TATA CONSULTANCY SERVICES

Experience certainty. IT Services
Business Solutions
Outsourcing

ABC Banking Case Study – OO Modeling and Implementation

Reference Case Study, January 2009

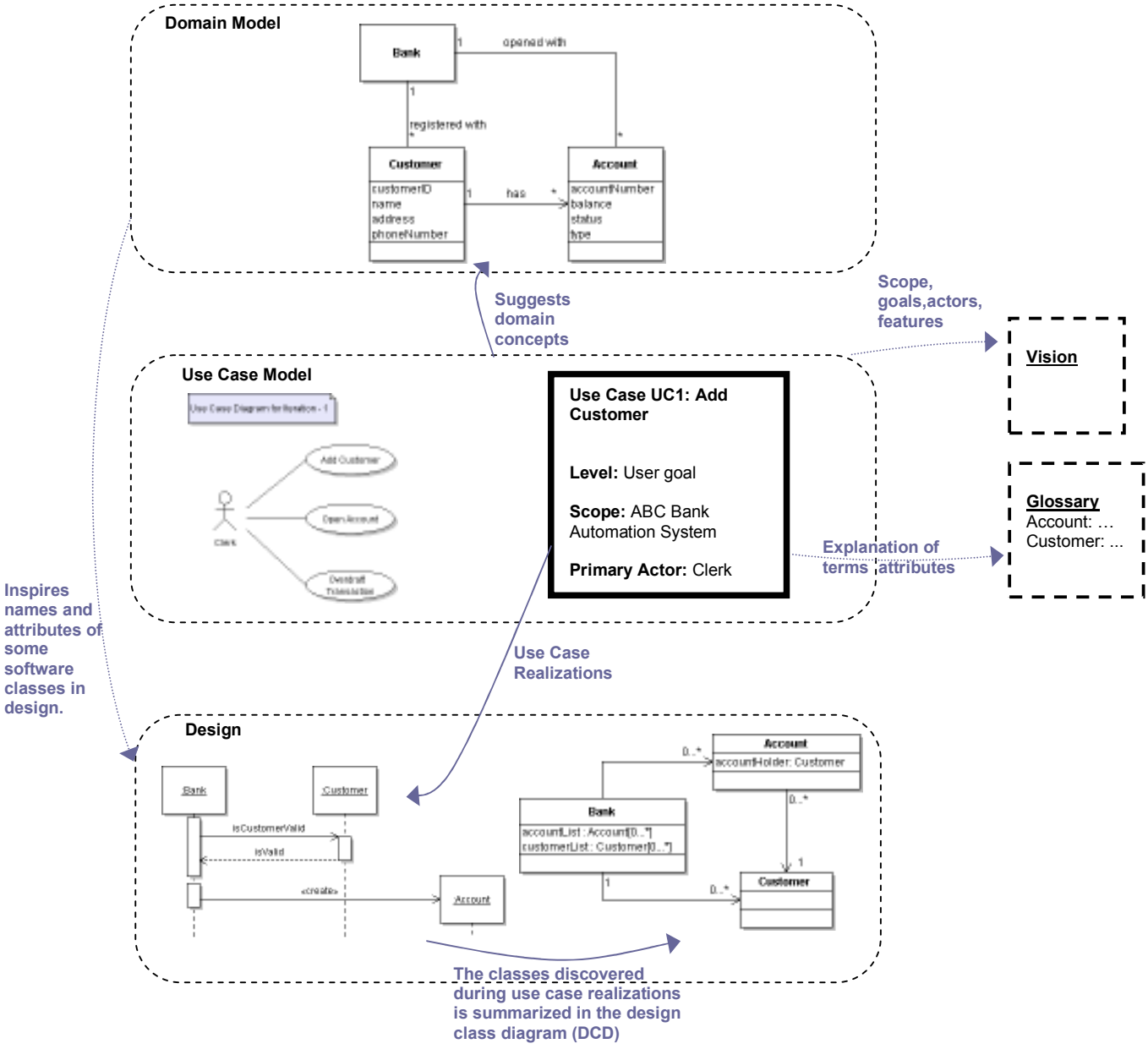
TCS - ILP

TCS Internal

Table of Contents

1. Relationship between UML artifacts
2. Vision
3. Iteration 1
    a. Use Case Model
        1. Use Case Diagram
        2. Use Cases
    b. Domain Model
    c. Design Model
        1. Use Case realizations(Sequence Diagram)
        2. Design Class Diagram
4. Appendix A
5. Appendix B
6. Appendix C

1. Relationship between UML artifacts



2. Vision: ABC Bank Automation System

Problem Statement:

ABC bank plans to automate its current account related activities. A customer can open a current account. There are two types of current accounts:-Cash Credit account and Regular Current account. Cash credit account holders can avail overdraft facility. They can withdraw 10% of their balance amount as overdraft. They cannot withdraw or avail an overdraft if existing overdraft is not paid back. Only privileged customers can open a Cash Credit account. For every 100 transactions made on a regular current account a fee of Rs.100/- is deducted from the account balance. A customer should first provide his details to register in the bank before opening an account. If required he can update his details later. Clerk can view details of registered customers. The customer may have one or more accounts in the bank. He can deposit or withdraw money from his account. He can also close his account. The clerk generates a statement of the transactions performed on an account to verify the account transactions.

Stakeholder Descriptions:

Stakeholder Summary:

- **ABC Bank:**
  The bank wants to automate its key areas of current account management with the help of latest technology. This includes automating customer registration, account opening and transaction handling processes. The bank wants to ensure fast and quality service to its customers and thereby wants to increase customer satisfaction.

- **Clerk:**
  Clerk wants to have a fast and reliable system that helps in carrying out daily transactions and related activities, in an efficient way.

- **Customer:**
  A customer with the bank expects reliable and fast service.

User Summary:

Clerk:
The user of ABC Bank Automation System will be a clerk, employed by the bank. He/she will have access to ABC Bank Automation System and will be trained to use the functionalities of the system.

Key high level goals and problems of stakeholders:
ABC Bank:
ABC Bank is a leading bank in the country. Their existing system for managing current accounts is manual. The manual process is time consuming and there are chances for errors to happen. The bank wants to have a reliable and efficient system for managing its current accounts.

Clerk:
The existing system of managing current accounts in ABC bank is manual. This is done by clerks employed by the bank. Manual management of current accounts is time consuming and gives a chance for errors to happen. Clerks employed with the bank want to have a reliable solution to manage the current account related activities.

Customer:
A customer with ABC bank expects fast and reliable service.

User level goals
Clerk:
A clerk at ABC bank, who handles current account related activities, wants a fast and reliable solution for the following operations:

- Adding a new customer
- Updating existing customer details
- Viewing existing customer details
- Opening a current account
- Handling transactions such as withdraw, deposit and overdraft for a current account
- Closing a current account

Product overview
The ABC Bank Automation System will be installed at ABC Bank offices. The system will be operated by clerks employed by the bank. It will provide services to make the current account management fast and reliable.

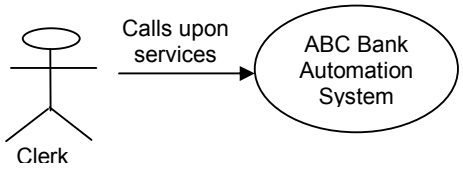


Figure 1: ABC Bank Automation System context diagram

Assumptions and dependencies

1) Customer can exist in the system even without an account.
2) The decision of allotting cash credit account to a customer is at the disposal of bank management.
3) Date is not handled.
4) Closing an account will not delete the account details from the system; instead the status of the account is set to “inactive”.
5) Joint Accounts are not managed.
6) There are three types of transactions possible for cash credit account holders: deposit, withdrawal and overdraft.
7) Transactions can only be done by cash.

Summary of System Features

- Add customer
- Update existing customer details
- View existing customer details
- Open Regular Current Account
- Open Cash Credit Account
- Handle Deposit Transaction
- Handle Withdraw Transaction
- Handle Overdraft Transaction
- Close Account
- Generate Account Transaction Report

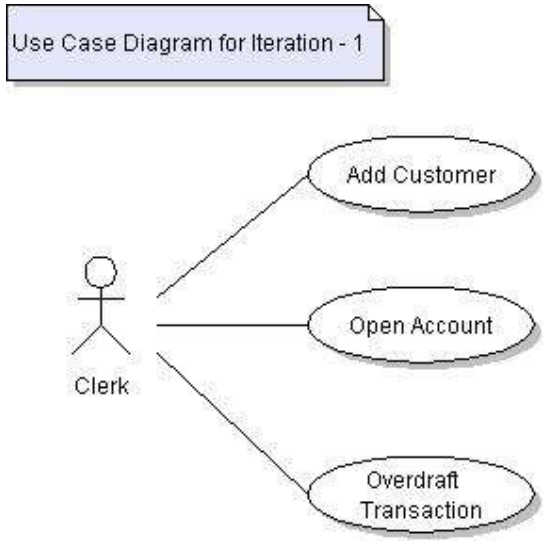
3. Iteration 1

In Iteration 1 only 4 system features (highlighted) are considered.

- **Add customer**
- Update existing customer details
- View existing customer details
- **Open Regular Current Account**
- **Open Cash Credit Account**
- Handle Deposit Transaction
- Handle Withdraw Transaction
- **Handle Overdraft Transaction**
- Close Account
- Generate Account Transaction Report

3a. Use Case Model: Iteration 1

3a.1 Use Case Diagram



3a.2 Use Cases

Use Case UC1: Add Customer

Level: User goal

Scope: ABC Bank Automation System

Primary Actor: Clerk

Stakeholders and interests:

- Customer:
  - Wants to register with ABC bank, for availing various current account related activities provided by the bank.
- Clerk:
  - Wants the system to accurately record the details provided by the customer to register with the bank.
- ABC Bank:
  - Wants to make customer registration process fast and reliable,

Preconditions:

- none

Success Guarantee: Customer details given by the customer is recorded. Customer Id is generated.

Main Success Scenario:

1. Customer arrives at ABC bank and provides details for registration.
2. Clerk enters the details into the system
3. System validates the customer details.
4. System Prompts for confirmation of registration.
5. Clerk Confirms registration.
6. System generates a customer Id for the customer
7. The customer details are saved in the system
8. Clerk provides the customer Id to the customer.

Extensions:

2a. Invalid Customer Details
   1. System displays an error message.
   2. System displays menu.

5a. Customer requests to stop registration.
   1. Clerk cancels registration process.
   2. System displays menu

7a. System fails to save the customer details provided.
   1. System displays an error message.
   2. System displays menu.

Use Case UC4: Open Account

Scope: ABC Bank Automation System

Level: User goal

Primary Actor: Clerk

Stakeholders and Interests:

- Customer:
  - Wants to open a Regular Current account or Cash Credit account with the bank.
  - Wants an account number with which his/her account can be uniquely identified.
- Clerk:
  - Wants to verify if the customer is registered already with the bank.
  - Wants to record account opening details in the system.
  - Wants to provide an account number to the customer on successful account opening.
- ABC Bank:
  - Wants to make account opening process fast and reliable.

Preconditions:

- Customer must be already registered in the system.

Success Guarantee:

- A Regular Current account or Cash Credit account is opened.
- Account number is generated.
- The account details are saved in the system.

Main Success Scenario:

1. A customer arrives at the bank for opening an account.
2. Customer provides his/her customer ID
3. Clerk enters the customerID.
4. System validates the customer ID.
5. Customer provides the type of account he wants to open.
6. Clerk selects the type of account to be opened.
7. Customer provides account opening details to the clerk
8. Clerk enters the details provided by the customer
9. Clerk Confirms Opening of Account
10. System generates the account number and records the account details based on the type of account selected.
11. Clerk provides account number to the customer

Extensions:

2a.Customer is not registered with the bank.

1. Clerks cancels account opening process
2. Clerk requests customer to complete the registration process before opening an account.

3a. When the customer ID provided is invalid:

1. System displays an error message; and gives an option to either initiate account opening process again, or discontinue account opening process

1a.Option to re-enter customer ID is selected.
1. Customer provides valid customer ID
2. Return to Main Success Scenario Step 3

1b. Option to discontinue account opening process is selected
1. Systems discontinues the process
2. System displays main menu

4a. Customer requests to open a cash credit account and he is not eligible for that.
1. Clerk cancels the account opening process.

8a. Customer requests to cancel account opening process.
3. Clerk cancels account opening process.
4. System displays main menu.

9a. System fails to save account details
1. System displays a message that account is not opened.
2. System displays main menu.

Use Case UC7: Overdraft

Scope: ABC bank automation system.

Level: User Goal

Primary Actor: Clerk

Stakeholders and Interests:

Customer:

- Wants to draw overdraft amount from ABC bank with minimum time and effort.
- Wants the account balance to be updated accurately.
- Wants receipt as a proof of the overdraft transaction.

Clerk:

- Wants the account number to be verified as a valid Cash Credit account.
- Wants to ensure that no overdraft payment is pending.
- Wants every overdraft transaction to be entered in the transaction journal so that he can retrieve back the account transaction details while generating the Account Transaction Report for a particular account.

ABC Bank:

Wants to accurately record the overdraft transactions and satisfy customer interests.

Preconditions:

- The customer should be registered with the bank
- The customer must have a valid Cash Credit account.
- There should not be any pending overdraft payment.

Success Guarantee / Post Conditions:

- The customers account balance is updated.
- The account transaction is recorded in the system.
- Receipt is generated / displayed.

Main Success Scenario (or basic flow):

1. Customer arrives at ABC bank and gives his account number and the amount to be drawn as overdraft.
2. Clerk enters the account number.
3. System validates the account number.
4. System validates whether the account is open.
5. System validates account type.
6. System checks for pending overdraft payment.
7. System calculates and displays the maximum overdraft amount available for withdrawal for the given account.
8. Clerk enters the amount to be drawn as overdraft.
9. System validates whether the overdraft amount is within eligible limit.
10. Clerk confirms overdraft transaction.

11. System generates the transaction ID and records the account transaction details.
12. System calculates the account balance after the overdraft.
13. System updates the account balance.
14. Clerk pays the amount to the customer.
15. Receipt is generated/ displayed.

Extensions (or alternate flows)

3a.Invalid Account Number (Not found in the system)
1. System displays an error message and gives an option to either enter valid account number or discontinue the transaction.
1a.Option to reenter account number is selected.
1. Customer provides valid account number
2. Return to Main Success Scenario step 2
1b. Option to discontinue the transaction is selected
1. System discontinues the transaction

4a.Account is closed
1. System displays an error message
2. System discontinues the transaction

5a.Not a cash credit account
1. System displays an error message
2. System discontinues the transaction

6a.Pending overdraft payment
1. System displays a message that the transaction cannot be performed since an overdraft payment is pending.
2. System discontinues the transaction

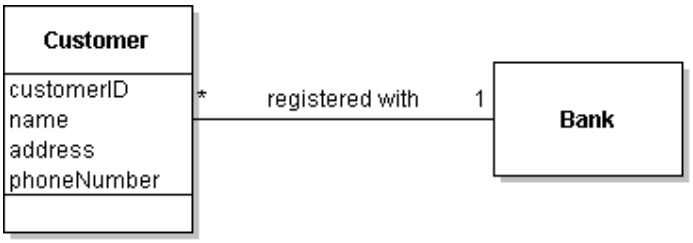
9a.Overdraft amount exceeds eligible limit.
1. System displays an error message
2. System discontinues the transaction

10a. Customer tells clerk to cancel the overdraft transaction
1. Clerk cancels the overdraft transaction.
2. System Displays main menu.

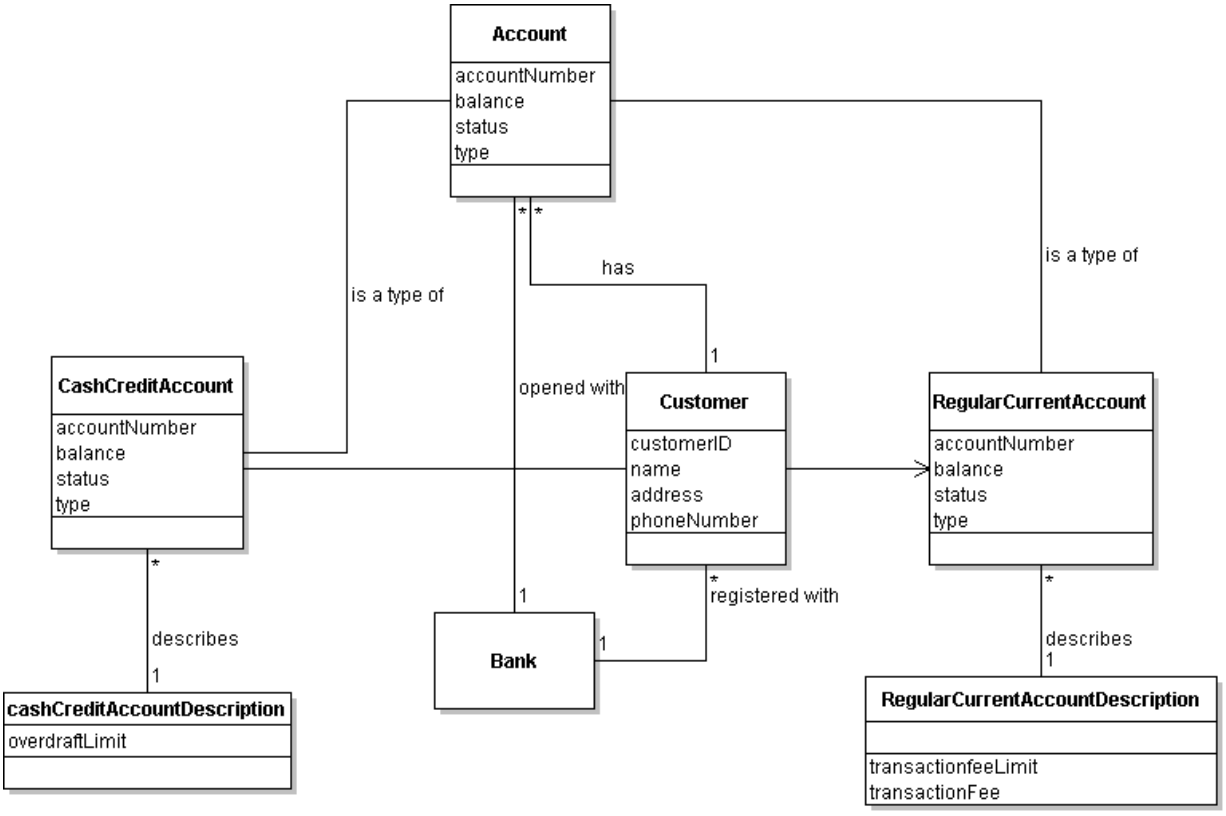
13a.System fails to update the account balance
1. System displays error message
2. System discontinues the transaction

3b Domain Model

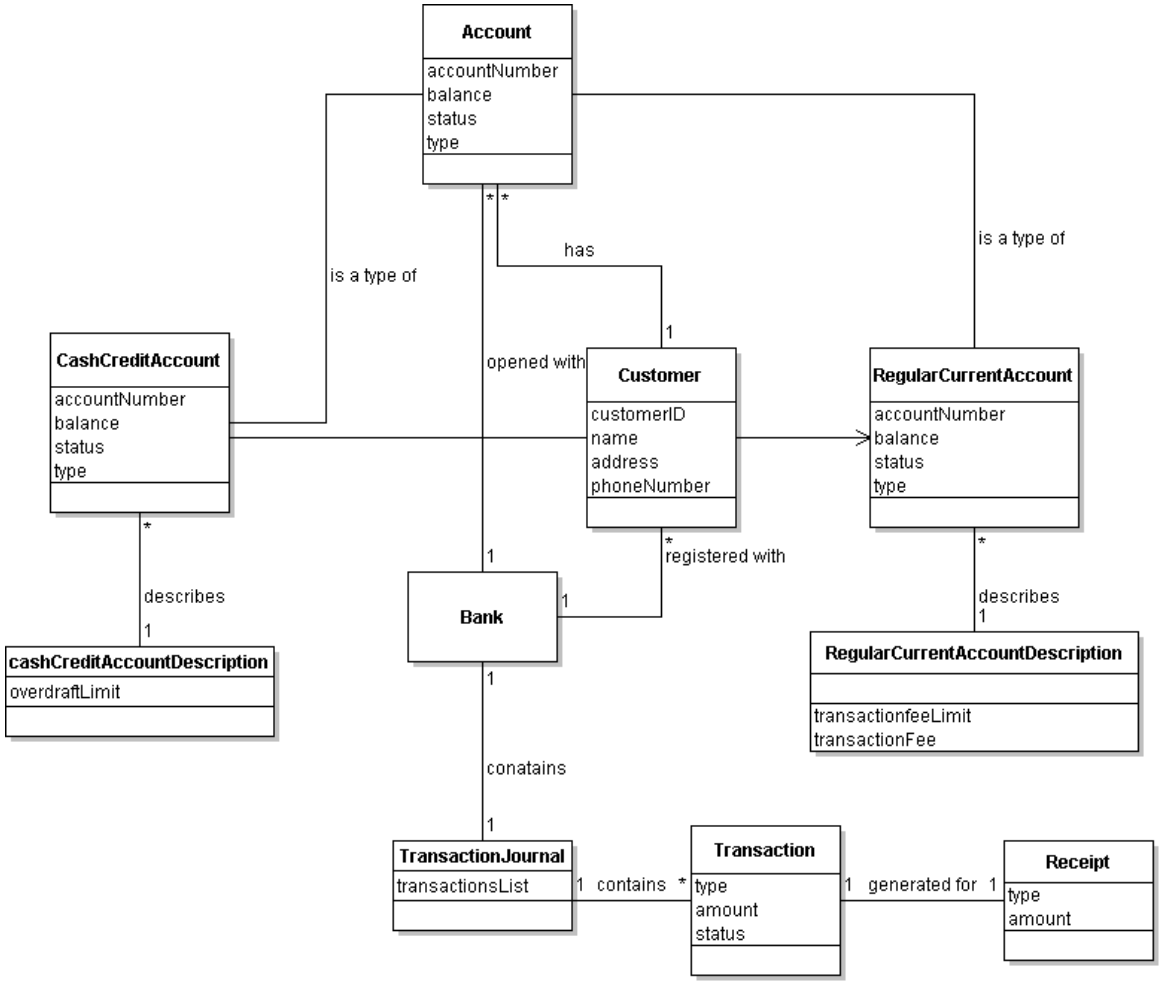
Reference:
1. Appendix A (Conceptual Classes discovered using category list)
2. Appendix B (Classes identified by noun-phrase analysis of use cases (UC1, UC4 & UC7)



Domain Level Class Diagram based on analysis of UC1



Domain Level Class Diagram based on analysis of UC1 & UC4

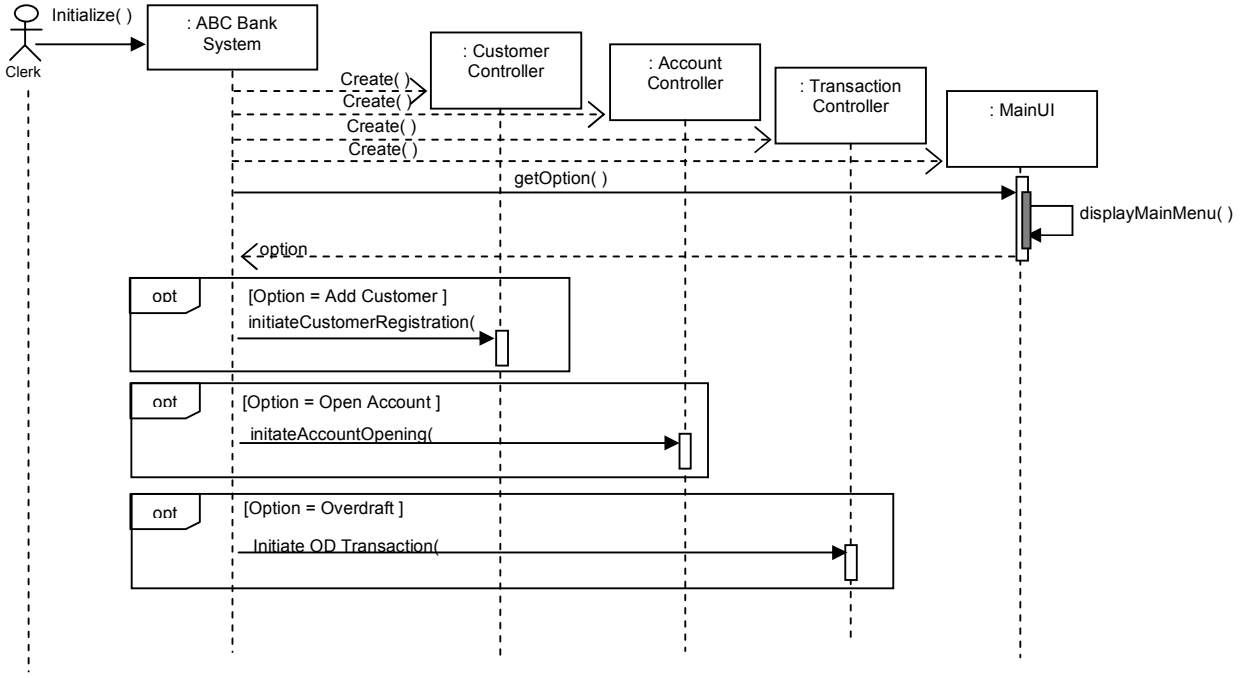


Domain Level Class Diagram based on analysis of UC1,UC4 & UC7

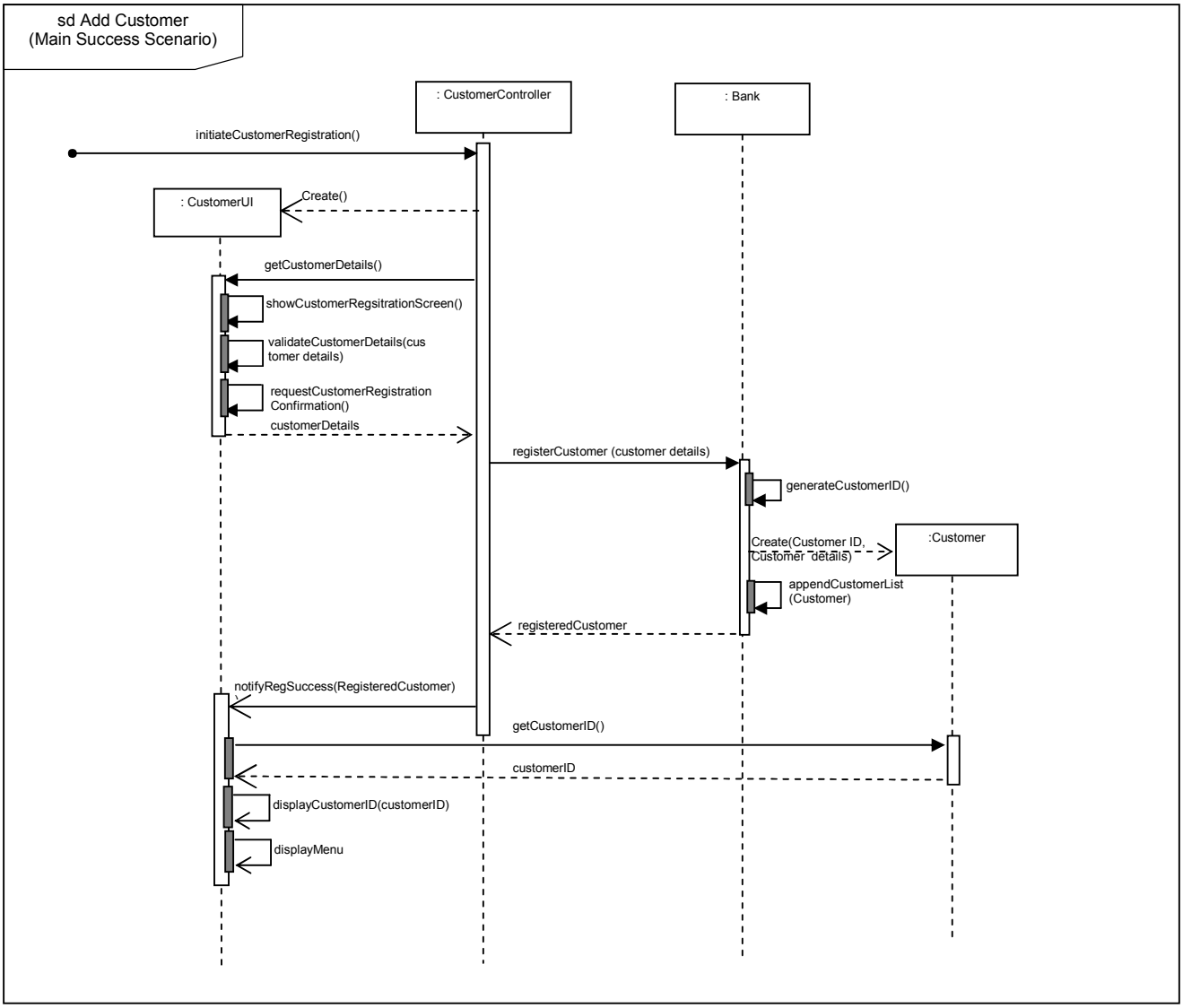
3c. Design Model

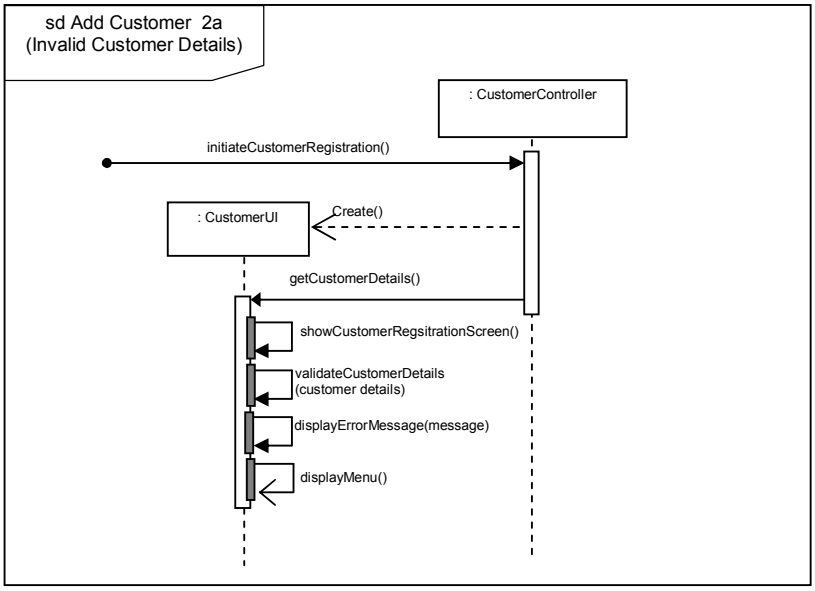
3c.1. Use Case Realization: Sequence Diagrams

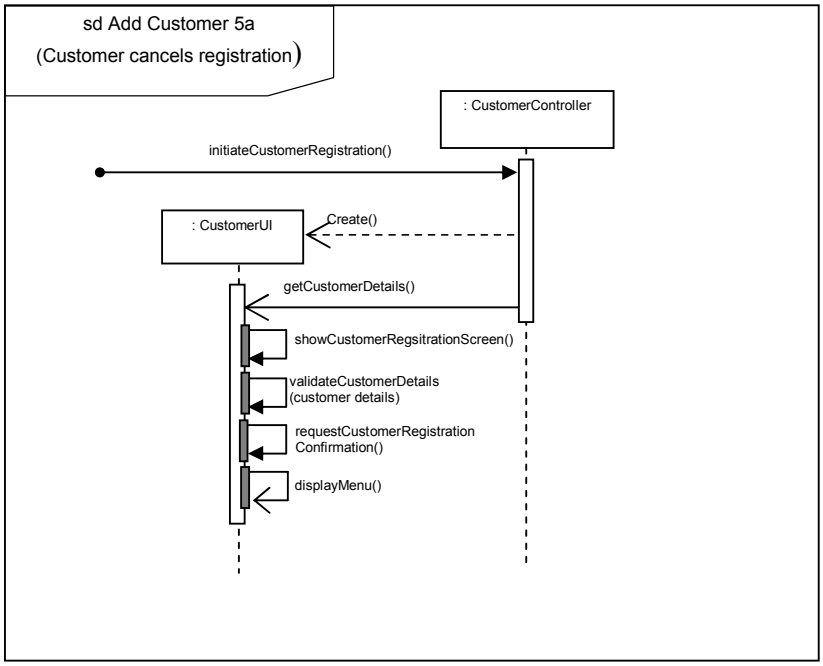
System Initialization Sequence Diagram

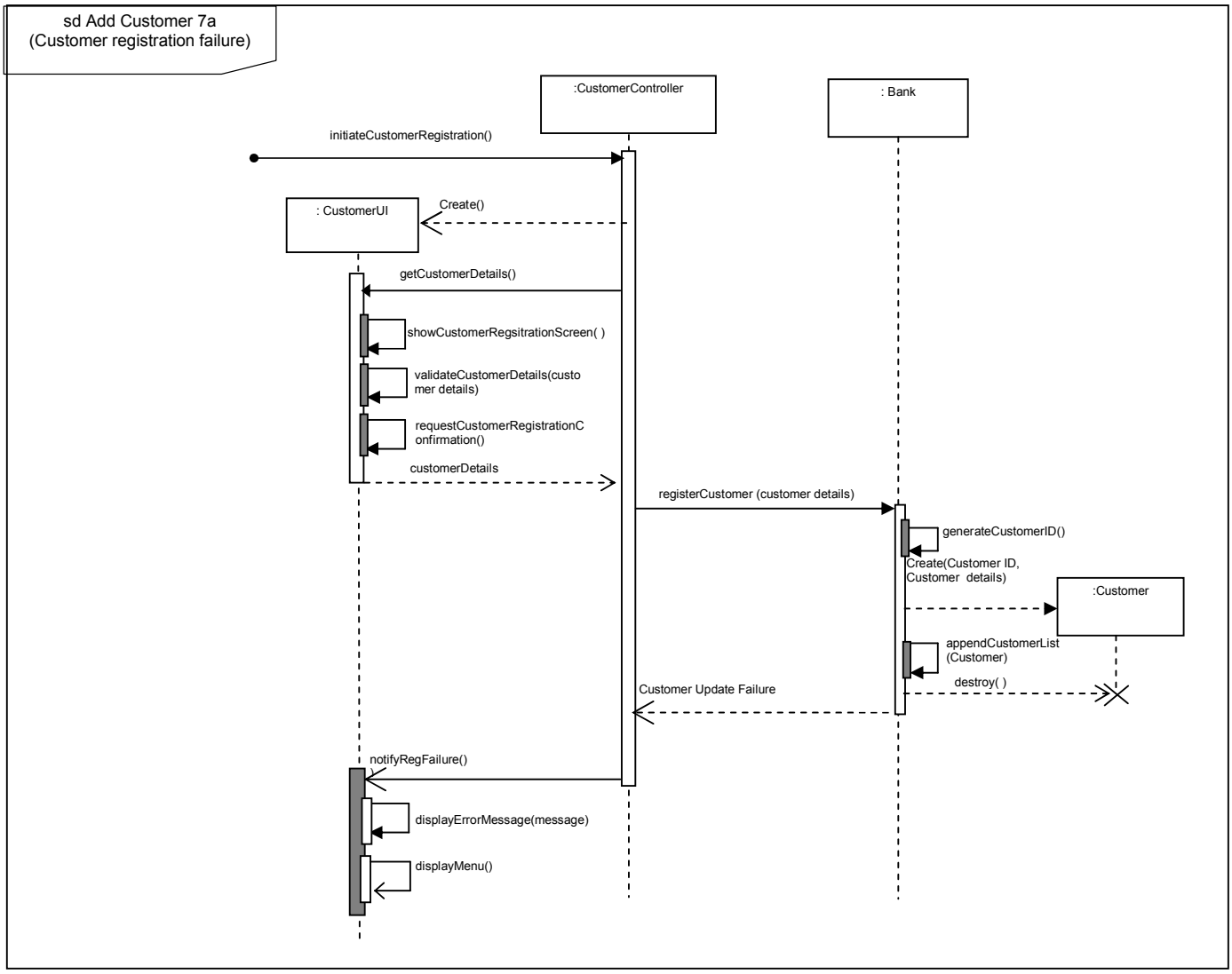


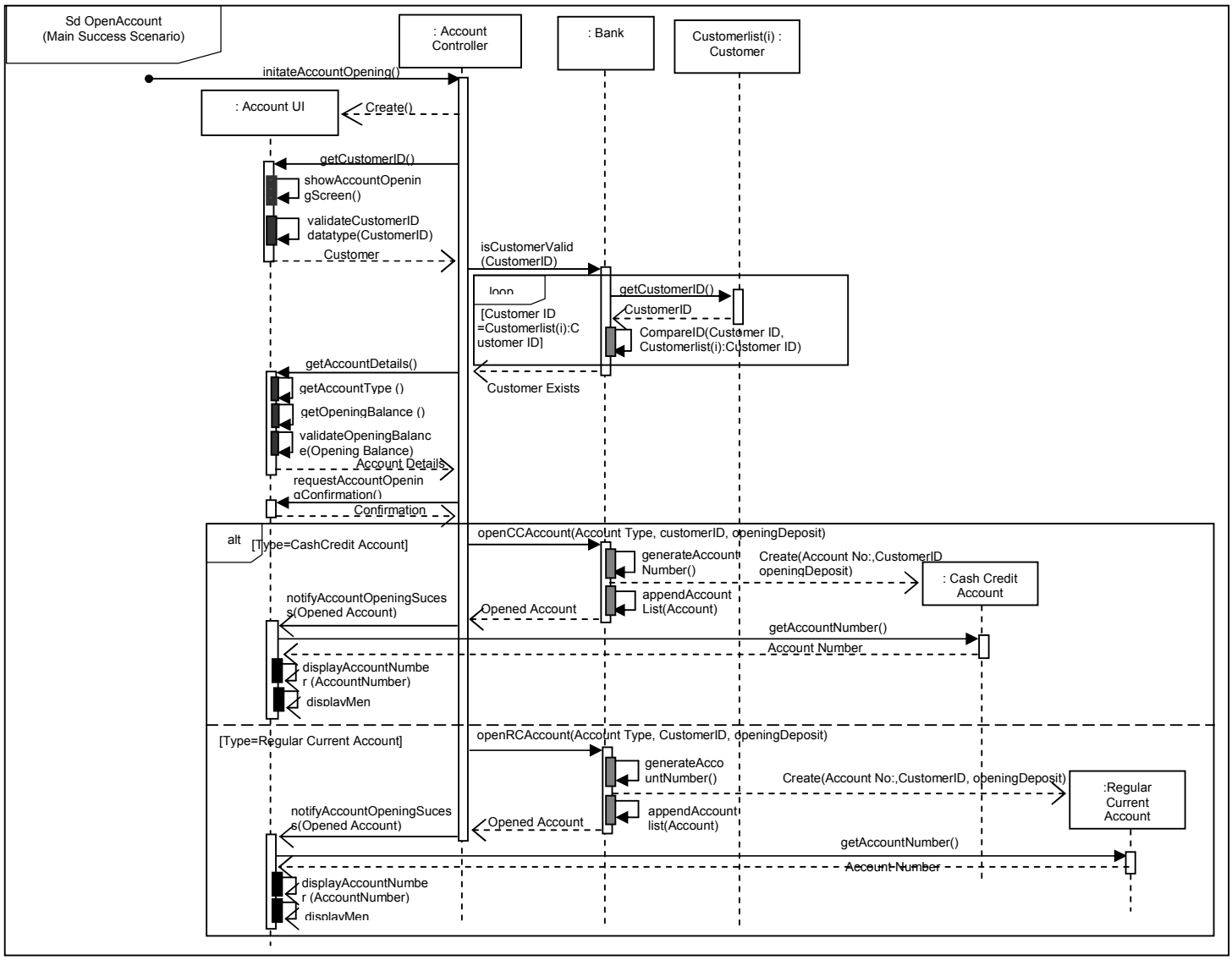
UCR1: Add Customer (Main & Alternate Scenarios)

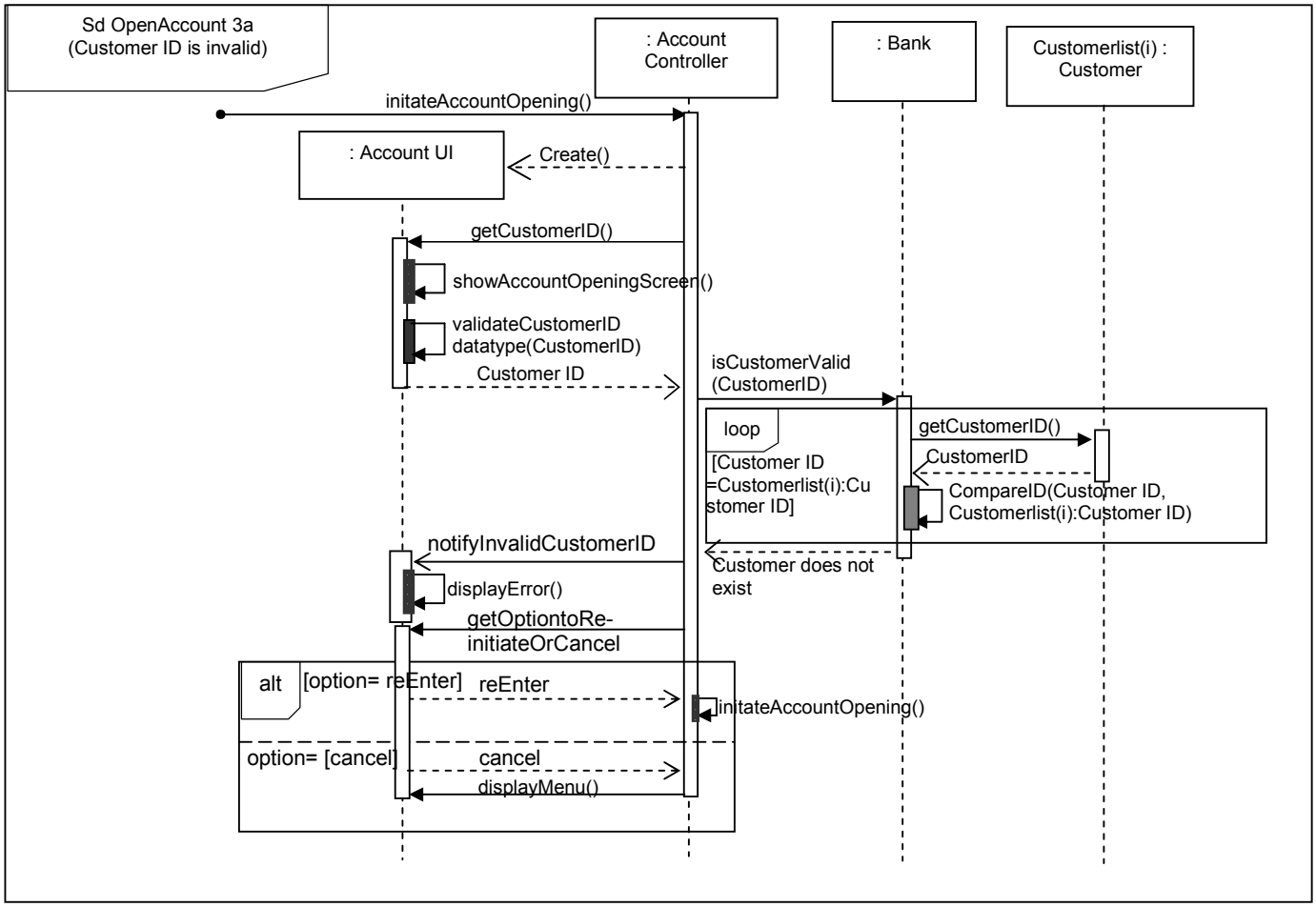


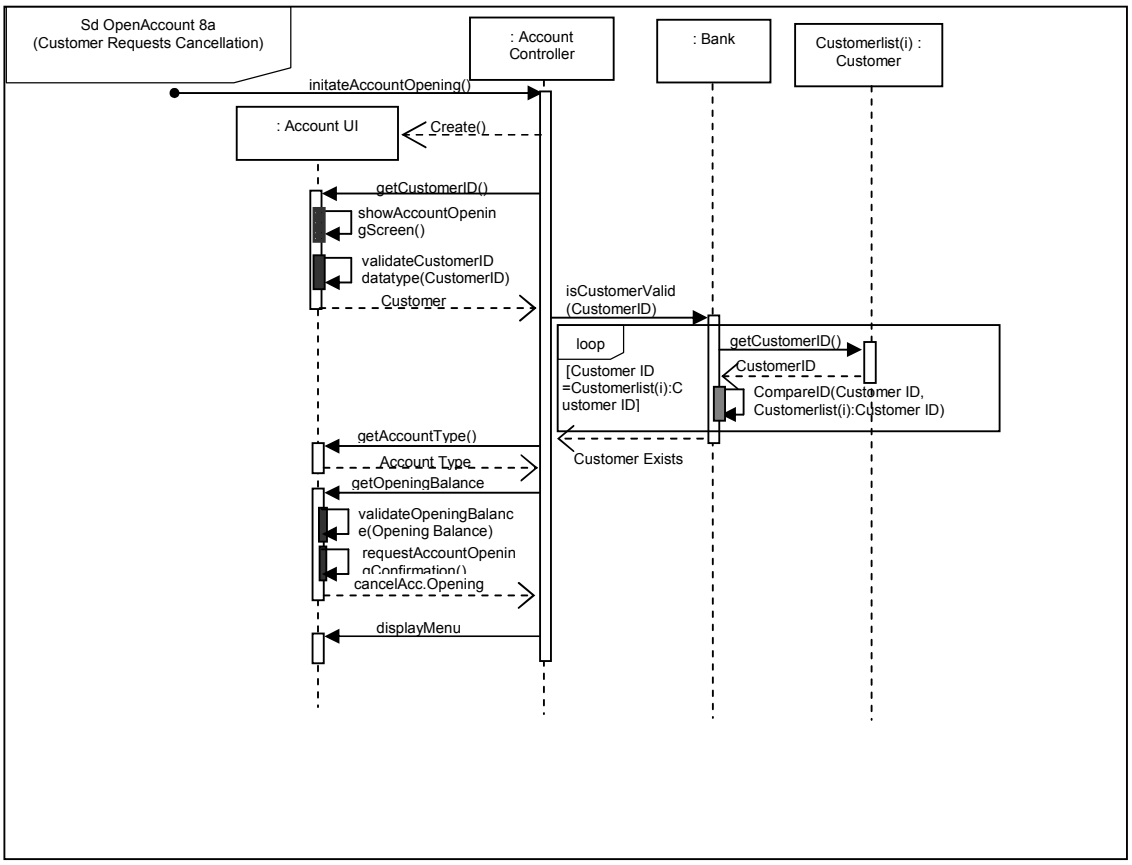

sd Add Customer 2a
(Invalid Customer Details)
: CustomerController
initiateCustomerRegistration()
: CustomerUI
Create()
getCustomerDetails()
showCustomerRegsitrationScreen()
validateCustomerDetails
(customer details)
displayErrorMessage(message)
displayMenu()

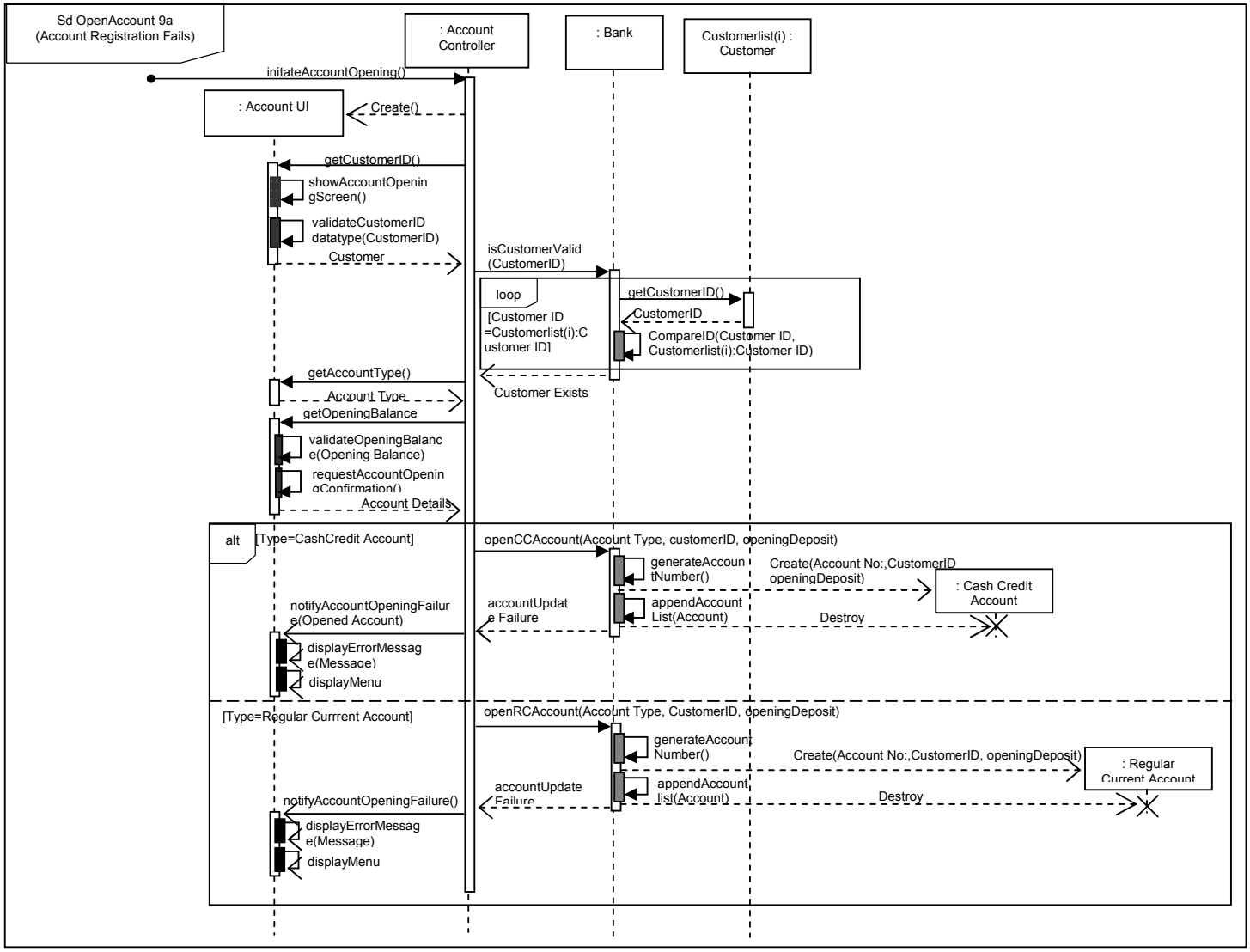

sd Add Customer 5a
(Customer cancels registration)
: CustomerController
initiateCustomerRegistration()
: CustomerUI
Create()
getCustomerDetails()
showCustomerRegsitrationScreen()
validateCustomerDetails
(customer details)
requestCustomerRegistration
Confirmation()
displayMenu()

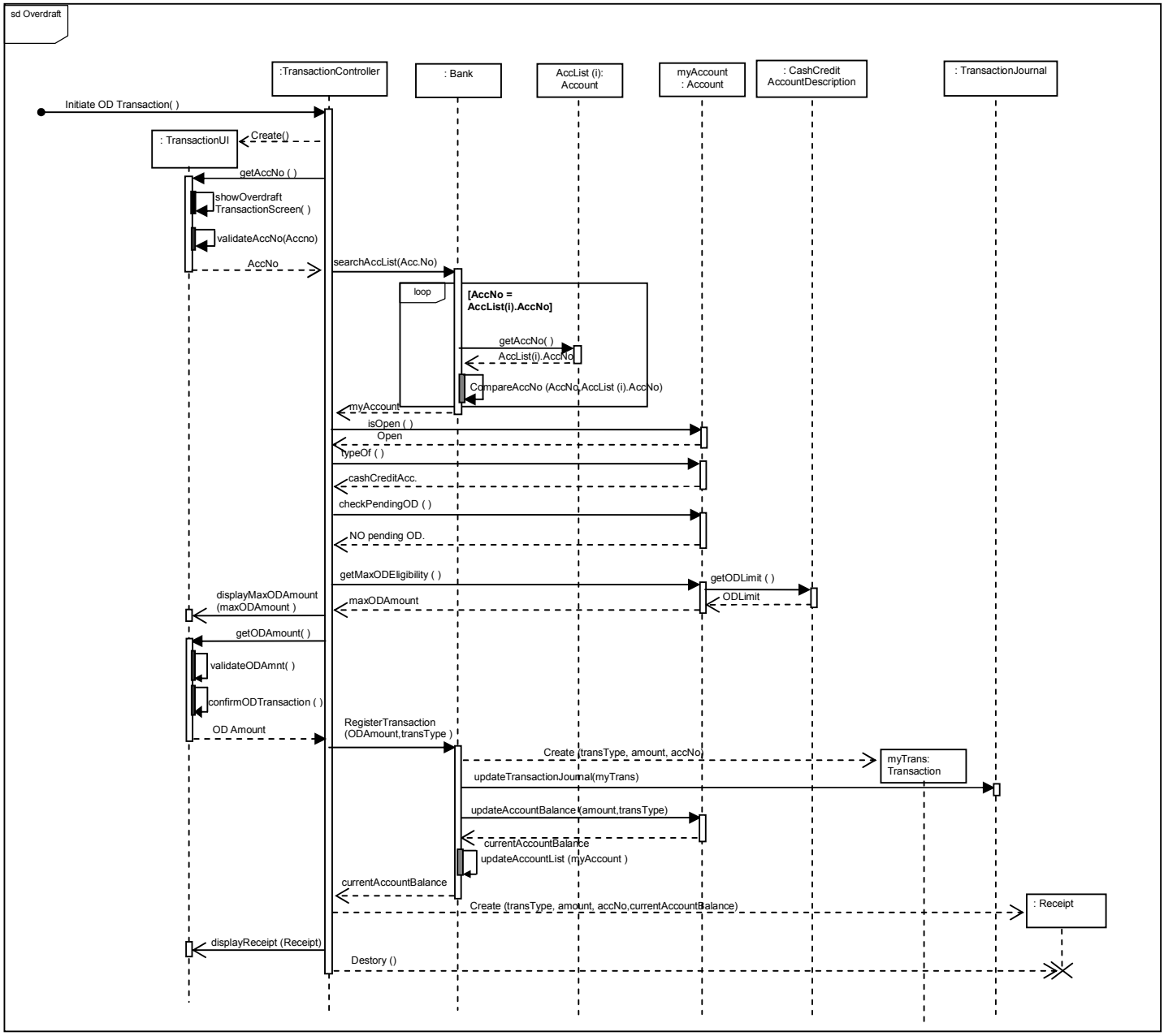

sd Add Customer 7a
(Customer registration failure)
:CustomerController
: Bank
initiateCustomerRegistration()
: CustomerUI
Create()
getCustomerDetails()
showCustomerRegsitrationScreen( )
validateCustomerDetails(custo
mer details)
requestCustomerRegistrationC
onfirmation()
customerDetails
registerCustomer (customer details)
generateCustomerID()
Create(Customer ID,
Customer details)
:Customer
appendCustomerList
(Customer)
destroy( )
Customer Update Failure
notifyRegFailure()
displayErrorMessage(message)
displayMenu()

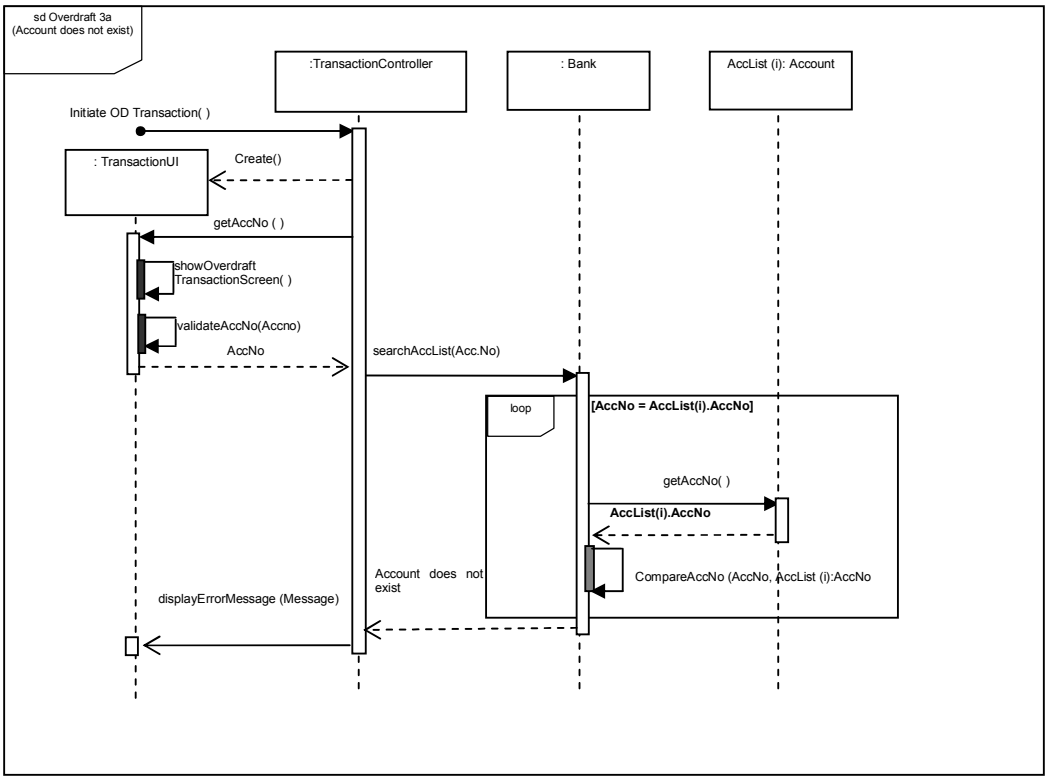

Sd OpenAccount
(Main Success Scenario)
: Account Controller
: Bank
Customerlist(i) : Customer
initateAccountOpening()
: Account UI
Create()
getCustomerID()
showAccountOpenin gScreen()
validateCustomerID datatype(CustomerID)
Customer
isCustomerValid (CustomerID)
loop
[Customer ID =Customerlist(i):C ustomer ID]
getCustomerID()
CustomerID
CompareID(Customer ID, Customerlist(i):Customer ID)
Customer Exists
getAccountDetails()
getAccountType ()
getOpeningBalance ()
validateOpeningBalanc e(Opening Balance)
Account Details
requestAccountOpenin gConfirmation()
Confirmation
alt
[Type=CashCredit Account]
openCCAccount(Account Type, customerID, openingDeposit)
generateAccount Number()
Create(Account No:,CustomerID openingDeposit)
: Cash Credit Account
appendAccount List(Account)
Opened Account
notifyAccountOpeningSuces s(Opened Account)
getAccountNumber()
Account Number
displayAccountNumbe r (AccountNumber)
displayMen
[Type=Regular Current Account]
openRCAccount(Account Type, CustomerID, openingDeposit)
generateAcco untNumber()
Create(Account No:,CustomerID, openingDeposit)
:Regular Current Account
appendAccount list(Account)
Opened Account
notifyAccountOpeningSuces s(Opened Account)
getAccountNumber()
Account Number
displayAccountNumbe r (AccountNumber)
displayMen

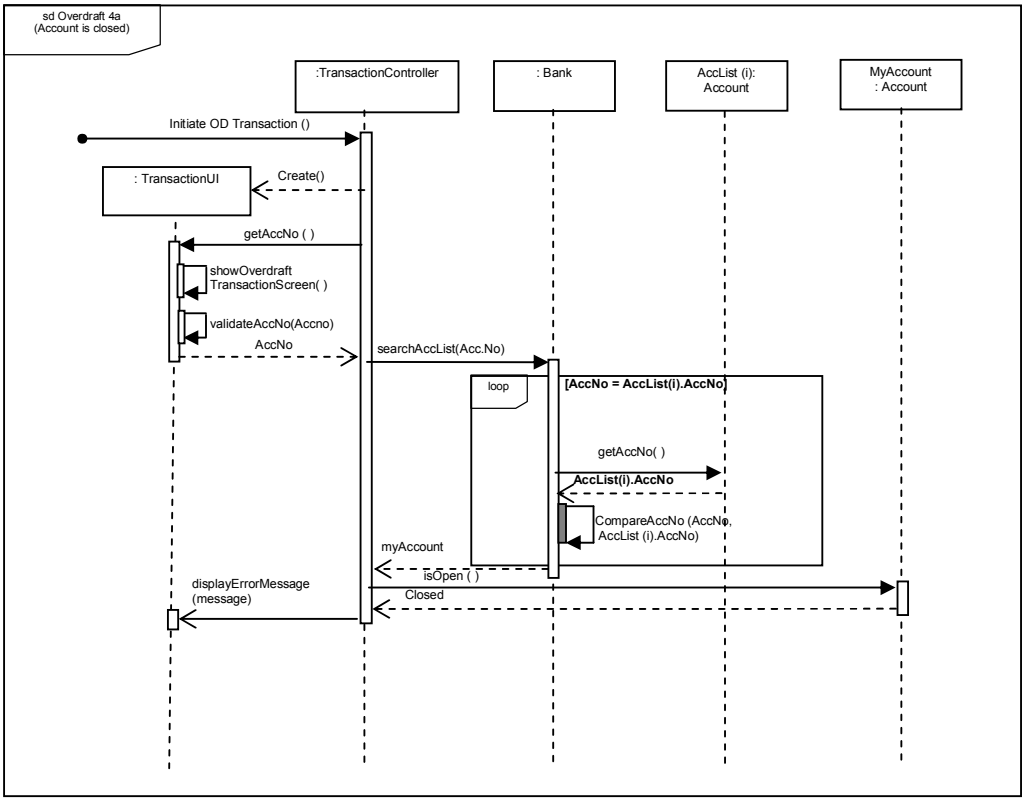

Sd OpenAccount 3a
(Customer ID is invalid)
: Account Controller
: Bank
Customerlist(i) : Customer
initateAccountOpening()
: Account UI
Create()
getCustomerID()
showAccountOpeningScreen()
validateCustomerID
datatype(CustomerID)
Customer ID
isCustomerValid
(CustomerID)
loop
[Customer ID =Customerlist(i):Customer ID]
getCustomerID()
CustomerID
CompareID(Customer ID, Customerlist(i):Customer ID)
Customer does not exist
notifyInvalidCustomerID
displayError()
getOptiontoRe-initiateOrCancel
alt
[option= reEnter]
reEnter
initateAccountOpening()
option= [cancel]
cancel
displayMenu()

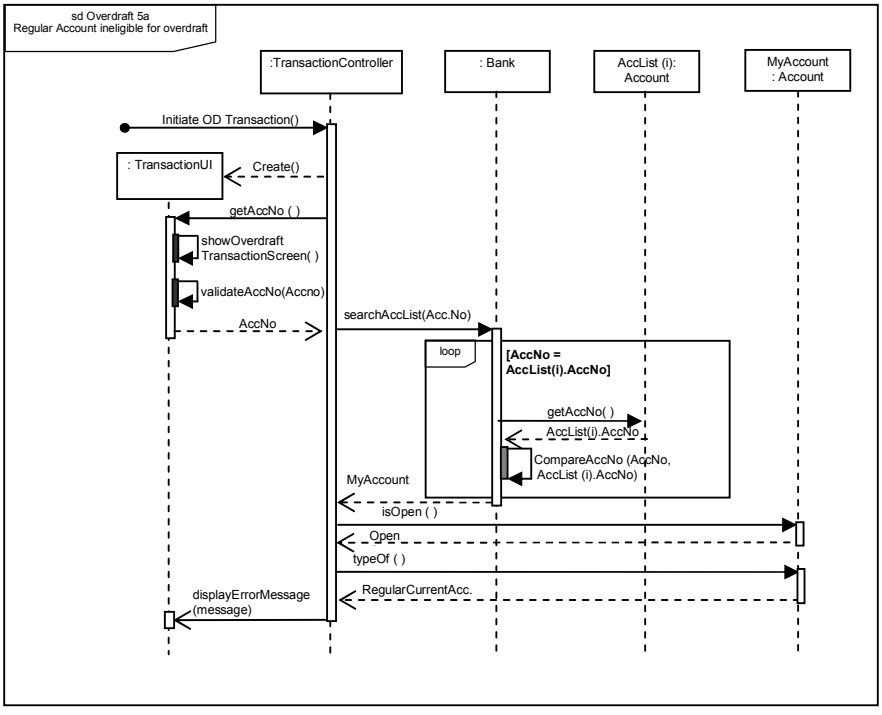

Sd OpenAccount 8a
(Customer Requests Cancellation)
: Account Controller
: Bank
Customerlist(i) : Customer
initateAccountOpening()
: Account UI
Create()
getCustomerID()
showAccountOpenin gScreen()
validateCustomerID datatype(CustomerID)
Customer
isCustomerValid (CustomerID)
loop
[Customer ID =Customerlist(i):C ustomer ID]
getCustomerID()
CustomerID
CompareID(Customer ID, Customerlist(i):Customer ID)
Customer Exists
getAccountType()
Account Type
getOpeningBalance
validateOpeningBalanc e(Opening Balance)
requestAccountOpenin gConfirmation()
cancelAcc.Opening
displayMenu

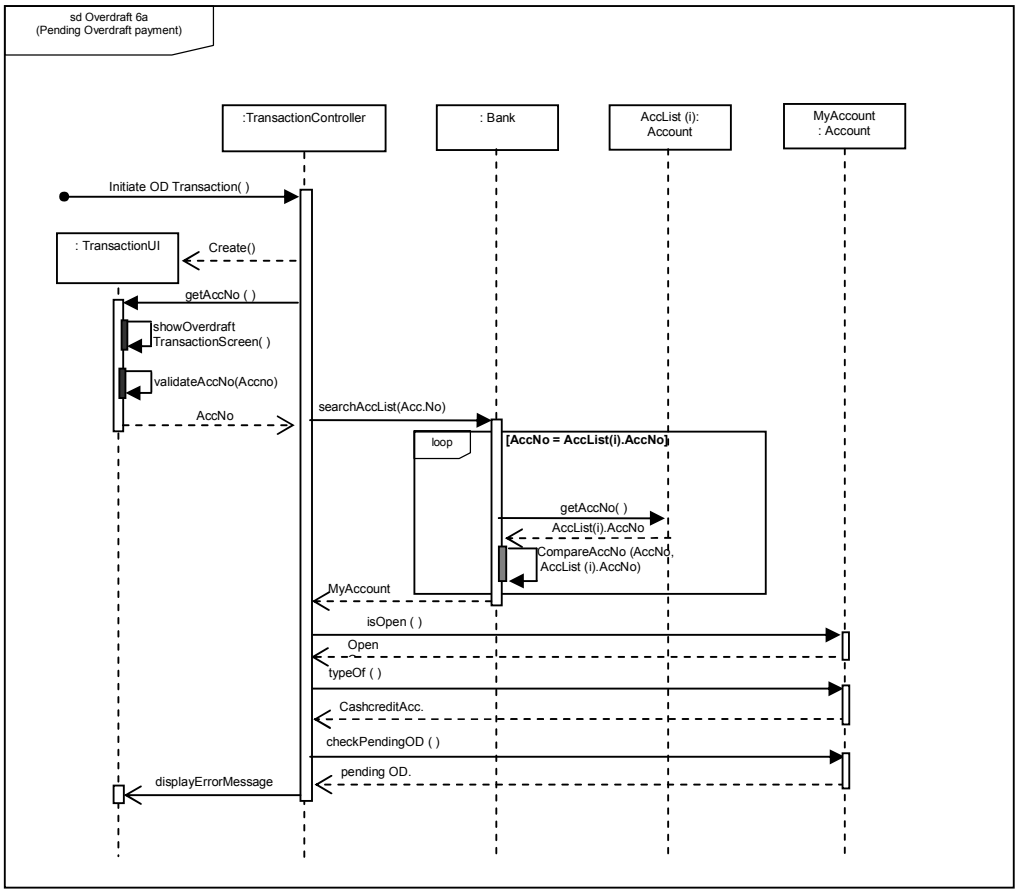

Sd OpenAccount 9a
(Account Registration Fails)
: Account Controller
: Bank
Customerlist(i) : Customer
initateAccountOpening()
: Account UI
Create()
getCustomerID()
showAccountOpenin gScreen()
validateCustomerID datatype(CustomerID)
Customer
isCustomerValid (CustomerID)
loop
[Customer ID =Customerlist(i):C ustomer ID]
getCustomerID()
CustomerID
CompareID(Customer ID, Customerlist(i):Customer ID)
Customer Exists
getAccountType()
Account Type
getOpeningBalance
validateOpeningBalanc e(Opening Balance)
requestAccountOpenin gConfirmation()
Account Details
alt
[Type=CashCredit Account]
openCCAccount(Account Type, customerID, openingDeposit)
generateAccoun tNumber()
Create(Account No:,CustomerID openingDeposit)
: Cash Credit Account
appendAccount List(Account)
Destroy
accountUpdat e Failure
notifyAccountOpeningFailur e(Opened Account)
displayErrorMessag e(Message)
displayMenu
[Type=Regular Currrent Account]
openRCAccount(Account Type, CustomerID, openingDeposit)
generateAccount Number()
Create(Account No:,CustomerID, openingDeposit)
: Regular Current Account
appendAccount list(Account)
Destroy
accountUpdate Failure
notifyAccountOpeningFailure()
displayErrorMessag e(Message)
displayMenu

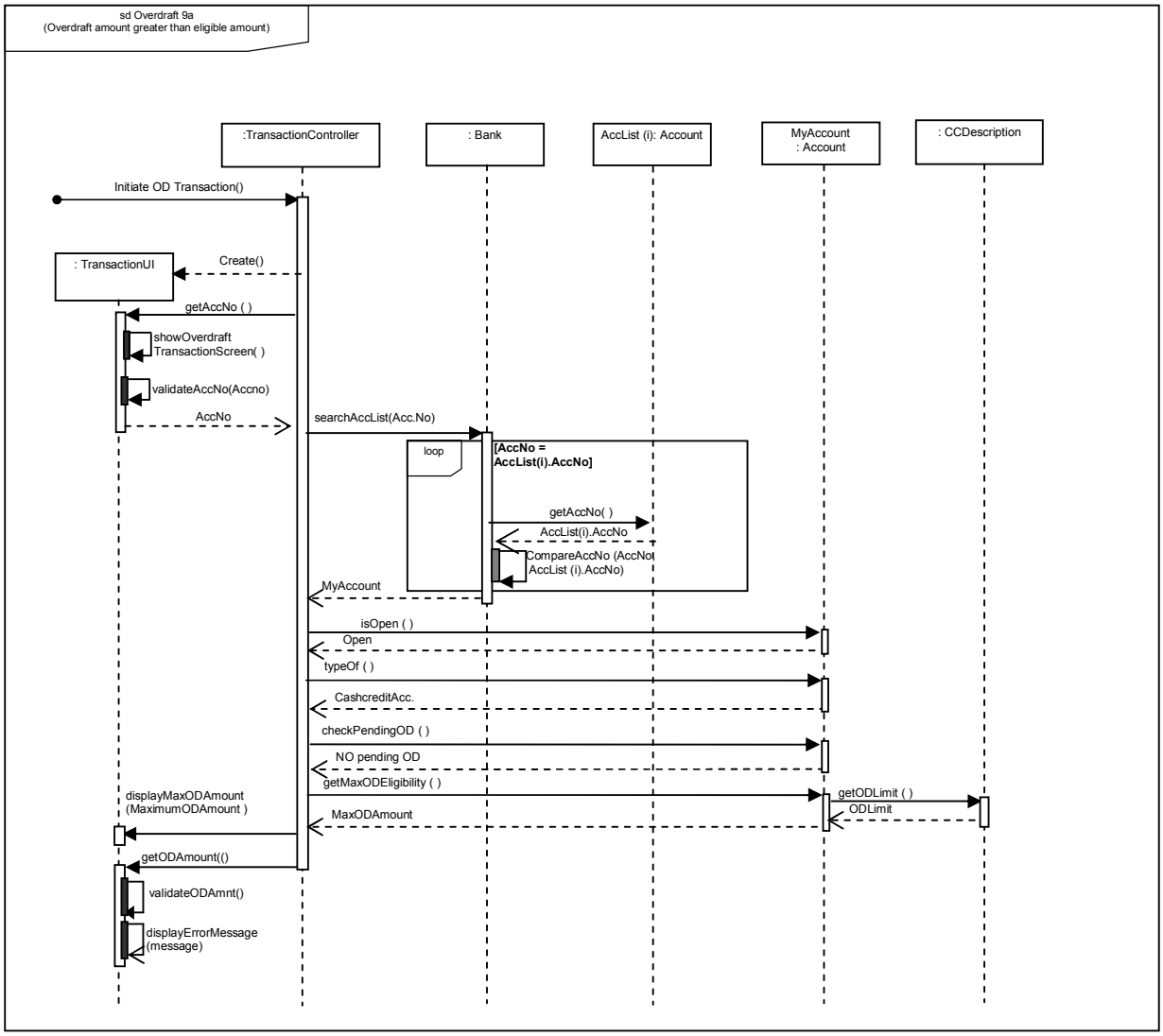

sd Overdraft
:TransactionController
: Bank
AccList (i): Account
myAccount : Account
: CashCredit AccountDescription
: TransactionJournal
Initiate OD Transaction( )
: TransactionUI
Create()
getAccNo ( )
showOverdraft TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo = AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo, AccList (i).AccNo)
myAccount
isOpen ( )
Open
typeOf ( )
cashCreditAcc.
checkPendingOD ( )
NO pending OD.
getMaxODEligibility ( )
getODLimit ( )
ODLimit
displayMaxODAmount (maxODAmount)
maxODAmount
getODAmount( )
validateODAmnt( )
confirmODTransaction ( )
OD Amount
RegisterTransaction (ODAmount,transType)
Create (transType, amount, accNo)
myTrans: Transaction
updateTransactionJournal(myTrans)
updateAccountBalance (amount,transType)
currentAccountBalance
updateAccountList (myAccount)
currentAccountBalance
Create (transType, amount, accNo,currentAccountBalance)
: Receipt
displayReceipt (Receipt)
Destory ()

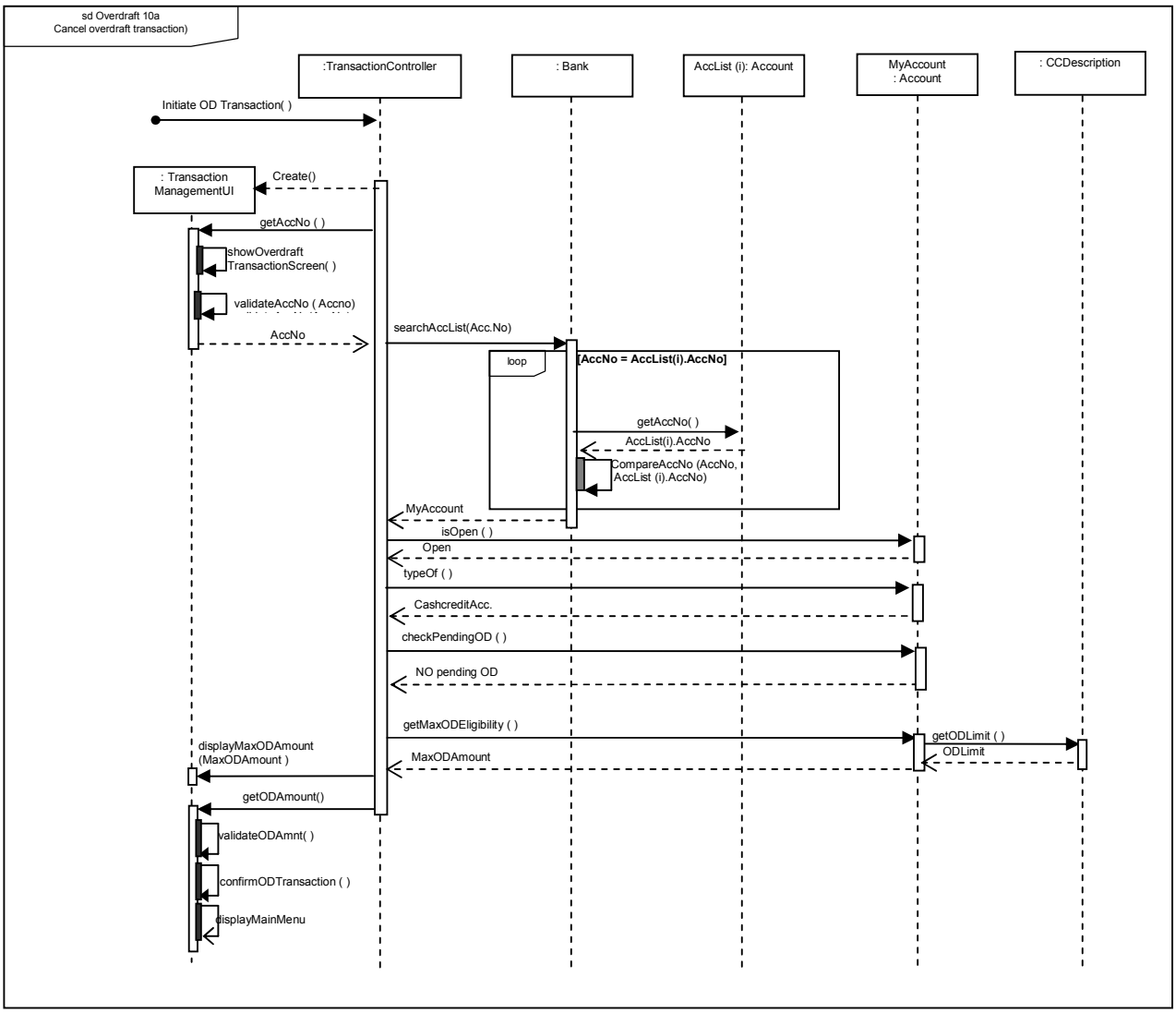

sd Overdraft 3a
(Account does not exist)
:TransactionController
: Bank
AccList (i): Account
Initiate OD Transaction( )
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo = AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo, AccList (i):AccNo
Account does not exist
displayErrorMessage (Message)

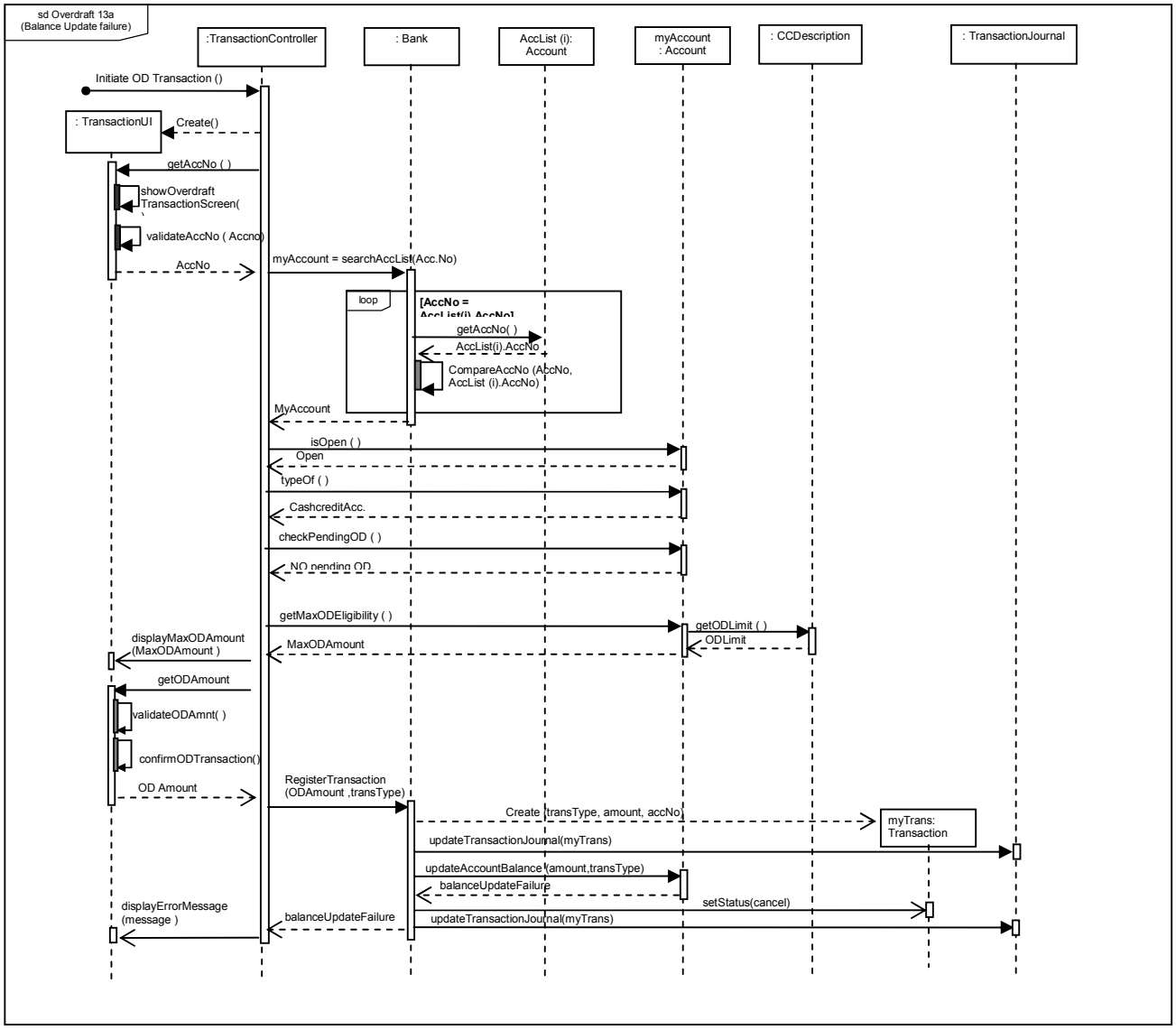

sd Overdraft 4a
(Account is closed)
:TransactionController
: Bank
AccList (i):
Account
MyAccount
: Account
Initiate OD Transaction ()
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo = AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo,
AccList (i).AccNo)
myAccount
isOpen ( )
displayErrorMessage
(message)
Closed


sd Overdraft 5a
Regular Account ineligible for overdraft
:TransactionController
: Bank
AccList (i):
Account
MyAccount
: Account
Initiate OD Transaction()
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo =
AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo,
AccList (i).AccNo)
MyAccount
isOpen ( )
Open
typeOf ( )
RegularCurrentAcc.
displayErrorMessage
(message)

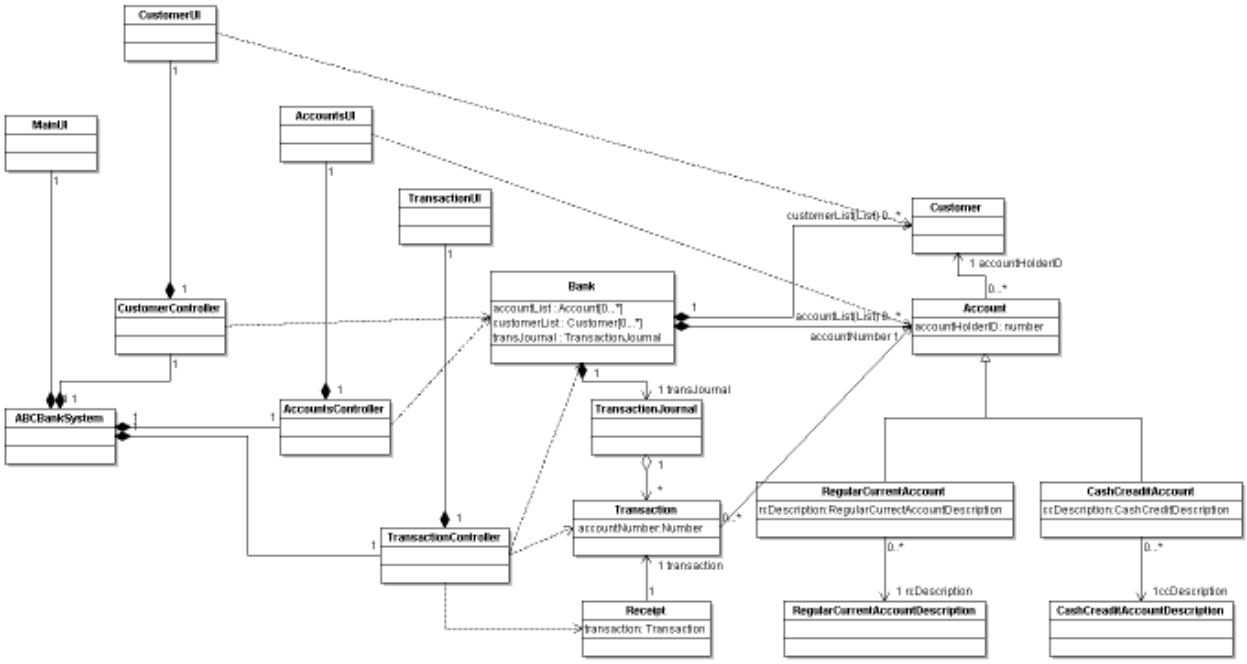

sd Overdraft 6a
(Pending Overdraft payment)
:TransactionController
: Bank
AccList (i):
Account
MyAccount
: Account
Initiate OD Transaction( )
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo = AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo,
AccList (i).AccNo)
MyAccount
isOpen ( )
Open
typeOf ( )
CashcreditAcc.
checkPendingOD ( )
pending OD.
displayErrorMessage


sd Overdraft 9a
(Overdraft amount greater than eligible amount)
:TransactionController
: Bank
AccList (i): Account
MyAccount
: Account
: CCDescription
Initiate OD Transaction()
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo(Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo =
AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo
AccList (i).AccNo)
MyAccount
isOpen ( )
Open
typeOf ( )
CashcreditAcc.
checkPendingOD ( )
NO pending OD
getMaxODEligibility ( )
getODLimit ( )
ODLimit
displayMaxODAmount
(MaximumODAmount)
MaxODAmount
getODAmount(()
validateODAmnt()
displayErrorMessage
(message)


sd Overdraft 10a
Cancel overdraft transaction)
:TransactionController
: Bank
AccList (i): Account
MyAccount
: Account
: CCDescription
Initiate OD Transaction( )
: Transaction
ManagementUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen( )
validateAccNo (Accno)
AccNo
searchAccList(Acc.No)
loop
[AccNo = AccList(i).AccNo]
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo,
AccList (i).AccNo)
MyAccount
isOpen ( )
Open
typeOf ( )
CashcreditAcc.
checkPendingOD ( )
NO pending OD
getMaxODEligibility ( )
getODLimit ( )
ODLimit
displayMaxODAmount
(MaxODAmount)
MaxODAmount
getODAmount()
validateODAmnt( )
confirmODTransaction ( )
displayMainMenu


sd Overdraft 13a
(Balance Update failure)
:TransactionController
: Bank
AccList (i):
Account
myAccount
: Account
: CCDescription
: TransactionJournal
Initiate OD Transaction ()
: TransactionUI
Create()
getAccNo ( )
showOverdraft
TransactionScreen(
validateAccNo (Accno)
AccNo
myAccount = searchAccList(Acc.No)
loop
[AccNo =
getAccNo( )
AccList(i).AccNo
CompareAccNo (AccNo,
AccList (i).AccNo)
MyAccount
isOpen ( )
Open
typeOf ( )
CashcreditAcc.
checkPendingOD ( )
NO pending OD
getMaxODEligibility ( )
getODLimit ( )
ODLimit
displayMaxODAmount
(MaxODAmount)
MaxODAmount
getODAmount
validateODAmnt( )
confirmODTransaction()
OD Amount
RegisterTransaction
(ODAmount ,transType)
Create (transType, amount, accNo)
myTrans:
Transaction
updateTransactionJournal(myTrans)
updateAccountBalance (amount,transType)
balanceUpdateFailure
setStatus(cancel)
displayErrorMessage
(message)
balanceUpdateFailure
updateTransactionJournal(myTrans)

3c.2.Design level Class Diagram



4. Appendix A (Conceptual Classes discovered using category list)

Conceptual Class Category	Identified Classes
Business Transactions	Deposit, Withdraw, Overdraft
Transaction Line Items	Transaction Amount
Product or service related to a transaction or transaction line item	Money, Account.
Where is transaction recorded?	Transaction Journal.
Roles of people related to the transaction; actors in the use case	Clerk, Customer.
Place of transaction	ABC Bank
Noteworthy events, often with a time or place we need to remember	NA
Physical objects	Money

Description of things	Regular Current Account, Cash Credit Account
Catalogs	NA
Containers of things(physical or information)	Transaction Journal.
Things in a container	Deposit, Withdraw or Overdraft Transactions
Other collaborating systems	NA
Financial instruments	Money
Organizations	ABC Bank
Rules and policies	Overdraft Payment Policy.
Records of finance, work, contracts, legal matters	Receipt, Transaction Journal, Transaction Report.

5. Appendix B (Classes identified by noun-phrase analysis of use cases (UC1, UC4 & UC7)

Nouns and Noun Phrases Identified from UC1: Add Customer

1. Customer Details
2. Customer
3. Customer Id
4. ABC Bank
5. Clerk **
6. System ***
7. Error Message

Class Candidates	Attribute Candidates
1.Customer	1. Customer Details
2.ABC Bank	2. Customer Id
	3. Error Message

*Note: Details of clerk is not required in this scenario, so clerk is not considered as a class.
**Note: System represents the whole system, its not a candidate for a class.

Nouns and Noun Phrases Identified from UC4: Open Account

Class Candidates	Attribute Candidates
1.Customer	1.Account Number
2.Account	2.Account Details
3.Regular Current Account	3.Account Type
4.Cash Credit Account	4.ABC Bank
	5.Customer ID
	6. Account opening Details.

Nouns and Noun Phrases Identified from UC7: Overdraft

~~1. ABC Bank Automation System.~~
~~2. Clerk~~
3. Customer
4. Overdraft Amount
~~5. ABC Bank~~
~~6. Time~~
~~7. Effort~~
8. Account Balance
9. Receipt
~~10. Overdraft Transaction*~~
11. Account Number
12. Cash Credit Account
~~13. Overdraft Payment~~
14. Account Transaction
15. Account Transaction Details
16. Transaction Journal
17. Account Transaction Report
18. Account
19. Account Type
20. Eligible Limit
21. Transaction ID
22. Error Message
23. Option
~~24. Valid Account Number~~
~~25. Withdrawal Transaction*~~

* We are not considering Overdraft Transaction & Withdrawal Transaction for classes; instead we define Transaction Type as an attribute of Transaction.

Class Candidates	Attribute Candidates
1.Customer	1.Overdraft Amount
2. Receipt	2.Account Balance
3. Cash Credit Account	3.Account Number
4. Account Transaction	4.Account Transaction Details
5. Account Transaction Report	5. Account Type.
6. Account	6. Eligible Limit
7.Transaction Journal	7.Transaction ID
	8. Error Message

6.Appendix C

GLOSSARY

Term	**Definition and Information**
Current account:	It is a transactional account offering methods to allow customers to distribute money. Customers of ABC Bank can withdraw or deposit money to their current account with the mediation of a Clerk. ABC Bank had two types of current accounts for its customers: Cash Credit account and Regular Current account.
Cash Credit account	This account is the primary method in which ABC Bank lends money against the security of commodities and debt. It runs like a Regular Current account except that the money that can be withdrawn from this account is not restricted to the amount deposited in the account. Instead, the account holder is permitted to withdraw a certain sum called "limit" in excess of the amount deposited in the account. Cash Credit account holders of ABC bank have an overdraft limit of 10% of their balance amount in their account.

Overdraft	Overdraft facility in ABC bank its Cash Credit account holders to overdraw money from a Bank account. The account holder can withdraw more money from a Bank Account than has been deposited in it.
Regular Current account	With a Regular Current account, a customer can access his/her account anytime during the bank working hours to perform cash transfer. Regular Current Account requires you to pay a fee of Rs.100/- for every 100 transactions made from that account.
Account	A bank account is a financial account with a banking institution recording the financial transactions between the customer and the bank and the resulting financial position of the customer with the bank. Bank accounts may have a positive balance where the bank owes money to the customer; or a negative balance where the customer owes the bank money.
Account Number	Account Number is a standard for uniquely identifying bank accounts in a Bank.
Customer Id	ABC Bank has a unique Customer ID for each of its customers. Every customer is the bank is provided with a customer ID. It is an identifier for a customer of the bank. The first time you create a new Customer ID you will need to register with the bank by providing some required personal details.
Registered customer	A customer of ABC bank is said to be a registered customer if he has provided his personal details to register with the bank and obtained a unique customer Id.
Transaction	Bank transactions handle money that goes into or out of customer account. Whenever money is added or subtracted in your bank accounts, it is recorded as a transaction.
Transaction Report	ABC bank is required to file for each deposit, withdrawal, overdraft which involves a transaction. Transaction report records the details of a transaction a stores it in the system for future reference.
Overdraft Repayment	Customer availing overdraft facility, agree to pay back the overdraft balance, including applicable overdraft fees due towards the bank.

TATA CONSULTANCY SERVICES

Experience certainty. IT Services
Business Solutions
Outsourcing

TATA CONSULTANCY SERVICES

www.tcs.com