**Rent-A-Ride**: Streamlined Car Rental Management Using Python Console

**A logo for a sports team

Description automatically generated**

A Console Project

In Partial Fulfillment of Course Requirements in

Computer Laboratory 1

Presented to:

**MARTZEL BASTE**

Computer Laboratory Professor

Presented by:

**Jeon Andreaux B. Ampad**

Proponent

**MAPUA MALAYAN COLLEGES MINDANAO**

Gen. Douglas MacArthur Hwy, Talomo, Davao City, 8000 Davao del Sur

1st Semester of AY 2023-2024

**Introduction**

1. **Problem Scenario**

In the competitive car rental industry, businesses face a significant challenge in efficiently managing their data. Traditional approaches to data management, including customer information, vehicle availability, and rental transactions, may lead to inefficiencies, errors, and difficulties in providing seamless services. As the volume of data grows, car rental businesses find it increasingly challenging to meet customer demands, optimize fleet operations, and ensure a smooth rental process.

Statement of the Problem:

1. Data Inefficiency and Errors:

Car rental businesses encounter inefficiencies and errors in managing extensive datasets related to customer details, vehicle availability, and rental transactions. Manual data handling may result in inaccuracies, leading to operational challenges and potential customer dissatisfaction.

1. Challenges in Meeting Customer Demands:

With the expanding volume of data, car rental companies struggle to effectively meet the dynamic demands of customers. The inability to swiftly access and update information hampers the ability to provide timely and tailored services, impacting overall customer satisfaction.

1. **Project Solution**

The Car Rental Console Application serves as a comprehensive solution to the data management challenges encountered by car rental businesses. It addresses key pain points and offers the following benefits:

**Streamlined Data Management:** The application provides a centralized platform for managing customer data, vehicle inventory, and rental transactions, ensuring a streamlined and organized approach to data management.

**Enhanced Customer Experience:** The application enhances the customer experience by facilitating easy reservation processes, providing accurate information on available vehicles, and ensuring seamless transactions, leading to higher customer satisfaction.

**Scalability:** The application is designed to accommodate the growing volume of data as the business expands, providing scalability and adaptability to changing market dynamics.

1. **Target Market: Automotive Rental Business**

The primary audience for the Car Rental Console Application includes automotive rental businesses of varying sizes, from local enterprises to larger regional or national chains. This may encompass:

**Local Automotive Rental Agencies:** Small to medium-sized car rental businesses operating in specific regions or neighborhoods.

**National Automotive Rental Chains:** Larger, national car rental chains with multiple locations seeking a standardized and efficient data management solution.

**Specialty Rental Services:** Businesses focusing on specific vehicle types, such as luxury cars, electric vehicles, or specialty rentals, aiming to optimize their operations.

By catering to the needs of car rental businesses struggling with data management, the application aims to empower these businesses with a robust and user-friendly tool to enhance their operations and overall efficiency.

1. **Similar Application**

As this console project requires the use of CRUD Operations. There are several applications that already exists, namely:

1. Hotel Management Systems:
2. Library Management Systems
3. Inventory Control Systems

While these similar applications exist, the Car Rental Console Project distinguishes itself through its tailored features for the unique requirements of the car rental industry. The adaptability of CRUD operations allows the project to share foundational concepts with various management systems, providing a versatile solution for businesses across different domains.

1. **Conceptual Framework**

The conceptual framework of the Car Rental Console Application is based on principles and theories from several domains, integrating them to create a cohesive and effective solution. Here are the key components of the conceptual framework:

**Database Management**

* Principles of relational database management systems guide the organization and storage of data. Utilizing concepts such as tables, relationships, and indexing ensures efficient data retrieval and management.

**Business Process Optimization**

* Principles of business process optimization inform features that enhance the efficiency of car rental operations. This includes functionalities such as efficiently updating user and car information, removing unwanted information, and searching data efficiently.

**User Interface (UI) Design:**

* Principles of user-centered design and human-computer interaction guide the creation of an intuitive and user-friendly interface. This includes considerations for information architecture, navigation, and error handling mechanisms to enhance the overall user experience.

1. **Scope and Limitations**

**Scope**

The project is designed to streamline essential CRUD (Create, Read, Update, Delete) operations within the context of a Car Rental Console Application. Its primary focus is on providing a user-friendly interface for managing vehicle rentals, customer information, and inventory. The application aims to enhance efficiency by simplifying routine tasks related to data manipulation in the car rental business.

**Limitations**

**Database Management Features:**

While the project excels in CRUD operations, it does not implement advanced database management features, such as a full-fledged Relational Database Management System (RDBMS). This means the project may not scale optimally for large-scale enterprises with complex data relationships.

**Efficiency Considerations:**

The program may not fully adhere to Object-Oriented Programming (OOP) principles, limiting its scalability and extensibility. The focus on CRUD operations may result in some constraints in terms of flexibility and code organization.

**User Interface Complexity:**

The project prioritizes simplicity in its user interface, which may be a limitation for users seeking more feature-rich interfaces or advanced functionalities beyond basic data manipulation.

1. **Project Definition**

**Programming Language & IDE**

* The project is implemented using Python, a versatile and widely-used programming language known for its simplicity and readability. The project was developed using PyCharm as the integrated development environment.

**Libraries**

Time library: this is used for managing time-related functions. Specifically in the project time was utilized to create the progress bar.

Rich Library: is a key component for crafting a visually appealing and feature-rich console interface. It provides tools for styling text, creating tables, and designing panels, contributing to an enhanced user interface.

**Dependency Management**

‘pip’ is the package installer for Python used for managing project dependencies.

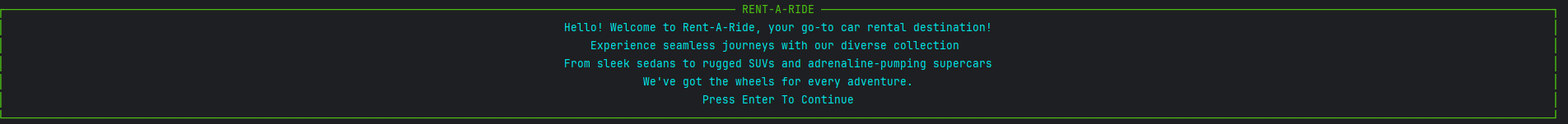
By combining these tools, libraries, and frameworks, the project achieves a balance between functionality, aesthetics, and maintainability, providing a robust and user-friendly console application for car rental management.

**Overview of the Project**

1. **Project/System Prototyping**

Rent-a-Ride is a console application designed to revolutionize data management for car rental businesses. With a sleek user interface, the application streamlines complex data processes, offering a seamless experience for both administrators and customers. Robust error handling ensures data accuracy. Scalable and adaptable, Rent-a-Ride grows with your business as it allows you to add more cars to your fleet. Elevate the customer experience with an innovative UI, making Rent-a-Ride the go-to solution for efficient and effective car rental operations.

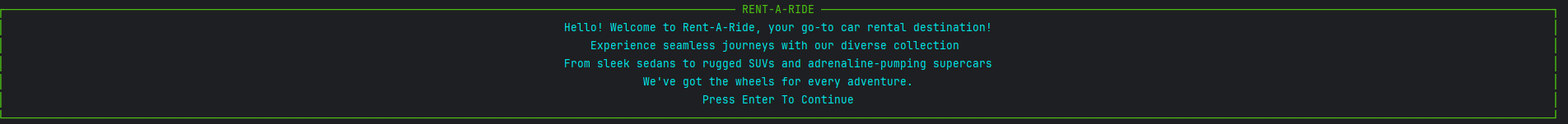
A screen shot of a computer

Description automatically generated****

**Functions of The Program**

There are 12 Functions of the Program namely:

1. WelcomeFunction():

****A screen shot of a computer code

Description automatically generated

* This function prints out a welcome message when called upon, the message is printed using the text and panel from the rich library.

1. Menu():



A screen shot of a computer

Description automatically generated

* The menu function displays the list of functions that is available in the program, again it is printed using the text and panel from the rich library

1. Choosemenu():

A black screen with colorful text

Description automatically generated

* Here is a small snippet of a code of the choosemenu(): function
* This code prints out a message asking to enter a choice based on the menu function, it then prints out the selected function and calls it.

A screenshot of a computer screen

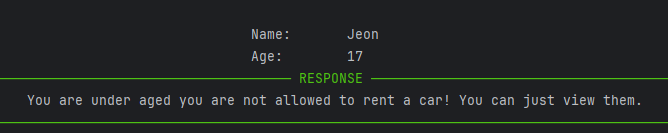
Description automatically generated

1. A screen shot of a computer

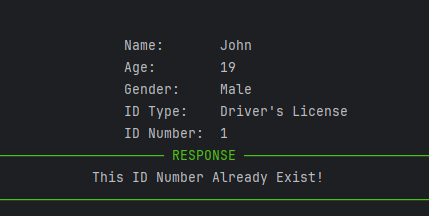
   Description automatically generatedAddCustomer():

* This function allows the user to add a customer with the details name, age, gender, id type and id number. And select a car on the list

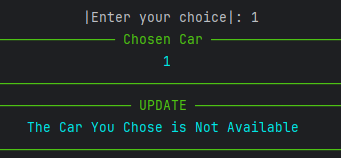
If the user inputs an age below 18, the program will print out a message.



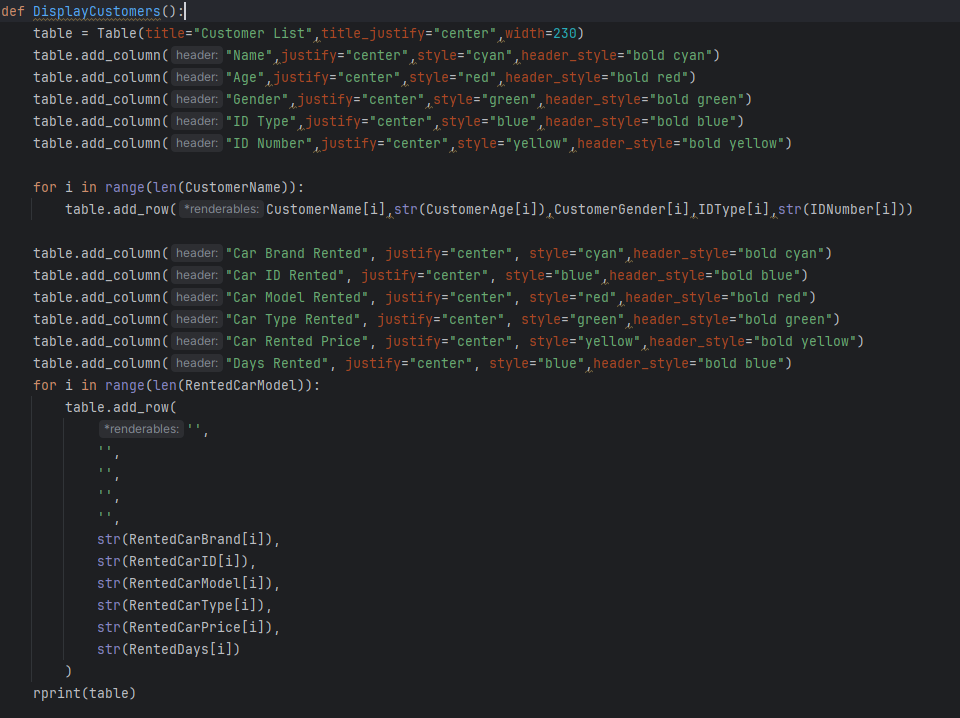
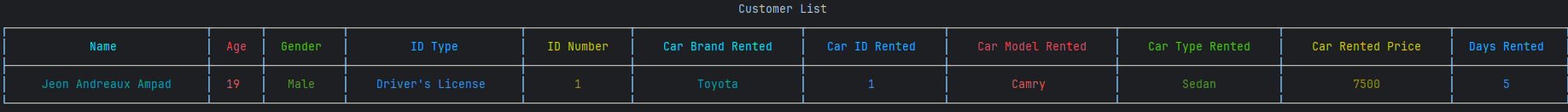
if the user inputs an already existing id number, the program will print out a message.



If the user inputs a car that is already rented, it will print out:



1. DisplayCustomers():



* This function works together with the AddCustomer() function as it uses the list in which the AddCustomer() appended the customers details. In the addCustomer, we used the pop method to remove the chosen car from the original list and appended it in a separate lists called Rented[].

1. CustomerReturn():

A screen shot of a computer

Description automatically generated

-In this function the customer returns the car as such the customer details will be removed from the list. In addition, the rented car will be appended back to its original list.

1. SearchCustomer()

A blurry image of a computer screen

Description automatically generated

* In this function the program asks to enter a customer detail to search, this can include the name, age, gender, id type and id number.

1. UpdateCustomer():

A screenshot of a computer program

Description automatically generated

* In this function the program will ask the user to type an ID Number to Update. If the ID number exists, it will prompt the user to input new details for the customer.