TicketTest

1. **testGetTicketID**:
   - Verifies that getTicketID correctly retrieves the ticket ID.
2. **testGetPrice**:
   - Checks if getPrice returns the correct initial price.
3. **testSetPrice**:
   - Ensures setPrice correctly updates the ticket price.
4. **testIsPaidStatus**:
   - Validates that isPaidStatus initially returns false.
5. **testSetPaidStatus**:
   - Ensures setPaidStatus updates the payment status correctly.
6. **testGetTimeArrived**:
   - Confirms that getTimeArrived provides a valid timestamp in ISO format.
7. **testSetTimeArrived**:
   - Ensures setTimeArrived updates the arrival time correctly.
8. **testCalculatePriceWithRate**:
   - Validates that calculatePrice calculates the total price based on hours parked and rate.
9. **testPrintTicket**:
   - Confirms printTicket outputs the ticket details to the console.

ManagerLogInTest

1. **testConstructorInitialization**:
   - Verifies that the constructor initializes fields correctly.
2. **testSetStatus**:
   - Ensures setStatus updates the manager's status.
3. **testValidateLogin_Valid**:
   - Tests validateLogin with a valid login, ensuring the status becomes "active."
4. **testValidateLogin_Invalid**:
   - Tests validateLogin with an invalid login, ensuring the status remains "inactive."
5. **testSelectGarage_Active**:
   - Ensures that a manager can select a garage when logged in.
6. **testSelectGarage_Inactive**:
   - Verifies that the system prompts the manager to log in if they attempt to select a garage without being active.
7. **testSelectGarage_InvalidState**:
   - Tests behavior when the manager has an invalid status.

GarageTest

1. **testParkVehicle**:
   - Confirms that vehicles can be parked if there is space available.
2. **testRemoveVehicle**:
   - Verifies that a parked vehicle can be removed.
3. **testCheckSpace**:
   - Checks if checkSpace accurately reflects availability.
4. **testGetCapacity**:
   - Ensures getCapacity returns the correct garage capacity.
5. **testGetSpacesTaken**:
   - Verifies the count of occupied spaces.
6. **testParkVehicleWhenFull**:
   - Ensures that no additional vehicles can be parked when the garage is full.
7. **testRemoveVehicleWhenEmpty**:
   - Confirms that vehicles cannot be removed from an empty garage.
8. **testCheckSpaceWhenFull**:
   - Validates that checkSpace returns false when the garage is full.
9. **testCheckSpaceWhenEmpty**:
   - Ensures that checkSpace returns true when the garage is empty.

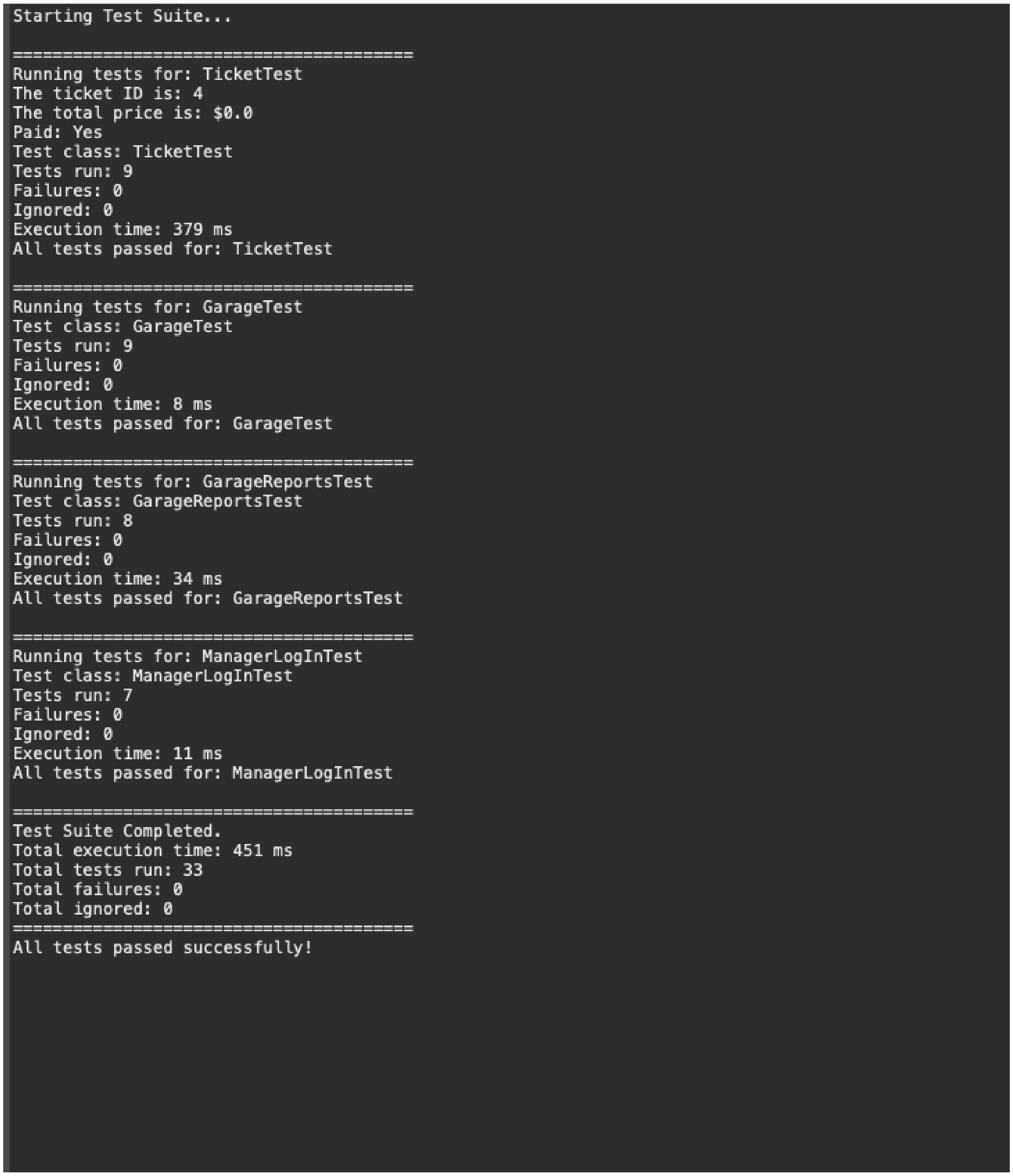
GarageReportsTest

1. **testGetDate**:
   - Verifies getDate provides a valid timestamp in ISO format.
2. **testGetID**:
   - Ensures getID returns a valid ID prefixed with "GR-".
3. **testGetSummary**:
   - Confirms getSummary is empty initially.
4. **testSetCarTracker**:
   - Validates that the car tracker increments correctly.
5. **testUpdatePaymentSummary**:
   - Ensures the payment summary updates with the correct value.
6. **testGetCarTracker**:
   - Confirms getCarTracker reflects the accurate number of cars tracked.
7. **testToString**:
   - Verifies that toString provides a detailed, accurate representation of the report.
8. **testSerialization**:
   - Tests that a GarageReports object can be serialized and deserialized correctly.

4. Summary of Tests

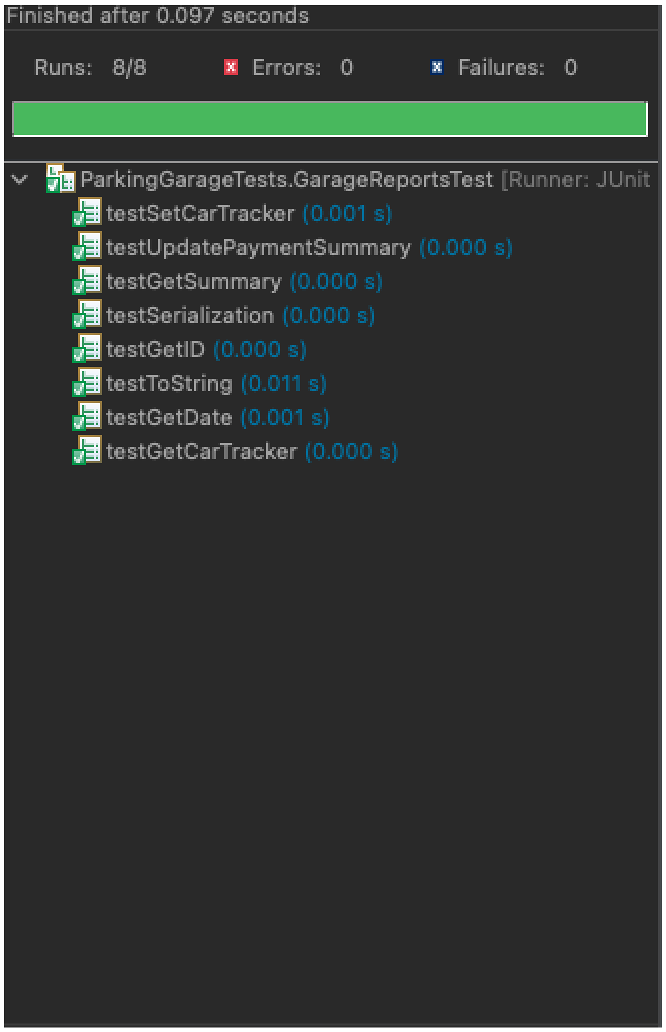
- The test cases cover the core functionality and edge cases of the Ticket, ManagerLogIn, Garage, and GarageReportsclasses. These tests validate that each class operates as intended, including correct initialization, method functionality, and error handling. By covering a wide range of scenarios, the tests ensure robustness and reliability of the system, particularly for real-world parking garage management applications. The comprehensive suite verifies system consistency, data integrity, and appropriate user interactions.

Test showcase:



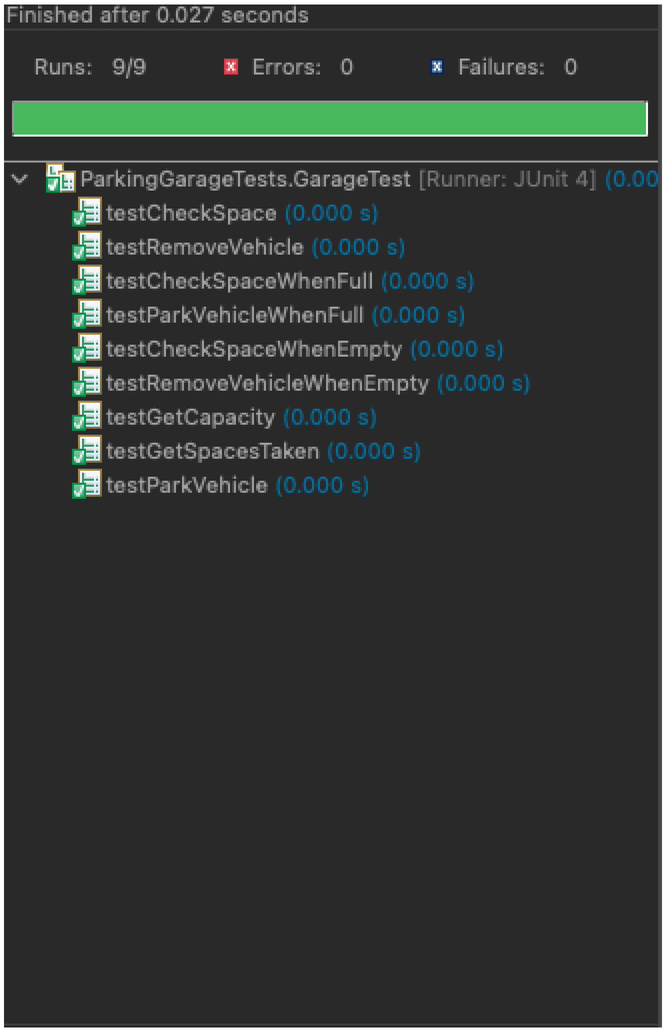
```
Starting Test Suite...

========================================
Running tests for: TicketTest
The ticket ID is: 4
The total price is: $0.0
Paid: Yes
Test class: TicketTest
Tests run: 9
Failures: 0
Ignored: 0
Execution time: 379 ms
All tests passed for: TicketTest

========================================
Running tests for: GarageTest
Test class: GarageTest
Tests run: 9
Failures: 0
Ignored: 0
Execution time: 8 ms
All tests passed for: GarageTest

========================================
Running tests for: GarageReportsTest
Test class: GarageReportsTest
Tests run: 8
Failures: 0
Ignored: 0
Execution time: 34 ms
All tests passed for: GarageReportsTest

========================================
Running tests for: ManagerLogInTest
Test class: ManagerLogInTest
Tests run: 7
Failures: 0
Ignored: 0
Execution time: 11 ms
All tests passed for: ManagerLogInTest

========================================
Test Suite Completed.
Total execution time: 451 ms
Total tests run: 33
Total failures: 0
Total ignored: 0
========================================
All tests passed successfully!
```

GarageReportsTest



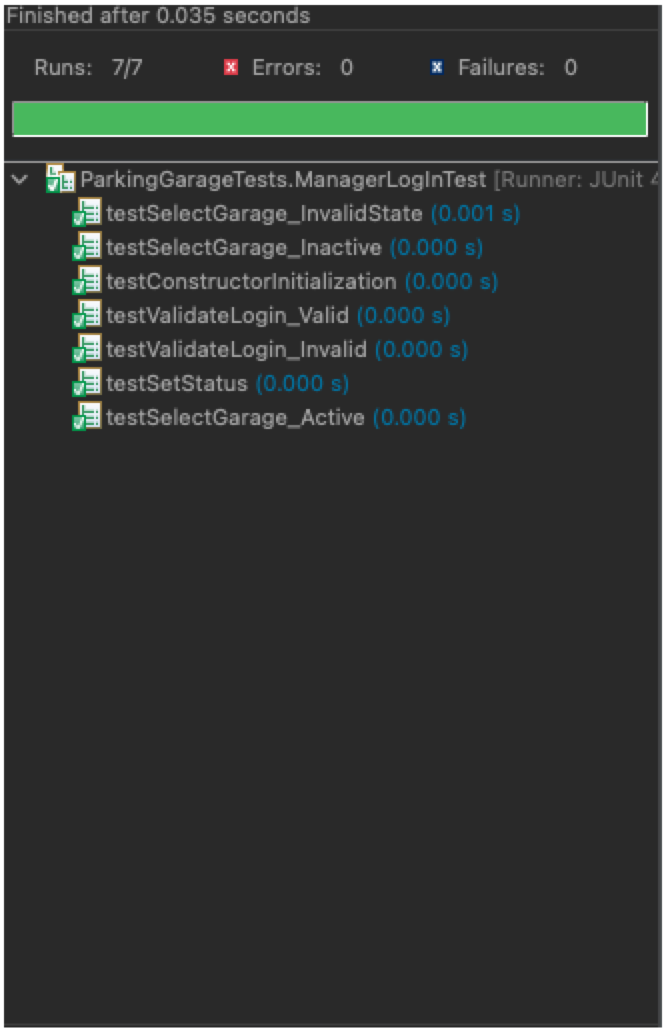
```
Finished after 0.097 seconds

Runs:  8/8      Errors:  0      Failures:  0

ParkingGarageTests.GarageReportsTest [Runner: JUnit
    testSetCarTracker (0.001 s)
    testUpdatePaymentSummary (0.000 s)
    testGetSummary (0.000 s)
    testSerialization (0.000 s)
    testGetID (0.000 s)
    testToString (0.011 s)
    testGetDate (0.001 s)
    testGetCarTracker (0.000 s)
```

GarageTest



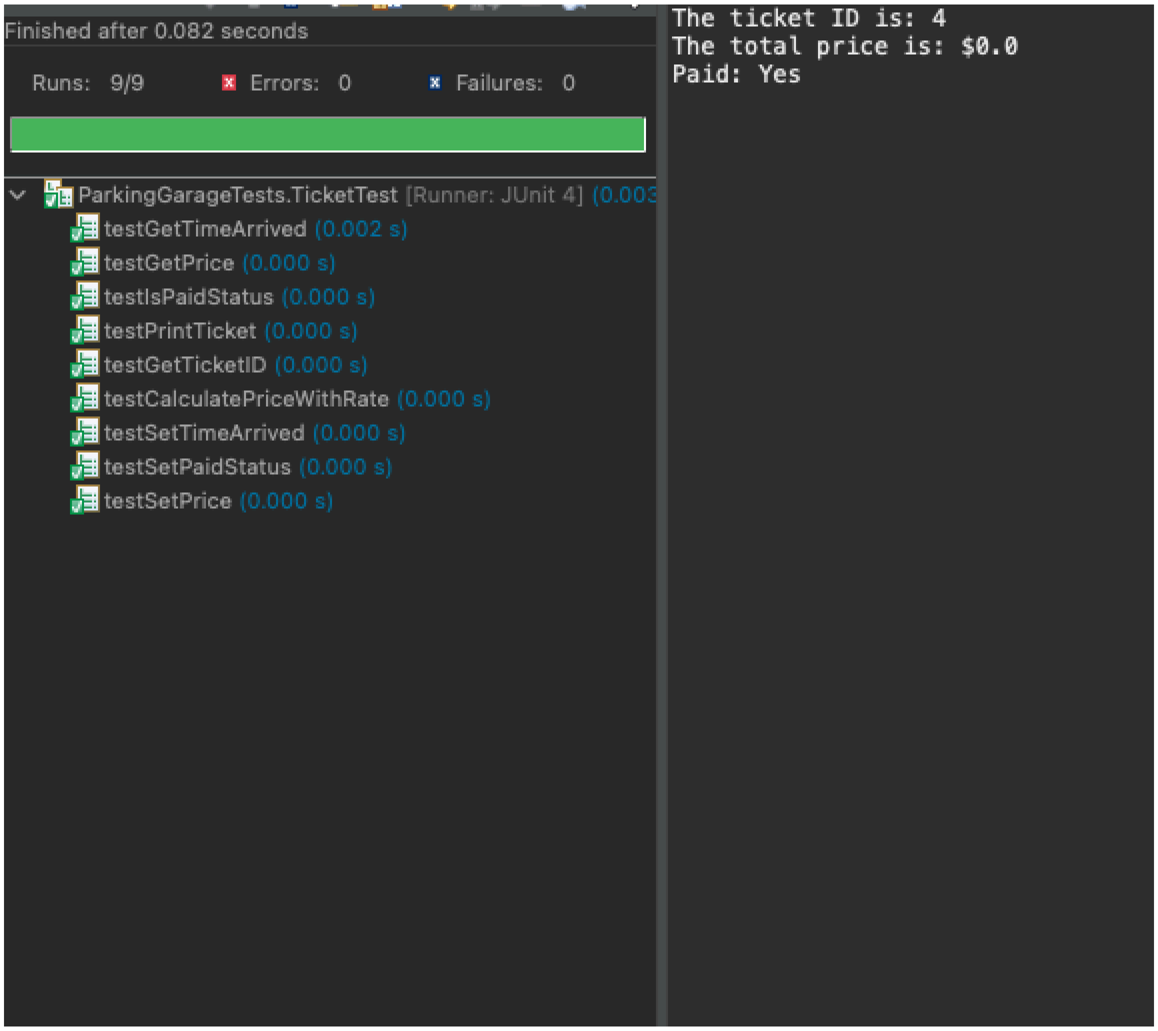
```
Finished after 0.027 seconds

Runs:  9/9      Errors:  0      Failures:  0

ParkingGarageTests.GarageTest [Runner: JUnit 4] (0.00
    testCheckSpace (0.000 s)
    testRemoveVehicle (0.000 s)
    testCheckSpaceWhenFull (0.000 s)
    testParkVehicleWhenFull (0.000 s)
    testCheckSpaceWhenEmpty (0.000 s)
    testRemoveVehicleWhenEmpty (0.000 s)
    testGetCapacity (0.000 s)
    testGetSpacesTaken (0.000 s)
    testParkVehicle (0.000 s)
```

ManagerLoginTest



```
Finished after 0.035 seconds

Runs: 7/7        Errors: 0        Failures: 0

ParkingGarageTests.ManagerLogInTest [Runner: JUnit 4
    testSelectGarage_InvalidState (0.001 s)
    testSelectGarage_Inactive (0.000 s)
    testConstructorInitialization (0.000 s)
    testValidateLogin_Valid (0.000 s)
    testValidateLogin_Invalid (0.000 s)
    testSetStatus (0.000 s)
    testSelectGarage_Active (0.000 s)
```

TicketTest


```
Finished after 0.082 seconds

Runs: 9/9    Errors: 0    Failures: 0

ParkingGarageTests.TicketTest [Runner: JUnit 4] (0.003
    testGetTimeArrived (0.002 s)
    testGetPrice (0.000 s)
    testIsPaidStatus (0.000 s)
    testPrintTicket (0.000 s)
    testGetTicketID (0.000 s)
    testCalculatePriceWithRate (0.000 s)
    testSetTimeArrived (0.000 s)
    testSetPaidStatus (0.000 s)
    testSetPrice (0.000 s)
```

```
The ticket ID is: 4
The total price is: $0.0
Paid: Yes
```