

Plus Points in Implementation (Overall Evaluation Criteria)

1. Authentication:

- Implement robust user authentication protocols to ensure secure access.

2. Cost Estimation - Time and Space:

- Conduct a thorough analysis of time and space complexity in the system.
- Utilize efficient algorithms and data structures to optimize both time and space requirements.

3. Handling System Failure Cases:

- Implement fault-tolerant mechanisms to address system failures.
- Employ backup and recovery strategies for data integrity.
- Develop comprehensive error recovery procedures to minimize downtime.

4. Object-Oriented Programming Language (OOPS):

- Choose a robust OOPS language for structured and modular code.
- Leverage OOPS principles such as encapsulation, inheritance, and polymorphism for maintainability and extensibility.

5. Trade-offs in the System:

- Clearly define and document trade-offs made during system design.
- Evaluate and communicate the rationale behind architectural and design decisions.
- Consider trade-offs in terms of performance, scalability, and maintainability.

6. System Monitoring:

- Implement comprehensive monitoring tools to track system performance.
- Utilize real-time dashboards and logging mechanisms to promptly identify and address issues.

7. Caching:

- Integrate caching mechanisms to enhance system response times.
- Utilize caching for frequently accessed data to reduce database load.
- Implement cache eviction policies for optimal resource utilization.

8. Error and Exception Handling:

- Develop a robust error and exception handling framework.
- Provide meaningful error messages for effective debugging.
- Regularly review and update error-handling strategies based on system usage patterns.

Instructions:

1. Read and Understand the Problem Statement:

- Carefully read the problem statement provided. Understand the requirements, inputs, expected outputs, and any constraints mentioned.

2. Choose a Programming Language:

- Select a programming language you are comfortable with and that is suitable for solving the problem described in the case study.

3. Design Your Solution:

- Plan the overall structure of your solution. Consider the algorithms, data structures, and any potential optimizations needed.

4. Write the Code:

- Implement your solution in code. Follow best practices for coding standards, such as meaningful variable names, proper indentation, and comments where necessary.
- Break down the problem into smaller functions or modules to improve code readability and maintainability.

5. Test Your Code:

- Test your code thoroughly with different sets of input data, including edge cases and boundary conditions.
- Ensure that your code produces the expected outputs for all test cases.

7. Document Your Code :

- Consider adding documentation or comments to explain the logic and purpose of your code, especially for complex parts or algorithms.

8. Submit Your Solution:

- Once you're satisfied with your code and it meets all the requirements, submit your solution on GitHub and share the GitHub link.

9. Demonstration:

- Include a demonstration video showcasing key features of the ride-sharing platform.
- Alternatively, use screenshots to visually highlight the user interface and functionality.

1. Visitor Management System

A **Visitor Management System (VMS)** plays a crucial role in ensuring workplace security, regulatory compliance, and a seamless visitor experience. This system efficiently tracks and manages visitors while maintaining strict access control. Below is a detailed breakdown of its functionalities:

I. Visitor Registration

When a visitor arrives at the workplace, their details are captured systematically to ensure security and record-keeping.

Information Captured:

- **Full Name** – The visitor's legal name for identification.
- **Contact Information** – Mobile number and/or email for communication and verification.
- **Purpose of Visit** – The reason for their visit (e.g., meeting with an employee, maintenance work, interview, delivery, etc.).
- **Host Employee Details** – Name and department of the employee they are visiting.
- **Company/Organization Name** – If the visitor represents a business, their company details are recorded.
- **Check-in and Check-out Time** – Automatic logging of entry and exit times for security tracking.
- **Mandatory Photo Capture** – A photograph is taken at the registration desk for identity verification.

◆ Process:

1. The **security guard** or **self-service kiosk** collects visitor details and captures their photo.
 2. The system automatically sends a request to the respective host employee for approval.
 3. A visitor badge (physical or digital QR code) is generated after approval.
-

II. Approval Workflow

To prevent unauthorized access, the system ensures that every visitor must be approved by the host employee.

✓ Approval Steps:

1. **Real-Time Notification:** The host employee receives an instant alert (email, SMS, or app notification) regarding the visitor request.
2. **Quick Approval/Rejection:** The employee can approve or deny the visitor's request through the mobile app, web portal, or an automated IVR system.
3. **Access Grant:** Upon approval, the visitor is allowed entry, and a digital or printed visitor pass is issued.
4. **Denied Access:** If rejected, the visitor is not permitted inside, and security is notified to handle the situation accordingly.

◆ Security Benefits:

- Ensures that only authorized visitors can enter the premises.
 - Reduces manual intervention and enhances operational efficiency.
 - Tracks approval history for audit purposes.
-

III. Pre-Approval for Convenience

To minimize last-minute approval delays, employees can pre-approve visitor requests.

✓ Features of Pre-Approval:

- Employees can schedule and approve visitor access in advance for a specific **date and time window** (e.g., 10 AM - 12 PM).
- Pre-approved visitors receive a **QR code/e-pass** via email or SMS, which they can scan upon arrival to bypass the manual approval process.
- If a visitor fails to check in within the approved window, the request **expires automatically** for security purposes.
- Admins can enforce rules, such as **pre-approval limits** (e.g., max 5 visitors per employee per day).

◆ Use Cases:

- Scheduling **client meetings** or **vendor visits** in advance.
 - Fast-tracking **frequent visitors**, such as maintenance personnel.
 - Reducing **employee workload** on the day of the visit.
-

IV. Mandatory Photo Capture

A crucial security measure that ensures every visitor's identity is visually verified before entering the premises.

✓ **Why is it Mandatory?**

- **Identity Verification** – Prevents impersonation by ensuring that the registered visitor matches the person entering.
- **Security Compliance** – Organizations can maintain visual records for auditing and safety purposes.
- **Incident Investigation** – In case of security breaches, captured photos help with tracking and investigation.
- **Badge Generation** – Visitor photos are displayed on digital/printed badges for easy identification by security personnel.

◆ **Process:**

1. The system prompts the security guard to capture a **live photo** of the visitor.
2. The captured image is **stored in a secure database** for future reference.
3. A **visitor badge with a photo** is generated, which can be checked at security checkpoints.
4. The visitor **must check out** upon exit to complete the visit record.