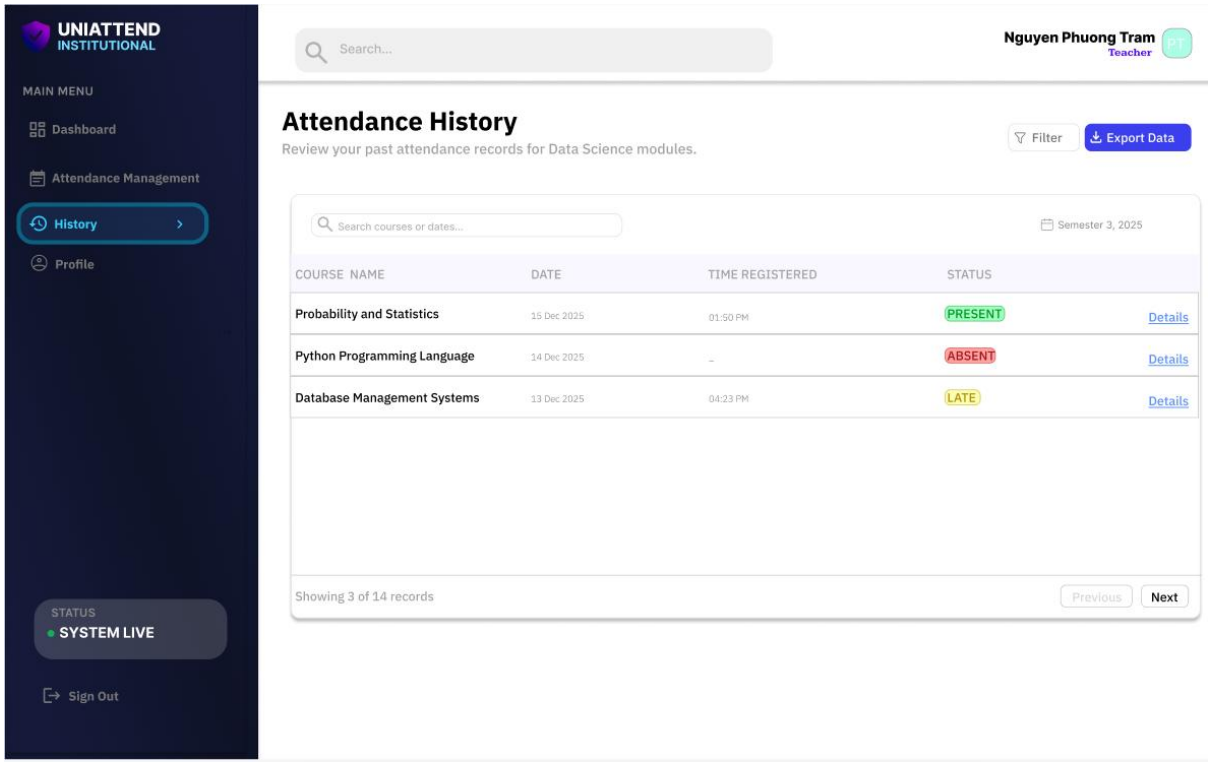
- Manage Attendance Sessions:



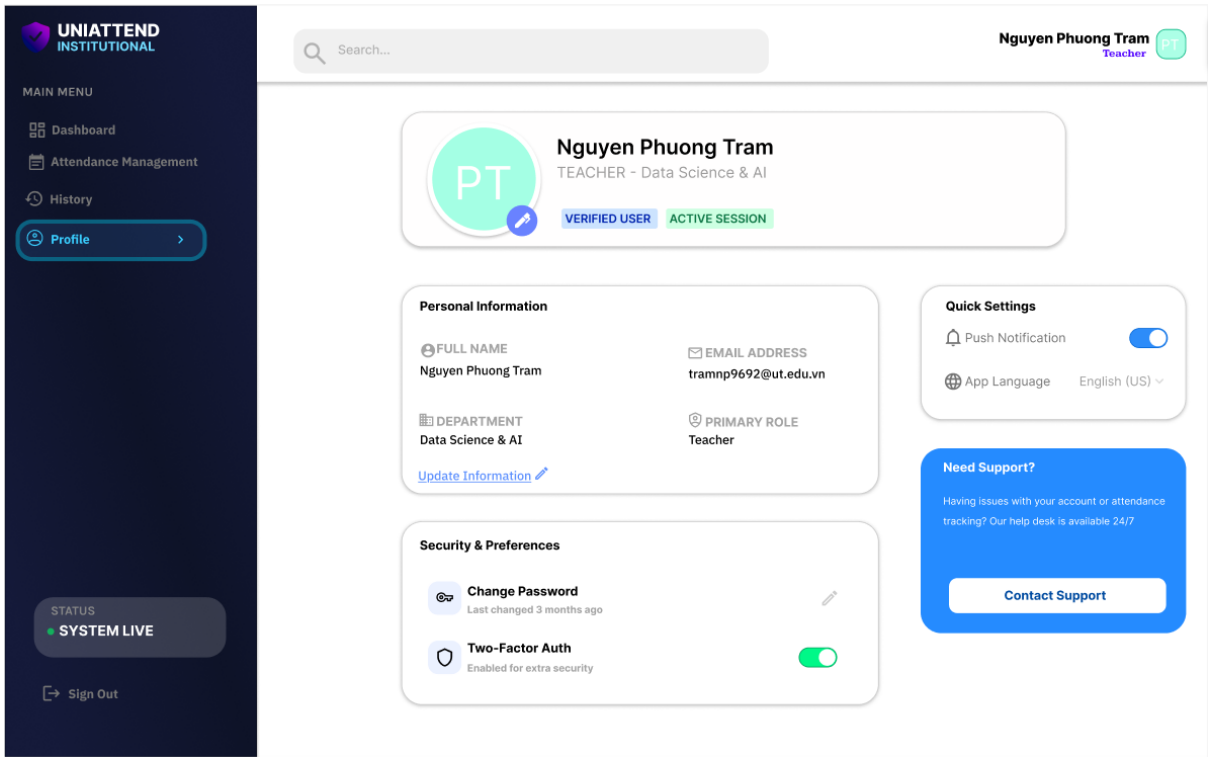
- Manage Attendance Sessions -View:



- View Attendance History:

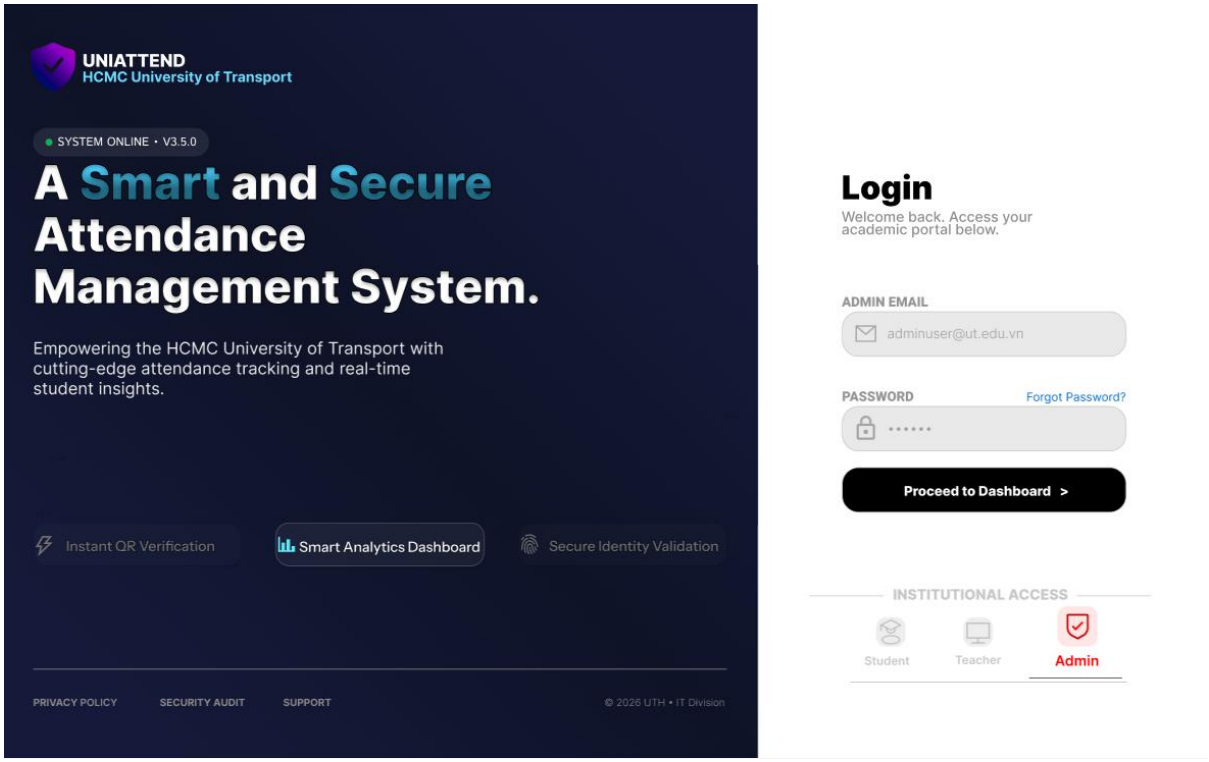


- Edit Profile:

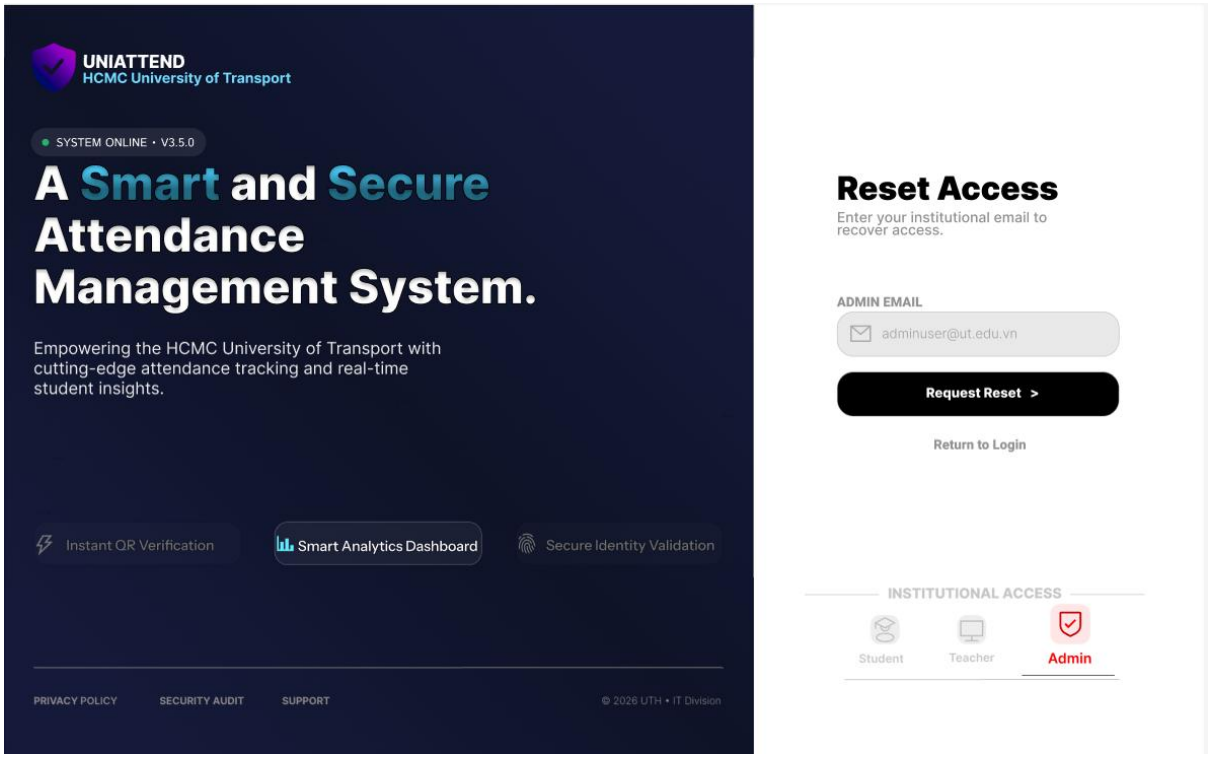


4. Admin.

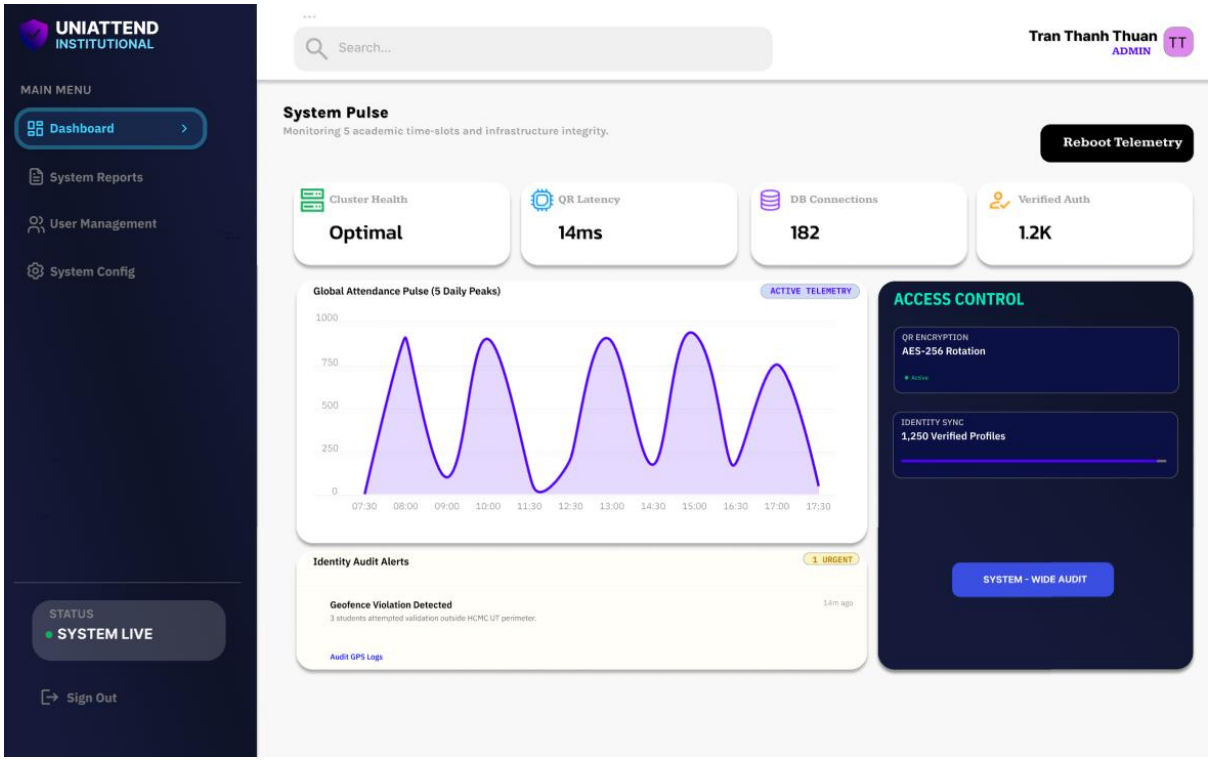
- Login:



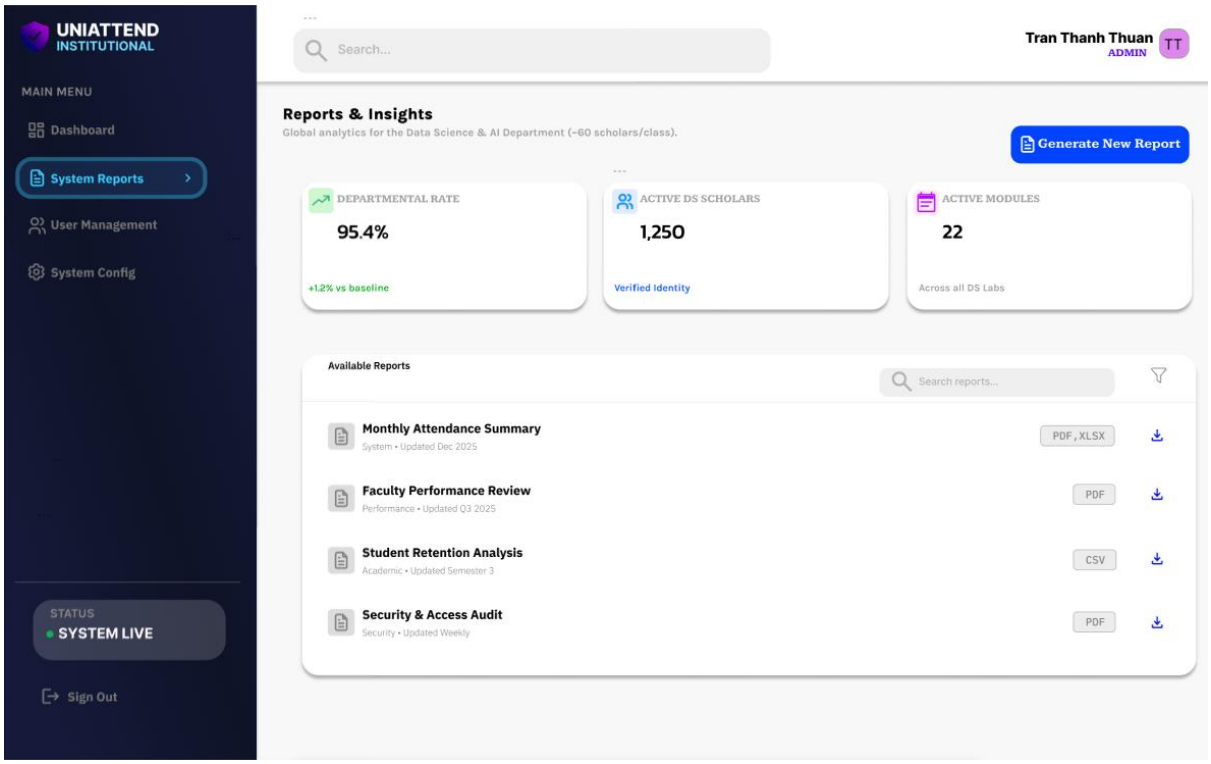
- Reset password:



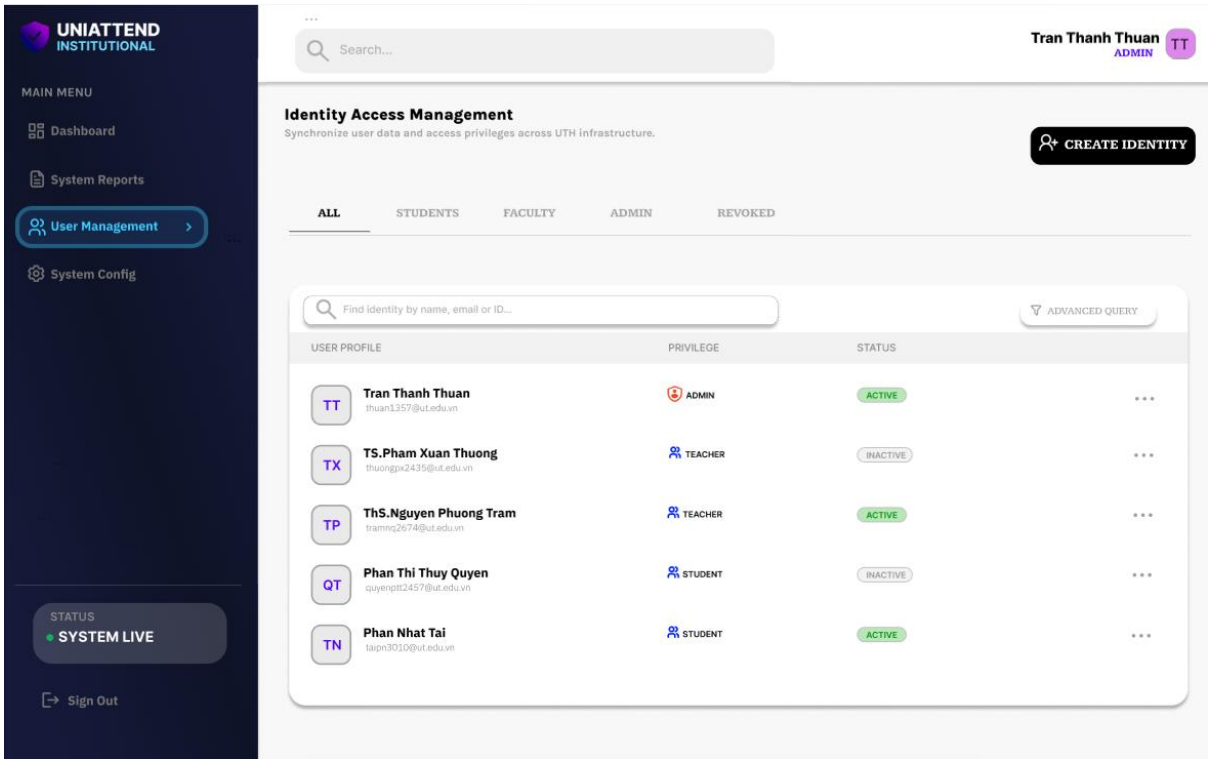
- View Dashboard:



- System Status:



- Manage User:



- Manage System: