Homework 4

Task: Consider the system described below. Evaluate the architectural design decisions with sensitivity points, trade-off points and the risk points.

1. ATM System Requirements:

The Interbank Consortium, a hypothetical financial institution, has directed its software development subsidiary, Interbank Software, to develop new services that support a network of automated teller machines (ATMs).

Customers use ATMs to make queries, withdrawals, deposits and funds transfers involving their accounts. Thieves or crooks must be prevented from interfering with these actions.

Interactions with the ATM would work like this:

1. The customer inserts his/her bank card into the ATM.
2. The ATM prompts the customer for a "password" which the user enters at the ATM.
3. The customer then selects an action to be performed; the selected action is then performed by the branch (perhaps causing dispersal of cash at the ATM).

Additionally, the bank would like to be able to use the system to maintain statistics about customers' behaviour in order to adapt its services to their needs, and also to send them some advertisements when they are using the system.

Functional requirements
The ATM system must be able to provide the following services to the customer. -A customer must be able to make a cash withdrawal from any suitable account linked to the card. -A customer must be able to make a deposit to any account linked to the card. -A customer must be able to make a transfer of money between any two accounts linked to the card. -A customer must be able to make a balance inquiry of any account linked to the card.

Non-functional requirements (NR)
-NR1: The ATM system should manage the presence of heterogeneous system. -NR2: The system must handle concurrent accesses to the same account correctly. -NR3: The security mechanisms be located, in the ATMs or a central network controller. -NR4: customer services should be available 999/1000 requests. -NR5: customer services should have a fast response time.

Qualities	Design issues
Heterogeneity	-communication mechanism between two banks
Concurrency	-the states of account -concurrent accesses to the same account -task communication
Security	-accessability to system resources -location of security mechanism per layer
Availability	-replication -synchronization for replication
Performance	-distributed transaction model

2. Some of the tactics and patterns used to design the system:

- Client-Server pattern
- Multi-Tier pattern
- Reduce the coupling between modules
    - assigning responsibilities to elements so that the majority of changes to the system will affect a small number of those elements (modifiability)
    - the business rules, database, and client should be localized into components
- Use an intermediary
    - to mediate the communication between the tiers by some abstract interface
    - A data access layer that uses Open DataBase Connectivity (ODBC) between the business rules and the database
    - A translation layer between the business rules and the client that understands Extensible Markup Language (XML)
    - With these intermediaries, it will be simple to add new databases or clients.
- Introduce concurrency
    - Make servers multithreaded so that they may execute multiple requests in parallel
- Increase resources
    - Deploy additional database servers and business rule servers
- Arbitrate between multiple resources to ensure that the processing load is distributed among the system's resources according to a chosen scheduling policy
    - Introduce a "load balancer" component

3. Some of the views of the system:

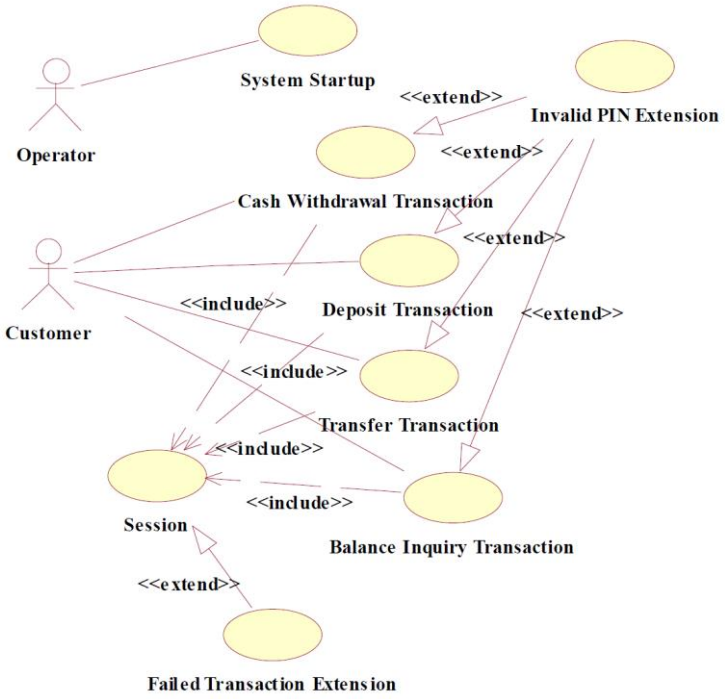


Figure 1. Primary purposes of the system (use case view).

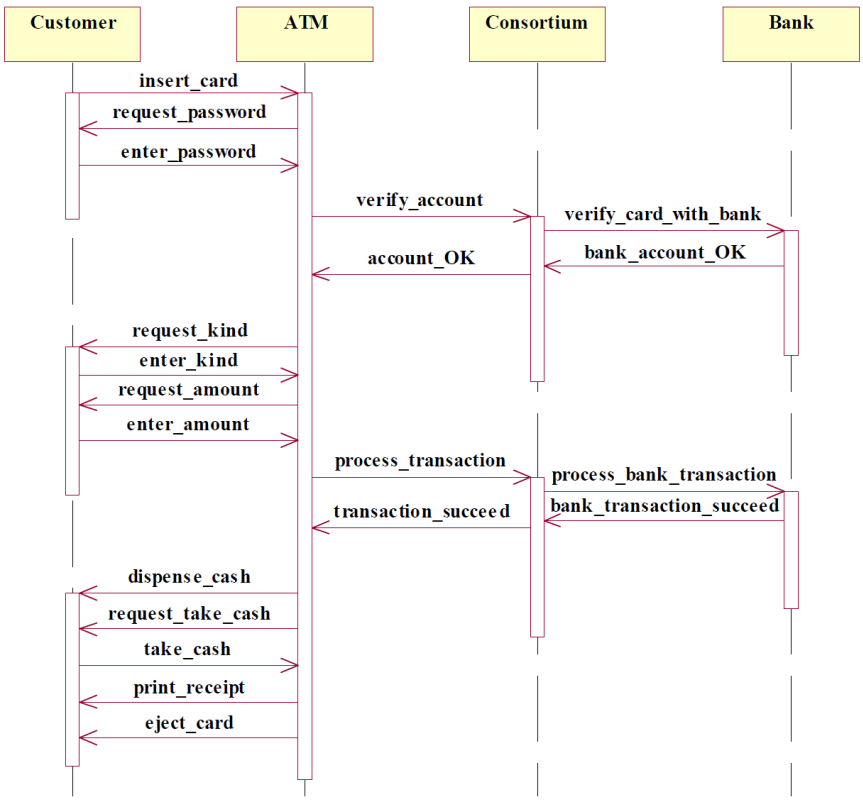


Figure 2. Basic sequence of actions during transactions (use case view).

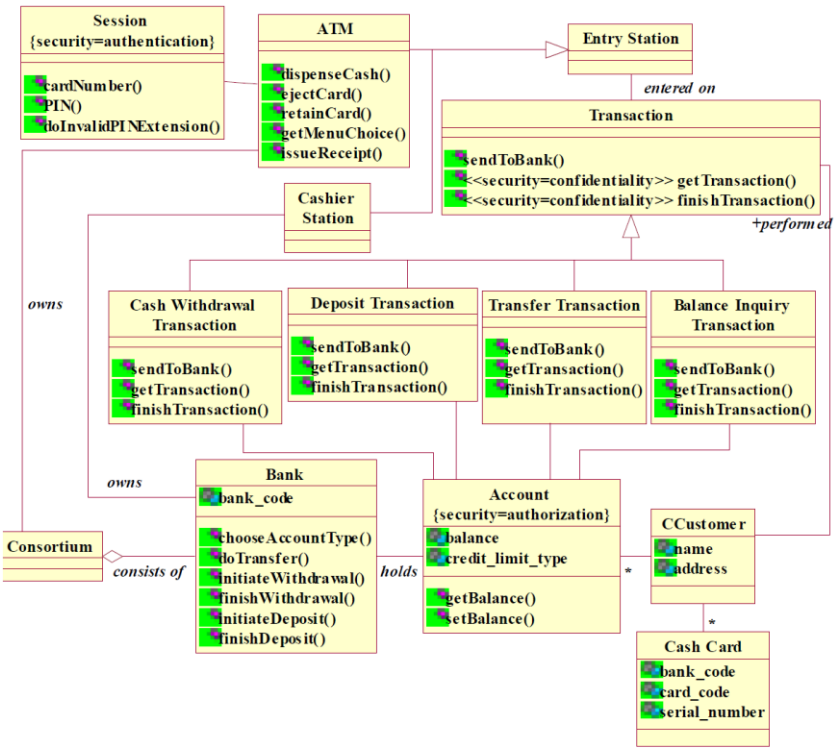


Figure 3. Logical structure of a system (logical view).

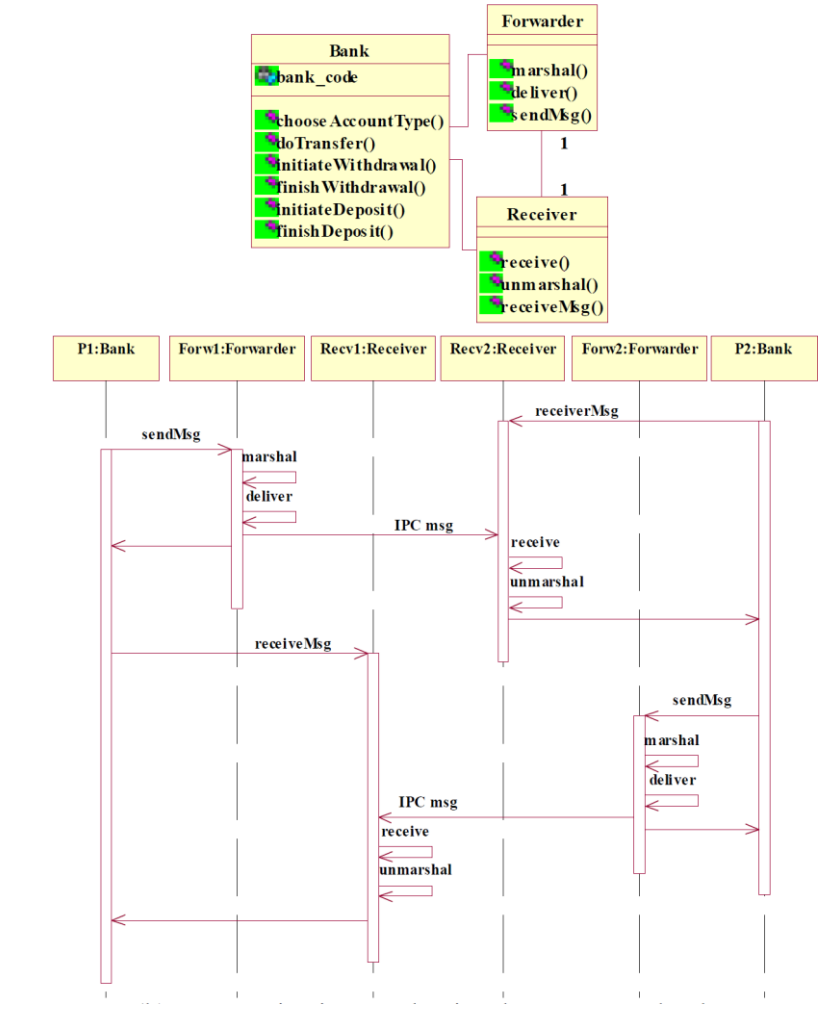


Figure 4. Communication mechanism between two banks (logical view).

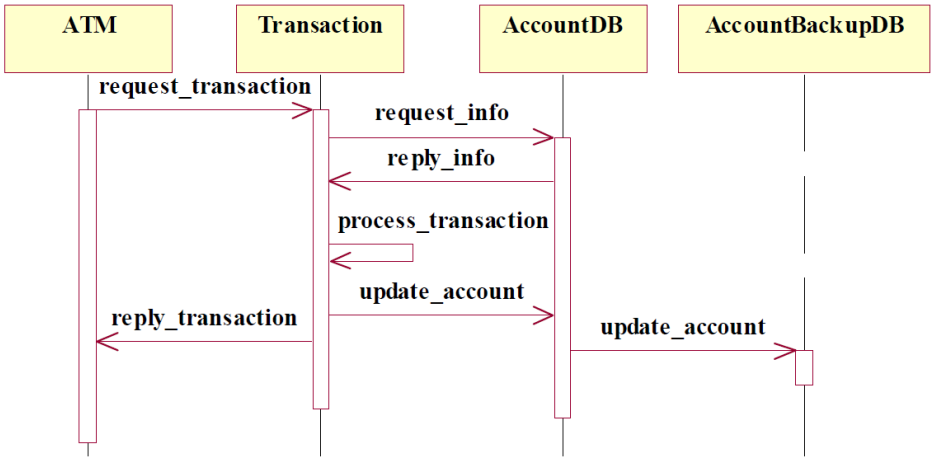


Figure 5. Replication and its synchronization (process view).

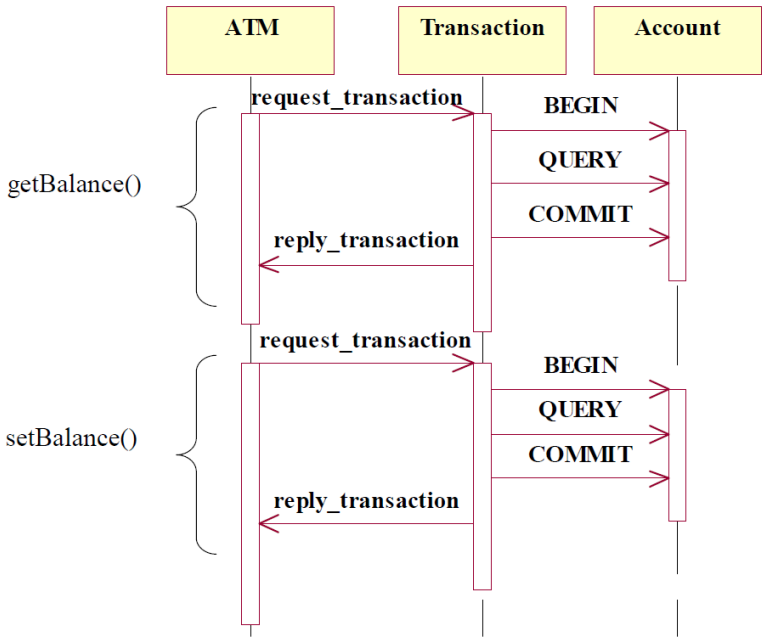


Figure 6. Per-operation transaction model (process view).

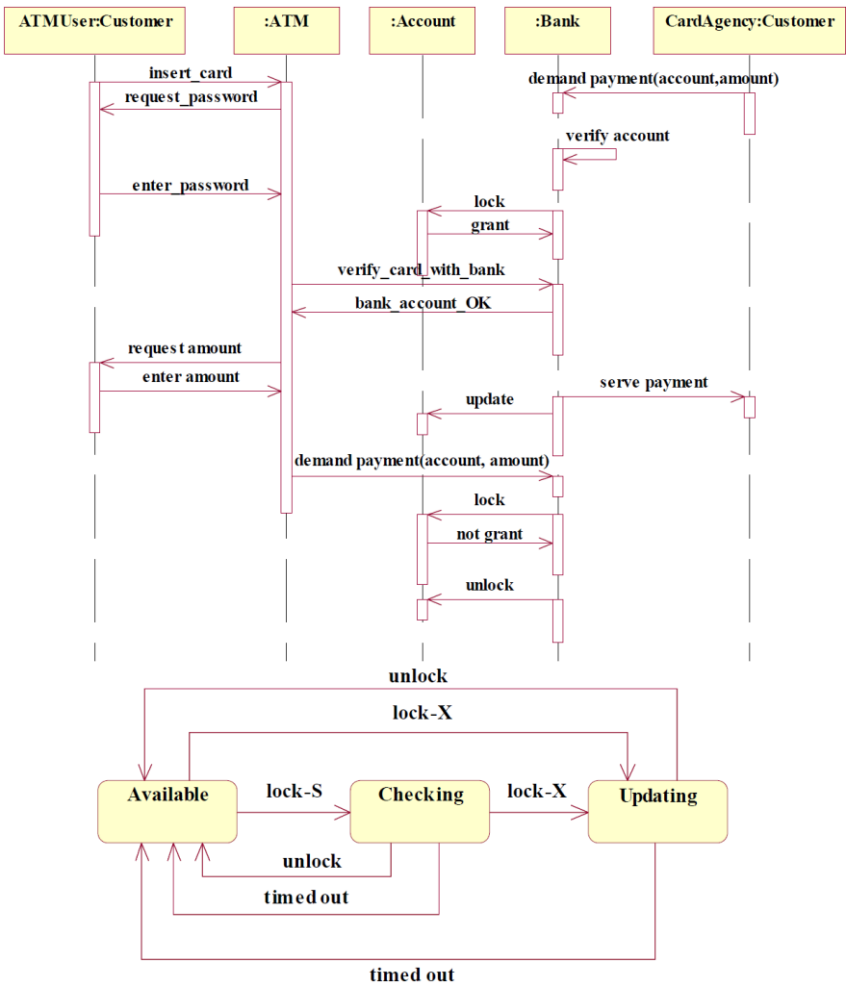
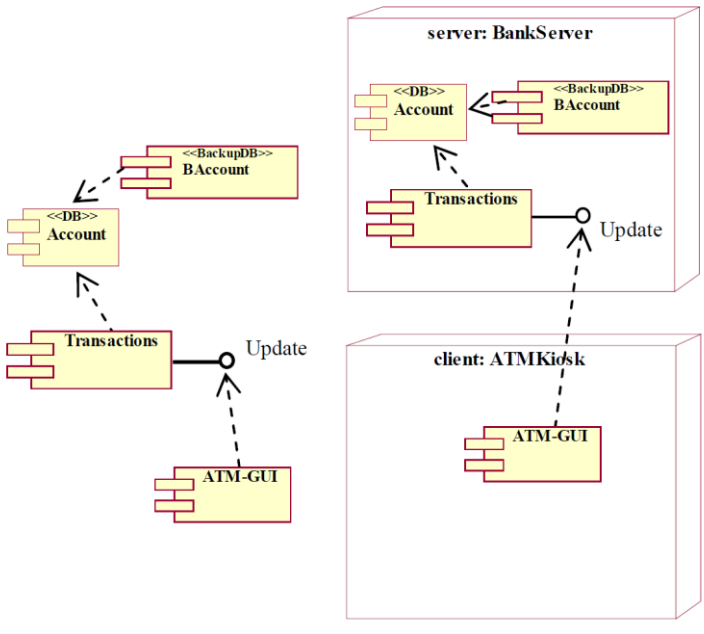


Figure 7. Concurrent access and the states of account (process view).



(a) development view (b) deployment view

Figure 8. Client-server structure.