System Design Collection

A comprehensive collection of System Design resources including High-Level Designs (HLD), Low-Level Designs (LLD), and general system design concepts.

📁 Folder Structure

```
SYSTEM DESIGN/
├── LLD/                        # Low-Level Design (Implementation
Code)
│   ├── Design-Patterns/        # Design Pattern Implementations (8
patterns)
│   └── [Complete Systems]      # Full system implementations (10
systems)
├── HLD/                        # High-Level Design (Architecture
Documents)
├── System-Design-Resources/    # General System Design Guides &
Concepts
└── README.md                   # This file
```

📂 LLD - Low-Level Design (18 Files)

🎨 Design-Patterns/ (8 Pattern Files)

Complete Java implementations with multiple real-world examples:

Creational Patterns

- `BuilderPattern.java` - 6 examples (House, Computer, Meal, Report, SQL Query, User Profile builders)
- `FactoryPattern.java` - 5 examples (Database, Notification, Payment, Document, Vehicle factories)
- `SingletonPattern.java` - 6 examples (Connection Pool, Config Manager, Logger, Cache, Thread Pool)

Structural Patterns

- `CompositePattern.java` - 5 examples (File System, Organization, Menu, Graphics, Product Catalog)
- `DecoratorPattern.java` - 5 examples (Coffee Shop, Text Formatting, Pizza, Notifications, Car Features)

Behavioral Patterns

- `ObserverPattern.java` - 5 examples (Email, Stock Market, Weather Station, Social Media, Events)
- `ChainOfResponsibilityPattern.java` - 5 examples (Spam Filter, Support Tickets, Logging, ATM, Auth)
- `StrategyPattern.java` - 5 examples (Sorting, Payment, Compression, Routes, Pricing)

💻 Complete System Implementations (10 Files)

- `ABTestingPlatform.java` - A/B testing platform with experiments and variants
- `ATMSystem.java` - ATM system with accounts, transactions, and state management
- `ElevatorSystem.java` - Elevator control system with scheduling
- `EmailSystem.java` - Email system with folders, attachments, and spam filtering
- `InstagramSystem.java` - Social media platform with posts, comments, follows
- `LoggingFramework.java` - Logging framework with multiple log levels and handlers
- `NotificationSystem.java` - Multi-channel notification system
- `OrderManagementDemo.java` - E-commerce order management system
- `ParkingLotSystem.java` - Parking lot management with multiple vehicle types
- `TwitterSystem.java` - Microblogging platform with tweets, follows, timeline

Total: 43+ Design Pattern Examples + 10 Complete System Implementations

---

📁 HLD - High-Level Design (11 Systems)

Architecture and design documents for scalable distributed systems:

Social & Content Platforms

- `instagram-system-design-HLD.md/pdf` - Photo sharing platform architecture
- `twitter-system-design-HLD.md/pdf` - Microblogging platform with feed generation
- `netflix-video-streaming-system-design-HLD.md/pdf` - Video streaming at scale

Infrastructure & Storage

- `distributed-key-value-store-system-design-HLD.md/pdf` - Distributed KV store like Dynamo/Cassandra
- `google-drive-system-design-HLD.md/pdf` - Cloud file storage and sync
- `url-shortener-system-design-HLD.md/pdf` - URL shortening service like bit.ly

Performance & Reliability

- `rate-limiter-system-design-HLD.md/pdf` - Rate limiting strategies and implementations
- `logging-metrics-system-design-HLD.md/pdf` - Observability infrastructure

Business Systems

- `ab-testing-platform-system-design-HLD.md/pdf` - Experimentation platform
- `marketing-campaign-system-design-HLD.md/pdf` - Campaign management system

Decision Guides

- `system-design-decision-guide-HLD.md/pdf` - Decision framework for system design

Also includes: `ab-testing-platform-system-design-LLD.pdf` - Low-level design complement

📂 System-Design-Resources (9 Guides)

Fundamental concepts, guides, and best practices:

Core Concepts

- `20-system-design-concepts-explained.md/pdf` - 20 essential system design concepts
- `system-design-concepts-deep-dive.md/pdf` - In-depth exploration of key concepts
- `system-design-terminology-guide.md/pdf` - Glossary of system design terms

Practical Guides

- `back-of-envelope-calculations-guide.md/pdf` - Estimation techniques for system capacity
- `database-selection-guide-MEDIUM.md/pdf` - Database selection criteria
- `database-selection-guide-system-design.md/pdf` - Comprehensive database guide

Architecture Patterns

- `rate-limiter-architecture-explained.md/pdf` - Rate limiting patterns and algorithms
- `global-file-storage-system-design.md/pdf` - Global storage system architecture

All guides include: Both Markdown (.md) and PDF (.pdf) versions

---

🎯 Quick Start

For Learning LLD Patterns:

```bash
cd LLD/Design-Patterns/
# Run any pattern file to see examples
java ObserverPattern.java
java FactoryPattern.java
java SingletonPattern.java
```

For Understanding HLD:

```bash
cd HLD/
# Open any system design document
open instagram-system-design-HLD.pdf
open rate-limiter-system-design-HLD.pdf
```

For System Design Fundamentals:

```
cd System-Design-Resources/
# Start with the concepts guide
open 20-system-design-concepts-explained.pdf
open back-of-envelope-calculations-guide.pdf
```

📊 Statistics

- **Total Files:** 56+ files
- **LLD Examples:** 43+ design pattern implementations + 10 complete systems
- **HLD Documents:** 11 complete system designs
- **Resource Guides:** 9 comprehensive guides
- **Format:** Markdown + PDF + Java

🎓 Learning Path

Beginner

1. Start with `System-Design-Resources/20-system-design-concepts-explained.md`
2. Learn basic patterns from `LLD/ObserverPattern.java` and `LLD/FactoryPattern.java`
3. Study simple system: `HLD/url-shortener-system-design-HLD.md`

Intermediate

1. Master all design patterns in `LLD/`
2. Study complex systems in `HLD/` (Instagram, Twitter, Netflix)
3. Practice with `System-Design-Resources/back-of-envelope-calculations-guide.md`

Advanced

1. Implement complete systems using pattern combinations
2. Design scalable distributed systems
3. Apply concepts from `system-design-decision-guide-HLD.md`

💡 Key Features

✅ **Comprehensive Coverage** - Covers all major design patterns and system designs
✅ **Production-Ready Code** - All LLD implementations are complete and runnable
✅ **Real-World Examples** - 5-6 practical examples per pattern
✅ **Well-Documented** - Extensive comments and explanations
✅ **Multiple Formats** - Markdown and PDF for easy reading
✅ **Scalable Designs** - HLD documents cover distributed system challenges

📝 Notes

- All Java files in `LLD/` are standalone and can be compiled/run independently
- Each design pattern file includes 5-6 real-world examples
- HLD documents include architecture diagrams, data flow, and scaling strategies
- All documents are available in both Markdown (.md) and PDF (.pdf) formats

🔗 Related Topics

- **Design Patterns:** Observer, Factory, Singleton, Builder, Strategy, Decorator, Composite, Chain of Responsibility
- **System Design:** Scalability, Load Balancing, Caching, Database Sharding, CAP Theorem
- **Distributed Systems:** Consistency, Availability, Partition Tolerance, Replication

Last Updated: November 9, 2025

Organization: Properly structured into LLD, HLD, and System-Design-Resources folders