

Weekly Progress Report

Name: Abhinav Sujit Pawar

Domain: Core Java

Date of submission: February 10, 2026

Week Ending: 01

I. Overview:

This week, the primary focus was on designing and developing the core architecture for a **Banking Information System** prototype using Java. The objective was to implement essential banking functionalities while ensuring clean code structure and basic security measures.

II. Achievements:

1. System Architecture Design:

- Designed a modular class structure comprising Account, Transaction, and BankingSystem classes to follow the Single Responsibility Principle.
- Defined data flows for account registration, fund transfers, and transaction logging.

2. Core Java Implementation:

- User Registration & Management:** Built a unique account generation system starting from ID 1001.
- Transaction Logic:** Implemented deposit and withdrawal methods with real-time balance updates.
- Fund Transfer:** Developed a secure peer-to-peer transfer logic that validates recipient existence and prevents self-transfers.

3. UI & Experience:

- Created a menu-driven Console Interface for seamless navigation between banking operations.
- Applied java.time for professional-grade formatting of account statements.

III. Challenges:

1. Input Validation:

- **Challenge:** Encountered InputMismatchException when non-numeric data was entered into deposit/withdrawal fields.
- **Resolution:** Implemented helper methods (getIntChoice, getDoubleInput) with try-catch blocks to sanitize user input and prevent system crashes.

2. Data Persistence Strategy:

- **Challenge:** Initial volatility of data stored in memory (RAM) being lost after program termination.
- **Ongoing Effort:** Evaluating the transition from temporary HashMap storage to a file-based or SQL database approach for the next phase.

IV. Learning Resources:

1. Java Collections Framework:

- Utilized HashMap documentation to understand efficient data retrieval for account lookups.
- Referenced ArrayList documentation for managing dynamic transaction histories.

2. Formatting & Logic:

- Leveraged java.time.format.DateTimeFormatter tutorials to create readable timestamps for the Account Statement feature.
- Studied Java logic patterns for implementing robust error handling and conditional checks.

V. Next Week's Goals:

1. Persistence & Security:

- Implement File I/O to save user data into a .txt or .csv file so information persists between sessions.
- Explore basic password hashing to improve security beyond plain-text storage.

2. Feature Expansion:

- Add a "Delete Account" or "Deactivate Account" feature.
- Develop a "Loan Eligibility" calculator module within the dashboard.

VI. Additional Comments:

The prototype successfully demonstrates all "Minimum Requirements" defined in the problem statement. The system is currently stable and provides a clear preview of the final application's flow and usability.