

# Aspect-Oriented Development of A Banking Management System

## **Fourth Turkish Aspect-Oriented Software Development Workshop**

Buğra M. Yıldız

Çağrı Toraman

Uğur Bilen



---

# Presentation Outline

---



- ◆ 1. Banking Management System
- ◆ 2. Design of BMS
- ◆ 3. Design Patterns
- ◆ 4. New Concern Requirements
- ◆ 5. Crosscutting Problem
- ◆ 6. AOP Solution with Aspects
- ◆ 7. Discussion / Conclusion

# 1. Banking Management System

- ◆ BMS is a complex system:

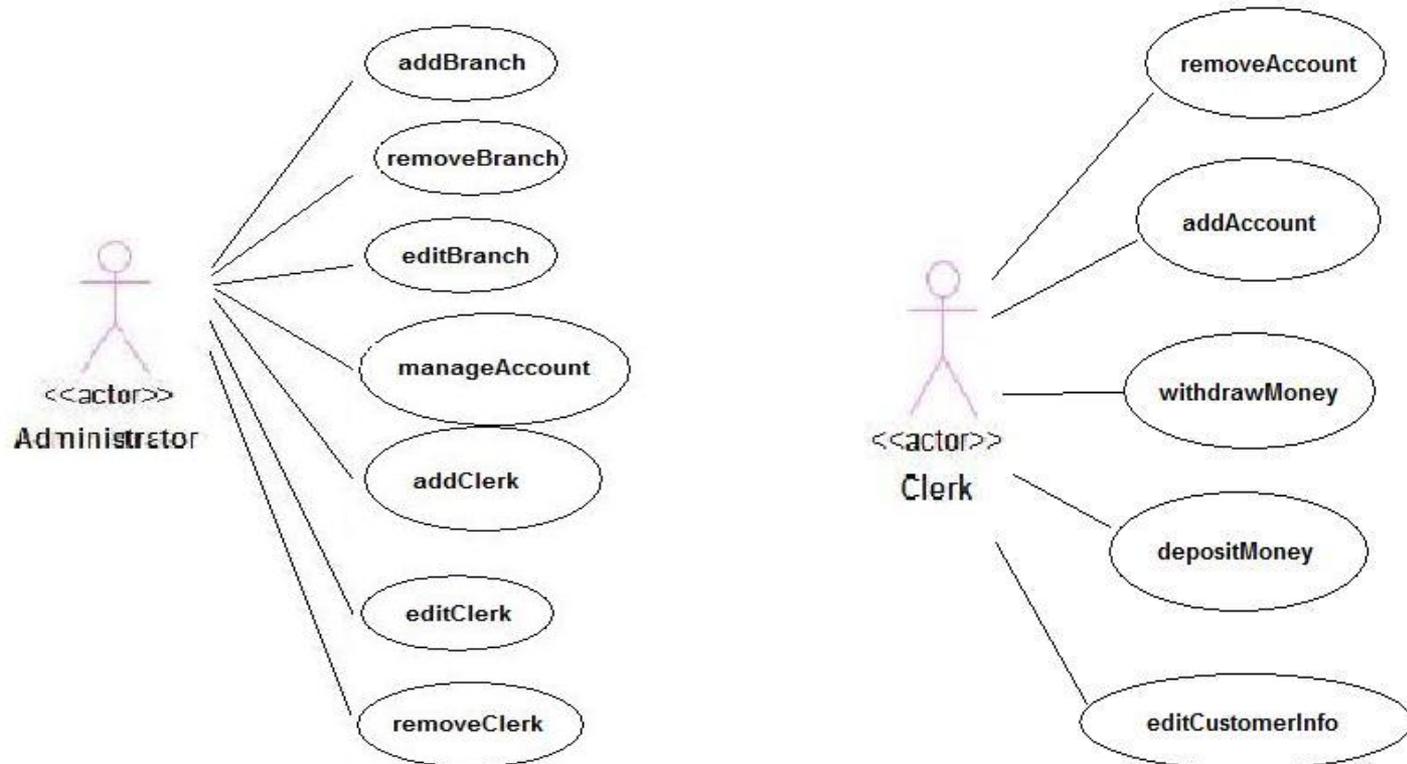
- Object Oriented,
- Client – Server Interaction,
- Complicated Database



- ◆ As an example case, we created a simple banking management application!

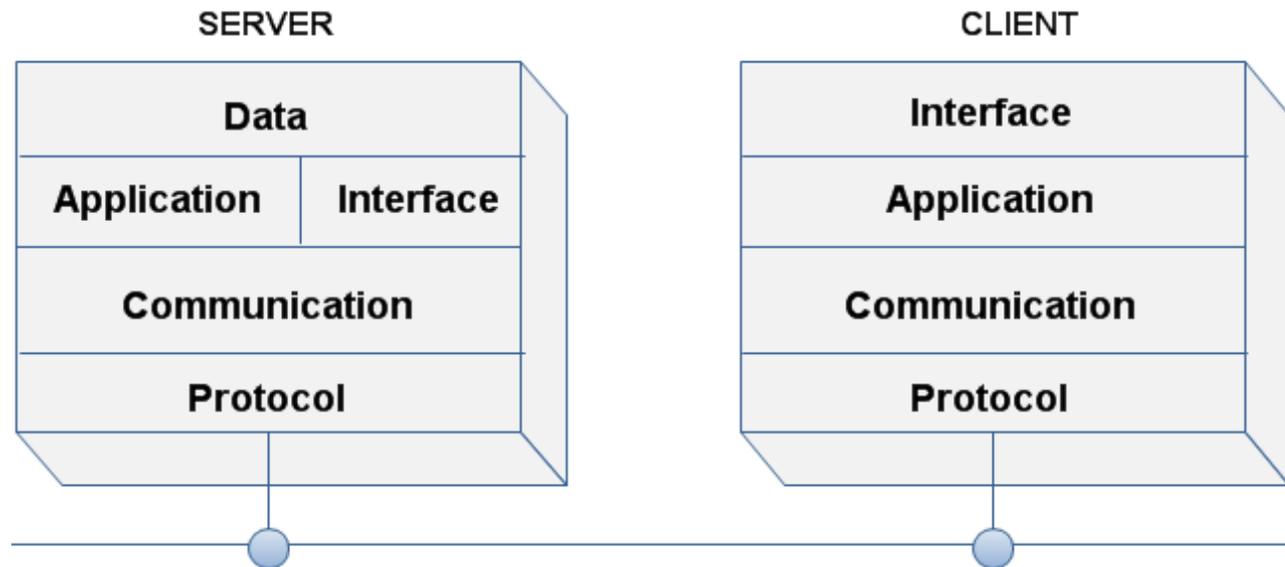
## 2. Design of BMS

### ◆ Requirement Analysis



## 2. Design of BMS

### ◆ Architectural Design

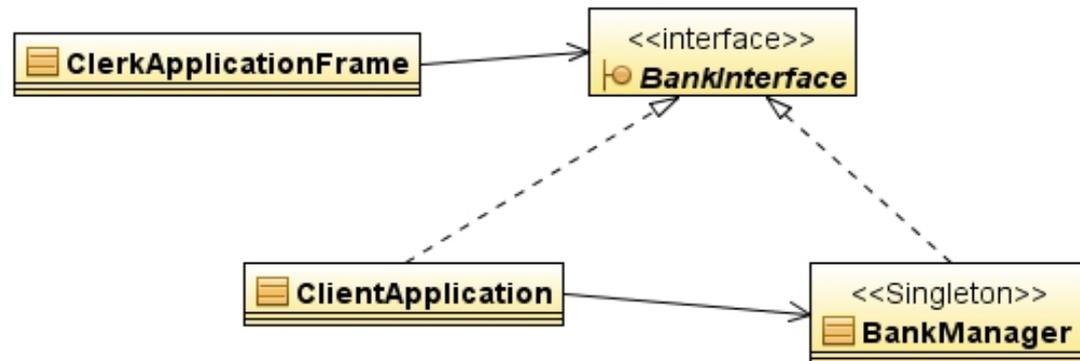


The layered diagram of the application

# 3. Design Patterns

## ◆ Applied Patterns

- Singleton
- Layers Pattern( Architectural Pattern)
- Proxy Pattern



# 4. New Concern Requirements

- ◆ Persistent database,
- ◆ Synchronized data entities,
- ◆ Monitoring of the system's server,
- ◆ Multi-language support,

# 5. Crosscutting Problem

- ◆ Multi-language:
  - Scattered through whole GUI components  
(6 classes – total 175 lines)
  - Tangled code on each GUI component
- ◆ Monitoring:
  - Scattered through 25 points (might increase)
  - Low cohesion

# 5. Crosscutting Problem

## ◆ Synchronization:

- Scattered through synchronized entities
- Tangled code on each synchronized entity

## ◆ Persistency:

- Scattered through lots of database-required points
- Low cohesion

# 6. AOP Solution with Aspects

## ◆ Demonstration on:

- Multi-Language Aspect
- Server Monitoring Aspect
- Synchronization Aspect
- Persistency Aspect



# 7.1 Discussion

- ◆ Aspect Mining vs Early Identification of Aspects.
- ◆ Trade-off between Aspect Oriented approach and Object Oriented approach.
- ◆ The difficulty of using pointcut syntax.
- ◆ Aspect Oriented Programming is not an alternative of Object Oriented Programming.

## 7.2 Conclusion

- ◆ In this work, we implemented some crosscutting concerns that can be encountered in most of the data-related complex systems.
- ◆ While we were identifying aspects, we followed two aspects identification methods that are Aspect Mining and aspect derivation from requirements.
- ◆ Finally, we realized that Aspect Oriented programming is not an alternative to the Object Oriented approach.

# Acknowledgment

- ◆ We would like to specially thank to Asst. Prof. Bedir Tekinerdoğan for giving us the opportunity to show ourselves in this workshop of the one of the leading software techniques that is Aspect-Oriented Software Development.

---

# Questions???

---

