Diagram, logo

Description automatically generated

MEHRAN UNIVERSITY

OF ENGINEERING & TECHNOLOGY

JAMSHORO, PAKISTAN

Subject: Software Construction and Development

(Lab 1 - Tasks Solution)

|  |  |
| --- | --- |
| **Roll No:** | **22SW028** |
| **Section:** | **I** |
| **Name** | **Arbab Hussain** |

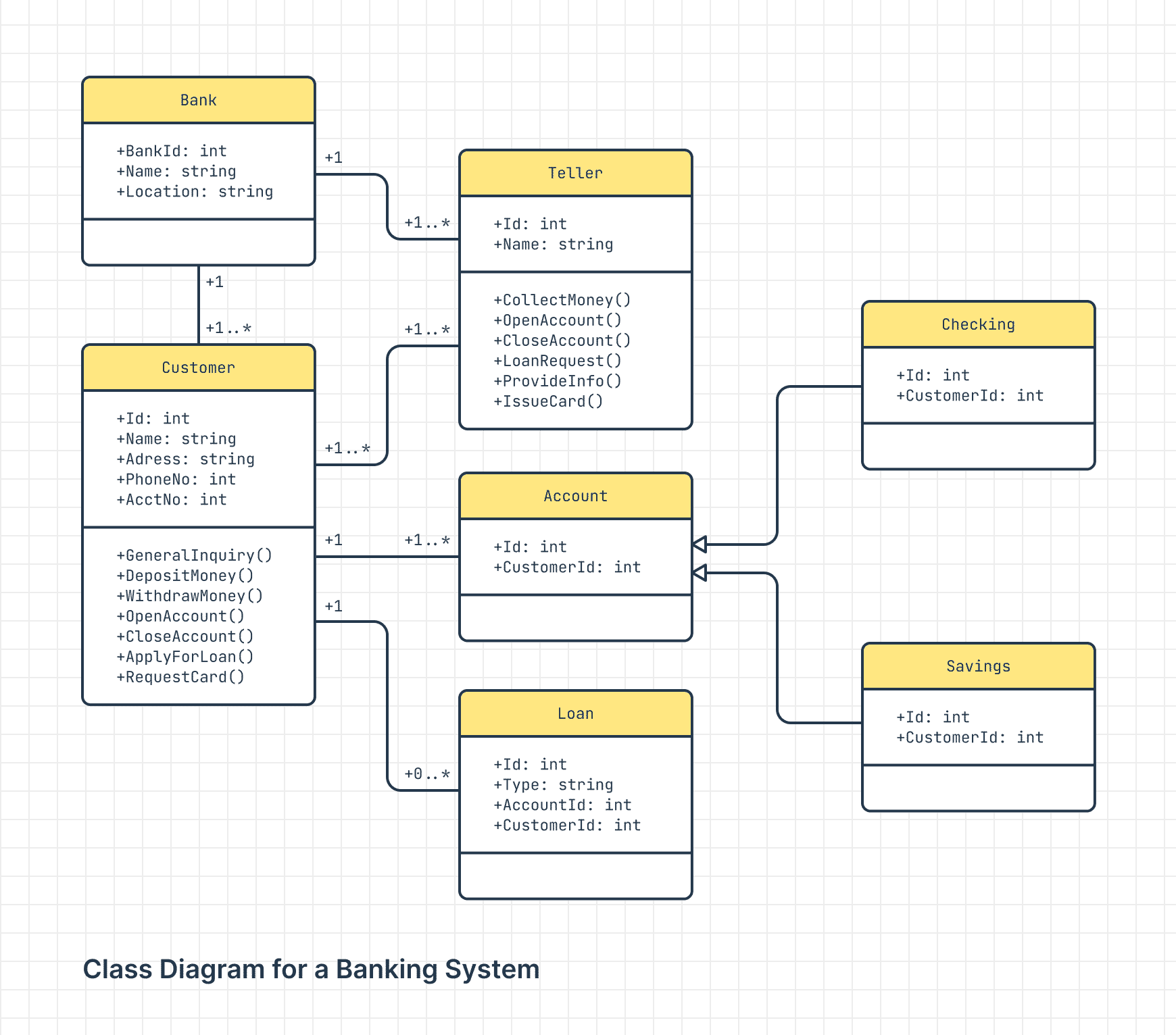
**LAB 01 TASKS**

**NOTE: THE DIAGRAMS ARE CREATED WITH A WEB APP NAMED LUCIDCHART**

**Task 1:**

Your first task is to analyze both examples and write scenarios. You must use appropriate Class Diagram Terms in your statements by understanding the meaning for each symbol drawn in the example.

Example 1:



**SCENARIO:**

A customer visits a bank to open an account. A teller processes the request and performs operations such as collecting initial deposits or issuing cards. The customer can open either a Checking or Savings account, both inheriting from the general Account class. If the customer applies for a loan, a loan record is created and linked to the customer. The bank manages multiple customers, accounts, and tellers while ensuring data integrity through the relationships depicted.

**Class Descriptions:**

**The Bank class** represents the main organization with attributes BankId, Name, and Location. It is associated with multiple Customers and Tellers.

**The Customer class** models users of the bank. It includes attributes such as Id, Name, Address, PhoneNo, and AcctNo. The methods include operations like DepositMoney(), WithdrawMoney(), ApplyForLoan(), and RequestCard().

**The Teller class** interacts with customers and performs operations such as CollectMoney(), OpenAccount(), CloseAccount(), and LoanRequest().

**The Account class** models a general bank account with attributes Id and CustomerId. The inheritance relationship connects it to Checking and Savings accounts, as indicated by the triangle symbol.

**The Loan class** handles banking loans, with attributes Id, Type, AccountId, and CustomerId. It has a composition relationship with the Customer class, as shown by the solid diamond symbol.

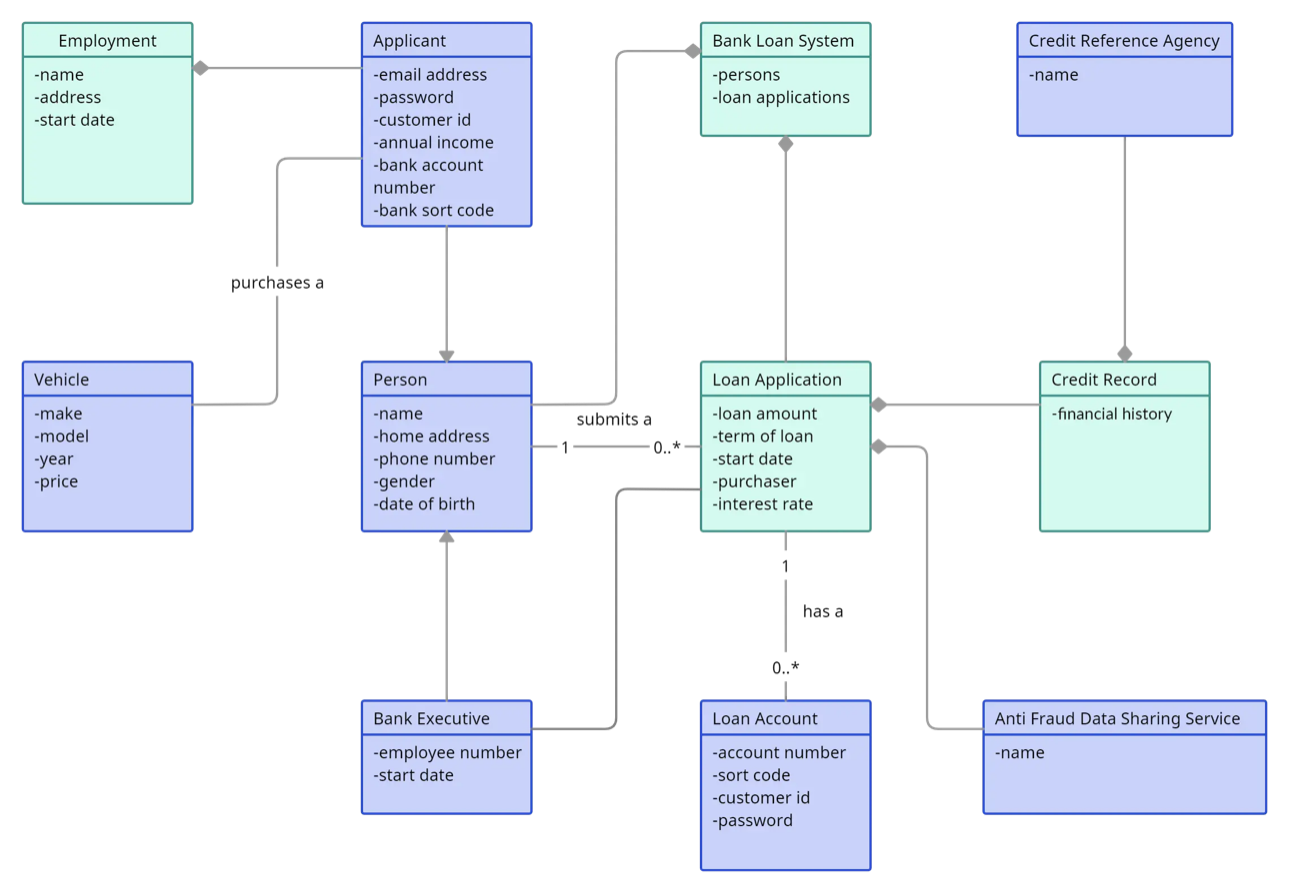
**Relationships:**

**Association:** The Bank is associated with Customer and Teller through a solid line. A bank can have multiple customers and tellers, indicated by the multiplicity 1..\*.

**Inheritance:** The Account class serves as a parent class for Checking and Savings, signified by a triangle symbol.

**Composition:** The Loan class has a composition relationship with the Customer, indicated by a solid diamond symbol. This means that if the Customer is deleted, the associated loans also cease to exist.

**Example 2:**



**SCENARIO:**

In a loan management system, an applicant provides their personal details, including employment and vehicle information. The bank processes the loan application, checks financial history through the credit record, and creates a loan account upon approval. The system maintains dependency on external services like the Anti-Fraud Data Sharing Service for validation and risk assessment.

**Class Descriptions:**

**The Person class** serves as a central entity with attributes such as name, home address, phone number, gender, and date of birth.

**The Applicant class** extends from Person, indicating an inheritance relationship. It adds attributes such as email address, password, customer id, annual income, bank account number, and bank sort code.

**The Loan Application** class handles attributes such as loan amount, term of loan, start date, purchaser, and interest rate. It is associated with Person, as indicated by the solid line with an arrow.

**The Bank Loan System** class aggregates Loan Applications and Persons. This aggregation is depicted by a hollow diamond, which shows that Bank Loan System manages multiple loan applications and persons.

**The Loan Account class**, which includes attributes like account number, sort code, customer id, and password, is dependent on the Loan Application, signified by a dashed arrow.

**The Employment and Vehicle** classes provide further contextual attributes like address, start date, make, model, year, and price. These classes are connected to Applicant via simple associations, as denoted by the solid lines.

**The Credit Record class** contains financial information and is associated with the Loan Application class. Additionally, it interacts with the Credit Reference Agency through a direct association.

**The Anti-Fraud Data Sharing** Service class represents external services, indicating dependency with the Loan Account class.

**Relationships:**

**Inheritance** relationship between Person and Applicant is indicated by the triangle symbol.

**Association** is used for connections such as Applicant to Employment and Loan Application to Credit Record.

**Aggregation** is used between Bank Loan System and Loan Applications, indicated by the hollow diamond symbol.

**Dependency** is represented with a dashed arrow from the Loan Account to the Anti -Fraud Data Sharing Service.

**Task 2:**

**Scenario 1: Library Management System**

Description: Design a UML class diagram for a Library Management System. The system should manage books, members, and staff.

Classes to Include:

1.Book:

Attributes: ISBN, Title, Author, Genre, Availability.

Methods: CheckOut(), Return().

2.Member:

Attributes: MemberID, Name, Address, PhoneNumber.

Methods: BorrowBook(), ReturnBook().

3.Staff:

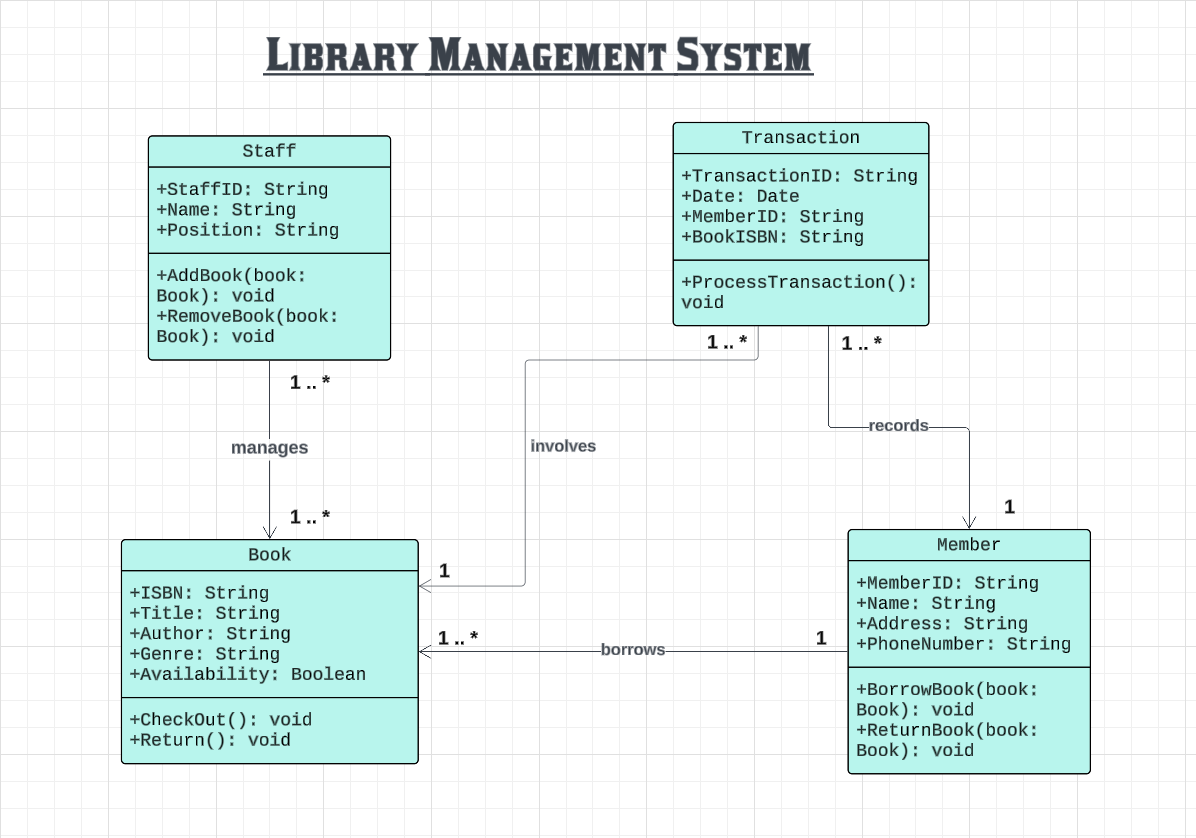
Attributes: StaffID, Name, Position.

Methods: AddBook(), RemoveBook().

4.Transaction:

Attributes: TransactionID, Date, MemberID, BookISBN.

Methods: ProcessTransaction().



**Scenario 2: Online Shopping System**

Description: Create a UML class diagram for an Online Shopping System where users can browse products, add them to a cart, and make purchases.

Classes to Include:

1.Product:

Attributes: ProductID, Name, Description, Price, StockQuantity.

Methods: UpdateStock().

2.User:

Attributes: UserID, Name, Email, Password.

Methods: Register(), Login(), Logout().

3.ShoppingCart:

Attributes: CartID, UserID, ListOfProducts.

Methods: AddProduct(), RemoveProduct(), Checkout().

4.Order:

Attributes: OrderID, UserID, Date, TotalAmount.

Methods: ConfirmOrder().

