



The Superior University

Operating Systems Lab – Project Documentation Template

Parking Slot Management System

Using FCFS

Group Members

 *Project can be completed individually or in a group of up to 4 members.*

List all team members with roll numbers:

- Chashman Javed (226)
- Yusra Fatima (254)
- Asima Allah Ditta (230)
- Ali Haider (233)

GitHub Repository

Both the Python code file and this documentation must be uploaded to a public GitHub repository.

GitHub Repository Link:

<https://github.com/Chashman312/Parking-Slot-system.-py.git>

🔧 Scheduling Algorithm Implemented

Tick the scheduling algorithm your group implemented:

- FCFS (First Come First Serve)
- SJF (Shortest Job First – Non-Preemptive)
- SJF (Preemptive)
- Round Robin

📄 Project Description

This project simulates a parking slot management system where vehicles are parked using the **First Come First Serve (FCFS)** principle.

- **Problem Solved:**

Efficiently manage limited parking slots and a waiting queue using FCFS order.

- **Inputs Required:**

- Total number of parking slots
- Vehicle plate number (at arrival)
- Slot number (at departure)

- **Outputs Generated:**

- Parking confirmation or waitlist message
- Slot release confirmation
- Waiting vehicle assignment
- Slot status and waiting queue display

- **Algorithm Implementation:**

When a vehicle arrives:

- If a slot is available, assign it.
- If all slots are full, add to the queue (FCFS).

When a vehicle departs:

- Mark the slot as free.
- Assign it to the next vehicle in the waiting queue, if any.

Output Screenshots

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Enter total number of parking slots: 3

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 1
Enter vehicle plate number: 789
Vehicle 789 parked at slot 1.

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 1
Enter vehicle plate number: 567
Vehicle 567 parked at slot 2.

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 1
Enter vehicle plate number: 123
Vehicle 123 parked at slot 3.
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 2
Enter slot number to vacate: 2
Slot 2 is now free.

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 3

--- Parking Slot Status ---
Slot 1: Occupied
Slot 2: Free
Slot 3: Occupied

--- Waiting Queue ---
No vehicles waiting.

--- Menu ---
1. Arrive Vehicle
2. Depart Vehicle
3. Display Parking Lot Status
4. Exit
Enter your choice: 4
Exiting system.
```

Code Structure & Explanation

Classes Used:

- o Vehicle: stores vehicle_id and plate_number.
- o Parkingslot: manages parking logic, queue, and slots.

Core Functions:

- o arrive_vehicle(): Handles vehicle entry.
- o depart_vehicle(): Handles vehicle exit.

- o `display_status()`: Shows current slot and queue state.

Libraries Used:

- o `collections.deque`: For efficient queue operations.

Performance Metrics

Metric	Value
Average Waiting Time	Based on queue size
Average Turnaround Time	Not Applicable
Time Quantum (if RR)	N/A

Challenges Faced

- Solved by using a temporary queue to filter out departing vehicles.
- Added proper validation and messaging for out-of-range or free slots.