

# **TEAM PROJECT REPORT**

## **BANK INFORMATION SYSTEM (BIS)**

INFSCI 2511

INFORMATION SYSTEM ANALYSIS &  
DESIGN AND EVALUATION

SPRING 2005

Group Members:

Mr. Derek Pawlikowsky

Miss Jaruwan Laptrakool

Mr. Kittipong Techapanichgul

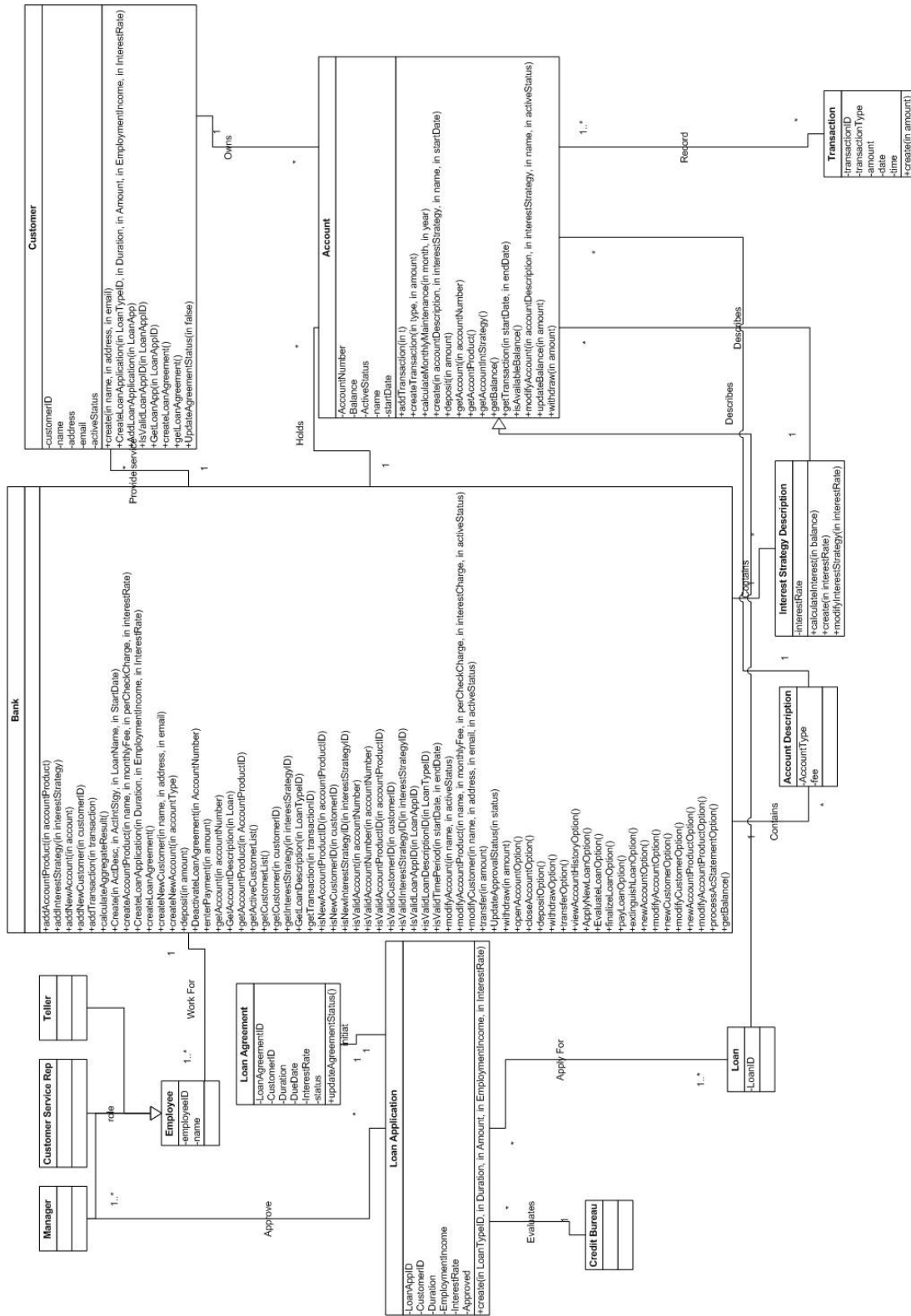
Mr. Warat Chesdavanijkul

<b>CLASS DIAGRAM.....</b>	<b>4</b>
<b>USE CASE 1: OPEN ACCOUNT .....</b>	<b>5</b>
FULL USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	5
SYSTEM SEQUENCE DIAGRAM .....	6
OPERATION CONTRACT .....	7
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	8
<b>USE CASE 2: CLOSE ACCOUNT.....</b>	<b>9</b>
FULL USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	9
SYSTEM SEQUENCE DIAGRAM .....	10
OPERATION CONTRACT .....	11
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	12
<b>USE CASE 3: DEPOSIT FUND.....</b>	<b>13</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	13
SYSTEM SEQUENCE DIAGRAM .....	14
OPERATION CONTRACT .....	15
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	16
<b>USE CASE 4: WITHDRAW FUND.....</b>	<b>17</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	17
SYSTEM SEQUENCE DIAGRAM .....	18
OPERATION CONTRACT .....	19
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	20
<b>USE CASE 5: TRANSFER FUND.....</b>	<b>21</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	21
SYSTEM SEQUENCE DIAGRAM .....	22
OPERATION CONTRACT .....	23
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	24
<b>USE CASE 6: VIEW ACCOUNT HISTORY.....</b>	<b>25</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	25
SYSTEM SEQUENCE DIAGRAM .....	26
OPERATION CONTRACT .....	27
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	28
<b>USE CASE 7: APPLY FOR LOAN .....</b>	<b>29</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	29
SYSTEM SEQUENCE DIAGRAM .....	30
OPERATION CONTRACT .....	31
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	32
<b>USE CASE 8: EVALUATE LOAN.....</b>	<b>33</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	33
SYSTEM SEQUENCE DIAGRAM .....	34
OPERATION CONTRACT .....	35
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	36
<b>USE CASE 9: FINALIZE LOAN.....</b>	<b>37</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	37
SYSTEM SEQUENCE DIAGRAM .....	38
OPERATION CONTRACT .....	39
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	40

<b>USE CASE 10: PAY LOAN INSTALLMENT .....</b>	<b>41</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	41
SYSTEM SEQUENCE DIAGRAM .....	42
OPERATION CONTRACT .....	43
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	44
<b>USE CASE 11: EXTINGUISH LOAN.....</b>	<b>45</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	45
SYSTEM SEQUENCE DIAGRAM .....	46
OPERATION CONTRACT .....	47
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	48
<b>USE CASE 12: MANAGE CUSTOMER (CREATE, MODIFY, DEACTIVATE A CUSTOMER) – CUSTOMER, TELLER AND SERVICE REP.....</b>	<b>49</b>
<b>CREATE NEW CUSTOMER.....</b>	<b>49</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	49
SYSTEM SEQUENCE DIAGRAM .....	50
OPERATION CONTRACT .....	51
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	52
<b>MODIFY CUSTOMER INFORMATION.....</b>	<b>53</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	53
SYSTEM SEQUENCE DIAGRAM .....	54
OPERATION CONTRACT .....	55
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	56
<b>DEACTIVATE CUSTOMER.....</b>	<b>56</b>
<b>USE CASE 13: MANAGE ACCOUNT INFORMATION MANAGE ACCOUNT INFORMATION (CREATE, MODIFY DEACTIVATE ACCOUNT INFO) – CUSTOMER, TELLER, SERVICE REP. ....</b>	<b>57</b>
<b>CREATE NEW ACCOUNT .....</b>	<b>57</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	57
SYSTEM SEQUENCE DIAGRAM .....	58
OPERATION CONTRACT .....	59
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	60
<b>MODIFY ACCOUNT INFORMATION.....</b>	<b>61</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	61
SYSTEM SEQUENCE DIAGRAM .....	62
OPERATION CONTRACT .....	63
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	64
<b>DEACTIVATE ACCOUNT.....</b>	<b>64</b>
<b>USE CASE 14: VIEW ACCOUNT PORTFOLIO .....</b>	<b>65</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	65
SYSTEM SEQUENCE DIAGRAM .....	66
OPERATION CONTRACT .....	67
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	68
<b>USE CASE 15: PROCESS ACCOUNT STATEMENT.....</b>	<b>69</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	69
SYSTEM SEQUENCE DIAGRAM .....	70
OPERATION CONTRACT .....	71
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	72

<b>USE CASE 16: MANAGE ACCOUNT PRODUCT (MANAGE NEW OR EXISTING BANK PRODUCTS, E.G., SENIOR CITIZENS CHECKING ACCOUNT. SAMPLE CHARACTERISTICS OF A PRODUCT INCLUDE A NAME, MONTHLY FEE, PER CHECK CHARGE, PERIODIC INTEREST RATE, ETC).....</b>	<b>73</b>
<b>CREATE NEW ACCOUNT PRODUCT .....</b>	<b>73</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	73
SYSTEM SEQUENCE DIAGRAM .....	74
OPERATION CONTRACT .....	75
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	76
<b>MODIFY ACCOUNT PRODUCT.....</b>	<b>77</b>
USE CASE DESCRIPTION WITH MAJOR ALTERNATIVE FLOWS.....	77
SYSTEM SEQUENCE DIAGRAM .....	78
OPERATION CONTRACT .....	79
SEQUENCE DIAGRAM FOR BASE SCENARIO .....	80
<b>DEACTIVATE ACCOUNT PRODUCT.....</b>	<b>80</b>

# Class Diagram



## Use Case 1: Open Account

### *Full Use Case Description with Major Alternative Flows*

**Pre Conditions:**

None

**Post Conditions:**

- Account is created.

**Basic Flow**

1. User selects open account option. →OpenAccountOption()
2. Go to [Create New Customer](#) use case.
3. Go to [Create New Account](#) use case.
4. Go to [Deposit Funds](#) use case
5. System presents options.

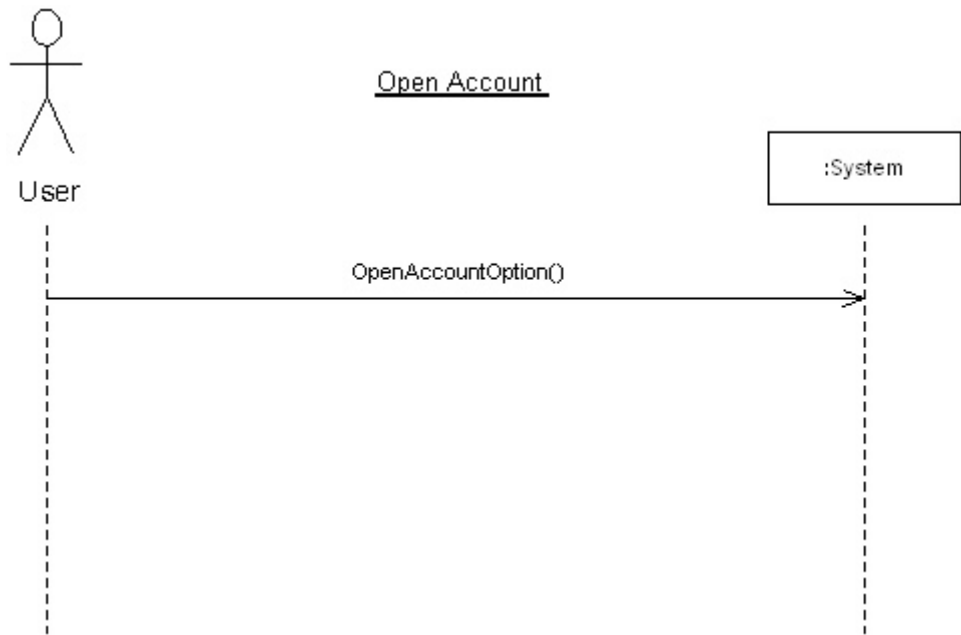
**Alternative Flow**

None

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

**System Sequence Diagram**



Lead Responsibility by Kittipong Techapanichgul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

## ***Operation Contract***

### **Contract C01: OpenAccount**

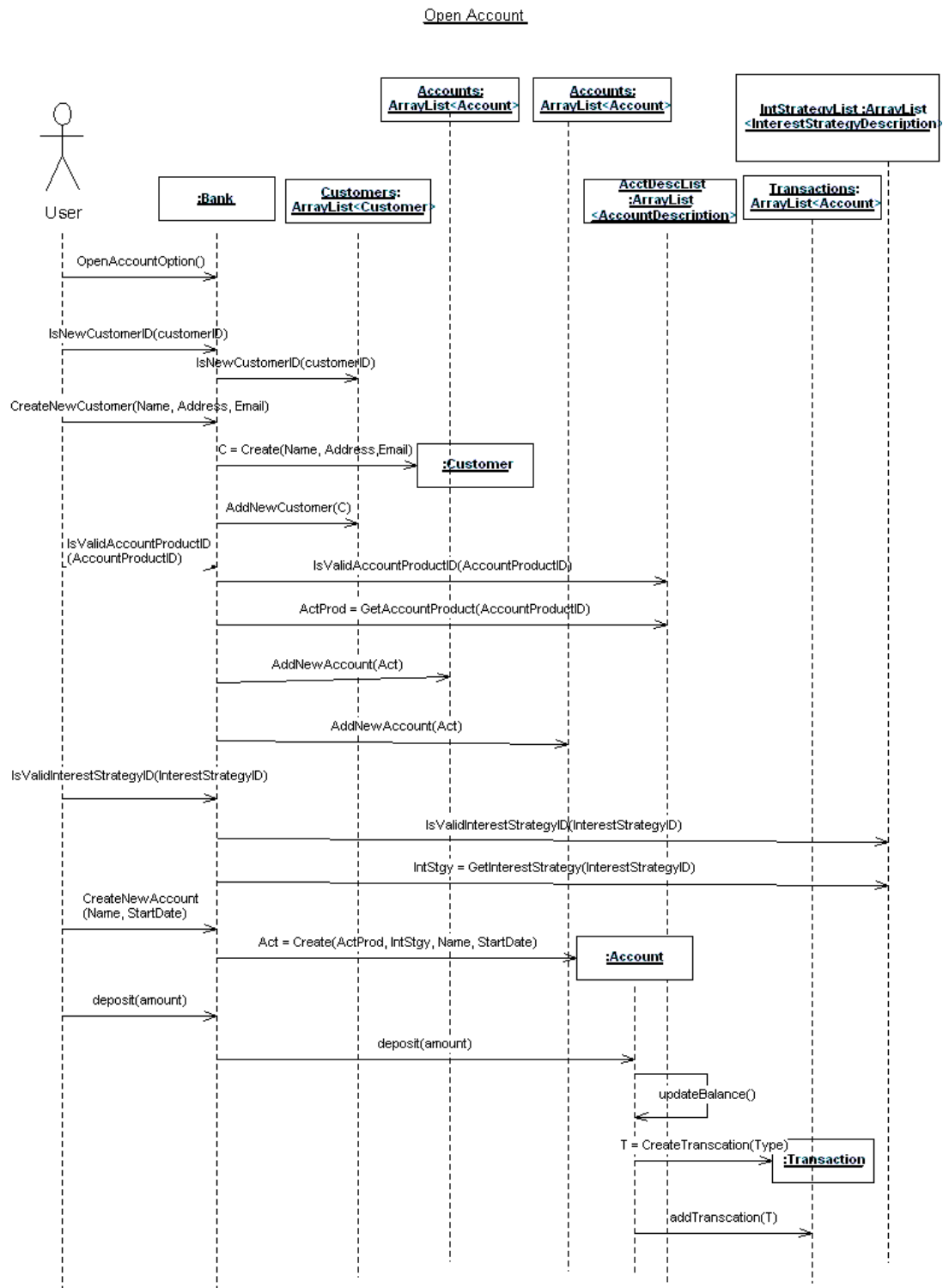
<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
OpenAccountOption()	-	-	-

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul



## Sequence Diagram for Base Scenario



Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

## Use Case 2: Close Account

### *Full Use Case Description with Major Alternative Flows*

#### **Pre condition**

- User is authenticated

#### **Post condition**

- Account is deactivated

#### **Basic Flow**

1. User selects close account option. →CloseAccountOption()
2. System requests customer identification
3. User enters customer identification →IsValidCustomer(customerID)
4. System verifies customer identification.
5. Go to [Withdraw Funds](#) use case.
6. Go to [Deactivate Account](#) use case.
7. System presents options.

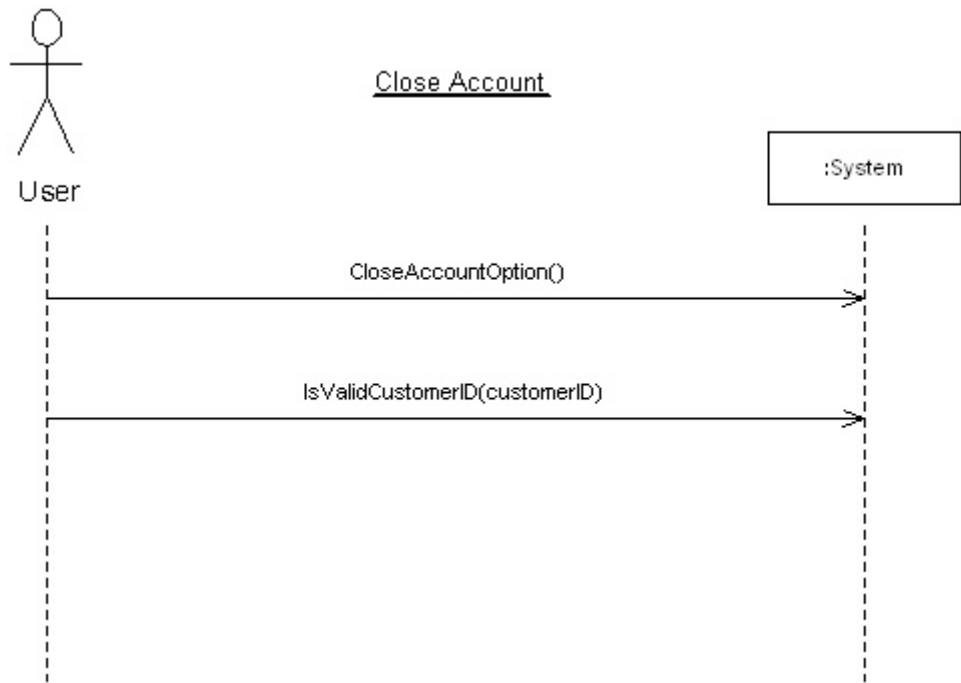
#### **Alternative Flow**

- 3a. User enters wrong customer identification.
  - System presents error message.
  - Return to step 2.

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

**System Sequence Diagram**



Lead Responsibility by Kittipong Techapanichgul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

## ***Operation Contract***

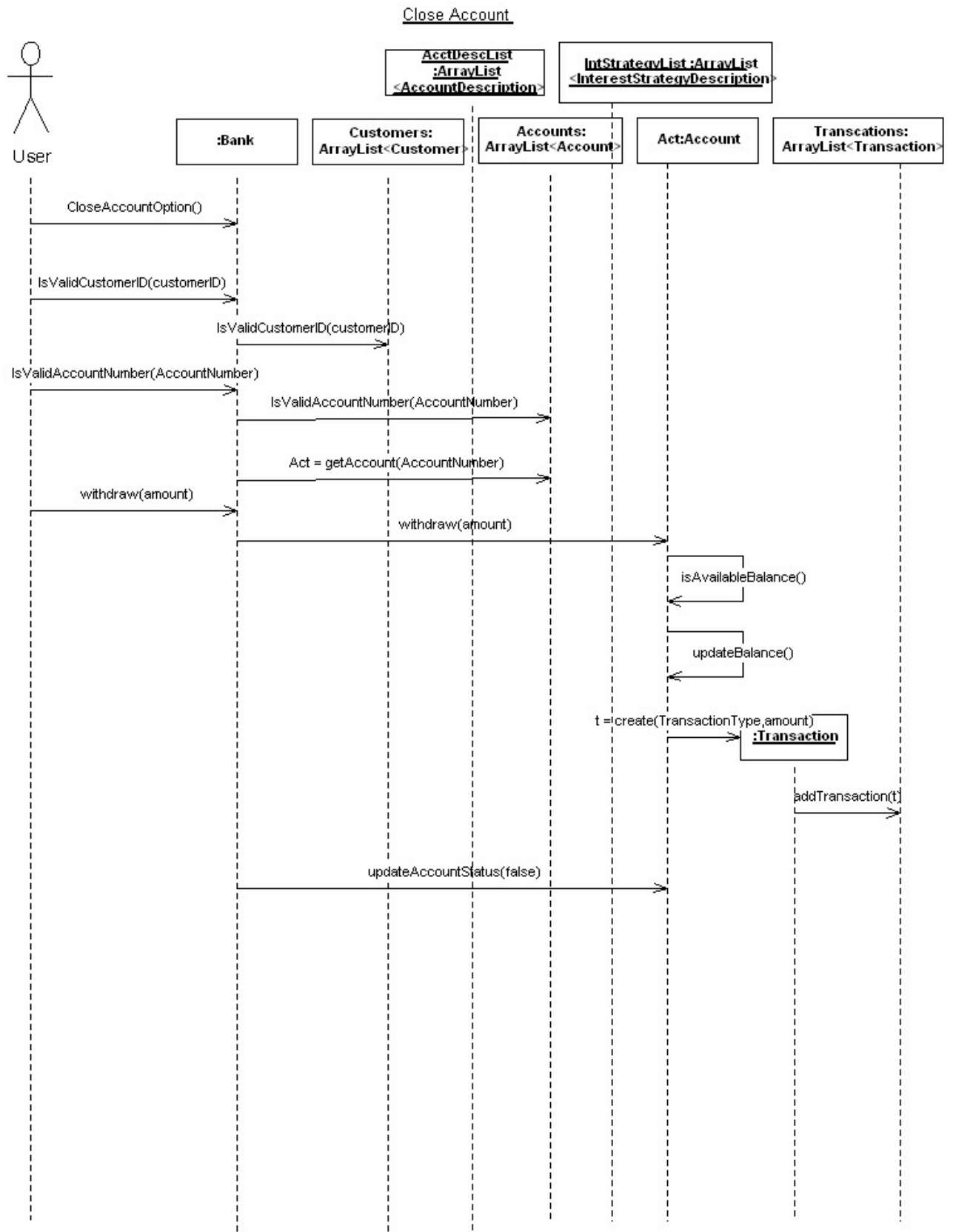
### **Contract C02: CloseAccount**

<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
CloseAccountOption()	-	-	-
IsValidCustomerID(customerID)	-	-	-

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

## Sequence Diagram for Base Scenario



## Use Case 3: Deposit fund

### *Use Case Description with Major Alternative flows*

**Pre Conditions:**

- User is authenticated.

**Post Conditions:**

- Balance is updated.

**Basic Flow**

1. User selects deposit fund option. →depositOption()
2. System requests account number.
3. User enters account number. →isValidAccountNumber(AccountNumber)
4. System verified account number.
5. System requests deposit amount.
6. User enters deposit amount. →deposit(amount)
7. System updates account balance.
8. System generates receipt.
9. System presents user option.

**Alternative Flow**

- a. The use case is requested from open account use case (account number is supplied.)
  1. System starts step 5. to 8.
  2. Resume open account use case.
- b. The use case is requested from transfer fund use case (account number is supplied.)
  1. System starts step 7.
  2. Resume transfer fund use case.
- c. The use case is requested from pay loan installment process (account number is supplied.)
  1. System start step 7.
  2. Resume pay loan installment use case.
- 4a. Account information is invalid<sup>1</sup>.
  1. System presents error message.
  2. Resume step 2.

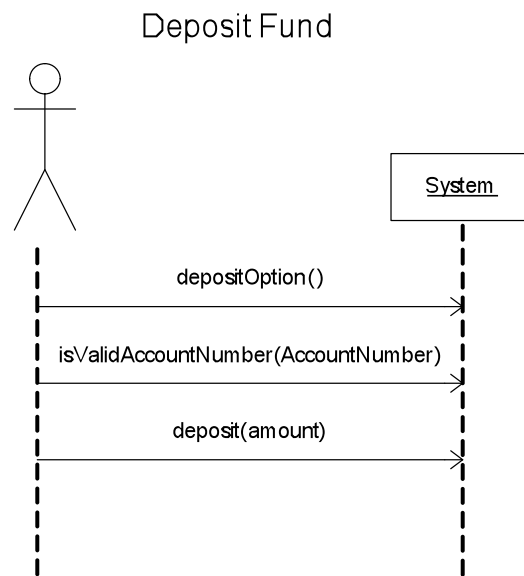
Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

---

<sup>1</sup> Invalid means the account is not found in the system or it is not in active status

## System Sequence Diagram



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## ***Operation Contract***

### **Contract C03: Deposit Fund**

<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
depositOption()	-	-	-
isValidAccountNumber(AccountNumber)	-	-	-
deposit(amount)	- Transaction	Associates Transaction with Account	- Transaction.Amount = amount - Acct.Balance = Acct.Balance + amount

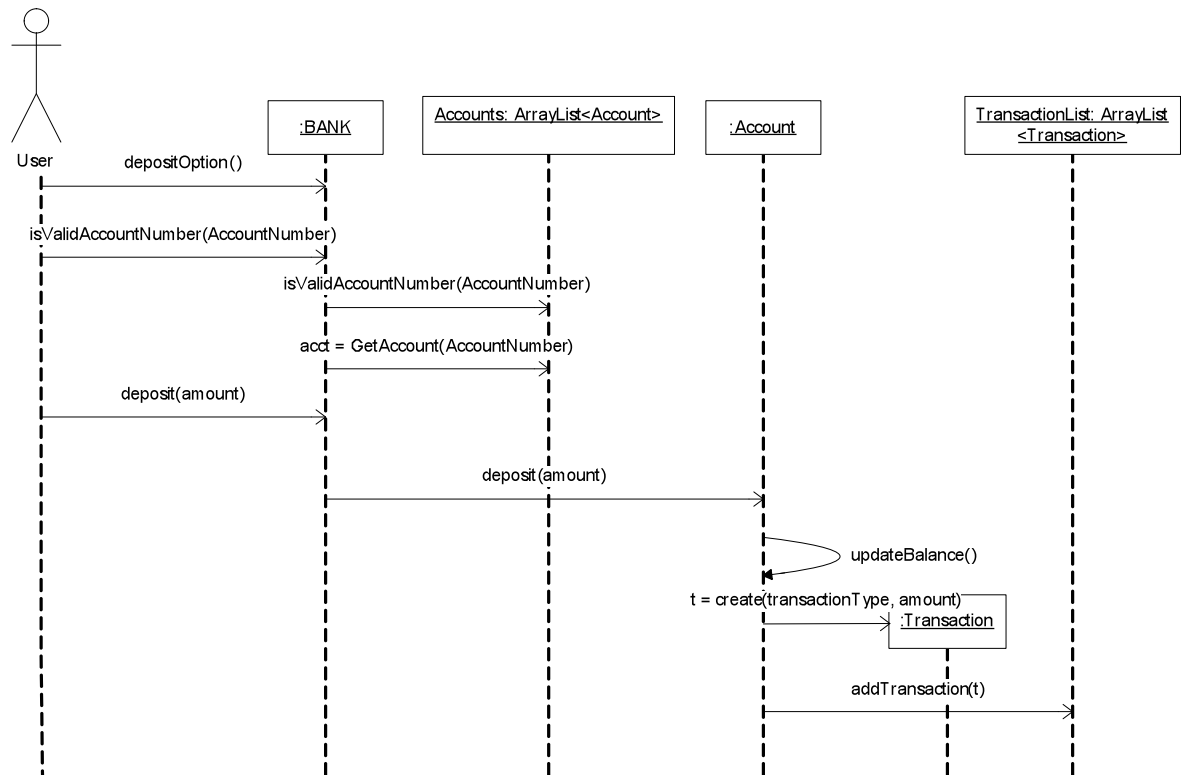
Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario

### 3. Deposit Fund



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Use Case 4: Withdraw fund

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- User is authenticated.

#### **Post Conditions:**

- Balance is updated.

#### **Basic Flow**

1. User selects withdraw fund option. →withdrawOption()
2. System request account number.
3. User enters account number. →isValidAccountNumber(AccountNumber)
4. System verifies account number.
5. System requests withdraw amount.
6. User enters withdraw amount. →withdraw(amount)
7. System verifies available balance.
8. System updates account balance.
9. System generates receipt.
10. System presents user option.

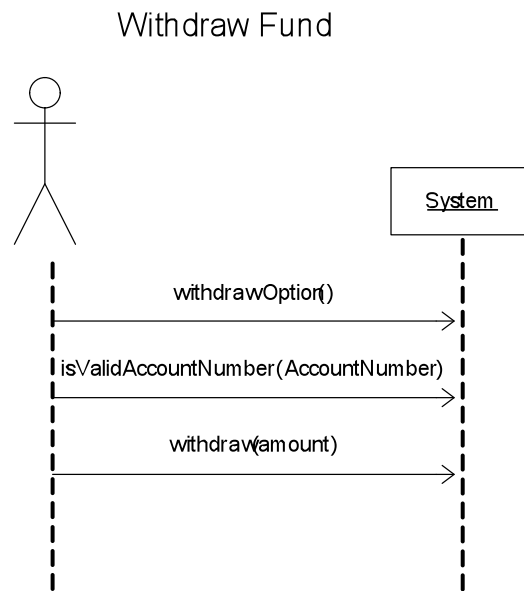
#### **Alternative Flow**

- a. The use case is requested from close account use case (account number is supplied.)
  1. System starts step 2. to 9.
  2. Resume close account process.
- b. The use case is requested from transfer fund use case (account number is supplied.)
  1. System starts step 8.
  2. Resume transfer fund process.
- 4a. Account information is invalid.
  1. System presents error message.
  2. Resume step 2.
- 7a. Available balance is lower than withdraw amount.
  1. System presents error message.
  2. Resume step 5.
- 7b. balance is equal to zero.
  - System presents error message.
  - Resume close account process.

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## ***Operation Contract***

### **Contract C04: Withdraw Fund**

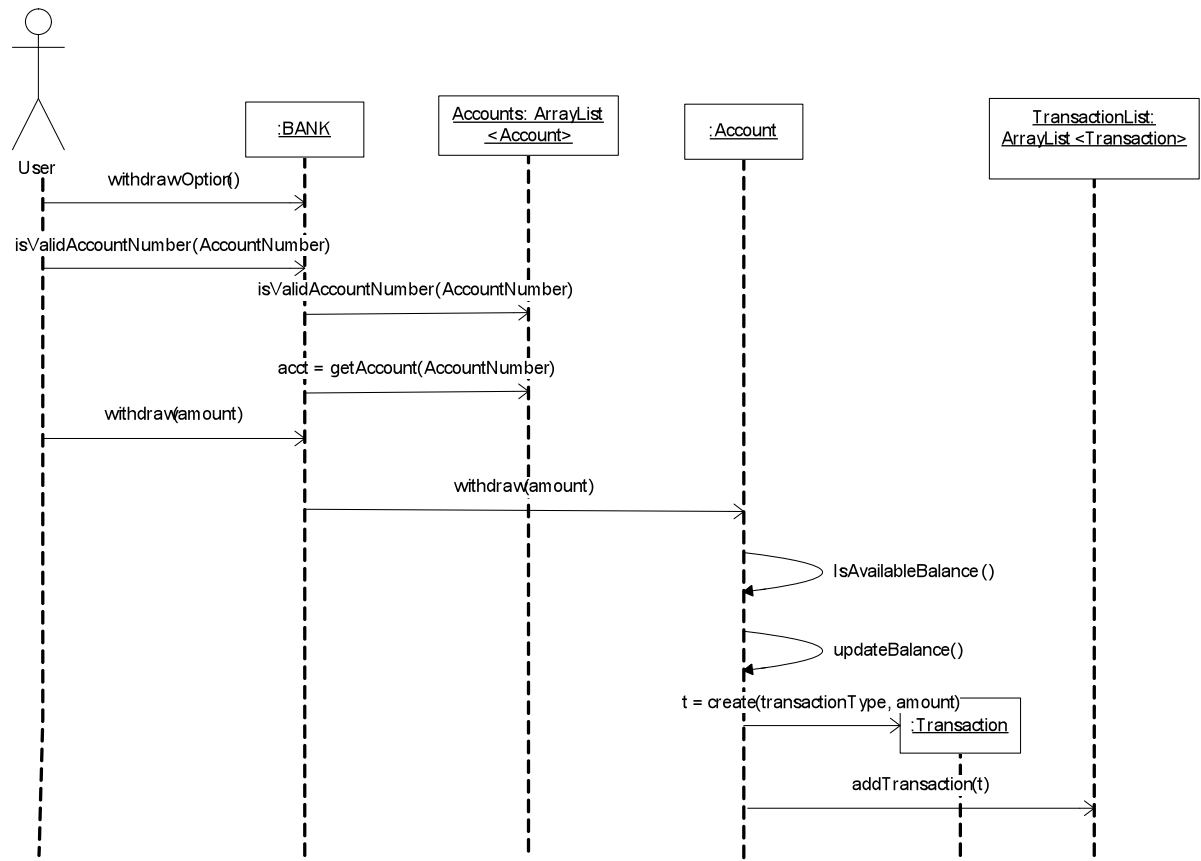
<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
withdrawOption()	-	-	-
isValidAccountNumber(AccountNumber)	-	-	-
withdraw(amount)	- Transaction	Associates Transaction with Account	- Transaction.Amount = amount - Acct.Balance = Acct.Balance - amount

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

**Sequence Diagram for Base Scenario**

**4. Withdraw Fund**



Lead Responsibility by Jaruwan Laptrakool  
Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Use Case 5: Transfer fund

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- User is authenticated.

#### **Post Conditions:**

- Balance is updated.
- Fund is successfully transferred

#### **Basic Flow**

1. User selects transfer fund option. →transferOption()
2. System requests source account number.
3. User enters source account number. →isValidAccountNumber(srcAccountNumber)
4. System verifies source account.
5. System requests destination account number.
6. User enters destination account number. →isValidAccountNumber(destAccountNumber)
7. System verifies destination account.
8. System requests amount to transfer.
9. User enters amount. →transfer(amount)
10. System verifies available balance in source account.
11. System updates account balances.
12. System generates receipt.
13. System presents user option.

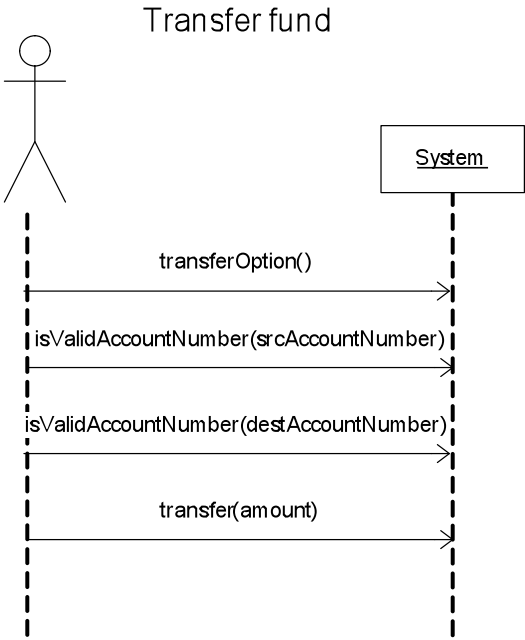
#### **Alternative Flow**

- 4a. Source account is invalid.
  1. System presents error message.
  2. Resume step 2.
- 7a. Destination account is invalid.
  1. System presents error message.
  2. Resume step 5.
- 10a. Source account's balance is not sufficient to transfer.
  1. System presents error message.
  2. Resume step 8.
- 10b. Amount exceeds the transferred limit.
  1. System presents error message.
  2. Resume step 8.

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

**System Sequence Diagram**



Lead Responsibility by Jaruwan Laptrakool  
Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

### Contract C05: Transfer Fund

Operation Name	Instances Created	Associations Formed	Attributes Modified
transferOption()	-	-	-
isValidAccountNumber(srcAccount Number)	-	-	-
isValidAccountNumber(destAccount Number)	-	-	-
transfer(amount)	- Transaction1 - Transaction2	- Associates Transaction1 with Account - Associates Transaction2 with Account	- Transaction1.Amount = amount - sourceAcct.Balance = sourceAcct.Balance – amount - Transaction2.Amount = amount - destAcctBalance = destAcct.Balance + amount

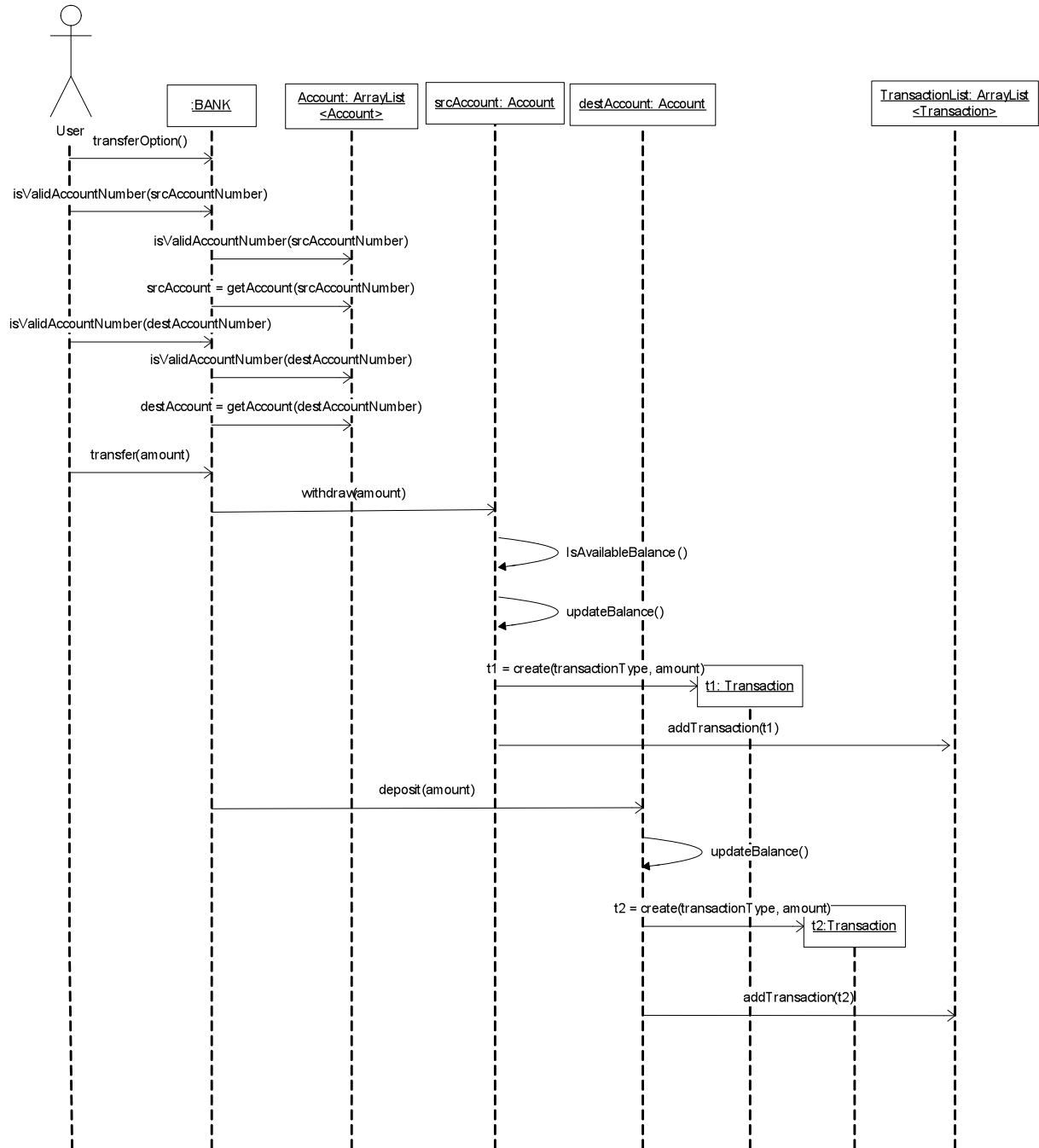
Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario

### 5. Transfer Fund



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Use Case 6: View Account History

### *Use Case Description with Major Alternative flows*

**Pre Conditions:**

- User is authenticated

**Post Conditions:**

- None

**Basic Flow**

1. User selects view account history option. →ViewAccountHistoryOption()
2. System requests account number.
3. User enters account number. →isValidAccountNumber(AccountNumber)
4. System verifies account number.
5. System requests start date and end date.
6. User enters start date and end date. →isValidTimePeriod(StartDate, EndDate)
7. System verifies time period.
8. System gets all transactions based on account number within time period.
9. System presents all the transactions.
10. System present user options

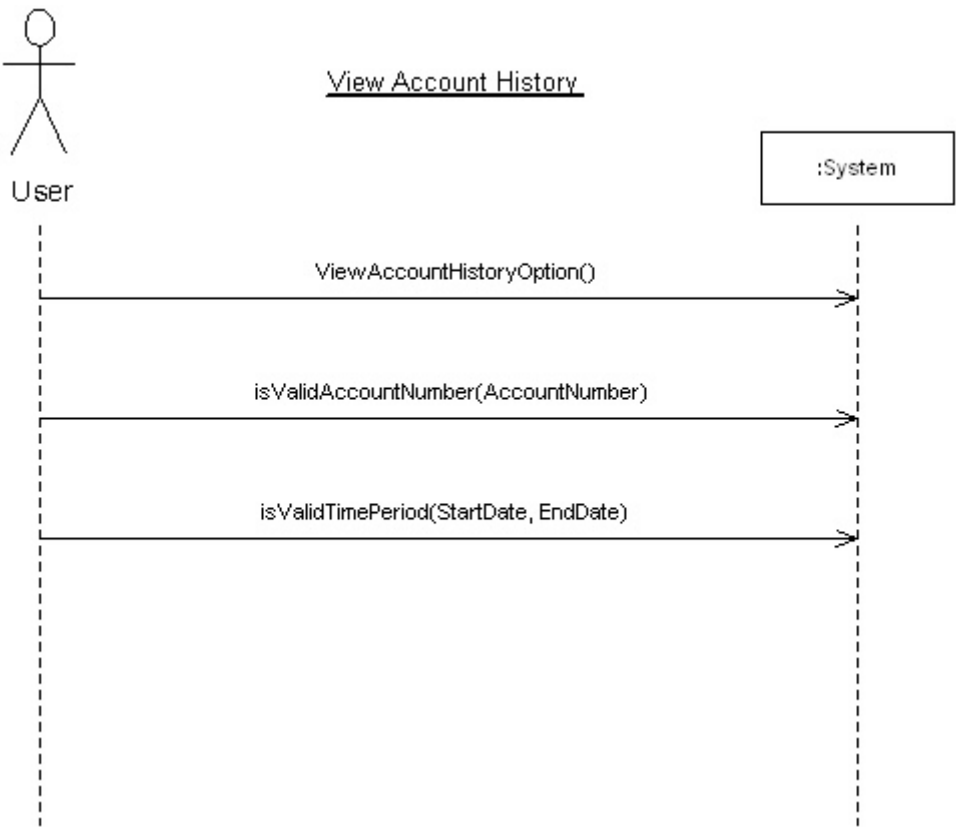
**Alternative Flow**

- 4a. User enters incorrect account number.
  1. System presents error message.
  2. Return to step 2.
- 7a. User enters invalid time period
  1. System presents error message.
  2. Return to step 5.

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Jaruwan Laptrakool

**System Sequence Diagram**



Lead Responsibility by Kittipong Techapanichgul  
Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Jaruwan Laptrakool

## ***Operation Contract***

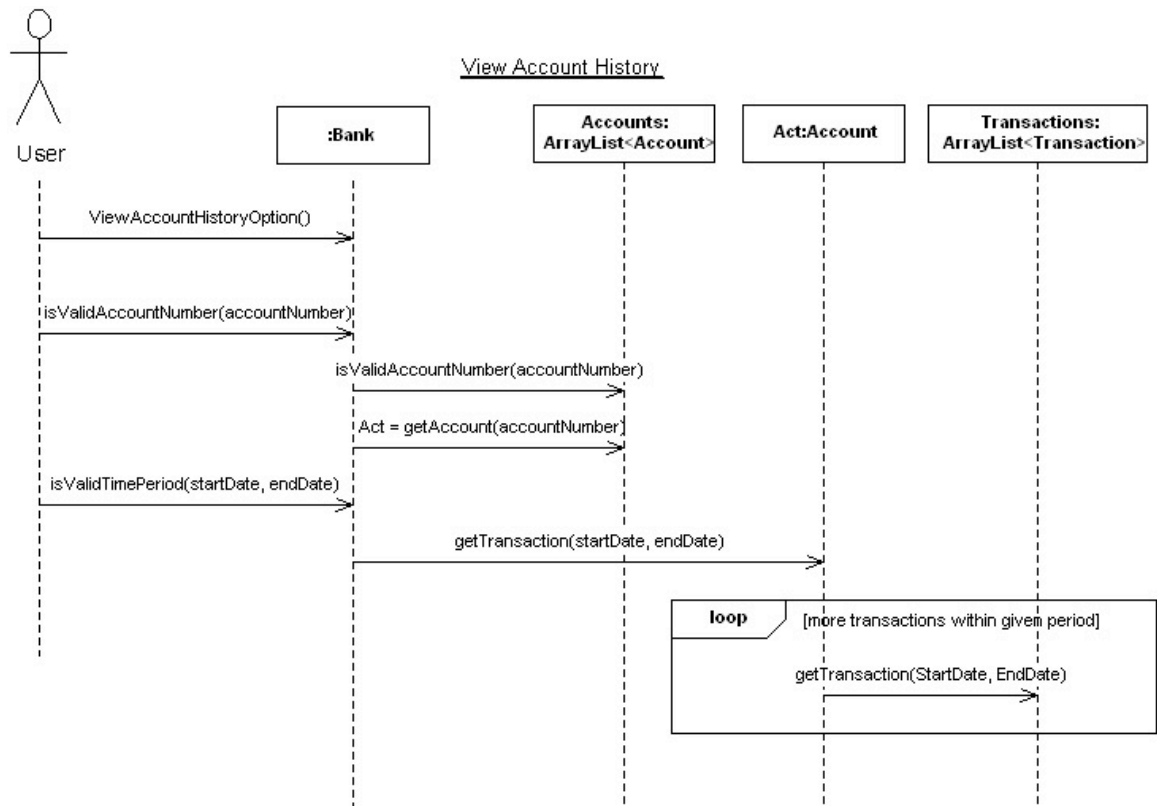
### **Contract C06: ViewAccountHistory**

<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
ViewAccountHistoryOption()	-	-	-
isValidAccountNumber(AccountNumber)	-	-	-
isValidTimePeriod(StartDate, EndDate)	-	-	-

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Jaruwan Laptrakool

## Sequence Diagram for Base Scenario



Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Jaruwan Laptrakool

## Use Case 7: Apply For Loan

### *Use Case Description with Major Alternative flows*

#### **Pre Condition**

None

#### **Post condition**

- Loan application is completed for evaluation.

#### **Basic Flow**

1. User selects apply for a loan option. → ApplyNewLoanOption()
2. System creates new loan application.
3. System request customer ID
4. User enters customer ID. → IsValidCustomerID(CustomerID)
5. System verifies customer ID.
6. System requests loan type ID.
7. User enters loan type ID. → IsValidLoanDescriptionID(LoanTypeID)
8. System verifies loan type ID.
9. System requests loan information.
  - Duration
  - Amount
  - Employment
  - Income
  - Interest rate
10. User enters loan information. → CreateLoanApplication(Duration, Amount, Employment Income, InterestRate)
11. System validates loan information.
12. System creates loan application.
13. System records logs for loan application.
14. System presents options.

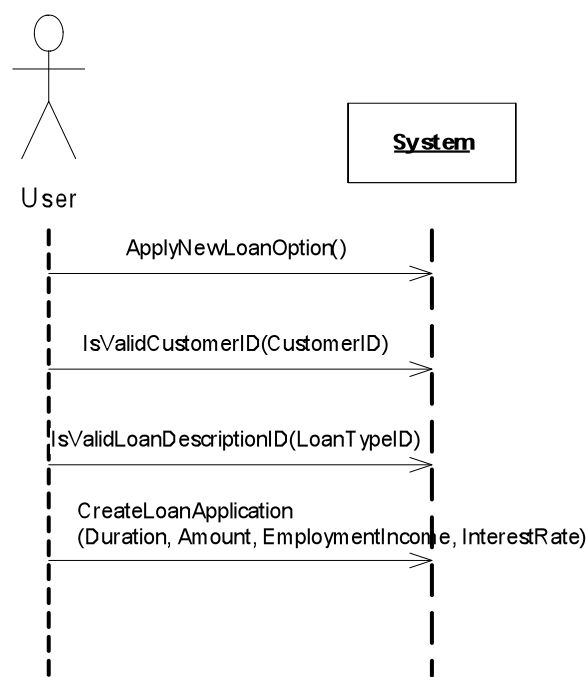
#### **Alternative Flow**

- 5a. Customer is new.
  1. Go to [Create New Customer](#) use case.
  2. Return to step 3.
- 8a. Loan type ID is invalid. (Not active or not exist)
  3. System presents invalid loan type ID
  4. Return to step 6
- 11a. Loan information is incomplete.
  1. System presents incomplete loan application.
  2. Return to step 9.

Lead Responsibility by Derek Pawlikowsky

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

**System Sequence Diagram**



Lead Responsibility by Derek Pawlikowsky  
Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

## Operation Contract

### Contract C07: ApplyForLoan

Operation Name	Instances Created	Associations Formed	Attributes Modified
ApplyNewLoanOption()			
IsValidCustomerID(customerID)			
IsValidLoanDescriptionID(LoanTypeID)			
CreateLoanApplication(Duration, Amount, employment Income, InterestRate)	- LoanApplication	- LoanApplication with bank - LoanApplication with customer - LoanDescription with LoanApplication	- LoanApplication. Duration = Duration - LoanApplication. Amount = amount - LoanApplication. EmploymentIncome = EmploymentIncome - LoanApplication. InterestRate = InterestRate - LoanApplication. Approved = false

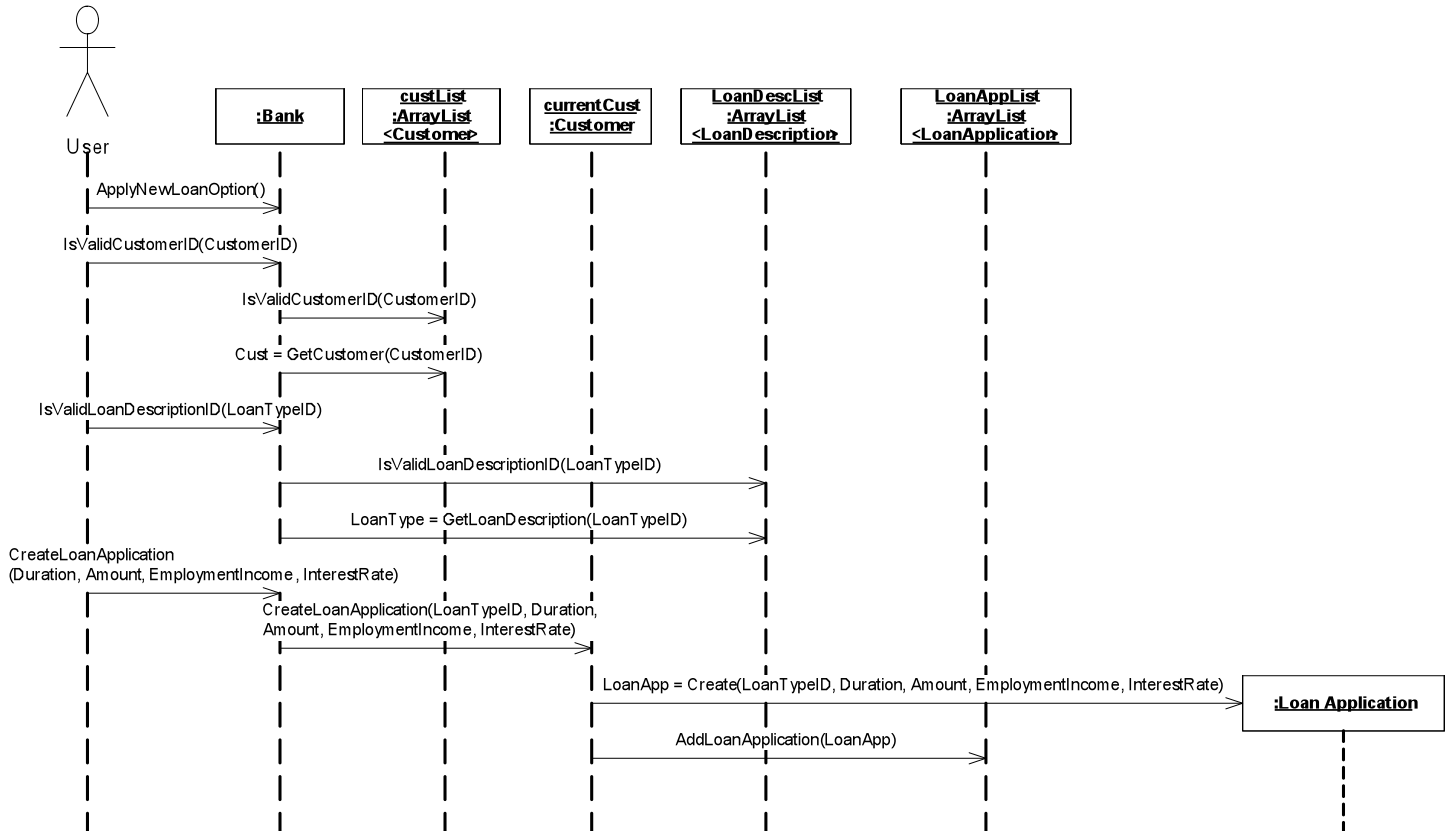
Lead Responsibility by Derek Pawlikowsky

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul



## Sequence Diagram for Base Scenario

Apply For Loan



Lead Responsibility by Derek Pawlikowsky

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

## Use Case 8: Evaluate Loan

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- User is authenticated
- Customer has applied for a loan

#### **Post Conditions:**

- Loan application status is updated

#### **Basic Flow**

1. User selects evaluate loan option. → evaluateLoanOption()
2. System requests customer ID.
3. User enters customer ID. → isValidCustomer(customerID)
4. System verifies customer ID.
5. System requests loan application ID.
6. User enters loan application ID. → isValidLoanAppID(loanAppID)
7. System verifies load application ID.
8. System submits customer and loan information to Credit Bureau system.
9. System requests credit score from Credit Bureau system.
10. System receives credit score from Credit Bureau system.
11. System displays customer information, loan information and his credit score.
12. System requests loan approval.
13. User enters loan application approval status. → updateApprovalStatus(status)
14. System updates loan application status
15. System presents user option.

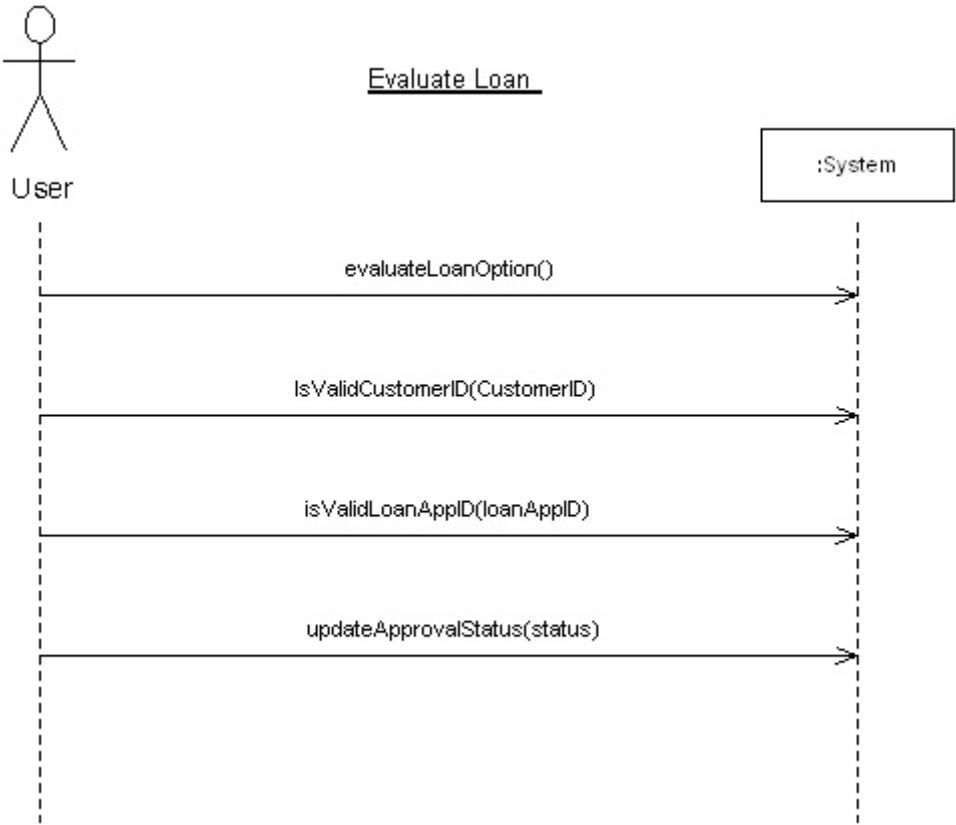
#### **Alternative Flow**

- 4a. Customer ID is invalid
  1. System presents error message.
  2. Resume step 2.
- 7a. Loan application ID is invalid
  1. System presents error message.
  2. Resume step 5.

Lead Responsibility by Derek Pawlikowsky

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

# System Sequence Diagram



Lead Responsibility by Derek Pawlikowsky  
 Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

## ***Operation Contract***

### **Contract C08: Evaluate Loan**

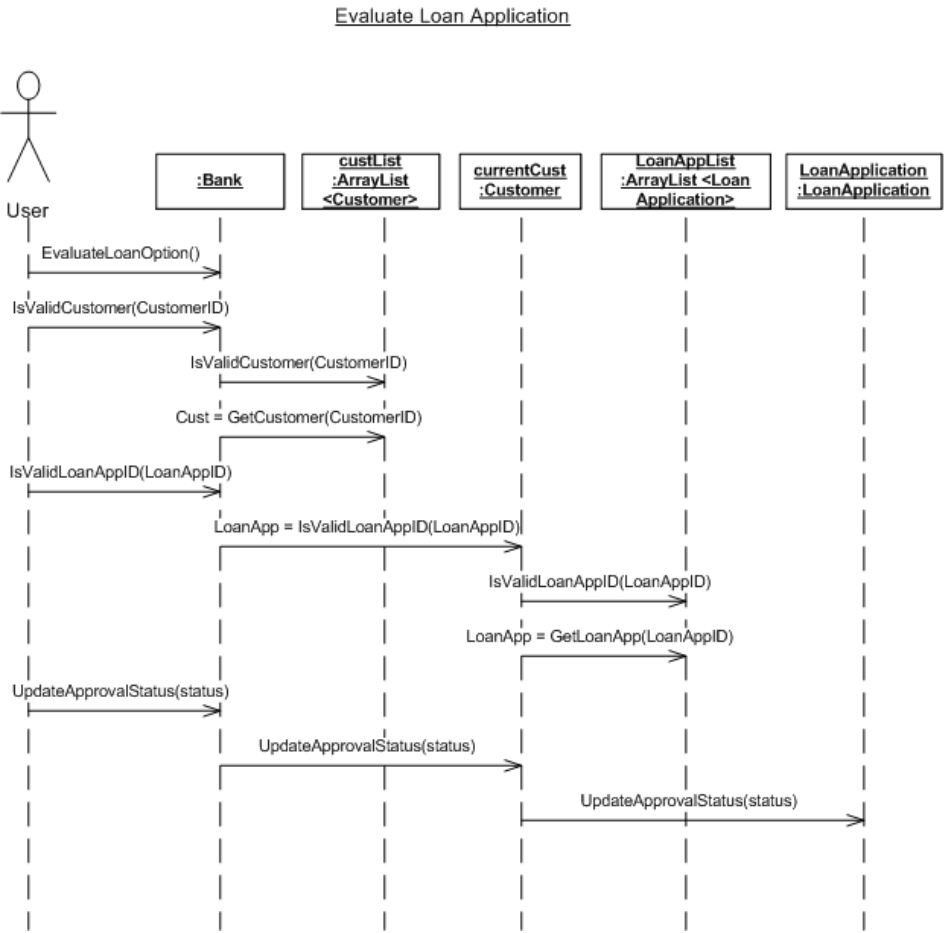
<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
evaluateLoanOption()	-	-	-
IsValidCustomerID(CustomerID)	-	-	-
IsValidLoanAppID(loanAppID)	-	-	-
updateApprovalStatus(status)	-	-	LoanApplication.Approved = status

Lead Responsibility by Derek Pawlikowsky

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

Sequence Diagram for Base Scenario

TEAM 3  
BIS Sequence Diagram  
UC08-Evaluate Loan Application



Lead Responsibility by Derek Pawlikowsky  
Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Warat Chesdavanijkul

## Use Case 9: Finalize Loan

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- User is authenticated.
- The loan is approved.

#### **Post Conditions:**

- Loan is created.

#### **Basic Flow**

1. User selects finalize loan option. → finalizeLoanOption()
2. System requests customer ID.
3. User enters customer ID. → IsValidCustomerID(CustomerID)
4. System verifies Customer ID.
5. System requests loan application ID.
6. User enters loan application ID. → IsValidLoanAppID(LoanAppID)
7. System verifies loan application ID.
8. System presents loan information.
9. System requests approval of loan.
10. User approves loan. → createLoanAgreement()
11. Go to [Create New Account](#) use case.
12. System generates the loan agreement.
13. System updates loan application status.
14. System present user options.

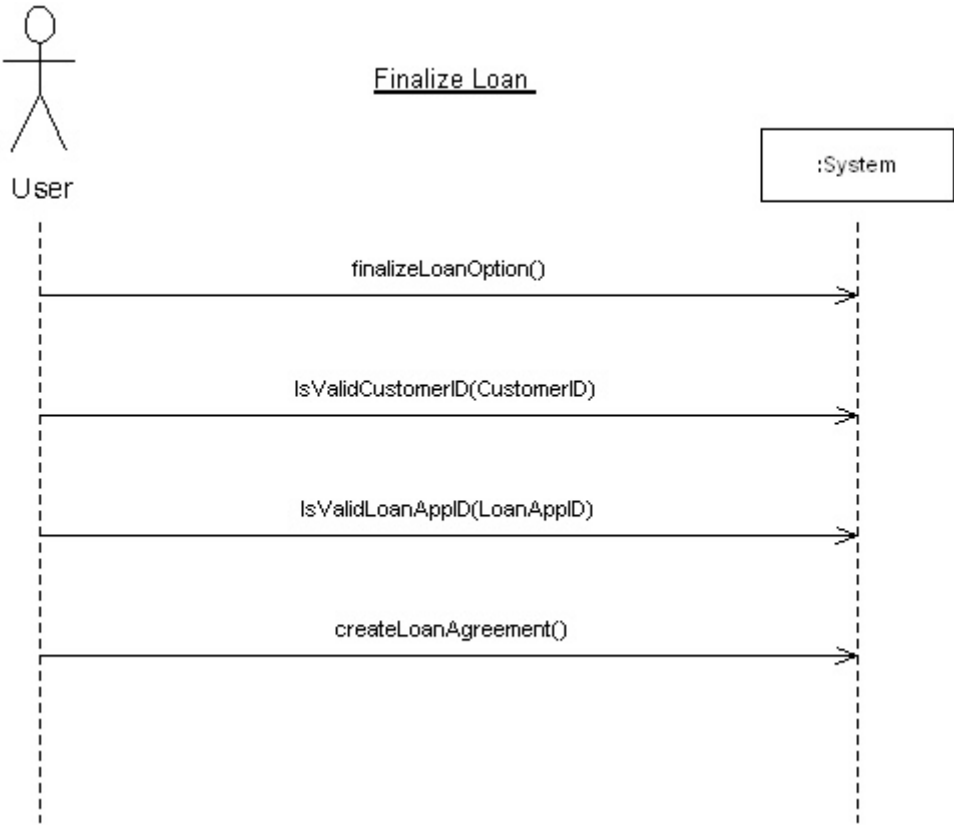
#### **Alternative Flow**

- 4a. Customer ID is invalid.
  1. System presents invalid customer ID.
  2. Return to step 2.
- 7a. Loan application ID is invalid.
  1. System presents invalid loan application ID.
  2. Return to step 4.
- 10a. User cancels approval of loan.
  1. Go to step 14.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Derek Pawlikowsky

**System Sequence Diagram**



Lead Responsibility by Warat Chesdavanijkul  
Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Derek Pawlikowsky

## ***Operation Contract***

### **Contract C09: Finalize Loan**

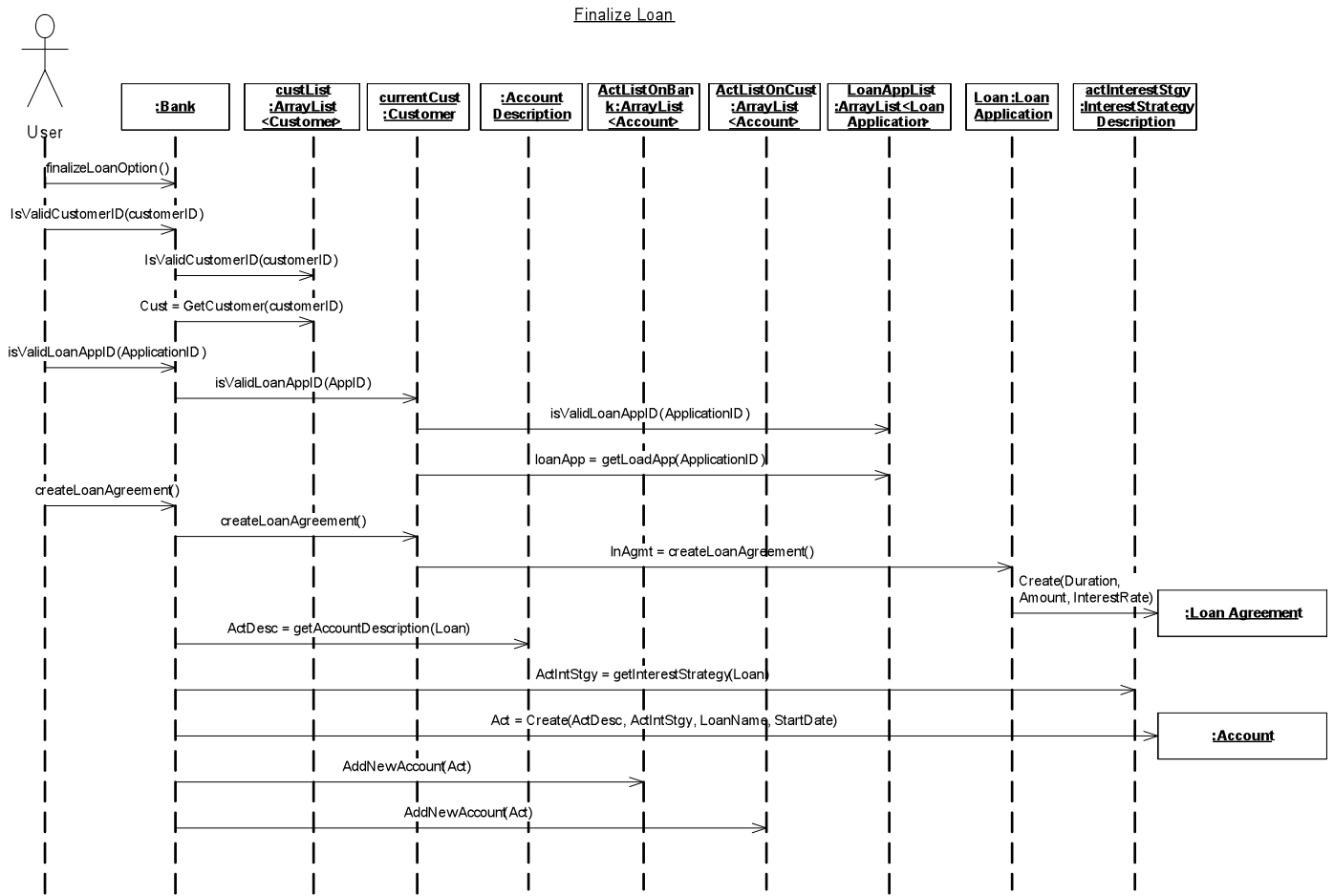
<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
finalizeLoanOption()	-	-	-
IsValidCustomerID(CustomerID)	-	-	-
IsValidLoanAppID(LoanAppID)	-	-	-
createLoanAgreement()	- LoanAgreement	LoanAgreement with Loan Application	-

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Derek Pawlikowsky



## Sequence Diagram for Base Scenario



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Derek Pawlikowsky

## Use Case 10: Pay loan installment

### *Use Case Description with Major Alternative flows*

#### **Pre conditions:**

- User is authenticated.

#### **Post conditions:**

- Loan balance is updated.

#### **Basic Flow**

1. User selects pay loan. → payLoanOption()
2. System requests loan account number.
3. User enters loan account number. → isValidAccountNumber(AccountNumber)
4. System verifies loan account.
5. User enters payment amount → enterPayment(amount)
6. System verifies payment amount.
7. System updates loan account balance.
8. System generates receipt.
9. System presents user option.

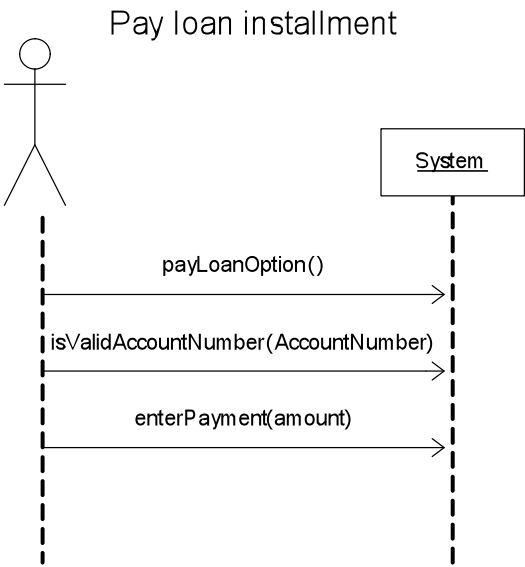
#### **Alternative Flow**

- 4a. Loan account number is invalid.
  1. System presents error message.
  2. Resume step 2.
- 6a. Payment amount is lower than minimum payment
  1. System present error message.
  2. Resume step 5.

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

**System Sequence Diagram**



Lead Responsibility by Jaruwan Laptrakool  
Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

### Contract C10: Pay Loan Installment

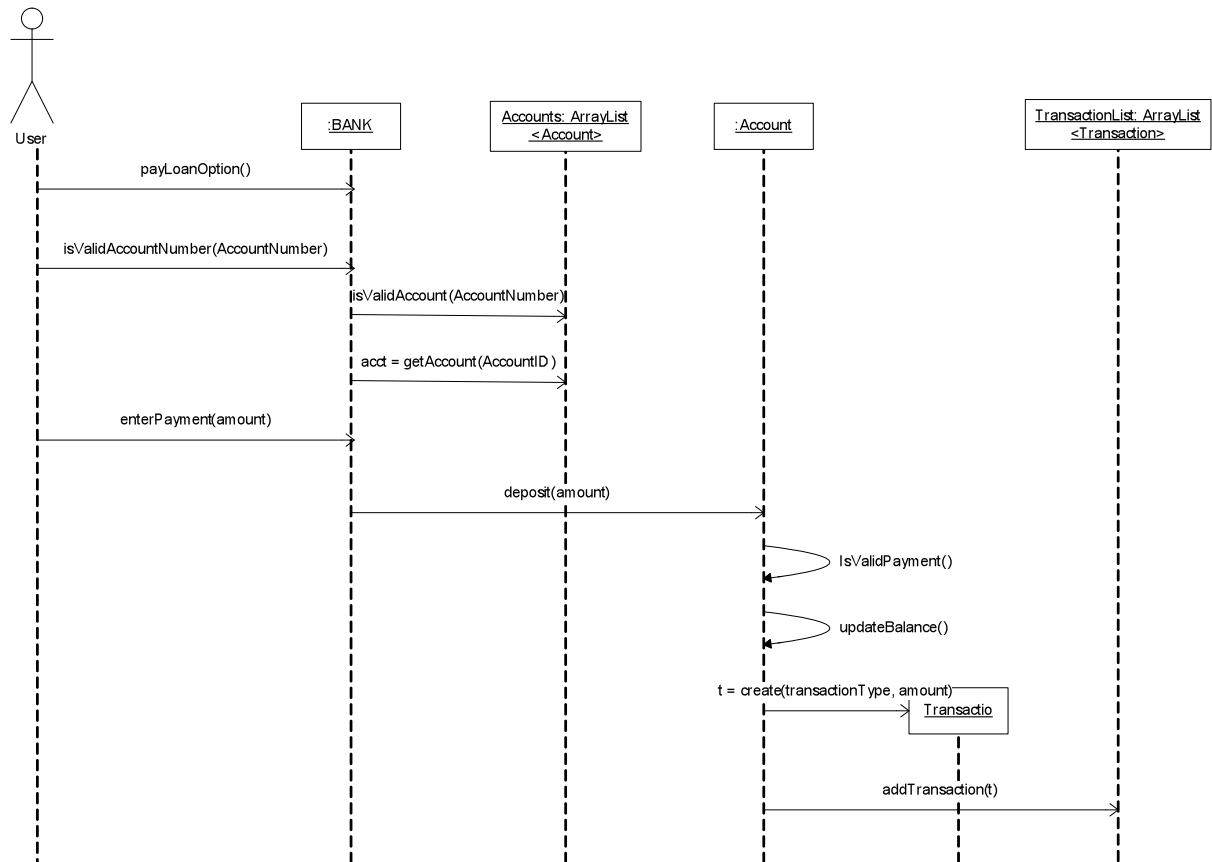
Operation Name	Instances Created	Associations Formed	Attributes Modified
payLoanOption()	-	-	-
isValidAccountNumber(AccountNumber)	-	-	-
enterPayment(amount)	- Transaction	Associates Transaction with LoanAccount	- Transaction.Amount = amount - loanAcct.Balance = loanAcct.Balance – amount

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Sequence Diagram for Base Scenario

### 10. Pay Loan Installment



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## Use Case 11: Extinguish Loan

### *Use Case Description with Major Alternative flows*

**Pre conditions:**

- User is authenticated.
- Loan account exists.

**Post conditions:**

- Loan account is closed.
- Loan agreement is deactivated.

**Basic Flow**

1. User selects extinguish loan option. → `extinguishLoanOption()`
2. System requests customer ID.
3. User enters customer ID. → `IsValidCustomerID(CustomerID)`
4. System verifies Customer ID.
5. System presents loan account information.
6. System request extinguished loan account.
7. User enters loan account number. → `IsValidAccountNumber(AccountNumber)`
8. System verifies loan account balance.
9. System deactivates loan account.
10. System deactivates loan agreement.
11. System presents user option.

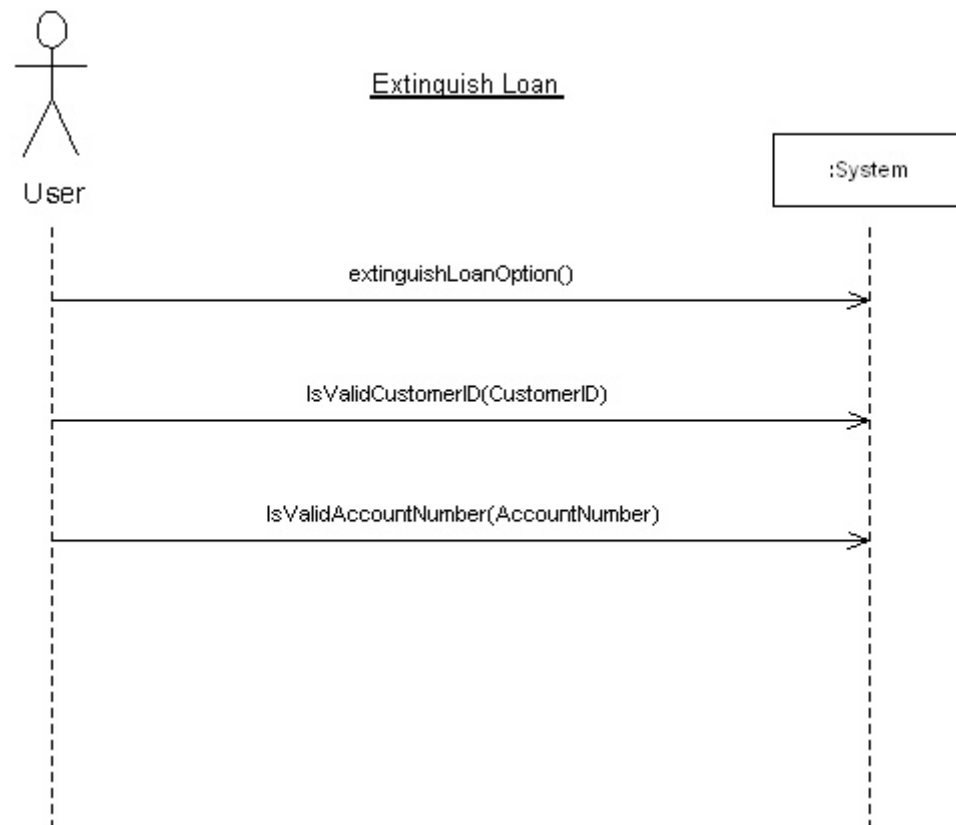
**Alternative Flow**

- 4a. Customer ID is invalid.
  3. System presents invalid customer ID.
  2. Return to step 2.
- 8b. balance is negative
  3. Go to Pay Loan Installment use case.
  4. Return to step 8.

Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram



Lead Responsibility by Jaruwan Laptrakool

Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul

**Operation Contract**

**Contract C11: Extinguish loan**

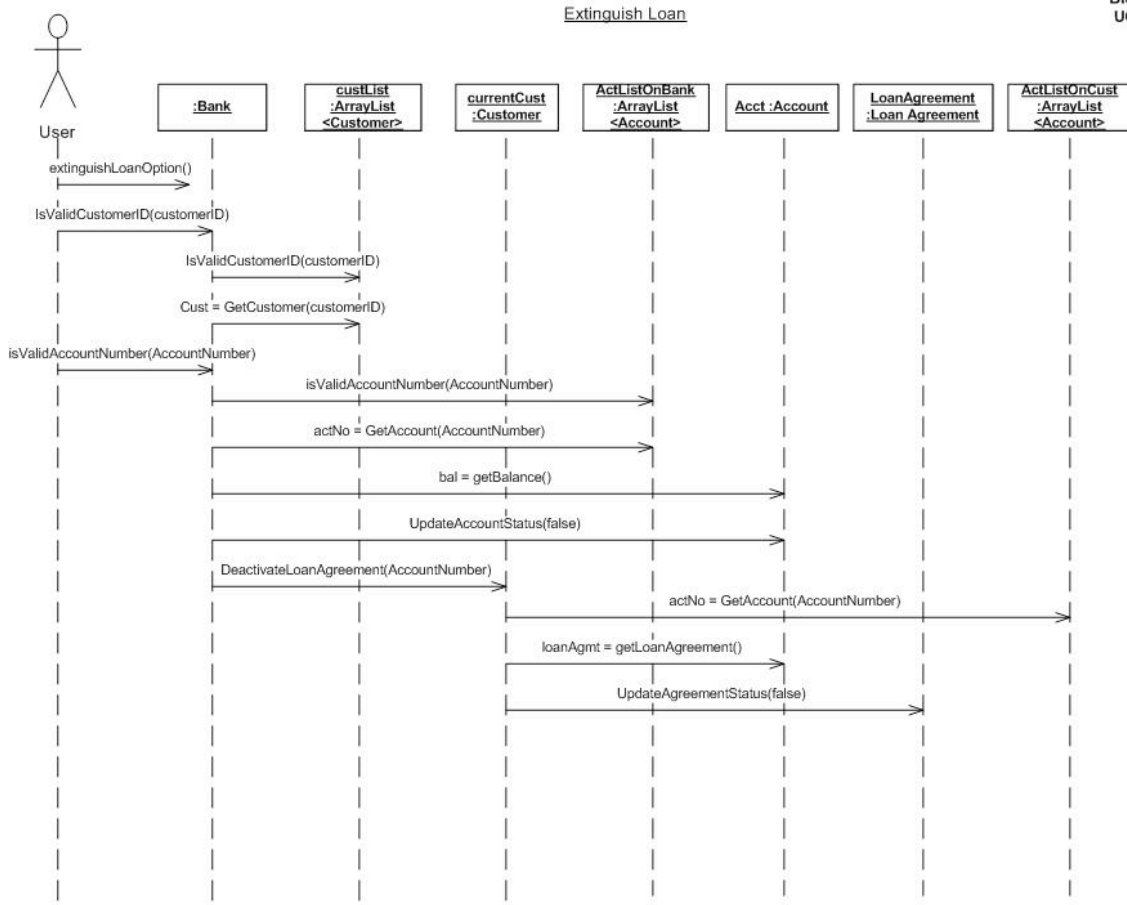
Operation Name	Instances Created	Associations Formed	Attributes Modified
extinguishLoanOption()	-	-	-
IsValidCustomerID(CustomerID)	-	-	-
IsValidAccountNumber(AccountNumber)	-	-	-

Lead Responsibility by Jaruwan Laptrakool  
Substantive assistance by Warat Chesdavanijkul, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario

TEAM 3  
BIS Sequence Diagram  
UC11-Extinguish Loan



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Kittipong Techapanichgul,, Jaruwan Laptrakool and Derek Pawlikowsky

## Use Case 12: Manage customer (Create, modify, deactivate a customer) – Customer, Teller and Service Rep.

### Create New Customer

#### *Use Case Description with Major Alternative flows*

##### **Preconditions:**

Service Rep. is authenticated.

##### **Post conditions:**

Customer is created.

##### **Basic Flow**

1. User selects create new customer option. → NewCustomerOption()
2. System requests customer ID. (SSN or Passport ID)
3. User enters customer ID. → IsNewCustomerID(CustomerID)
4. System verifies customer ID.
5. System requests customer information.  
User enters customer information. → CreateNewCustomer(Name, Address, Email)
6. System verifies customer information.
7. System creates new customer
8. System presents user options.

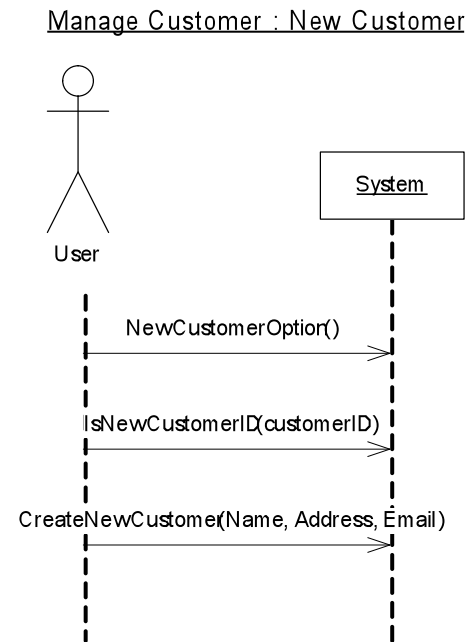
##### **Alternative flow**

- a. The use case is requested from [Open Account](#) use case
  1. System starts step 2. to 7.
  2. Resume open account process.
- b. The use case is requested from [Apply for Loan](#) use case
  2. System starts step 2. to 7.
  2. Resume open account process.
- 4a. System detects duplicate customer ID
  1. System presents customer ID already exist on the system.
  2. Return to step 9.
- 7a. System detects missing information.
  1. System presents error message.
  2. Return to step 5.
- 7b. System detects duplicate customer
  3. System presents customer already exist on the system.
  4. Return to step 5.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

### Contract C121: CreateNewCustomer

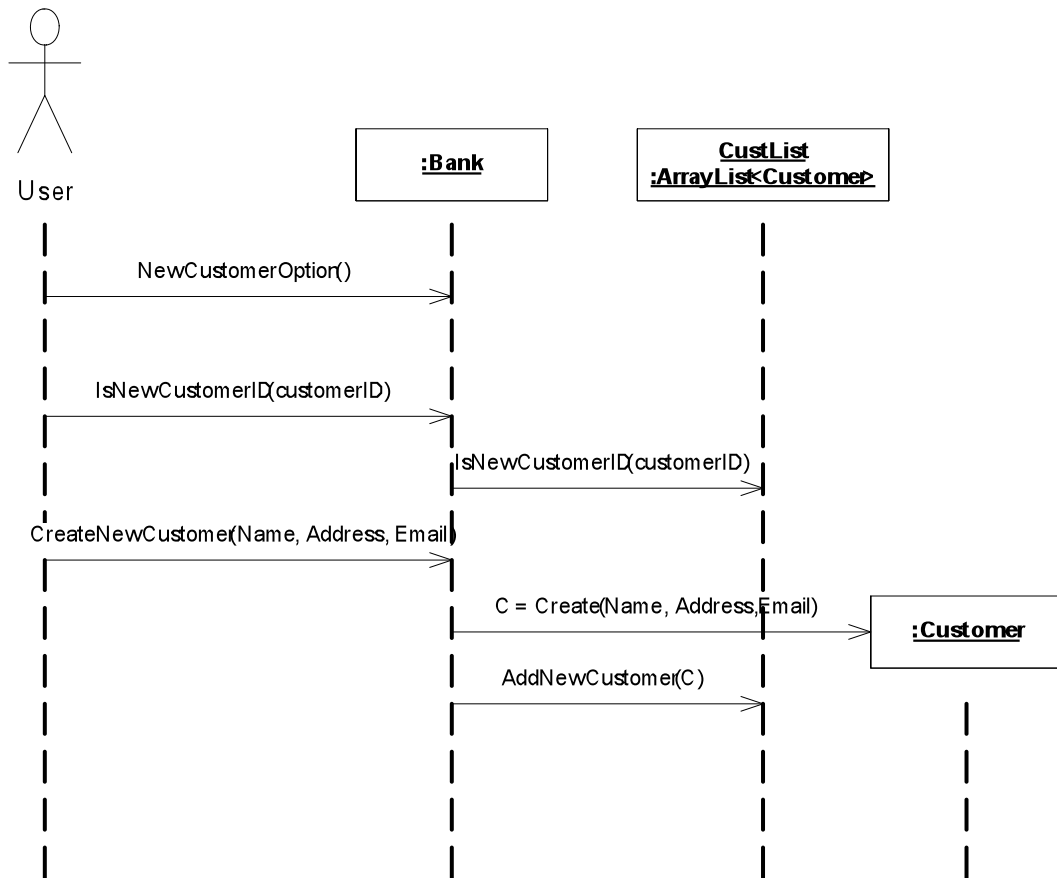
Operation Name	Instances Created	Associations Formed	Attributes Modified
NewCustomerOption()			
IsNewCustomerID(CustomerID)			
CreateNewCustomer(Name, Address, Email)	- Customer	- Customer with Bank	- Customer.active = True - Customer.Name = Name - Customer.Address = Address - Customer.Email = Email

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Sequence Diagram for Base Scenario

Manage Customer : New Customer



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Modify Customer Information

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- Service Rep is authenticated.

#### **Post Conditions:**

Customer information is modified.

#### **Basic Flow**

1. User selects modify customer option. → ModifyCustomerOption()
2. System requests customer ID. (SSN or passport ID)
3. User enters customer ID. → IsValidCustomerID(CustomerID)
4. System verifies customer ID.
5. System presents customer information.
6. System requests changes in customer information and status of the customer.
7. User enters customer information and customer status. → ModifyCustomer(Name, Address, Email, ActiveStatus)
8. System verifies customer information.
9. System updates customer information.
10. System updates customer status.
11. System presents user options.

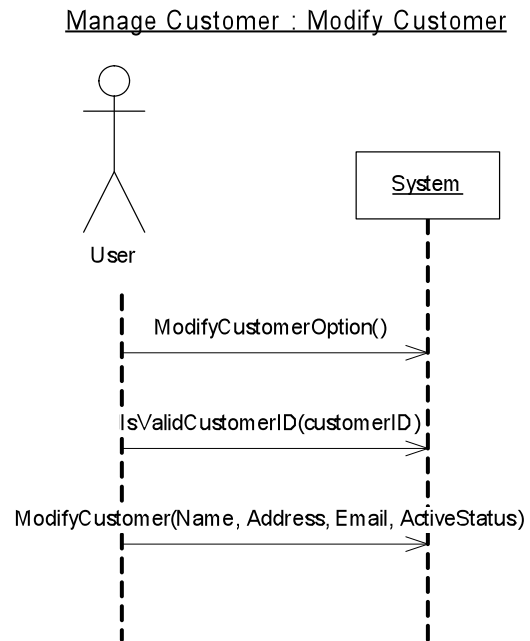
#### **Alternative flow**

- 4a. Invalid customer ID.
  1. System presents error message.
  2. Return to step 2.
- 7a. System detects missing information.
  1. System presents error message.
  2. Return to step 5.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## ***Operation Contract***

### **Contract C122: ModifyCustomerInformation**

<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
ModifyCustomerOption()			
IsValidCustomerID(CustomerID)			
ModifyCustomer(Name, Address, Email, ActiveStatus)			<ul style="list-style-type: none"> <li>- Customer.Name = Name</li> <li>- Customer.Address = Address</li> <li>- Customer.Email = Email</li> <li>- Customer.Active = ActiveStatus</li> </ul>

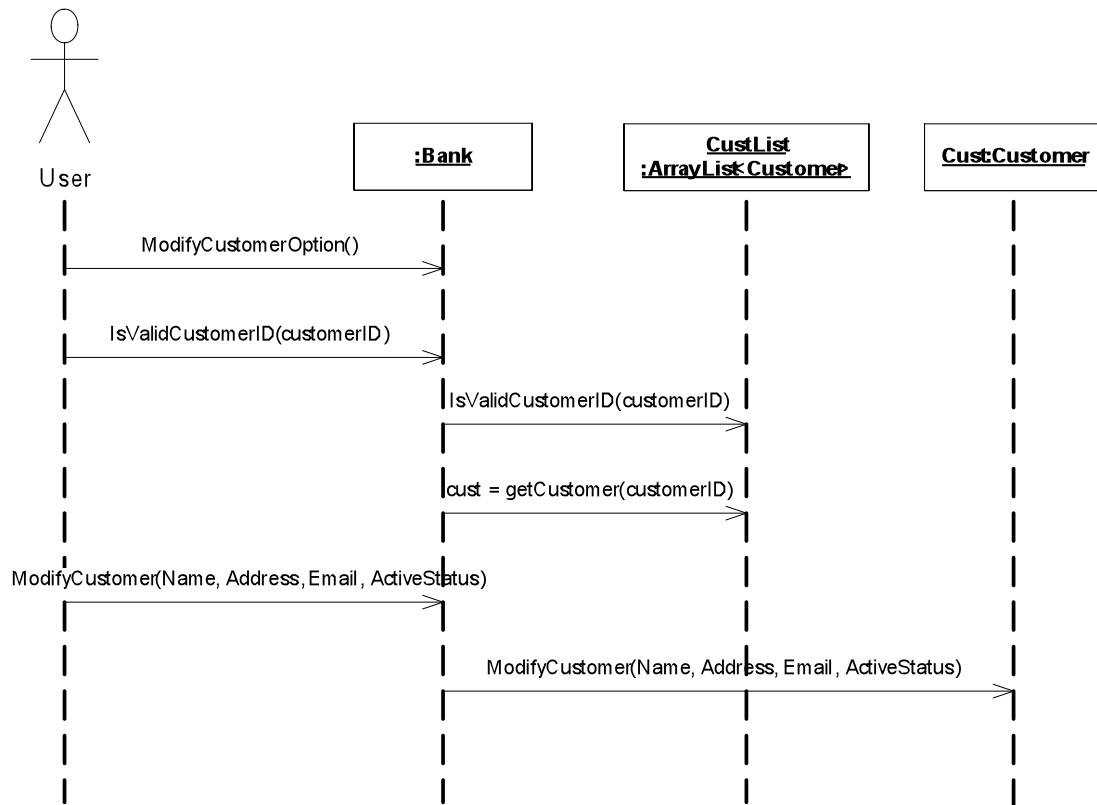
Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario

Manage Customer : Modify Customer



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Deactivate customer

This is already implemented on modify customer use case.

## **Use Case 13: Manage account information** Manage account information (Create, modify deactivate account info) – Customer, Teller, Service Rep.

### **Create New Account**

#### ***Use Case Description with Major Alternative flows***

##### **Preconditions:**

Service Rep. is authenticated.

##### **Post conditions:**

Account is created.

##### **Basic Flow**

1. User selects create new account option. → NewAccountOption()
2. System requests customer ID.
3. User enters customer ID. → IsValidCustomerID(CustomerID)
4. System verifies customer ID.
5. System requests account product.
6. User enters account product ID. → IsValidAccountProductID(AccountProductID)
7. System verifies account product ID.
8. System requests interest strategy ID.
9. User enters interest strategy ID. → IsValidInterestStrategyID(InterestStrategyID)
10. System verifies interest strategy ID.
11. System requests account information.  
User enters account information. → CreateAccount(Name, StartDate)
12. System verifies account information.
13. System creates new account.
14. System presents account number.
15. System presents user options.

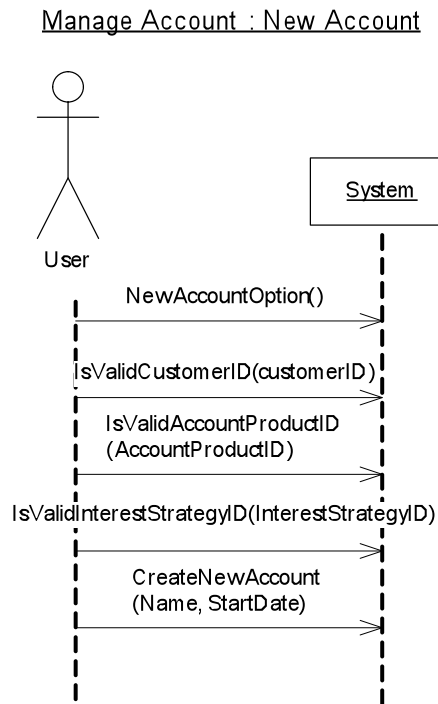
##### **Alternative Flow**

- a. The use case is called from Open Account use case (CustomerID is supplied)
  1. System starts step 5. to 15.
  2. Resume open account process.
- 4a. Customer ID invalid (not exist or not in active status).  
System presents invalid customer ID.  
Return to step 2.
- 7a. Account Product ID invalid (not exist or not in active status).
  1. System presents invalid Product.
  2. Return to step 5.
- 10a. InterestStrategy ID is invalid (not exist or not in active status).
  1. System presents invalid interest strategy ID.
  2. Return to step 8.
- 13a. System detects missing information.  
System presents error message.  
Return to step 11.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## ***Operation Contract***

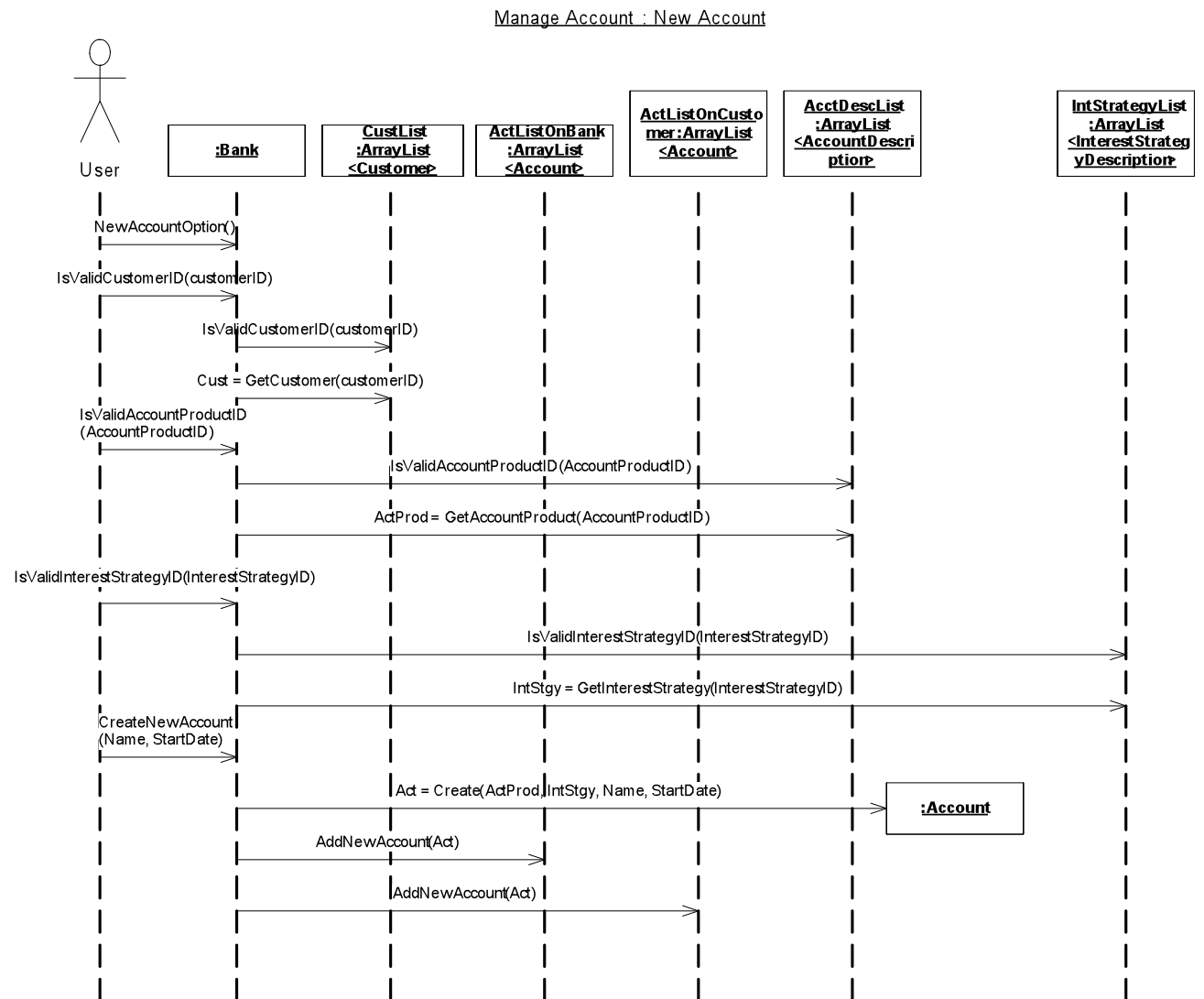
### **Contract C131: CreateNewAccount**

<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
NewAccountOption()			
IsValidCustomerID(CustomerID)			
IsValidAccountProductID(AccountProductID)			
IsValidInterestStrategyID(InterestStrategyID)			
CreateNewAccount(Name, StartDate)	- Account	- Account with Bank - Account with AccountDescription - Account with Customer	- AccountDescription.Type = Type

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Sequence Diagram for Base Scenario



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Modify Account Information

### *Use Case Description with Major Alternative flows*

**Pre conditions:**

- Service Rep is authenticated.

**Post conditions:**

- Account information is modified.

**Basic Flow**

1. User selects modify account option. → ModifyAccountOption()
2. System requests account number.
3. User enters account number. → IsValidAccountNumber(AccountNumber)
4. System verifies account number.
5. System requests change in account product.
6. User enters account product ID. → IsValidAccountProductID(AccountProductID)
7. System verifies account product ID.
8. System requests interest strategy ID.
9. User enters interest strategy ID. → IsValidInterestStrategyID(InterestStrategyID)
10. System verifies interest strategy ID.
11. System requests changes in account.
12. User enters changes in account information. → ModifyAccount(Name, ActiveStatus)
13. System verifies changes in account information.
14. System updates account information.
15. System updates account active status.
16. System presents user options.

**Alternative Flow**

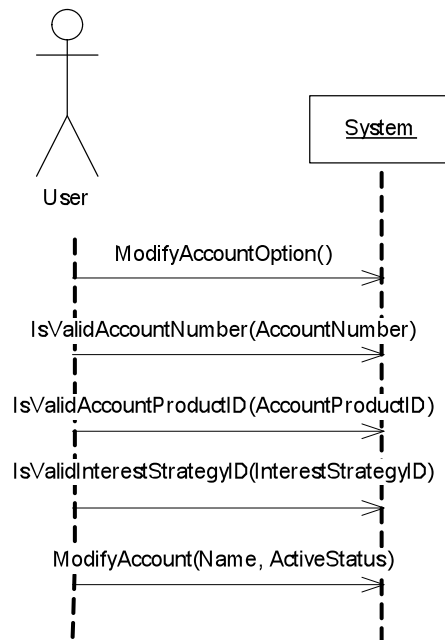
- a. The use case is called from **Close Account** Use Case (AccountNumber is supplied, purpose is to deactivate account)
  1. System starts step 12. to 15.
  2. Resume close account process.
- 4a. Account number is invalid (not exist or not in active status).
  1. System presents invalid account number
  2. Return to step 2.
- 7a. Account Product ID is invalid (not exist or not in active status).
  1. System presents invalid account product ID.
  2. Return to step 5.
- 10a. InterestStrategy ID is invalid (not exist or not in active status).
  1. System presents invalid interest strategy ID.
  2. Return to step 8.
- 13a. System detects missing information.
  1. System presents error message.
  2. Return to step 14.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram

Manage Account : Modify Account



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

### Contract C132: ModifyAccountInformation

Operation Name	Instances Created	Associations Formed	Attributes Modified
ModifyAccountOption()			
IsValidAccountNumber(AccountNumber)			
IsValidAccountProductID(AccountProductID)			
IsValidInterestStrategyID(InterestStrategyID)			
ModifyAccount(Name, ActiveStatus)			<ul style="list-style-type: none"> <li>- Account.AccountProductID = AccountProductID</li> <li>- Account.Name = Name</li> <li>- Account.Active = ActiveStatus</li> </ul>

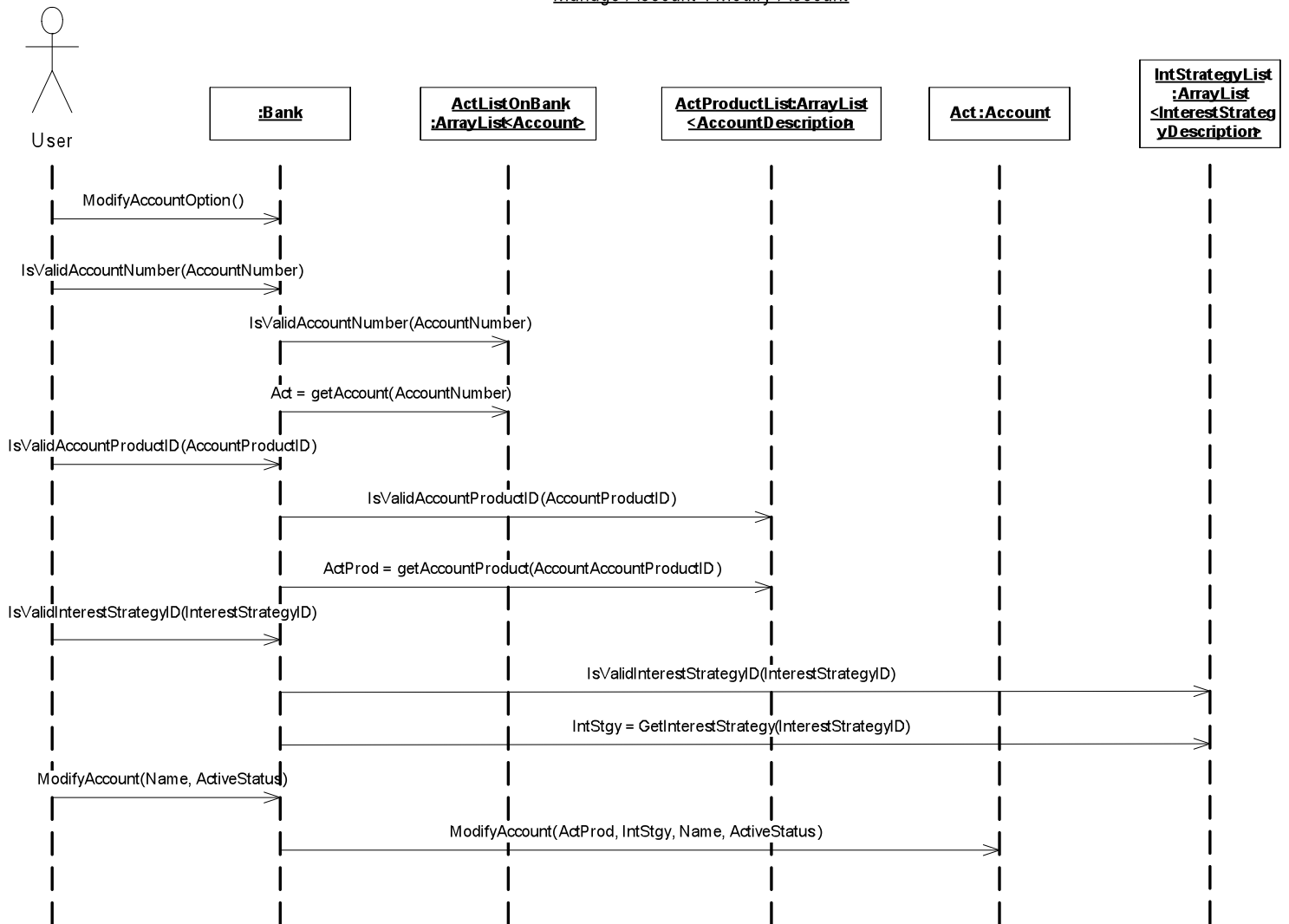
Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario

Manage Account : Modify Account



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Deactivate account

This is already implemented on modify account information use case.

## Use Case 14: View Account Portfolio

### *Use Case Description with Major Alternative flows*

**Pre Conditions:**

- User is authenticated

**Post Conditions:**

- None

**Basic Flow**

1. User selects view account portfolio option. →ViewAccountPortfolioOption()
2. System requests account product ID.
3. User enters account product ID. →isValidAccountProductID(accountProductID)
4. System verifies account product ID.
5. System gets a list of active customers.
6. For each customer, system gets all his active accounts based on account type.
7. System gets each account's balance.
8. System calculates aggregate results.
9. System presents aggregate results.
10. System present user options.

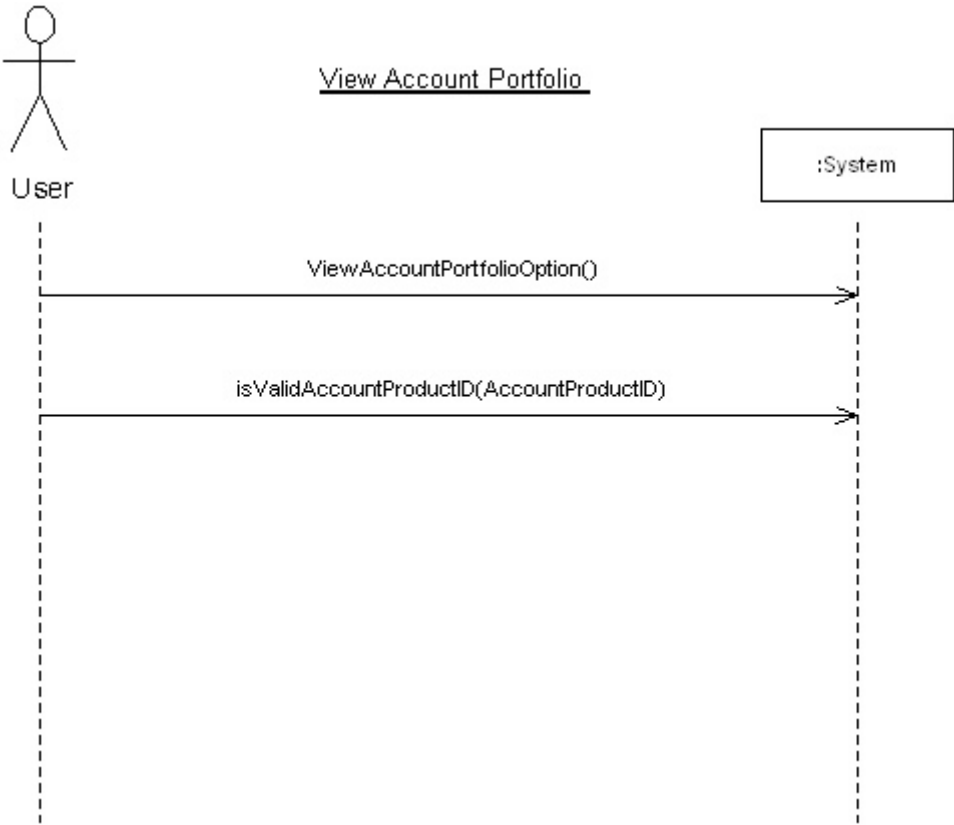
**Alternative Flow**

- 3a. User enters incorrect account type.
  1. System presents error message.
  2. Return to step 2.
- 5a. There is no any active customer.
  1. System presents error message.
  2. Return to step 2.
- 6a. Customer has no active account.
  1. System presents error message.
  2. Return to step 2.

Lead Responsibility by Kittipong Techapanichgul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

**System Sequence Diagram**



Lead Responsibility by Kittipong Techapanichgul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

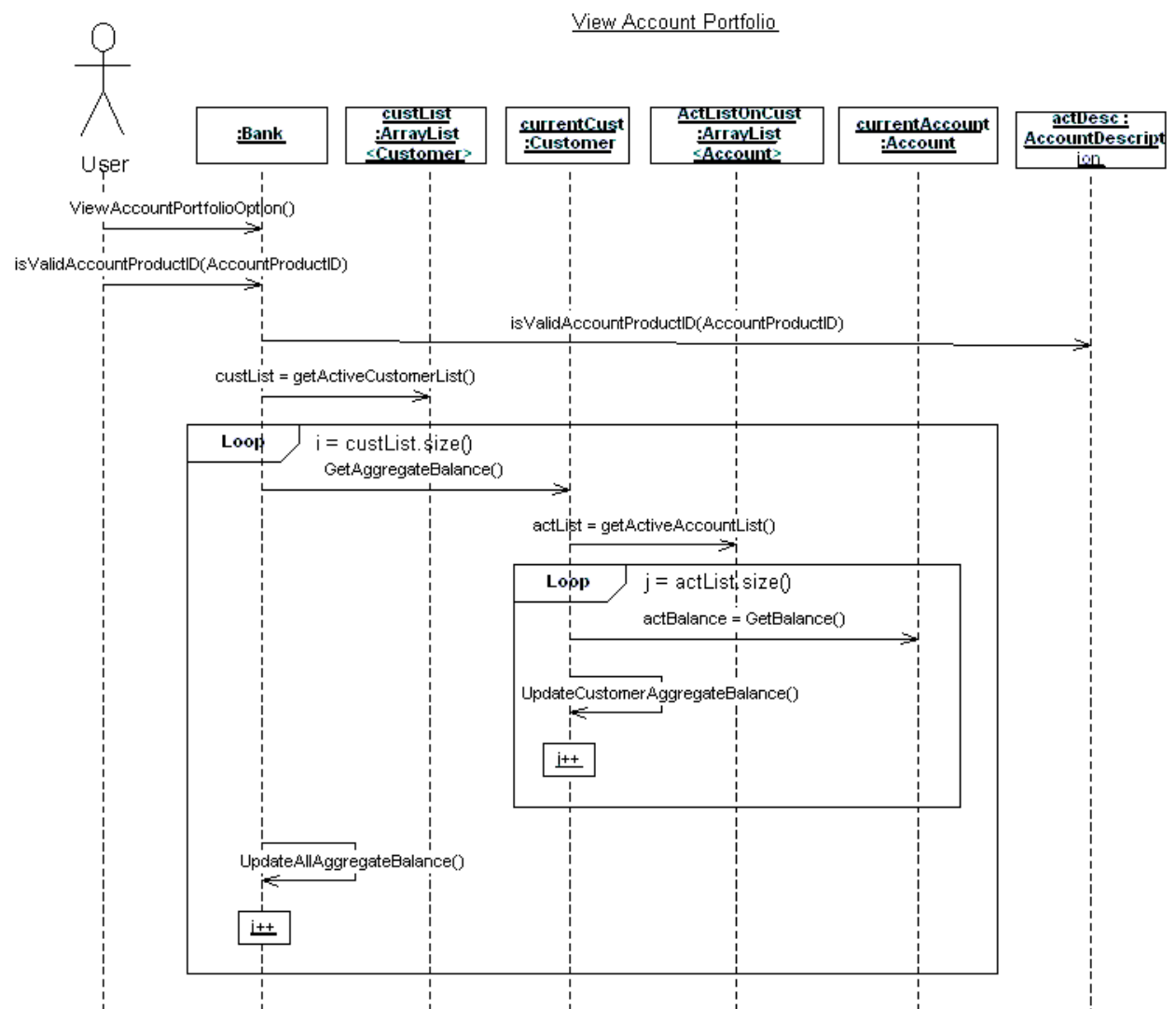
# **Operation Contract**

## **Contract C14: ViewAccountPortfolio**

Operation Name	Instances Created	Associations Formed	Attributes Modified
ViewAccountPortfolioOption()	-	-	-
isValidAccountProductID(AccountProductID)	-	-	-

Lead Responsibility by Kittipong Techapanichgul  
 Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Warat Chesdavanijkul

## Sequence Diagram for Base Scenario



## Use Case 15: Process Account Statement

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- Time reach the expiration of a monthly time period.

#### **Post Conditions:**

Balance, interest and fee of each account is calculated and updated.

#### **Basic Flow**

1. System requests start of monthly maintenance.
2. User selects monthly maintenance option. → ProcessACStatementOption(Month, Year)
3. For every account, system calculates fee.
4. For every account, system calculates interest.
5. For every account, system generates transaction for monthly maintenance.
6. For every account, system updates balance.

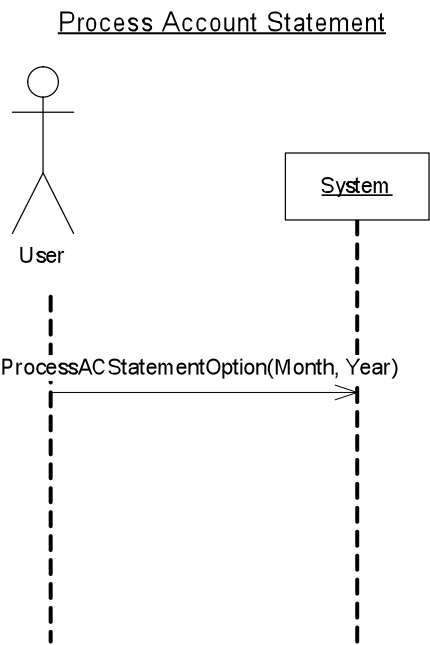
#### **Alternative Flow**

None

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

**System Sequence Diagram**



Lead Responsibility by Warat Chesdavanijkul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## ***Operation Contract***

### **Contract C15: Process Account Statement**

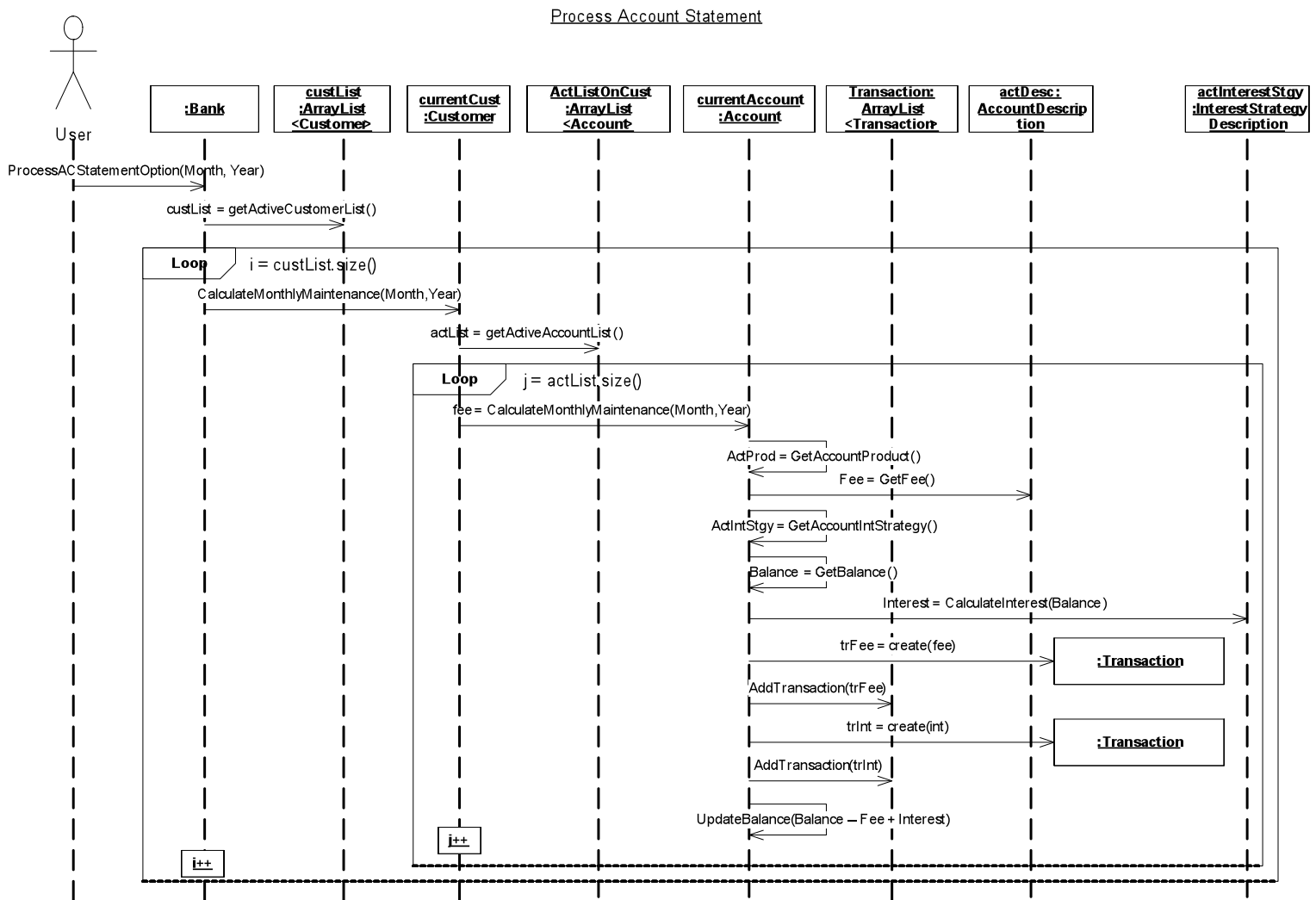
<b>Operation Name</b>	<b>Instances Created</b>	<b>Associations Formed</b>	<b>Attributes Modified</b>
ProcessACStatementOption(Month, Year )	- Transaction for every account.	- Each transaction with each Account	- Account.balance = CalculateNewBalance

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul



## Sequence Diagram for Base Scenario



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

**Use Case 16: Manage Account Product** (Manage new or existing bank products, e.g., Senior Citizens Checking Account. Sample characteristics of a product include a name, monthly fee, per check charge, periodic interest rate, etc)

## Create New Account Product

### *Use Case Description with Major Alternative flows*

**Pre Conditions:**

Service Rep. is authenticated.

**Post Conditions:**

Account product is created.

**Basic Flow**

1. User selects create new account product option. → NewAccountProductOption()
2. System requests account type.
3. User enters the new account product ID. → IsNewAccountProductID(AccountTypeID)
4. System verifies product ID.
5. System requests interest strategy ID.
6. User enters interest strategy ID. → IsNewInterestStrategyID(InterestStrategyID)
7. System verifies interest strategy ID.
8. System requests account product information.  
User enters account product information. → CreateNewAccountProduct(Name, MonthlyFee, PerCheckCharge, InterestRate)
9. System verifies account product information.
10. System creates new account product.
11. System presents user options.

**Alternative Flow**

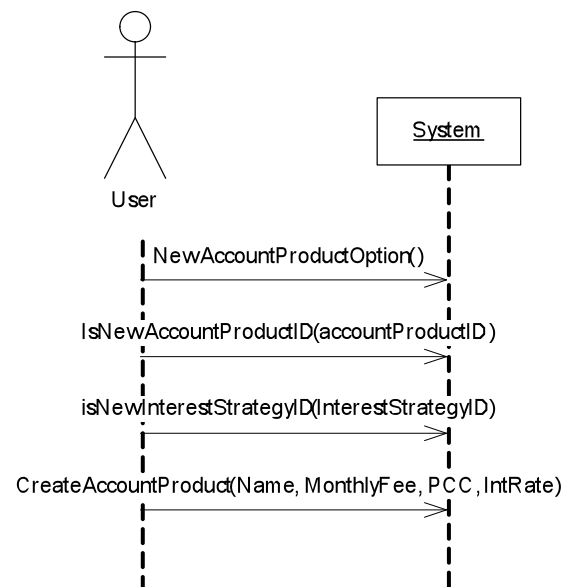
- 4a. System detects existing account product ID.  
System presents the account product already exist on the system.  
Return to step 2.
- 7a. InterestStrategy ID is invalid (not exist or not in active status).  
System presents invalid interest strategy ID.  
Return to step 5.
- 10a. System detects missing information.
  1. System presents error message.
  2. Return to step 8.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram

### Manage Account Product : New Account Type



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

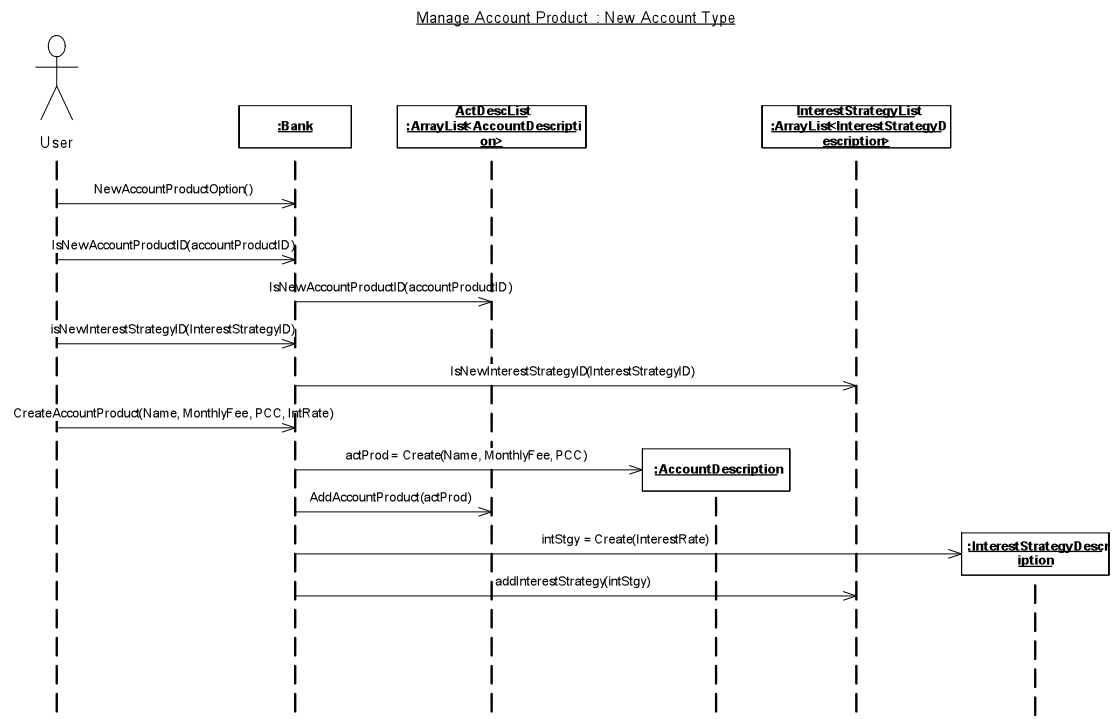
### Contract C161: Create New Account Product

Operation Name	Instances Created	Associations Formed	Attributes Modified
NewAccountProductOption()			
IsNewAccountProductID(AccountTypeName)			
IsNewInterestStrategyID(InterestStrategyID)			
CreateAccountProduct(Name, MonthlyFee, PerCheckCharge, InterestRate)	<ul style="list-style-type: none"> <li>- AccountDescription</li> <li>- InterestStrategyDescription</li> </ul>	<ul style="list-style-type: none"> <li>- AccountDescription with Bank</li> <li>- InterestStrategy with Bank</li> </ul>	<ul style="list-style-type: none"> <li>- AccountDescription.Name = Name</li> <li>- AccountDescription.MonthlyFee = MonthlyFee</li> <li>- AccountDescription.PerCheckCharge = PerCheckCharge</li> <li>- AccountDescription.active=true</li> <li>- InterestDescription.InterestRate = InterestRate</li> </ul>

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

Sequence Diagram for Base Scenario



Lead Responsibility by Warat Chesdavanijkul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Modify Account Product

### *Use Case Description with Major Alternative flows*

#### **Pre Conditions:**

- Service Rep is authenticated.

#### **Post Conditions:**

- Account product is modified.

#### **Basic Flow**

1. User selects modify account product option. → ModifyAccountProductOption()
2. System requests account product ID.
3. User enters account product ID. → IsValidAccountProductID(AccountProductID)
4. System verifies account product ID.
5. System requests interest strategy ID.
6. User enters interest strategy ID. → IsValidInterestStrategyID(InterestStrategyID)
7. System verifies interest strategy ID.
8. System requests changes in account product information and the status of the account.
9. User enters account product information and the account status. →  
ModifyAccountProduct(MonthlyFee, PerCheckCharge, InterestRate, ActiveStatus)
10. System verifies account product information.
11. System updates account product information.
12. System updates account product status.
13. System presents user options.

#### **Alternative Flow**

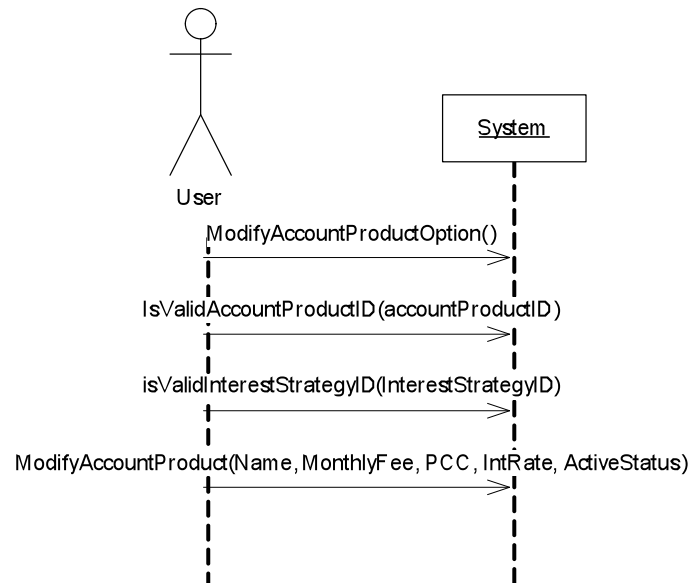
- 4a. Invalid account product ID.
  3. System presents invalid account product ID.
  4. Return to step 2.
- 7a. InterestStrategy ID is invalid (not exist or not in active status).
15. System presents invalid interest strategy ID.
16. Return to step 5.
  - 10a. System detects missing information.
    3. System presents error message.
    4. Return to step 8.

Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## System Sequence Diagram

### Manage Account Product : Modify Account Type



Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

## Operation Contract

### Contract C162: Modify Account Product

Operation Name	Instances Created	Associations Formed	Attributes Modified
ModifyAccountProduct()			
IsValidAccountProductID(AccountProductID)			
IsValidInterestStrategyID(InterestStrategyID)			
ModifyAccountProduct(Name, MonthlyFee, PerCheckCharge, InterestRate, ActiveStatus)			<ul style="list-style-type: none"> <li>- Account.MonthlyFee = MonthlyFee</li> <li>- Account.PerCheckCharge = PerCheckCharge</li> <li>- Account.InterestRate = InterestRate</li> <li>- Account.Active = ActiveStatus</li> </ul>

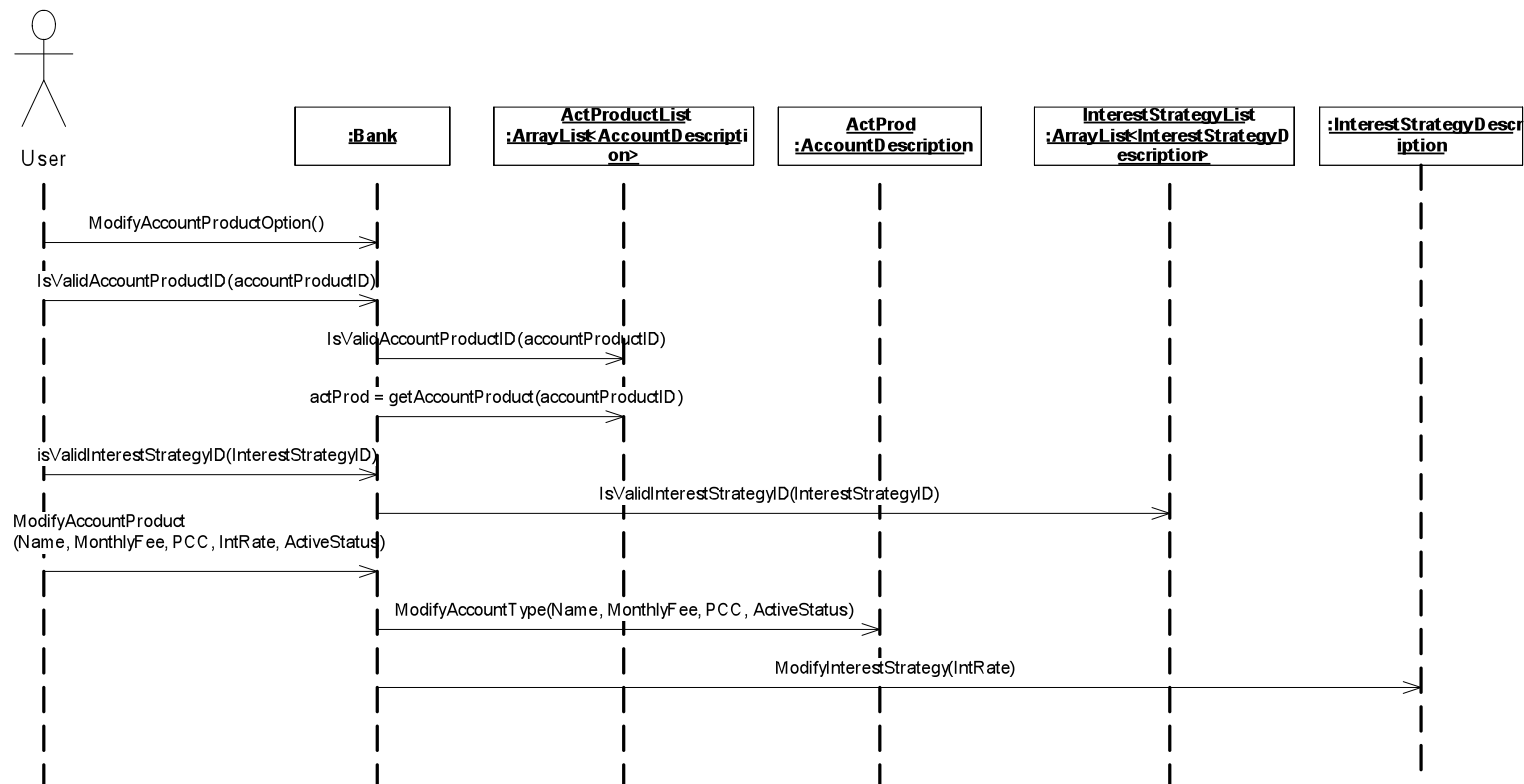
Lead Responsibility by Warat Chesdavanijkul

Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul



Sequence Diagram for Base Scenario

Manage Account Product : Modify Account Type



Lead Responsibility by Warat Chesdavanijkul  
Substantive assistance by Jaruwan Laptrakool, Derek Pawlikowsky and Kittipong Techapanichgul

Deactivate Account Product

This is already implemented on modify account product use case.