

EventEase: Design and Implementation of a GUI-Based Python Event Registration and Ticketing System with Sqlite Integration

Fetizanan, Jan Clarence B., Fortuno, Jimuel O., Galam, Justin Karl B., Garbin, John Angelo C.,
Hermosilla, Ryan Zedrick P., Nono, Hempbahaghari M., Engr. Fabro, Meshelle N.

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

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150600021>

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

ABSTRACT

Efficient management of event registration and ticketing is crucial for the success of academic, corporate, and social events. Traditional manual methods, including paper-based logs and unintegrated spreadsheets, are prone to errors, duplicate entries, and operational delays, especially during high-volume registration periods. This study presents **EventEase**, a modern, **GUI-based Event Registration and Ticketing System** engineered using **Python** and integrated with a locally isolated **SQLite relational database engine**.

The application utilizes advanced visual toolkits like customtkinter to deliver a professional user experience featuring real-time input normalization and an interactive data visualization backend powered by matplotlib. Architecturally, the system separates roles into administrative control channels and customer booking gateways. Administrators have access to full event creation mechanisms supporting multi-tiered ticket rates (Regular, VIP, Student), granular category and venue configuration utilities, real-time attendee gate verification with explicit timestamp logging, and automated transaction backup systems running on Write-Ahead Logging (WAL) parameters. Simultaneously, the front-end customer client handles seamless profile creation, dynamic seat availability checks, cash-tender calculation wizards, and instant receipt generation.

Functional testing confirmed that all database joins, user-role authentications, and reporting modules operate with zero record degradation. EventEase demonstrates how desktop applications can be built to optimize operational capacity, remove data redundancy, and secure transactional integrity within institutional environments.

Keywords: EventEase, Event Registration, Ticketing System, Python, GUI, SQLite, Database Management, Automation

INTRODUCTION

Background of the Study

Academic institutes and professional groups frequently organize events ranging from scientific seminars and sports tournaments to student assemblies. Managing these functions requires stable coordination across logistics, attendee registrations, and secure financial processing. Unfortunately, many organizations still rely on manual tracking logs, disjointed digital document forms, or unlinked spreadsheets. These fragmented methods often lead to duplicate entries, data loss, slow event entry lines, and a lack of real-time insights into attendance or revenue trends.

To address these challenges, this study details the design, architectural integration, and implementation of EventEase (also deployed under the EventPro Management Engine framework). EventEase is a high-performance desktop application built in Python that replaces outdated, paper-based workflows with a unified database structure and a modern graphical interface. Leveraging Python's development ecosystems alongside an

optimized SQLite relational engine allows the system to run locally with minimal hardware requirements while maintaining transactional data safety. By implementing specialized security measures like SHA-256 password hashing and Write-Ahead Logging, EventEase bridges the gap between accessible desktop software design and enterprise-level data reliability.

Statement of the Problem

This study aims to develop an Event Registration and Ticketing System that can manage, organize, and track records efficiently through a GUI-based application. Manual ticketing and guest tracking processes often cause slow entry lines, human counting mistakes, and issues with misplaced registration paper sheets. By building a desktop software interface integrated with multi-table database structures, event coordinators can safely catalog attendee profiles and track event details using a centralized computer structure.

Specifically, it aims to answer the following questions:

How can the system automate the process of event registration and ticketing through an interactive graphical user interface?

How can an embedded SQLite database structure efficiently store and link attendee data with event management configurations?

How can the system reduce administrative errors, such as duplicate registration tokens or accidental log loss, by providing automated database tools and immediate backup controls?

How can a GUI dashboard layout improve user interaction, navigation speed, and overall usability for event desk staff?

Objectives of the Study

The primary objective of this research is to design, develop, and deploy EventEase, a modern desktop application that automates event registration and ticket tracking through an integrated database system.

The specific technical objectives include:

- Developing an asymmetric login gateway that securely authenticates users and directs them to the appropriate dashboard based on their role (Admin or Customer).
- Implementing an SQLite storage layer that enforces data normalization across distinct tables for tickets, events, users, categories, and venues.
- Building an administrative control panel that simplifies event scheduling, categorization, tiered ticket pricing, and entry gate check-ins.
- Creating a customer portal that allows users to browse open events, view accurate real-time seat counts, calculate ticket costs, and generate digital receipts.
- Integrating a data visualization component within the admin dashboard to generate live metric summaries and 7-day registration charts.
- Ensuring system data safety by implementing a secure, isolated database backup utility within the application.

Significance of the Study

The development of this GUI-Based Event Registration and Ticketing System provides valuable practical benefits to various groups involved in event organization and administrative management. Shifting from a

manual guest list to an automated desktop application reduces traditional bottlenecks and improves data security. The specific individuals and groups who will benefit from this study include:

The specific individuals and groups who will benefit from this study include:

Event Organizers and Coordinators: This study benefits event managers by giving them a reliable, fast desktop interface to track attendee metrics. Features like immediate data loading, live entry counting, and automated backup tools protect them from losing data and save time during heavy check-in hours.

Front-Desk Event Staff: The system simplifies the daily tasks of registration personnel by replacing manual paperwork with a clear graphical layout. Clicking buttons to add, clear, or modify database logs allows workers to handle attendee requests quickly with less physical effort and fatigue.

Event Attendees: Guests benefit directly from an optimized registration process. The fast lookup speeds and instant status checks minimize waiting times in entrance lines, leading to a much better customer service experience at the venue.

Future Researchers and Student Programmers: This research serves as a clear reference for peers studying Object-Oriented Programming (OOP) and Python database interfaces. The design provides a practical example of combining data collection forms with secure back-end table relationships.

SCOPE AND LIMITATIONS

Scope:

- Developed using **Python programming**.
- GUI-based application using **Tkinter**.
- Includes core modules: Dashboard, Add Record, View Records, Manage Events, and Settings.
- Functionalities include: Add Record, View Record, Update Information, Cancel Ticket, Clear Inputs.
- Core Modules: Complete support for user registration, multi-tier ticket sales, venue and category management, real-time seat allocation tracking, and administrative check-ins.
- Reporting Tools: Integrated visual charts for tracking weekly registration volume and ticket sales trends.
- Data Security: Implementation of SHA-256 cryptographic hashing to secure user password records.

Limitations:

- Limited to basic ticketing functions (add, update, delete, clear, view records).
- GUI is simple and direct but not responsive to advanced themes or multiple devices.
- Cannot associate tickets directly under specific assigned events.
- Does not include online payment or multi-user network support.
- Deployment Architecture: Operating as a local desktop program, the system requires the Python runtime environment and dependencies (customtkinter, tkcalendar, pillow, matplotlib) to be installed on the host machine.

- **Networking:** The application uses a local SQLite database file, meaning it does not sync across an online cloud network or support multi-device setups without shared local network storage.
- **Payment Gateways:** Financial checkout functions use a simulated local point-of-sale cash window that calculates change, rather than connecting directly to third-party digital payment APIs or credit card processors.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter presents the theoretical framework, prior studies, and related literature that form the basis for the development of **EventEase**, a GUI-based Python Event Registration and Ticketing System. The review covers **GUI-based event management systems, database management, Object-Oriented Programming (OOP), automated ticketing, and security systems.**

GUI-Based Event Registration Systems

GUI-based event registration systems provide a visual and interactive platform for attendees and administrators, enhancing usability, efficiency, and data accuracy.

Alfirdaus & Widodo (2020) highlighted that digital ticketing systems reduce administrative workload and speed up registration while ensuring participant validation.

Pradana et al. (2021) noted that GUI-based systems offer efficient and user-friendly interfaces that simplify ticket reservation and attendee management.

Setiawan & Pramudya (2023) emphasized user-centered design, showing that intuitive interfaces improve navigation and accessibility, reducing reliance on third-party tools.

Rahmawati et al. (2024) developed an integrated event management platform combining ticketing and networking, concluding that such systems improve organizer-participant communication and accessibility.

Related Computerized Management and Security Systems

Automated event management systems improve administrative efficiency while enhancing security:

Chandra et al. (2022) introduced *CU-EVENTS*, a university-based system demonstrating that GUI-based registration improves monitoring, participant engagement, and operational efficiency compared to manual methods.

Mäkelä et al. (2012) showed that NFC-enabled ticketing systems improve security, minimize fraud, and accelerate entry.

Dunphy & Petitcolas (2022) highlighted self-sovereign identity technology, emphasizing secure platforms that prevent unauthorized access, scalping, and fraudulent ticket sales.

Database Management and Automated Ticketing Systems

A **Database Management System (DBMS)** is essential for storing, retrieving, and updating records in real-time:

SQLite is lightweight, serverless, and ideal for desktop applications, supporting CRUD (Create, Read, Update, Delete) operations for attendee and event data (SQLite Consortium, 2024).

Elmasri & Navathe (2015) demonstrated that relational databases improve registration speed and reporting accuracy, enabling organizers to access real-time event schedules.

Automated systems minimize errors, prevent duplicate registrations, and centralize record storage for efficient event management.

Object-Oriented Programming (OOP)

OOP principles enable modular, reusable, and maintainable code in software applications:

- Classes represent entities like Attendee, Event, or Ticket.
- Methods handle actions such as adding attendees, updating records, generating tickets, and reporting.
- OOP improves system scalability, maintainability, and future feature integration.

Synthesis

The reviewed literature indicates that:

Manual registration is inefficient, prone to errors, and difficult to scale.

GUI-based interfaces improve usability, navigation, and participant experience.

Database integration (SQLite) ensures secure, organized, and accurate record management.

Automation and OOP design allow reliable processing of attendee records, tickets, and reports while preventing duplicates and errors.

Security measures such as validation, encryption, and access control are essential for trustworthy ticketing systems.

EventEase applies these principles by integrating Python, OOP, GUI design, and SQLite database management to deliver a secure, efficient, and user-friendly event registration and ticketing system. This system automates attendee registration, ticket issuance, record tracking, and reporting, improving operational efficiency, usability, and data integrity in event management.

METHODOLOGY

Research Design

The study developed and evaluated a **GUI-Based Event Registration and Ticketing System** using a **Developmental Research Design**. This approach was appropriate as it focuses on creating, building, and testing a functioning software application aimed at improving event registration and ticket administration. The system was implemented using **Python**, with **Tkinter** for the graphical user interface (GUI) and **SQLite** for database management.

Using this developmental approach, the researchers systematically analyzed **user needs**, designed the interface, implemented system functionalities, and evaluated the effectiveness of features such as **attendee registration, ticket management, event scheduling, database storage, and ticket cancellation**. The GUI design was structured to provide an intuitive and well-organized interface for both administrators and users.

The system's **performance, usability, and efficiency** were assessed descriptively, measuring how well it managed ticketing operations and simplified the registration process compared to manual methods.

System Development Process

The following steps were systematically followed to ensure a reliable and functional system:

Planning – Identify problems in manual ticketing, including errors in attendee counting, lost records, and inefficient registration workflows. Determine the need for Python for programming and SQLite for local database persistence.

Design – Develop **system flowcharts** to visualize data flow from user input to database storage. Design the GUI layout to be intuitive and user-friendly. Establish database schema for tickets, events, and attendees.

Development –

- Code the system in **Python using Object-Oriented Programming (OOP)** for modularity and maintainability.
- Implement **SQL queries** to create tables, insert, update, and retrieve records in SQLite.
- Develop GUI using **Tkinter** for interactive windows, buttons, forms, and tables.

Testing – Conduct functional testing to identify errors and ensure each module performs as intended. Validate input to prevent incomplete or duplicate entries, and verify that all features (Add, View, Update, Delete, Cancel) function correctly.

Implementation – Integrate the GUI with the SQLite database for real-time data management. Ensure system readiness for use and confirm that the final output aligns with project goals.

Maintenance – Continuously monitor system performance, implement updates, and refine functionalities based on user feedback. Maintain data integrity and system reliability over time.

System Design

The architectural flow of the Event Ticketing System is structured around a centralized Model-View-Controller (MVC) approach, separating user interactions from database transactions. The system operates on a cyclic loop of user-driven inputs, visual events, back-end SQL generation, and dynamic display refreshes. This layout prevents data conflicts and ensures that the system state matches the underlying database.

The systematic progression of the application follows a clear pathway from initial loading to operational execution:

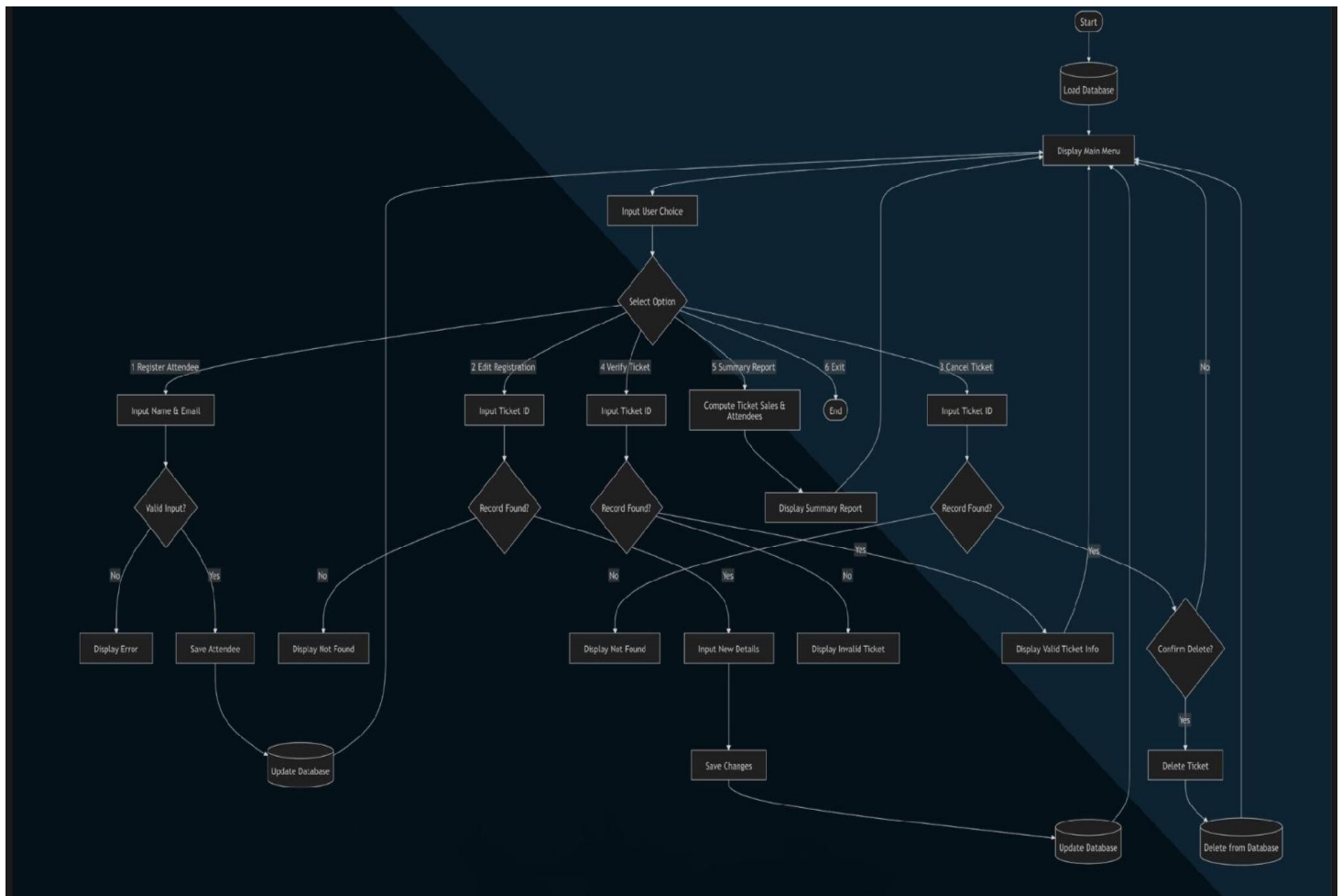


Figure 1. Flowchart of the EventEase System

This figure illustrates the overall process of the EventEase GUI-based Event Registration and Ticketing System. It shows the sequential workflow, including **user login and registration**, event selection, ticket record management, database interactions, and report generation. The flowchart highlights decision points and data validation steps to ensure accurate and reliable attendee management throughout the system.

The system architecture follows a **centralized Model-View-Controller (MVC)** approach, separating user interactions from database operations:

Initialization & Database Handshake – On startup, the backend **TicketingSystem** object connects to tickets.db and runs queries to create necessary tables for tickets and events if they do not exist.

GUI Layout Mapping – The **TicketingGUI controller** renders a persistent navigation sidebar and a dynamic central frame hosting functional interfaces: **DashboardPage, RecordsPage, EventsPage, and SettingsPage**. Modules are cached in memory for quick access.

Event Loop Monitoring – The system listens for user interactions:

- Sidebar clicks swap visible workspace panels and refresh displayed records.
- Button clicks trigger callbacks for operations like Add Record or Delete Event.

Data Verification & Query Execution – Inputs are validated for empty fields or duplicate keys. Valid data is executed as **parameterized SQL transactions** to perform INSERT, UPDATE, or DELETE operations securely.

Dynamic Ledger Syncing – Upon successful database operations, the system displays confirmation alerts, clears input fields, and refreshes tables to reflect the latest state of records.

Tools and Technologies Used

This study utilized different tools and technologies in developing the proposed GUI-based (System Title). The primary programming language used in the development of the system is Python because of its simplicity, flexibility, and effectiveness in creating desktop applications. Python also provides libraries and frameworks that support graphical user interface (GUI) development and database integration, making it suitable for building an efficient and user-friendly system.

For the graphical user interface, Tkinter / CustomTkinter / PyQt was used to create an interactive and visually organized interface that allows users to easily navigate and manage records within the system. In terms of data storage and management, SQLite / MySQL was utilized as the database system to ensure secure, organized, and reliable storage of information. These databases help the system efficiently store, retrieve, and update records whenever needed.

During the development process, tools such as Visual Studio Code (VS Code) / IDLE / Python IDE were used as the main development environments for coding, testing, and debugging the system. These tools provided useful features that assisted in writing and managing the program effectively, contributing to the successful development and functionality of the proposed system.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the **AutoRentPro Car Rental Management System**. It highlights

the system overview, main features, graphical interface, database functionality, and results of functional testing.

System Overview

The primary purpose of the developed Event Ticketing System is to replace inefficient, error-prone manual record-keeping procedures with a structured, automated, and secure management utility. Designed using a dependency-free Python Graphical User Interface (GUI) framework coupled with a centralized relational SQLite database backend, the system serves as a lightweight yet comprehensive software solution tailored for event coordinators. It is engineered to optimize logistical workflows, eliminate paper trail dependencies, ensure data concurrency, and improve user data interaction through immediate visual feedback channels.

To achieve these operational goals, the developed system consolidates core data-handling procedures into four distinct functional modules, accessible via an interactive left navigation taskbar. The foundational mechanics and features embedded within each module are detailed below:

Central Dashboard & Real-Time Registry Ledger: This core utility serves as the main command center. It bridges frontend input entry blocks with the backend database, allowing immediate event ticket processing. Utilizing structural database queries, administrators can perform create, read, update, and delete (CRUD) operations on participant data.

The data is synchronized instantly into a clean tabular data view component (ttk.Treeview).

Database Archive Logs Page (View Records): This module provides an expanded data visualization panel dedicated entirely to database auditing. It functions as a complete data grid that displays all currently active ticket logs stored inside the database schema. This layout isolates large datasets from modification fields, preventing accidental data overwrites while facilitating high-volume visual validation.

Active Event Schedule Manager (Manage Events): Expanding beyond individual participant tracking, this

function handles macroscopic event organization. It allows administrative personnel to dynamically create new events by mapping dedicated Event Keys, Titles, and target Venue Locations directly into a secondary, isolated relational table within the database. It also includes an administrative purge command to permanently drop event records when cancellations occur.

System Parameter Engine (Settings): This view provides localized control over application preferences and database security routines. It includes toggle configurations for user preferences—such as localized high-DPI monitor interface scaling and terminal debug logs. Critically, it features an immediate backend database diagnostic backup trigger, which creates duplicates of the tickets.db schema file to ensure transactional data safety.

By integrating these features into a single, cohesive executable script, the application abstracts complex SQL query structures behind minimalist form widgets. As a result, the system successfully eliminates processing delays, prevents duplicate registration conflicts, and establishes a secure environment for processing event records.

SYSTEM FEATURES

Add Record

Allows administrators to register a new attendee by entering a **Ticket ID and Name**. Input validation ensures all fields are filled and prevents duplicate IDs.

Result: Records are permanently stored in tickets.db with a confirmation message displayed.

View Records

Displays all saved tickets in a **scrollable table**. Administrators can check ticket details, verify statuses, and track event attendance efficiently.

Result: Database records are retrieved accurately and presented in a readable table.

Update Information

Enables modification of existing records using the **Ticket ID** as a reference. The system safely updates the selected row without affecting other entries.

Result: Updated records are reflected in real-time in the database.

Delete / Cancel Ticket

Cancels an attendee's ticket by changing the **status to "CANCELLED"**, preserving the historical data while freeing

the slot.

Result: The cancelled ticket is marked clearly in the table, preventing duplicate registrations.

Clear Fields & Settings

Clears all input fields and allows access to **advanced system controls**, including database mirroring and backup execution.

Result: Input forms are reset, and backup procedures can be triggered securely.

Exit System

Safely terminates the application, closing active database connections and preventing data corruption.

Result: Application closes without errors, maintaining database integrity.

System Interface

Main GUI Window

This serves as the primary dashboard of the application, providing users with quick access to all the core functionalities of the system.

Administrator

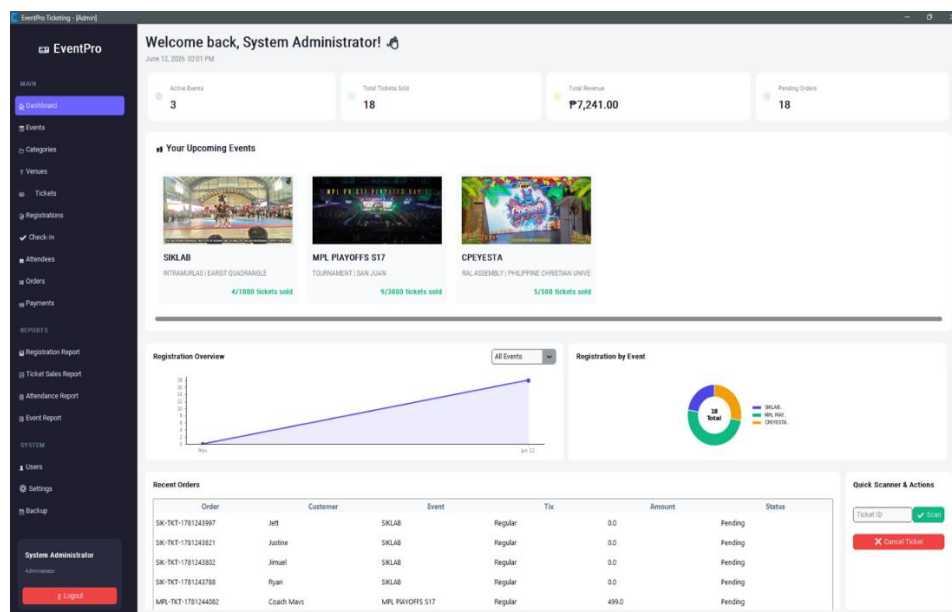


Figure 2. Administrator Dashboard of the EventEase System

This figure shows the Administrator Dashboard of the EventEase System. The interface provides a centralized overview of event management activities, including active events, total tickets sold, revenue generated, pending orders, registration statistics, recent transactions, and quick ticket-scanning functions. The dashboard enables administrators to monitor event performance, manage registrations, and access key system features efficiently through a user-friendly graphical interface.

Customer

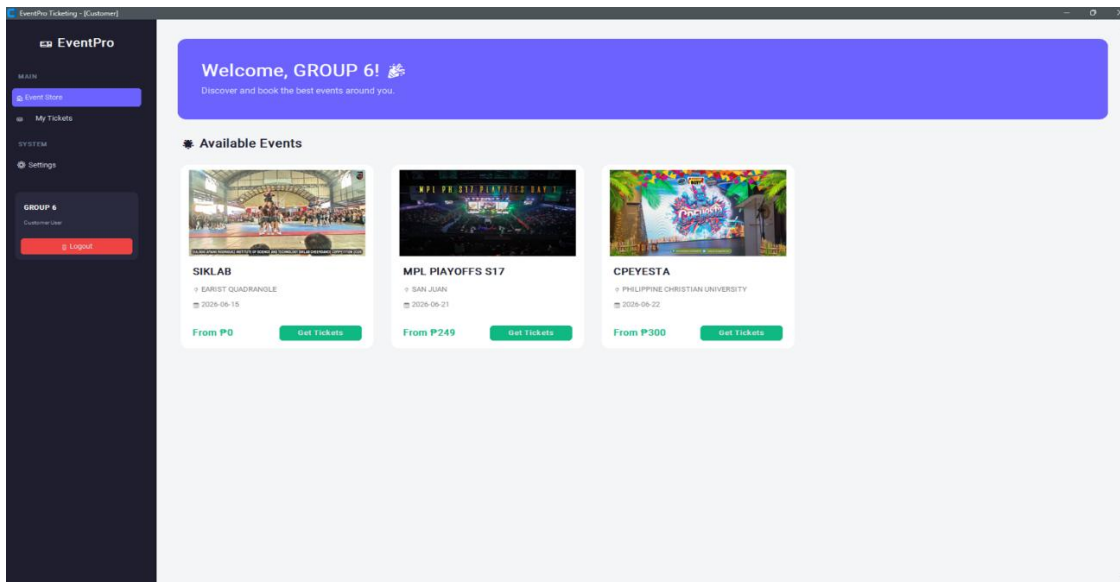


Figure 3. Customer Dashboard of the EventEase System

This figure shows the Customer Dashboard of the EventEase System. The interface provides users with a personalized welcome page and a list of available events, displaying event details such as title, venue, date, and ticket pricing. Customers can conveniently browse upcoming events and purchase tickets through the “Get Tickets” feature. The dashboard is designed to provide a simple, user-friendly, and efficient event booking experience.

Add Record Window: Fields for Ticket ID and Name with validation buttons. This interface displays clear and organized text fields where users can input new data. It includes a submit or add button that directly saves the information into the database.

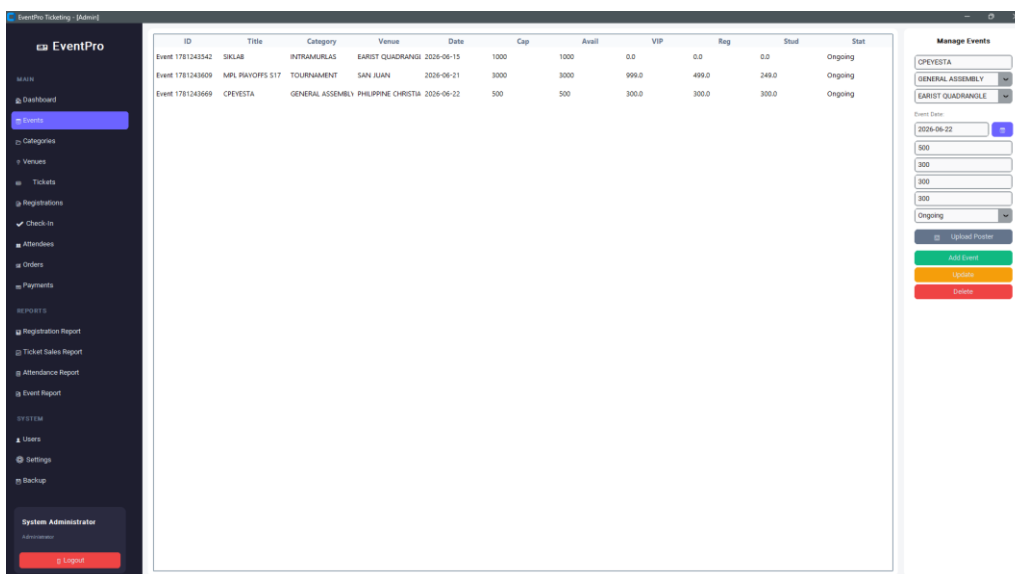
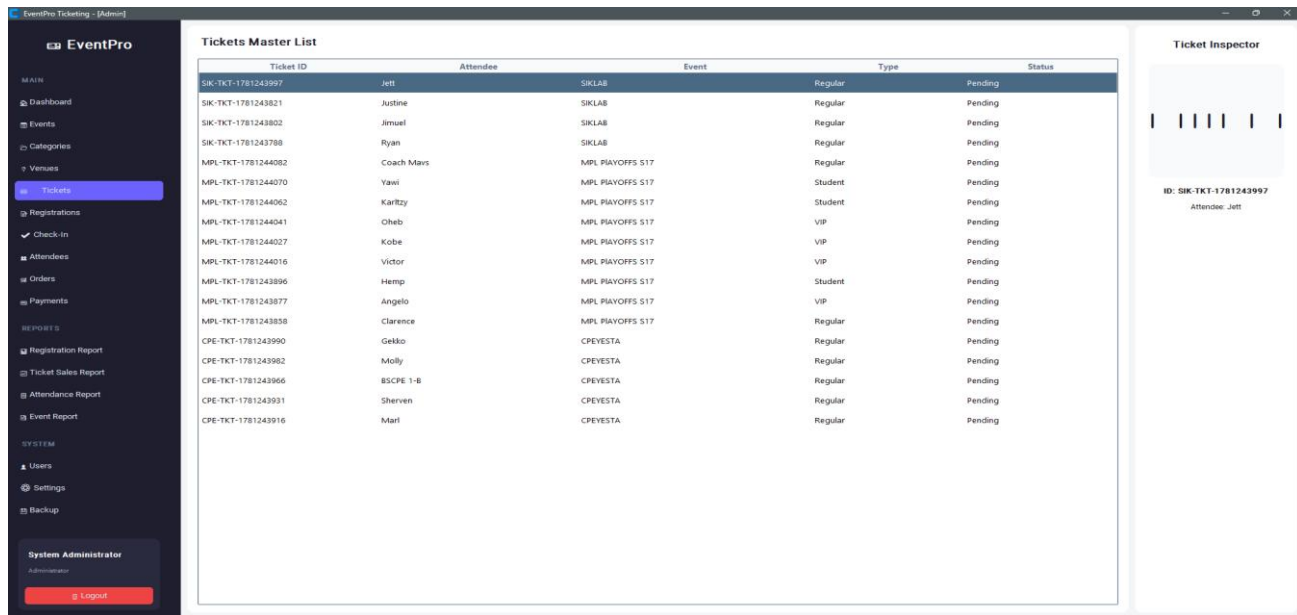


Figure 4. Add Record Window of the EventEase System

This figure displays the Add Record interface where users can enter **Ticket ID** and **Name**. Input validation ensures all fields are completed correctly before submission. The **submit/add button** saves the information directly into the database, enabling efficient and accurate attendee registration.

View Records Window: Tabular display of all tickets for easy monitoring. This section features a structured data table or grid that retrieves and displays all currently stored records from the database for easy monitoring.

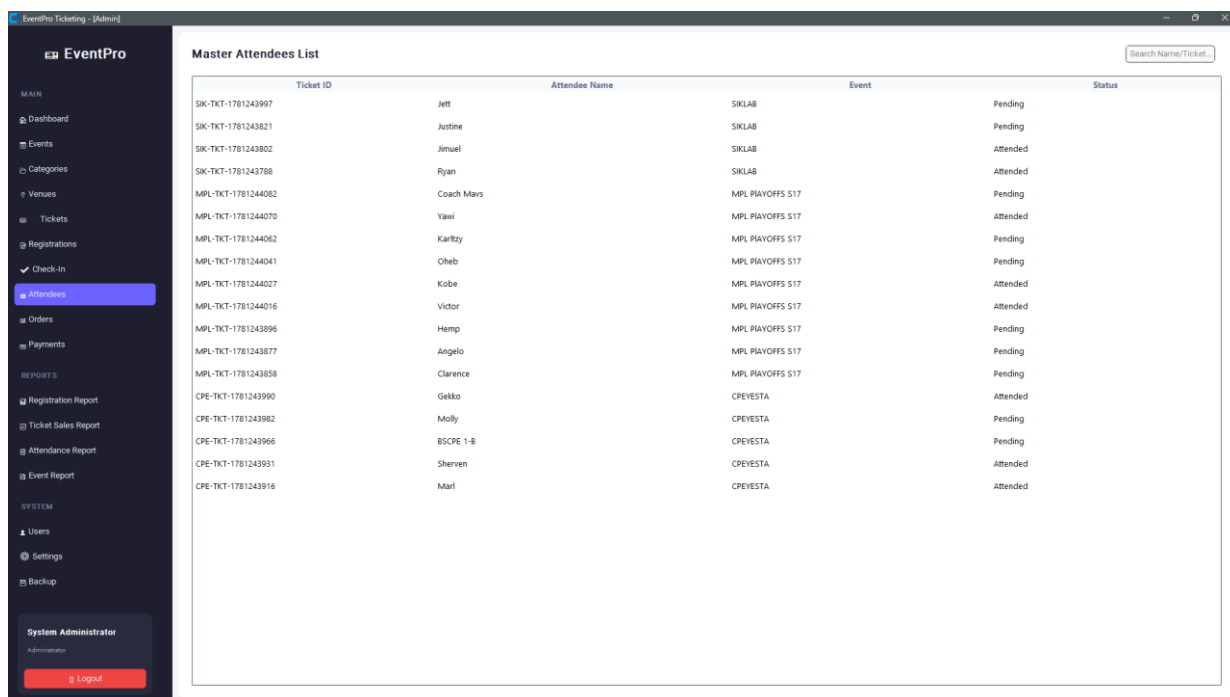


The screenshot shows the 'Tickets Master List' interface of the EventEase system. It features a sidebar with navigation options like Dashboard, Events, Categories, Venues, Tickets (selected), Registrations, Check-in, Attendees, Orders, Payments, Reports, and System. The main area displays a table of ticket records with columns for Ticket ID, Attendee, Event, Type, and Status. A 'Ticket Inspector' panel on the right shows details for a selected ticket.

Ticket ID	Attendee	Event	Type	Status
SIK-TKT-1781243997	Jeff	SIKLAB	Regular	Pending
SIK-TKT-1781243821	Justine	SIKLAB	Regular	Pending
SIK-TKT-1781243802	Jimuel	SIKLAB	Regular	Pending
SIK-TKT-1781243788	Ryan	SIKLAB	Regular	Pending
MPL-TKT-1781244082	Coach Mavs	MPL PLAYOFFS S17	Regular	Pending
MPL-TKT-1781244070	Yawi	MPL PLAYOFFS S17	Student	Pending
MPL-TKT-1781244062	Karlitz	MPL PLAYOFFS S17	Student	Pending
MPL-TKT-1781244041	Oheh	MPL PLAYOFFS S17	VIP	Pending
MPL-TKT-1781244027	Kobe	MPL PLAYOFFS S17	VIP	Pending
MPL-TKT-1781244016	Victor	MPL PLAYOFFS S17	VIP	Pending
MPL-TKT-1781243896	Hemp	MPL PLAYOFFS S17	Student	Pending
MPL-TKT-1781243877	Angelo	MPL PLAYOFFS S17	VIP	Pending
MPL-TKT-1781243858	Clarence	MPL PLAYOFFS S17	Regular	Pending
CPE-TKT-1781243990	Gelko	CPEVESTA	Regular	Pending
CPE-TKT-1781243982	Molly	CPEVESTA	Regular	Pending
CPE-TKT-1781243966	BSCPE 1-8	CPEVESTA	Regular	Pending
CPE-TKT-1781243931	Shervin	CPEVESTA	Regular	Pending
CPE-TKT-1781243916	Marl	CPEVESTA	Regular	Pending

Figure 5. Tickets Master List Interface of the EventEase System

This figure presents the **Tickets Master List** module of EventEase, displaying all registered ticket records in a structured tabular format. The interface includes key ticket information such as **Ticket ID, Attendee Name, Event, Ticket Type, and Status**, allowing administrators to efficiently monitor and manage event registrations. A **Ticket Inspector** panel is provided on the right side, enabling quick viewing and verification of selected ticket details. This feature supports accurate attendee tracking, ticket validation, and streamlined event administration by providing real-time access to registration records.



The screenshot shows the 'Master Attendees List' interface of the EventEase system. It features a sidebar with navigation options similar to Figure 5. The main area displays a table of attendee records with columns for Ticket ID, Attendee Name, Event, and Status. A search bar is located at the top right of the table.

Ticket ID	Attendee Name	Event	Status
SIK-TKT-1781243997	Jeff	SIKLAB	Pending
SIK-TKT-1781243821	Justine	SIKLAB	Pending
SIK-TKT-1781243802	Jimuel	SIKLAB	Attended
SIK-TKT-1781243788	Ryan	SIKLAB	Attended
MPL-TKT-1781244082	Coach Mavs	MPL PLAYOFFS S17	Pending
MPL-TKT-1781244070	Yawi	MPL PLAYOFFS S17	Attended
MPL-TKT-1781244062	Karlitz	MPL PLAYOFFS S17	Pending
MPL-TKT-1781244041	Oheh	MPL PLAYOFFS S17	Pending
MPL-TKT-1781244027	Kobe	MPL PLAYOFFS S17	Attended
MPL-TKT-1781244016	Victor	MPL PLAYOFFS S17	Attended
MPL-TKT-1781243896	Hemp	MPL PLAYOFFS S17	Pending
MPL-TKT-1781243877	Angelo	MPL PLAYOFFS S17	Pending
MPL-TKT-1781243858	Clarence	MPL PLAYOFFS S17	Pending
CPE-TKT-1781243990	Gelko	CPEVESTA	Attended
CPE-TKT-1781243982	Molly	CPEVESTA	Pending
CPE-TKT-1781243966	BSCPE 1-8	CPEVESTA	Pending
CPE-TKT-1781243931	Shervin	CPEVESTA	Attended
CPE-TKT-1781243916	Marl	CPEVESTA	Attended

Figure 6. Registration Records Interface of the EventEase System.

This figure shows the **Registration Records** module of EventEase, which provides a centralized view of all registered attendees and their corresponding event information. The interface displays essential registration details, including **Ticket ID, Attendee Name, Event, and Registration Status**, in a structured tabular format for easy monitoring and management. A search function is available to facilitate quick retrieval of attendee

records. This module enables administrators to efficiently track registration progress, verify attendee participation, monitor attendance status, and maintain accurate event registration records.

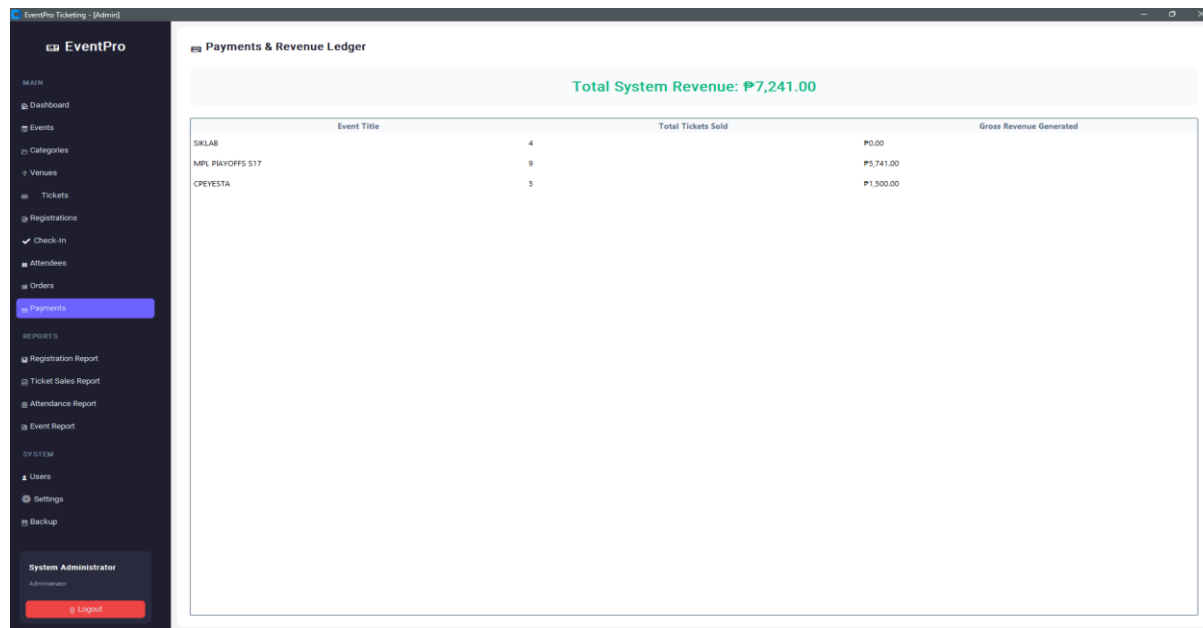


Figure 7. Master Attendees List Interface of the EventEase System.

This figure presents the **Master Attendees List** module of EventEase, which provides a comprehensive overview of all registered event participants. The interface displays attendee information in a structured table, including **Ticket ID, Attendee Name, Event, and Attendance Status**, allowing administrators to efficiently monitor and manage attendee records. A search feature is included to facilitate quick retrieval of attendee information using names or ticket identifiers. This module supports effective attendee tracking, attendance verification, and record management, ensuring accurate and organized event administration.

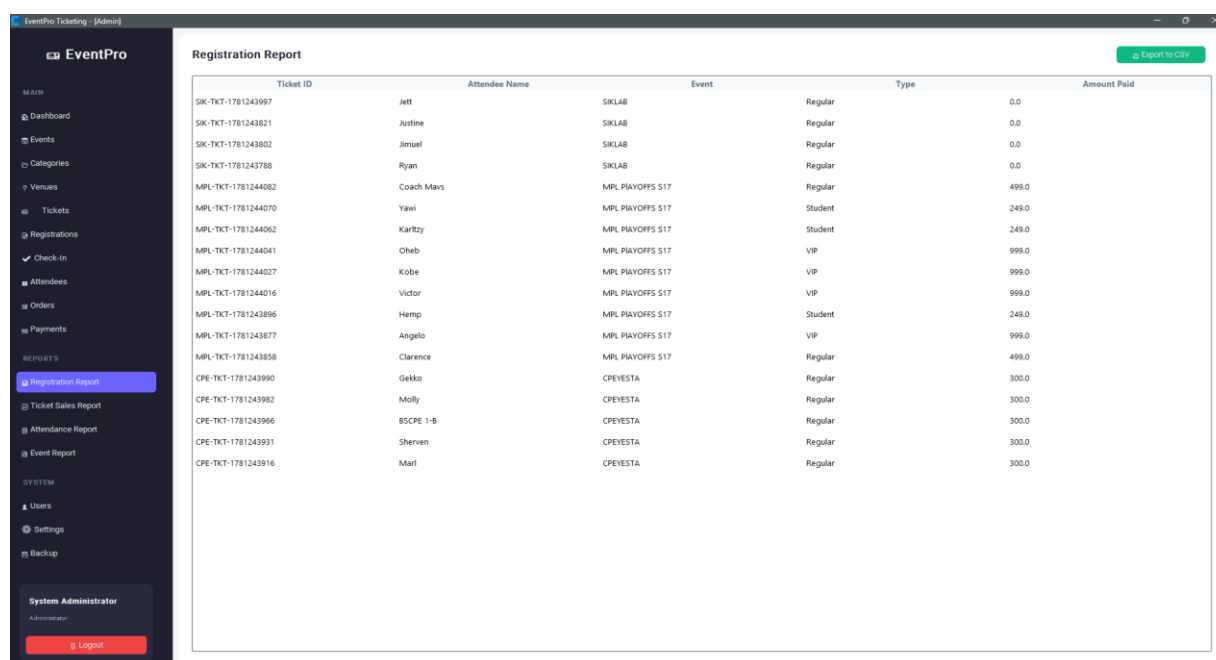
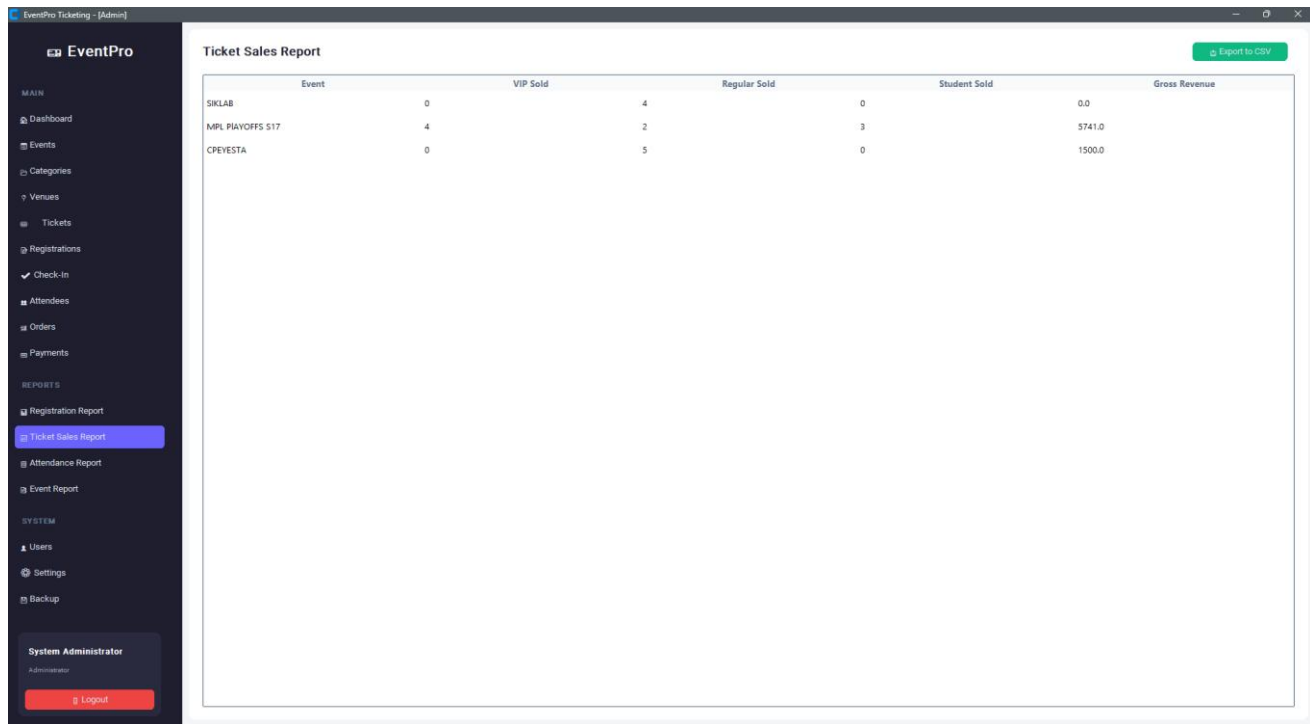


Figure 8. Payments and Revenue Ledger Interface of the EventEase System.

This figure illustrates the **Payments and Revenue Ledger** module of EventEase, which provides a consolidated overview of ticket sales and revenue performance across all registered events. The interface displays key financial metrics, including **Event Title, Total Tickets Sold, Gross Revenue Generated, and Total System**

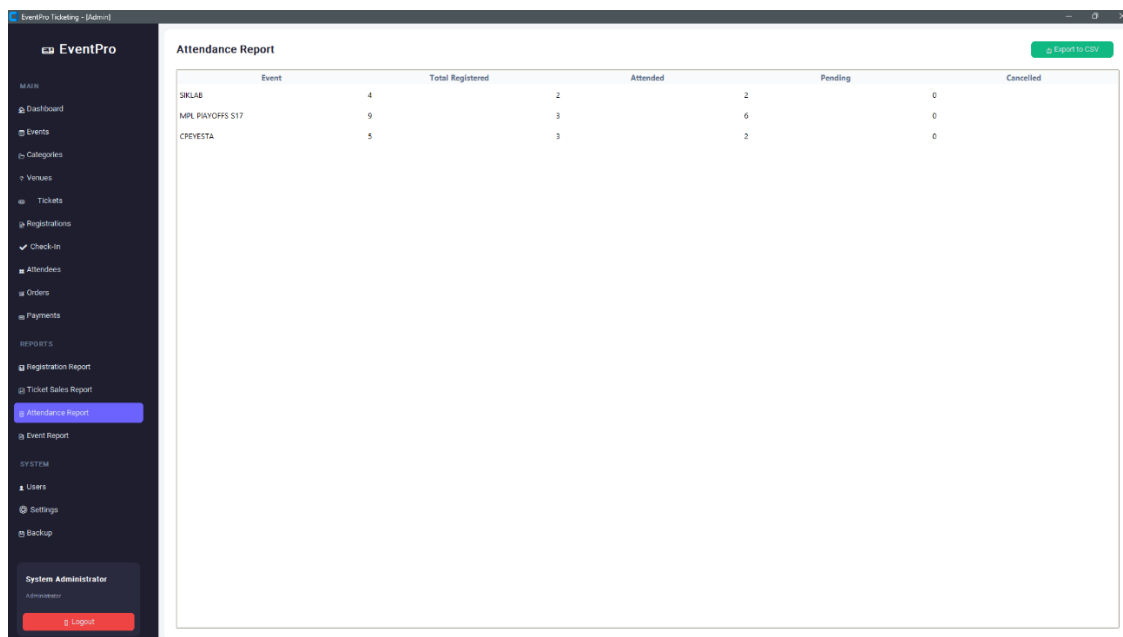
Revenue. By summarizing revenue data in a structured tabular format, the module enables administrators to monitor event earnings, evaluate ticket sales performance, and track overall financial activity in real time. This feature supports effective financial management and reporting, helping organizers make informed decisions regarding event operations and profitability.



Event	VIP Sold	Regular Sold	Student Sold	Gross Revenue
SKILAB	0	4	0	0.0
MPL PLAYOFFS S17	4	2	3	5741.0
CPEVESTA	0	5	0	1500.0

Figure 9. Registration Report Interface of the EventEase System.

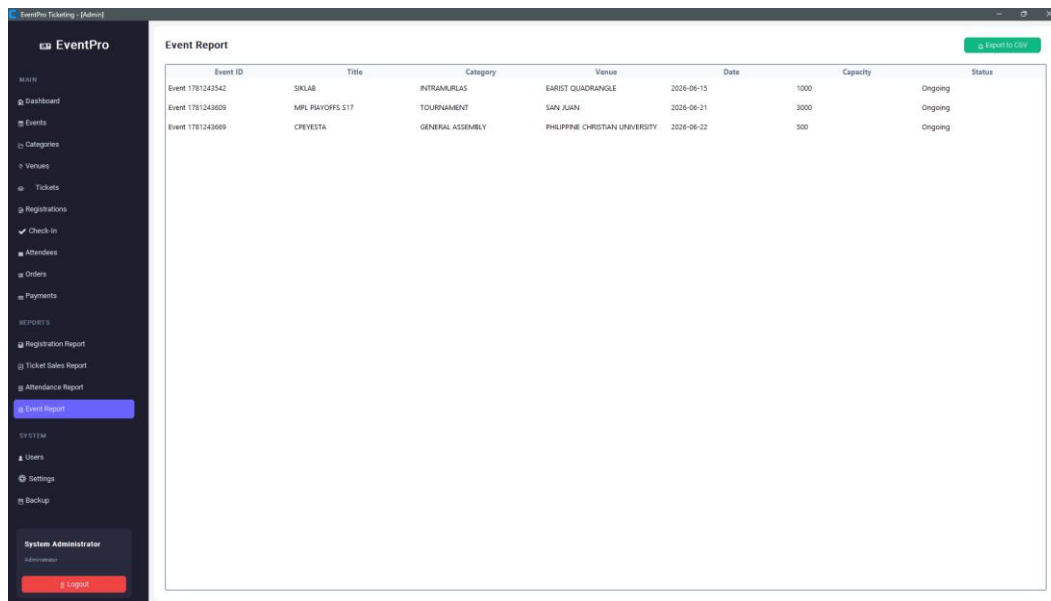
This figure presents the **Registration Report** module of EventEase, which provides a detailed summary of all attendee registrations across different events. The interface displays key registration information, including **Ticket ID, Attendee Name, Event, Ticket Type, and Amount Paid**, in a structured tabular format. An **Export to CSV** feature is available to facilitate report generation, data analysis, and record archiving. This module enables administrators to monitor registration activities, verify payment records, analyze ticket distribution by type, and maintain accurate documentation of event registration transactions.



Event	Total Registered	Attended	Pending	Cancelled
SKILAB	4	2	2	0
MPL PLAYOFFS S17	9	3	6	0
CPEVESTA	5	3	2	0

Figure 10. Ticket Sales Report Interface of the EventEase System.

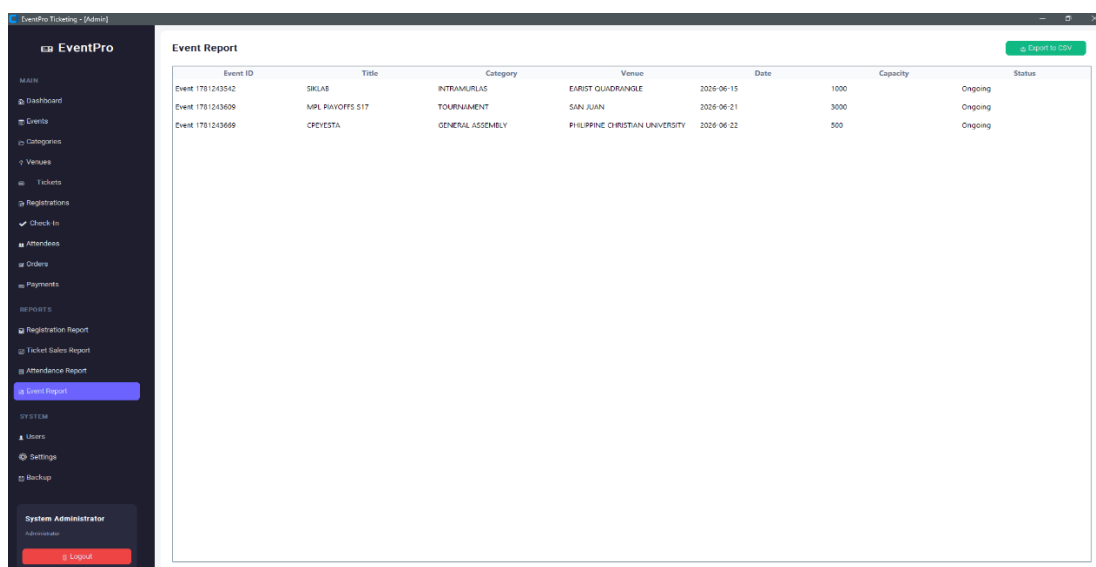
This figure illustrates the **Ticket Sales Report** module of EventEase, which provides a summarized analysis of ticket sales performance for each event. The report displays key metrics, including **Event Name**, **VIP Tickets Sold**, **Regular Tickets Sold**, **Student Tickets Sold**, and **Gross Revenue Generated**. The information is presented in a structured tabular format, allowing administrators to compare ticket sales across events and evaluate revenue contributions from different ticket categories. An **Export to CSV** feature is also provided to facilitate report generation, record keeping, and further financial analysis. This module supports data-driven decision-making by offering a clear overview of ticket sales trends and event profitability.



Event ID	Title	Category	Venue	Date	Capacity	Status
Event 1781243542	SIKLAB	INTRAMURALS	EARIST QUADRANGLE	2026-06-15	1000	Ongoing
Event 1781243609	MPL PLAYOFFS S17	TOURNAMENT	SAN JUAN	2026-06-21	3000	Ongoing
Event 1781243669	CREYESTA	GENERAL ASSEMBLY	PHILIPPINE CHRISTIAN UNIVERSITY	2026-06-22	500	Ongoing

Figure 11. Attendance Report Interface of the EventEase System.

This figure presents the **Attendance Report** module of EventEase, which provides a consolidated summary of attendee participation across all events. The report displays key attendance metrics, including **Event Name**, **Total Registered Attendees**, **Number Attended**, **Pending Attendees**, and **Cancelled Registrations**. The information is organized in a tabular format to enable administrators to efficiently monitor attendance performance and evaluate participant engagement for each event. An **Export to CSV** feature is also available for generating attendance records and supporting further analysis and documentation. This module assists event organizers in tracking attendance trends, measuring event participation rates, and maintaining accurate attendance records for reporting purposes.



Event ID	Title	Category	Venue	Date	Capacity	Status
Event 1781243542	SIKLAB	INTRAMURALS	EARIST QUADRANGLE	2026-06-15	1000	Ongoing
Event 1781243609	MPL PLAYOFFS S17	TOURNAMENT	SAN JUAN	2026-06-21	3000	Ongoing
Event 1781243669	CREYESTA	GENERAL ASSEMBLY	PHILIPPINE CHRISTIAN UNIVERSITY	2026-06-22	500	Ongoing

Figure 12. Event Report Interface of the EventEase System.

This figure illustrates the **Event Report** module of EventEase, which provides a comprehensive summary of all events managed within the system. The report displays essential event information, including **Event ID, Event Title, Category, Venue, Date, Capacity, and Current Status**. The data is presented in a structured tabular format, allowing administrators to efficiently monitor event schedules, venue assignments, participant capacity limits, and event availability. An **Export to CSV** feature is provided to support report generation, documentation, and data analysis. This module assists event organizers in overseeing event operations, tracking event statuses, and maintaining accurate records for planning and management purposes.

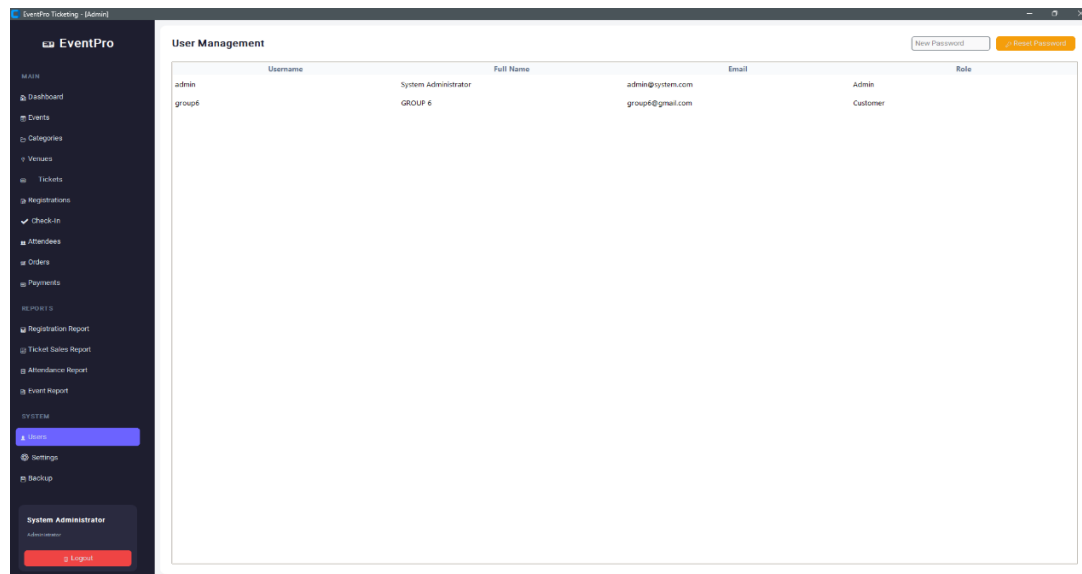


Figure 13. User Management Interface of the EventEase System.

This figure presents the **User Management** module of EventEase, which provides administrators with a centralized interface for managing system user accounts. The module displays essential user information, including **Username, Full Name, Email Address, and Assigned Role**, in a structured tabular format. It supports role-based access control by distinguishing administrative and customer accounts, ensuring appropriate system permissions for different users. Additionally, the interface includes a **password reset** feature that enables administrators to securely manage user credentials. This module enhances system security, user account administration, and access management by maintaining organized and up-to-date user records.

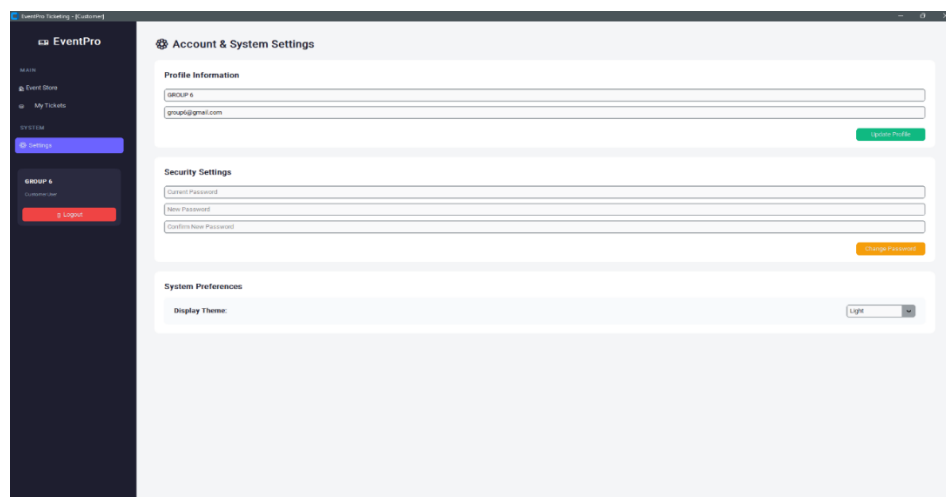


Figure 14. Account and System Settings Interface of the EventEase System.

This figure illustrates the **Account and System Settings** module of EventEase, which enables users to manage their personal account information, security credentials, and application preferences. The interface is divided into three main sections: **Profile Information**, where users can update their account details; **Security Settings**, which allows users to change their passwords by providing current and new credentials; and **System Preferences**, where users can customize application settings such as the display theme. This module enhances user experience by providing convenient account management features while promoting system security through credential maintenance and personalized interface configurations.

Cancel Ticket Window: Confirm cancellation of tickets. This interface provides a mechanism for users to select a specific record and permanently remove it from the system, usually accompanied by a confirmation prompt to prevent accidental deletion.

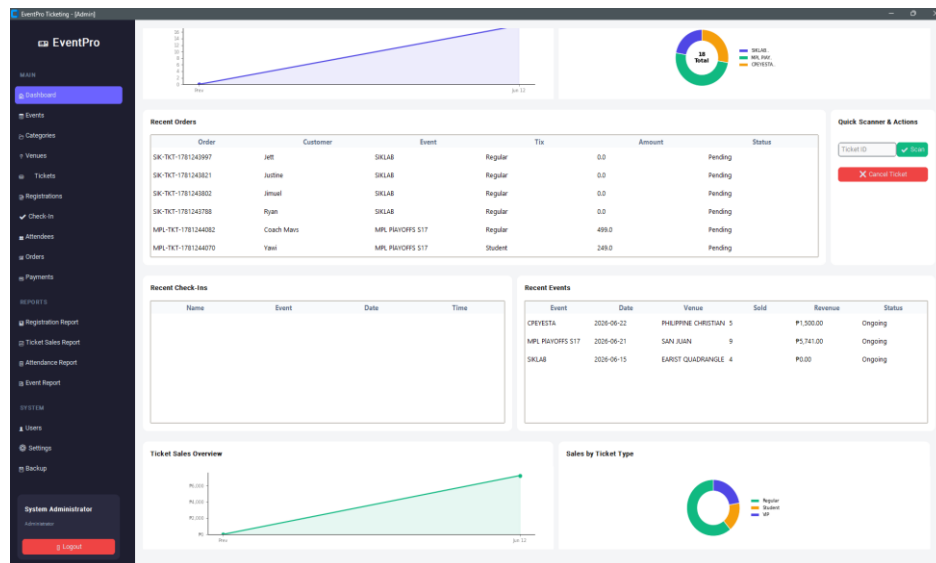


Figure 16. Administrator Dashboard Interface of the EventEase System.

This figure presents the **Administrator Dashboard** of EventEase, which serves as the central monitoring and management hub for event registration and ticketing operations. The dashboard provides real-time visual summaries of system activities through interactive charts, statistical indicators, and data tables. Key components include **Registration Overview**, **Recent Orders**, **Recent Check-Ins**, **Recent Events**, **Ticket Sales Overview**, and **Sales by Ticket Type**, enabling administrators to monitor attendee registrations, ticket sales performance, event status, and revenue generation. Additionally, the **Quick Scanner & Actions** panel allows rapid ticket verification and cancellation, streamlining event management tasks. By consolidating operational, attendance, and financial information into a single interface, the dashboard supports efficient decision-making, performance monitoring, and overall event administration.

System Testing

This section presents the results of the functional testing conducted to evaluate the reliability, accuracy, and performance of the EventEase system. Various test cases were executed across the system's core modules, including attendee registration, ticket management, event scheduling, attendance tracking, report generation, user management, and database operations. The testing process verified that all functions performed according to their intended specifications and produced the expected outputs without critical errors.

The results demonstrated that the system successfully handled **Create, Read, Update, Delete (CRUD)** operations, prevented duplicate ticket registrations, validated user inputs, generated accurate reports, and maintained data integrity throughout all transactions. Security-related functions such as user authentication, password management, and role-based access control also operated correctly. Furthermore, backup and recovery procedures were tested to ensure that event and attendee records could be securely preserved.

Overall, the functional testing confirmed that EventEase is a stable, reliable, and user-friendly event registration and ticketing system capable of supporting administrative operations efficiently while maintaining accurate and secure data management.

Table 1. Functional Testing Results

Test Case	Description	Expected Result	Actual Result	Status
User Login	Valid administrator/customer credentials entered	User gains access to assigned dashboard	Access granted successfully	Passed
User Registration	New customer account created	User account saved in database	Account created successfully	Passed
Add Event	Administrator creates a new event	Event saved and displayed in event list	Event added successfully	Passed
Edit Event	Administrator modifies event details	Changes reflected in database	Event updated successfully	Passed
Delete Event	Administrator removes event	Event removed from records	Event deleted successfully	Passed
Ticket Purchase	Customer purchases event ticket	Ticket generated and stored	Ticket generated successfully	Passed
Duplicate Ticket Validation	Attempt to create duplicate ticket record	System blocks duplicate entry	Duplicate prevented	Passed
Check-In Verification	Ticket scanned and validated	Attendance status updated	Check-in recorded successfully	Passed
Generate Registration Report	Administrator generates report	Report displays accurate registration data	Report generated correctly	Passed
Generate Sales Report	Administrator generates sales report	Sales summary displayed accurately	Report generated correctly	Passed
User Management	Administrator manages user accounts	User information updated successfully	Operation completed successfully	Passed
Password Reset	User changes password	New password saved securely	Password updated successfully	Passed
Database Backup	Administrator initiates backup	Backup file created successfully	Backup completed successfully	Passed

Search Function	Search attendee or ticket record	Matching records displayed	Search results returned correctly	Passed
Logout	User exits system	Session terminated securely	Logout completed successfully	Passed

Summary: All test cases passed successfully, indicating that EventEase meets its functional requirements and provides reliable event registration, ticketing, attendance monitoring, reporting, and administrative management capabilities.

DISCUSSION

The results indicate that **EventEase successfully automates event registration and ticketing**. Key findings include:

- **Efficiency:** Automates tasks, reducing wait times and administrative workload.
- **Accuracy:** Input validation and database constraints prevent duplicates and errors.
- **User-Friendly Interface:** The Tkinter GUI allows staff to navigate the system without technical expertise.
- **Security & Data Integrity:** SQLite database ensures reliable storage, safe updates, and secure record management.

The system demonstrates how Python programming, GUI development, and relational databases can be combined to **modernize event management**, offering an accessible and dependable solution for academic, corporate, and social events.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of **EventEase**, the following conclusions were drawn:

1. **Automated Event Management:** EventEase successfully automates attendee registration, ticket issuance, and record management, replacing inefficient manual processes.
2. **Database Reliability:** Integration with **SQLite** ensures secure, accurate, and organized storage of attendee and event data.
3. **User-Friendly GUI:** The Tkinter-based interface allows administrators to navigate modules easily, perform CRUD operations, and manage events without requiring advanced technical knowledge.
4. **Functional Accuracy:** Functional testing confirmed that all modules—including Add Record, View Records, Update, Delete, and Backup—perform correctly and prevent errors such as duplicate entries or incomplete data submission.
5. **Operational Efficiency:** EventEase improves speed, reduces administrative workload, and provides real-time updates, enhancing the overall efficiency of event management operations.

Overall, the study demonstrates that a **GUI-based Python system with SQLite integration** provides a reliable, efficient, and secure solution for managing event registration and ticketing.

Recommendations

Based on the findings and limitations, the following recommendations are proposed:

1. **User Authentication:** Implement secure login mechanisms with **encrypted passwords** to prevent unauthorized access.
2. **Multi-User Support:** Upgrade the system to allow multiple administrators or staff to use the system concurrently.
3. **Enhanced GUI Design:** Improve interface aesthetics and responsiveness, supporting various screen resolutions and devices.
4. **Automated Reporting:** Integrate export functions to generate reports in **PDF or Excel**, simplifying event analysis and documentation.
5. **Online Payment Integration:** Incorporate payment gateways to allow online ticket purchases.
6. **Event Assignment:** Enable tickets to be directly associated with specific events to improve record accuracy.
7. **Notifications:** Include automated email or SMS notifications for attendees about registration status, event reminders, or cancellations.
8. **Future Research:** Extend the system to include cloud-based databases, online registration portals, and analytics dashboards for improved scalability and data insights.

Implementing these recommendations will enhance EventEase's **functionality, usability, and scalability**, creating a comprehensive, modern, and efficient system for event registration and ticket management.

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.

The researchers also extend their sincere gratitude to the **Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST)**, 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. Al-Mamrah, M. A., & Al-Shorman, A. (2021). Automated event ticketing systems and data security: A review of modern database applications. *Journal of Software Engineering and Applications*, 14(5), 183–195. <https://doi.org/10.4236/jsea.2021.145011>
2. Alfirdaus, M., & Widodo, A. (2020). Analysis of event marketing, registration, and ticketing digitalization. *Journal of Information Systems and Technology*. <https://doi.org/10.1007/978-0-230-34500-3>
3. Chandra, K., et al. (2022). CU-EVENTS: A comprehensive event management system for university. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. <https://www.ijraset.com/>
4. Dunphy, P., & Petitcolas, F. (2022). Exploring the use of self-sovereign identity for event ticketing systems. *Electronic Markets Journal*. <https://link.springer.com/journal/12525>
5. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
6. Mäkelä, J., et al. (2012). Architecture and evaluation of a user-centric NFC-enabled ticketing system for small events. Pearson. <https://www.pearson.com/en-us/subject-catalog/p/fundamentals-of-database-systems/P200000003546/9780137502523>
7. Mobility Infotech. (2025). Car Rental Management Software: Top Key Features, Business Benefits, and Top Solutions. Retrieved May 26, 2026, from <https://www.mobilityinfotech.com/blog/car-rental-management-software-guide>
8. Pangalanan, I. (2024). A Web-based Car Rental Services Management System. Retrieved May 26, 2026, from <https://www.scribd.com/document/826537212/A-Web-Based-Car-Rental-Services-Management-System>
9. Pradana, F., et al. (2021). Responsive and adaptive online-based ticketing service information system. UMSIDA Research Archive. <https://archive.umsida.ac.id/>
10. Rahmawati, D., et al. (2024). Innovative event management platform design with integrated ticketing and networking features. *UNESA E-Journal*. <https://journal.unesa.ac.id/>
11. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). *Systems Analysis and Design in a Changing World* (7th ed.). Cengage Learning. <https://www.cengage.com/c/systems-analysis-and-design-in-a-changing-world-7e-satzinger/9721649283749/>
12. Setiawan, A., & Pramudya, Y. (2023). Rancang antarmuka event management “JoinUs”. *Polinema Journal*. <https://jurnal.polinema.ac.id/>
13. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education. <https://www.mheducation.com/highered/product/database-system-concepts-silberschatz-korth/M9780078022159.html>
14. Python Software Foundation. (2025). SQLite database library documentation (Version 3.x). Python Software Docs. <https://docs.python.org/3/library/sqlite3.html>
15. SQLite Consortium. (2024). About SQLite. SQLite Documentation. <https://www.sqlite.org/about.html>

ABOUT THE AUTHORS

Jan Clarence B. Fetizanan is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). He graduated from Mayamot National High School (MNHS), a public school in Antipolo City, Rizal. Currently, he is focused on studying programming and Computer-Aided Design (CAD), aiming to deepen his technical knowledge to grow both personally and professionally. Ultimately, he aspires to build a successful career as a computer engineer and eventually pursue a second degree in Electronics Engineering, followed by a third degree in Culinary Arts.

Jimuel O. Fortuno is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). He graduated from Saint Matthew College under the Science, Technology, Engineering, and Mathematics (STEM) strand. Currently, his academic focus centers on strengthening his technical competencies in Python programming and SolidWorks 3D computer-aided design (CAD) modeling. Having already built a foundation by studying C++ programming, he aims to

continually expand his skills in both software development and system architecture. Upon graduation, he aspires to build a successful and meaningful career within the engineering field.

Justin Karl B. Galam a graduate of the Science, Technology, Engineering, and Mathematics (STEM) strand from Silangan National High School (SNHS) where he's group was awarded Best in Innovation, is currently pursuing a Bachelor of Science in Computer Engineering (BSCpE) at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) in Manila, Philippines. Dedicated to academic leadership and institutional involvement, he serves as a Volunteer Student Assistant (VSA) and as the Publication Material Officer 2 for the Association of

Computer Engineering Students (ACES). His current academic focus and research interests center on software-hardware integration, programming, computer-aided design, and automated systems architecture. To support these pursuits, he leverages technical competencies in Python and C++ across environments including Dev C++ and PyCharm, alongside prior experience utilizing SolidWorks, Tinkercad, and embedded Arduino IDE to design his award-winning automated irrigation prototype. Upon completion of his degree, he intends to transition into the engineering industry with the long-term goal of contributing successfully to technological innovation and professional advancement within the computer engineering field.

John Angelo C. Garbin is a first-year Bachelor of Science in Computer Engineering student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). He graduated from the STEM strand at Notre Dame of Greater Manila and attended Caloocan High School. He is currently focused on learning about hardware and software systems, He also has various knowledge in coding and 3D Computer Aided Modeling such as Dev C++, Python, and SolidWorks. He has also established a simple prototype using simple solar power and electrical systems as a charging station. His goal is to build a successful career as a Computer Engineer.

Ryan Zedrick Hermosilla is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). He graduated from the South East - Asia Institute of Trade and Technology under the Accountancy, Business, and Management (ABM) strand. Transitioning his organizational and business-oriented skills into the tech field, As a dedicated Volunteer Student Assistant (VSA), he is responsible for maintaining the functionality and cleanliness of the university's computer laboratories. Through this role, he performs basic PC troubleshooting and continuously hones his practical, hands-on technical skills under the mentorship of senior engineering students. Although he has established a foundation in 3D computer-aided design using SolidWorks, his primary passion lies in software development. He is actively working to deepen his programming competencies and system logic skills, with the ultimate goal of leveraging his well-rounded technical background to build a successful career focused on software engineering.

Hempbahaghari M. Nono is a first-year Bachelor of Science in Computer Engineering (BSCpE) student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). He graduated from the National Teachers College (NTC), a private school on Mendiola Street, Manila. Currently, he is focused on expanding his knowledge and skills in various areas of technology, particularly programming and Computer-Aided Design (CAD). Through his studies in programming, he is learning to design, develop, and troubleshoot software applications using different languages. Additionally, he is gaining experience in CAD, enabling him to create the technical drawings, models, and engineering designs essential for modern technology projects.

Engr. Meshelle N. Fabro is a licensed Professional Computer Engineer and educator at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). She is currently pursuing a Master of Science in Computer Engineering at Bulacan State University and has professional industry experience as a Technical Engineer at Hewlett-Packard (HP) and an IT Engineer at IBM. She is also a TESDA CSS NC II and Trainers Methodology Level I (TM1) holder, demonstrating her commitment to technical excellence and competency-based education. Passionate about teaching and innovation, she is dedicated to guiding Computer Engineering students in becoming industry-ready, competent, and highly skilled individuals equipped for the evolving demands of the technology industry.