

UNIVERSITY OF TECHNOLOGY, JAMAICA

School of Computing and Information Technology

Object-Oriented Programming Project

R. Andrews / M. March

Group Assignment (2 - 4 persons to each group)

Given Week of January 27, 2020

Due Week of March 16, 2020

Cars4Rent is a successful vehicle rental company in the parish of Kingston. Due to an increase in vehicle rental during the summer period, the owner would like an automated system through which persons can use to view and rent vehicles. You have been asked to implement the proposed system.

There are three types of vehicles, these are Car, Truck, and Bike. For each vehicle, the system records the license plate number, brand, model, year, color, engine size, transmission (A or M), mileage, the number of seats, rental status and rate per day. For bikes, the number of helmets provided (1 or 2) is stored and for trucks the system tracks the towing capacity in kg.

The four main functions of the application are:

1. View Vehicles
2. Search Vehicles
3. My Rentals
4. Return Vehicle

- **View Vehicles**

The user should be able to view all the vehicles available (see excel sheet for dataset) and select the vehicle they want to rent. All the relevant vehicle information must be displayed in a readable format (e.g. table or grid). After the vehicle has been selected the system should capture and store the following rental information to a text file:

- License plate #
- Name
- Home address
- Phone #
- Date borrowed
- Expected return date

- **Search Cars**

The user should be able to search for a vehicle using the following attributes:

- License plate #
- Brand
- Model
- Year

When the user selects the vehicle from the search results, the system should store the rental information to a text file. The search should be performed for **one attribute** at a time.

- **My Rental**

The My Rental screen, displays all the vehicles you have rented. The user should not be able to rent more than 3 vehicles.

- **Return Vehicle**

The return vehicle screen should allow the user to specify which vehicle they are returning using the license plate #. The system should capture the following information when a vehicle is being returned:

- License plate #
- Actual return date
- Current mileage

A receipt should be displayed to the user, showing the cost breakdown. The total owed is calculated by multiplying the rate per day by the number of days the vehicle was borrowed. If the actual return date exceeds the expected return date, the user should be charged an additional \$1000 for each day over the limit. The formulas are shown below:

- $\text{Total Owed} = (\text{Date Borrowed} - \text{Actual Return Date}) * \text{Rate per day}$
- $\text{Late Fee} = \text{IF Actual Return Date} > \text{Expected Return Date}$
 $\text{THEN } (\text{Expected Return Date} - \text{Actual Return Date}) * 1000$
- $\text{Final Cost} = \text{Total Owed} + \text{Late Fee (If any)}$

After the vehicle has been returned, it should be removed from My Rentals, the rental status should change as well as the mileage.

Required:

Perform an object-oriented analysis on the proposed system described above, and then design the system using the unified modelling language (UML). Utilize composition and inheritance in your design to increase reusability and reduce system complexity.

Grading Scheme (100 marks): General Mark Breakdown

- **Documentation (20 marks)**
 - Group Report (outlining contribution(s) of each member) [3 marks]
 - Object-Oriented Analysis and Design of system [12 marks]
 - User Manual [5 marks]
- **Source Code (30 marks)**
 - Comments [3 marks]
 - Each file should have details for the student(s) who wrote the file
 - Practice use of self-commenting files (i.e. proper variable and method naming)
 - Proper use of inline and method comments where necessary
 - Naming Convention [2 marks]
 - Pascal Case should be used for naming classes
 - Camel Case should be used for variable and method naming
 - Ensure class files are named appropriately
 - Object-Oriented Programming Techniques [20 marks]
 - Inheritance, Polymorphism, Composition
 - Method overriding and overloading
 - Use of Files [5 marks]
 - Proper implementation of appropriate file management
- **Functionality (50 marks)**
 - Robustness [10 marks]
 - User Input validation
 - Error / Exception Handling
 - Program Navigation (i.e. Menu System)
 - User Interface
 - Ease of User Interaction [6 marks]
 - Appropriate Notifications (i.e. error and information messages) [4 marks]
 - System Functionality Implemented
 - View Vehicles [5 marks]
 - Search Vehicles [10 marks]
 - My Rentals [5 marks]
 - Return Vehicle [10 marks]

Extra Marks (10 marks):

A project that satisfies the program's functional requirements can gain additional 10 marks:

- + 10 marks – Awarded for use of colour and graphics to enhance the look and feel of the Program.

Submission:

Zip all your documents and upload to the link below on or before the due date of March 16, 2020.

<https://www.dropbox.com/request/ebQ4ZaVzTM4juW9Rk76g>

Your zip file should contain all Documentation (User Manual, OOA & OOD), Source Files and Executable. The name of the zip file should be in this format:

GROUPEADERSID#.zip

Each group will have to do a 10 minute interview with their tutor to receive a grade.

Late Submission:

Any project submitted after the due date will be late and 5 % will be deducted for each day late. Additionally late projects will not be considered for extra marks. Saturday and Sunday will count as one day late.

Group members:

No group should exceed four members, any group exceeding the limit will receive a 20% deduction in their overall grade.