Tunku Abdul Rahman University Of Management and Technology

BMCS2023

Object-Oriented Programming

Assignment
2025/2026

Programme : RISY1S3
Tutorial Group : Group 3
Date Submitted to Tutor : 1/8/2025

Team Members:

NO	STUDENT NAME	STUDENT ID
1	LYE WEI LUN	24WMR15116
2	OSCAR LIM ZHENG YOU	24WMR14330
3	NEESHWRAN A/L VEERACHELVAN	24WMR13795
4	TEH GUAN CHEN	24WMR16748
5	CHIN WEN WEI	24WMR15026

NO	Team member	Task allocated	Overall Contribution (%)
1.	LYE WEI LUN	Cover Page, Description of Idea, Modules, UML diagram Draft	20
2.	OSCAR LIM ZHENG YOU	Modules, UML diagram Draft	20
3.	NEESHWRAN A/L VEERACHELVAN	Modules, UML diagram Draft	20
4.	TEH GUAN CHEN	Modules, UML diagram Draft	20
5.	CHIN WEN WEI	Modules, UML diagram Draft	20

Coursework Declaration

We confirm that we have read and shall comply with all the terms and conditions of TAR University of Management and Technology's plagiarism policy.
We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is our own properly derived work.

Signature :					
Name :	LYE WEI LUN	OSCAR LIM ZHENG YOU	NEESHWRAN A/L VEERACHELVAN	TEH GUAN CHEN	CHIN WEN WEI
Date :	26/7/2025	26/7/2025	26/7/2025	26/7/2025	26/7/2025

Description of Idea---------- 1
Module handled by each member----------2
1) Customer Module (Oscar Lim Zheng You)----------2
2) Vehicle Module (Lye Wei Lun)---------- 3
3) Payment Module (Teh Guan Chen)----------4
4) Rental Module (Chin Wen Wei)----------6
5) Duration Module (Neeshwaren A/L Veerachelvan)---------- 7
6) Database Module(Lye Wei Lun)---------- 8
Draft UML diagram---------- 9

Description of Idea

HillClimmer Sdn Bhd, our company is themed around providing vehicle rental service in the hilly part of Pahang, Malaysia for extreme motorsports enthusiasts. The nature of our business is to allow users to rent our Mountain-suited vehicle to use for a short period of time to entertain themselves with mountain activities such as Mountain-biking and dirt bike, buggy ride, 4x4 Crossover vehicle for long travel and camping purposes. Currently, we are using a system whereby our customer will book through a phone call and all the rental tracking is all on the papertrail. The mountains of papertrail for our booking system has made it hard to efficiently process every rental and track them.

The proposed rental system will allow users to easily book a new booking through our automated system, in which they will be able to register themselves, have that info recorded, and relogin, input their license info and birthdate to track their age for safety checks. Afterwards they may select an extensive amount of vehicles which vary the rental period for each of them. For example, mountain-bikes may have a 1 day rental period whereby 4x4 crossover trucks can be rented up to a week at a time. The customer will be paying for their rental and may choose whether to save their payment for future purchase that makes their next rental easier. Afterwards, our customer with their choice of mountain related activity and travel, the remaining period of their usage will be tracked by our system, and remind them if they have reached 80% and notify them to return the vehicle on time. Afterwards the vehicle will be added back into our inventory for our next customer to rent. Written in Java using the Spring boot framework.

Module handled by each member

1) Customer Module (Oscar Lim Zheng You)
Class: Customer, SafetyCheck
Description:
This module manages customer information and safety validation. The Customer class is responsible for storing and managing customer details such as customerID, name, phoneNo, email, licenseID, age, and registrationDate. Moreover, the SafetyCheck class provides a short quiz that customers must pass before renting or viewing any vehicle types to ensure they fully understand the safety rules.

Class: Customer
Object/Variable:

- **customerID :** Unique ID for the customer
- **name :** Customer full name
- **phoneNo :** Contact number
- **email :** Email address
- **licenseID :** Driving license number
- **age :** Customer age
- **registrationDate :** Date of registration

Method:

- **getCustomerID() :** Returns customer ID
- **getName() :** Returns name
- **getPhoneNo() :** Returns phone number
- **getEmail() :** Returns email address
- **getLicenseID() :** Returns license number
- **getAge() :** Returns age
- **getRegistrationDate() :** Returns registration date
- **setPhoneNo(String phoneNo) :** Updates phone number
- **setEmail(String email) :** Updates email
- **setLicenseID(String licenseID) :** Updates license number
- **toString() :** Displays all customer info

Class: SafetyCheck
Object/Variable:

- **quizID :** Unique ID for quiz
- **questionList[] :** Questions in the quiz
- **correctAnswers[] :** Correct answers for checking
- **minPassScore :** Minimum score required to pass

Method:

- **startQuiz() :** Starts the quiz
- **checkScore(int score) :** To check the correct score answers
- **isPassed(int score) :** Returns true if the score passes

2) Vehicle Module (Lye Wei Lun)
Class: Vehicle, VehicleManager
Description: The module covers the handling of the vehicle with sub-classes of vehicle type handled by Vehicle class, and the management of vehicle i.e. returning or adding a new vehicle into the system will be handled by Manager class.

Class: Vehicle
Object/Variable:

- **vehicleID :** Unique ID that has information on vehicle type, model and numbering system.
- **vehicleType :** To specify the vehicle type of Mountain Bike, Dirt Bike, Buggy, and Crossover vehicle.
- **vehicleCon :** Condition of the vehicle condition record it before and after for record purpose.
- **vehicleModel :** Brand and model of each vehicle to separate each vehicle type with its pricing.
- **modelPricing :** The specified pricing for each vehicle model and brand.

General Methods:

- **getAvailable() :** To view the current available
- **displayPricing() :** To display pricing on vehicle that is selected:
- **returnVehicle() :** To allow rented vehicle to be added back to current lineup of vehicle
- **updateCondition(String newCondition) :** To update on vehicle condition after each rental
- **getVehicleDetails() :** To view a summary of a vehicle's past rental history, condition and related details.

Class: VehicleManager
Object/Variable:

- **managerID :** Unique identifier for Vehicle manager according to the level of authorization that they have.
- **authorizeLv :** Authorization level specification on permission that they can do.
- **managerName :** Manager's name
- **changeCount :** Records on how many changes that each manager have added.
- **modifierRules:** Rules on what each Manager can do.

General Method:

- **addVehicle(Vehicle newVehicle) :** To add a new or existing vehicle that has been processed yet into the existing pool of vehicles.
- **removeVehicle(String vehicleID) :** To remove a vehicle from the existing pool for vehicle retiring purposes or updating manual booking.
- **setVehiclePricing(String vehicleID, double newPrice) :** To update on each vehicle pricing.
- **setAuthorization(int requiredAuthLevel) :** To set a new authorization value.
- **getAuthorizeLevel():** To view each managed authorization level and description of role.
- **resetChangeCount() :** To reset change count.
- **setVehicleDetails(String newDetails):** To change vehicle details.

3) Payment Module (Teh Guan Chen)
Class: Payment, Invoice, TransactionManager
Description: This module handles all payment-related processes in the system. It manages customer payments, generates invoices, tracks transaction status, and calculates totals including taxes or discounts.

Class: Payment
Object/Variable:

- **paymentID** : Unique ID for each successful transaction
- **totalAmount** : Total amount paid
- **paymentMethod** : Mode of payment (e.g., Credit Card, Cash, eWallet)
- **paymentStatus** : Paid, Pending, Failed
- **timestamp** : Date & time of payment
- **customerID** : Reference to the customer

Method:

- **processPayment()** : Initiates the payment process
- **refundPayment()** : Processes refund
- **getPaymentDetails()** : Returns receipt of a payment
- **updateStatus(String newStatus)** : Updates the payment status

Class: Invoice
Object/Variable:

- **invoiceID** : Unique ID for each invoice
- **vehicleID** : Reference to the vehicle
- **customerID** : Reference to the customer
- **issueDate** : Invoice issued date
- **dueDate** : Payment due date
- **itemList** : List of billed items (e.g., rental fee, add-on fee)
- **amount** : Total amount after calculating itemList, taxAmount, and discount
- **taxAmount** : Tax is calculated according to government's law
- **discount** : Apply discount if any

Method:

- **generateInvoice()** : Creates an invoice after confirmation of rental
- **addItem(String item, double price)** : Adds a billable item
- **calculateTotal()** : Calculates the final amount.
- **applyDiscount(double amount)** : Adds a discount to the invoice
- **getInvoiceDetails()** : Returns a full invoice summary.

Class: TransactionManager
Objects/Variables:

- **transactionManagerID** : Unique ID for transaction manager
- **transactionList** : Stores all completed/pending/failed transactions
- **totalEarnings** : Sum of all transactions

Methods:

- **recordTransaction(Payment p)** : Saves a successful transaction
- **getTransactionSummary()** : Provides a summary of all transactions
- **getTransactionByID(String paymentID) :** Finds a transaction using ID

4) Rental Module (Chin Wen Wei)
Class: Rental, RentalManager
Description: This module manages the rental transactions between customers and vehicles, handling the entire rental process for businesses.

Class: Rental
Object/Variable:

- **rentalID:** Unique identifier for each rental transaction
- **customerID:** Reference to the customer making the rental
- **vehicleID:** Reference to the vehicle being rented
- **rentalDate :** Start date/time of rental
- **returnDate :** Expected return date/time
- **actualReturnDate :** When vehicle was actually returned
- **rentalStatus:** Current status (reserved, active, completed, cancelled)
- **damageReport:** Any damage noted at returndamageReport: Any damage noted at return
- **paymentStatus :** Track if rental is paid

Method:

- **startRental():** Changes status from reserved to active
- **completeRental():** Finalizes rental transaction
- **generateInvoice():** Creates rental invoice document
- **generateDamageReport(String report):** Records any vehicle damage
- **extendRental(Date newReturnDate):** Modifies new rental period
- **getRentalDetails():** Returns summary of rental information

Class: RentalManager
Object/Variable:
managerID: Unique identifier for rental manager
rentalProcessed: Count of rentals handled by this manager
dailyRevenue: To track daily rental income
activeRentals: List of currently active rentals

Method:

- **createRental(Customer customer, Vehicle vehicle, DateRange dates):** Creates new rental record
- **cancelRental(String rentalID):** Cancels a reserved rental
- **processReturn(String rentalID):** Handles vehicle return process
- **generateDailyReport():** Creates summary of daily rentals
- **findAvailableVehicles(DateRange dates, VehicleType type):** Return available vehicles
- **updateRentalStatus(String rentalID, RentalStatus newsStatus):** Modifies rental state
- **viewRentalHistory(String customerID/vehicle):** Retrieves past rentals
- **calculateTotalRevenue(DateRange period):** Computes income for period

5) Duration Module (Neeshwaren A/L Veerachelvan)
Class: Reminder, RentalPeriod
Description: This module is useful for involving time management, scheduling and rental systems. It handles scheduling and tracking of reminders on specific time intervals or due dates. It also manages start and end dates of rental durations, and can calculate total rental time or check if a rental is still available.

Class : Reminder
Object/Variables:
reminderID : Unique ID to identify each reminder on which it can be retrieved,updated or deleted.
reminderTime : Date and time when the reminder should trigger
rentalID : Refers to which rental that also shows the customer that is involved

Methods:
startCountdown() : To initiate the countdown
sendReminder() : Alerts the customer with a warning/reminder
scheduleReminder() : Sets a new reminder time

Class : RentalPeriod
Object/Variables:
startTime : The beginning of rental period
endTime : The scheduled end of rental period
status : Shows the current status of rental (active, completed,overdue)
duration : Shows the rental time that has been used
rentalID : Reference to the rental that has been made

Methods:
getDuration() : Calculates total rental duration
isValidPeriod() : Rental is active if rental is ongoing
isOverdue() : Checks if the current time has passed the rental period

6) Database Module(Lye Wei Lun)
Class: Database

Draft UML diagram

Draw using Excalidraw