Test Case ID	Test Case Summary	Related Requirement	Pre-Conditions	Test Procedure	Test Data	Priority	Expected Result	Actual Result	Status	Remarks	Test Case Author	Executed By	Test Environment	Frequency	Reporting Date
TC-0001	Check if the "getCurrentDate" method of the "DateTime" class returns the current date in the expected format.	Date and Time manipulation functionality	1. Date Time class must be implemented 2. getCurrentDate() method should be implemented	1. Call the "getCurrentDate" method to retrieve the current date and format it as "dd/MM/yyyy". 2. Then compare the result with the expected date format string.	1. An instance of the DateTime class representing current date 2. Expected date format of dd/MM/yyyy (13/03/2023)	High	13/03/2023	13/03/2023	Passed	If the formatting of the date and if the date mathces with the current date then the test is successful	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0002	Check whether the "getFormattedDate" method of the "DateTime" class returns the specific date in the specified format.	Date and Time manipulation functionality	1. Date Time class must be implemented 2. getFormattedDate() method should be implemented	1. Call the "getFormattedDate" method to retrieve the specific date and format it as "dd/MM/yyyy". 2. Then compare the result with the expected date format string.	1. An instance of the DateTime class representing specific date 2. Expected date format of dd/MM/yyyy (15/03/2024)	Medium	15/03/2024	15/03/2024	Passed	If the formatting of the date and if the date mathces with the specified date then the test is successful	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0003	Check whether the method that calculates difference between two dates correctly	Date and Time manipulation functionality	1. Date Time class must be implemented 2. diffDays() method should be implemented	1. Instantiate two objects of the DateTime class 2. Call the "diffDays" method to get the difference between two dates in days 3. Then compare the result with the expected date format string.	1. Two instance of the DateTime class representing specific date 2. Start Date: 10/03/2024 3. End Date: 20/03/2024	Medium	10	10	Passed	If the difference is correct, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0004	Check whether the eight digit date functionality works accordingly or not	Date and Time manipulation functionality	1. Date Time class must be implemented 2. getEightDigitDate() method should be implemented	1. Call the "getEightDigitDate" method to get the specific date in eight digit format 2. Then compare the result with the expected date format string.	1. An instance of the DateTime class representing specific date 2. 25/03/2024	Low	25032024	25032024	Passed	If the formatting is in eight digits and matches with the date, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0005	Check whether getting the correct day from a specific date works or not	Date and Time manipulation functionality	1. Date Time class must be implemented 2. getNameOfDay() method should be implemented	1. Call the "getNameOfDay" method to get the specific day of the given date 2. Then compare the result with the expected date format string.	1. An instance of the DateTime class representing specific date 2. 25/03/2024	Low	Monday	Monday	Passed	If the output is the correct date, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0006	Check whether the default constructor of the DateTime class gets properly called or not	Date and Time manipulation functionality	1. Date Time class must be implemented 2. The class should have a default constructor implemented	1. Instantiate an object of the DateTime class 2. Then compare the result with the current date and the "getTime" method of the DateTime class	1. An instance of the DateTime class representing current date	Medium	TRUE	TRUE	Passed	If the constructor is implemented correctly, the test will be successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0007	Check whether the Vehicle ID returns the correct ID	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. getVehicleId() method should be implemented	1. Instantiate an object of the Vehicle class with initial values 2. Then compare the result with the returned value of "getVehicleId" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: C_001, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Medium	C_001	C_001	Passed	If the method returns the correct VehicleId, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0008	Check whether the rent functionality works for a specific vehicle given that the vehicle is availibile for renting	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. rent() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the number of rent days would be greater than zero 3. Then compare the result with the returned value of "rent()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: C_002, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. customerId: customer1, day: 10, month: 3, year: 2024, numOfRentDay: 5	Medium	TRUE	TRUE	Passed	If the method returns indicating that vehicle can be rented, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0009	Check whether the rent functionality works for a specific vehicle given that the vehicle is not availibile for renting	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. rent() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the number of rent days would be greater than zero 3. Then compare the result with the returned value of "rent()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: C_003, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. customerId: customer2, day: 10, month: 3, year: 2024, numOfRentDay: 0	Medium	TRUE	TRUE	Passed	If the method indicates that vehicle can not be rented, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0010	Check whether the vehicle maintanance functionality works for a specific vehicle given that the vehicle is available (Not rented and also not under maintanance)	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. performMaintenance() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the status would be zero 3. Then compare the result with the returned value of "performMaintenance()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: V_001, Year: 2022, Make: Vapid, Model: Speedo, Status:0, Vehicle Type: {Seats:7}	Medium	TRUE	TRUE	Passed	If the method indicates that vehicle can be put under maintanance, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0011	Check whether the vehicle maintanance functionality works for a specific vehicle given that the vehicle is rented or is already under maintanance	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. performMaintenance() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the status would be zero 3. Then compare the result with the returned value of "performMaintenance()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: V_001, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:7}	Medium	TRUE	TRUE	Passed	If the method indicates that vehicle can not be put under maintanance, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0012	Check whether the toString says a vehicle is available when it's actually available	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. toString() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the status would be set to zero (Indicating that it's available) 3. Then compare the result with the returned value of "toString()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: V_005, Year: 2022, Make: Vapid, Model: Speedo, Status:0, Vehicle Type: {Seats:8}	Medium	V_005:2022:Vapid:Speedo:15:Available	C_005:2022:Karin:Kurama:5:Available	Passed	If the toString method indicates that vehicle is available, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0013	Check whether the toString says a vehicle is Rented when it's actually Rented	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. toString() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. In this case the status would be set to one (Indicating that it's Rented) 3. Then compare the result with the returned value of "toString()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: V_002, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:8}	Medium	V_002:2022:Vapid:Speedo:15:Rented	V_002:2022:Vapid:Speedo:15:Rented	Passed	If the toString method indicates that vehicle is rented, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0014	Check whether the details of a rented vehicle is correct	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. getDetails() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: C_006, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Low	Vehicle ID:C_006 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available	Vehicle ID:C_006 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available	Passed	If the details are shown in the correct way with correct details, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0015	Check whether the details of a not rented vehicle is correct	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. getDetails() method should be implemented	1. Instantiate an object of the Vehicle class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: V_003, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:8}	Low	Vehicle ID: C_006 Year: 2022 Make: Karin Model: Kurama Number of Seats: 5 Status: Available	Vehicle ID: C_006 Year: 2022 Make: Karin Model: Kurama Number of Seats: 5 Status: Available	Passed	If the details are shown in the correct way with correct details, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0016	Check whether the last index of the Rental Records returns the correct element of the Vehical Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Rental Record class must be implemented 4. getLastElementIndex() method should be implemented	1. Instantiate an object of the Vehicle class 2. Then compare the result with the returned value of "getLastElementIndex()" method of the Vehicle class	1. An instance of the Vehicle class representing a vehicle 2. VehicleId: C_007, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Low	-1	-1	Passed	If the last element is indeed in the last index, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0017	Check the late fee calculation for car where the return has been due for positive amount of days in Car class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. getLateFee() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "getLateFee()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_001, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. Start Date: 10/03/2024 4. End Date: 15/03/2024	Medium	487.5	487.5	Passed	If the late fees match, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0018	Check the late fee calculation for car where the return has been not been due at all in Car class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. getLateFee() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "getLateFee()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_002, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. Start Date: 10/03/2024 4. delta: 0.01	Low	0	0	Passed	If the late fees match, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024

Test Case ID	Test Case Summary	Related Requirement	Pre-Conditions	Test Procedure	Test Data	Priority	Expected Result	Actual Result	Status	Remarks	Test Case Author	Executed By	Test Environment	Frequency	Reporting Date
TC-0019	Check whether the car return functionality works if the car can actually be returned in the Car Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. returnVehicle() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "returnVehicle()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_002, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. records[0]: [RentId: R_001, { day: 15, month: 3, year: 2024}, { day: 20, month: 3, year: 2024}]	Medium	TRUE	TRUE	Passed	If the car can be returned, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0020	Check whether the car return functionality works if the car can not be returned in the Car Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. returnVehicle() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "returnVehicle()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_002, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. { day: 15, month: 3, year: 2024}	Medium	TRUE	TRUE	Passed	If the car can not be returned, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0021	Check whether the car maintanance functionality works if the car actually be put into maintanance in the Car Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. completeMaintenance() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "completeMaintenance()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_005, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Medium	TRUE	TRUE	Passed	If the car can be put into maintanance, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0022	Check whether the car maintanance functionality works if the car not be put into maintanance in the Car Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. completeMaintenance() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "completeMaintenance()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_006, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Low	TRUE	TRUE	Passed	If the car can not be put into maintanance, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0023	Check whether details gotten from an empty rental record yeilds correct response	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. getDetails() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_007, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5}	Medium	Vehicle ID:C_007 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Vehicle ID:C_007 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0024	Check whether details gotten from a non empty (populated) rental record yeilds correct response	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. getDetails() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_008, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. {day: 15, month: 3, year: 2024}	Medium	Vehicle ID:C_008 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Vehicle ID:C_008 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0025	Check whether details gotten from a null rental record yeilds correct response	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. getDetails() method should be implemented 4. Car class must be implemented	1. Instantiate an object of the Car class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Car class	1. An instance of the Car class representing a Car 2. VehicleId: C_010, Year: 2022, Make: Karin, Model: Kurama, Status:0, Vehicle Type: {Seats:5} 3. car.records[0]: null	Low	Vehicle ID:C_010 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Vehicle ID:C_010 Year: 2022 Make:Karin Model:Kurama Number of Seats:5 Status:Available RENTAL RECORD:empty	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0026	Check the late fee calculation for Van where the return has been due for positive amount of days in Van class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. getLateFee() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "getLateFee()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_001, Year: 2022, Make: Vapid, Model: Speedo, Status:0, Vehicle Type: {Seats:7} 3. Start Date: 15/03/2024 4. End Date: 20/03/2024	High	1495	1495	Passed	If the late fees match, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0027	Check whether the Van return functionality works if the Van can actually be returned in the Van Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. returnVehicle() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "returnVehicle()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_002, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:7} 3. records[0]: [RentId: R_001, { day: 15, month: 3, year: 2024}, { day: 20, month: 3, year: 2024}]	Medium	TRUE	TRUE	Passed	If the Van can be returned, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0028	Check whether the Van maintanance functionality works if the Van actually be put into maintanance in the Van Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. completeMaintenance() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "completeMaintenance()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_004, Year: 2022, Make: Vapid, Model: Speedo, Status:2, Vehicle Type: {Seats:8}	Medium	TRUE	TRUE	Passed	If the Van can be put into maintanance, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0029	Check Whether the toString method provides matching details for a particular van of the Van class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. toString() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "toString()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_006, Year: 2022, Make: Vapid, Model: Speedo, Status:0, Vehicle Type: {Seats: 10, {day: 10, month: 3, year: 2024}}	Medium	V_006:2021:Vapid:Speedo: 15:Available:01/03/2024	V_006:2021:Vapid: Speedo:15:Available: 01/03/2025	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0030	Check whether details gotten from a non empty (populated) rental record yeilds correct response for the Van Class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. getDetails() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_002, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:9, {day: 10, month: 3, year: 2024}} 3. records[0]: [RentId: R_001, { day: 15, month: 3, year: 2024}, { day: 20, month: 3, year: 2024}]	Medium	Vehicle ID:V_008 Year: 2022 Make:Vapid Model:Speedo Number of Seats:8 Status:Available Last maintenance date: 10/03/2024 RENTAL RECORD: Record ID: R_001 Rent Date: 15/03/2024 Estimated Return Date: 20/03/2024 ----------	Vehicle ID:V_008 Year: 2022 Make:Vapid Model:Speedo Number of Seats:8 Status:Available Last maintenance date: 10/03/2024 RENTAL RECORD: Record ID: R_001 Rent Date: 15/03/2024 Estimated Return Date: 20/03/2024 ----------	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024
TC-0031	Check whether details gotten from a null rental record yeilds correct response for the Van class	Vehicle Renting System	1. Vehicle Type class must be implemented 2. Vehicle class must be implemented 3. Van class must be implemented 4. getDetails() method should be implemented	1. Instantiate an object of the Van class with initial values. 2. Then compare the result with the returned value of "getDetails()" method of the Van class	1. An instance of the Van class representing a Van 2. VehicleId: V_002, Year: 2022, Make: Vapid, Model: Speedo, Status:1, Vehicle Type: {Seats:9, {day: 10, month: 3, year: 2024}}	Low	Vehicle ID:V_008 Year: 2022 Make:Vapid Model:Speedo Number of Seats:8 Status:Available Last maintenance date: 10/03/2024 RENTAL RECORD: Record ID: R_001 Rent Date: 15/03/2024 Estimated Return Date: 20/03/2024 ----------	Vehicle ID:V_008 Year: 2022 Make:Vapid Model:Speedo Number of Seats:8 Status:Available Last maintenance date: 10/03/2024 RENTAL RECORD: Record ID: R_001 Rent Date: 15/03/2024 Estimated Return Date: 20/03/2024 ----------	Passed	If the response matches, the test is successful.	Mashrur Ahsan	Mashrur Ahsan	IDE (Integrated Development Environment): IntelliJ IDEA Build Tool: Apache Maven Testing Framework: JUnit 4 Operating System: Windows 11 Java Development Kit (JDK): JDK 21	1	13 March 2024