

ACADEMY NEXUS

College Management System in Python Django



SUVARNA DHAVALÉ

MODERN COLLEGE OF ARTS, SCIENCE & COMMERCE
Pune - 411005

ACKNOWLEDGEMENT

It gives us a great pleasure to acknowledge the contribution of all the people who have directly or indirectly helped us in the development of our project for the MSC(CA) - II .

We would like to express a deep sense of gratitude to our faculty members for their continuous encouragement and guidance which shaped our efforts.

We would like to express our sincere regards to **Prof. Shamkant Deshmukh sir**, Vice Principal and Head, Dept of Computer Science at Modern College, Pune . We are great thankful to our project guide **Prof. Kalyani Salla** madam for guiding us and constant encouragement for this project.

We also thank all respected faculty members and staff of Modern College of Arts, Science and Commerce.

Suvarna Dhavale.

INDEX

SR NO.	Title	Page NO.
1	Introduction	4
	1.1 Overview	
	1.2 Purpose	
	1.3 Scope	
2	Problem Definition	5
	2.1 Challenges in College Management	
	2.2 Limitations of Existing Systems	
3	Existing System and Need for Computerization	6,7
	3.1 Overview of Existing System	6
	3.2 Justification for Computerization	7
4	Scope and Objectives of the Proposed System	8,9
	4.1 Scope of the Proposed System	8
	4.2 Objectives	9
5	Requirement Gathering and Platform Details	10,11
	5.1 Requirement Gathering and Anticipation	10
	5.1 Platform (H/W, S/W) with Version Details	11
6	Analysis and Design Specifications (Object-Oriented Approach)	12-22
	6.1 Entity-Relationship Diagrams	12
	6.2 Use Case Diagrams	13
	6.3 Class Diagrams	14
	6.4 Sequence Diagrams	15,16,17
	6.5 System Architecture	18
	6.6 Activity Diagram	19,20,21
	6.7 Database Schema	22
7	Input/Output Screens	23-26
	7.1 Admin Panel Screens	23
	7.2 Staff Panel Screens	24
	7.3 Student Panel Screens	25,26

8	Testing Strategies and Decision Tools	27,28
	8.1 Unit Testing	27
	8.2 Integration Testing	28
	8.3 User Acceptance Testing	28
9	Limitations, Conclusion, and Future Enhancements	29
	9.1 Limitations and Drawbacks	
	9.2 Conclusion and Future Enhancements	
10	User Manual, and References & Bibliography	30,31
	10.1 User Manual (Installation Guide, User Guide)	30
	10.2 References & Bibliography	31

1. Introduction

1.1 Overview:

The College Management System project is an innovative endeavor aimed at revolutionizing the administrative processes within educational institutions. Developed using Python with the Django Web Framework, this system offers a comprehensive suite of features tailored to meet the diverse needs of colleges and universities. By leveraging modern technology and best practices in software development, the system provides an intuitive and efficient platform for managing various administrative tasks.

1.2 Purpose:

The primary purpose of the College Management System is to streamline and automate the cumbersome administrative processes prevalent in educational institutions. By digitizing tasks such as student enrollment, attendance tracking, and result management, the system aims to reduce manual effort, minimize errors, and enhance overall efficiency. Additionally, the system facilitates improved communication and collaboration among students, staff, and administrators, fostering a more cohesive and productive learning environment.

1.3 Scope:

The scope of the College Management System encompasses a wide range of functionalities and features designed to address the key challenges faced in college management. From student record management to course scheduling and examination management, the system offers comprehensive solutions tailored to meet the specific requirements of educational institutions. Furthermore, the system's scalability and flexibility ensure adaptability to varying institutional needs and future enhancements.

2. Problem Definition

2.1 Challenges in College Management:

Educational institutions face numerous challenges in managing administrative tasks efficiently. Manual processes such as paper-based record-keeping and manual attendance tracking are time-consuming and prone to errors. Additionally, communication bottlenecks and data fragmentation hinder effective collaboration among stakeholders. These challenges result in inefficiencies and difficulties in maintaining accurate and up-to-date information, highlighting the need for a more streamlined and automated approach to college management.



2.2 Limitations of Existing Systems:

Existing systems for college management often lack the robust features and scalability required to meet the evolving needs of modern educational environments. Traditional methods and basic software solutions suffer from limitations such as data fragmentation, lack of integration, and usability issues. These shortcomings hinder institutions from harnessing the full potential of technology to improve administrative processes and enhance overall effectiveness.

3. Existing System and Need for Computerization

3.1 Overview of Existing System

The existing system for managing college operations relies heavily on manual and paper-based processes, which lead to several inefficiencies. Records for students, staff, courses, and attendance are maintained in physical registers, making the system prone to errors and the misplacement of important documents. Administrative tasks such as student registration, attendance tracking, leave management, and result processing are time-consuming due to their manual nature. Accessing specific information is cumbersome and slow, often requiring a physical presence at the administrative office. Communication relies on notice boards and verbal announcements, resulting in missed information and delayed updates. Additionally, data is scattered across various departments, making it difficult to obtain a consolidated view of student and staff data. The security of physical records is also a concern, as they are vulnerable to loss, theft, or damage.

3.2 Justification for Computerization

Computerizing the college management system is crucial to address these issues effectively. Automation of processes significantly reduces the time and effort required for administrative tasks, allowing staff to focus on more critical activities. A digital system enables centralized storage of all college-related information, ensuring easy access and retrieval, and facilitating better coordination among different departments. Automation also minimizes human errors, ensuring higher accuracy in managing college operations. Digital platforms enhance communication through instant notifications and alerts, ensuring that students and staff receive timely updates. Computerized systems offer robust security features, protecting sensitive information from unauthorized access and loss. Stakeholders can access real-time information from anywhere, enhancing decision-making and responsiveness. Advanced reporting and data visualization tools provide valuable insights into attendance patterns, academic performance, and resource utilization, aiding in strategic planning and management. Finally, a computerized system can easily scale to accommodate the growing needs of the college, including increasing student enrollments and expanding course offerings.

4. Scope and Objectives of the Proposed System

4.1 Scope of the Proposed System

The proposed College Management System aims to streamline and automate the various administrative and academic processes within the institution. The system encompasses a wide range of functionalities to manage student, staff, and course information efficiently. It provides a centralized platform for managing student registrations, attendance tracking, leave applications, and result processing. The system also facilitates efficient communication through notifications and feedback mechanisms. Key features include:

- **Student Management:** Registration, profile updates, attendance tracking, leave management, and viewing notifications.
- **Staff Management:** Registration, profile updates, attendance tracking, result management, leave management, and communication with students and admin.
- **Admin Management:** Full control over the system, including course, subject, session management, staff and student record management, notification sending, and overseeing attendance and leave records.
- **Data Visualization:** Graphical representations of attendance, results, and other key metrics for easy analysis and reporting.
- **Security and Accessibility:** Secure login and data protection measures with real-time access for authorized users.

4.2 Objectives

The primary objectives of the proposed College Management System are:

1. **Efficiency Improvement:** Automate manual processes to save time and reduce administrative workload.
2. **Centralized Data Management:** Maintain all records in a centralized database for easy access and retrieval.
3. **Enhanced Accuracy:** Reduce human errors in record-keeping and data processing.
4. **Effective Communication:** Provide a platform for timely notifications and feedback between students, staff, and administration.
5. **Data Security:** Implement robust security measures to protect sensitive information.
6. **Real-Time Information Access:** Enable stakeholders to access up-to-date information from any location.
7. **Comprehensive Reporting:** Offer advanced reporting tools for better decision-making and strategic planning.
8. **Scalability:** Ensure the system can accommodate future growth and expansion of the college.
9. **User-Friendly Interface:** Design an intuitive and easy-to-use interface for all user types.
10. **Improved Resource Management:** Optimize the allocation and use of resources such as classrooms, equipment, and staff.

5. Requirement Gathering and Platform Details

5.1 Requirement Gathering and Anticipation

The requirement gathering process involves understanding the needs and expectations of the stakeholders, including students, staff, and administrators. This is achieved through surveys, interviews, and reviewing existing systems. Key requirements identified are:

- **User Authentication and Authorization:** Secure login system for students, staff, and admins with appropriate access levels.
- **Student Management:** Features for registration, profile updates, attendance tracking, leave applications, and viewing results.
- **Staff Management:** Capabilities for managing student results, taking attendance, applying for leaves, and sending notifications.
- **Admin Management:** Comprehensive control over course, subject, session management, and the ability to manage student and staff records.
- **Notification System:** A robust mechanism for sending and receiving notifications between students, staff, and admin.
- **Data Visualization:** Tools for graphical representation of attendance and performance data.
- **Feedback Mechanism:** A system for collecting and responding to feedback from students and staff.
- **Scalability and Flexibility:** The ability to handle increasing amounts of data and user requests as the college grows.

5.2 Platform (H/W, S/W) with Version Details

The platform for the College Management System consists of both hardware and software components. The following details outline the necessary hardware and software along with their versions:

Hardware Requirements:

- **Server:** A reliable server with the following minimum specifications:
 - Processor: Intel Core i5 or equivalent
 - RAM: 8 GB or higher
 - Storage: 500 GB SSD
 - Network: High-speed internet connection
- **Client Machines:** Computers for users (students, staff, admin) with the following specifications:
 - Processor: Intel Core i3 or equivalent
 - RAM: 4 GB or higher
 - Storage: 250 GB HDD/SSD
 - Display: 1024x768 resolution or higher

Software Requirements:

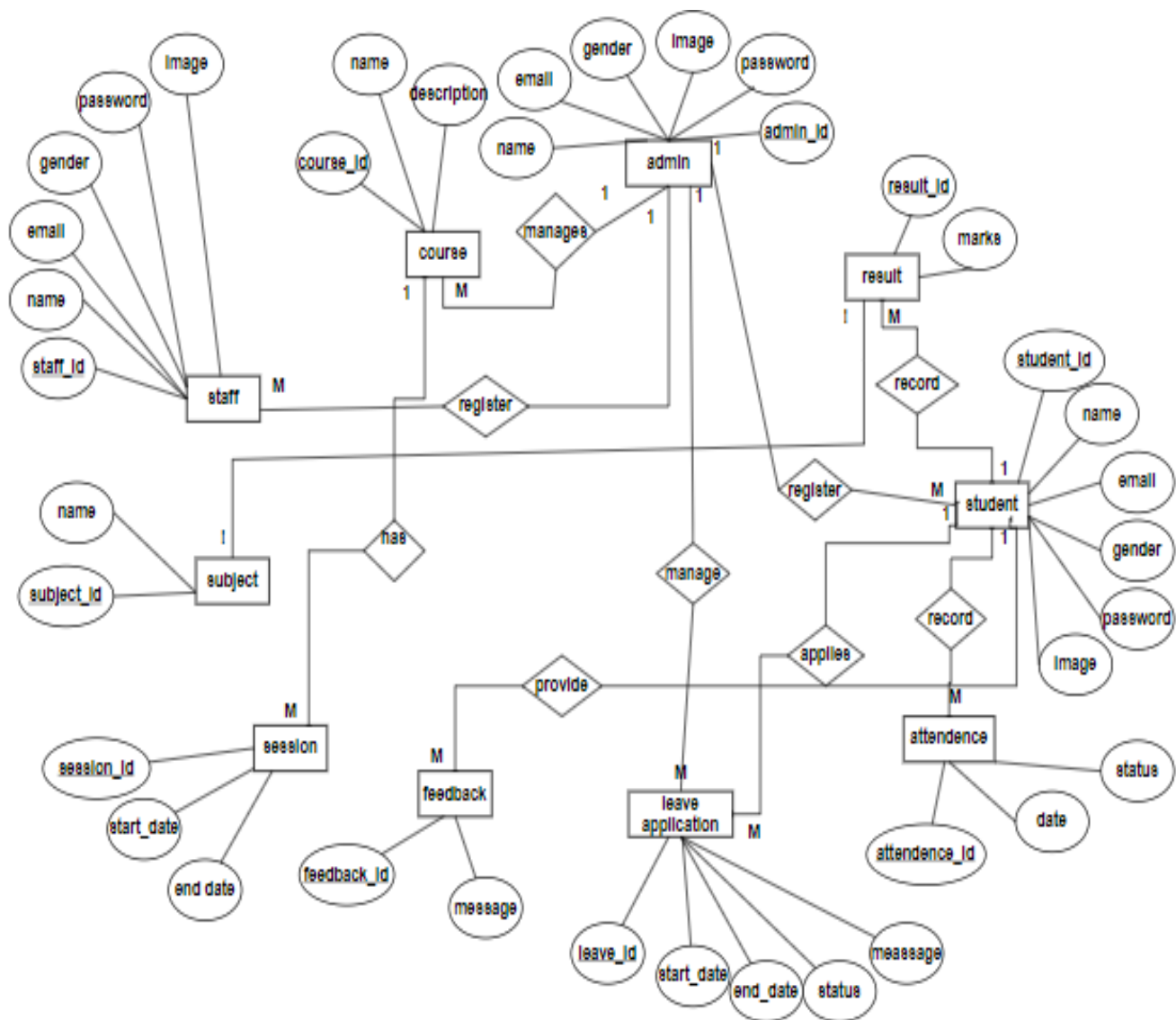
- **Operating System:**
 - Server: Ubuntu 20.04 LTS or any compatible Linux distribution
 - Client: Windows 10 or higher, macOS Catalina or higher, or Linux
- **Web Server:** Apache 2.4 or Nginx 1.18
- **Database:** SQLite 3.32 or higher
- **Programming Language:** Python 3.8 or higher
- **Web Framework:** Django 3.1 or higher
- **Frontend Technologies:**
 - HTML5, CSS3 (Bootstrap 4)
 - JavaScript (Vanilla JS or frameworks like React/Vue.js if needed)
- **Version Control:** Git 2.28 or higher
- **IDE/Code Editor:** VS Code, PyCharm, or any preferred IDE for Python development
- **Browser Compatibility:** Latest versions of Chrome, Firefox, Safari, and Edge

This platform ensures the system is robust, scalable, and capable of meeting the needs of all stakeholders effectively.

6. Analysis and Design Specifications

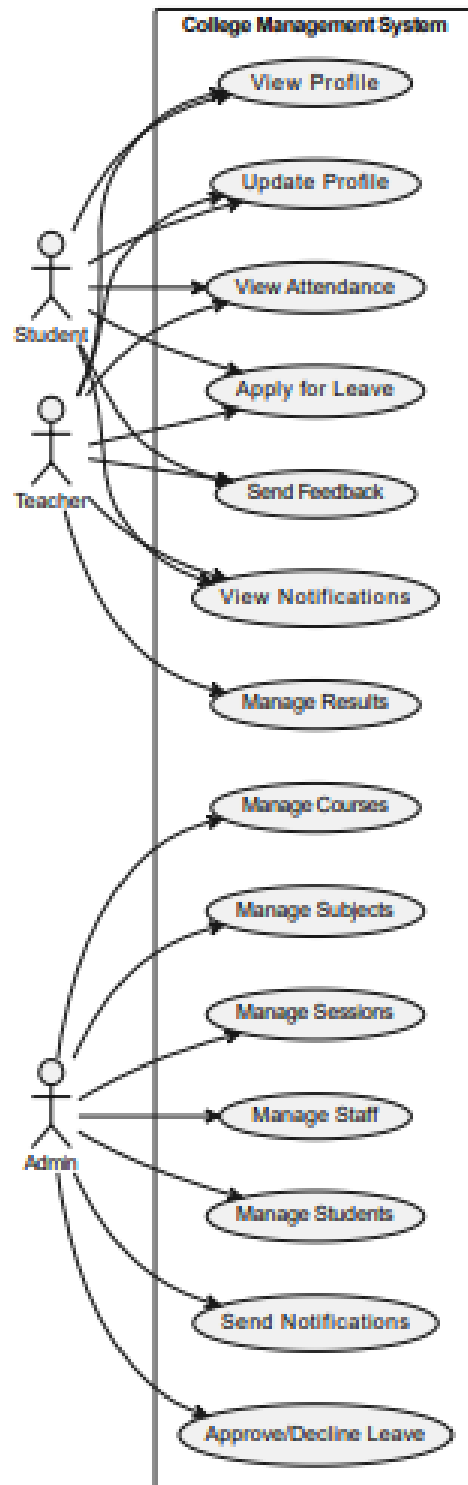
(Object-Oriented Approach)

6.1 Entity-Relationship Diagrams

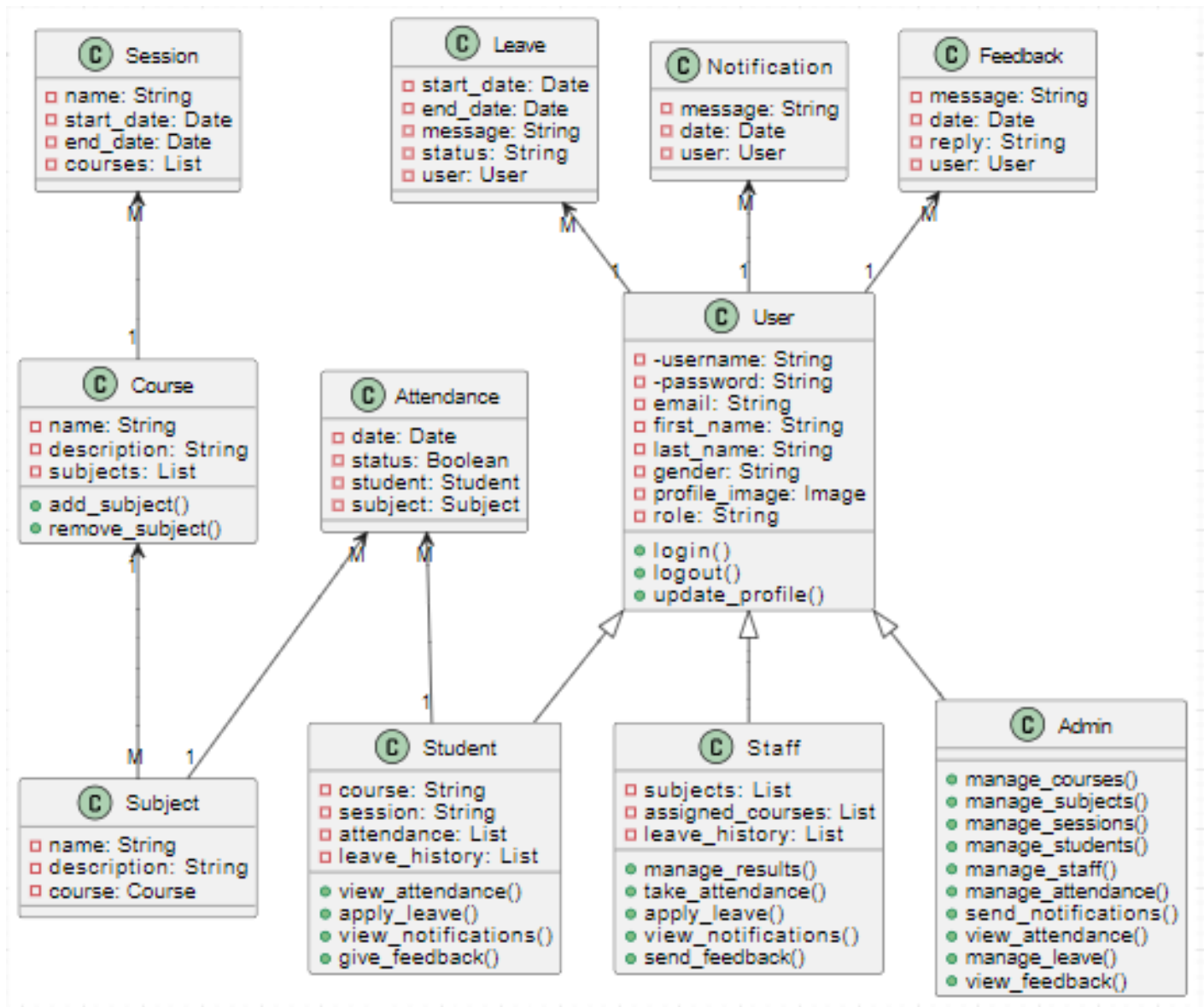


6.2 Use Case Diagrams

USE CASE DIAGRAM

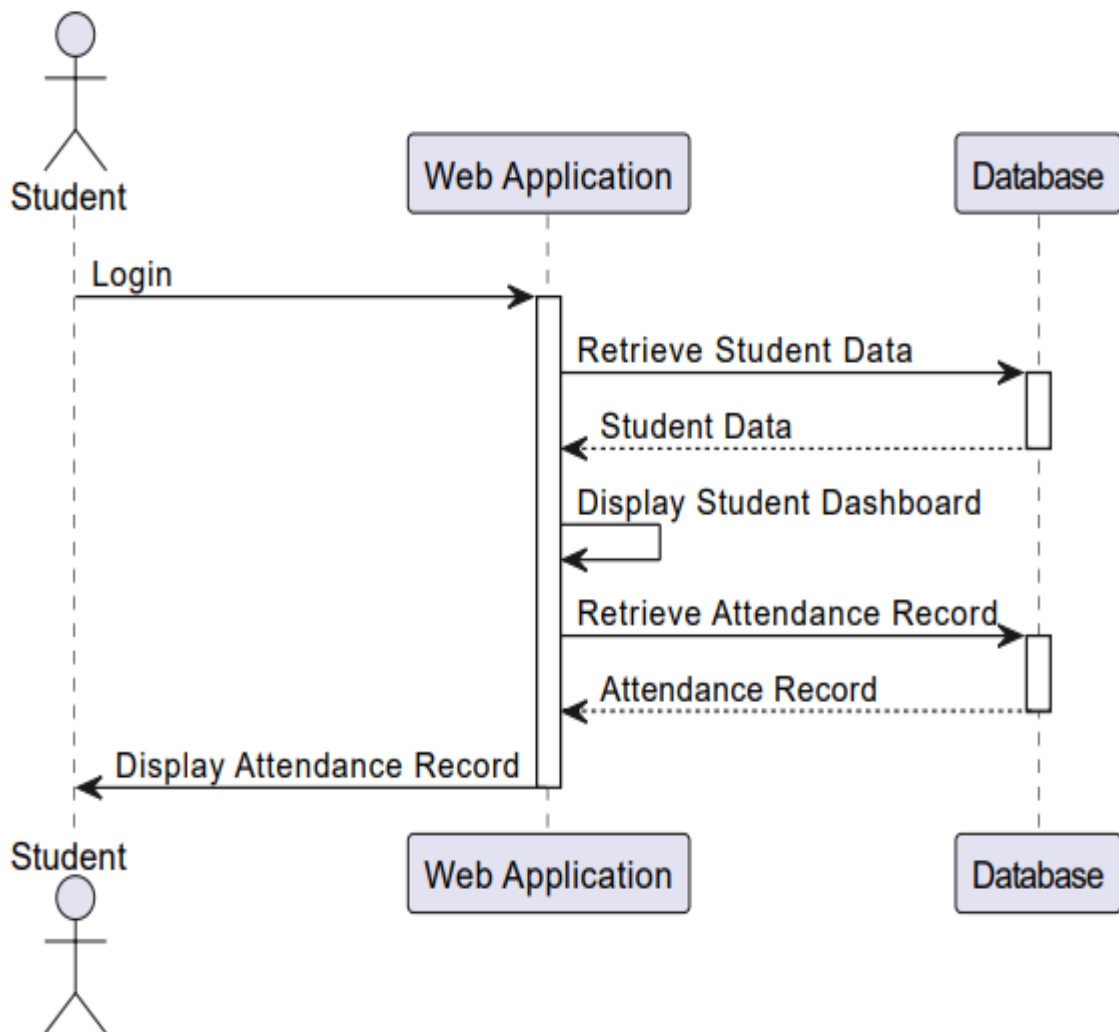


6.3 Class Diagrams

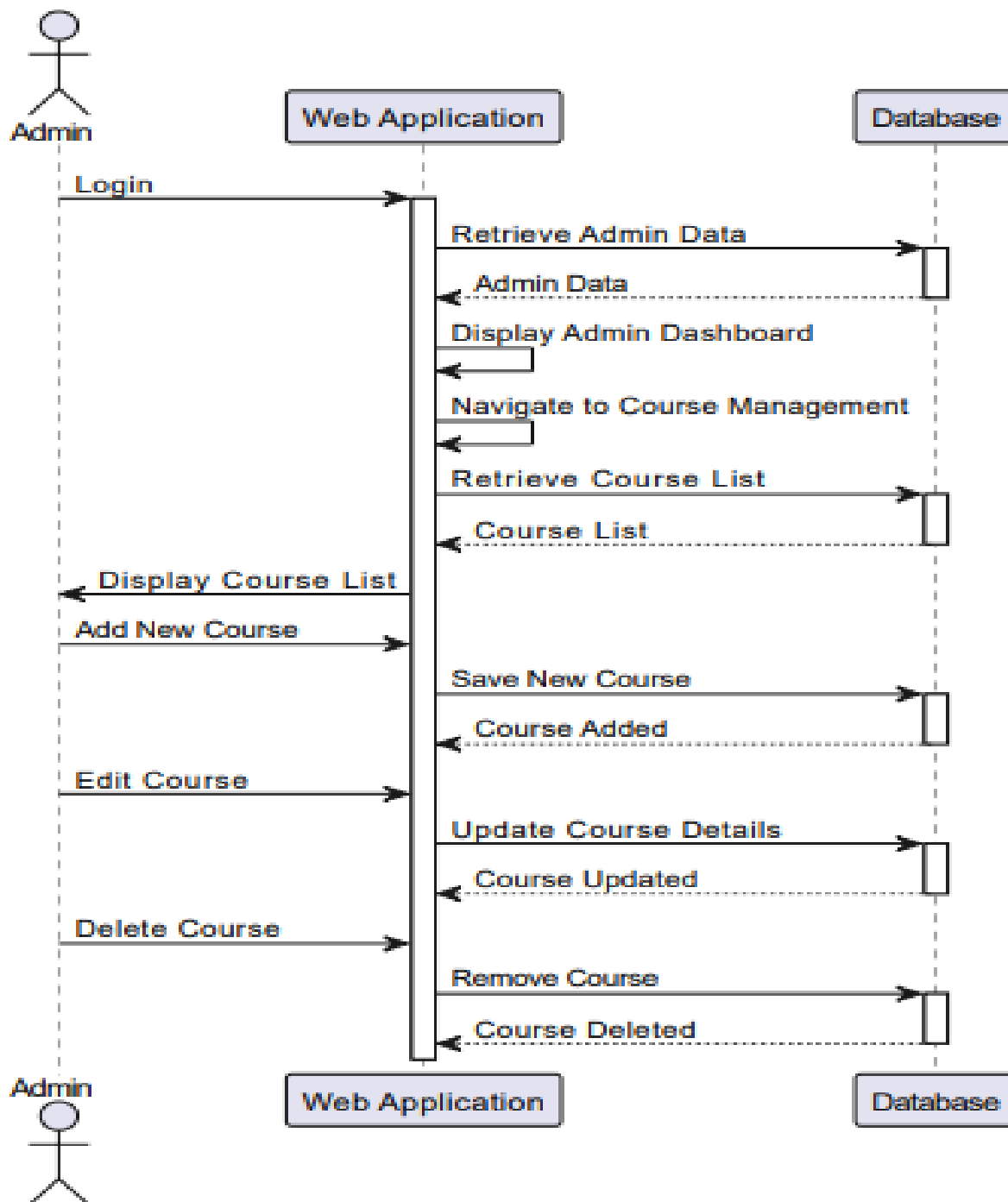


6.4 Sequence Diagrams

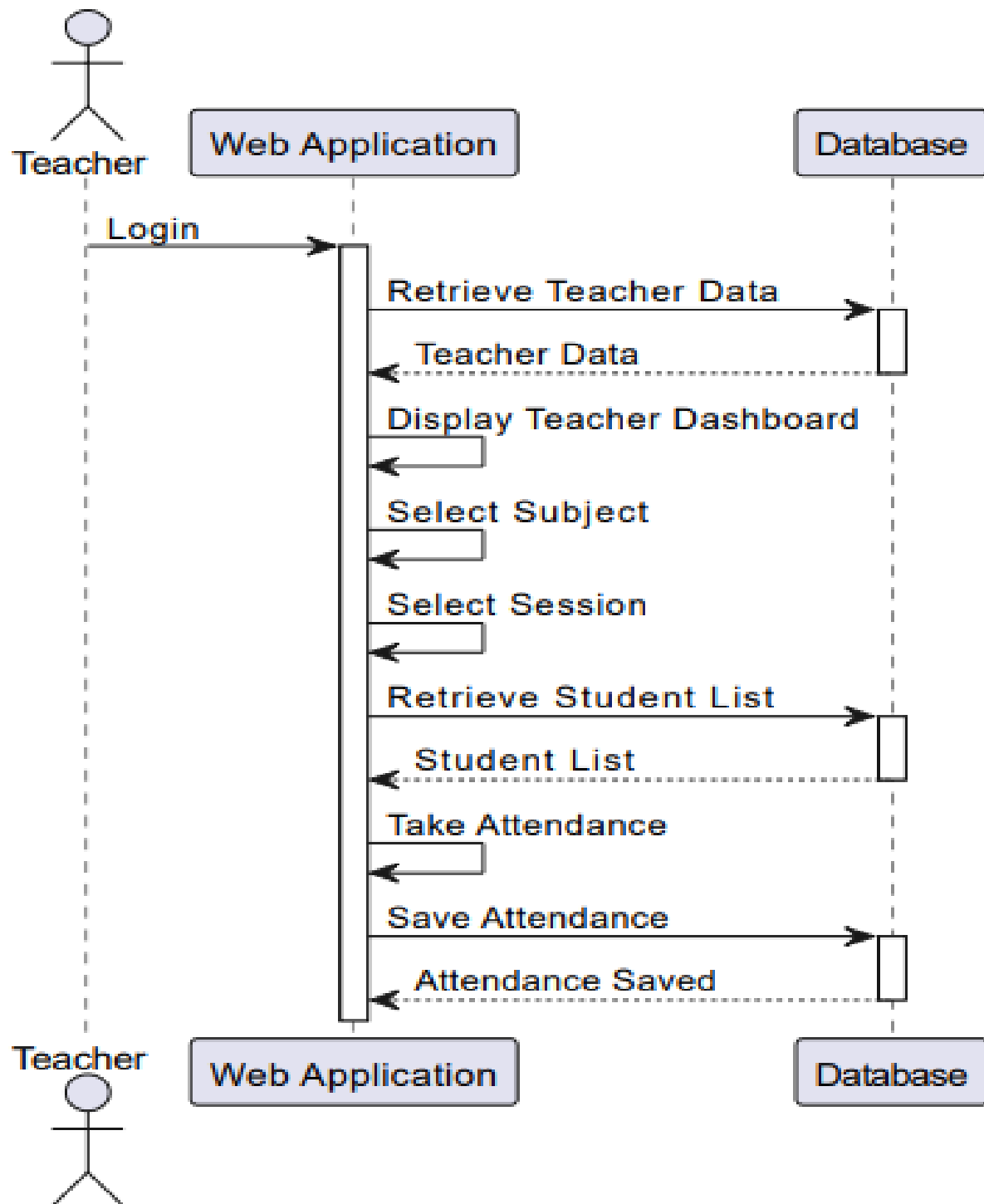
Student



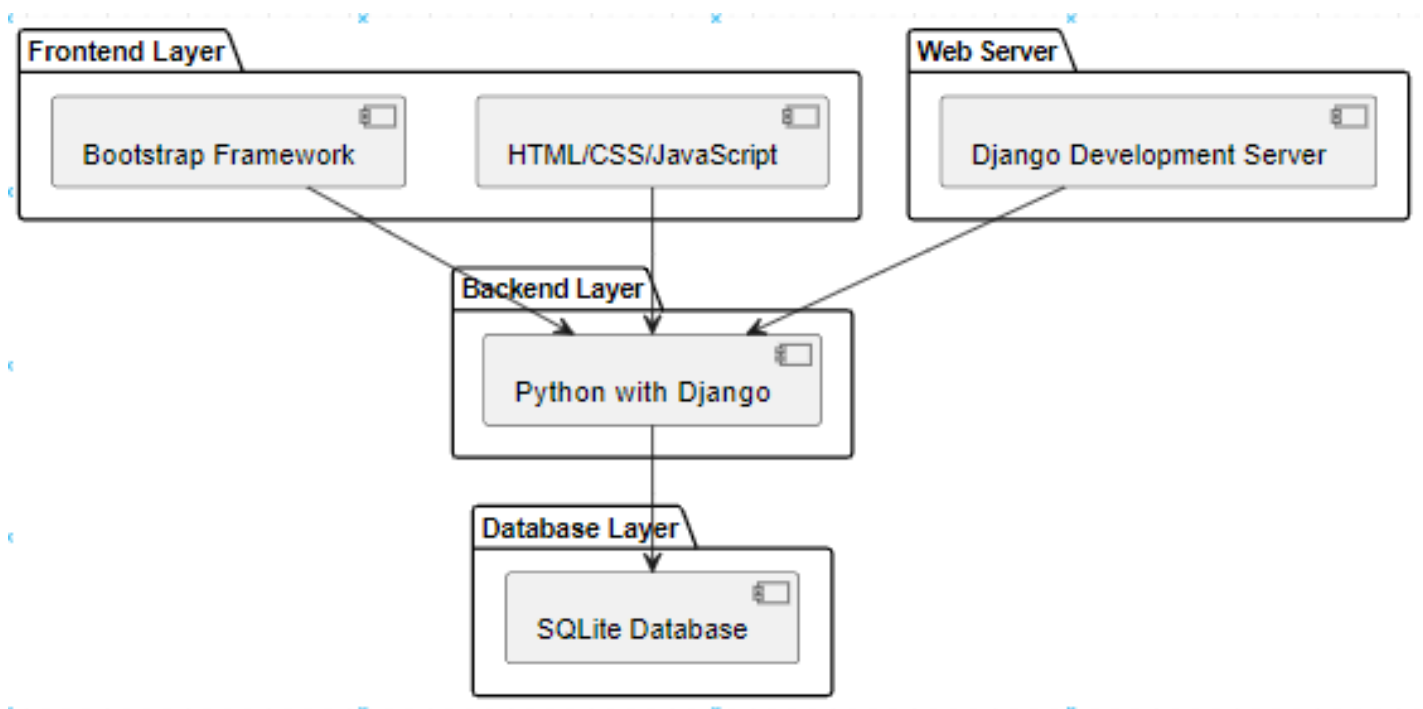
Admin



Teacher

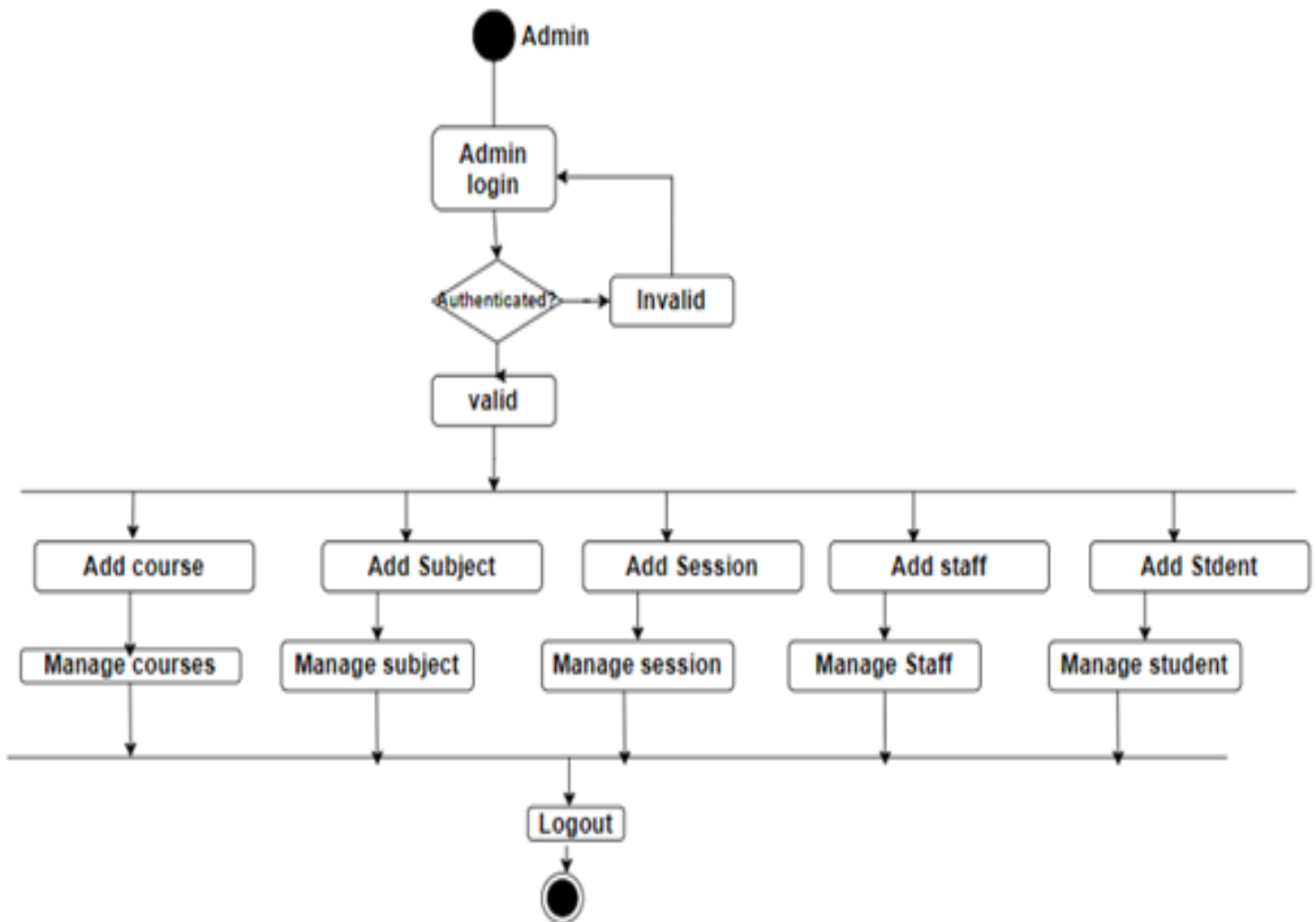


6.5 System Architecture

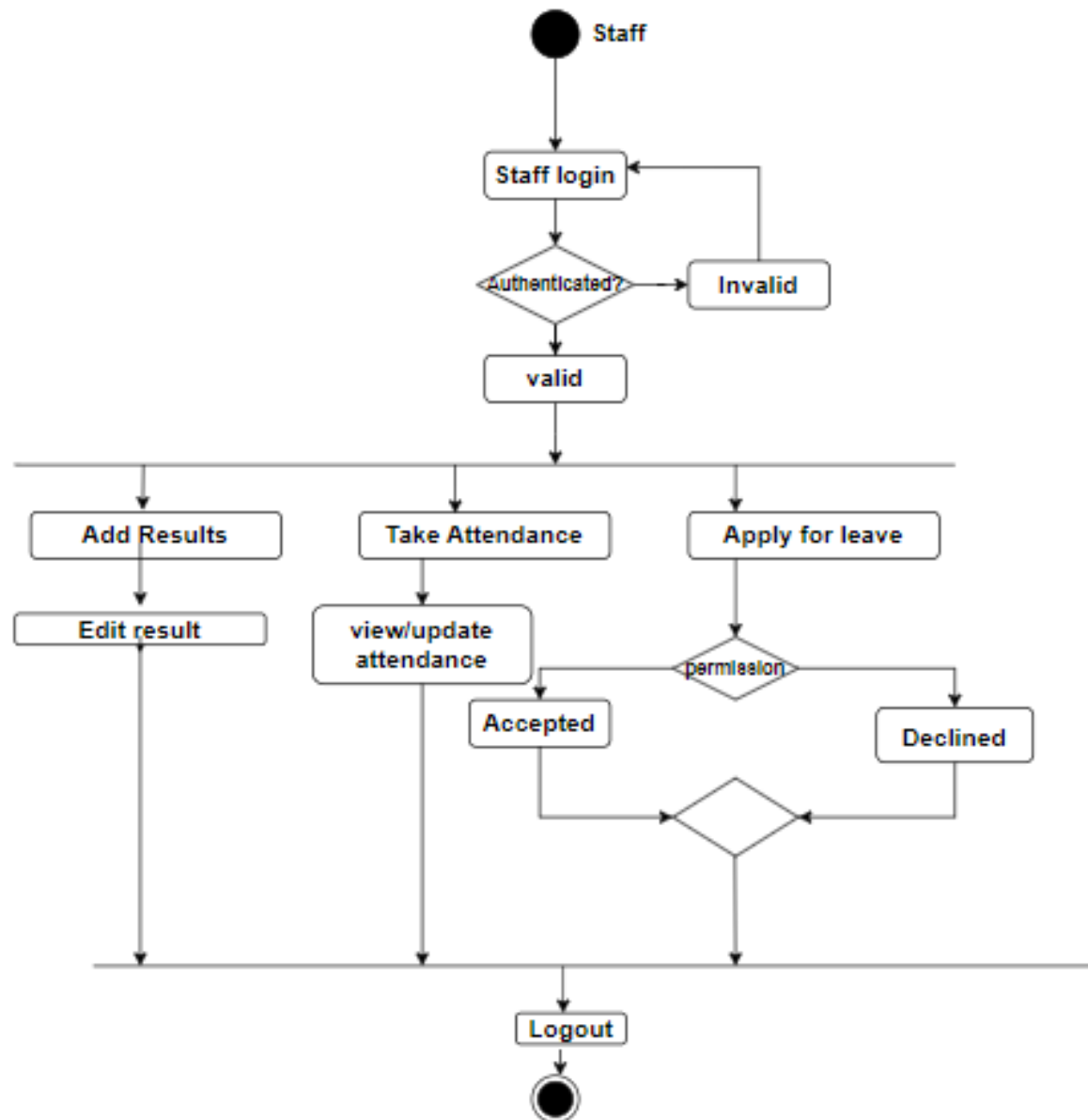


6.6 Activity Diagram

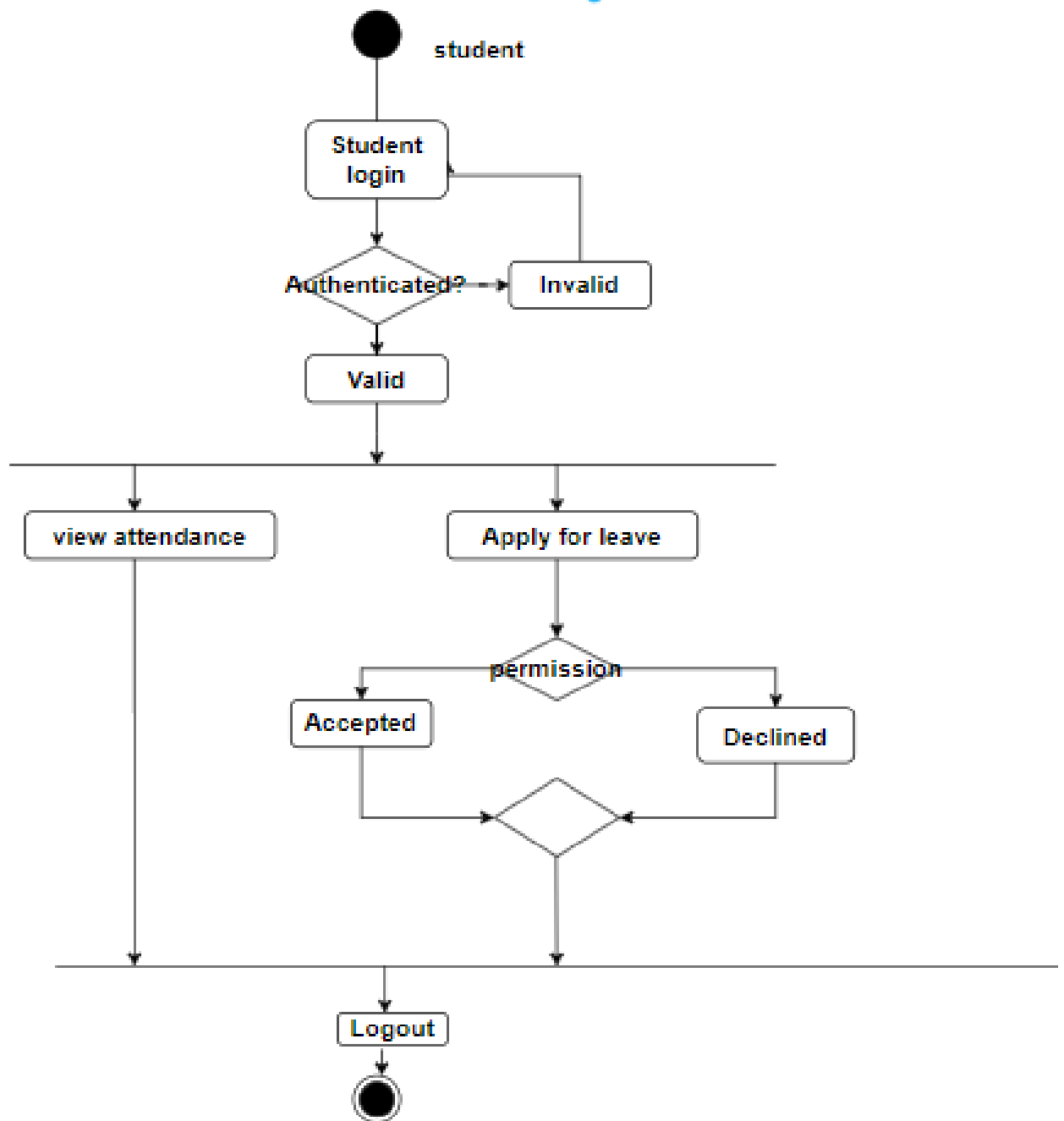
Admin



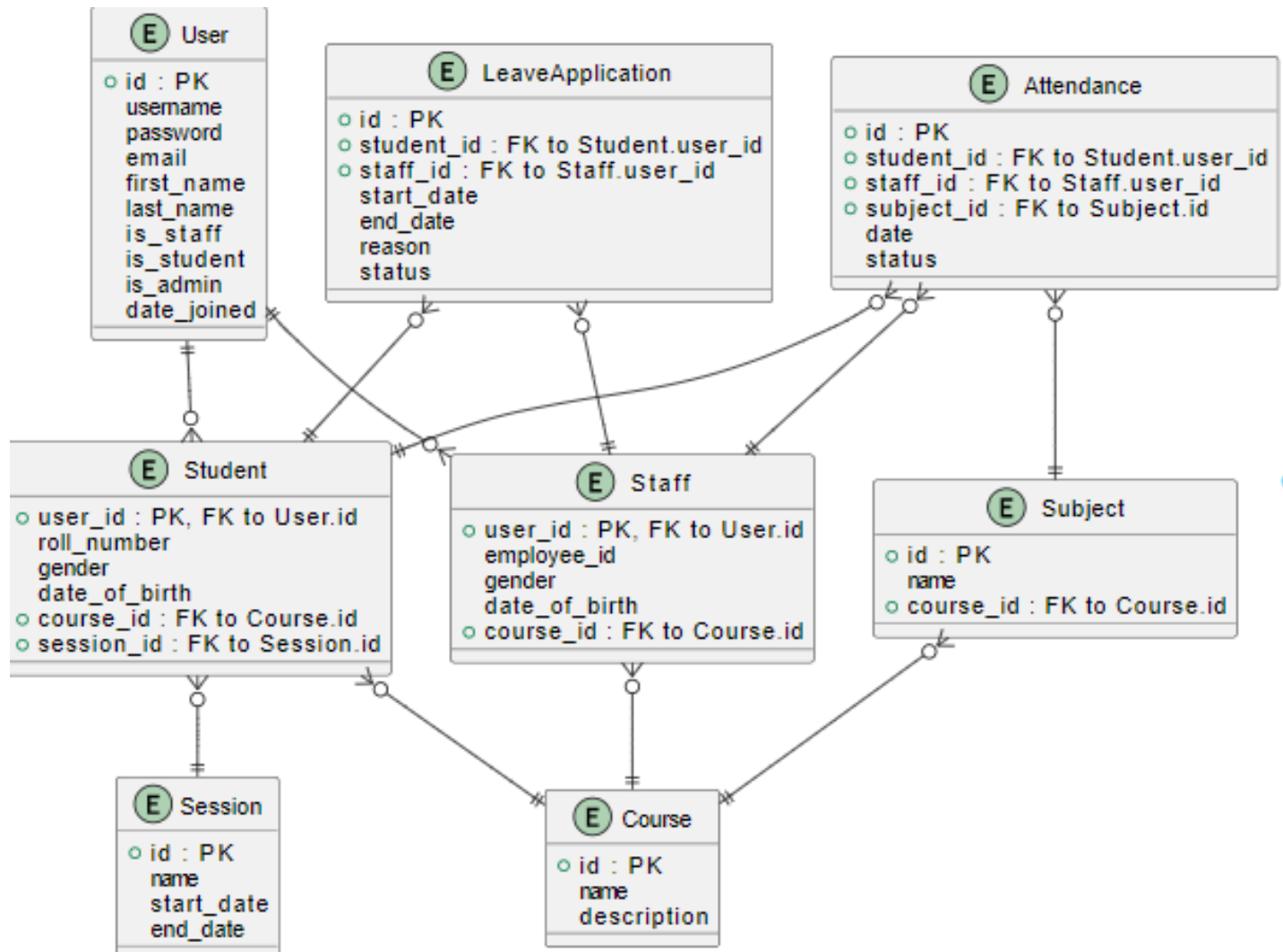
Staff



Student

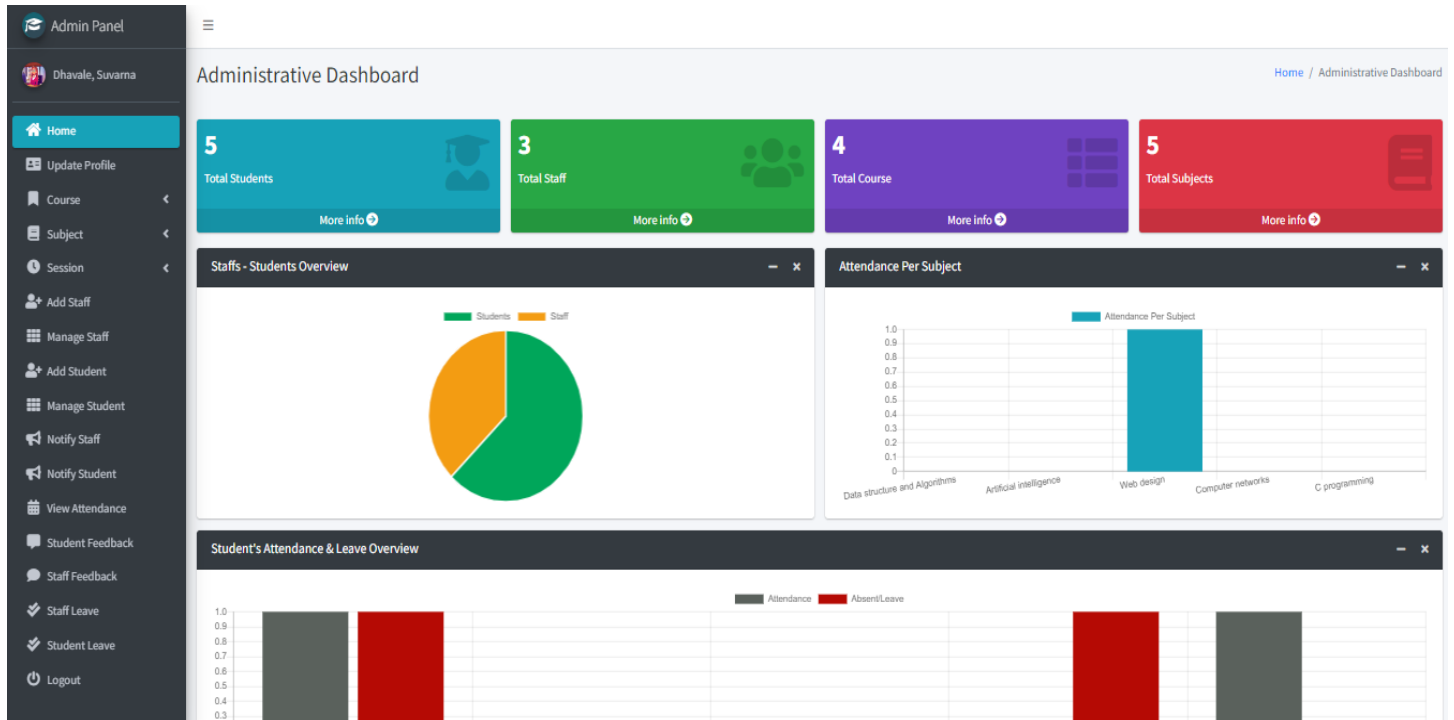


6.7 Database Schema



7. Input/Output Screens

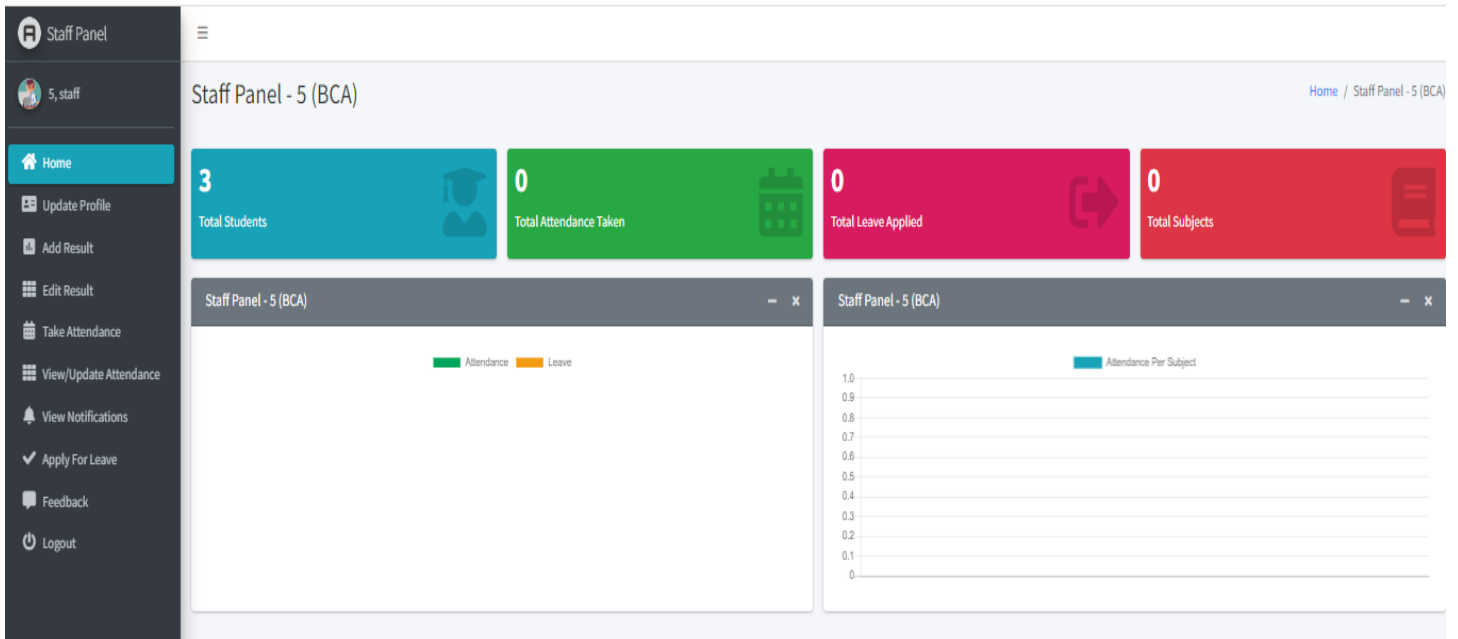
7.1 Admin Panel Screens



The Manage Students screen displays a table of student records. The table includes columns for ID, Full Name, Email, Gender, Course, Avatar, and Actions.

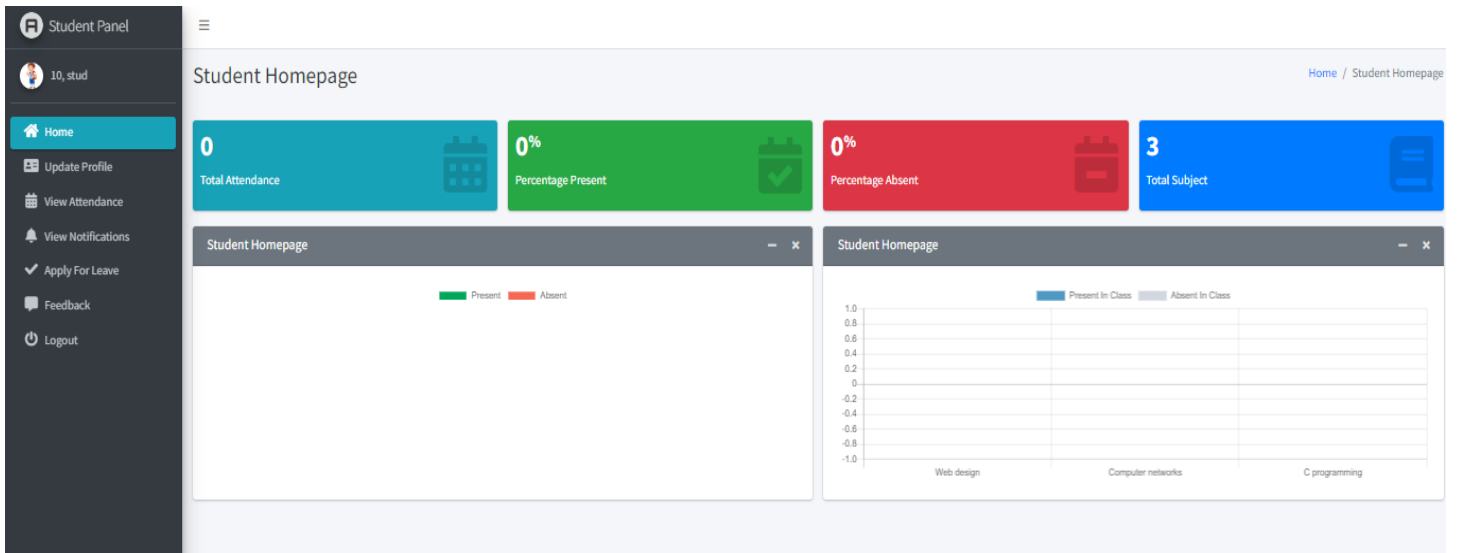
#	Full Name	Email	Gender	Course	Avatar	Actions
1	1, stud	stud1@gmail.com	M	BCA		Edit Delete
2	2, stud	stud2@gmail.com	M	BSC		Edit Delete
3	3, stud	stud3@gmail.com	F	B.Tech		Edit Delete
4	4, stud	stud4@gmail.com	M	BCA		Edit Delete
5	5, stud	stud5@gmail.com	M	BCA		Edit Delete
6	10, stud	stud10@gmail.com	M	BCA		Edit Delete

7.2 Staff Panel Screens



The screenshot shows the 'Take Attendance' screen. The sidebar is identical to the previous screenshot, with 'Take Attendance' highlighted in blue. The main content area has a header 'Take Attendance' and a breadcrumb 'Home / Take Attendance'. Below the header is a form with two dropdown menus: 'Subject' and 'Session Year'. Below the dropdowns is a green button labeled 'Fetch Students'.

7.3 Student Panel Screens




The screenshot shows the 'Apply for leave' screen within the 'Student Panel'. The sidebar is identical to the previous screen, with 'Apply For Leave' now selected in the menu. The main content area is titled 'Apply for leave' and contains a form with the following elements:


- Date:** A text input field with the placeholder 'dd-----yyyy' and a calendar icon on the right.
- Message:** A large text area for writing the leave message.
- Apply For Leave:** A prominent green button.
- Leave History:** A table with the following structure:

ID	Date	Message	Status


Login page

ACADEMY NEXUS





☐ I'm not a robot


reCAPTCHA
[Privacy](#) * [Terms](#)

☐ **Remember Me**

[Forgot Password?](#)

Log In

8. Testing Strategies and Decision Tools

8.1 Unit Testing

Unit Testing focuses on the smallest parts of the software, often individual functions or methods, to ensure they work correctly. For the College Management System in Django, unit tests can be created for various models, views, and utilities.

Strategies:

1. Test Models:

- Ensure all fields in the models are correctly set up.
- Verify relationships (ForeignKey, ManyToManyField) between models.
- Validate constraints and default values.

2. Test Views:

- Test each view function or class-based view to ensure correct rendering of templates.
- Verify context data passed to templates.

3. Test Forms:

- Check form validation for both correct and incorrect inputs.
- Ensure form save methods perform the intended operations.

4. Test Utilities:

- Test helper functions and utilities to ensure they return expected results.

Tools:

- **Django's Test Framework:** Provides a rich set of tools for testing Django applications.
- **Unittest:** Python's built-in testing framework.
- **Pytest:** A more flexible testing framework that can be used with Django

8.2 Integration Testing

Integration Testing ensures that different parts of the system work together. For the College Management System, integration tests can verify the interaction between models, views, templates, and forms.

Strategies:

1. **Test Data Flow:**
 - Ensure data flows correctly between different parts of the application (e.g., model updates reflecting in views).
2. **Test End-to-End Scenarios:**
 - Simulate user interactions from the frontend to the backend to verify the complete functionality of features.

Tools:

- **Selenium:** For end-to-end testing of web applications.
- **Django's LiveServerTestCase:** For running live tests against the server.

8.3 User Acceptance Testing

User Acceptance Testing (UAT) ensures that the system meets the business requirements and is ready for use by end users. It often involves real-world scenarios and end-user feedback.

Strategies:

1. **Test Real Scenarios:**
 - Verify the system by simulating real user workflows (e.g., student registration, attendance marking, etc.).
2. **Feedback Loops:**
 - Gather feedback from actual users and incorporate changes as needed.
3. **Test on Different Devices:**
 - Ensure the application works well across different devices and browsers.

Tools:

- **Manual Testing:** By actual end-users.
- **Automated UAT Tools:** Like TestRail or Zephyr for managing test cases and results.
- **BrowserStack:** For cross-browser testing.

9. Limitations, Conclusion, and Future Enhancements

9.1 Limitations and Drawbacks

While the College Management System is robust and feature-rich, it has certain limitations:

- **Scalability Issues:** As the number of users increases, the system may face performance issues due to its reliance on SQLite, which is not optimal for high-traffic applications.
- **Limited Customization:** The current version lacks advanced customization options for different institutions' unique requirements.
- **Security Concerns:** Basic security measures are implemented, but additional layers such as multi-factor authentication and advanced encryption are not present.
- **Mobile Compatibility:** The system is not fully optimized for mobile devices, potentially affecting user experience on smartphones and tablets.

9.2 Conclusion and Future Enhancements

Conclusion

The College Management System built with Django is a comprehensive solution for managing student and staff information, attendance, and notifications. It offers a user-friendly interface with distinct panels for students, staff, and administrators, making it easy to handle various college records and activities. Despite its limitations, the system provides a solid foundation for digital management in educational institutions.

Future Enhancements

To improve the system, several enhancements can be considered:

- **Database Upgrade:** Migrating from SQLite to a more robust database system like PostgreSQL to handle higher traffic and improve performance.
- **Enhanced Security:** Implementing advanced security features such as multi-factor authentication, encryption, and regular security audits.
- **Customization Options:** Adding features for more extensive customization to cater to different institutional needs.
- **Mobile Optimization:** Developing a mobile-friendly version or a dedicated mobile app to enhance accessibility and user experience on various devices.

10. User Manual, and References & Bibliography

10.1 User Manual (Installation Guide, User Guide)

Installation Guide

1. **Create Virtual Environment:** Navigate to the project root folder, create, and activate a virtual environment.
2. **Install Requirements:** Run `pip install -r requirements.txt` to install necessary dependencies.
3. **Database Migrations:** Execute `python manage.py makemigrations` and `python manage.py migrate` to set up the database.
4. **Run Server:** Start the server with `python manage.py runserver` and access the application via `http://127.0.0.1:[PORT_NUMBER]/`.
5. **Create Superuser:** Create an admin user using `python manage.py createsuperuser`.

User Guide

- **Admin Panel:** Manage courses, subjects, sessions, staff, students, and notifications. View attendance records and handle leave applications.
- **Staff Panel:** Manage student results, take attendance, apply for leave, and view notifications.
- **Student Panel:** View and update profiles, check attendance records, apply for leave, and view notifications.

10.2 References & Bibliography

Online Documentation :

- Django Documentation
- Bootstrap Documentation
- Selenium Documentation
- Python Unittest Documentation
- Pytest Documentation

Books :

- "Two Scoops of Django 3.x: Best Practices for the Django Web Framework" by Daniel Roy Greenfeld and Audrey Roy Greenfeld
- "Django for Professionals: Production websites with Python & Django" by William S. Vincent
- "Python Testing with pytest: Simple, Rapid, Effective, and Scalable" by Brian Okken
- "Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript" by Harry J.W. Percival
- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Robbins

Links :

- <https://www.google.com/>
- <https://www.drawio.com/>
- <https://www.youtube.com/>
- <https://www.w3schools.com/python/>