Part A – Banking System Documentation

1. Requirements Elicitation

1.1 Functional Requirements

1. Customer Management: Register and authenticate customers.
2. Account Management: Open Savings, Investment, and Cheque accounts.
3. Transactions: Deposit and withdraw funds (except withdrawal from Savings).
4. Interest Processing: Apply monthly interest (0.05% Savings, 5% Investment).
5. System Services: Display balances and transaction history.

1.2 Non-Functional Requirements

1. Security: Secure login and transaction validation.
2. Performance: Complete operations within 2 seconds.
3. Usability: User-friendly JavaFX GUI.
4. Scalability: Support future account types.
5. Reliability: Ensure atomic transactions (no partial updates).

2. Structural UML Modelling

2.1 Use Case Diagram

Actors: Customer, Admin.
Use cases: Register/Login, Open Account, Deposit, Withdraw, View Balance, View Transactions, Earn Interest.

2.2 Class Diagram

Classes: Customer, Account (abstract), SavingsAccount, InvestmentAccount, ChequeAccount, Interface: InterestBearing.
Demonstrates abstraction, inheritance, encapsulation, polymorphism, and interfaces.

3. Behavioural UML Modelling

3.1 Sequence Diagrams

Examples: Login sequence (Customer → GUI → Controller → DAO → Database). Deposit funds sequence (Customer → GUI → Controller → Account → DAO → Database).

3.2 State Diagram

Example: Account lifecycle with states Active → Interest Applied → Updated Balance, triggered by monthly timer.

Appendix: Mock Interview Record

Lecturer (Client): What core services must the system provide?
Student (Analyst): Customers can open accounts, deposit, withdraw, and earn interest.
Lecturer: Any restrictions for accounts?
Student: Savings – no withdrawals. Investment – minimum deposit required. Cheque – requires employer details.