

# Project X: Automated Attendance System

# Team BurDangBall

Christopher Burgos

Joomer Jake Caballero

Leopoldo Dumadangon IV

Cristian Villanueva

**R1. User Roles and Access Control**

R1.1. The system shall support three primary roles: Lecturer, Student, and Administrator.

R1.2. Only registered lecturers shall be able to record attendance.

R1.3. Only registered devices shall be authorized for attendance recording.

R1.4. Administrators shall have full control over system data (CRUD operations on students, lecturers, devices, and courses).

**R2. Attendance Recording and Data Persistence**

R2.1. The system shall allow lecturers to record student attendance using a registered mobile device, tablet, or computer.

R2.2. Attendance data shall be stored on a cloud-based MySQL database.

R2.3. Attendance shall be linked to a course, identifying the student, lecturer, date, and time.

R2.4. Attendance shall be retrievable in real-time.

**R3. Device Registration and Tracking**

R3.1. A lecturer shall be able to register multiple devices.

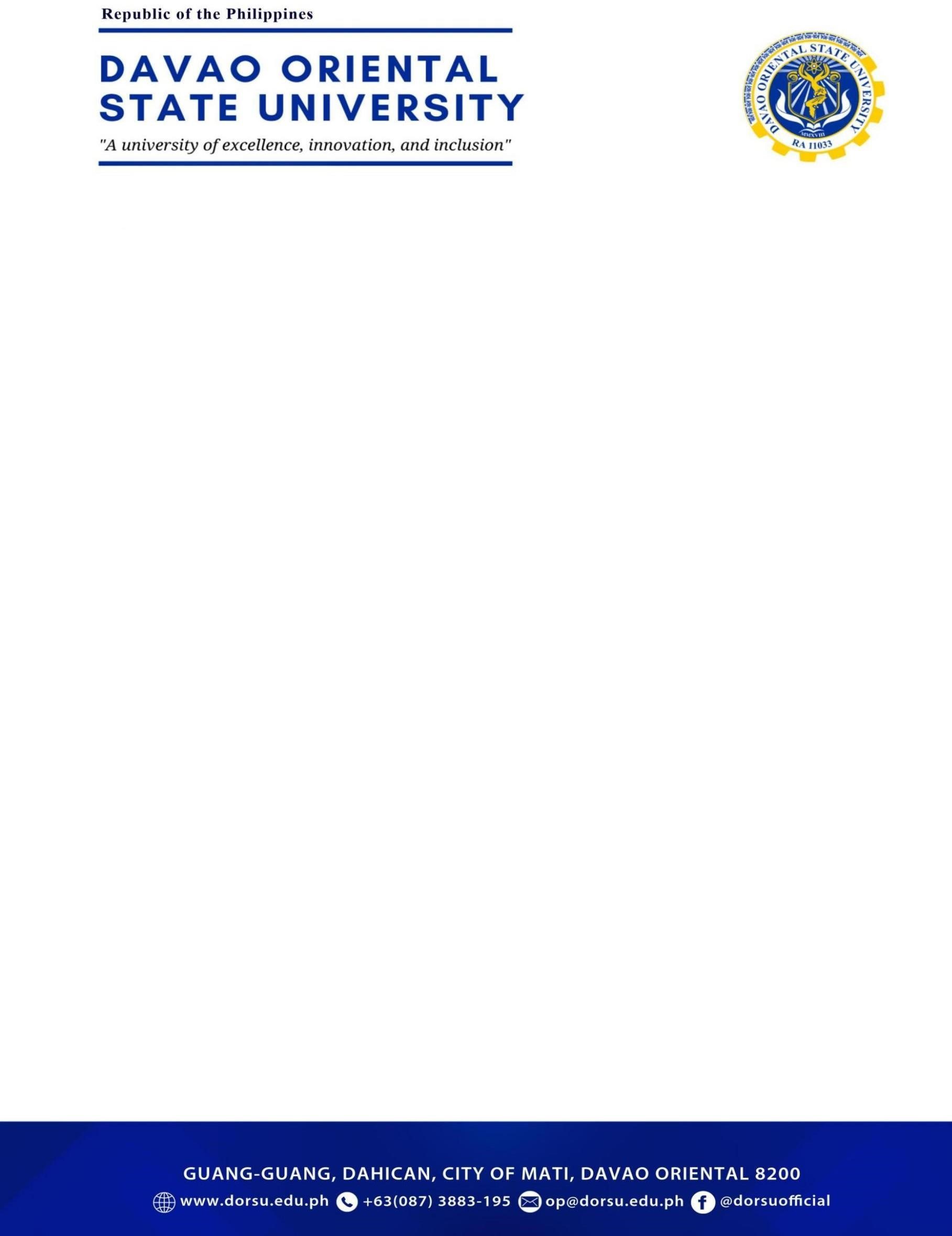
R3.2. The system shall track the location of registered devices in case a lecturer needs assistance.

R3.3. Only registered devices shall be able to access the attendance system.

**R4. Student Enrollment and Management**

R4.1. Students shall be able to enroll in courses through the system.

R4.2. The system shall store and manage student details, including:



R4.2.1. Name

R4.2.2. University ID

R4.2.3. Profile Picture (captured via the system)

R4.3. The system shall allow adding, updating, and deleting student records.

**R5. Reporting and Data Access**

R5.1. The system shall provide the following reports:

R5.1.1. Attendance reports (by student, by course, by date range).

R5.1.2. List of students per course.

R5.1.3. List of lecturers and their registered devices.

R5.2. Reports shall be accessible to lecturers and administrators.

**R6. Photo Capture and Storage**

R6.1. The system shall allow lecturers to capture passport-style photos of students.

R6.2. Photos shall be stored as files, and student IDs and names shall be stored in the database.

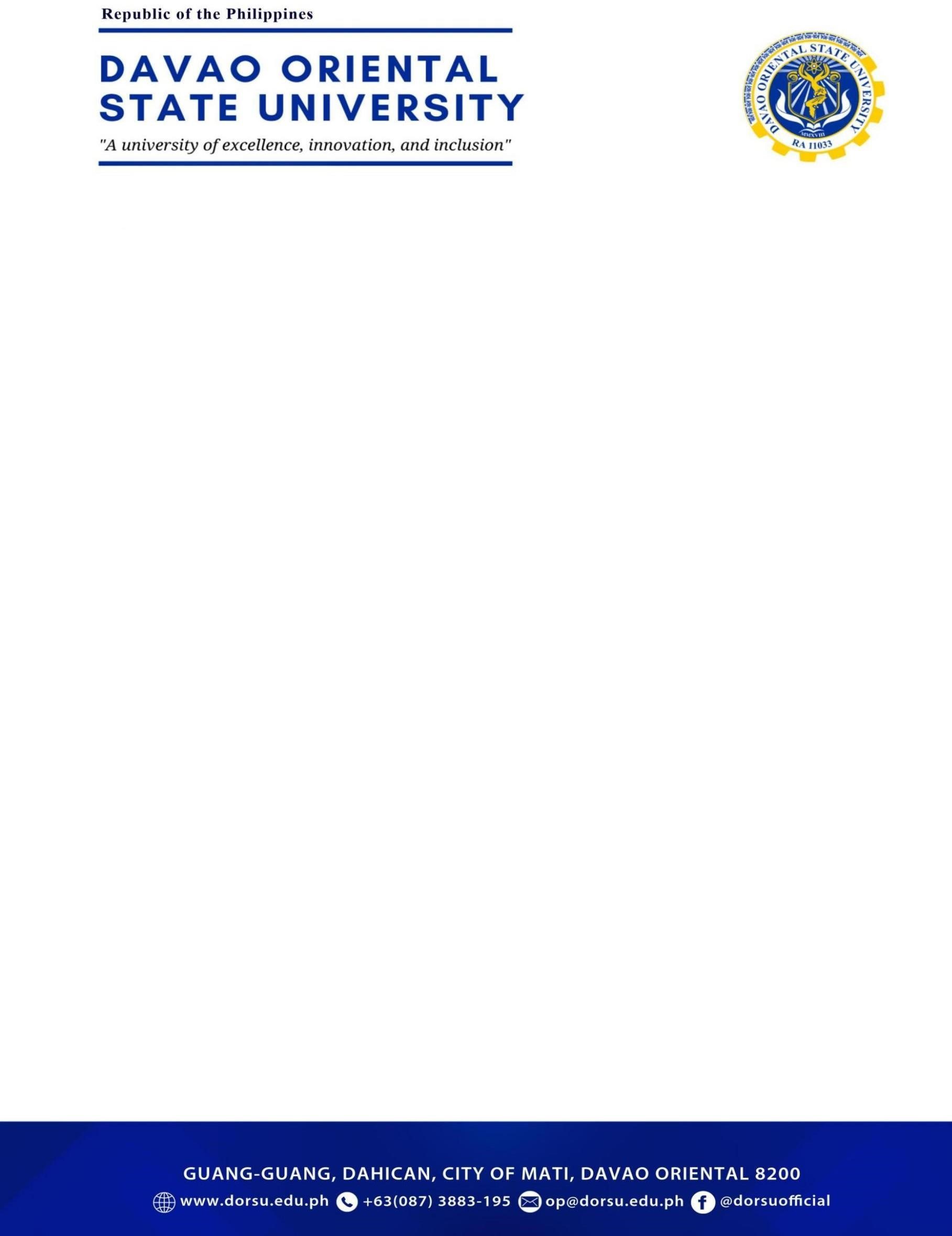
R6.3. The system shall ensure secure access to stored images.

**R7. System Architecture and API Access**

R7.1. The system shall use a REST API to interact with the database.

R7.2. All user interactions (attendance, enrollment, reporting) shall be handled through the API.

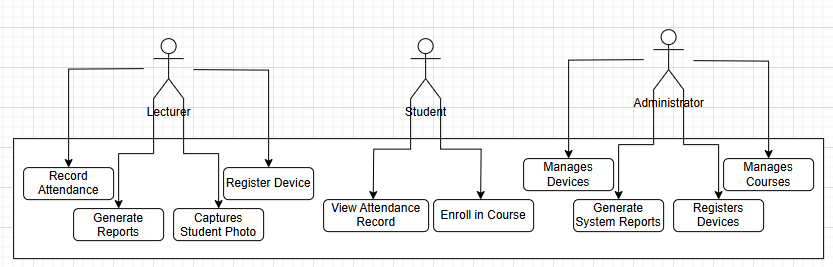
R7.3.API requests shall be authenticated and authorized.



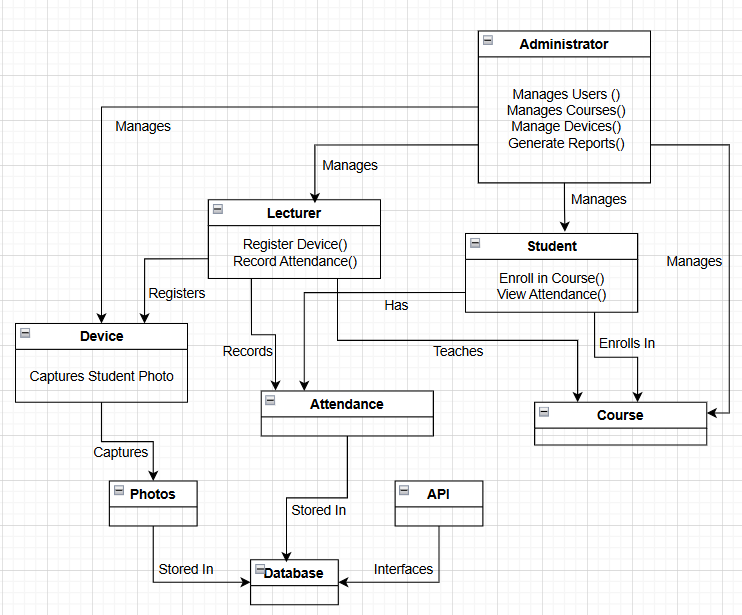
**8. Testing Criteria**

R8.1. Unit Testing: Each function (attendance logging, student enrollment, etc.) shall be tested in isolation.

R8.2. System Testing: The system shall be tested end-to-end to validate workflows.

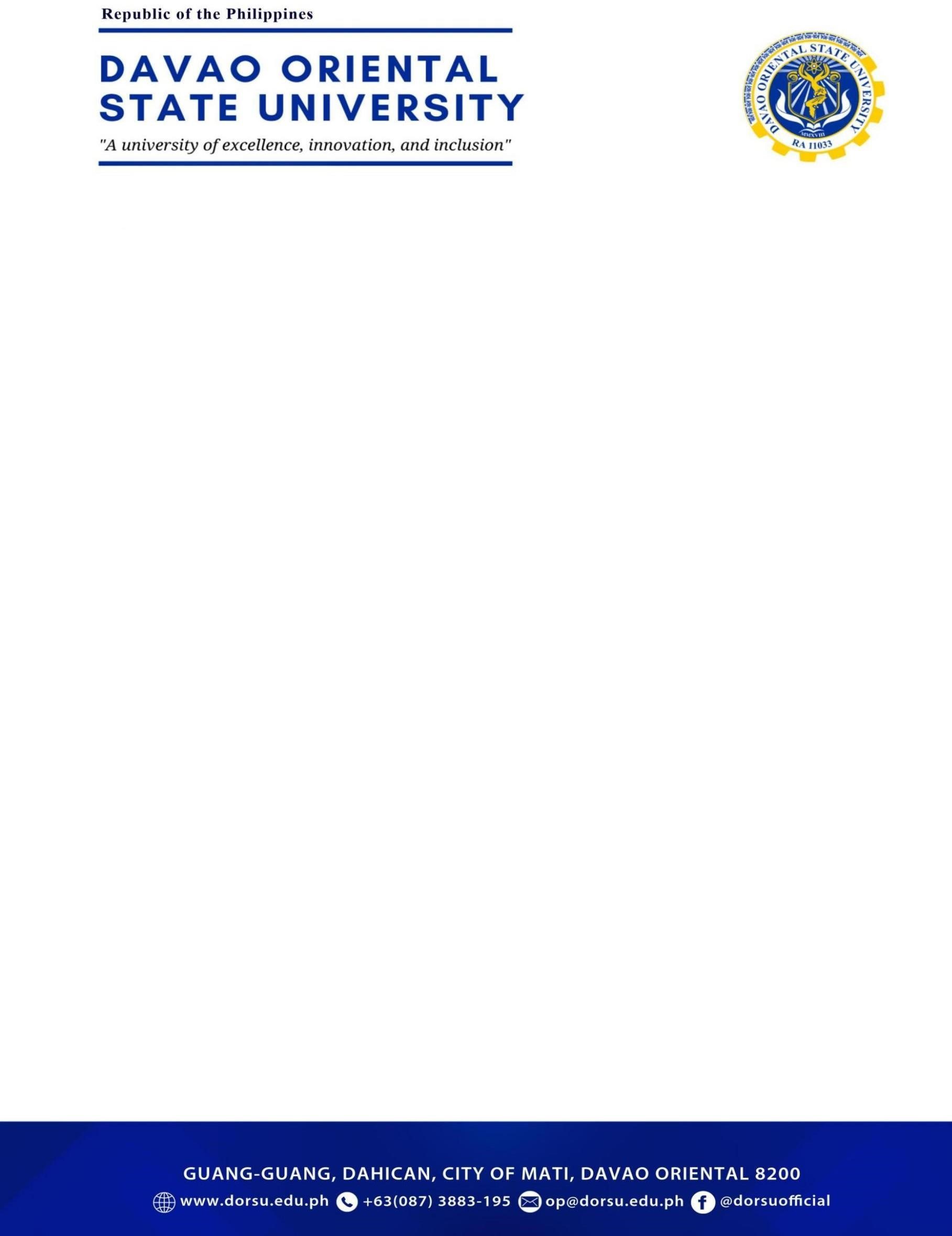
R8.3. User Acceptance Testing (UAT): The system shall be tested against the above assertions to confirm it meets user needs.

USE CASE DIAGRAM



**CONCEPTUAL UML CLASS DIAGRAM**

**Frontend (Client Side)**



* Web & Mobile App (React/Flutter) - Interface for lecturers, students, and administrators.
* Camera Module - Captures student photos for attendance records.

**Backend (Server Side)**

* REST API (Node.js/Express) - Handles requests from the frontend.
* Authentication Module - Manages user login/logout.
* Attendance Processing Module - Records student attendance.
* Report Generation Module - Generates attendance reports.
* Device Tracking Module - Locates registered devices.

**Database (Cloud-Based MySQL)**

* User & Role Management - Stores user details (students, lecturers, admins).
* Course & Enrollment Data - Tracks which students are enrolled in which courses.
* Attendance Records - Stores attendance data.
* Photo Storage - Stores student photos for verification.

**External Services (Optional Enhancements)**

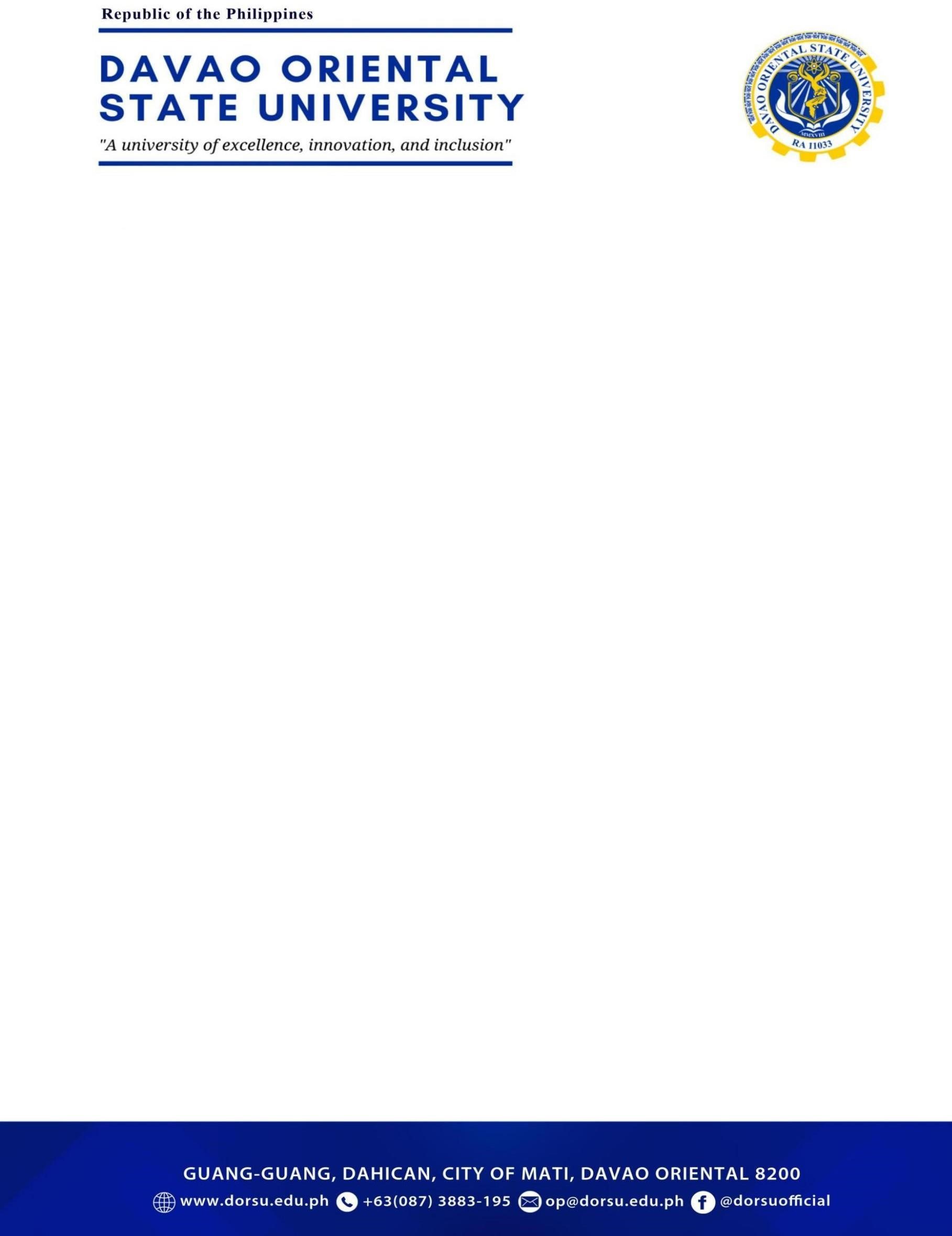
* Cloud Storage (AWS S3, Firebase Storage) - Stores student images.
* GPS Tracking API (Google Maps API) - Locates lecturer devices.
* Notification Service (Twilio, Firebase Cloud Messaging) - Sends reminders for attendance.

## Use Case Scenario: Automated Attendance System

**Use Case: Mark Student Attendance**

**Actors:**

* **Primary Actor:** Lecturer
* **Secondary Actors:** Student, Administrator **Preconditions:**
* The lecturer must be logged into the system.
* The class session must be scheduled in the system.



* Students must be registered in the system.

**Main Flow:**

1. The lecturer logs into the system.
2. The lecturer selects the course and class session.
3. The system displays the list of enrolled students.
4. The lecturer chooses the attendance marking method (e.g., QR code, biometric, manual).
5. Students mark their attendance using the selected method.
6. The system validates the student’s presence.
7. Attendance records are updated in real-time.
8. The lecturer submits the attendance record.
9. The system stores the attendance data in the database.
10. The system generates an attendance report.

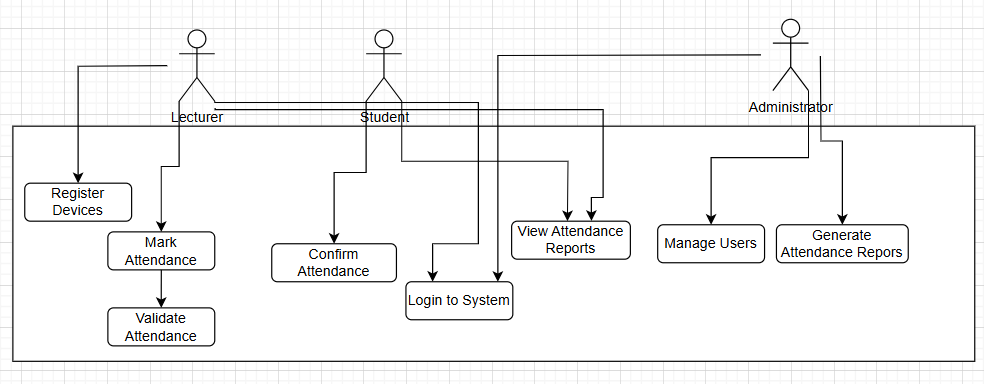
**Alternate Flows:**

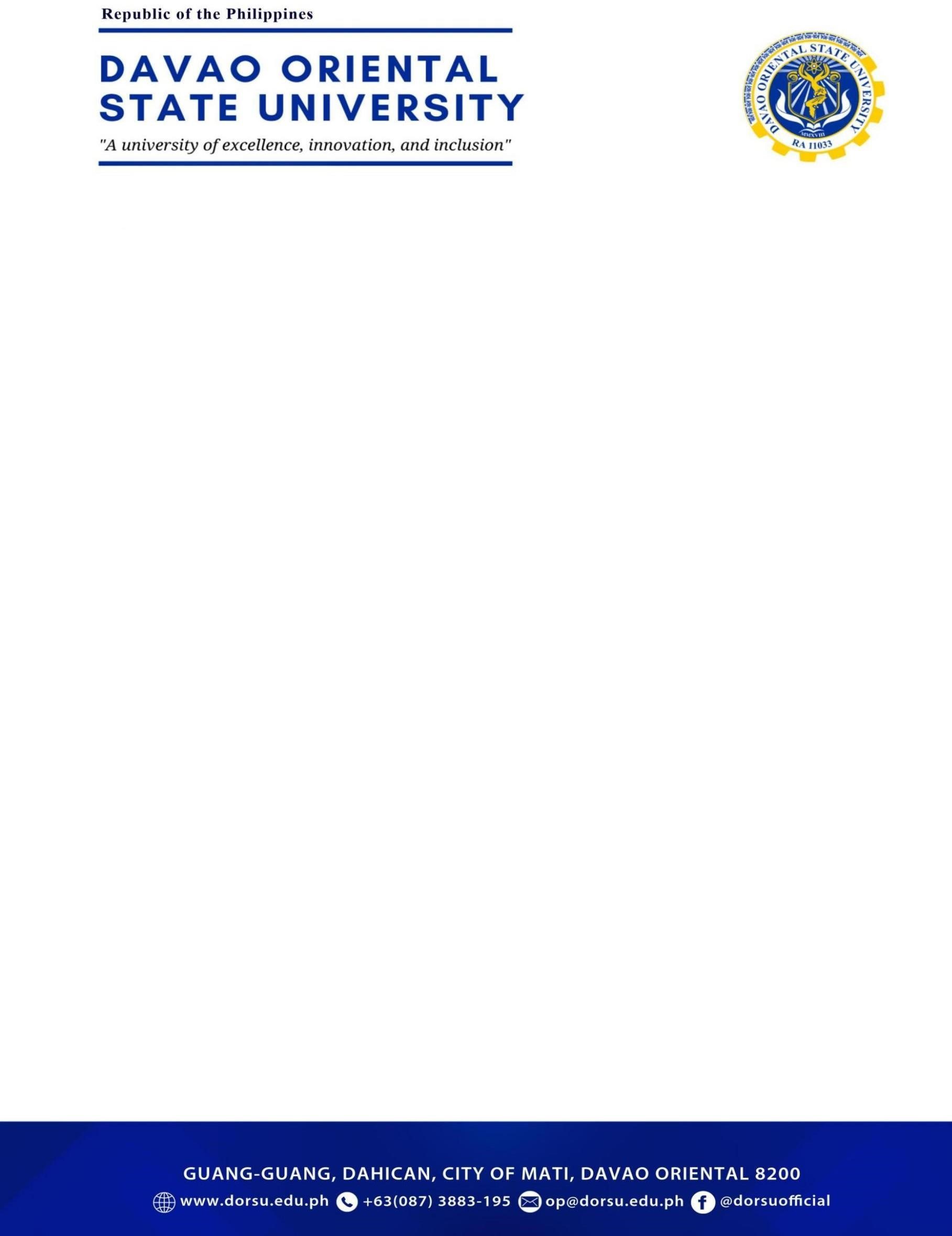
* **Invalid Student Attempt:** If a student who is not registered for the course tries to mark attendance, the system denies access and notifies the lecturer.
* **Missed Attendance:** If a student fails to mark attendance within the given timeframe, the system marks them absent.
* **Offline Mode:** If the internet is unavailable, the system stores attendance data locally and syncs once connected.

**Postconditions:**

* A diagram of a company

  AI-generated content may be incorrect.Attendance records are successfully stored in the database.
* Reports are accessible to lecturers and administrators.
* Students can view their attendance status

**Use Case Scenario: Generate Attendance Report**



**Actors:**

* **Primary Actor:** Administrator
* **Secondary Actor:** Lecturer

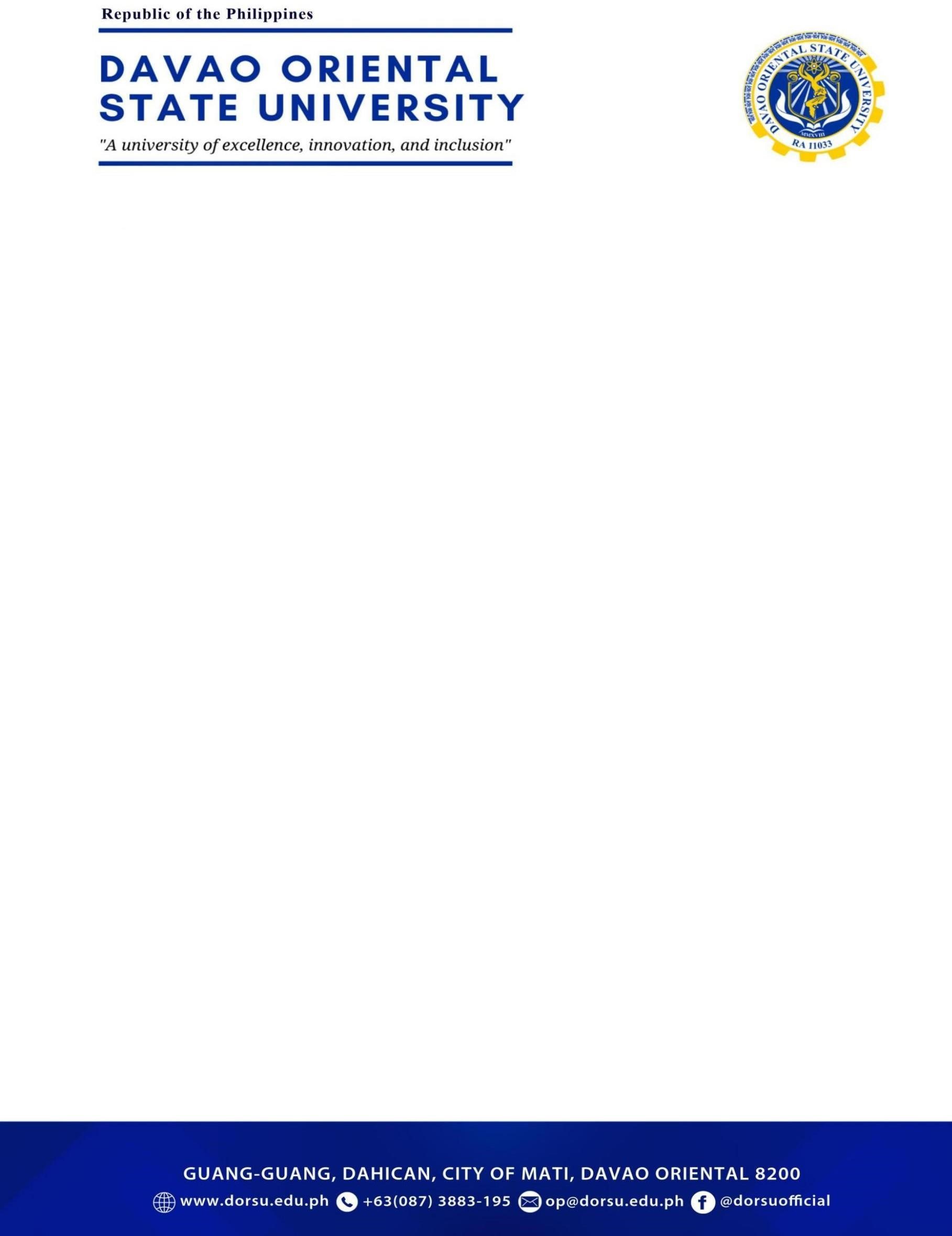
**Preconditions:**

* The administrator or lecturer must be logged into the system.
* Attendance records must be available in the system.

**Main Flow:**

1. The administrator logs into the system.
2. The administrator navigates to the **Reports** section.
3. The system provides filters (e.g., by course, date range, student, lecturer).
4. The administrator selects the desired filters.
5. The administrator requests to generate the report.
6. The system processes the request and retrieves relevant attendance data.
7. The system formats the report (e.g., table, PDF, CSV).
8. The system displays the generated report.
9. The administrator downloads or prints the report.

**Alternate Flows:**



* **Invalid Date Range:** If the selected date range has no attendance records, the system notifies the user and prompts them to adjust the filters.
* **Export Options:** The administrator may choose to export the report in different formats (e.g., CSV, PDF).
* **Access Restriction:** A lecturer can only generate reports for the classes they are assigned to, while administrators have full access.

**Postconditions:**

* The requested attendance report is generated and available for download.
* The administrator or lecturer can use the report for record-keeping or further analysis.

