

AutoRentPro: Design and Implementation of a GUI-Based Python Car Rental Management System with SQLite Integration

Doploso, John Abel S., Hombrebueno, Jonathan D., Honrado, Franz Gabriel G., Magalona, Aron Luis T., Montero, James Adrian D., Aldepolla, Patrick Sean H., Cariaga, John Kenth S., 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.2026.150600089>

Received: 10 June 2026; Accepted: 15 June 2026; Published: 07 July 2026

ABSTRACT

The management of car rental operations, including vehicle inventory, customer reservations, and rental transactions, is often prone to errors when conducted manually. Delays in booking, inaccurate record-keeping, and inefficient tracking of vehicle availability can lead to customer dissatisfaction and operational inefficiency. This study presents **AutoRentPro**, a **GUI-based Python Car Rental Management System** integrated with an **SQLite database** to automate rental processes and enhance data accuracy. The system allows administrators to manage vehicle information, process customer bookings, track rental transactions, and generate reports on vehicle usage and availability. Object-Oriented Programming (OOP) principles were applied to ensure modularity, scalability, and maintainability of the code. Functional testing confirmed that the system successfully handles vehicle management, customer registration, booking processing, and report generation, while the GUI facilitates an intuitive user experience. AutoRentPro improves efficiency, reduces human error, and provides a reliable platform for managing car rental operations in small to medium-sized enterprises. Limitations include the absence of online payment integration and fleet telematics, which may be considered for future enhancement.

Keywords: AutoRentPro, Car Rental System, Python, SQLite, GUI, Object-Oriented Programming, Vehicle Management, Reservation System

INTRODUCTION

Background of the Study

The advancement of technology has significantly transformed business operations, including the car rental industry. Traditional manual car rental processes often result in delays, errors, and difficulties in managing rental records and vehicle availability. These inefficiencies can lead to customer dissatisfaction, missed bookings, and mismanagement of vehicle resources. To address these challenges, businesses increasingly adopt **computerized management systems**, which provide faster, more accurate, and more convenient services (Mobility Infotech, 2025).

The proposed **Car Rental Management System**, titled **AutoRentPro**, is a **GUI-based application** developed using **Python, Tkinter, and SQLite**. It is designed to streamline rental operations by allowing users to efficiently add, view, update, delete, and search records while ensuring accurate data management through **input validation** (Pangalinan, I., 2024). In addition, the system introduces a feature enabling car owners to partner with rental organizations by offering their vehicles for rent on **daily, weekly, or monthly terms**, benefiting both the rental organization and vehicle owners through improved service and additional income opportunities (Mobility Infotech, 2025).

AutoRentPro enhances convenience by providing **real-time vehicle information** and an organized, user-friendly interface for rental management. The importance of computerized systems became particularly evident during the COVID-19 pandemic, as businesses shifted to digital platforms for safer, more efficient, and contactless transactions. Through this study, the researchers aim to develop a **reliable and efficient car rental management system** that modernizes and improves the overall rental process while ensuring accurate and secure record-keeping.

Statement of the Problem

The primary objective of this study is to develop a Car Rental Management System that improves the process of managing rental records through automation. Specifically, the study seeks to answer the following questions:

1. How does the system perform in managing car rental records with respect to:
 - 1.1 Adding new rental records
 - 1.2 Viewing existing records
 - 1.3 Updating records
 - 1.4 Deleting records
2. How effective is the system in handling user input validation, particularly:
 - 2.1 Detecting incomplete inputs
 - 2.2 Detecting invalid inputs (e.g., non-numeric contact numbers)
 - 2.3 Preventing duplicate ticket IDs
3. How efficient is the system in **retrieving and searching rental records** based on:
 - 3.1 Ticket ID search function
 - 3.2 Accuracy of displayed data
4. How usable is the system in terms of:
 - 4.1 Ease of use of the interface
 - 4.2 User-friendliness of the system design

Objectives of the Study

General Objective: To develop a **GUI-based Car Rental Management System** using Python and SQLite databases that automates and improves car rental operations.

Specific Objectives:

1. To design the Car Rental Management System using a flowchart and system planning.
2. To develop the system using Python programming language.
3. To create a **graphical user interface (GUI)** using Tkinter.
4. To integrate an **SQLite database** for storing rental records and renter information.
5. To implement functionalities such as **adding, viewing, updating, deleting, and searching records**.
6. To validate user input to ensure accurate and reliable record management.
7. To test the functionality and usability of the system to ensure it meets operational requirements.

Significance of the Study

This study demonstrates how **technology and computerized systems** can improve the **efficiency and convenience of car rental transactions**. The AutoRentPro system provides a more organized and accessible way of managing vehicles, customer data, and rental transactions compared to traditional manual methods.

The study may benefit the following groups:

- **Organization / Car Rental Companies:** AutoRentPro helps manage rental records efficiently by enabling easy storage, updating, searching, and deletion of customer and vehicle information, reducing manual work and minimizing errors.
- **Students / Developers:** The system serves as a reference for learning system development using Python, Tkinter, and SQLite, demonstrating the integration of GUI and database technologies.
- **Drivers / Renters:** Ensures rental information is properly recorded and managed, making transactions more organized and reducing misunderstandings.
- **Future Researchers:** Provides guidance and reference for those developing or improving similar car rental management systems or database-driven applications.

Scope and Limitations

Scope:

- The system is developed using the **Python programming language**.
- **SQLite** is used as the database to store rental records.
- The system includes a **GUI-based application** developed with Tkinter.
- It manages **car rental information and renter details**.
- Functions include:
 - Add Record
 - View Record
 - Update Record
 - Delete Record
 - Search Record

Limitations:

- The system allows **only one contact number per renter**.
- **Online booking and payment features** are not included in the current implementation.

- Delivery tracking, automated email notifications, and advanced security protocols are excluded but can be considered for future versions.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter discusses the relevant theories, previous studies, and literature that provide a foundation for the development of the **AutoRentPro Car Rental Management System**. The review covers car rental management systems, database integration, Object-Oriented Programming (OOP), graphical user interfaces (GUI), and related system development practices.

Car Rental Management Systems

Car rental systems are digital platforms that allow businesses to manage vehicle inventory, bookings, customer information, and rental transactions efficiently. Traditional manual methods often result in lost records, inaccurate bookings, and delayed processing, which reduce customer satisfaction and operational efficiency (Mobility Infotech, 2025).

Automated car rental management systems improve:

- **Vehicle tracking and availability**
- **Customer registration and management**
- **Rental transaction accuracy**
- **Report generation for revenue and fleet usage**

Studies show that computerized car rental systems enhance operational efficiency and reduce errors, providing better services to both customers and operators (Pangalanan, 2024).

Database Integration

A robust **database system** is critical for maintaining accurate vehicle and customer records. **SQLite** is commonly used for small- to medium-scale applications because it is lightweight, file-based, and easily integrates with Python. Database integration enables:

- Persistent storage of vehicle and customer records
- Efficient CRUD (Create, Read, Update, Delete) operations
- Real-time updates for rentals and vehicle availability
- Reliable report generation for decision-making

Kumar and Singh (2021) emphasize that relational databases, such as SQLite or MySQL, ensure data consistency, security, and accessibility for business management systems.

Object-Oriented Programming (OOP)

OOP organizes code into **classes and objects**, making it modular, reusable, and easier to maintain. In the development of AutoRentPro:

- **Classes** represent entities such as Vehicle, Customer, or RentalTransaction
- **Methods** handle operations like adding vehicles, processing bookings, and generating reports
- OOP allows for scalability, future enhancements, and structured code management

Bhavani et al. (2025) highlight that OOP principles increase functionality and reduce complexity in small business applications, ensuring maintainable and scalable system design.

Graphical User Interface (GUI)

A user-friendly **GUI** improves system usability, especially for non-technical users. **Tkinter** in Python allows developers to create interactive forms, buttons, tables, and menus for a seamless user experience.

GUI design benefits include:

- Easy navigation between modules for vehicle management, booking, and reporting
- Visual feedback for operations like adding, updating, or deleting records
- Improved user satisfaction and reduced training requirements

Shneiderman et al. (2016) emphasize that well-designed GUIs increase efficiency and reduce errors in business management systems.

Related Studies

1. **Mobility Infotech (2025):** Computerized car rental systems improve vehicle and transaction management, allowing efficient record-keeping and customer service.
2. **Pangalinan (2024):** Emphasized Python and SQLite integration for building GUI-based management systems that automate booking, vehicle tracking, and reporting.
3. **Drazen et al. (1982):** Demonstrated the benefits of automation in organizational record-keeping and transaction accuracy, applicable to small business management systems.
4. **Eu-Opensci (2026):** Web-based e-commerce and rental systems require database integration and GUI design for operational efficiency.
5. **Shneiderman et al. (2016):** Highlighted the importance of GUI design in improving usability and efficiency in computerized management systems.

Synthesis

The reviewed literature indicates that:

- **Manual car rental processes** are inefficient and prone to errors.
- **Python and OOP** provide modularity and maintainability for business management systems.
- **Database integration (SQLite/MySQL)** ensures persistent, reliable storage of vehicles, customers, and transactions.

- **GUI design** improves usability, reduces errors, and facilitates efficient navigation.

The **AutoRentPro Car Rental Management System** applies these principles by integrating vehicle management, customer bookings, rental transactions, and reporting into a Python-based OOP system with SQLite database and Tkinter GUI, providing an efficient, reliable, and user-friendly platform for small- to medium-sized car rental businesses.

METHODOLOGY

Research Design

This study will employ a developmental research design since the main objective is to design, develop, and test a functional Car Rental Management System. This type of research focuses on creating a system and improving it through testing until it performs its intended functions correctly. The study involves the development of a computerized system using Python programming language, Tkinter for the graphical user interface, and SQLite for database management.

The system was tested to determine its functionality in handling car rental records such as adding, viewing, updating, deleting, and searching data. The development process included checking the system's accuracy, reliability, and usability to ensure that it works properly.

System Development Process

The researchers used the System Development Life Cycle (SDLC) in developing the GUI-Based Car Rental System. This process served as a guide in creating the system from planning up to implementation. Through this method, the researchers were able to organize the development of the project properly and make sure that the system would function effectively.

Planning - The researchers first identified the problems experienced in the manual management of car rental records. Since recording information manually can lead to errors and lost data, the researchers decided to create a system that would make the process faster, easier, and more organized.

Design - The system was tested several times to check if all features were functioning correctly. The researchers identified errors and fixed them immediately to improve the performance and reliability of the system. Testing also helped ensure that the data entered into the system was stored and displayed accurately.

Development - During this phase, the actual coding and development of the system were done. The researchers used the Python programming language with Tkinter for the GUI and SQLite for the database. Different functions of the system were created and connected to ensure that the system would operate properly.

Testing - The system was tested several times to check if all features were functioning correctly. The researchers identified errors and fixed them immediately to improve the performance and reliability of the system. Testing also helped ensure that the data entered into the system was stored and displayed accurately.

Implementation - Once the system was completed and tested, it was presented and implemented. The researchers demonstrated how the system works and explained its different features. The system helped make the management of rental records more efficient and organized compared to the manual process.

System Flowchart / Design

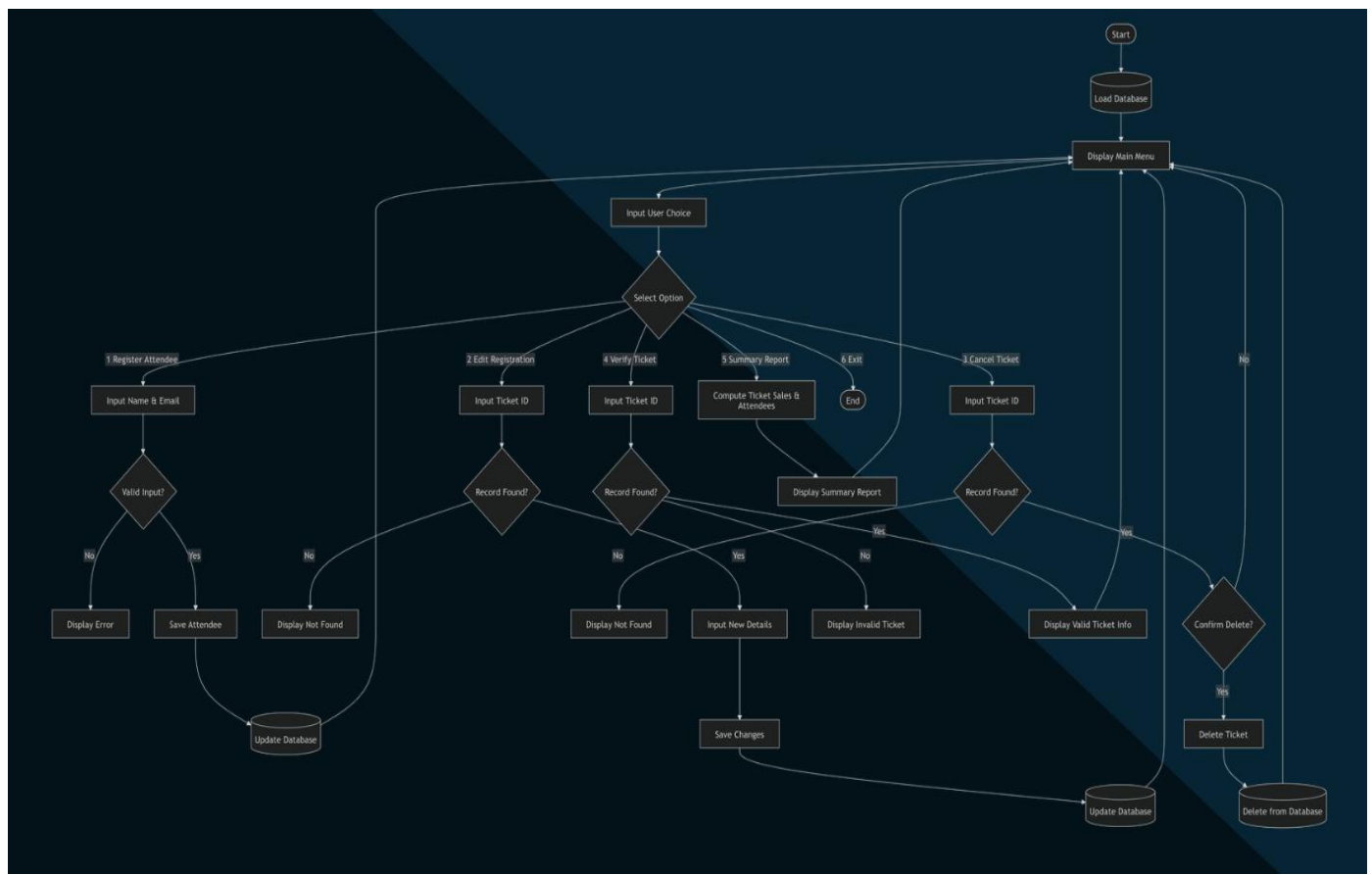


Figure 1. Overall System Flowchart of the AutoRentPro Car Rental Management System

This figure illustrates the full process flow of the AutoRentPro system. It starts with loading the database and displaying the main menu, then proceeds to modules for registering attendance, editing reservations, verifying tickets, generating summary reports, and canceling bookings. Decision points and validations are included to ensure accurate record management, reliable data updates, and proper transaction handling within the car rental management system.

Add Car: Prompts the user to enter new vehicle details. It loops back if the inputs are invalid; otherwise, it saves and confirms the new entry before returning to the Main Menu.

Edit Car Record: The system searches for the specific car record. If found and validated, the user updates the details, saves the changes, and returns to the Main Menu. If invalid, it kicks them back to the menu.

Delete Car Record: The system pulls up the record and asks for confirmation to delete. If confirmed (YES), deletes the record, saves the changes, and goes back to the Main Menu. If cancelled (NO), it simply returns to the menu.

Search Car: Allows the user to input a car query, displays the search results, and then returns to the Main Menu.

Report: Generates and displays a summary report of the car rental data, then routes back to the Main Menu.

Exit: Safely saves all current session data to the database and cleanly terminates the program (End).

Tools and Technologies Used

Python – Python was used as the main programming language because of its simplicity, efficiency, and strong

support for database integration and GUI development. It is also beginner-friendly, making it suitable for developing the Car Rental Management System.

Tkinter – Tkinter was utilized for creating the graphical user interface (GUI) of the system. Since it is built into Python, it allows the development of simple, functional, and user-friendly interfaces.

SQLite – SQLite was used as the system's database because it is lightweight, easy to set up, and effective for storing and managing small to medium-sized data such as customer and rental records.

Python IDE – A Python Integrated Development Environment (IDE) was used to provide a convenient platform for writing, testing, and debugging code, which helped make the development process faster and more organized.

This methodology ensures that AutoRentPro is a **robust, reliable, and user-friendly car rental management system**, combining Python programming, OOP, GUI design, and database integration to automate and streamline vehicle and customer management operations.

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

AutoRentPro is a desktop-based application developed using **Python, Tkinter, and SQLite**. The system automates car rental operations including:

- Vehicle registration and management
- Customer registration and management
- Booking and rental transactions
- Report generation for vehicle usage and revenue

The system is designed for **administrators, rental staff, and customers**, providing a user-friendly GUI to improve usability and efficiency.

This chapter explains the developed Car Rental Management System and how it works. The system was made to help organize and manage car rental records using Python, Tkinter, and SQLite. It allows the user to add new rental details, view all saved records, update information, delete unwanted data, and search for specific ticket IDs in a simple way.

The system runs through a graphical user interface where users can easily enter and manage data without complicated steps. All records are saved in a database so the information is stored properly and can still be accessed anytime. Overall, the system was created to replace manual recording and make the process of managing car rentals faster, easier, and more accurate.

System Features

Add Record. Feature allows the user to add new rental information into the database. The user needs to enter the ticket ID, car model, renter name, and contact number. The system checks if the fields are complete and if the ticket ID is already used.

View Records. Feature shows all saved rental records in a table. This helps the user easily see and monitor all customer information stored in the system.

Update Record. Feature allows the user to edit or change existing rental information. The user selects a record from the table and updates the information when needed.

Delete Record. Feature allows the user to remove a selected record from the database. The system asks for confirmation first before deleting the record.

Clear Fields. Feature clears all the text fields in the form. This helps the user quickly remove current inputs and enter new data.

Exit System. Feature allows the user to close the application safely. It helps properly end the program when the user is done using the system.

Vehicle Management Module

Administrators can manage vehicle information including:

- Vehicle ID, type, model, rental price, and availability
- Adding, updating, deleting, and viewing vehicle records

Result: Vehicle records are correctly stored, updated, and deleted in the SQLite database.

Customer Management Module

Allows administrators to register and manage customer details:

- Customer ID, name, contact, and rental history
- Functions include adding, editing, deleting, and searching customer records

Result: Customer information is accurately maintained, and search functions return correct results.

Booking and Rental Module

Customers can reserve vehicles for rent:

- Record transactions including vehicle, rental duration, and total cost
- Update vehicle availability automatically
- Generate receipts summarizing the transaction

Result: Transactions are recorded accurately, vehicle status is updated in real-time, and receipts are generated correctly.

Reports and Dashboard

Administrators can generate reports for:

- Total vehicles and availability

- Total rentals and revenue
- Customer activity and rental history

Result: Reports reflect the actual database records, allowing quick analysis of rental operations.

System Interface

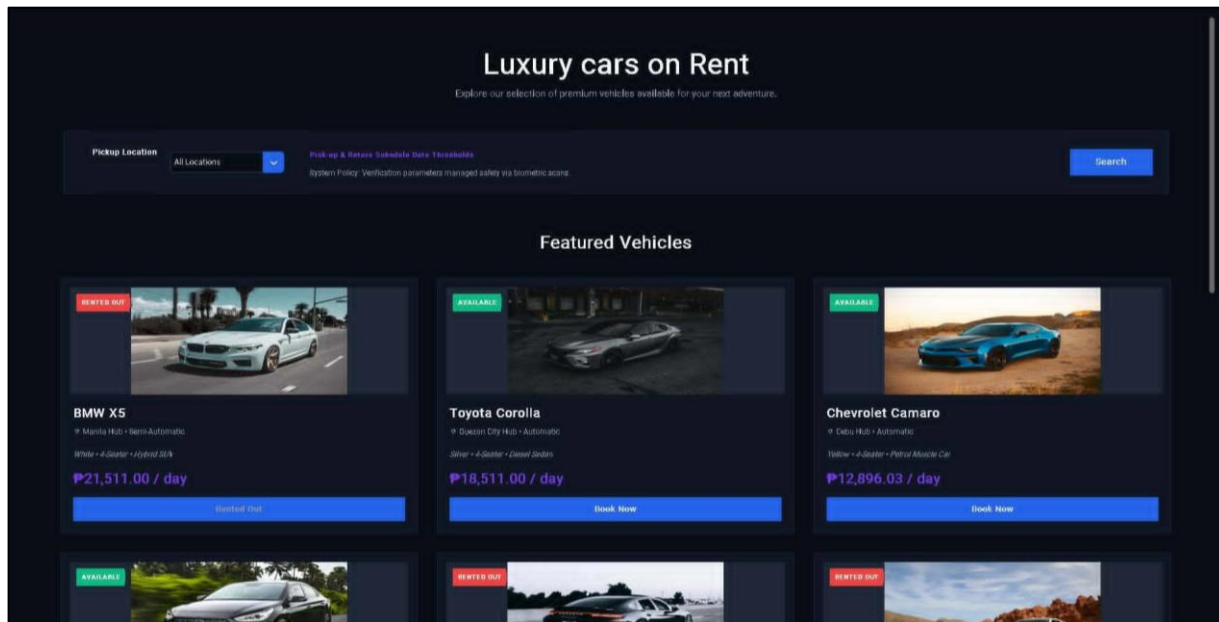


Figure 2. AutoRentPro Car Rental Management System Main Interface

This figure shows the main interface of the AutoRentPro system. Users can **add, view, update, delete, search, or clear car rental records** using the input fields for car model, renter name, contact, and ticket ID. The table displays existing rental records, allowing administrators to efficiently manage and track car rental transactions.

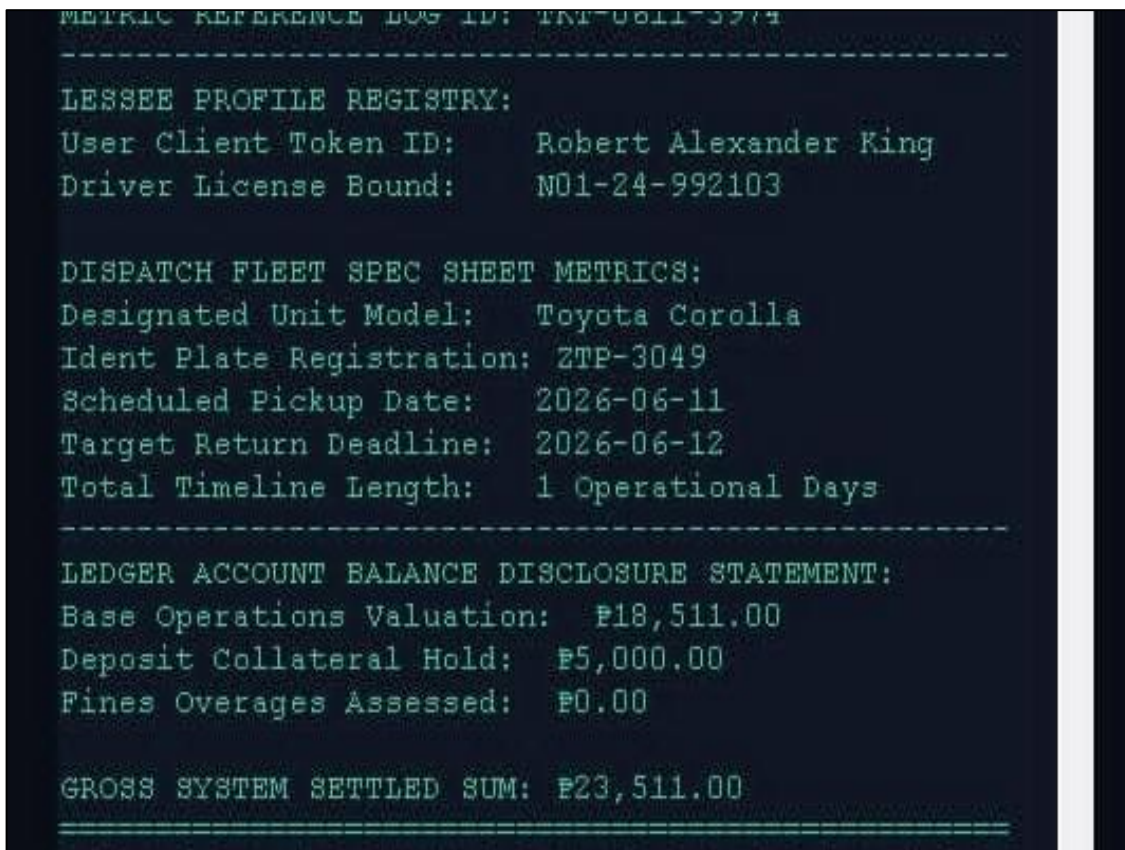


Figure 3. Input Validation Error in AutoRentPro Car Rental Management System

This figure illustrates the system's input validation feature. When a user enters an invalid car model or incorrect details, an **error message** is displayed to prevent incorrect data from being saved. This ensures the integrity of the rental records and maintains accurate management of vehicle and customer information.

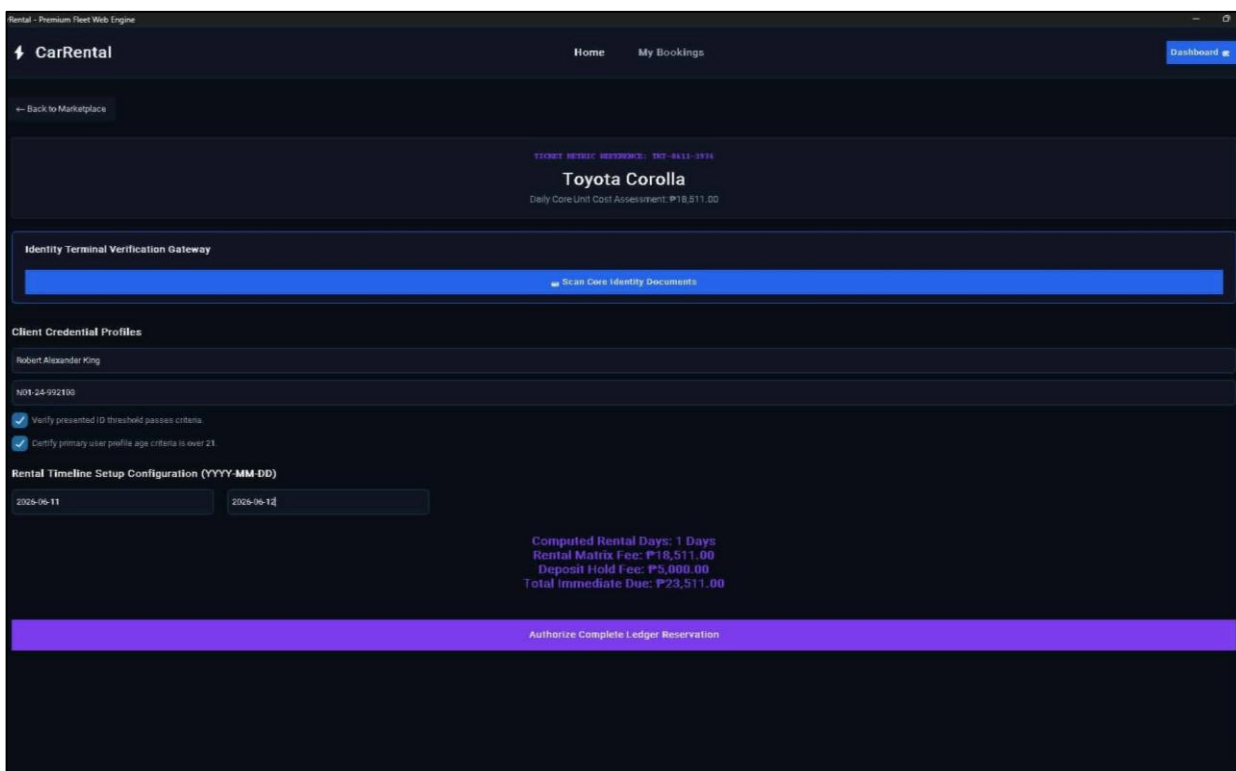


Figure 4. Successful Record Addition in AutoRentPro Car Rental Management System

This figure shows a successful transaction where a new car rental record has been added to the system. The confirmation message ensures that the data has been correctly saved to the database, providing immediate feedback to the administrator and maintaining accurate rental records.

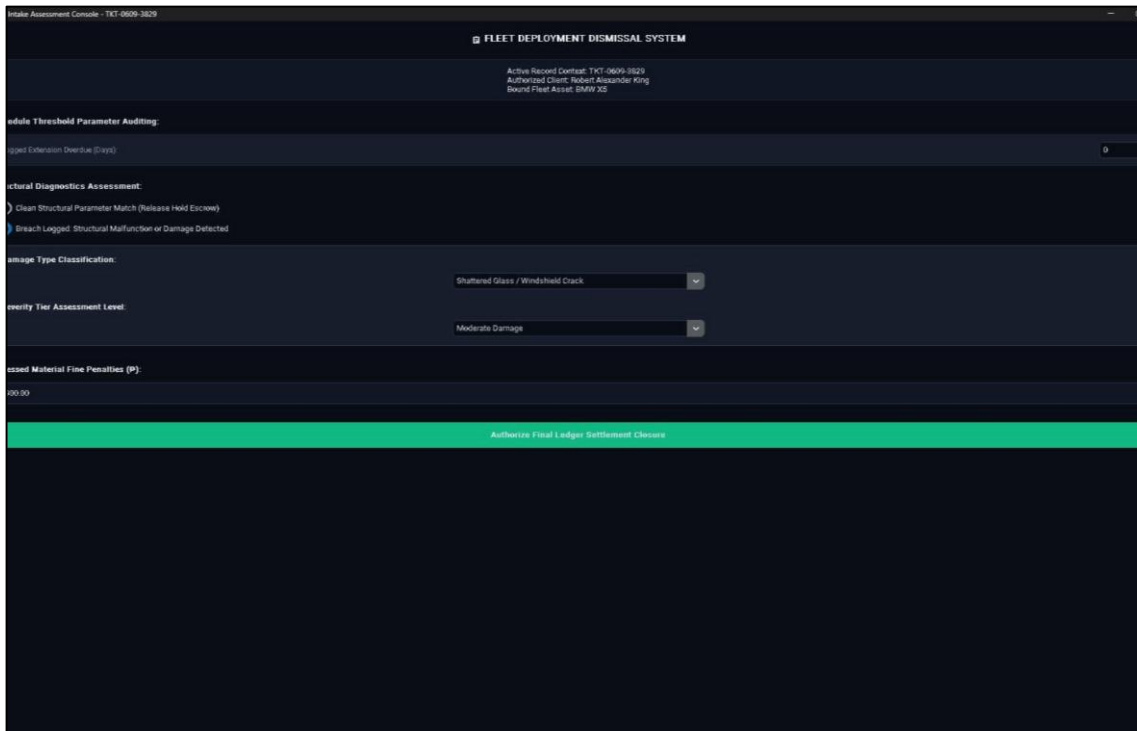


Figure 5. Successful Record Deletion in AutoRentPro Car Rental Management System

This figure shows a deleted car rental record in the system. A confirmation message is displayed to indicate that the selected record has been successfully removed from the database, ensuring accurate and up-to-date management of vehicle and customer information.

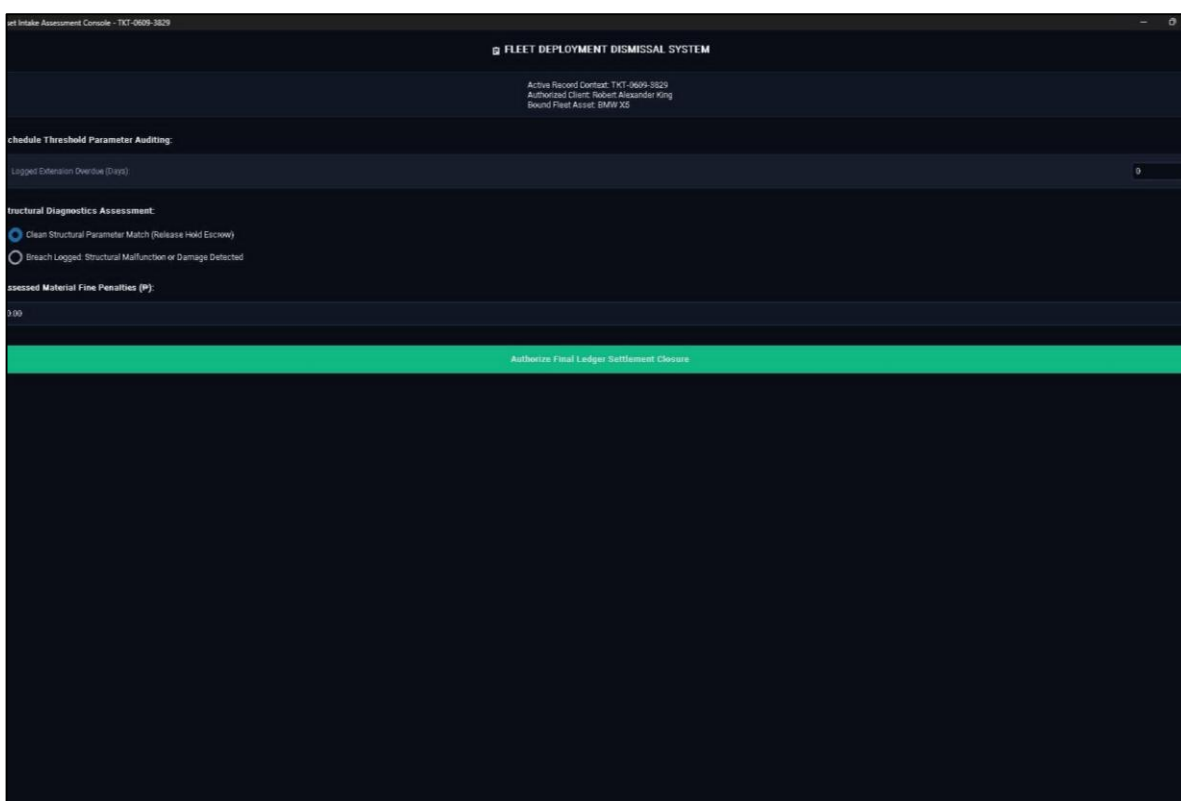


Figure 6. Input Validation for Required Fields in AutoRentPro Car Rental Management System

This figure shows the system’s input validation when attempting to add a record with missing information. An error message is displayed indicating that **all fields are required**, preventing incomplete data from being saved and ensuring accurate and reliable record management.

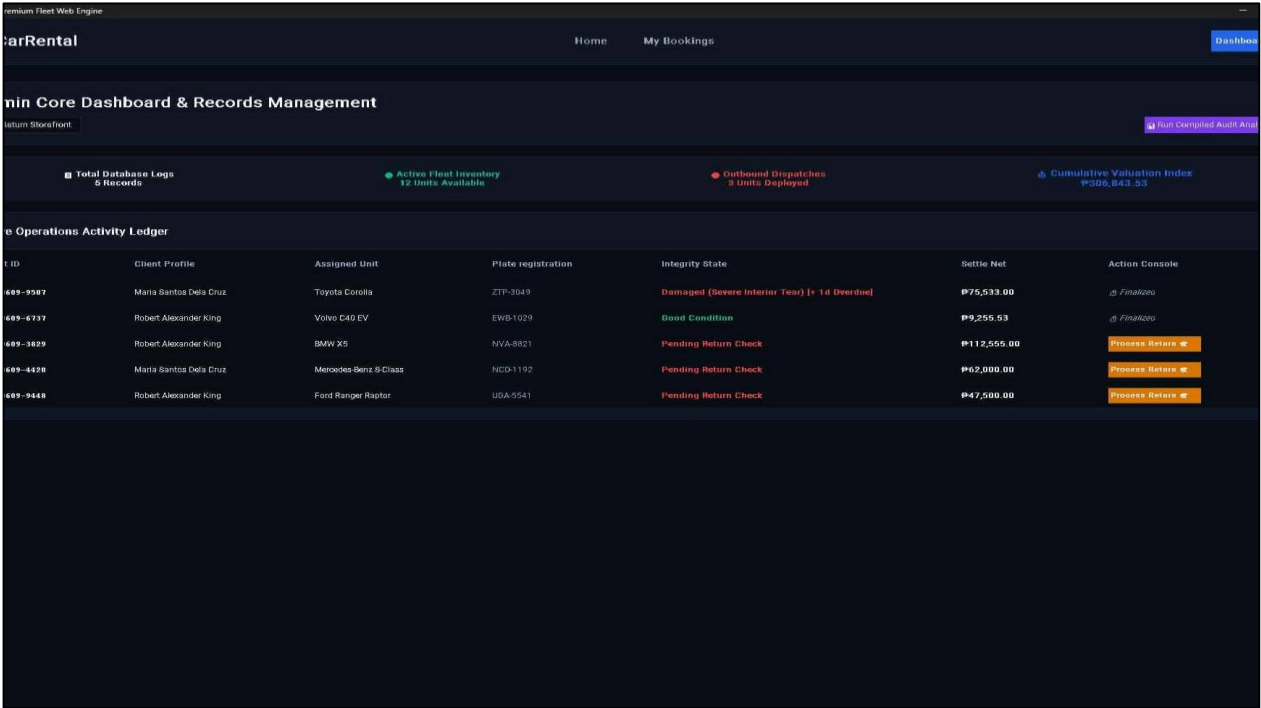
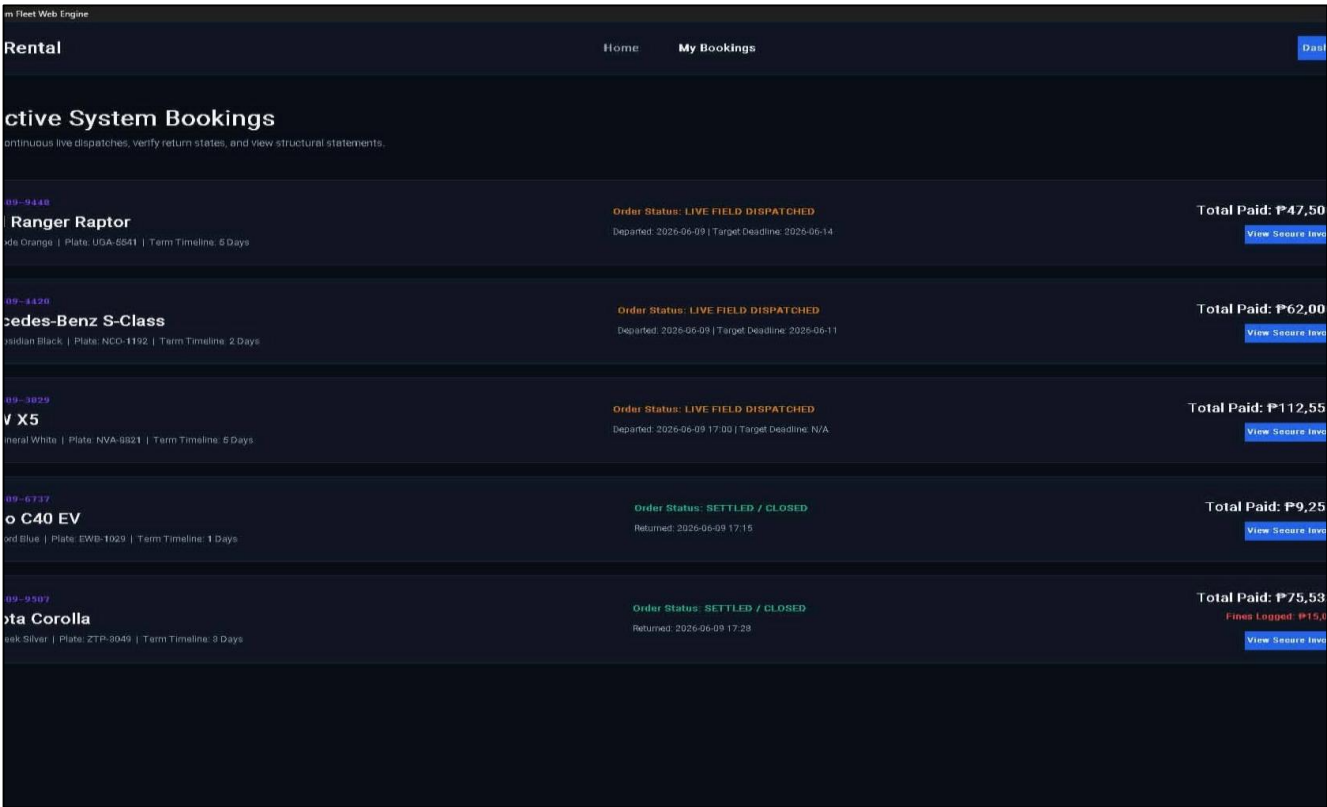


Figure 7. Successful Record Update in AutoRentPro Car Rental Management System



This figure shows the system’s record update feature. When a user modifies an existing car rental record, a confirmation message appears indicating that the **record has been successfully updated** in the database. This ensures data accuracy and reliability for vehicle and customer information management.

Figure 8. Successful Record Search in AutoRentPro Car Rental Management System

This figure illustrates the search functionality of the system. When a user searches for a record by ticket ID or car model, the system locates and displays the matching record along with a confirmation message. This feature ensures that rental records can be quickly retrieved and verified efficiently.

Functional Testing Results

The system was tested to check if all features of the Car Rental Management System are working properly. The researchers used different test cases such as adding records, deleting records, updating data, searching, and entering invalid or incomplete inputs. This was done to ensure that the system can handle errors and perform its functions correctly.

Functional testing was conducted for all modules. Sample test cases include:

Test Case	Action	Expected Result	Actual Result	Status
Blank Input	Left a required field empty	System prompts user	Prompt displayed	PASS
Add Vehicle	Entered valid vehicle info	Record saved in database	Saved successfully	PASS
Update Vehicle	Edited vehicle details	Database reflects changes	Updated successfully	PASS
Delete Vehicle	Removed a vehicle record	Record deleted	Deleted successfully	PASS
Add Customer	Entered valid customer info	Record saved	Saved successfully	PASS
Update Customer	Edited customer details	Database reflects changes	Updated successfully	PASS
Delete Customer	Removed customer record	Record deleted	Deleted successfully	PASS
Book Vehicle	Processed rental transaction	Vehicle status updated; transaction recorded	Transaction successful	PASS
Generate Report	Requested summary	Reports accurate	Correct report generated	PASS
Login Validation	Entered invalid credentials	Access blocked; error displayed	Error message shown	PASS

All modules passed functional testing, confirming that AutoRentPro operates reliably, accurately, and efficiently.

DISCUSSION OF FINDINGS

The results indicate that AutoRentPro:

- **Automates car rental operations** by replacing manual record-keeping
- **Improves accuracy** in vehicle, customer, and transaction records
- **Provides real-time updates** of vehicle availability and rental status
- **Generates accurate reports** for management decision-making
- **Offers a user-friendly GUI** to simplify navigation and system use

Overall, the system enhances operational efficiency, minimizes human error, and ensures a reliable platform for small- to medium-scale car rental businesses.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The developed Car Rental Management System successfully showed how Python, Tkinter GUI, and SQLite database can be used to manage rental records in an easy and organized way. The system allows users to add, view, update, delete, and search records through a simple and user-friendly interface. It also includes validation features that help prevent errors like incomplete inputs and duplicate ticket IDs. Based on the development, testing, and evaluation of **AutoRentPro**, the following conclusions were drawn:

1. **Automated Rental Management:** AutoRentPro successfully automates car rental operations, including vehicle management, customer registration, booking processing, and transaction tracking, reducing manual errors and improving operational efficiency.
2. **Database Accuracy:** Integration with **SQLite** ensures that vehicle, customer, and transaction records are stored accurately, providing reliable information for administrators and users.
3. **User-Friendly Interface:** The GUI developed using **Tkinter** allows easy navigation between modules, ensuring that administrators and customers can efficiently perform operations without technical expertise.
4. **Functional Reliability:** Testing confirmed that the system performs as expected in adding, updating, deleting, and searching records, while also validating user inputs to prevent errors.
5. **Operational Efficiency:** AutoRentPro improves the speed and accuracy of rental record management and supports real-time updates, ensuring smooth operation of car rental services.

Overall, AutoRentPro provides a **reliable, efficient, and user-friendly platform** for managing small to medium-sized car rental operations.

Recommendations

Based on the study and system limitations, the following recommendations are proposed:

1. **Online Payment Integration:** Include online payment gateways such as GCash, PayPal, or credit card support to allow seamless transaction processing.
2. **Multi-User Access:** Implement role-based access for administrators, staff, and customers to enhance security and operational control.
3. **Delivery and Fleet Tracking:** Integrate vehicle tracking or telematics for real-time monitoring of rented vehicles.
4. **Email Notifications:** Include automated notifications for booking confirmation, payment reminders, and rental expirations.
5. **System Analytics:** Add reporting features to visualize rental trends, revenue statistics, and customer activity.
6. **Mobile Compatibility:** Develop a mobile version or web interface to enable remote access for customers and administrators.

7. **Regular Maintenance:** Periodically update and test the system to improve functionality, security, and usability.
8. **Future Research:** Researchers may extend this system by integrating AI-based features, predictive analytics for vehicle availability, or cloud-based storage for scalability.

Implementing these recommendations will enhance AutoRentPro's functionality, usability, and scalability, providing a **comprehensive and modern solution** for car rental businesses.

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. E. Diavastis, K. A. Chrysafis, and G. C. Papadopoulou, "Determinants of Accounting Information Systems Success: The Case of the Greek Hotel Industry," *International Journal of Financial Studies*, vol. 12, no. 2, p. 42, Jun. 2024. <https://scholar.google.com/scholar?q=Determinants+of+Accounting+Information+Systems+Success+Diavastis+2024&hl=en-US>
2. J. T. Ogbiti and W. Aaron, "Development of a web-based car rental management system," *Science World Journal*, vol. 19, no. 3, pp. 797-807, Oct. 2024. <https://scholar.google.com/scholar?q=Development+of+a+webbased+car+rental+management+system+Ogbiti&hl=en-US>
3. C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*, 13th ed. Boston, MA, USA: Cengage Learning, 2018. https://scholar.google.com/scholar?q=Coronel+Morris+Database+Systems+Design+Implementation+Management+13th+ed&hl=en-US#d=gs_qabs&t=1779963978770&u=%23p%3DZj5DmmRKIRwJ
4. R. Ramakrishnan, *Database Management Systems*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2020. https://scholar.google.com/scholar?q=Ramakrishnan+Database+Management+Systems+4th+ed&hl=en-US#d=gs_qabs&t=1779964017406&u=%23p%3DSYMZqNzC6XwJ
5. G. Van Rossum and F. L. Drake, *The Python Language Reference Manual*. Bristol, UK: Network Theory Ltd., 2018. <https://docs.python.org/3/reference/index.html>
6. M. Lutz, *Learning Python: Powerful Object-Oriented Programming*, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2019. https://scholar.google.com/scholar?q=Lutz+Learning+Python+Powerful+ObjectOriented+Programming+5th+ed&hl=en-US#d=gs_qabs&t=1779964065325&u=%23p%3D74ajtm4ETLAJ

7. B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, and N. Elmqvist, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Boston, MA, USA: Pearson, 2016.
https://scholar.google.com/scholar?q=Shneiderman+Designing+the+User+Interface+Strategies+for+Effective+HumanComputer+Interaction+6th+ed&hl=enUS#d=gs_qabs&t=1779964080972&u=%23p%3DY5juZQ32oB4J

ABOUT THE AUTHORS

John Abel Doploso is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is a responsible and hardworking student who is committed to fulfilling his academic responsibilities and achieving his goals. Abel values teamwork, discipline, and cooperation, which help him perform effectively in both individual and group activities. He continuously seeks opportunities to improve his knowledge and develop new skills that will prepare him for a productive career in the field of computer engineering.

Jonathan Hombrebueno is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is a dedicated student who values commitment, perseverance, and continuous growth. Jonathan possesses good multitasking and time-management skills that help him balance his academic responsibilities effectively. Despite challenges, he remains focused on improving himself and working toward his long-term goals as a future computer engineer.

Franz Gabriel G. Honrado is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is a hardworking and dedicated student who strives to fulfill his academic responsibilities to the best of his abilities. He is committed to enhancing his knowledge and skills in the field of computer engineering, particularly in developing practical experience and technical expertise. He enjoys exploring how various technologies and electronic devices work, often taking them apart to gain a deeper understanding of their components and functions. His curiosity and passion for learning continue to drive him toward growth and improvement in his chosen field. Through perseverance, continuous learning, and dedication, he aims to build a successful career and make meaningful contributions to the field of computer engineering.

Aron Luis T. Magalona is currently taking up Bachelor of Science in Computer Engineering at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is a motivated and responsible student who is committed to achieving excellence in his studies. Aron demonstrates adaptability and perseverance when facing challenges and is always eager to learn new knowledge and skills. He looks forward to applying what he learns in college to contribute positively to the field of computer engineering.

James Adrian D. Montero is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is a dedicated student who is eager to learn and deeply committed to pursuing his professional goals in the tech industry. His academic interests focus heavily on mastering computer systems, software architectures, and coding languages like C++ and Python. With a strong motivation for continuous self-improvement and technical skill development, he strives to become a proficient and successful innovator in the field of computer engineering.

Patrick Sean H. Aldepolla is a first-year Computer Engineering student at the Eulogio Amang Rodriguez Institute of Science and Technology. He is currently improving his skills in programming, computer architecture, and upcoming technologies. He is developing his technical skills and obtaining useful engineering experience through project-based learning and classroom instruction. He is dedicated to attaining academic success and hopes to develop into a capable computer engineer who can help promote technology in the future.

John Kenth Cariaga is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is dedicated to his studies and consistently demonstrates responsibility, determination, and a strong work ethic. Kenth values continuous learning and actively seeks opportunities to enhance his

skills and broaden his knowledge. As he progresses in his academic journey, he remains focused on preparing himself for future opportunities and success in his chosen profession.

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.