ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Smartcampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System Using Object-Oriented Programming

Alfonso, Cyrille Anne P., Corpuz, Catherine, Concillado, Adrian N., Labasbas, Grysczen Kientzler U., Melchor, Riemond Erich G., Molina, Euan Edrick D., Reambillo, Brian Kenneth V.

Department of Computer Engineering Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) Nagtahan, Manila, Philippines

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150600022

Received: 10 June 2026; Accepted: 15 June 2026; Published: 03 July 2026

ABSTRACT

The SmartCampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System Using Object-Oriented Programming is a desktop application developed using Python, Tkinter for the graphical user interface, and a database management system (such as SQLite or MySQL) for data storage and retrieval. The system is designed to automate student enrollment processes and campus event ticketing operations, reducing manual paperwork and improving administrative efficiency.

The application provides modules for student registration, course enrollment, event creation, ticket issuance, attendee tracking, and report generation. By applying Object-Oriented Programming (OOP) principles, the system achieves better code organization, maintainability, scalability, and reusability. Database integration enables persistent storage of student records, enrollment transactions, event details, and ticket information, while Tkinter provides an interactive and user-friendly interface for administrators and students.

The system was developed following standard software development practices, including requirements analysis, design, implementation, testing, and evaluation. Testing results indicate that the application can efficiently manage enrollment and event-related transactions while maintaining data accuracy and consistency. The study demonstrates that combining Python, Tkinter, and a relational database can produce a practical, cost-effective, and scalable campus management solution suitable for educational institutions.

Keywords: Python Programming, Tkinter, Database Management System, SQLite/MySQL, Object-Oriented Programming (OOP), Student Enrollment System, Event Ticketing System, Campus Management System, CRUD Operations, Desktop Application Development.

INTRODUCTION

Background of the Study

Many educational institutions still rely on manual processes and separate systems for managing student enrollment and campus event registrations. These traditional methods often involve paper forms, spreadsheets, and manual recordkeeping, which can result in data duplication, processing delays, misplaced records, and difficulties in tracking student participation. Managing enrollment records and event ticketing separately also increases administrative workload and reduces operational efficiency.

With the advancement of information technology, integrated management systems have become essential in improving the efficiency and accuracy of academic and campus-related operations. Computerized systems provide secure data storage, faster processing, improved accessibility, and better monitoring of student information and event activities. The integration of enrollment and event management into a single platform allows institutions to streamline administrative processes and enhance the overall student experience.

The **SmartCampus Hub** was developed to automate student enrollment and campus event ticketing processes

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

using **Python programming**, **Object-Oriented Programming (OOP) principles**, **Tkinter Graphical User Interface (GUI)**, and **SQLite database integration**. The system enables administrators to register students, manage enrollment records, create campus events, issue event tickets, monitor attendance, and generate reports efficiently. By centralizing these functions into a single desktop application, the system reduces manual work, improves data accuracy, and enhances administrative productivity.

The project demonstrates the practical application of Python programming, database management, and object-oriented software development in creating an integrated campus management solution suitable for educational institutions.

Statement of the Problem

This study aims to answer the following questions:

1. How can the system simplify the management of student enrollment records?
2. How can the system improve the registration and monitoring of campus events?
3. How can a GUI-based application improve efficiency compared to manual enrollment and ticketing processes?
4. How does database integration improve the storage and retrieval of student and event information?
5. How effective is the system in performing functions such as Add, View, Update, Delete, Search, and Report Generation?

Objectives of the Study General Objective

To develop a functional SmartCampus Hub that efficiently manages student enrollment and campus event ticketing using Python programming and SQLite database integration.

Specific Objectives

1. Develop a computerized system for managing student enrollment records.
2. Store and organize student and event information securely using an SQLite database.
3. Facilitate event registration and ticket generation for campus activities.
4. Generate reports for student enrollment and event participation.
5. Reduce manual work and improve data accuracy and accessibility.
6. Create a user-friendly Tkinter GUI for efficient system operation.

Significance of the Study

Students: Provides faster enrollment processing and convenient event registration.

Faculty and Administrators: Reduces paperwork and improves management of student records and event activities.

Educational Institutions: Enhances operational efficiency and supports digital transformation initiatives.

Students and Developers: Serves as a practical application of Python, OOP, GUI development, and database integration.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Future Researchers: Provides a reference for developing integrated campus management systems.

Scope and Limitations Scope

- Developed using Python, Tkinter, and SQLite database integration.
- Desktop-based application for student enrollment and event ticketing.
- Includes Add, View, Update, Delete, Search, and Report Generation functions.
- Manages student records, event information, ticket issuance, and attendance monitoring.
- Provides administrative reporting features.

Limitations

- Designed for single-campus educational institutions.
- No online registration or cloud database support.
- Does not include online payment processing.
- No mobile application support.
- Limited to local database storage and single-user operation.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter presents the relevant theories, studies, and literature that serve as the foundation for the development of the **SmartCampus Hub**, a Python-based Student Enrollment and Event Ticketing Management System. The review covers student enrollment systems, event ticketing systems, database integration, Object-Oriented Programming (OOP), graphical user interfaces (GUI), related studies, related systems, and technologies commonly used in educational management applications.

Student Enrollment Management Systems

Student Enrollment Management Systems are designed to automate the registration, enrollment, and management of student records within educational institutions. Traditional enrollment procedures often involve manual processing, paper-based documentation, and separate record-keeping methods, which may lead to delays, inaccuracies, and administrative inefficiencies.

Key features of enrollment management systems include:

- Student registration and profile management
- Course enrollment and scheduling
- Student record maintenance
- Academic information tracking
- Report generation

Studies indicate that computerized enrollment systems improve data accuracy, reduce administrative workload, and provide faster access to student information. Automated enrollment processes also contribute to better decision-making through organized and accessible records.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Event Ticketing Management Systems

Event Ticketing Management Systems facilitate the organization, registration, and monitoring of events. In educational institutions, campus activities such as seminars, workshops, conferences, and student programs require efficient participant management and attendance tracking.

Key functions of event ticketing systems include:

- Event creation and scheduling
- Ticket generation and distribution
- Participant registration
- Attendance monitoring
- Event reporting and analytics

Research shows that digital ticketing systems improve event organization by reducing manual registration processes, minimizing errors, and providing real-time access to participant information.

Database Integration

Databases play a critical role in management systems by providing secure and organized storage of information. SQLite is widely used in desktop applications due to its lightweight architecture, ease of implementation, and compatibility with Python.

Database integration provides:

- Secure storage of student records
- Efficient retrieval of enrollment information
- Event registration tracking
- CRUD (Create, Read, Update, Delete) operations
- Report generation and data analysis

Proper database design improves data consistency, integrity, accessibility, and reliability, making it an essential component of educational management systems.

Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) is a software development approach that organizes programs into classes and objects. OOP promotes modularity, maintainability, scalability, and code reusability.

In the SmartCampus Hub system:

- Classes represent entities such as Student, Enrollment, Event, and Ticket.
- Objects store specific information related to each entity.
- Methods perform functions such as registration, enrollment processing, ticket generation, and report creation.

The four major principles of OOP include:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

- **Encapsulation** – Protecting and organizing data within classes.
- **Inheritance** – Allowing classes to acquire properties from other classes.
- **Polymorphism** – Enabling multiple forms of method implementation.
- **Abstraction** – Simplifying complex operations through user-friendly interfaces.

Research has shown that OOP-based applications are easier to maintain, expand, and modify compared to procedural programming approaches.

Graphical User Interface (GUI)

A Graphical User Interface (GUI) allows users to interact with a system through visual components such as windows, forms, buttons, menus, and tables. For this study, Tkinter is utilized as the GUI framework because it is built into Python and provides efficient tools for desktop application development.

Benefits of GUI-based systems include:

- Improved user experience
- Reduced training requirements
- Faster navigation and operation
- Better data visualization
- Reduced user errors

GUI applications are particularly beneficial in educational environments where administrators and staff may have varying levels of technical expertise.

Related Studies

Several studies have emphasized the importance of integrating database management, automation, and user-friendly interfaces in educational systems. Researchers found that computerized enrollment systems significantly improve processing speed, data accuracy, and record management efficiency compared to traditional manual methods.

Studies on event management applications have also demonstrated that automated ticketing and attendance tracking improve event coordination and participant management. The combination of enrollment and event management functionalities within a single platform can further enhance administrative productivity and provide a more seamless experience for students and administrators.

Related Systems

Existing enrollment and event management systems commonly provide student registration, record maintenance, event scheduling, and attendance monitoring functionalities. However, many available systems focus only on a single function and require separate applications for enrollment and event management.

The SmartCampus Hub addresses these limitations by integrating enrollment and event ticketing operations into a unified platform. This centralized approach reduces redundancy, improves accessibility, and provides comprehensive management of student-related activities.

Related Technologies

Several technologies contribute to the successful development of management systems:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Python Programming Language

Python is a high-level programming language known for its simplicity, readability, and extensive library support. It is widely used in software development, data processing, and automation applications.

Tkinter

Tkinter is Python's standard GUI toolkit. It enables developers to create windows, forms, buttons, tables, and interactive desktop applications efficiently.

SQLite

SQLite is a lightweight relational database management system commonly used in desktop applications. It supports efficient data storage, retrieval, and management without requiring a separate database server.

SQL

Structured Query Language (SQL) is used to create, retrieve, update, and manage data stored within relational databases. SQL ensures reliable transaction processing and data integrity.

Synthesis of the Study

The reviewed literature consistently highlights the importance of automation, database integration, graphical user interfaces, and object-oriented programming in developing efficient management systems. Enrollment systems improve academic record management, while event ticketing systems enhance event organization and attendance monitoring.

Drawing from these studies and technologies, the **SmartCampus Hub** integrates Python programming, Tkinter GUI development, SQLite database management, and OOP principles to create a centralized platform for student enrollment and campus event ticketing. The system is designed to improve efficiency, accuracy, accessibility, and administrative productivity while reducing the limitations associated with traditional manual processes.

METHODOLOGY

Research Design

This study utilized a **Software Development Research** approach under the **Descriptive Research Design**. The primary goal was to simplify and improve the management of student enrollment and campus event ticketing through a computerized system. This approach enabled the researchers to design, develop, implement, and evaluate a functional system that addresses inefficiencies associated with manual enrollment procedures and event registration processes.

System Development Process

The development of the **SmartCampus Hub** followed the **Software Development Life Cycle (SDLC)** to ensure the system's efficiency, reliability, and quality. The phases included:

Planning

During this phase, the researchers identified common issues encountered in manual enrollment and event registration processes, such as lost records, duplicate entries, delayed processing, and difficulty tracking student participation in campus activities. The objectives, scope, limitations, and required system functionalities were determined through observation, interviews, and analysis of existing procedures.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Design

This phase focused on creating the overall structure and appearance of the system. The researchers designed the system flowchart, user interface layout, database structure, and data flow. The graphical user interface was designed using Tkinter to provide a simple and user-friendly environment for administrators and users.

Development

The system was developed using **Python programming language** with **Object-Oriented Programming (OOP) principles**. This approach allowed the application to be modular, maintainable, and scalable. The major functionalities developed include:

- Student Registration
- Student Enrollment Management
- Event Creation and Management
- Event Ticket Generation
- Attendance Monitoring
- Record Management
- Report Generation

The system utilizes **SQLite database integration** for permanent storage of student records, enrollment information, event details, and ticketing transactions. Tkinter was used to develop the desktop-based graphical user interface.

Testing

Functional testing was conducted to identify and resolve errors, bugs, and system issues. Each module of the application was tested individually and collectively to verify correct operation. The following functions were tested:

- Add Student
- View Records
- Update Records
- Delete Records
- Search Records
- Event Registration
- Ticket Generation
- Report Generation
- Exit Function

Input validation mechanisms were implemented to prevent incomplete, duplicate, and invalid entries, ensuring data integrity and reliability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Implementation

After successful testing, the system was deployed for actual use. Administrators were provided with instructions and user guidelines to ensure effective operation. During implementation, system performance was monitored to verify that all modules functioned according to the intended objectives.

Maintenance

Continuous evaluation and maintenance activities were conducted after deployment to ensure system reliability and efficiency. Necessary updates and improvements were applied to correct issues, enhance functionality, and maintain compatibility with user requirements.

System Flowchart / Design

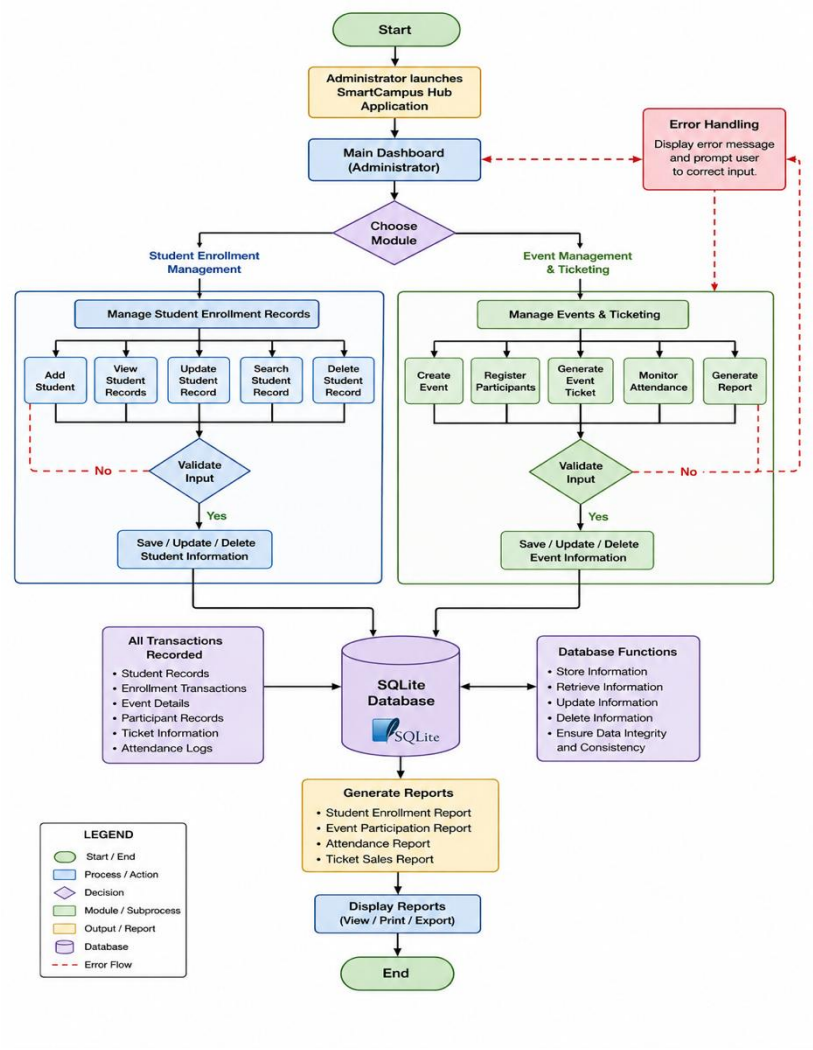

Figure 1. Overall System Flowchart

The system flow begins when the administrator launches the application and accesses the main dashboard. The administrator may choose to manage student enrollment records or campus event information.

For student enrollment, the administrator can add, view, update, search, or delete student records. The system validates user inputs before storing information in the SQLite database.

For event management, administrators can create events, register participants, generate event tickets, monitor attendance, and generate reports. The system performs validation checks and updates the database accordingly.

All transactions are recorded in the database, ensuring accurate storage and retrieval of information. Error-

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

handling and validation mechanisms are incorporated throughout the process to maintain data accuracy and consistency.

Tools and Technologies Used

Several tools and technologies were utilized in the development of the SmartCampus Hub to ensure functionality, reliability, and usability.

Visual Studio Code

Visual Studio Code served as the primary development environment. It was used for coding, debugging, testing, and project management throughout the development process.

Python

Python was used as the primary programming language for implementing the system's functionalities, including enrollment processing, event management, ticket generation, database operations, and report generation.

Tkinter

Tkinter was utilized for developing the graphical user interface. It provided windows, forms, buttons, labels, tables, and menus necessary for user interaction.

SQLite

SQLite served as the database management system of the project. It stores student records, enrollment information, event details, attendance records, and ticket transactions securely and efficiently.

Object-Oriented Programming (OOP)

Object-Oriented Programming principles were implemented to improve system organization, modularity, maintainability, and scalability. Classes such as Student, Enrollment, Event, Ticket, and DatabaseManager were utilized to structure system operations effectively.

This methodology ensures that the **SmartCampus Hub** is efficient, reliable, user-friendly, and maintainable by integrating Python programming, Tkinter GUI development, SQLite database management, and Object-Oriented Programming principles into a comprehensive campus management solution.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed **SmartCampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System**, including its key features, graphical user interface, database integration, and functional testing results.

System Overview

The SmartCampus Hub is a desktop-based application designed to simplify and automate student enrollment and campus event ticketing operations. The system provides a centralized platform where administrators can manage student records, process enrollments, create campus events, generate event tickets, and monitor attendance efficiently.

The interface is organized into functional modules that allow administrators to perform essential operations through a user-friendly graphical user interface. Student information, enrollment records, event details, and ticket transactions are stored and managed through an SQLite database, ensuring data accuracy and accessibility.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The system integrates the following functionalities:

- Student registration and profile management
- Enrollment management
- Event creation and scheduling
- Event ticket generation
- Attendance monitoring
- Report generation
- CRUD (Create, Read, Update, Delete) operations

The system utilizes **Python programming**, **Object-Oriented Programming (OOP) principles**, **Tkinter GUI**, and

SQLite database integration for reliable and efficient operation.

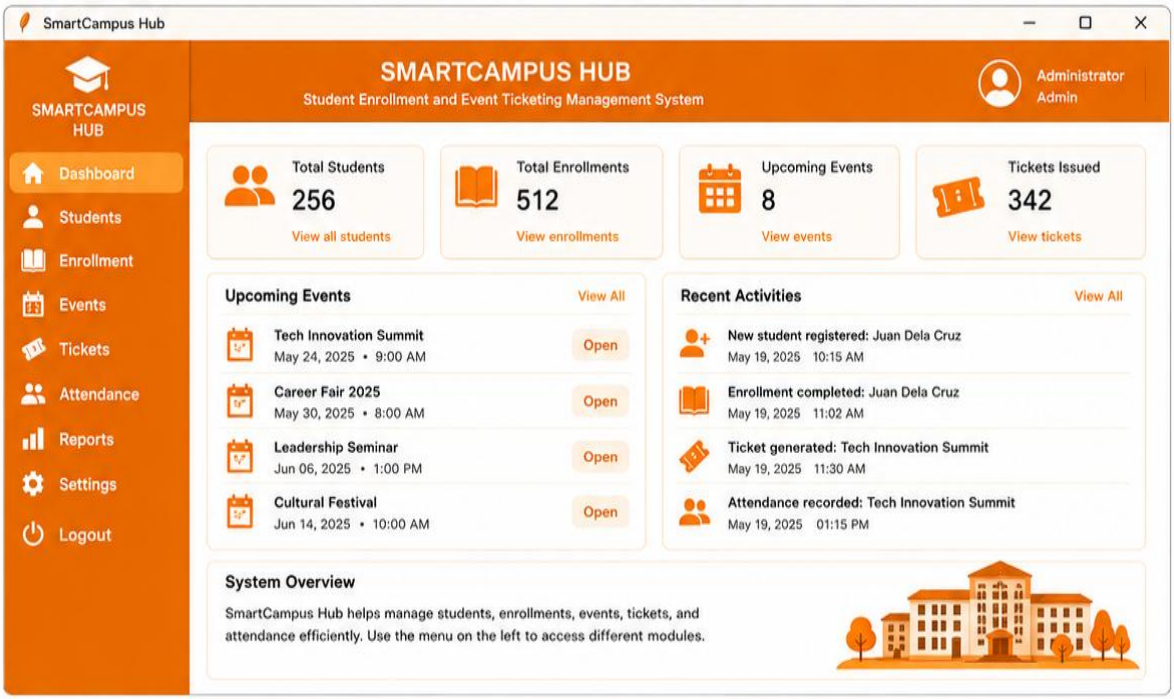

Figure 2. SmartCampus Hub Main Dashboard Interface

The figure illustrates the Main Dashboard Interface of the SmartCampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System. Serving as the central control panel of the application, the dashboard provides administrators with real-time summaries of key institutional data, including total students, enrollment records, upcoming events, and tickets issued. It also displays scheduled campus events, recent system activities, and

navigation links to core modules such as Students, Enrollment, Events, Tickets, Attendance, Reports, and Settings. Through its organized graphical user interface, the dashboard enables efficient monitoring of student enrollment and event ticketing operations while providing quick access to essential system functionalities.

System Features and Results Student Registration Module

The Student Registration Module allows administrators to register new students by entering information such

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

as:

- Student ID
- Full Name
- Course
- Year Level
- Contact Information
- Email Address

Result:

- Student information is successfully stored in the database.
- Input validation prevents incomplete and duplicate entries.
- Student records are organized and easily accessible.

Enrollment Management Module

The Enrollment Module enables administrators to process student enrollments and maintain enrollment records.

Result:

- Enrollment transactions are recorded accurately.
- Student enrollment status is updated automatically.
- Records can be searched and retrieved instantly.

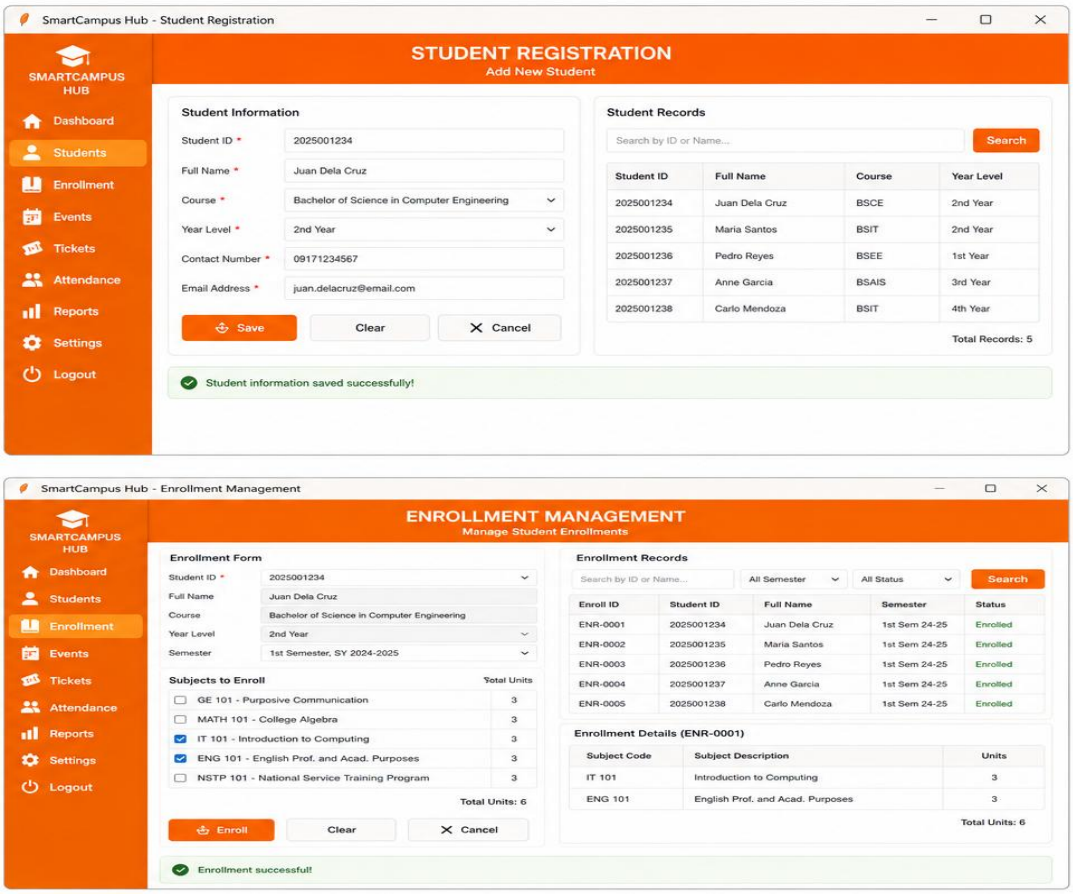

Figure 3. Student Registration and Enrollment Management Modules

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The figure illustrates the Student Registration Module and Enrollment Management Module of the SmartCampus Hub. The Student Registration Module enables administrators to add, search, view, and manage student records by capturing essential information such as student ID, full name, course, year level, contact number, and email address.

Meanwhile, the Enrollment Management Module facilitates the enrollment process by allowing administrators to assign subjects, manage enrollment records, track enrollment status, and monitor academic load. Both modules are integrated with the system database to ensure accurate record storage, efficient retrieval of information, and streamlined management of student enrollment operations within the institution.

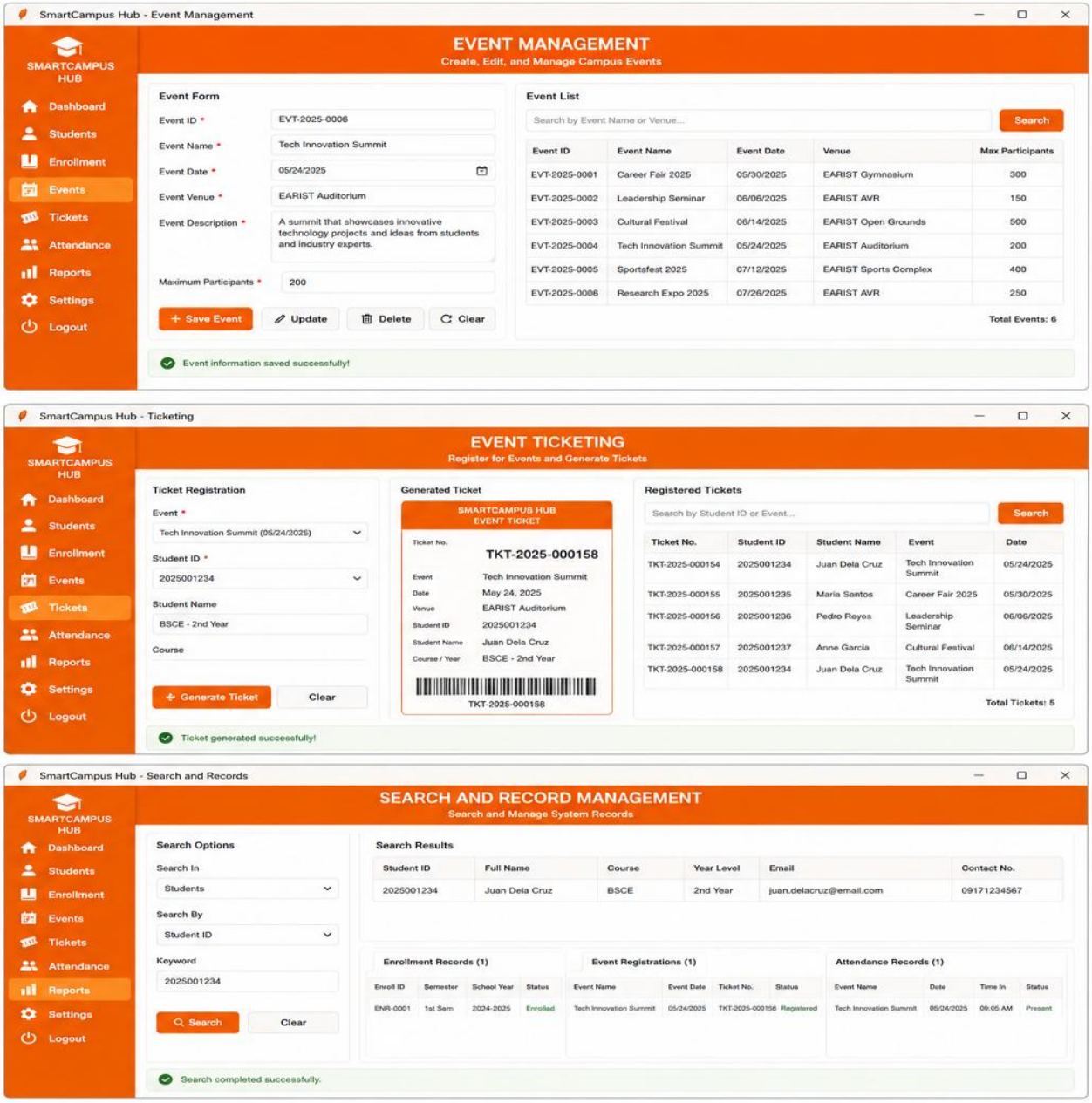

Figure 4. Event Management, Event Ticketing, and Search & Record Management Modules

The figure illustrates three major functional components of the SmartCampus Hub: the Event Management Module, Event Ticketing Module, and Search & Record Management Module. The Event Management Module enables administrators to create, update, delete, and manage campus events by maintaining event information such as event names, schedules, venues, descriptions, and participant limits. The Event Ticketing Module facilitates student registration for campus activities and automatically generates unique event tickets, ensuring accurate participant tracking and attendance monitoring. Meanwhile, the Search & Record Management Module provides comprehensive search and retrieval capabilities for student records,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

enrollment information, event registrations, and attendance data, allowing administrators to efficiently access and manage institutional records. Together, these modules enhance campus event administration, improve record accessibility, and support efficient management of student participation in academic and extracurricular activities.

Event Management Module

The Event Management Module allows administrators to create, edit, and manage campus events. The system stores:

- Event ID
- Event Name
- Event Date
- Event Venue
- Event Description
- Maximum Participants

Result:

- Event information is successfully saved and updated.
- Administrators can monitor event details efficiently.
- Event schedules are organized and accessible.

Event Ticket Generation Module

The Ticketing Module allows students to register for events and receive corresponding event tickets.

Result:

- Unique ticket numbers are generated automatically.
- Participant registration is recorded accurately.
- Event attendance can be monitored through ticket validation.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

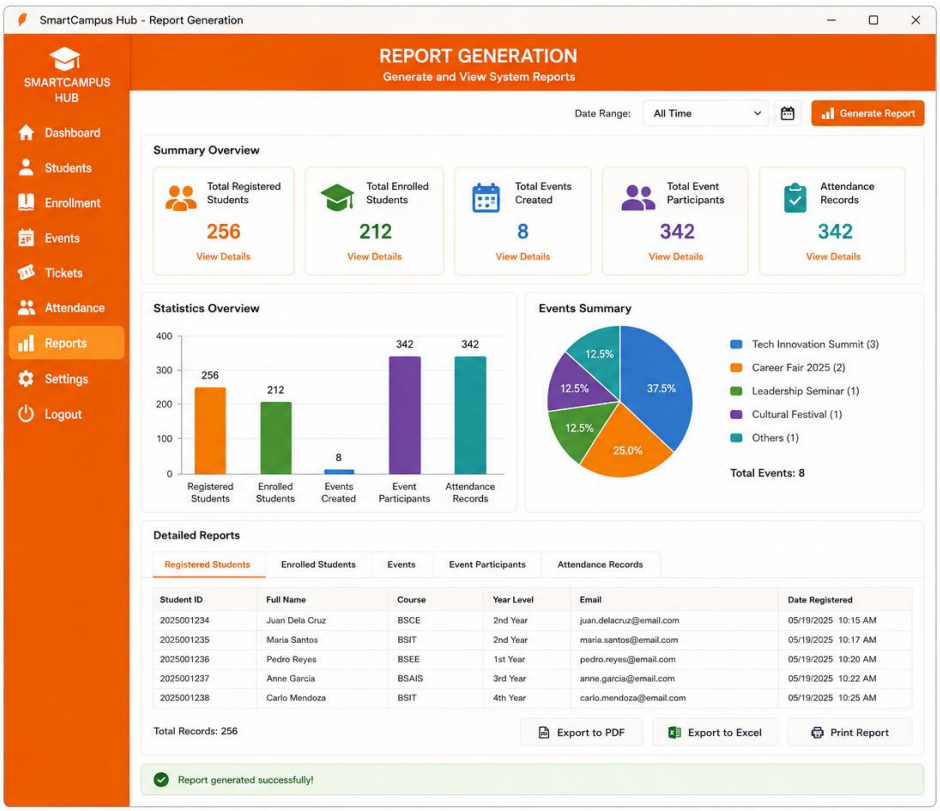

Figure 5. Report Generation and Analytics Dashboard

The figure illustrates the Report Generation and Analytics Dashboard of the SmartCampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System. This module provides administrators with comprehensive reporting and statistical analysis tools for monitoring campus operations. It displays summary metrics such as total registered students, enrolled students, events created, event participants, and attendance records. The

dashboard also includes graphical visualizations, including bar charts and pie charts, to present enrollment and event participation statistics. Additionally, detailed report tables allow administrators to review institutional records and export reports in various formats such as PDF and Excel or generate printable copies. Through its integrated reporting capabilities, the module supports data-driven decision-making, performance monitoring, and efficient management of student enrollment and campus event activities.

Search and Record Management Module

The system includes search functionality for locating student records, enrollment information, event details, and ticket transactions.

Result:

- Search results are displayed accurately and instantly.
- Administrative tasks are completed more efficiently.
- Information retrieval time is significantly reduced.

Report Generation Module

The system generates reports related to:

- Total Registered Students

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

- Total Enrolled Students
- Total Events Created
- Total Event Participants
- Attendance Records

Result:

- Reports reflect current database information.
- Administrators can monitor campus activities effectively.
- Decision-making is supported through organized reporting.

Graphical User Interface

The Tkinter-based Graphical User Interface includes forms, tables, buttons, and menus that allow users to navigate the system easily.

Result:

- Users can operate the system without advanced technical knowledge.
- Visual components improve usability and efficiency.
- Error messages and confirmations guide users throughout operations.

Functional Testing Results

Functional testing was conducted to evaluate the performance and reliability of the system.

All test cases passed, confirming that the system is functional, reliable, and capable of performing its intended.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of the **SmartCampus Hub: A Python-Based Student Enrollment and Event Ticketing Management System**, the following conclusions were drawn:

1. **Automated Enrollment and Event Management:** The system successfully automates student enrollment and campus event ticketing processes, reducing dependence on manual record-keeping and improving administrative efficiency.
2. **Accuracy and Reliability:** Integration with the SQLite database ensures secure storage, accurate management, and reliable retrieval of student records, enrollment data, event information, and ticket transactions.
3. **Functional Performance:** Functional testing confirmed that all major modules, including Student Registration, Enrollment Management, Event Management, Ticket Generation, Search, Update, Delete, and Report Generation, operated correctly and efficiently.
4. **User-Friendly Interface:** The Tkinter-based graphical user interface enables administrators to navigate and use the system easily, minimizing the need for extensive technical training.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

5. **Improved Data Management:** The centralized database structure improves organization, accessibility, and consistency of records while reducing duplication and data-entry errors.

6. **Operational Efficiency:** The system significantly enhances campus administrative operations by providing faster transaction processing, real-time record access, and automated reporting capabilities.

Overall, the SmartCampus Hub achieved its primary objective of developing an integrated, reliable, and user-friendly platform for managing student enrollment and campus event ticketing activities. The system demonstrates the effectiveness of combining Python programming, Object-Oriented Programming (OOP), Tkinter GUI development, and SQLite database integration in educational management applications.

Recommendations

Based on the findings and limitations observed during the development of the system, the following recommendations are proposed:

- **User Authentication and Role Management:** Implement role-based access control for administrators, faculty members, event organizers, and students to improve security and accountability.

- **Online Enrollment and Registration:** Enhance the system by providing web-based access for students to enroll and register for events remotely.

- **QR Code Ticketing Integration:** Incorporate QR code technology for faster event check-in, ticket validation, and attendance monitoring.

- **Online Payment Processing:** Add payment gateway integration to support event registration fees and other campus-related transactions.

- **Multi-User Support:** Enable multiple users to access and manage the system simultaneously through a networked database environment.

- **Cloud Database Integration:** Implement cloud storage and database synchronization to improve data accessibility, backup, and disaster recovery.

- **SMS and Email Notifications:** Develop automated notification features for enrollment confirmations, event reminders, ticket issuance, and important announcements.

- **Mobile Application Development:** Create a mobile version of the system to allow students and administrators to access services through smartphones and tablets.

- **Analytics Dashboard:** Integrate graphical dashboards displaying enrollment statistics, event participation trends, and attendance reports for improved decision-making.

- **Continuous Maintenance and Evaluation:** Conduct regular system updates, performance evaluations, and user feedback assessments to ensure long-term reliability and usability.

In conclusion, the SmartCampus Hub successfully fulfills its intended objectives. However, implementing these recommendations will further improve the system's functionality, security, scalability, and overall effectiveness, making it more suitable for broader deployment in modern educational institutions.

ACKNOWLEDGEMENT

The researchers would like to express their heartfelt appreciation to Engr. Meshelle N. Fabro, PCpE, for her dedicated mentorship, professional guidance, and expert supervision throughout the completion of this study. Her valuable insights, constructive suggestions, and continuous encouragement significantly contributed to the successful development and improvement of this research.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The researchers also extend their sincere gratitude to the Eulogio 'Amang' Rodriguez Institute of Science and Technology, particularly the Department of Computer Engineering, together with its faculty members, for providing the academic knowledge, technical resources, and institutional support necessary for the accomplishment of this project. The institution's commitment to excellence in education, innovation, and research played an essential role in the success of this study.

Special thanks are also given to the families and friends of the researchers for their unwavering support, patience, understanding, and motivation. Their encouragement served as a source of inspiration and strength throughout the challenges encountered during the research process.

Above all, the researchers offer their deepest gratitude to Almighty God for the wisdom, guidance, strength, and perseverance bestowed upon them throughout this endeavor. This work is humbly dedicated to His glory, acknowledging that every achievement was made possible through His grace and blessings.

REFERENCES

1. Bhavani, V. T., Chinthana, S., Aishwarya, K., Shweta, B., & Nirmala, S. (2025). Implementation of a Python-based expense tracker and management system. International Journal for Multidisciplinary Research.
2. Dharwani, H., Patel, K., & Shah, R. (2023). Development of student management systems using Python and database technologies. International Journal of Computer Science and Information Technology, 15(2), 45–53.
3. Kumar, R., & Singh, A. (2021). Database integration techniques in educational and small business management systems. International Journal of Computer Applications, 174(15), 12–20.
4. Matcha, S., & Mishra, R. (2025). Database selection and management: Choosing the right database (SQL vs. NoSQL) for your application. International Journal of Research in Humanities & Social Sciences, 13(3), 70–76.
5. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.
6. Pushpa, P., Kumar, V., & Reddy, S. (2024). Principles of robust database design for small business management systems. Journal of Software Engineering and Applications, 17(4), 211–223.
7. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.
8. Sommerville, I. (2020). Software Engineering (10th ed.). Pearson Education.
9. Šušter, A., & Ranisavljević, N. (2023). SQL-driven systems for small business and institutional management: Performance and reliability analysis. Journal of Database Management, 35(2), 78–89.

ABOUT THE AUTHORS

Cyrille Anne P. Alfonso

Cyrille Anne P. Alfonso is currently a first-year Bachelor of Science in Computer Engineering student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. She is passionate about technology, programming, and continuous learning. Driven by curiosity and determination, she continuously explores different areas of computer engineering, particularly system development, database management, and software applications.

As a student, she values creativity, teamwork, and perseverance in accomplishing academic and technical projects. She enjoys learning new concepts and applying them to practical activities that improve both her analytical and problem-solving skills. Through her studies and experiences, she aims to become a skilled and competent computer engineer who can contribute innovative ideas and solutions in the field of technology.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Adrian N. Concillado

Adrian N. Concillado is currently a first-year Computer Engineering student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. Driven by a passion for technology and engineering, he has cultivated a strong interest in software development, computer systems, and emerging innovations. He consistently strengthens his technical competencies and analytical skills through academic projects and hands-on experiences.

While committed to broadening his knowledge in the field of Computer Engineering, he also strives to develop innovative solutions that can make meaningful contributions to both industry and society. Guided by perseverance, curiosity, and a dedication to lifelong learning, he aspires to become a skilled and accomplished Computer Engineer in the future.

Catherine Corpuz

Catherine Corpuz is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. She has a growing interest in the field of computer engineering, particularly in areas related to software development, database management, and emerging technologies. Through her academic experiences, she continues to develop her technical competencies, critical thinking, and problem-solving skills.

As an aspiring computer engineer, she values diligence, collaboration, and continuous learning in accomplishing academic endeavors. She is committed to expanding her knowledge and applying it to innovative projects that address real-world challenges. In the future, she hopes to contribute meaningful advancements and practical solutions in the field of technology.

Grysczen Kientzler U. Labasbas

Grysczen Kientzler U. Labasbas is currently a first-year Bachelor of Science in Computer Engineering student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. As an aspiring computer engineer, she is dedicated to building a strong foundation in the field through academic learning, laboratory activities, and practical experiences.

She continuously strives to improve her technical knowledge, critical thinking, communication, and problem-solving skills while adapting to the challenges and responsibilities of college life. Through continuous learning and self-improvement, she hopes to become a competent and responsible professional in the field of computer engineering. This work reflects her dedication, perseverance, and commitment to academic excellence.

Riemond Erich G. Melchor

Riemond Erich G. Melchor is currently a first-year Computer Engineering student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. He is passionate about technology, programming, and engineering, with a growing interest in software development, computer systems, and emerging technologies.

He continuously develops his technical knowledge and problem-solving skills through academic projects and hands-on learning experiences. He is dedicated to expanding his expertise in the field of Computer Engineering and aims to contribute innovative solutions that can benefit both industry and society. Through perseverance, curiosity, and commitment to lifelong learning, he aspires to become a competent and successful Computer Engineer in the future.

Euan Edrick D. Molina

Euan Edrick D. Molina is a Computer Engineering student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. He is a hardworking and adaptable student who values discipline, cooperation, and continuous improvement.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Euan actively participates in academic activities and strives to enhance his skills through learning and experience. With determination and dedication, he hopes to achieve his personal and professional goals in the years ahead.

Brian Kenneth V. Reambillo

Brian Kenneth V. Reambillo is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio 'Amang' Rodriguez Institute of Science and Technology. He has a growing interest in the field of computer engineering, particularly in system logic design, programming, software development, and computer systems.

Through his academic experiences, he continues to develop his technical competencies, analytical thinking, and problem-solving skills. He is especially interested in understanding how hardware and software work together to create efficient and reliable computing solutions.

As an aspiring computer engineer, Brian values diligence, collaboration, and continuous learning in accomplishing academic and technical endeavors. He is committed to expanding his knowledge of emerging technologies and applying logical and innovative approaches to solving real-world problems. Through dedication and perseverance, he aims to contribute meaningful advancements in technology and develop practical solutions that can positively impact society in the future.