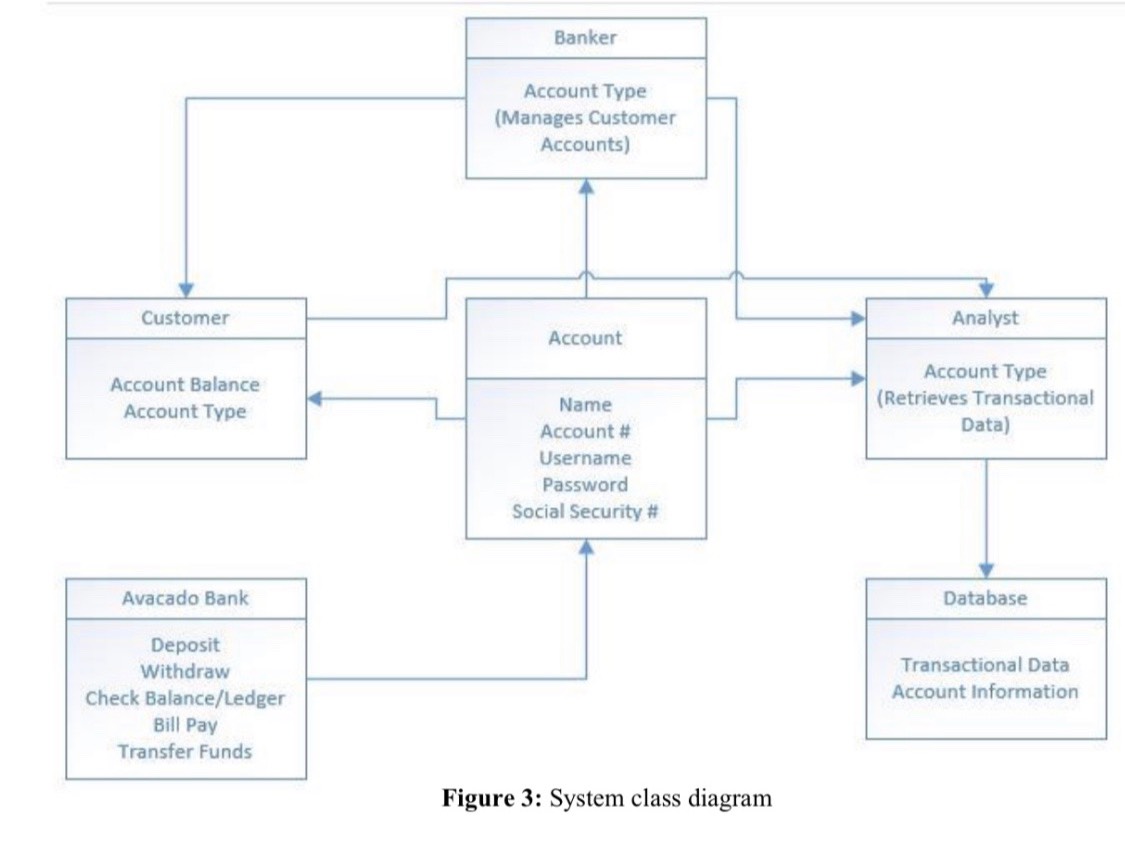
ARCHITECTURE MODELING

TASK-1- Measure the class Diagram coupling and cohesion-

Basic class Diagram



**Measurement of Cohesion based on above class diagram:**

Calculations:

Formula**- Cohesion for a class(n) = Classes on which n directly-indirectly potentially depends/Nc**

Where Nc-total no. of classes

* For Banker class:

Cohesion = 3/6=0.5

* For Customer class:

Cohesion= 2/6= 0.34

* For Avocado Bank:

Cohesion = 5/6=0.83

* For Analyst class:

Cohesion = 1/6=0.16

* For Database class:

Cohesion = 0/6=0

* For Account class:

Cohesion = 4/6=0.67

|  |  |  |
| --- | --- | --- |
| Classes | Directly or indirectly depends on: | Cohesion Values: |
| Banker | Customer, Analyst, Database | 0.5 |
| Customer | Analyst, Database | 0.34 |
| Avocado Bank | Account, customer, banker, analyst, Database | 0.83 |
| Analyst | Database | 0.16 |
| Database | - | 0 |
| Account | Customer, Analyst, Database, Banker | 0.67 |

**COHESION VALUES**

**Coupling :**

A coupling measure between classes, which class directly coupled with other class

Flow P (C2, C1) = Slice (P, c1, V c1) | N (c2) / N (C1)

Flow *P* (*c*2, *c*1) implies that information flow from class *c*2 to class *c*1 in a modular *P*, employee information modular.

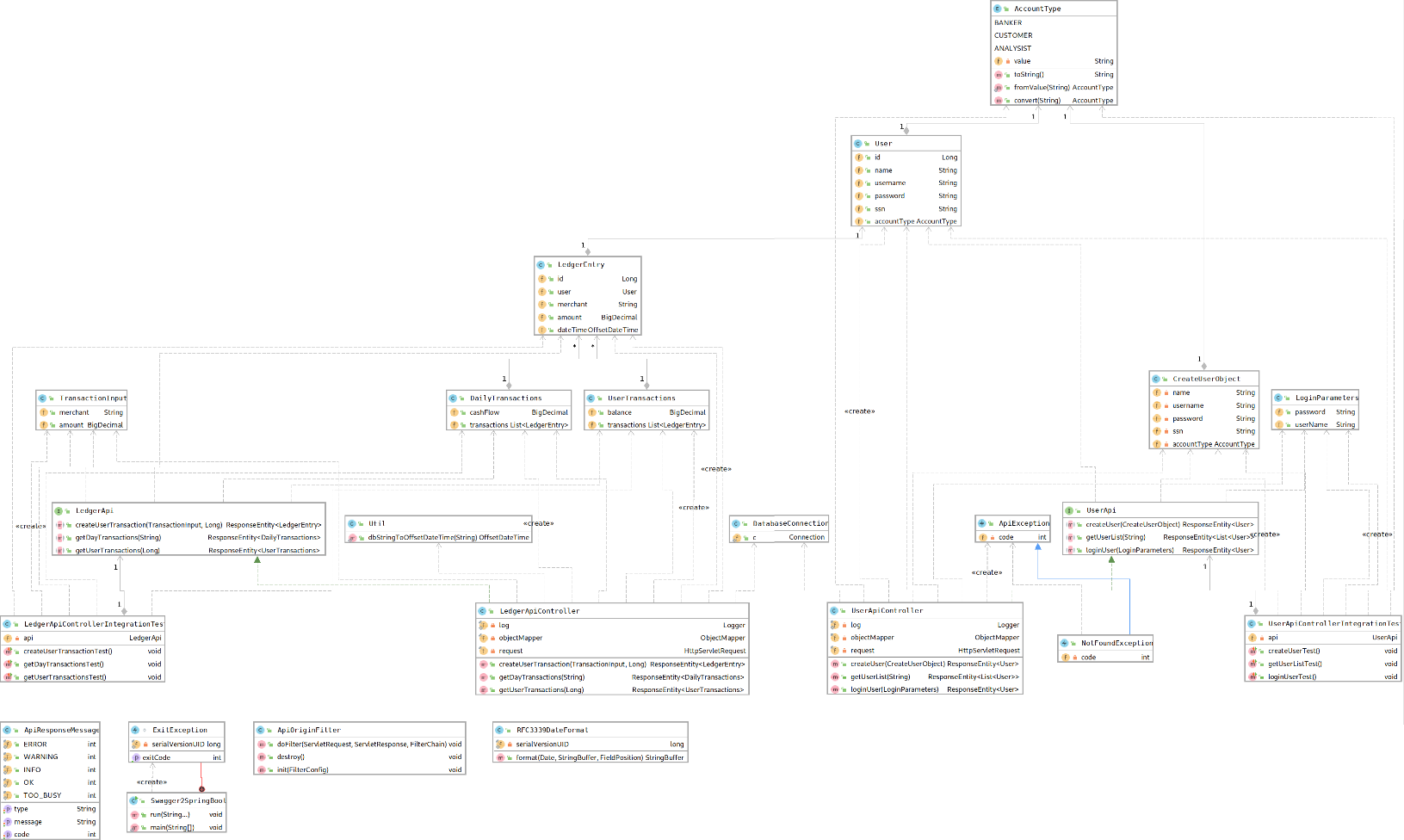
Coupling P (C1, C2) = Flow p (C1, C2) +Flow p (C2, C1)/N(C1) N(C2)

Note: The instructor did not want us to calculate coupling values

TASK-2- Design Patterns chosen for the system

|  |  |  |
| --- | --- | --- |
| **Major Issues During Implementation** | **Design Patterns used to rectify the issues** | **Reasons for the Design patterns chosen** |
| Database Connectivity | Singleton Pattern  (Creational Design Pattern) | It is used for the database connection so that it can be easily accessed from anywhere in the program. |
| Implementation of HTTP API | Strategy Pattern (Behavioral Design Pattern) | Since it lets the algorithm vary independently from clients that use it. |
| To provide controller with Object Mapper (to translate between JSON and Java objects | Dependency Injection pattern | It is used since we did not create the Object Mapper but instead ask the provider to create one for us. |

Note: The design patterns that has been used in order to improve the system that might change the way the classes interact with each other or the relationships between the classes. The improved version of class diagram is as follows:



**Updated Class Diagram with all the relationships and Multiplicities specified.**