National Institute of Technology, Patna

Object-Oriented Programming in Java

OOPS MINOR PROJECT

PROJECT TITLE:- STUDY STREAK HUB

Submitted by:

S.No.	Name	Roll No.
1.	AMGOTH KALYAN	2406122
2.	AYUSH RAJ YADAV	2406160
3.	SAIRAJ RAITHATHA	2406165

3rd Semester, B.Tech CSE
Batch: 2024-2028

Submitted to:
Prof. Krishan Kumar Sethi
Department of Computer Science and Engineering

Academic Year 2025-2026

1. Introduction

StudyStreak Hub is a structured study-accountability system designed to help students maintain consistent self-study habits through group-based motivation. Users join groups that set daily study-hour targets. If at least 75% of the group members meet the requirement, the group's streak continues; otherwise, it resets. Members failing to meet their daily target for three consecutive
days are automatically removed, ensuring discipline and fairness.

The system integrates several technical mechanisms, including Java Serialization for persistent storage, SHA-256 password hashing for secure authentication, custom exceptions for controlled error handling, simulated logical date progression for daily evaluation, and detailed activity logging. The project is developed using modular, object-oriented Java principles suitable for
academic learning and potential real-world extension.

2. Real-Life Problem This System Addresses

Students often struggle to maintain consistent study habits, especially when studying in groups where structure and accountability are weak. In real scenarios, several practical issues arise:

- **No dependable way to track daily study performance**, leading to confusion about who is actually keeping up with the group goals.
- **Frequent cases of duplicate or dishonest study logs**, since manual reporting can easily be misused.
- **Streak calculations done manually become inaccurate**, causing disputes and losing the motivation that streaks normally bring.
- **No transparent record of past activity**, making it difficult to review progress or identify falling performance early.
- **Rules are inconsistently enforced** because there is no automated system ensuring fairness across all members.
- **Group study environments lack accountability mechanisms**, resulting in members gradually becoming irregular without consequences.

These real-life challenges highlight the need for a system that can **automatically track study activity**, **enforce group discipline consistently**, and **maintain secure, persistent records** of both user and group performance. Such a platform creates fairness, boosts motivation, and helps students stay committed to their study goals.

3. Proposed Solution

StudyStreak Hub implements a secure, rule-driven, and automated study-tracking platform. The key objectives include:

3.1 Secure User Management

- User registration and authentication using SHA-256 password hashing
- Prevention of duplicate accounts
- Controlled access through validated login

3.2 Group Formation and Administration

- Creation of study groups with configurable daily targets
- Admin-controlled member approval and removal
- Management of join requests

3.3 Daily Study Logging

- Each user can submit only one log per day
- Study hours validated within a strict 0–24 range

3.4 Automated Streak Evaluation

- Streak continues only if ≥75% of members meet the target
- Otherwise, streak resets automatically

3.5 Performance Enforcement

- Tracking of user failure count across days
- Automatic removal upon three consecutive failures

3.6 Persistent Storage

- Full serialization of users and groups
- Temporary-file based atomic writes to prevent corruption

3.7 Transparent Activity Tracking

- Activity logs include registrations, group events, evaluations, and removals
- Leaderboard generation based on daily study performance

Together, these modules create a robust system that enforces consistency, transparency, and fairness in group study environments.

4. System Modules

4.1 User Module

Handles all user-related operations including:

- Username, password hash, and security details
- Daily study-hour status and last log date
- Tracking of streak failures
- Managing admin privileges for respective groups
- Prevention of duplicate log submissions

4.2 Group Module

Manages the lifecycle and functioning of study groups:

- Group creation, admin assignment, and daily target configuration
- Member lists and pending join requests
- Tracking of daily study logs
- Applying the 75% streak rule
- Leaderboard generation using comparators
- Automatic removal of inactive or rule-violating members

4.3 Authentication Module

Responsible for secure system entry:

- Registration and login workflows
- SHA-256 hashing via SecurityUtil
- Validation of credentials and secure session handling

4.4 Study Logging Module

Controls all log submissions:

- Validates day and hour constraints

- Updates user and group study records
- Triggers leaderboard updates

4.5 Evaluation Module

Executes automated end-of-day evaluations based on simulated date:

- Checks member compliance with daily targets
- Updates streak status (continue or reset)
- Increments user failure counts
- Performs automatic removals based on failure thresholds
- Logs evaluation results

4.6 Storage Module

Ensures reliable data persistence:

- Java Serialization using ObjectInputStream/ObjectOutputStream
- Use of temporary files and Files.move() for atomic writes
- Reconstruction of transient structures on load

4.7 Activity Logging Module

Ensures transparency and traceability:

- Logs all major system events in activity_log.txt
- Timestamps included for clarity
- Covers user registration, group creation, log submissions, streak updates, and removals

5. Java Concepts Implemented

5.1 Object-Oriented Programming Concepts

- **Encapsulation**: All fields are private; password hashes are never exposed.
- **Abstraction**: Separate classes handle hashing, storage, and logging.
- **Inheritance**: Custom exceptions are derived from Java’s base Exception class.
- **Polymorphism**: Comparator-based sorting for leaderboards and exception behavior.

5.2 File Handling

- Serialization for complex objects such as users and groups
- FileWriter, BufferedWriter, and PrintWriter for activity logs
- Safe writes using temporary files to avoid corruption
- NIO-based path and file management

5.3 Exception Handling

Custom exceptions ensure controlled handling of invalid actions, improving user experience and system reliability. Examples include:

- UserAlreadyExistsException
- InvalidLoginException
- GroupNotFoundException
- LateLogException
- InvalidHoursException
- DuplicateJoinRequestException
- UserNotInGroupException
- NotGroupAdminException

5.4 Java Serialization

Serialization is used extensively to maintain persistent data across sessions:

- User profiles, group data, and logs stored as objects
- Handles nested structures efficiently
- Skips transient or volatile fields

5.5 Collections and Algorithms

- HashMap for quick user and group lookups
- ArrayList for managing members and join requests
- Stream API & Comparator for leaderboard sorting
- Efficient iteration and search operations

5.6 Date Handling

- LocalDate used to validate daily logs
- Prevent multiple logs per day
- Simulates system date advancement for controlled testing
- Acts as the basis for streak evaluation logic

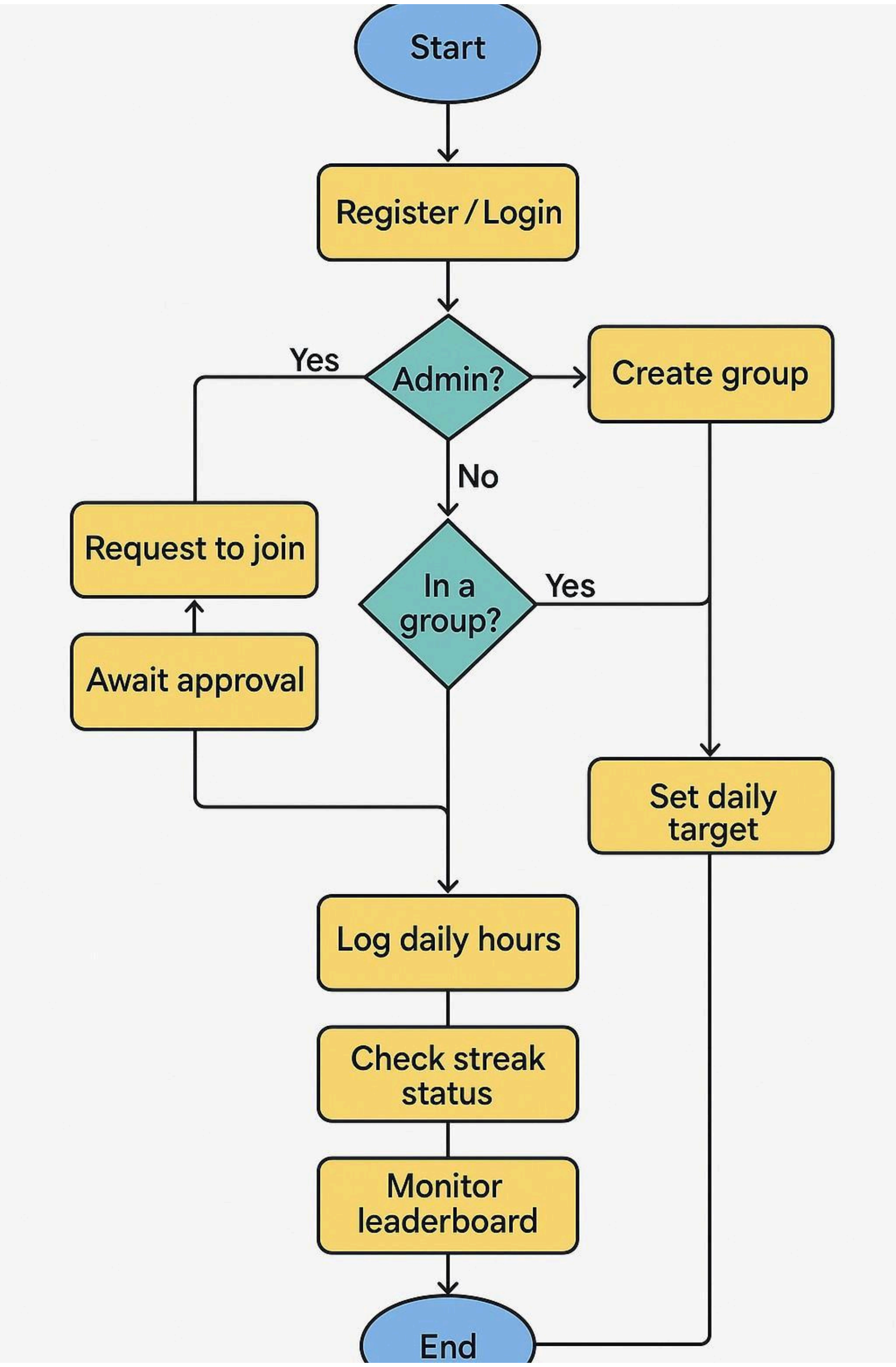
5.7 Security

- SHA-256 hashing ensures passwords are irreversible
- Hexadecimal encoding for safe storage
- No plain-text passwords stored anywhere

6. Contributions

Name	Roll Number	Contributions
Amgoth Kalyan	**2406130**	Kalyan implemented the custom exception-handling classes used throughout the system, performed final testing and output validation, and resolved critical debugging issues to ensure stable system behavior. He also assisted in structuring and preparing the final project report.
Ayush Raj Yadav	**2406160**	Ayush handled the core implementation of the StudyStreak system, developing the file-handling mechanism serving as the project's database, and implementing daily streak validation, rule enforcement, and the complete activity-logging system. He designed and implemented the console-based user interface, authored major system architecture documentation, and finalized and refined the entire project report.
Sairaj Raithatha	**2406165**	Sairaj originated the core idea for the project and implemented the group-level streak computation, user authentication flow, and the admin–user role structure. He developed the workflow for daily evaluations, managed group operations such as member handling and automatic user removal, created the internal file structure, prepared theoretical sections, and contributed to the problem statement and system design.

7. Flowchart



Start
Register / Login
Admin?
Yes
Create group
No
Request to join
In a group?
Yes
Await approval
Set daily target
Log daily hours
Check streak status
Monitor leaderboard
End