****

**CP 317 – Software Engineering**

Spring 2024

Group 26

CP317 Group Project

Member 1 Name: Ranjot Sandhu ID: 169020301

Member 2 Name: Rishubh Gusain ID: 169044443

Member 3 Name: Jack Jarrett ID: 210576140

Member 4 Name: Aditya Moon ID: 210399020

Member 5 Name: Ryan Wilson ID: 169038861

### Table of Contents

[**Table of Contents 1**](#_2w6kokph9r8o)

[**Summary 2**](#_9huqedfkiwn6)

[**Introduction 3**](#_rys7kbh7vir6)

[**Problem Definition 3**](#_oacghjnm23op)

[**Functional Requirements 3**](#_e2k16f42srqp)

[**Non-Functional Requirements 4**](#_gi8u6prsislv)

[**Limitations 5**](#_n8e5cs90a0u3)

[**Assumptions 5**](#_bffhk2wqyhaw)

[**Function and Interaction Between Components 6**](#_t5xggv4idj1d)

[**Use Case Diagram & UML Class Diagrams 8**](#_l1uml0jhsljb)

[**Component & Sequence Diagrams 8**](#_b7tpd9tkfae5)

[**Citations 11**](#_4hrlpdimxdm6)

**Video For The Report -** [**https://drive.google.com/file/d/1WSJnRSevXEyuv6SIRPHrqcsVPQawJh3V/view?usp=sharing**](https://drive.google.com/file/d/1WSJnRSevXEyuv6SIRPHrqcsVPQawJh3V/view?usp=sharing)

### Summary

Our group has developed a Rental Agency Management System to simplify the process of renting cars, trucks, and luxury vehicles. A user-friendly application that provides easy-to-use rental vehicle cost estimates, our system enhances transparency and helps customers make informed decisions.

The current car rental process can often be described as unclear and confusing, leading to frustration among consumers. Our application solves these challenges by offering a user-friendly interface that simplifies vehicle rentals.

In the requirements and analyzing phases, we identified functional requirements, including collecting user inputs for key metrics and customer identification. The application must also compute the rental costs and present them to the consumer with the option to book or cancel. Non-functional requirements considered were programming in Java, establishing a focus on modular programming, and accuracy of rental calculations.

Despite its advantages, the application has limitations, including reliance on accurate user input, not being suited for unique rental scenarios, and mitigating the human element by focusing on simplicity of design. We had to make assumptions about the environment and information used in our application. These assumptions included that users have experience interacting with existing computer programs, rental agencies have inventory management to avoid overbooking, and finally, we assumed vehicle rental price estimates from current services in the market.

Our Rental Agency Management System streamlines and creates a more valuable vehicle rental experience while simplifying and increasing transparency.

### Introduction

Our group has developed a Rental Agency Management System that seeks to simplify the process of renting cars, trucks, and luxury vehicles. Our user-friendly application enables consumers to easily navigate different rental options allowing them to find the solution that is best suited for them. Users can calculate rental fees based on vehicle type, rental duration, and expected mileage. By using these metrics, our system provides accurate cost estimates, making it easier for customers to understand their rental options.

### Problem Definition

The traditional car rental process can often be confusing and overly complicated for many consumers, thus leading to frustration with a lack of transparency in the pricing of rental agreements. Customers frequently encounter difficulties understanding rental fees, navigating complex terms, and estimating costs based on their needs (Enterprise, 2024). Our Rental Agency Management System looks to address these challenges by providing an easy-to-use interface that simplifies the rental process. Users can calculate rental fees based on key metrics to determine costs such as vehicle type, rental duration, and expected miles driven. Our application enhances transparency and enables customers to make informed decisions. Our solution improves the overall rental experience but also helps increase efficiency by reducing the potential for misunderstandings over pricing.

### Functional Requirements

In the requirement and design phases during the development life cycle of our project, we determined the following functional requirements of our application that must dictate the required outputs of the application. The application must accept inputs to gather the necessary variables to determine the cost of renting a vehicle. These inputs included the type of car, rental length, and the expected miles driven. Along with accepting the mentioned inputs, we must also gather user information to assign the rental plan customer's name and driver's licence number. The application should be able to handle errors collecting user inputs and provide clear messages to the user to advise the user on the acceptable inputs that should be entered. Upon gathering user inputs, the application should calculate the vehicle rental costs using adjusting for unique inputs from the user such as vehicle type, rental duration, and the expected miles driven. After calculating these costs the program must present the user with a confirmation screen with the associated cost of the vehicle rental with options to cancel or proceed to confirm the rental. By implementing the mentioned functional requirements we can create an effective system to collect the necessary information, handle errors, calculate rental costs, and provide clear transparency to consumers.

### Non-Functional Requirements

In the process of designing our application, it was necessary to consider the non-functional requirements that determine the desired properties that we wanted to implement. Coding our project, we wanted to use Java due to familiarity among team members along with its ease of use and strong compatibility with object-oriented programming. Consistency and accurate projection of rental car fees to ensure our application helps deliver our mission of simplification and creating transparency between consumers and rental car services. Another requirement was to create our program with modularity as a focus of design. Modularity helped enhance the application’s scalability, maintenance, and encapsulation. By implementing our non-functional requirements, we can have confidence in the longevity of our application to ensure we provide value to consumers of our application.

### Limitations

Although our rental vehicle application creates a user-friendly experience streamlining the rental vehicle process, it has some limitations. The dependency on accurate user input, as any discrepancies in the information collected and accurate information, may cause disruptions in the rental process. Additionally, the system may not support all types of rental scenarios, such as rentals with unique conditions or long-term leases, which could restrict its appeal to a larger audience. As a result of using our application that simplifies the rental process by removing the human element, it may reduce the impact of customer service interaction in support of simplicity and efficiency. Overall, while our application aims to enhance the vehicle rental experience, continual improvement and maintenance will be useful to maximize its ability to deliver optimal effectiveness.

### Assumptions

In the development of our vehicle rental application, we had to make the following assumptions. The first assumption was that users would have familiarity interacting with computer program interfaces and submit the appropriate information. We also assumed that the vehicle rental agencies had internal inventory management systems to reduce the potential for overbooking vehicles in our application. Finally, we made basic assumptions on pricing vehicle rental agreements using prices we gathered from relevant vehicle rental services such as Communauto and Zip Car. However, these prices would likely differ depending on the vehicle rental service's pricing models. These assumptions were key in the development of our application but will need to be continually reassessed to enhance the overall rental experience.

### Function and Interaction Between Components

| **Component Name** | **Component Description** | **Component Attributes** | **Interaction with other Components** |
| --- | --- | --- | --- |
| VehicleType | Presents all available vehicles that consumers can rent form the vehicle rental service | vehicleType: enum | VehicleType writes the different vehicle types to CarRentalGUI  VehicleType writes the different vehicle types to CarRentalService |
| Customer | Defines the personal user information that should be collected from the application | name: String  license: String | Customer writes customer fields to CarRentalGUI |
| CarRentalService | Calculate the rental cost for various vehicle types based on user metrics of vehicle type, rental duration, miles driven. | ratePerDay: double  baseMilage: int | Reads parameters from CarRentalGUI  Writes total cost of vehicle rental to CarRentalGUI  Reads the different vehicle types from vehicleTypes |
| CarRentalGUI | Connects all of the interfaces to each other and creates the interface that will be displayed to the user | N/A | Reads different vehicle types from VehicleType  Reads personal customer information from Customer  Writes parameters to CarRentalGUI and reads the total cost output |
| CarRentalApp | Responsible for initializing and launching the application | N/A | N/A |

### Use Case Diagram & UML Class Diagram

The use case diagram showcases the use cases and actors of the car rental system, and highlights the relationships between them. The UML class diagram showcases the different java classes of the system, as well as their attributes, operations, and relationships.

<https://lucid.app/lucidchart/6f25bd40-50f4-4966-8745-011f8d3cff19/edit?viewport_loc=-1192%2C-709%2C6214%2C2750%2C.Q4MUjXso07N&invitationId=inv_bab52657-56a4-44ef-8cb8-c593752f9f39>

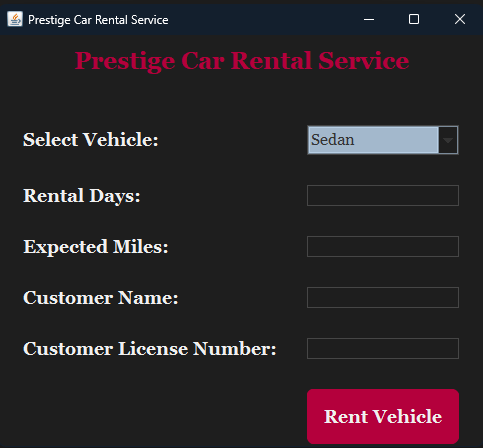
### Component & Sequence Diagrams

The two component diagrams showcase the internal structures of the CarRentalService and CarRentalGUI components, including their functionalities and sub-components. The two sequence diagrams showcase the control flow of the “Select Vehicle Type” and “Submit Rental Information” use cases, highlighting the flow of data through different actors as the use case is completed.

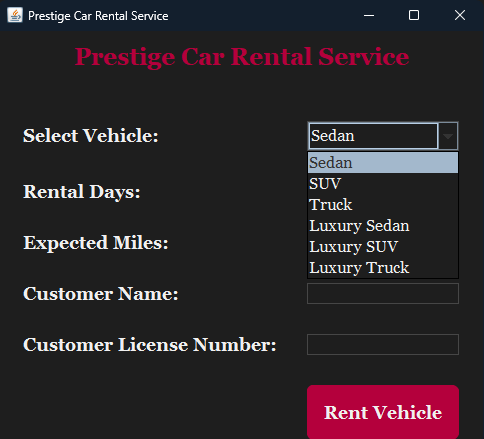
<https://lucid.app/lucidchart/0d5ca04a-8312-4b63-903e-a0d052208c31/edit?viewport_loc=-2658%2C-1124%2C5120%2C2266%2CHWEp-vi-RSFO&invitationId=inv_e2decc43-db77-4968-979a-08b005b5d1cf>

Inputs And Outputs

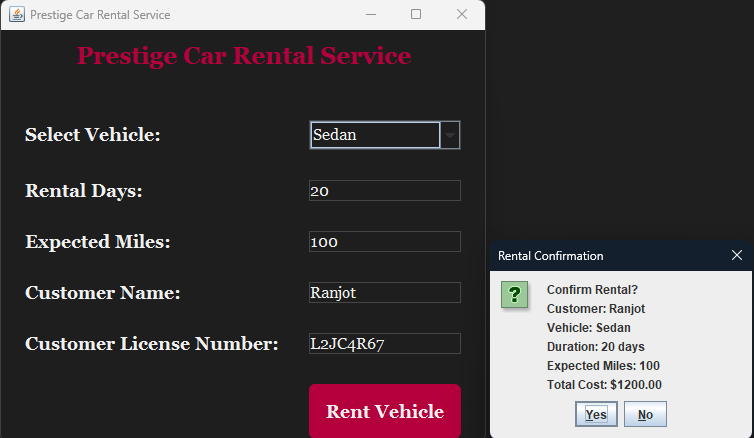
This is the main page of the software where the user interacts to fill in the needed information for the rental of the car.



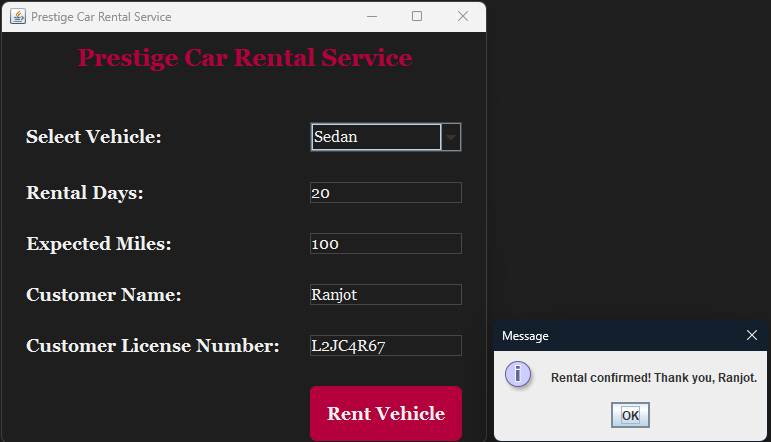
This image shows that there is a drop down menu for the user to select what kind of car they would like to rent out.



Here is the information all filled out with a confirmation tab that opens up for the user to confirm the information they have inputted into the system.



This last image is just a rental order confirmation screen where a confirmation tab opens up with the user's name that they inputted.



### Citations

“Car Sharing Alternative for Daily and Hourly Car Rental.” *Zipcar*, my.zipcar.com/search. Accessed 4 Aug. 2024.

enterprise. *The Most Common Issues with Car Rental And ...*, cdn.asp.events/CLIENT\_Northsta\_5B1737B2\_5056\_B733\_83AD112B2FDD2BEF/sites/Business-Travel-Show-2021/media/libraries/whites-papers-1/12816-2020-08-Enterprise-General Purpose-The Most Common Issues with Car Rentals and How to Avoid Them-ENG.pdf. Accessed 4 Aug. 2024.

“Rates.” *Communauto Toronto*, 14 May 2024, toronto.communauto.com/rates/.