Design Documentation

Distributed Vehicle Reservation Management System

(DVRMS)

By Yichen Huang 40167688

1. System Overview and Architecture

The DVRMS is a distributed system composed of three office servers (MTL, WPG, BNF) and two client roles (Manager and Customer). Clients communicate with their home office server via Java RMI, while inter-server communication uses UDP for loose coupling and scalability.

By restricting each client to communicate only with its home office server, the system simplifies client-side logic and centralizes cross-office coordination within the server layer.

2. Role Separation and Access Control

Role separation is enforced using user ID patterns. Managers can add, remove, and list vehicles, while Customers can reserve, update, cancel, search vehicles, and check budgets. Validation is performed at both client and server sides.

3. Data Structures Design

VehicleRecord objects maintain vehicle information, available quantity, active reservations, and a FIFO waiting list. Reservation records store customer and date information, while waiting list entries queue pending requests fairly.

Each vehicle maintains its own reservation list and waiting list, which localizes synchronization and ensures fairness when handling concurrent reservation requests.

4. Concurrency Control

Critical sections related to reservation creation, inventory updates, cancellations, and waiting list assignment are synchronized at the vehicle-record level to prevent race conditions while allowing parallel access to different vehicles.

5. Inter-Server Communication (UDP)

UDP is used for cross-office vehicle search and remote reservation checks. The home server aggregates responses from other offices before returning results to the client.

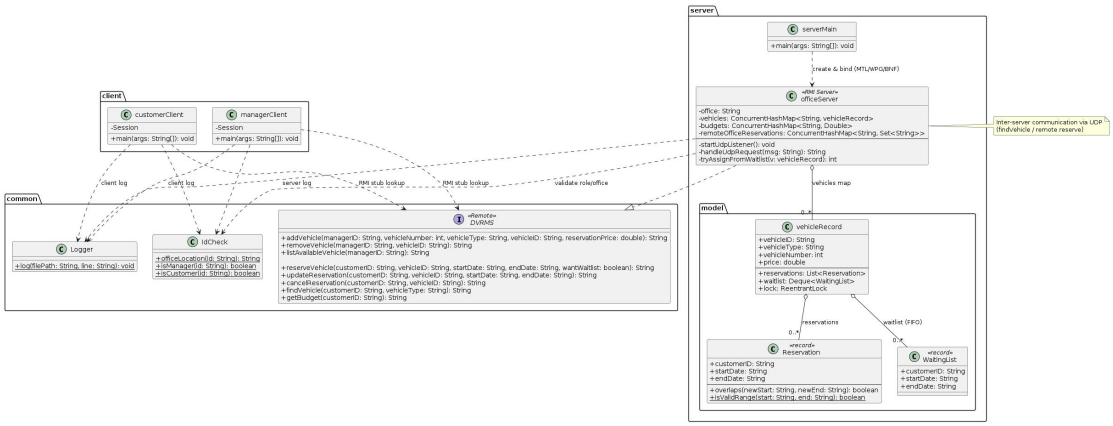
6. Test Scenarios

The system is validated through local operations, remote operations, budget enforcement, date validation, and integrated inventory and waiting list scenarios.

7. Most Important and Difficult Part

The integration of inventory management, waiting list logic, and concurrency control was the most challenging aspect of the assignment. Correctness and fairness had to be ensured under concurrent access.

8.UML Diagram



9. Sequence Flow and Edge Case Handling (Enhanced)

This section provides an end-to-end sequence explanation of a typical reservation scenario involving inventory exhaustion and waiting list assignment. When a customer submits a reservation request, the home server first validates the request and checks local inventory. If sufficient quantity exists and no date conflict is detected, the reservation is confirmed and inventory is decremented. If the vehicle is unavailable due to exhausted inventory, the request may be added to the waiting list. When a cancellation or vehicle removal occurs, the server automatically checks the waiting list and assigns the vehicle to the first eligible request.

The system also handles several edge cases, including invalid date ranges, insufficient customer budget, duplicate reservation attempts, and remote office reservation limits. Each edge case is explicitly validated to ensure system robustness and predictable behavior under erroneous or conflicting requests.

10.Testing Execution and Results

Server Main
Local: MTL, WPG, BNF

Manager Test
--- MTL add ---
vehicle number: 1, vehicle ID: MTL1001, vehicle type: Sedan, reservation price: $120.0
vehicle number: 1, vehicle ID: MTL1002, vehicle type: Sedan, reservation price: $130.0
vehicle number: 2, vehicle ID: MTL9999, vehicle type: Truck, reservation price: $1500.0
MTL list:
MTL9999 Truck 1500.0
MTL1001 Sedan 120.0
MTL1002 Sedan 130.0

--- WPG add ---
vehicle number: 1, vehicle ID: WPG2001, vehicle type: Sedan, reservation price: $110.0
vehicle number: 1, vehicle ID: WPG2002, vehicle type: Sedan, reservation price: $115.0
vehicle number: 5, vehicle ID: WPG2003, vehicle type: SUV, reservation price: $210.0
WPG list:
WPG2001 Sedan 110.0
WPG2003 SUV 210.0
WPG2002 Sedan 115.0

--- BNF add ---
vehicle number: 1, vehicle ID: BNF3001, vehicle type: Sedan, reservation price: $105.0
vehicle number: 1, vehicle ID: BNF3002, vehicle type: Sedan, reservation price: $108.0
vehicle number: 1, vehicle ID: BNF3003, vehicle type: SUV, reservation price: $220.0
BNF list:
BNF3003 SUV 220.0
BNF3002 Sedan 108.0
BNF3001 Sedan 105.0

Customer Test

Customer ID: MTLU1111

--- Local Testing---
1. Reserve MTL1001 01-05 Feb 2026 - Success
2. Reserve MTL1001 03-04 Feb 2026 - Fail (date overlap)
3. Update reserve date MTL1001 06-07 Feb 2026 - Success
4. Update reserve date MTL1001 08-97 Feb 2026 - Fail (end date > start date)
5. Update reserve date MTL1001 999999 - Fail (Invalid date format)
6. Cancel reservation MTL1001 - Success
7. Reserve after cancel MTL1111 06-07 Feb 2026 - Success
8. Reserve after cancel MTL1001 03-04 Feb 2026 - Fail (auto-assigned to waiting customer)
9. Reserve MTL9999 - Fail (Customer budget not enough)
10. Cancel reservation MTL1002 - Fail (non-existent reservation)
11. Reserve MTL1001 32-13 2026, 01 Feb 2026 - Fail (invalid date)

--- UDP Testing---
1. FIND Sedan across 3 offices
2. Remote reserve WPG2001 06-07 Feb 2026 - Success
3. Remote reserve WPG2002 08-09 Feb 2026 - Fail (only 1 per remote office)
4. Remote reserve BNF3001 10-11 Feb 2026 - Success
5. Remote reserve BNF3002 12-13 Feb 2026 - Fail (only 1 per remote office)

--- Waiting List Testing---
Customer A = MTLU1111
Customer B = MTLU2222

1. A reserve MTL1001 01-05 Mar 2026 - Success
2. B reserve MTL1001 01-05 Mar 2026 - Fail (can't reserve same slot, add in waiting list)
3. A cancel MTL1001 - Success (Auto-assigned B from waiting list)
4. A reserve MTL1001 01-05 Mar 2026 - Fail (B already assigned to the car)
5. A reserve MTL1002 01-05 Apr 2026 - Success
6. B reserve MTL1002 01-05 Apr 2026 - Fail (can't reserve same slot, add in waiting list)
7. Manager MTLM1111 add inventory for MTL1002 -(no auto-assignment, capacity unchanged)
8. Manager MTLM1111 create new MTL2000 - Success
9. A reserve MTL2000 01-05 Apr 2026 - Success

--- Manager remove Testing---
Customer C = MTLU3333
Customer D = MTLU4444

1. Manager MTLM1111 create MTL2001 - Success
2. C reserve MTL2001 01-05 Jun 2026 - Success
3. D reserve MTL2001 01-05 Jun 2026 - Fail (can't reserve same slot, add in waiting list)
4. Manager MTLM1111 remove MTL2001 - Success (active reservation refunded, waiting list requests removed)
5. C reserve MTL2001 - Fail (no such car)

Manager Client
--- Manager adding Testing---
ManagerID MTLM1111
1. Add/update MTL1001, sedan, 1, 120 - Success with list
2. Add/update MTL1002, sedan, 1, 130 - Success with list
3. Add/update MTL3333, sedan, 1, 100 - Success with list
4. Add/update MTL5555, sedan, 2, 100 - Success with list
5. Add/update MTL9999, truck, 3, 1500 - Success with list

Switching ManagerID WPGM1111
6. Add/update WPG2001, sedan, 1, 110 - Success with list
7. Add/update WPG2002, sedan, 1, 115 - Success with list
8. Add/update WPG2003, suv, 2, 210 - Success with list

Switching ManagerID BNFM1111
9. Add/update BNF3001, sedan, 1, 105 - Success with list
10. Add/update BNF3002, sedan, 1, 108 - Success with list

11. Add/update BNF3003, suv, 2, 220 - Success with list

--- Manager removing Testing---
ManagerID MTLM1111

Customer Client
--- Local Testing---
Customer ID: MTLU1111
1. Check Budget (5) - show 1000
2. Reserve MTL1001, 01022026, 05022026 - Success
3. Reserve MTL1001, 03022026, 04022026 - Fail (date overlap)
4. Reserve MTL9999 - Fail (budget not enough)
5. Update MTL1001, 06022026, 07022026 - Success
6. Update MTL1001, 08022026, 07022026 - Fail (invalid date range)
7. Update MTL1001, 999999, 999999 - Fail (invalid date format)
8. Cancel MTL1001 - Success
9. Cancel MTL1002 - Fail (non-existent reservation)

--- UDP Testing---
1. Find sedan - list all 3 server
2. Reserve WPG2001, 06022026, 07022026 - Success
3. Reserve WPG2002, 08022026, 09022026 - Fail (only 1 per remote office)
4. Reserve BNF3001, 10022026, 11022026 - Success

--- Waiting List Testing---
1. Reserve MTL1001, 01032026, 05032026 - Success
2. Switch user MTLU2222 - show budget
3. Reserve MTL1001, 01032026, 05032026 - Fail
4. Asking to join waiting list - yes
5. Switching user MTLU1111 - show budget
6. Cancel MTL1001 - Success
7. Reserve MTL1001 - Fail (already assigned to MTLU2222)
8. Reserve MTL5555, 01042026, 05042026 - Success
9. Switch user MTLU2222 - show budget
10. Reserve MTL5555, 01042026, 05042026 - Success
11. Switch user MTLU3333 - show budget
12. Reserve MTL5555, 01042026, 05042026 - Fail (unavailable quantity = 0)
13. Asking to join waiting list - no

--- Manager remove Testing---
Customer C = MTLU3333
Customer D = MTLU4444

1. Reserve MTL3333, 01052026, 05052026 - Success
2. Switching user MTLU4444 - show budget
3. Reserve MTL3333, 01052026, 05052026 - Fail
4. Asking to join waiting list - yes
5. (managerClient) MTLM1111 remove MTL3333