DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

ASSIGNMENT 1

Student Name: Amrit Dey
UID: 23BCS12543
Branch: CSE
Section/Group: KRG_1A
Semester: 6th
Date of Performance: 03/02/2026
Subject Name: System Design
Subject Code: 23CSH-314

Q1 : Explain SRP and OCP in detail with proper examples.

1. **Single Responsibility Principle (SRP)**

Definition

- SRP states that a class should have only one responsibility.
- A class should have only one reason to change.
- Mixing multiple functionalities in one class increases complexity and maintenance difficulty.

Example of SRP Violation

- A BankService class handling deposit, withdrawal, loan details, passbook printing, and notifications violates SRP.
- The class manages multiple unrelated tasks, making it difficult to maintain and update.

```
class BankService {
public void deposit() {}
public void withdraw() {}
public void printPassbook() {}
public void getLoanInfo() {}
public void sendOTP() {}
 }
```

Fix (Applying SRP)

- Split the class into multiple classes based on responsibility.
- Each class handles a single functionality.

```
class PrinterService {
 public void printPassbook() {}
}
class LoanService {
public void getLoanInfo() {}
}
class NotificationService {
 public void sendOTP() {}
}
```

Benefits of SRP

- Improves code readability and organization.
- Makes testing and debugging easier.
- Reduces risk of unexpected errors.

2. Open-Closed Principle (OCP)

Definition

- OCP states that software should be open for extension but closed for modification.
- New features should be added without changing existing code.
- This prevents breaking stable and tested functionality.

Example of OCP Violation

- A notification service using conditional statements violates OCP.
- Adding new notification types requires modifying existing code.

```
class NotificationService {
public void sendOTP(String method) {
 if(method.equals("EMAIL")) {}
 if(method.equals("SMS")) {}
}
}
```

Fix (Applying OCP)

Step 1: Create Interface

```
interface NotificationService { void sendOTP(); }
```

Step 2: Create Separate Implementations

```
class EmailNotification implements NotificationService { public void sendOTP() {} } class
SMSNotification implements NotificationService { public void sendOTP() {} }
```

Benefits of OCP

- Existing code remains unchanged.
- New features can be added easily.
- Improves scalability and flexibility.

Q2. Discuss in detail about the violations in SRP and OCP along with their fixes.

SRP Violation

SRP violations commonly occur when a class grows over time and starts handling multiple unrelated responsibilities. Such classes are often called God Classes. These classes are difficult to maintain, test, and extend.

Violation

- A class performs multiple tasks
- Example: Authentication + Logging + Database operations

Problems

- Difficult to maintain
- High coupling
- Hard to test

Fix

- ➢ Split responsibilities into separate classes

OCP Violation

OCP violations occur when developers uses the conditional logic to handle the variations. This forces modification of existing code whenever a new feature is added.

Violation

- Use of if–else or switch for behavior changes
- Modifying existing code for new functionality

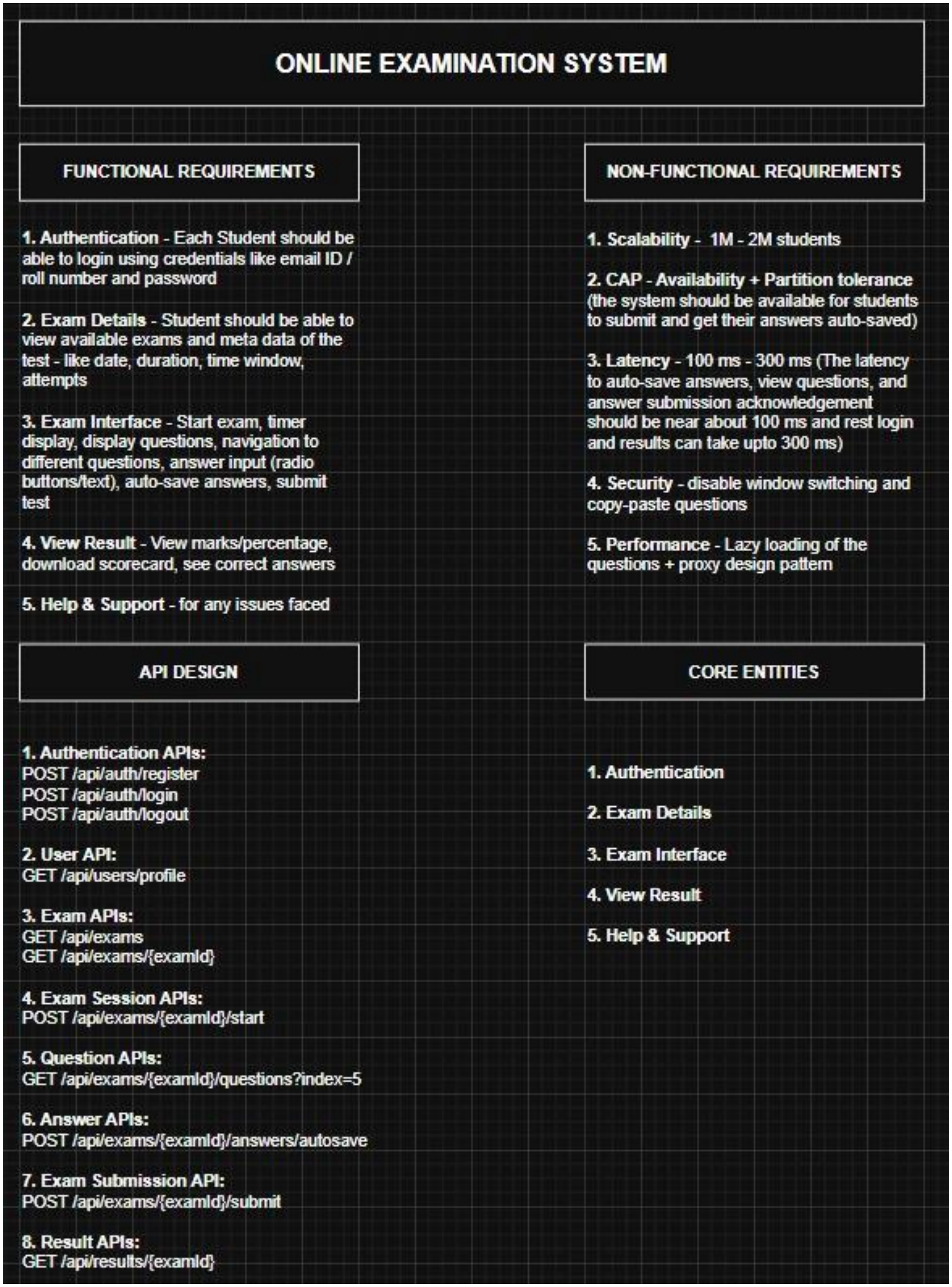
Problems

- Risk of introducing bugs
- Code becomes rigid

Fix

- ➢ Use interfaces, inheritance, and polymorphism

Q3. Design an HLD for an Online Examination System applying these principles.



ONLINE EXAMINATION SYSTEM

FUNCTIONAL REQUIREMENTS

1. **Authentication** - Each Student should be able to login using credentials like email ID / roll number and password

2. **Exam Details** - Student should be able to view available exams and meta data of the test - like date, duration, time window, attempts

3. **Exam Interface** - Start exam, timer display, display questions, navigation to different questions, answer input (radio buttons/text), auto-save answers, submit test

4. **View Result** - View marks/percentage, download scorecard, see correct answers

5. **Help & Support** - for any issues faced

NON-FUNCTIONAL REQUIREMENTS

1. **Scalability** - 1M - 2M students

2. **CAP - Availability + Partition tolerance** (the system should be available for students to submit and get their answers auto-saved)

3. **Latency** - 100 ms - 300 ms (The latency to auto-save answers, view questions, and answer submission acknowledgement should be near about 100 ms and rest login and results can take upto 300 ms)

4. **Security** - disable window switching and copy-paste questions

5. **Performance** - Lazy loading of the questions + proxy design pattern

API DESIGN

1. Authentication APIs:
POST /api/auth/register
POST /api/auth/login
POST /api/auth/logout

2. User API:
GET /api/users/profile

3. Exam APIs:
GET /api/exams
GET /api/exams/{examId}

4. Exam Session APIs:
POST /api/exams/{examId}/start

5. Question APIs:
GET /api/exams/{examId}/questions?index=5

6. Answer APIs:
POST /api/exams/{examId}/answers/autosave

7. Exam Submission API:
POST /api/exams/{examId}/submit

8. Result APIs:
GET /api/results/{examId}

CORE ENTITIES

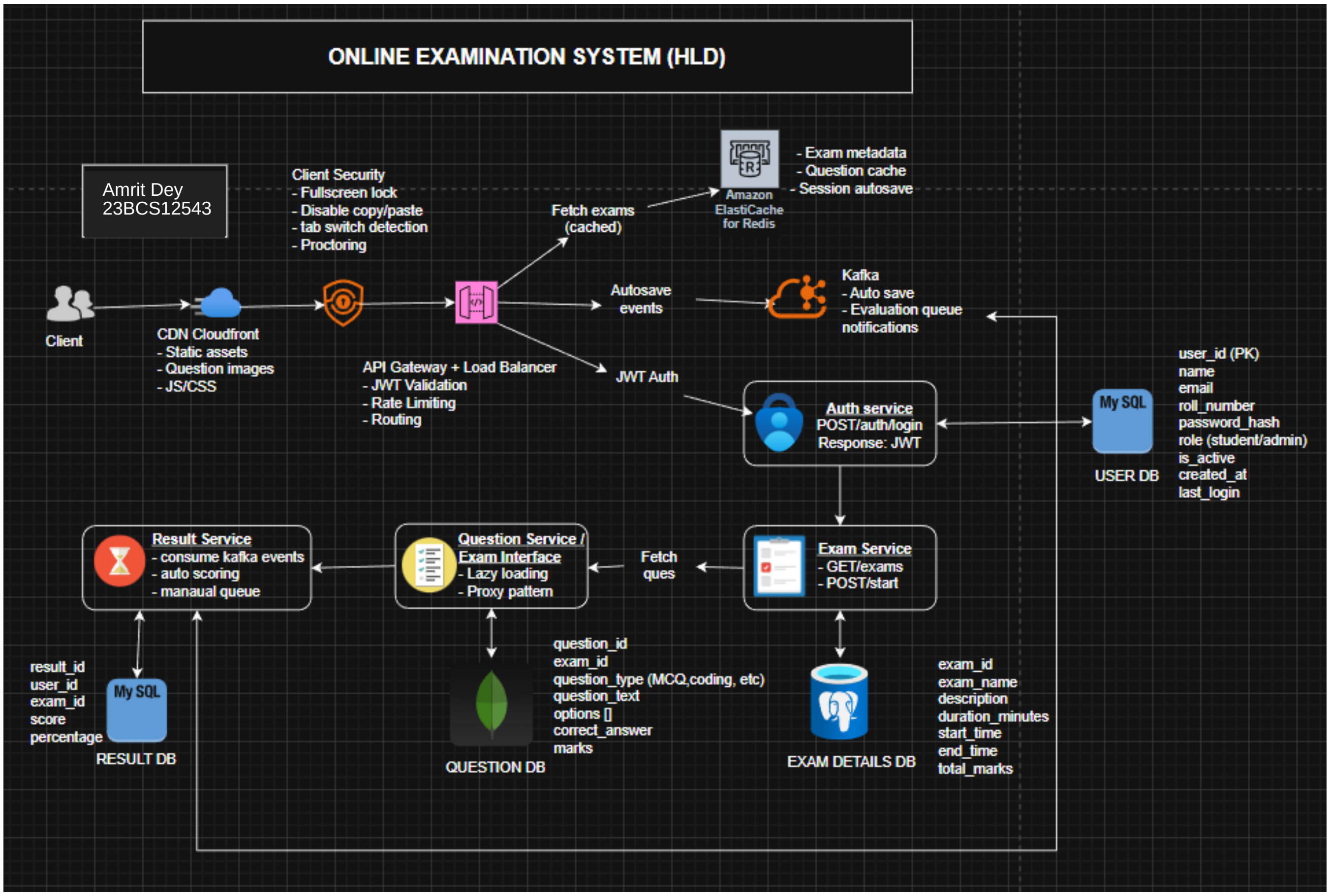
1. **Authentication**

2. **Exam Details**

3. **Exam Interface**

4. **View Result**

5. **Help & Support**


ONLINE EXAMINATION SYSTEM (HLD)
Amrit Dey
23BCS12543
Client Security
- Fullscreen lock
- Disable copy/paste
- tab switch detection
- Proctoring
Amazon ElastiCache for Redis
- Exam metadata
- Question cache
- Session autosave
Fetch exams (cached)
Client
CDN Cloudfront
- Static assets
- Question images
- JS/CSS
Autosave events
Kafka
- Auto save
- Evaluation queue
notifications
API Gateway + Load Balancer
- JWT Validation
- Rate Limiting
- Routing
JWT Auth
Auth service
POST/auth/login
Response: JWT
My SQL
USER DB
user_id (PK)
name
email
roll_number
password_hash
role (student/admin)
is_active
created_at
last_login
Result Service
- consume kafka events
- auto scoring
- manaual queue
Question Service / Exam Interface
- Lazy loading
- Proxy pattern
Fetch ques
Exam Service
- GET/exams
- POST/start
result_id
user_id
exam_id
score
percentage
My SQL
RESULT DB
QUESTION DB
question_id
exam_id
question_type (MCQ,coding, etc)
question_text
options []
correct_answer
marks
EXAM DETAILS DB
exam_id
exam_name
description
duration_minutes
start_time
end_time
total_marks