



Design Document

For AB-SERVICE DIRECTORY

Prepared for

Fresher05

Monday, October 20, 2014

Version 0.1

Prepared by

Pham Trung Hai

Revision and Signoff Sheet

Change Record

Date	Author	Version	Change reference
09/18/2011	Pham Trung Hai	0.1	Create new
10/11/2011	nganptq	0.1	4, 5
10/12/2011	Sonbh1	0.1	7.1
10/12/2011	Nganlt1	0.1	7.2
10/12/2011	Hungnm7	0.1	7.3, 7.4
10/12/2011	Cuongnv10	0.1	7.5
10/12/2011	Minhnn3	0.1	7.6, 7.7

Reviewers

Name	Version	Position	Date
Nguyen Trung Kien	0.1	QA	08-Sep-2010
Nguyen Minh Tien	0.1	QA	08-Sep-2010

Table of Contents

1	Introduction	31
1.1	Purpose.....	31
1.2	Scope.....	31
1.3	Intended Audiences and Document Organization	31
1.4	Acronyms and Abbreviations	31
1.5	References	32
2	Architecture design	32
2.1	Application Logical Architecture	32
2.1	User case view	32
2.1.1	Overview	32
2.1.2	Description	33
2.2	Architectural Representation	33
2.2.1	Presentation Layer.....	34
2.2.2	Business Layer	34
2.2.3	Data Access Layer.....	35
2.3	Packages/Components view	35
2.2.4	UI Components	35
2.2.5	Business Entities	36
2.2.6	Data Access Object	36
2.2.7	Exceptions Lib	36
2.2.8	General Lib	36
2.2.9	Logging Lib	36
3	Technical Solutions	37
3.1	Exception handling mechanism	37
3.2	Logging mechanism.....	37
4	Database design HuyVQ4	37
4.1	Entity Relationship Diagram	37
4.2	Schema.....	37
4.3	Detail Schema	38
4.3.1	Customer	38
4.3.2	Account.....	38
4.3.3	Card	38
4.3.4	OverDraft Limit.....	38
4.3.5	Withdraw Limit.....	38
4.3.6	ATM	39
4.3.7	Money	39
4.3.8	Stock.....	39
4.3.9	LogType.....	39
4.3.10	Log	39
4.3.11	Config.....	40
5	Application Security	40
5.1	Main functions 's security matrix.....	40
6	Interface Design	41
6.1	Layout	41
6.2	Themes.....	41
7	Details function design	41
7.1	Use case 01: Validation – HuyVQ4	41
7.1.1	Validate Card	41
7.1.2	Authentication	46
7.2	Use case 02: Withdraw Money - LuyenNV	51
7.2.1	Withdraw	51
7.2.2	Dispense Cash	Error! Bookmark not defined.
7.3	Use case 03: Check Balance – LuyenNV.....	57

7.4	Use case 04: View History – HuyVQ4	61
7.4.1	View History	61
7.5	Use case 05: Cash Transfer – DucNA9	66
7.6	Use case 06: Change PIN – DucNA9.....	79
7.7	Use case 07: Logging – HuyVQ4	84
7.7.1	Logging	84
8	Configuration	86
8.1	Application Configuration.....	86
8.2	System Configuration	87
9	Packaging and Deployment	87
10	Appendix.....	87
10.1	Microsoft Enterprise library.....	87

1 Introduction

1.1 Purpose

This document provides a technical approach for implementing ATM application, using a number of different architectural diagrams to view different aspects of the system. It also presents the decisions that will govern how the application will be built from .NET framework with C# programming perspective.

This document's aim is to describe the architecture at high level, in which the system will be described as a set of functional layers, with technologies suggested for each layers.

1.2 Scope

This document is prepared for the ATM simulator application in scope of the Assignemtn of FRESHER11.

1.3 Intended Audiences and Document Organization

This document is intended for:

- ❖ Development team: Developers
- ❖ Customer Representatives: Responsible to review & approve the document.

Below are main sections of the document:

- ❖ **Introduction** : This section describes the general introduction of this document
- ❖ **Architecture Design** : This section describes the high-level technical assessments and decisions for the application.
- ❖ **Data Design**: This section describes in detail how data is structured and manipulated in this application.
- ❖ **Interface Design**: This section describes in detail how UI is designed in general (layout , theme).
- ❖ **Application Security**: This section describes security matrix in detail
- ❖ **Configuration**: This section describes all configuration needed for the application to function properly.
- ❖ **Packaging and Deployment**: This section describes how applications could be packaged and deployed.

1.4 Acronyms and Abbreviations

#	Item	Description
1	ATM	Automated Teller Machine
2	PL	Presentation Layer
3	BLL	Business Logic Layer
4	DAL	Data Access Layer
5	DAO	Data Access Object, this object is responsible for attaching to a system, extracting some information, based on specific requirements, and creating a value object.
6	VO	Value Object
7	BE	Business Entities

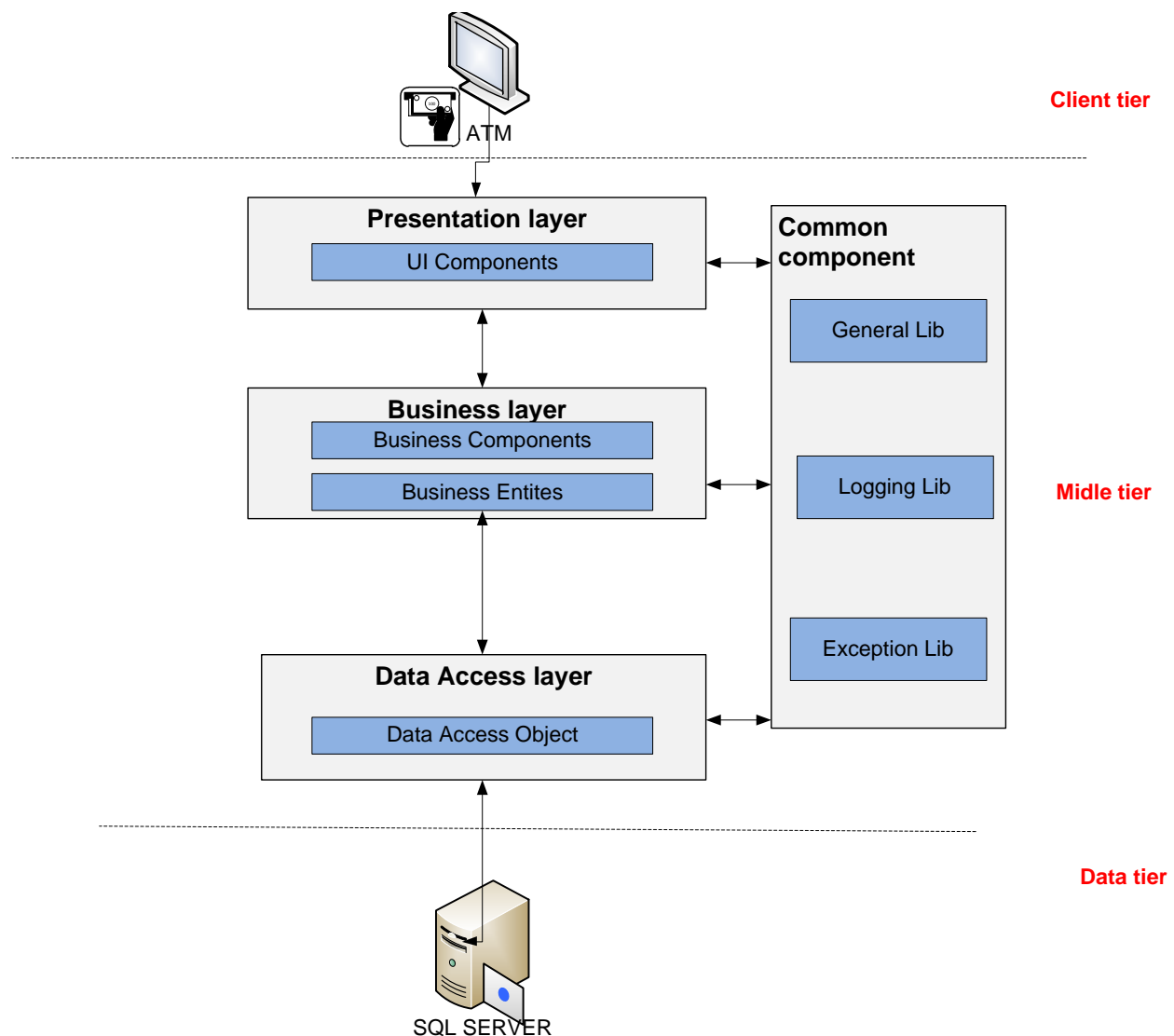
1.5 References

#	Title	Version	File Name / Link	Description
1	.NET Project_Software Design Document	1.0	.NET Project_Software Design Document.docx	
2				

Table 1.1: List of References

2 Architecture design

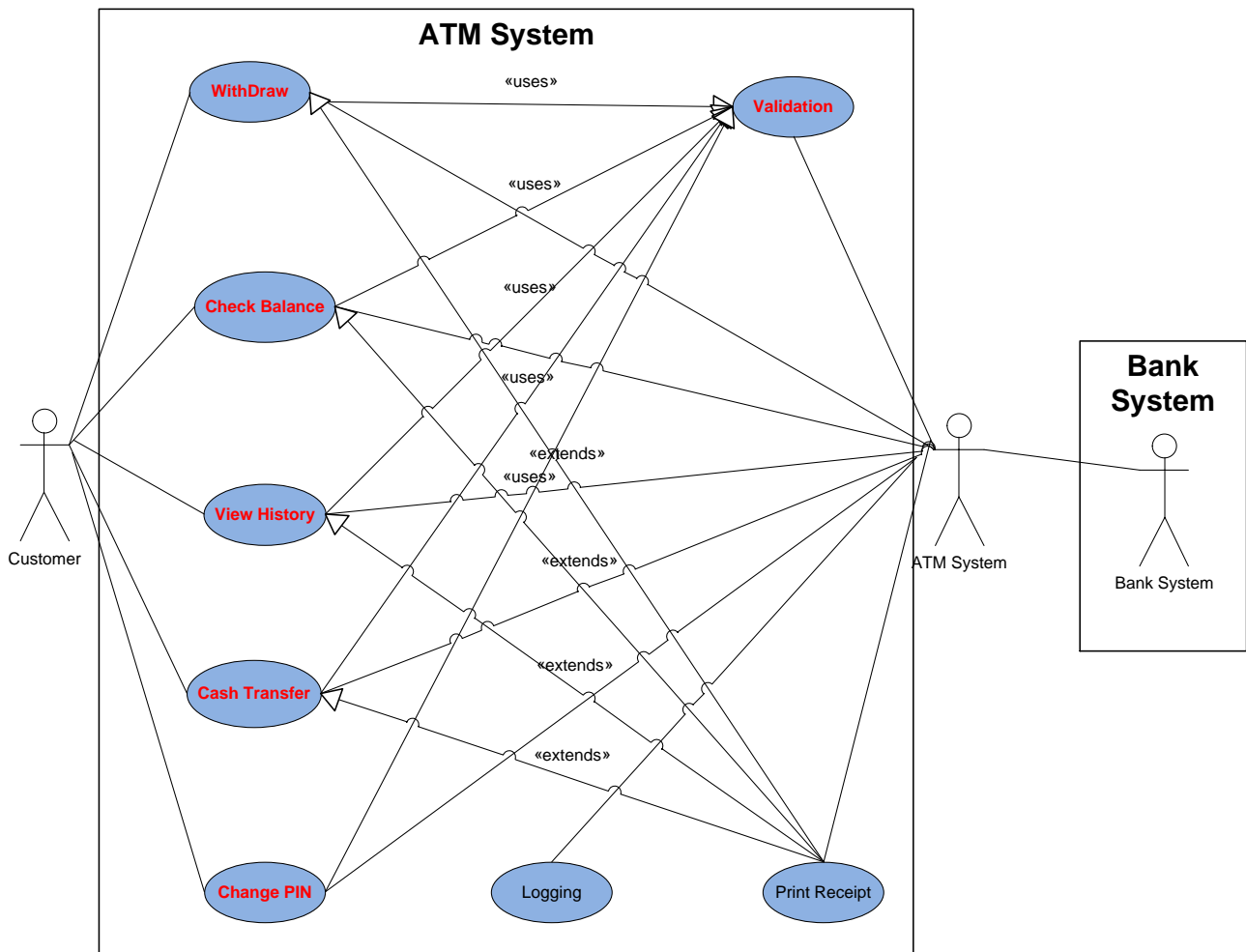
2.1 Application Logical Architecture



2.1 User case view

2.1.1 Overview

The following use cases are required to be included in the ATM simulator application



2.1.2 Description

UC	Name	Description
UC01	Validation	Validate customer card and PIN customer entered
UC02	Withdraw	Allow customer to withdraw money
UC03	Check balance	Allow customer to check their account balance
UC04	View history	Allow customer to view their succesful transaction
UC05	Cash Transfer	Allow customer to transfer their cash to another account in accepted bank system
UC06	Change PIN	Allow customer to change their PIN
UC07	Logging	System write log

2.2 Architectural Representation

The following diagram shows the primary tiers in the proposed n-tier architecture. This diagram shows the main layers in this architecture and the vision of how they fit together.

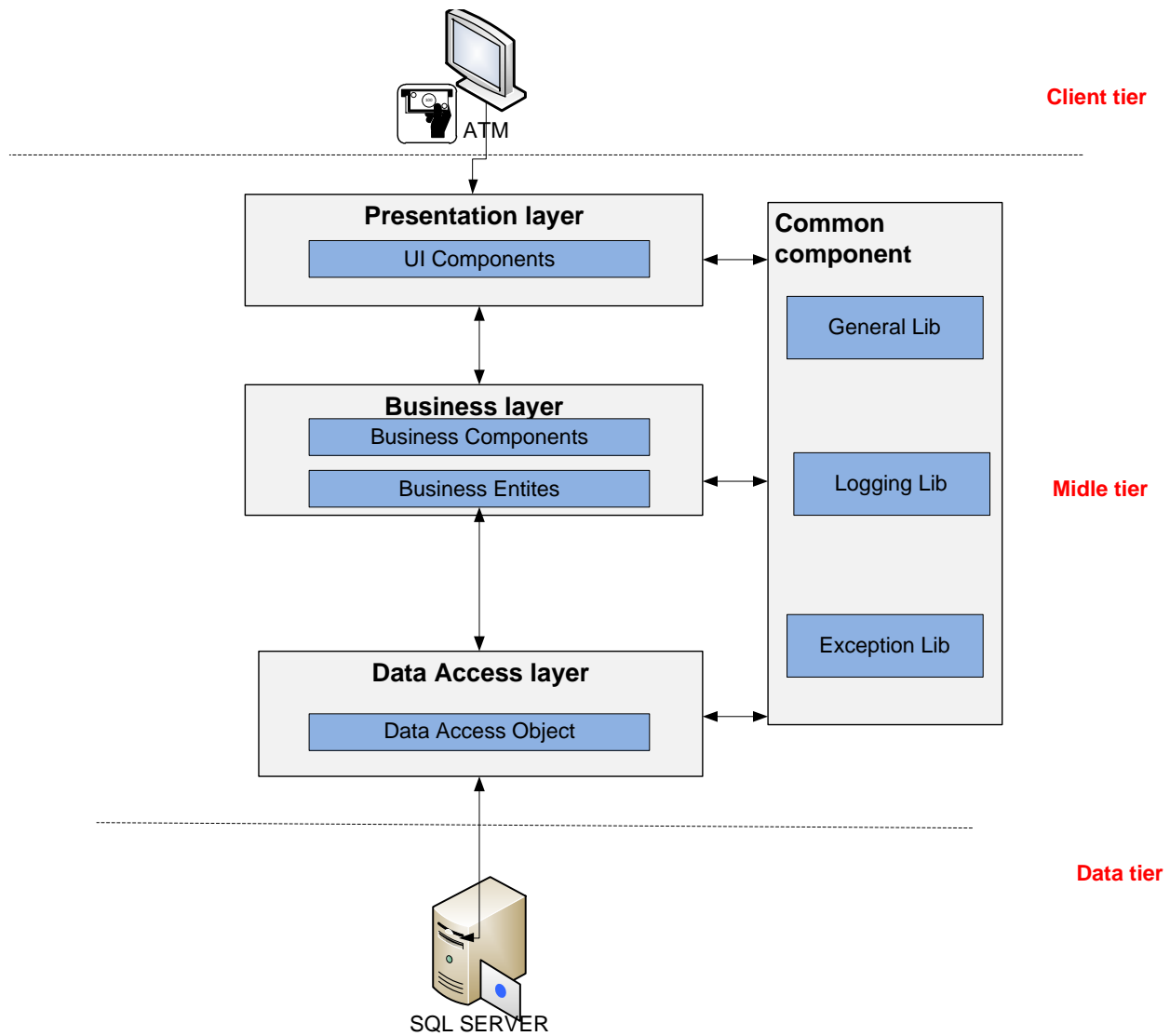


Figure 1 – N-tier architecture of ATM simulator system

2.2.1 Presentation Layer

This layer controls the display to the end user. The development framework is based on .NET Model architecture. The framework is responsible for:

- Managing requests/responses from/to the clients.
- Controlling display to the end user.
- Performing UI validation.
- Handling exceptions from other layers.

2.2.2 Business Layer

This layer manages the business processing rules and logic.

- Handling application business logic and business validation.
- Managing transactions.
- Allowing interfaces for interaction with other layers.
- Managing dependencies between business level objects.
- Adding flexibility between the presentation and the persistence layer so they do not directly communicate with each other.

- Exposing a context to the business layer from the presentation layer to obtain business services.
- Managing implementations from the business logic to the persistence layer.

2.2.3 Data Access Layer

This layer manages access to persistent storage. The primary reason to separate data access from the rest of the application is that it is easier to switch data sources and share Data Access Objects (DAOs) between applications.

This layer manages reading, writing, updating, and deleting stored data.

2.3 Packages/Components view

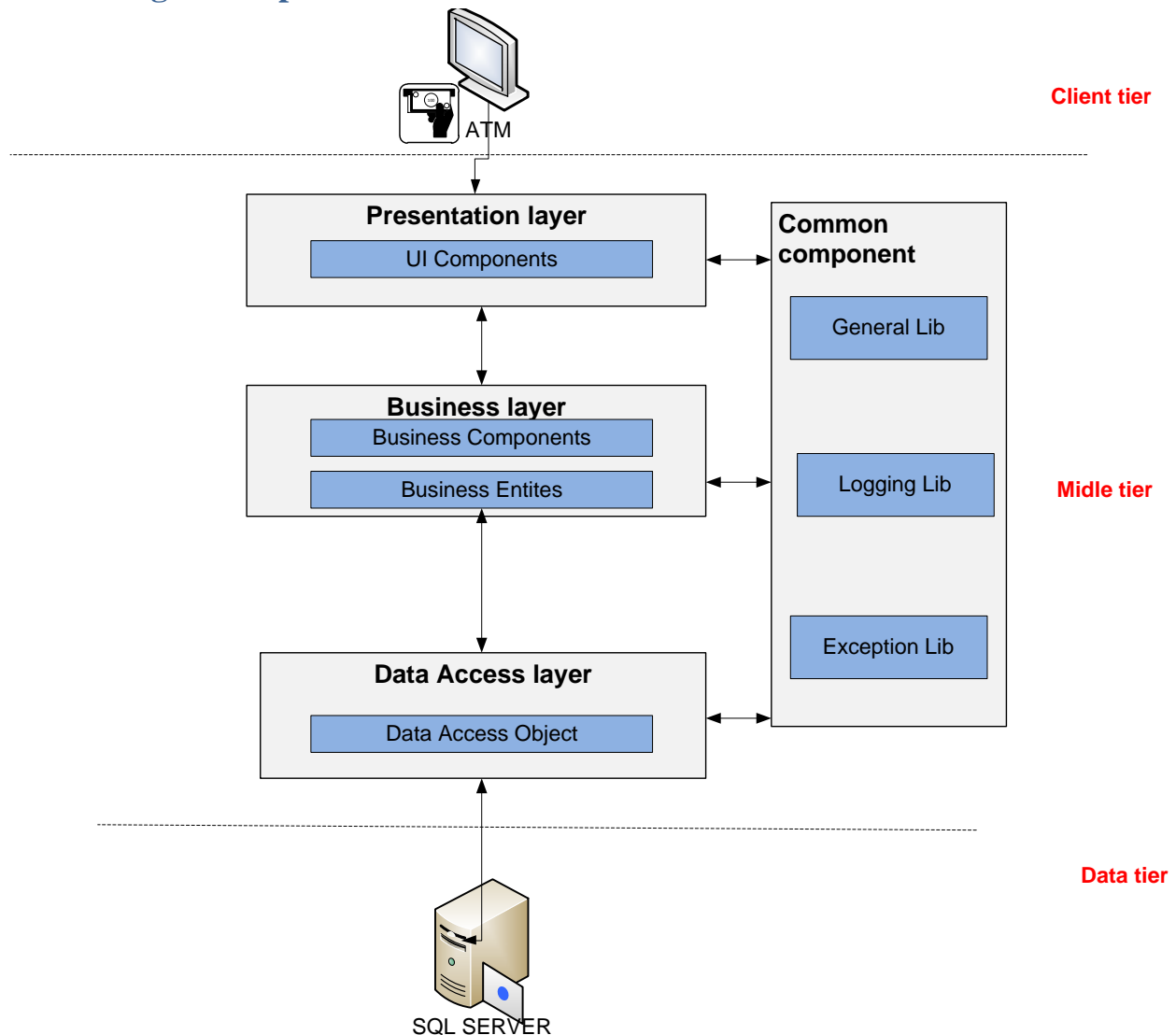


Figure 2 – Packages/Component view of ATM simulator system

2.2.4 UI Components

This package includes the implementation for the .NET architecture proposed to be used in the Presentation Layer to handle the display to the end user.

Validation: All validation of incoming requests parameters to the server should be validated using .NET client side control .

2.2.5 Business Entities

This package includes the implementation of business objects. **Business Entities** (BE) layer is used to perform the business operations. The Business Entities layer will access the DAO to access database. Transactions should be managed within this business layer.

2.2.6 Data Access Object

This package includes the implementation of Data Access Object. Using Linq to SQL classes here to make the application more flexible to access database. Linq to SQL classes includes basic functions to work with database: *select, insert, update, delete*.

2.2.7 Exceptions Lib

This package will include all general exceptions that will typically used by more than one package. The try-catch clauses should be kept to a minimum.

2.2.8 General Lib

This package includes all utilities .NET classes will be wisely used in the modules.

2.2.9 Logging Lib

This package includes implemented logging classes.

3 Technical Solutions

3.1 Exception handling mechanism

The try-catch clauses should be kept to a minimum.

Whenever an exception raise , system will log exception detail to Windows event and redirect user to Error page to display standard error message!

3.2 Logging mechanis

4 Database design HuyVQ4

4.1 Entity Relationship Diagram

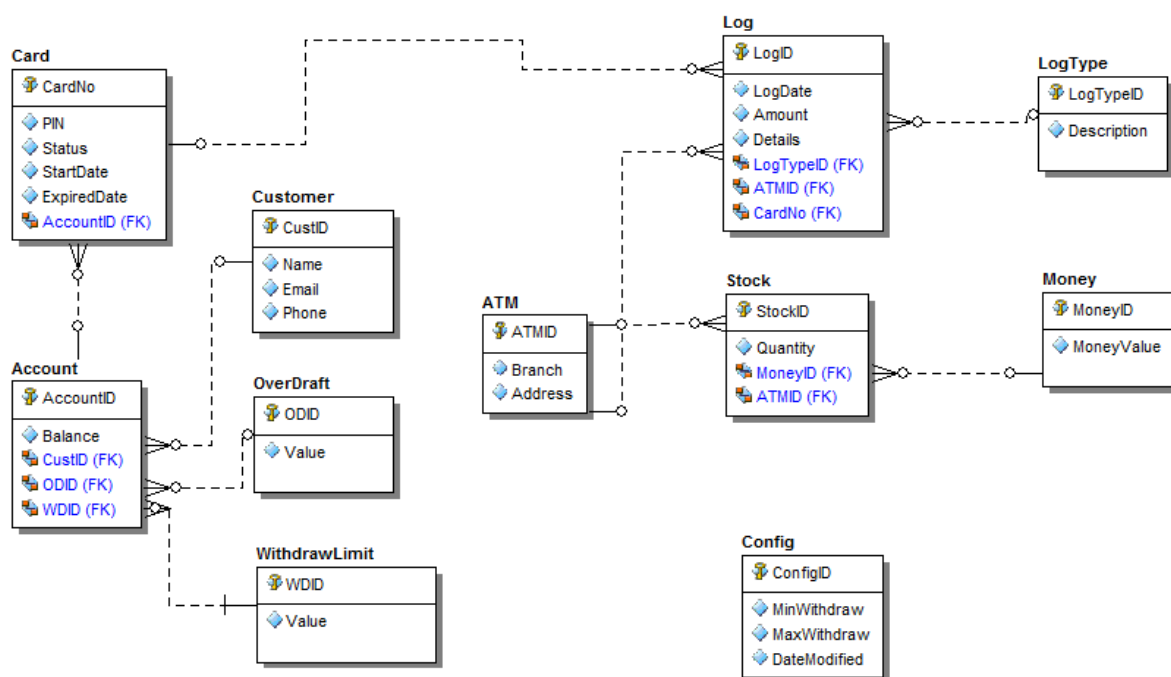


Figure 3 – Data overview

4.2 Schema

Overview

STT	Name	Description
1	Customer	List all Customer
2	Account	List all account use in system
3	Card	List all ATM card use in system
4	OverDraft	Amount of money that an account can over draft
5	WithdrawLimit	Limit of amount that an account can withdraw for a day
6	ATM	List all ATM use in system
7	Money	Kind of money and value
8	Stock	Kind of money and quantity of each kind stored in each ATM
9	Log	Log any transaction of customer
10	LogType	Kind of Log: Withdraw, transfer, check balance, change pin
11	Config	Store all system config: minwithdraw, maxwithdraw, number

		records per search results.
--	--	-----------------------------

4.3 Detail Schema

4.3.1 Customer

Customer				
	Field name	Type	Max Length	Description
1	CustID	int		Store ID of Customer
2	Name	nvarchar	100	Store name of Customer
3	Phone	Varchar	50	Store phone number of Customer
4	Email	Varchar	100	Store email address of Customer
5	Addr	Nvarchar	200	Address of Customer

4.3.2 Account

Account				
	Field name	Type	Max Length	Description
1	AccountID	int		Store ID of Account
2	CustID	Int		Store ID of Customer
3	AccountNo	Varchar	50	Store Account Number
4	ODID	Int		Store ID of OverDraft Limit
5	WDID	Int		Store ID of WithDraw Limit
6	Balance	Decimal		Store amount of money left in account

4.3.3 Card

Card				
	Field name	Type	Max Length	Description
1	CardNo	Varchar	16	Store Card number
2	Status	Varchar	30	Store status of card: block, normal ...
3	AccountID	Int		Store Account ID
4	PIN	Varchar	6	Store the PIN code of Card
5	StartDate	Datetime		Date of created
6	ExpiredDate	Datetime		Date of expire.
7	Attempt	Int		Entered invalid PIN times of customer

4.3.4 OverDraft Limit

OverDraft				
	Field name	Type	Max Length	Description
1	ODID	Int		ID of overdraft limit
2	Value	Decimal		Value of overdraft limit that an account can overdraft

4.3.5 WithDraw Limit

OverDraft				
	Field name	Type	Max Length	Description
1	WDID	Int		ID of withdraw limit

2	Value	Decimal		Value of withdraw limit that an account can withdraw for a day.
---	-------	---------	--	---

4.3.6 ATM

ATM				
	Field name	Type	Max Length	Description
1	ATMID	Int		ID of ATM
2	Branch	Nvarchar	50	Branch name
3	Address	Nvarchar	100	Location of ATM

4.3.7 Money

Money				
	Field name	Type	Max Length	Description
1	MoneyID	Int		ID of money use in system
2	MoneyValue	Decimal		Value of kind of money
3	Address	Nvarchar	100	Location of ATM

4.3.8 Stock

Stock				
	Field name	Type	Max Length	Description
1	StockID	Int		ID of stock record
2	MoneyID	Int		Store ID of kind of money
3	ATMID	Int		Store ID of ATM
4	Quantity	Int		Store quantity of each kind of money in each ATM

4.3.9 LogType

LogType				
	Field name	Type	Max Length	Description
1	LogTypeID	Int		ID of log type use in system
2	Description	Nvarchar	100	Name or details of log type

4.3.10 Log

Log				
	Field name	Type	Max Length	Description
1	LogID	Int		ID of log record
2	LogTypeID	Int		ID of log type
3	ATMID	Int		ID of ATM
4	CardNo	Varchar	16	Number of ATM Card
5	LogDate	Datetime		Date when transaction happen.
6	Amount	Decimal		Amount of transaction
7	Details	Varchar	100	Description about transaction

4.3.11 Config

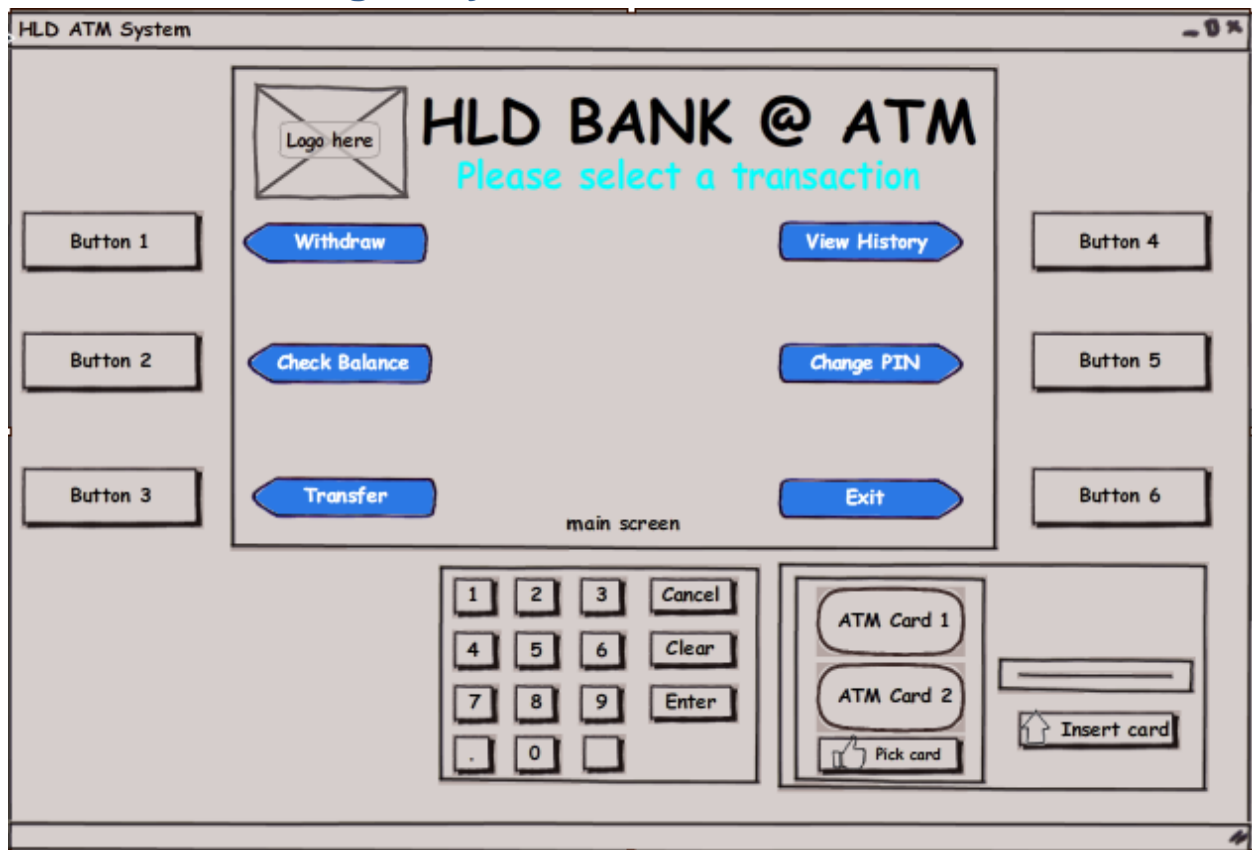
Config				
	Field name	Type	Max Length	Description
1	DateModified	Datetime		Date of modified
2	MinWithdraw	Decimal		Limit minimum of each withdrawal
3	MaxWithdraw	Decimal		Limit maximum of withdrawal for a day
4	NumPerPage	Int		Number of record will display per page in search results screen

5 Application Security

5.1 . Main functions 's security matrix

Usercase	Actor	Customer	ATM
Validation			x
Withdraw		x	x
CashTransfer		x	x
CheckBalance		x	x
ChangePIN		x	x
ViewHistory		x	x
Logging			x

6 Interface Design HuyVQ4



6.1 Layout – HuyVQ4

- A Main screen is aligned top center: use to display all the screen transaction to the customer
 - Bank logo on the top left of main screen
 - Bank Name on the top center of main screen
- There are six (6) fixed button (Button1 -> 6): 3 on the left and 3 on the right: different function with each screen transaction
- Calculator keyboards are under the main screen: button 0 to 9 to input number 0 to 9.
 - Cancel button: used to cancel any transaction.
 - Clear button: used to refresh the input text.
 - Enter button: used to submit the input or confirm customer action.

6.2 Themes

7 Details function design

7.1 Use case 01: Validation – HuyVQ4

7.1.1 Validate Card

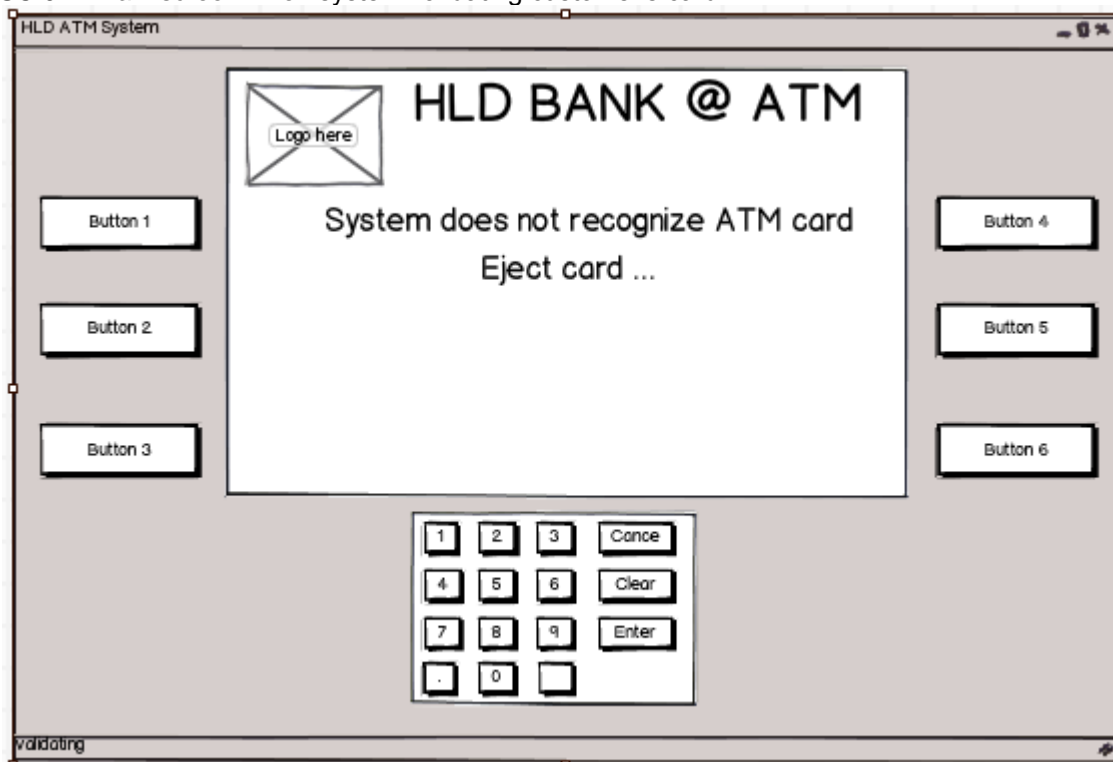
Name	Validate Card
Description	This use case allows ATM system to check the cards which is inputted by user is valid or not.
Actor	ATM System

Trigger	When user clicked on 'Insert Card' button at main screen.
Pre-condition	The card has been inputted into ATM machine.
Post-condition	If the card is valid then next step "Authenticaton" is activated, eject the card if it is invalid.




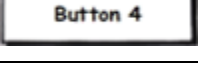
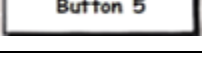
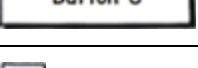













Screen Design & Data Description



SC 01 : Main screen when system validating customer's card



SC 02 : Screen when system does not recognize customer's card is a ATM card or customer insert a card this bank system does not support.

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen

Activities Flow

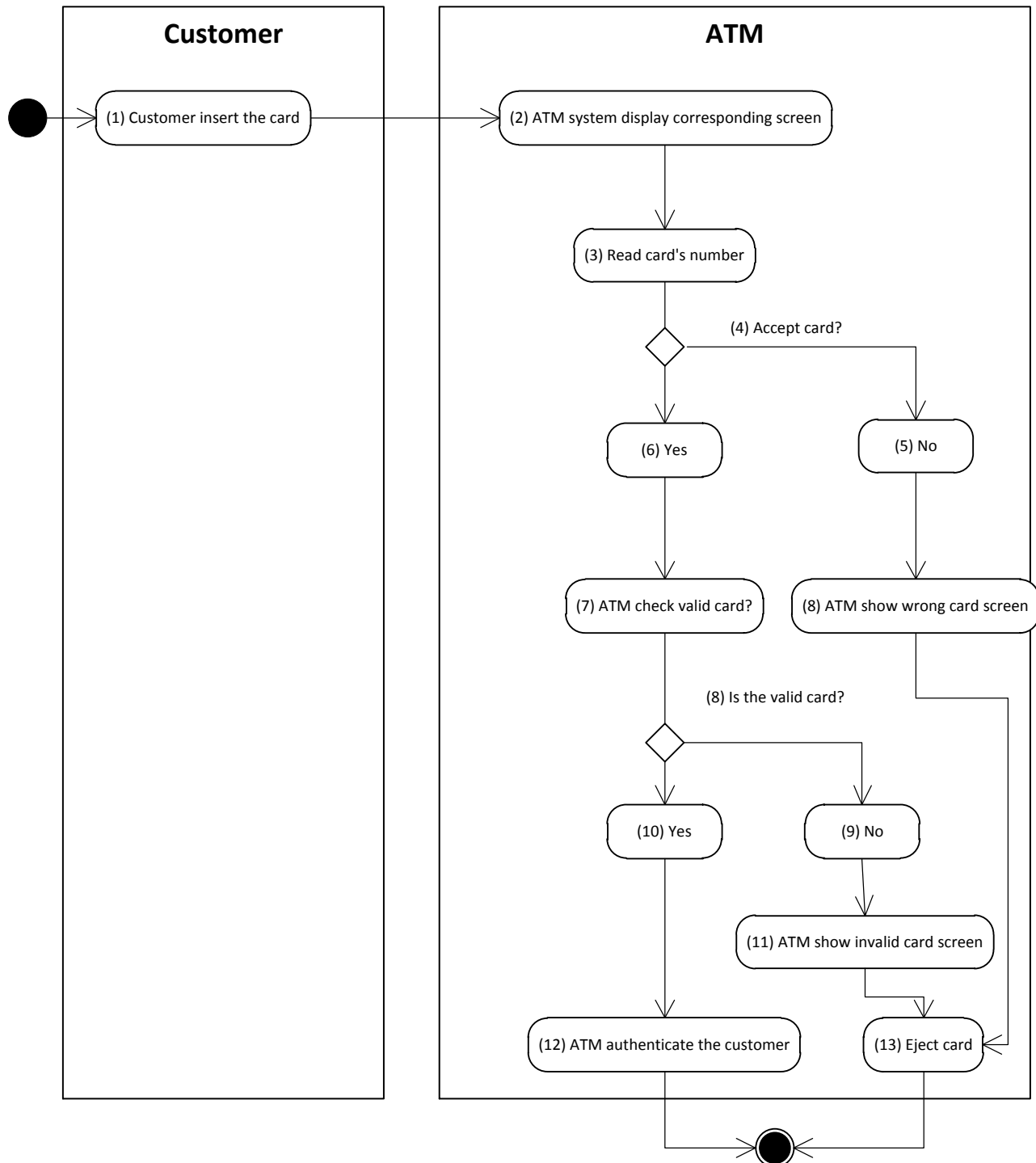
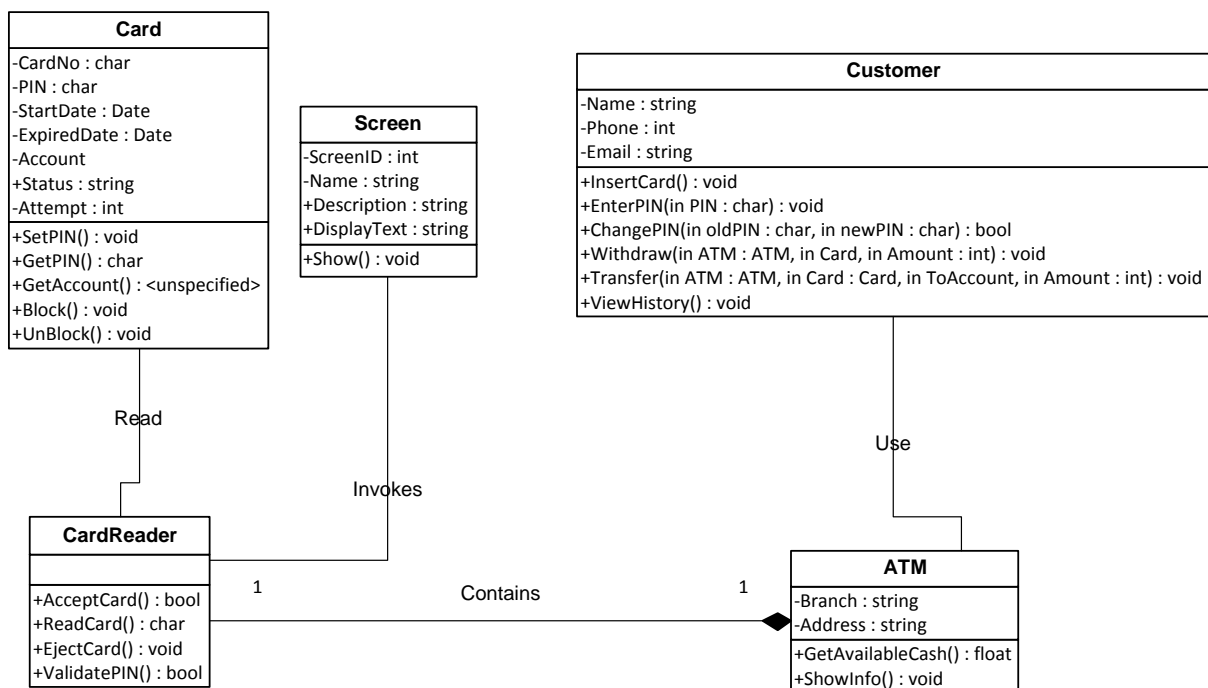


Figure 4.1:Activities Flow

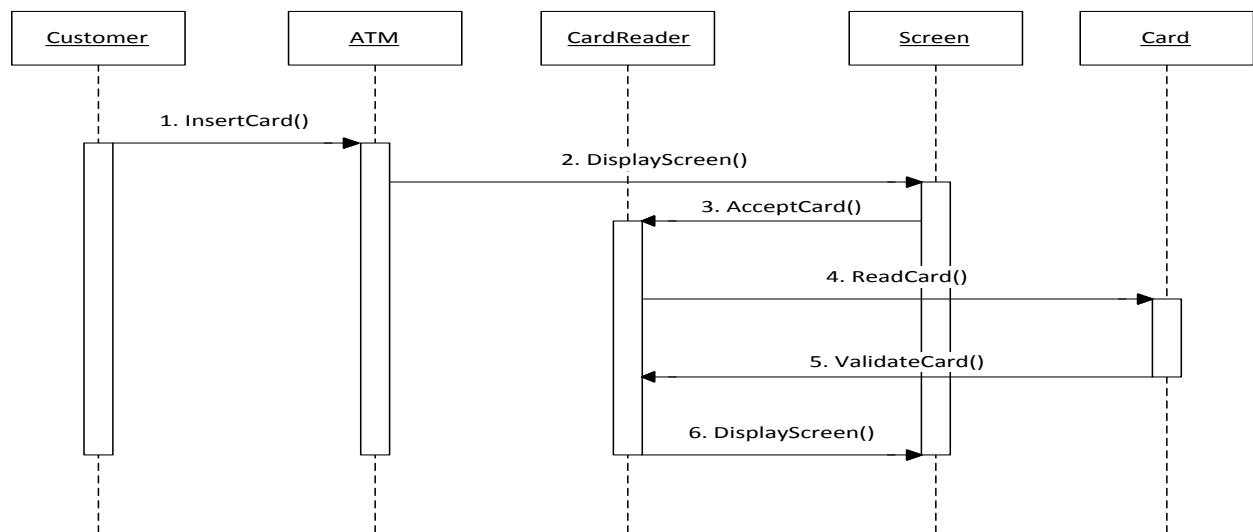
Detail Processing

Activity	BR Code	Description
(3)	BR01	Checking rules: <ul style="list-style-type: none"> ❖ IF <Card Reader> cannot read the card's number THEN <ul style="list-style-type: none"> ➢ Set <<ShowedScreen>> = [Wrong Card Screen]. ➢ Send request to eject card.
(7)	BR02	Checking rules: <ul style="list-style-type: none"> ❖ When user inserted the right card into ATM <ul style="list-style-type: none"> ➢ Get card information from database with card number like the number read from the card was inserted by customer. ❖ IF Card number does not match with any card number in database THEN <ul style="list-style-type: none"> ➢ Set <<ShowedScreen>> = [Invalid Card Screen]. ➢ Send request to eject card.

Class diagram



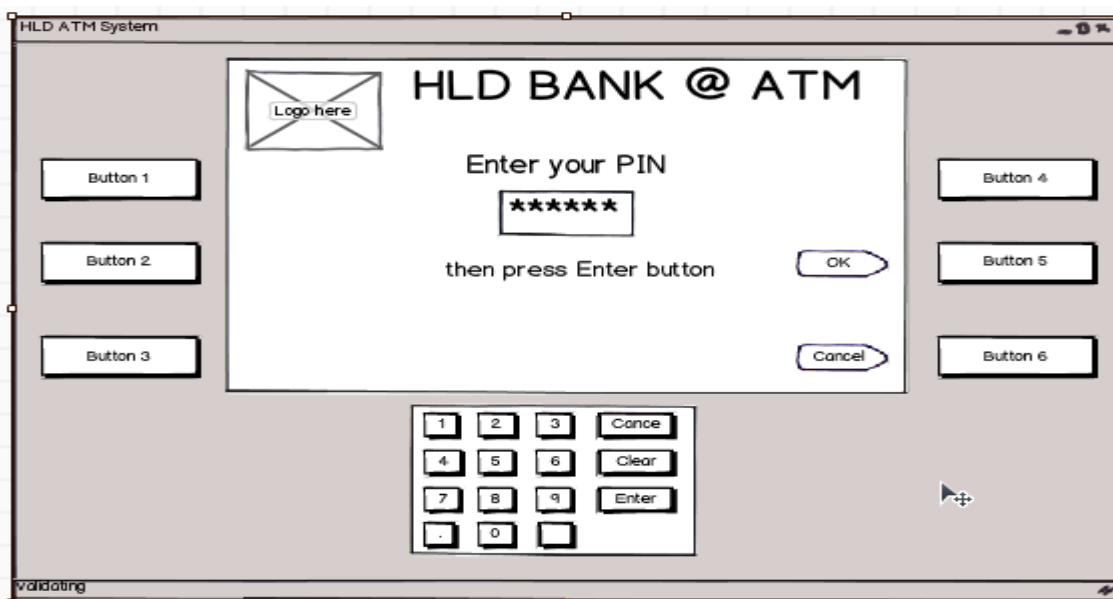
Sequence diagram



7.1.2 Authentication





Name	Authentication
Description	This use case allows ATM system to check the PIN which is inputted by customer is valid or not.
Actor	ATM System
Trigger	When user clicked on 'Enter' button or 'Submit'button at [Input PIN] screen.
Pre-condition	The card has been inputted into ATM machine.
Post-condition	Customer was authenticated successfully, ATM system display the select transaction screen.

Screen Design & Data Description



SC 01 : Screen when customer enter their PIN

Item	Type	Description
	button	Submit customer PIN
	button	Cancel function, return to pre-screen
	button	Cancel function, return to pre-screen
	button	Clear text at text box
	button	Submit customer PIN
	Textbox	Input customer PIN
Others button	Button	Do nothing at this screen
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer

Item	Type	Description
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer
	button	Accept input of customer

SC 02 : Screen when customer enter WRONG PIN, system requires customer to re-enter PIN.



SC 03 : Screen when system validating customer PIN.

Item	Type	Description
All button	Button	Do nothing at this screen

Activities Flow

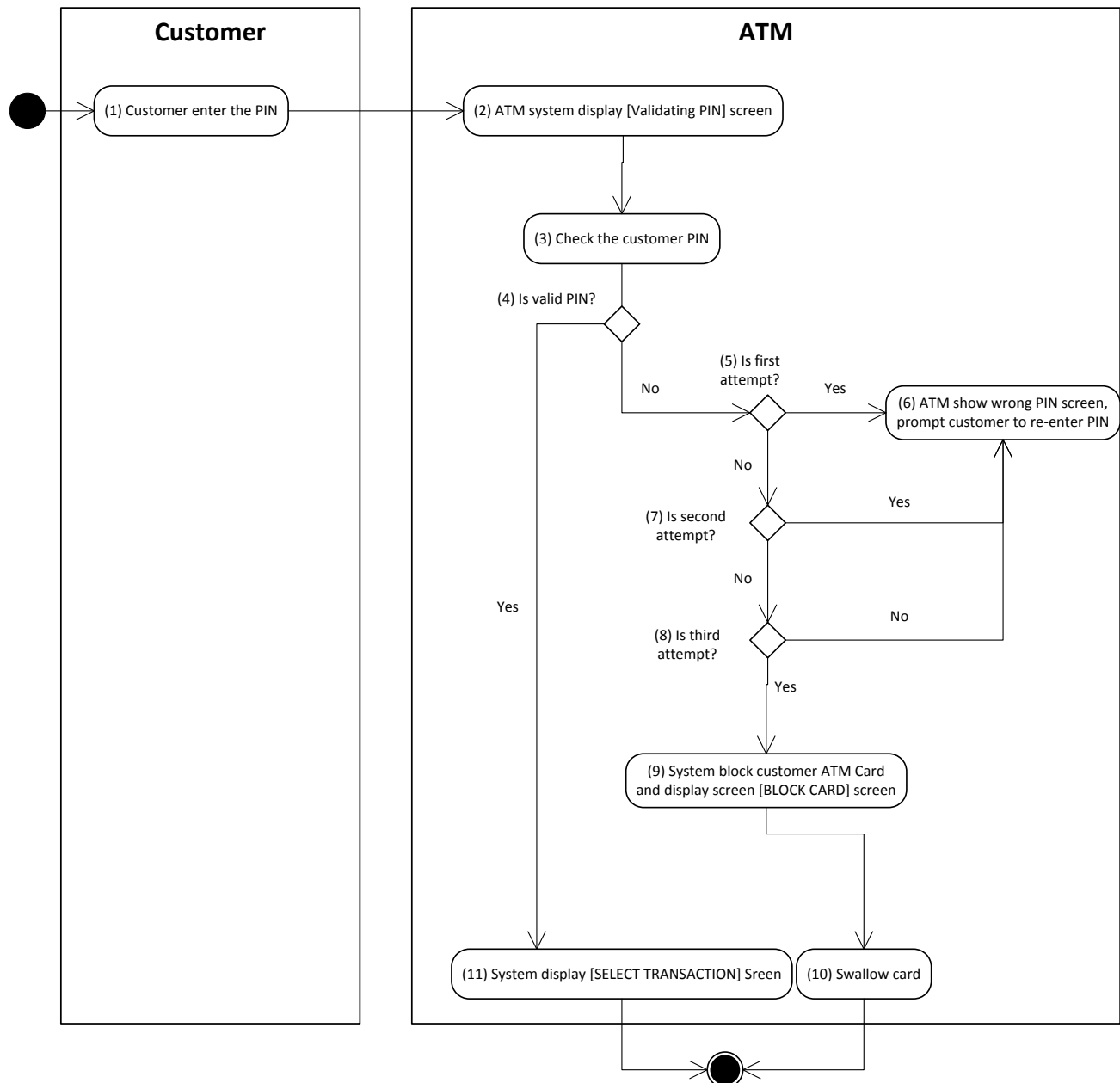


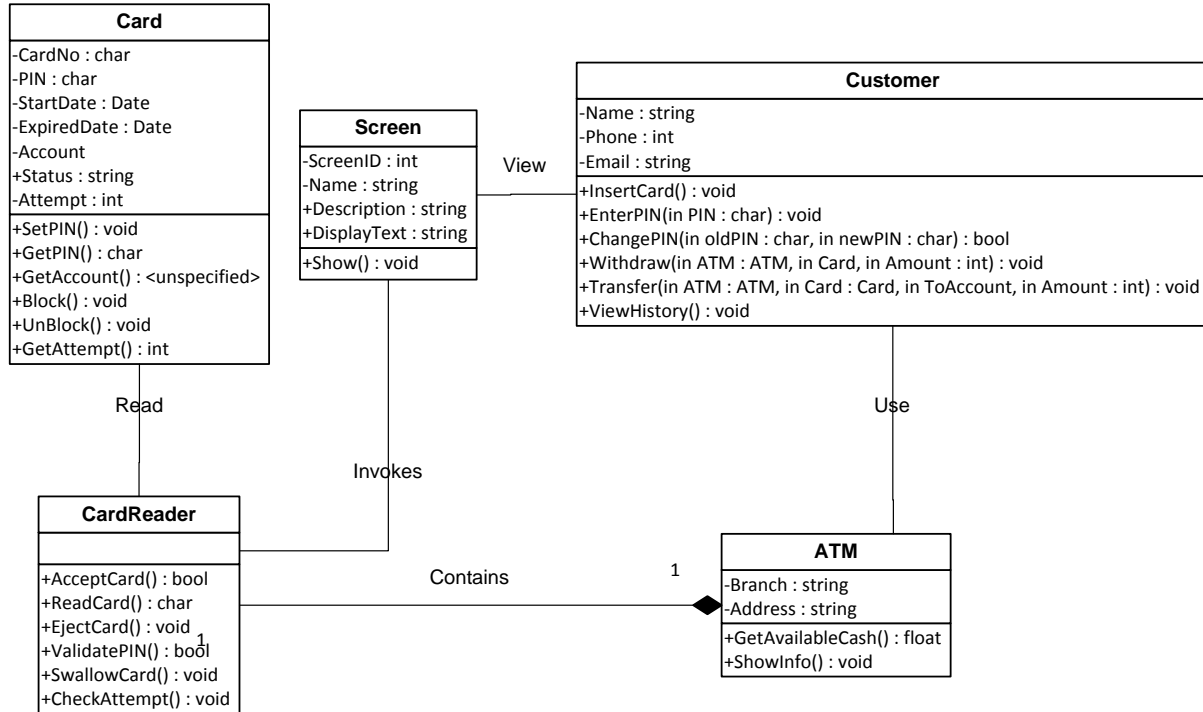
Figure 4.2:Activities flow for authentication

Detail Processing

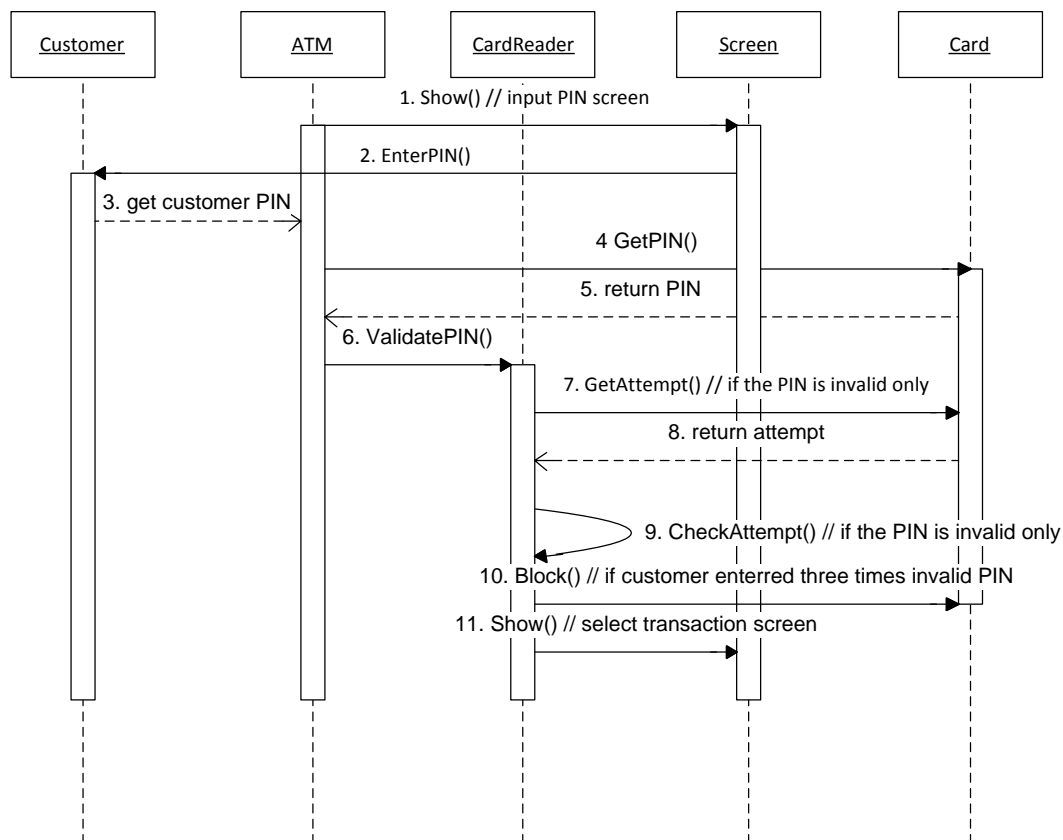
Activity	BR Code	Description
(4)	BR01	Checking rules: <ul style="list-style-type: none"> ❖ Check PIN <ul style="list-style-type: none"> ➢ Get the PIN of Customer Card from Database. ➢ Compare the PIN get from database with the PIN customer just entered. ❖ IF the PIN customer inputted does not match with the PIN in database of customer Card THEN <ul style="list-style-type: none"> ➢ Set <<ShowedScreen>> = [Wrong PIN Screen]. ➢ Prompt customer to re-enter PIN. ❖ IF the customer has inputted three times wrong PIN THEN

- Set <<ShowedScreen>> = [Block Card Screen].
- Set <Status> of Card = "Block".
- Swallow card.

Class diagram



Sequence diagram



7.2 Use case 02: Withdraw Money - LuyenNV

7.2.1 Withdraw

Name	Withdraw money
Description	This use case allows customer to withdraw moneys.
Actor	Customers
Trigger	When user clicking on 'Withdraw' button in the screen.
Pre-condition	After Validation success, customer input amount money which they want to withdraw.
Post-condition	Receive money, write log into system, decide print receipt.

Screen Design & Data Description

HLD ATM System

Logo here

HLD BANK @ ATM

Choose amount below or enter your amount you want

500.000	1.000.000
2.000.000	2.500.000
3.000.000	Enter other

1

2

3

Cancel

4

5

6

Clear

7

8

9

Enter

.

0

Button 1

Button 2

Button 3

Button 4

Button 5

Button 6

Main screen

created with Balsamiq Mockups - www.balsamiq.com

SC01 : Main screen when customer access "Withdraw"

HLD ATM System

Logo here

HLD BANK @ ATM

Enter your amount you want to withdrawal below, number you enter have to divisible to 50.000

OK

Cancel

1

2

3

Cancel

4

5

6

Clear

7

8

9

Enter

.

0

Button 1

Button 2

Button 3

Button 4

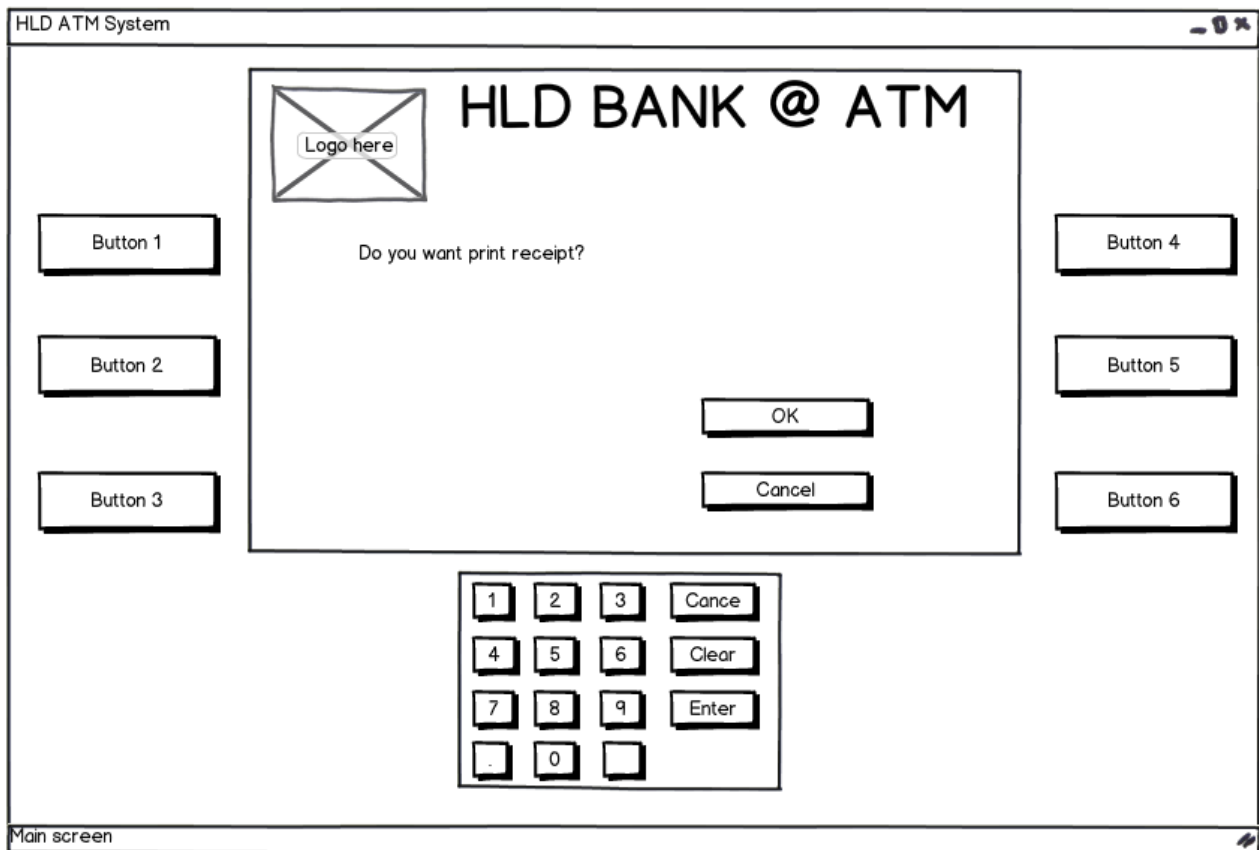
Button 5

Button 6

Main screen




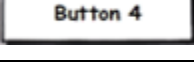
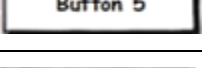
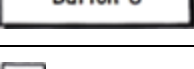


created with Balsamiq Mockups - www.balsamiq.com












SC02: Screen display when customer choose “Enter other” from SC01 screen.



created with Balsamiq Mockups - www.balsamiq.com

SC03: Screen confirm “Print receipt” when customer choose “OK” from SC 01 screen or “OK” from SC 02 screen

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Function “OK”
	button	Function “Cancel”
	button	Number “1” SC03
	button	Number “2” SC03

Item	Type	Description
	button	Number “3” SC03
	button	Number “4” SC03
	button	Number “5” SC03
	button	Number “6”
	button	Number “7”
	button	Number “8”
	button	Number “9”
	button	Number “0”
	button	Function “Cancel”
	button	Clear text input
	button	Function “OK”

Activities Flow

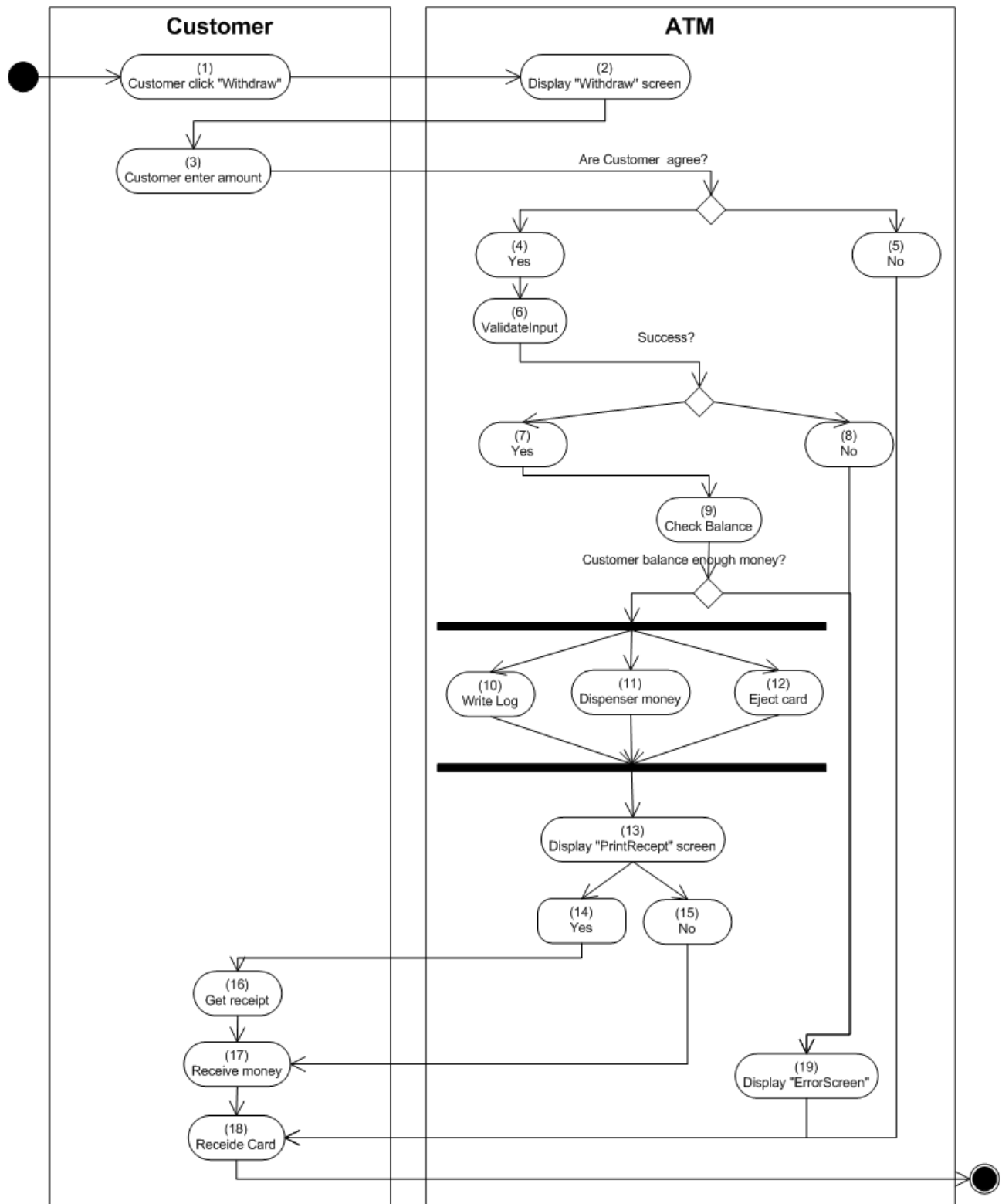


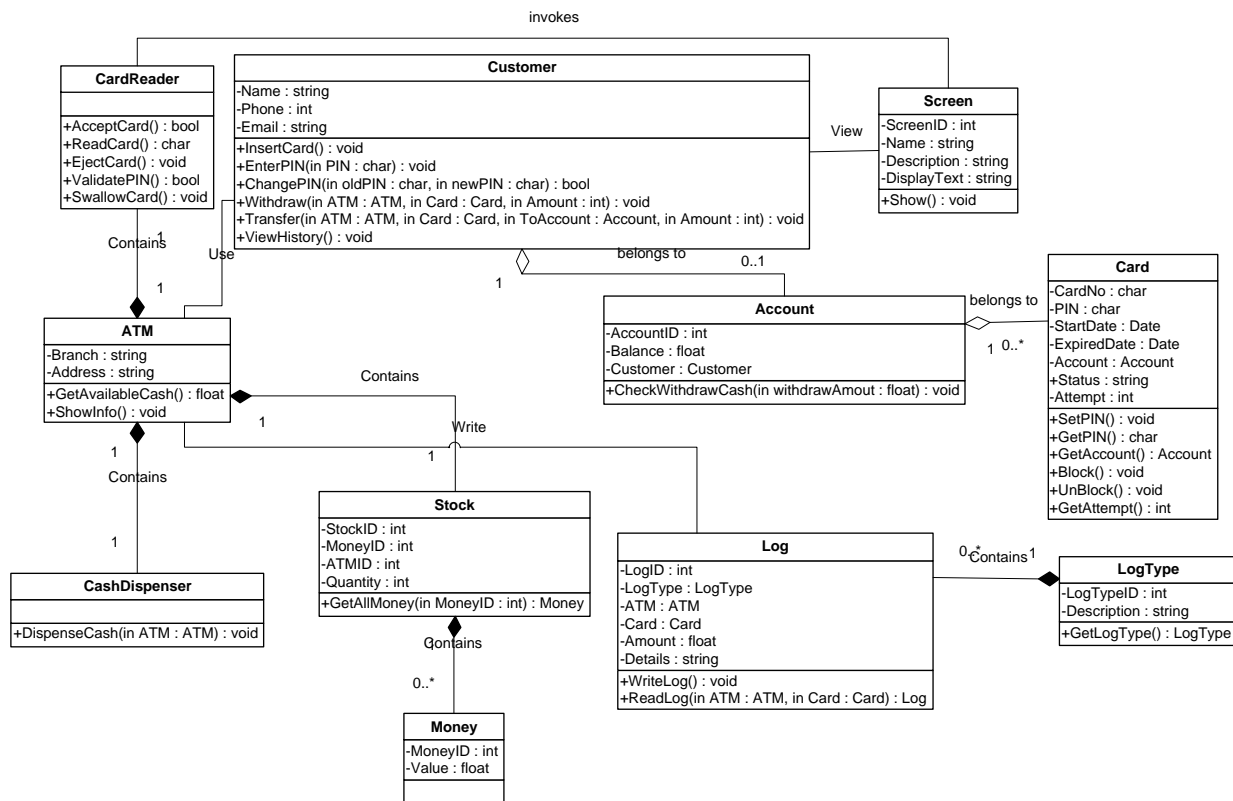
Figure 4.3:Activities flow of withdraw

Detail Processing

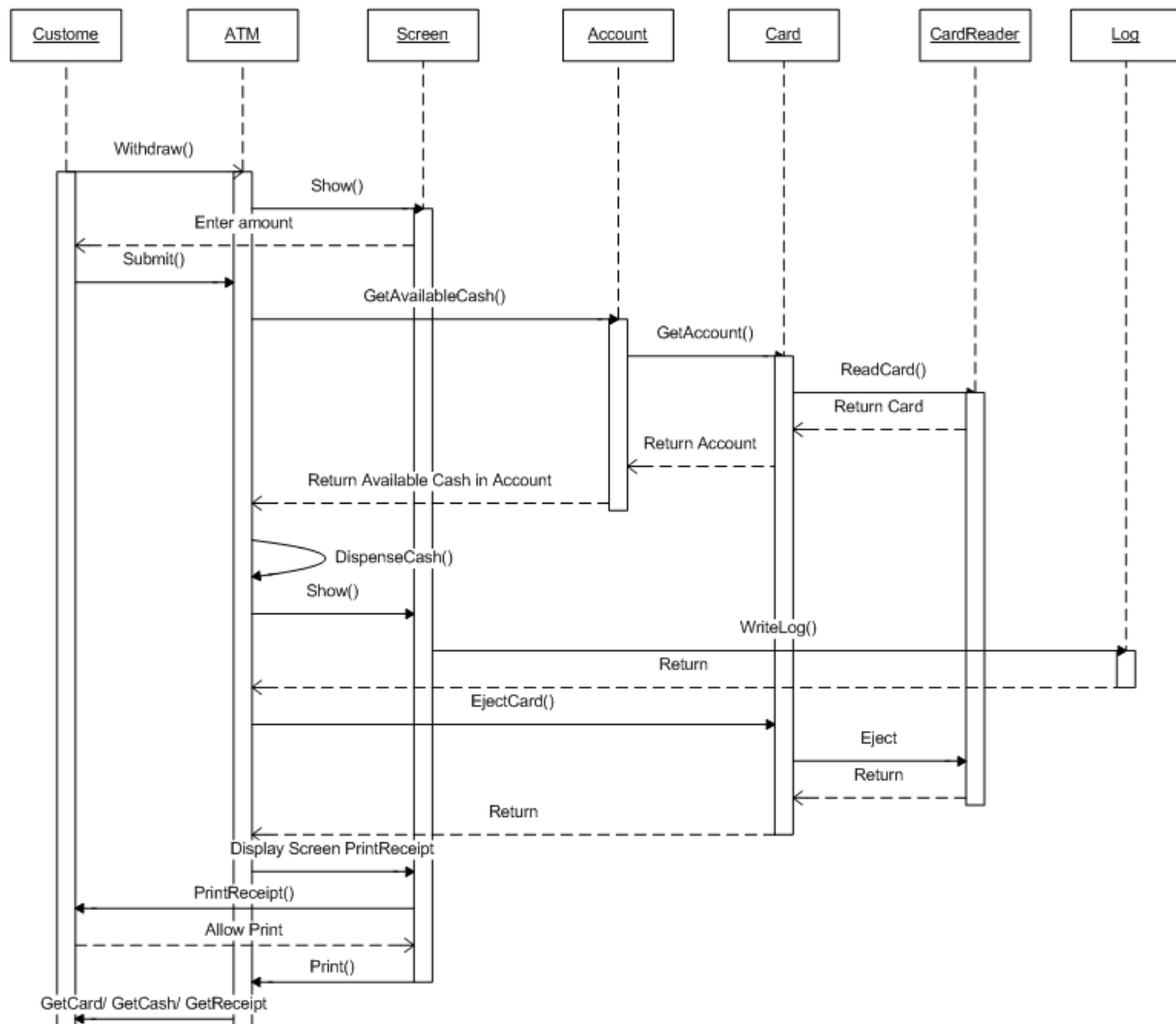
Activity	BR Code	Description
(6)	BR01	ValidateInput: <ul style="list-style-type: none"> ❖ System check balance:

		<ul style="list-style-type: none"> ➤ IF enterCash > MinValue ➤ OR enterCash <MaxValue ➤ OR enterCash mod 50.000 <> 0 THEN <ul style="list-style-type: none"> ○ Set <<ShowedScreen>> = [Withdraw Failed Screen] ○ Return FALSE
(9)	BR02	<p>Check balance:</p> <ul style="list-style-type: none"> ➤ IF enterCash < AccountBalance THEN <ul style="list-style-type: none"> ○ Set <<Account Balance>> = <<Account Balance>> - enterCash ○ Write Log. ➤ ELSE <ul style="list-style-type: none"> ○ Set <<ShowedScreen>> = [Withdraw Failed Screen]
(11)	BR03	<p>Dispenser money:</p> <ul style="list-style-type: none"> ❖ Calculate enterCash customer enter and MoneyType and Value, number of this MoneyType in this ATM, Return Cash to customer

Class diagram



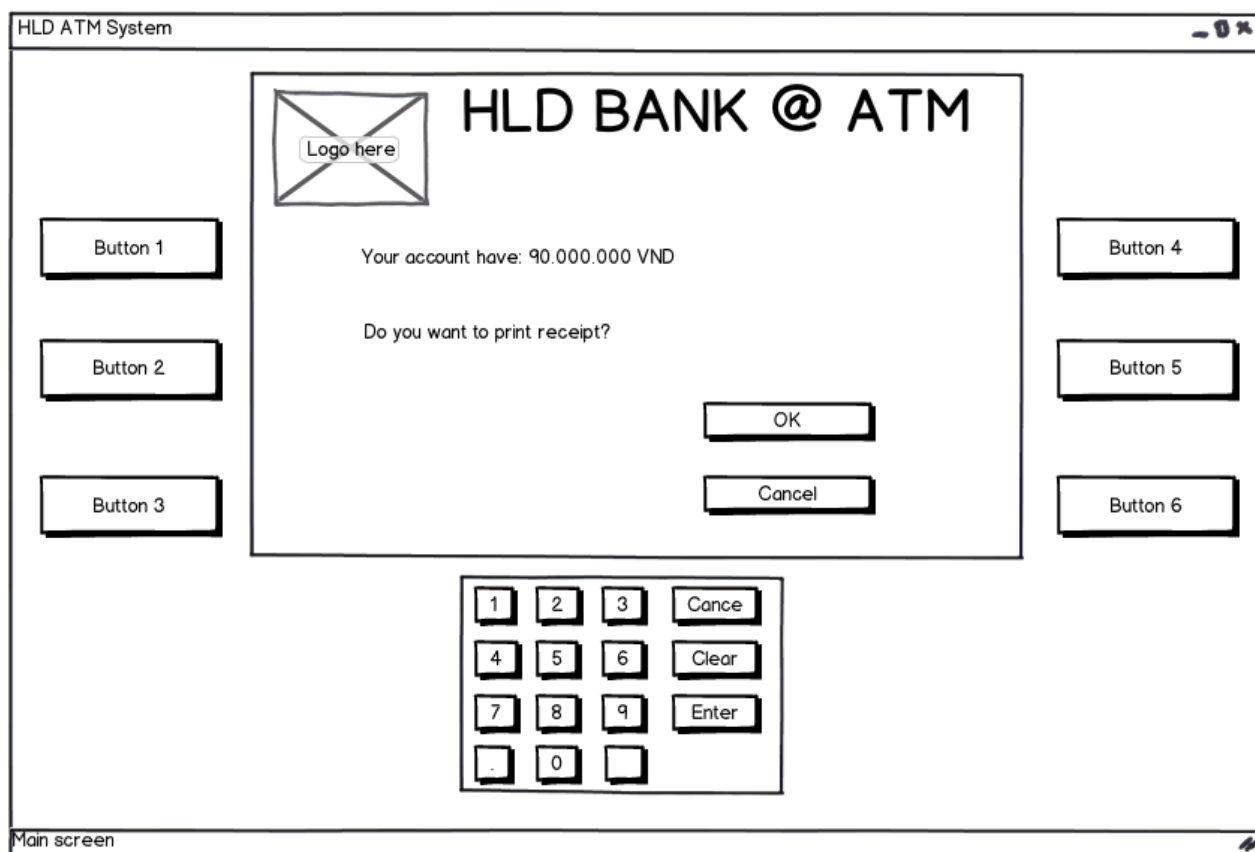
Sequence diagram



7.3 Use case 03: Check Balance – LuyenNV

Name	CheckBalance
Description	This use case allows Customer to check their Balance.
Actor	Customer
Trigger	When Customer click “Check balance” button in MainScreen
Pre-condition	Customer had Validation to ATM.
Post-condition	ATM system display balance of Customer.









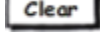
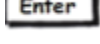
Screen Design & Data Description



created with Balsamiq Mockups - www.balsamiq.com

SC 01 : Display Balance of customer and confirm PrintReceipt

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do not print receipt
	button	Do nothing at this screen
	button	Allow print receipt

Activities Flow

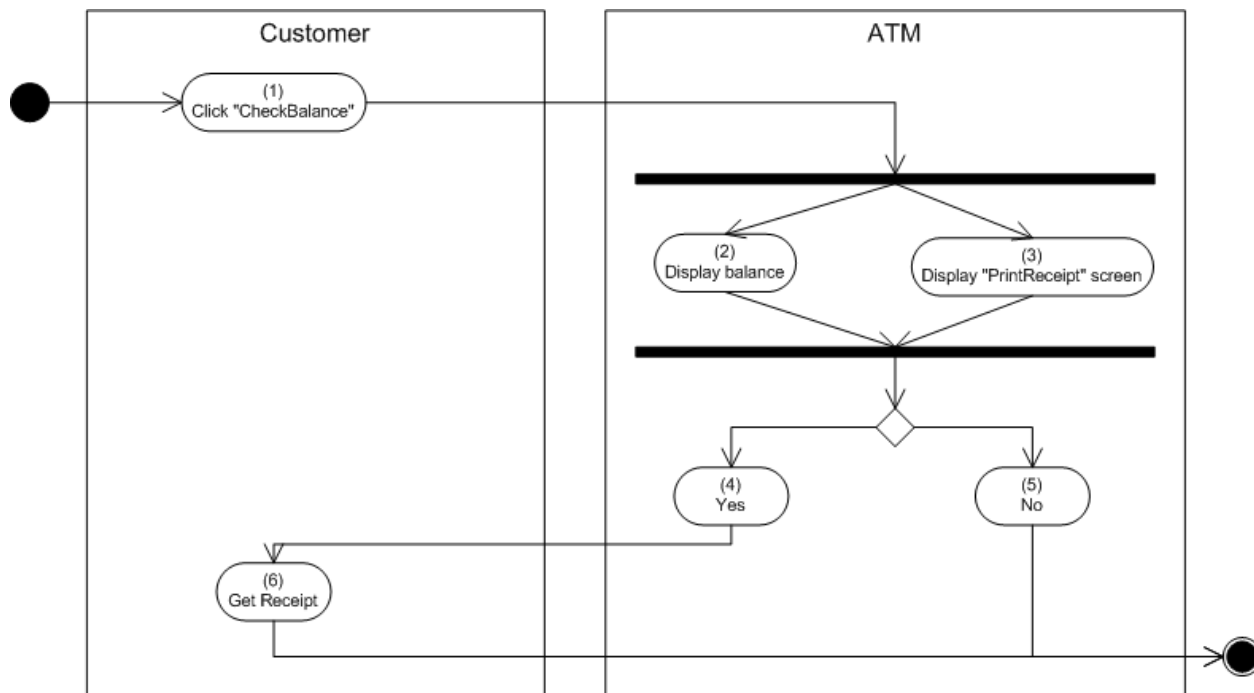


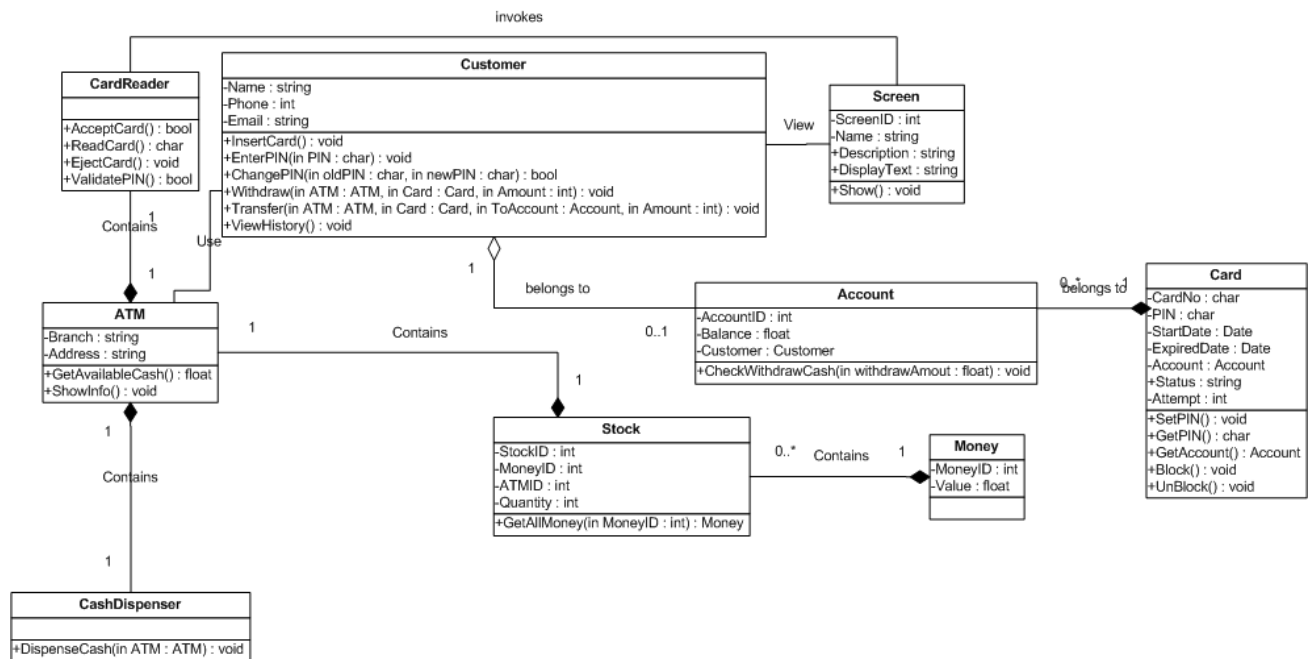
Figure 4.4:Activities flow of check balance

Detail Processing

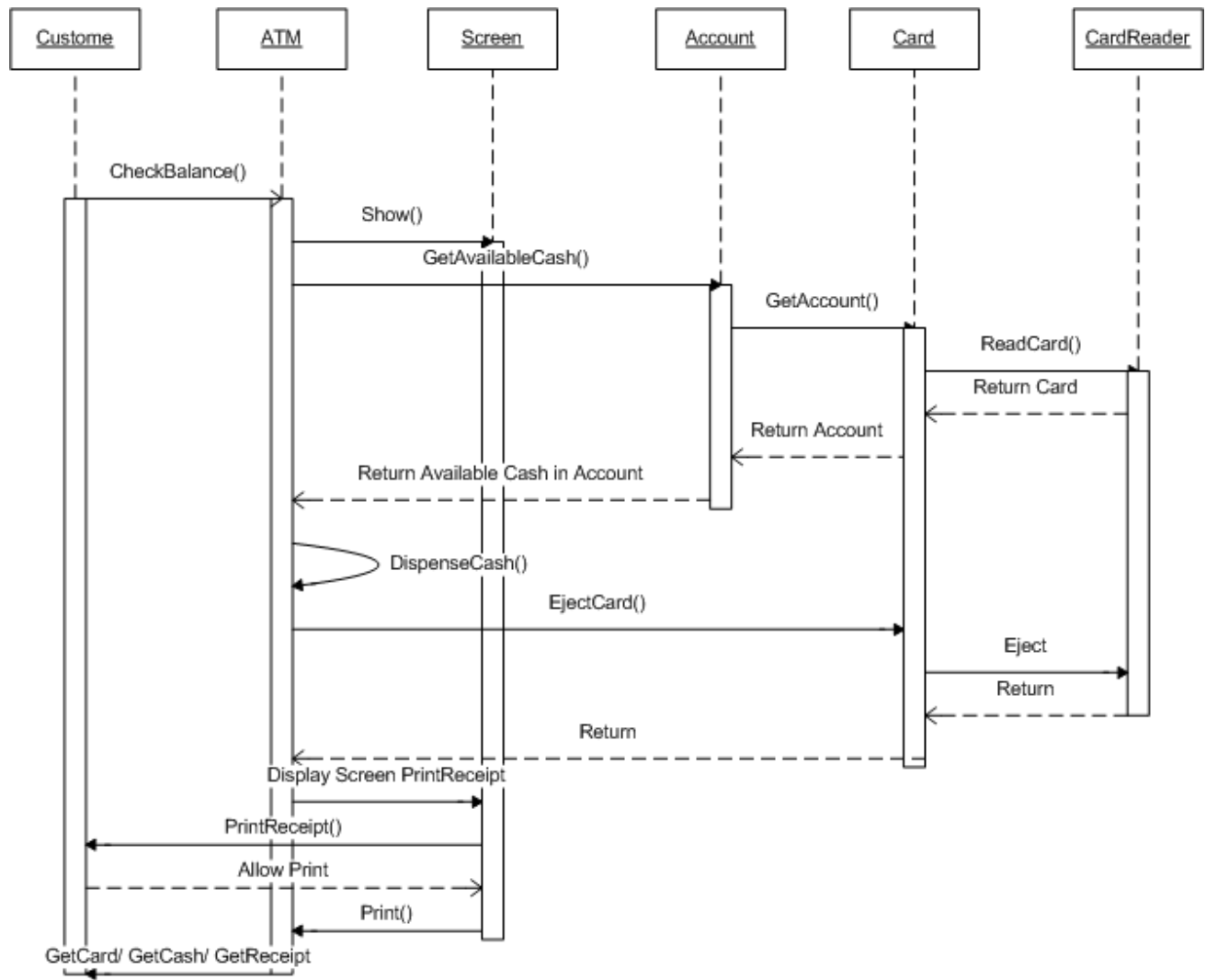
Activity	BR Code	Description
(2)	BR01	Display balance:

➤ Get balance of customer from database and display to screen.

Class diagram



Sequence diagram

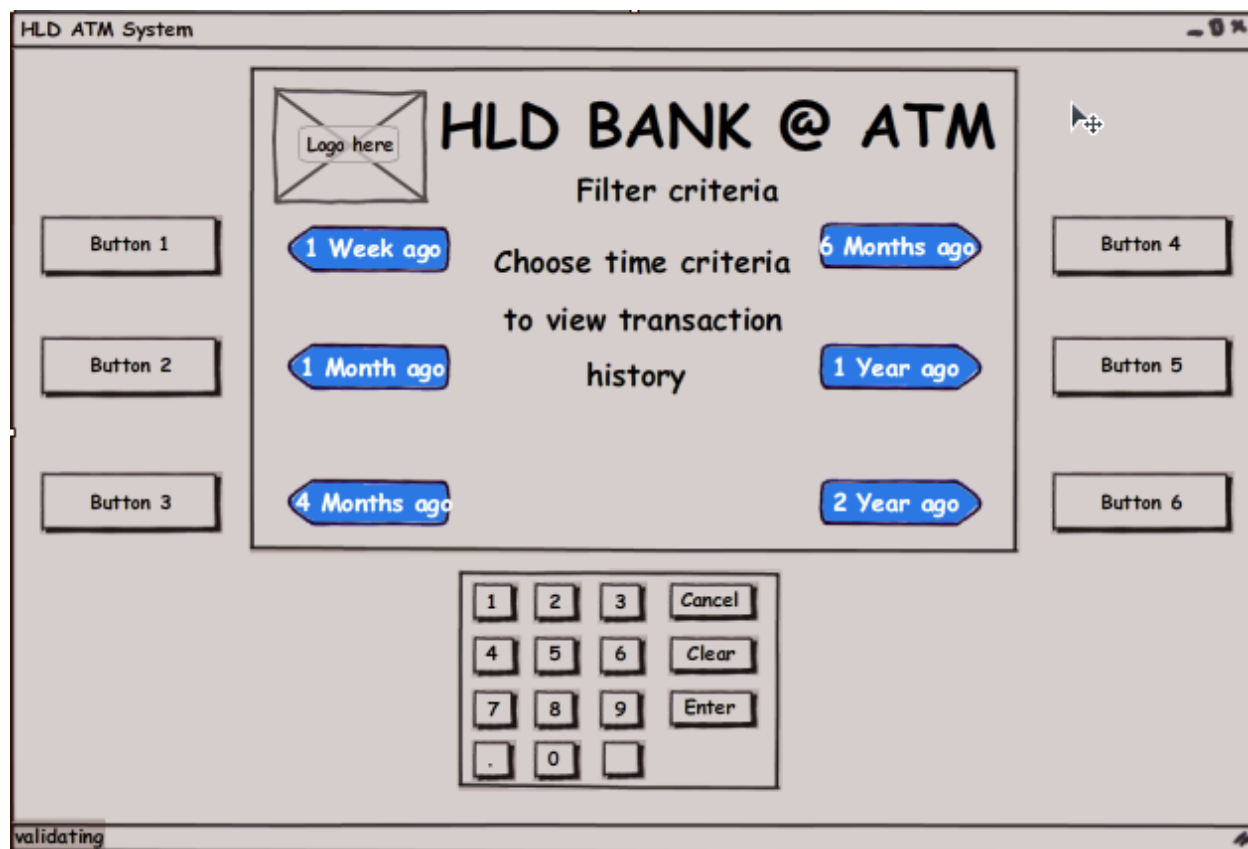


7.4 Use case 04: View History – HuyVQ4




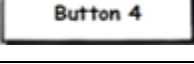
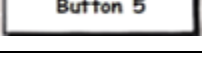
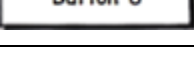
7.4.1 View History

Name	View history
Description	This use case allows customer to view all transaction has been done.
Actor	Authenticated Customer
Trigger	When user clicked on 'View History' button at [Select Transaction] screen.
Pre-condition	Customer was authenticated succesfully.
Post-condition	All transaction has been done by the customer will show.

Screen Design & Data Description



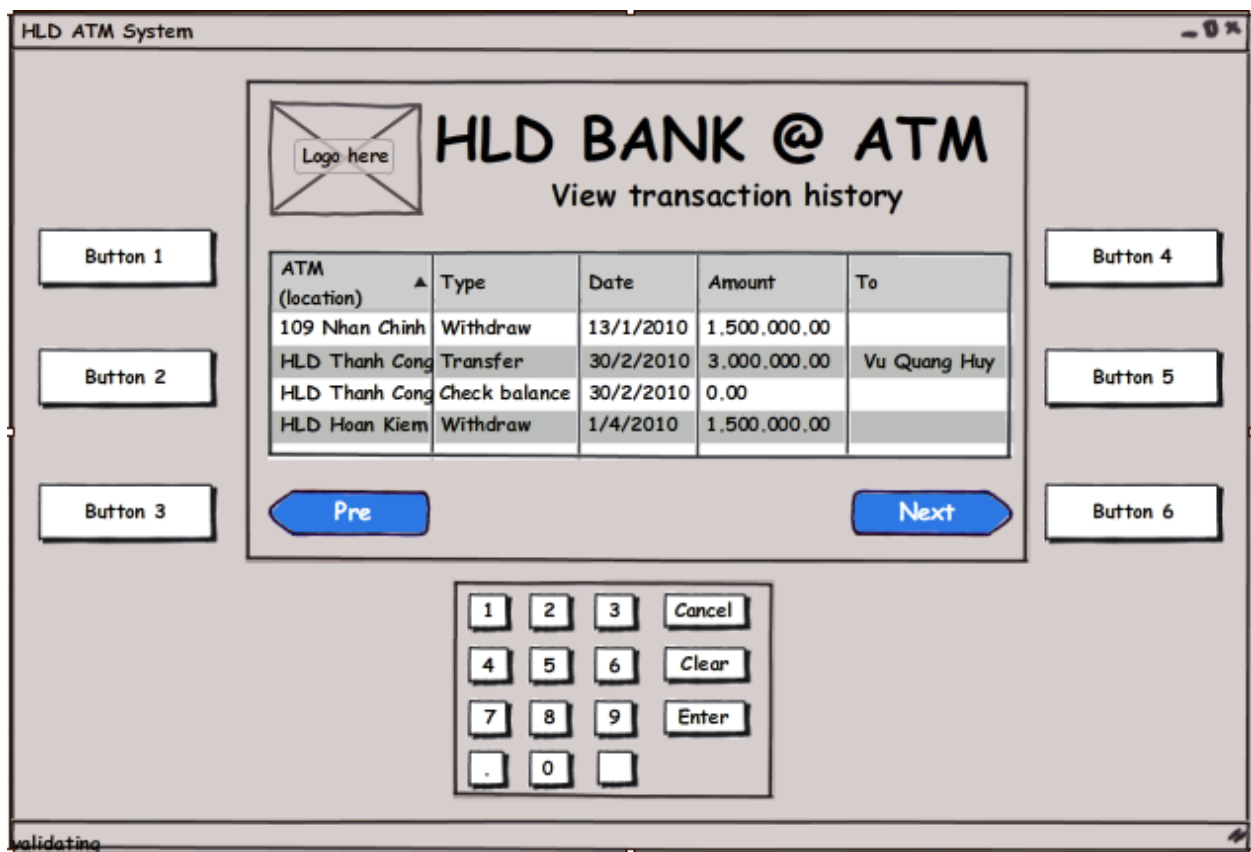
SC 01 : Screen list the time criterias to customer to choose

Item	Type	Description
	button	Set TimeCriteria = Current Date – 7 day
	button	Set TimeCriteria = Current Date – 30 day
	button	Set TimeCriteria = Current Date – 120 day
	button	Set TimeCriteria = Current Date – 180 day
	button	Set TimeCriteria = Current Date – 365 day
	button	Set TimeCriteria = Current Date – 700 day
Others button	Button	Do nothing at this screen





SC 02 : Waitted screen when system is searching.

Item	Type	Description
All button	Button	Do nothing at this screen



SC 03 : Screen show search results paged.

Item	Type	Description
	button	Show previous search results paged.
	button	Show next search results paged.
Others button	Button	Do nothing at this screen

Activities Flow

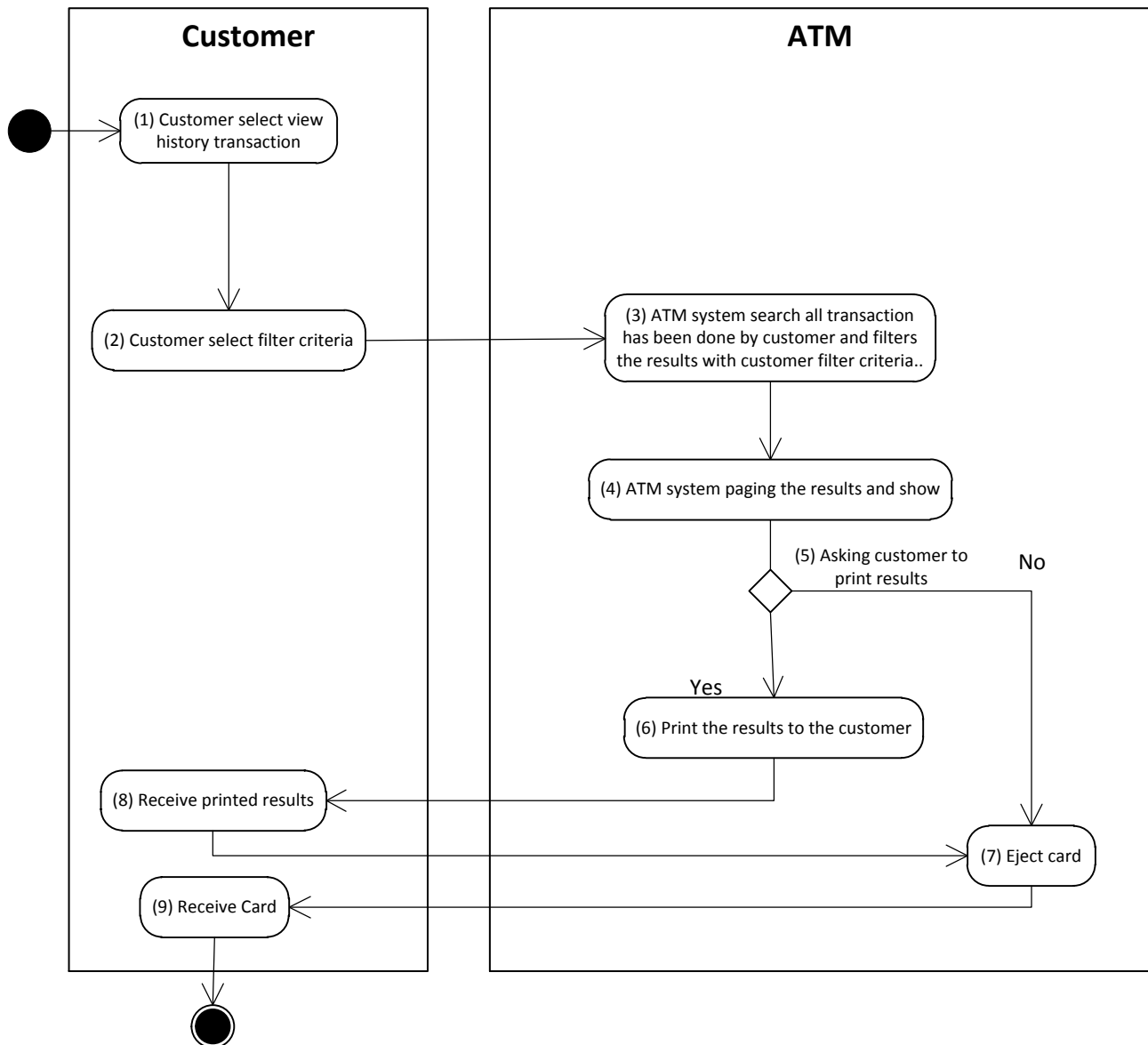


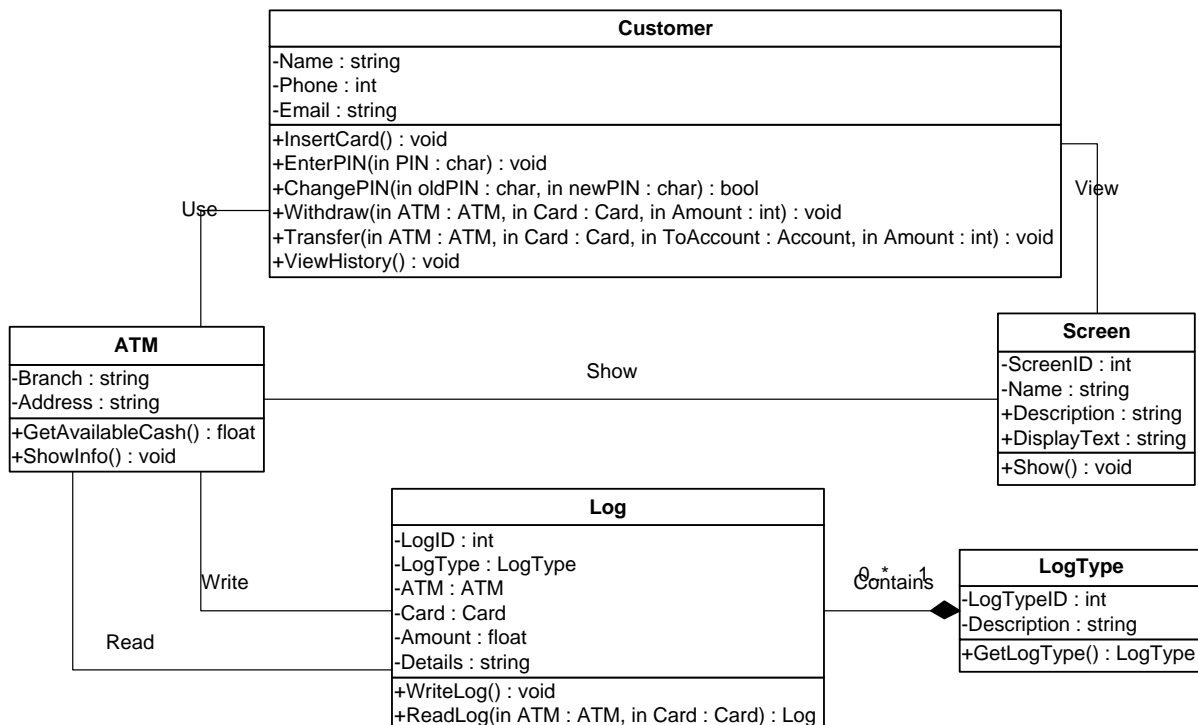
Figure 4.5:Activities flow of check balance

Detail Processing

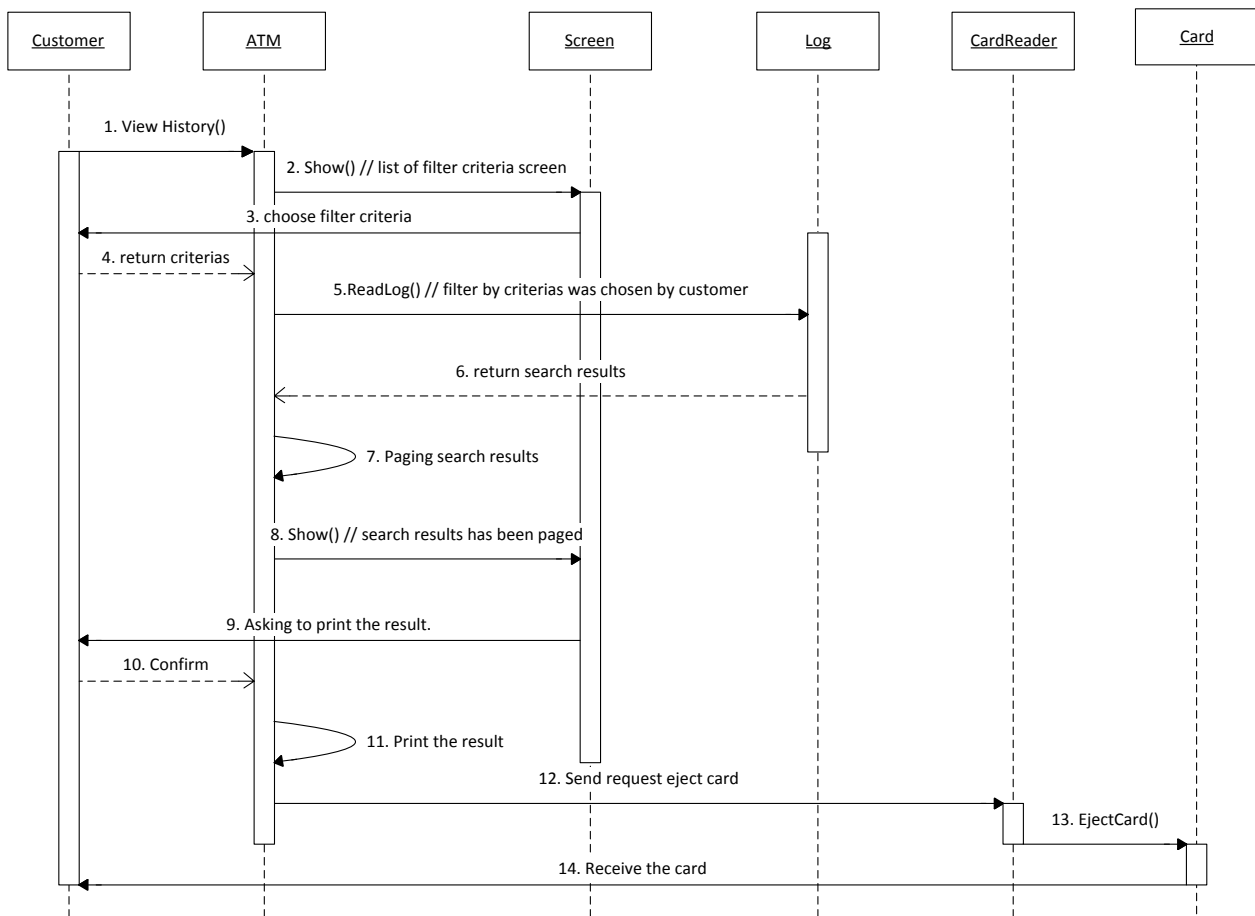
Activity	BR Code	Description
----------	---------	-------------

(3)	BR01	Checking rules: <ul style="list-style-type: none"> ❖ Search transaction/ log <ul style="list-style-type: none"> ➤ Set <<Criteria>> = [Filter Criteria] the customer has chosen. ➤ System search from database all transaction/ log with: <ul style="list-style-type: none"> ▪ <<LogStatus>> = [Successful] ▪ <<LogDate>> = <<Criteria.Date>> ▪ <<LogType>> in ([Withdraw], [Transfer], [CheckBalance], [ChangePIN])
(4)	BR02	Paging rules <ul style="list-style-type: none"> ❖ Paging <ul style="list-style-type: none"> ➤ Set <<NumPerPage>> = [Number records per page] in system configuration. ➤ Paging the search results follow <<NumPerPage>>

Class diagram



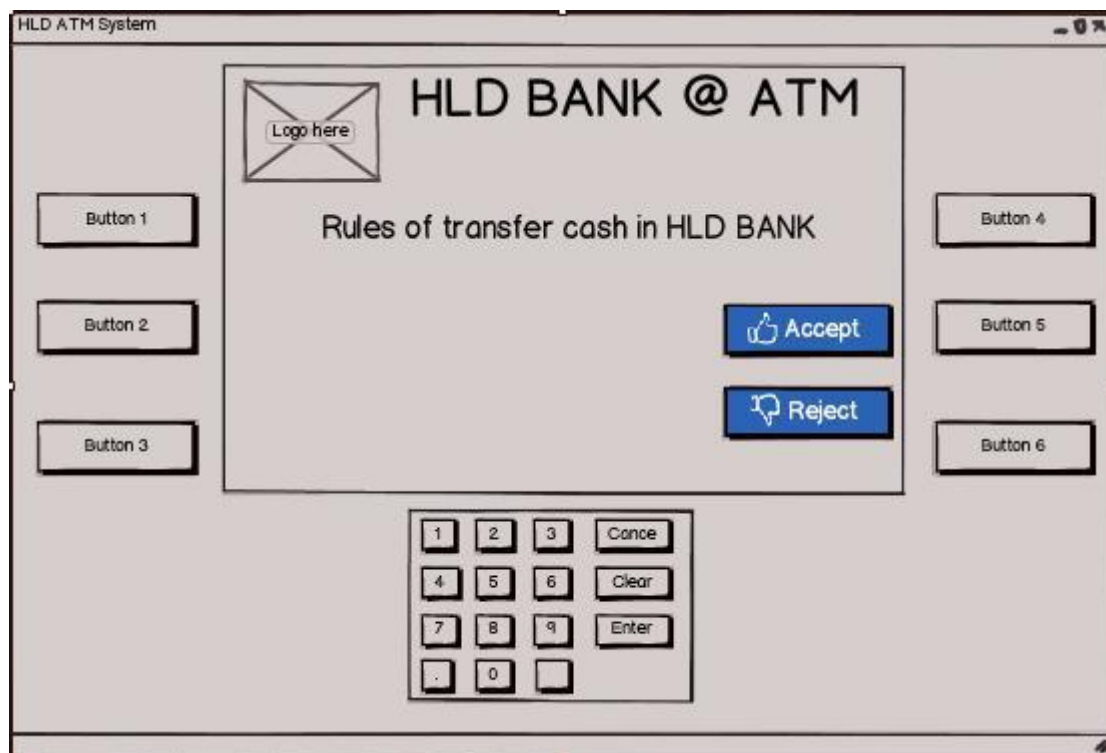
Sequence diagram





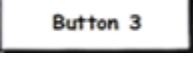
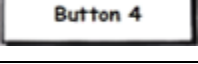
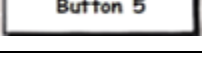
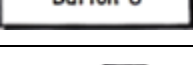




7.5 Use case 05: Cash Transfer – DucNA9

Name	Cash Transfer
Description	This use case allows customer transfer cash from his/her account to another account
Actor	Customer
Trigger	When user clicked on 'Cash Transfer' button at main screen.
Pre-condition	After validation success, customer enter account and amount of cash which want to transfer.
Post-condition	Write log into system, decide print receipt (Yes/No)

Screen Design & Data Description



Screen 01: Main screen when require customer enter account want to transfer cash, show rules of transfer


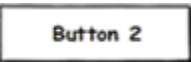








Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept rules of transfer cash in ATM
	button	Reject and stop transaction
	button	Do nothing at this screen
	button	Stop transaction
	button	Do nothing at this screen
	button	Do nothing at this screen

Screen 02: Display account transfer and receive account

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept input of customer
	button	Stop transaction
	button	Clear text in textbox
	button	Submit receive account ID which want to transfer

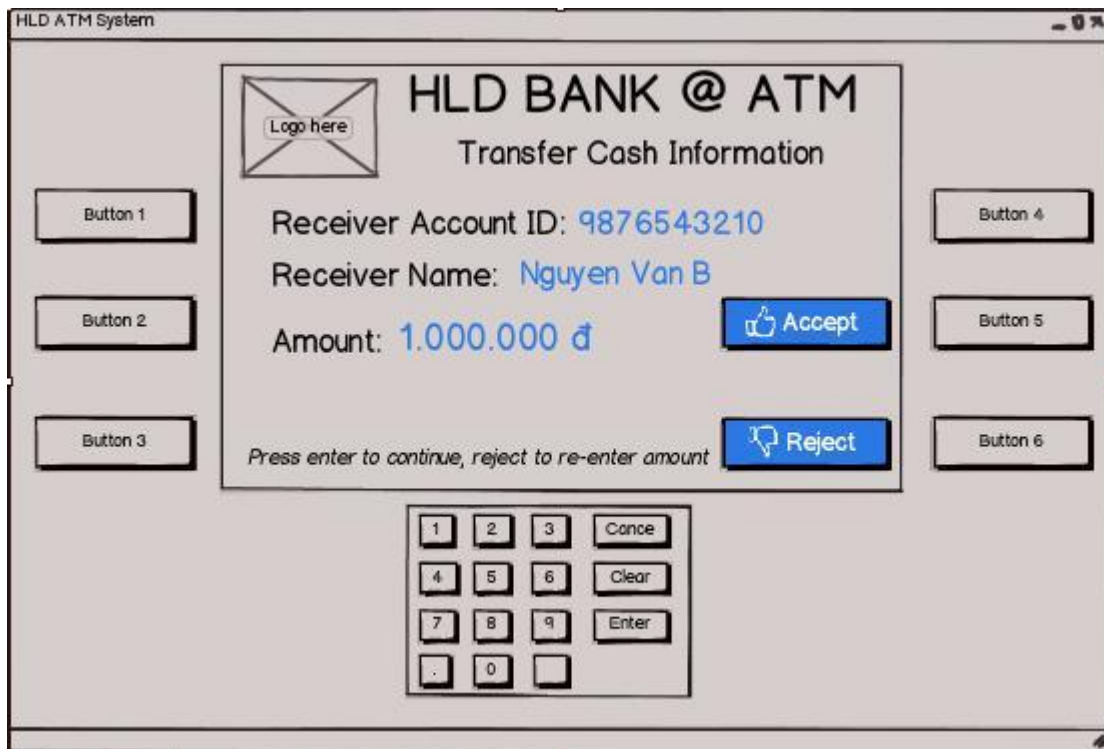


Screen 03: Display information about receive account




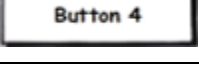
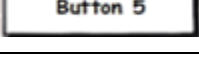
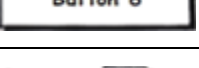
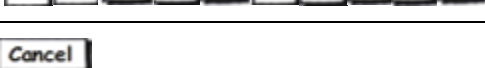



Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Continue transfer cash
	button	Back to enter receive account ID
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen

Screen 04: Display allow customer enter amount which want to transfer

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept input of customer
	button	Do nothing at this screen
	button	Clear text in textbox
	button	Submit amount and continue



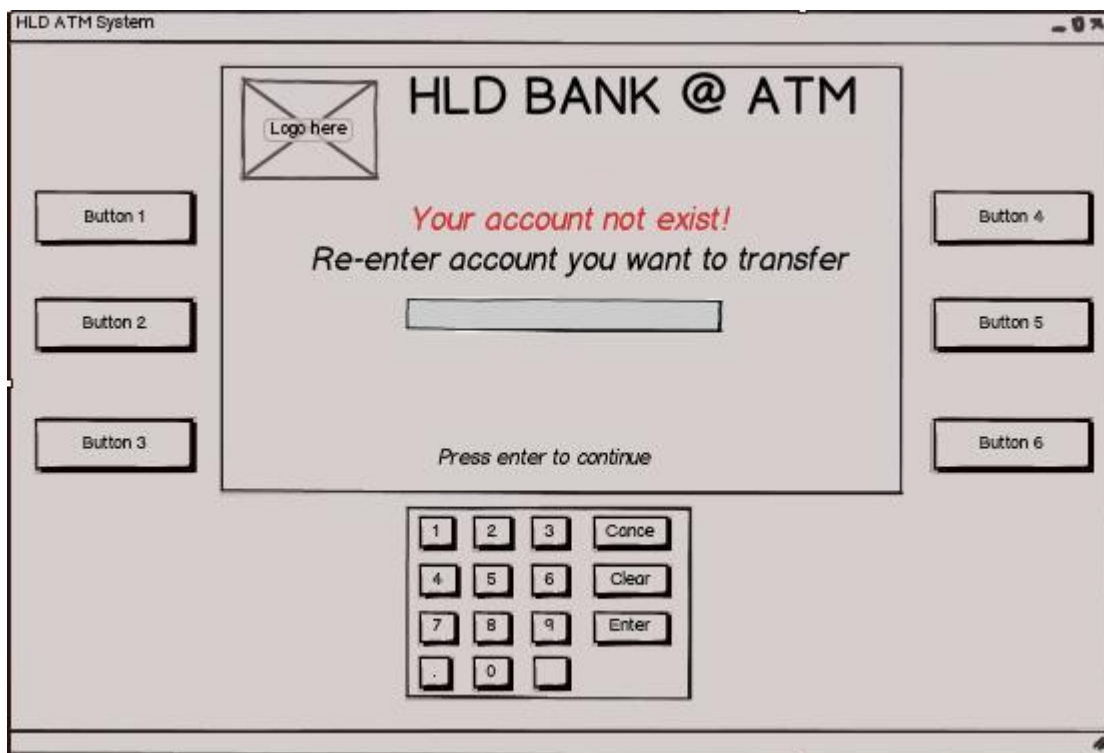
Screen 05: Display screen allow customer confirm transfer cash

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept to transfer cash
	button	Reject to transfer cash and back to previous screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen




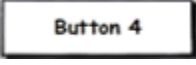

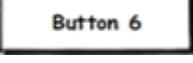






Screen 06: Display screen allow customer choose receive receipt or not

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept to receive receipt
	button	Reject to receive receipt
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen


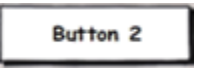
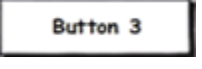
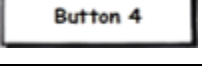
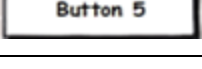
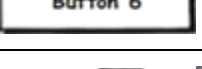






Screen 07: Display screen to notice customer that receive account not exist and re-enter account to transfer

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept input of customer
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Continue transaction



Screen 08: Display screen to notice customer that his/her account isn't enough money to transfer and re-enter amount he/she want to transfer

Item	Type	Description
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Accept input of customer
	button	Do nothing at this screen
	button	Do nothing at this screen
	button	Continue transaction

Activities Flow

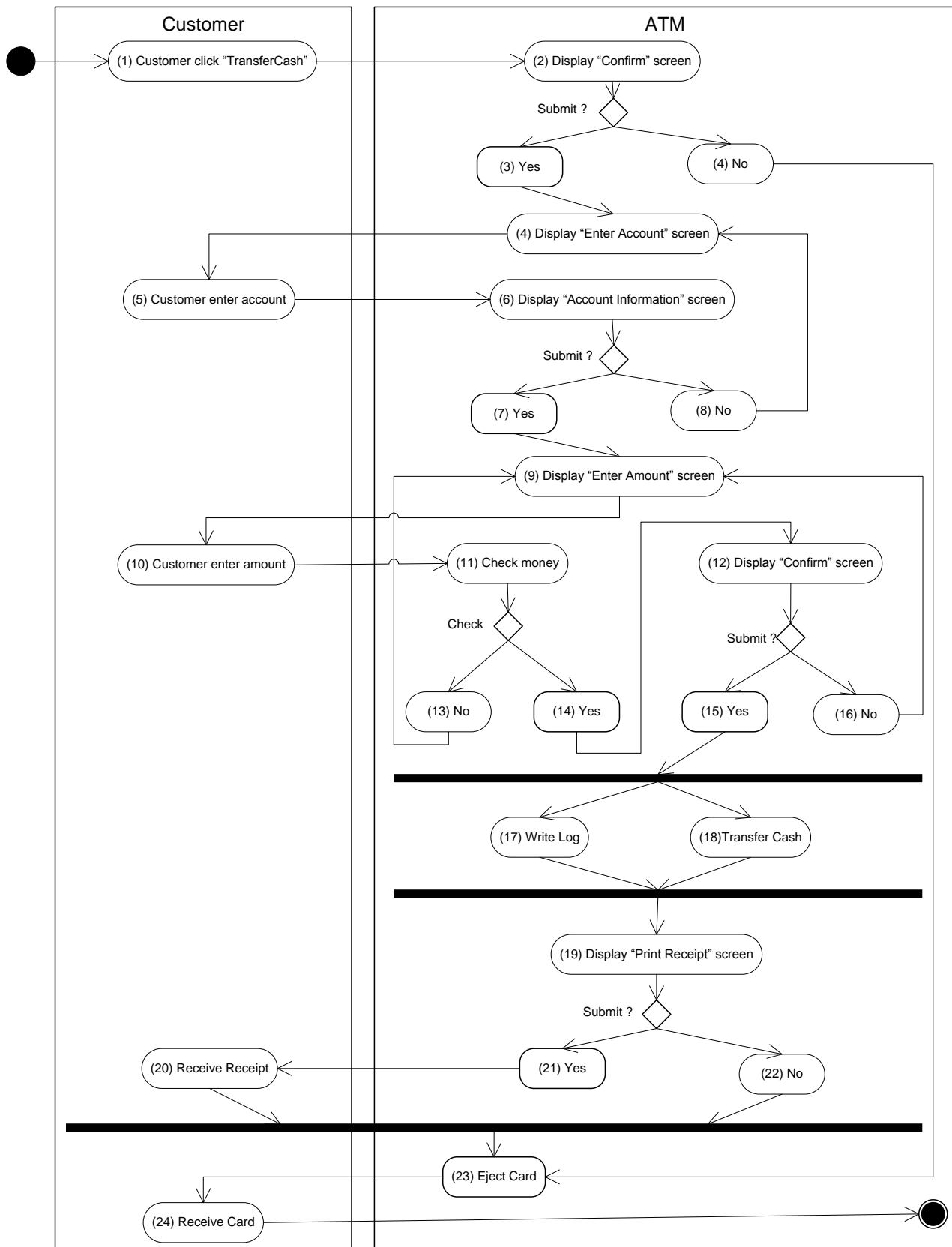
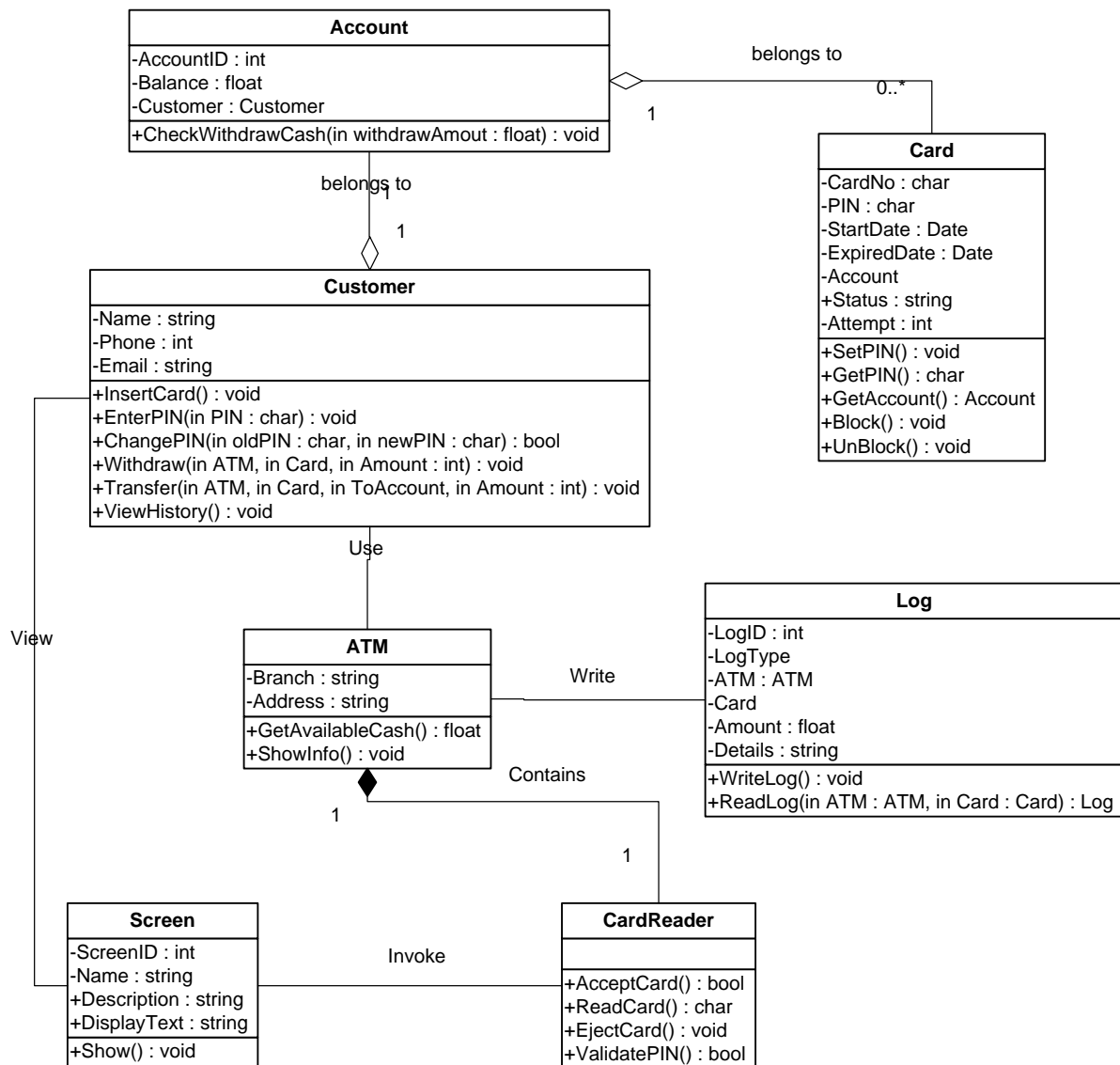


Figure 4.6:Activities flow of Cash transfer

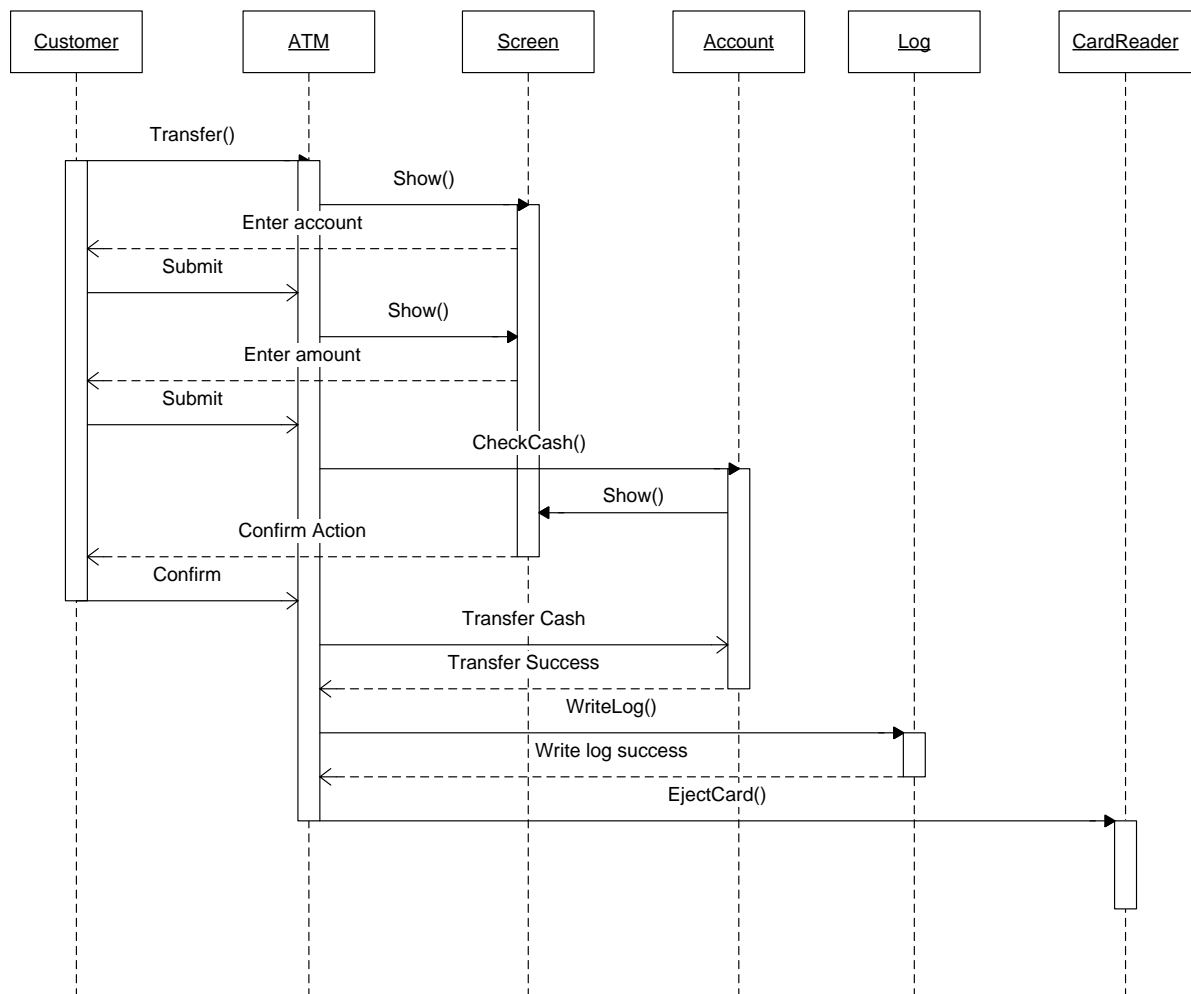
Detail Processing

Activity	BR Code	Description
(11)	BR01	Check Money <ul style="list-style-type: none">❖ IF <Customer> accept amount have just entered THEN<ul style="list-style-type: none">➤ Get balance of this account and compare with amount he/she want to transfer➤ IF (Balance > amount) THEN<ul style="list-style-type: none">▪ Continue transaction➤ ELSE<ul style="list-style-type: none">▪ DISPLAY "Your account not enough money to transfer"▪ Back to previous screen to enter another amount
(18)	BR02	Transfer Cash: <ul style="list-style-type: none">❖ IF <Customer> click accept transfer cash THEN<ul style="list-style-type: none">➤ Get amount and balance of send account and receive account➤ Plus amount to balance of receive account and subtract balance of send account

Class diagram



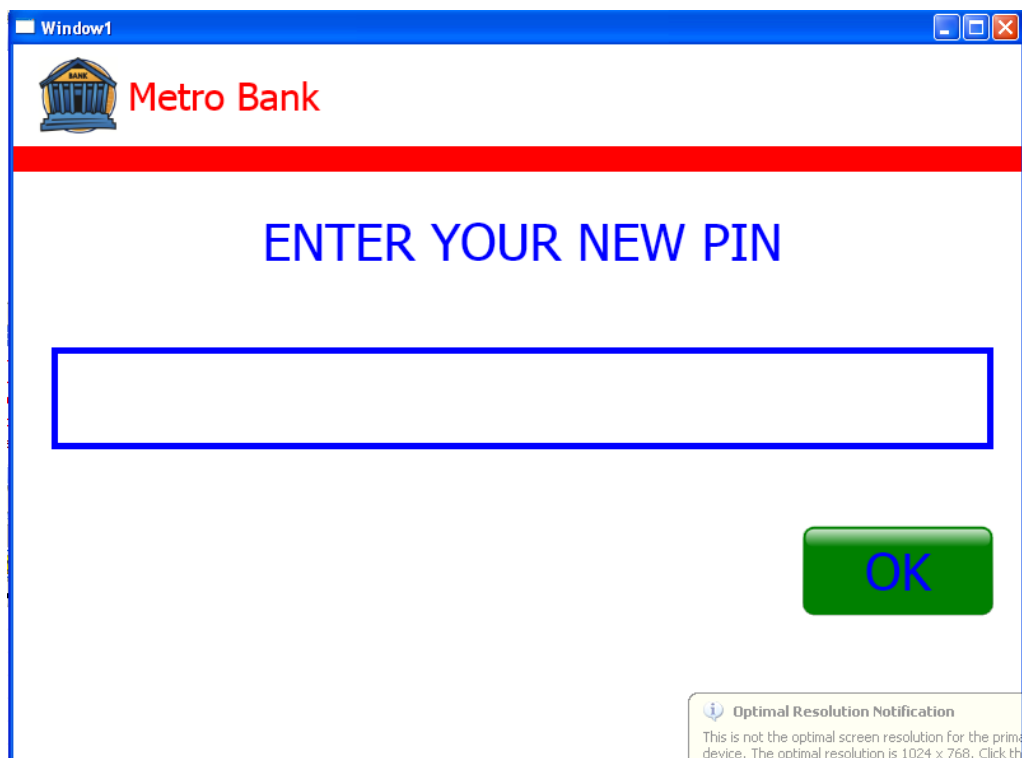
Sequence diagram




7.6 Use case 06: Change PIN – CuongLH3

Name	Change PIN
Description	This use case allows customer to change his/her PIN
Actor	Customers
Trigger	When user clicking on 'Change PIN' button in the screen.
Pre-condition	After Validation success
Post-condition	PIN of customer will change


Screen Design & Data Description



Screen 01: Display screen require customer enter old PIN

Item	Type	Description
	button	Submit old PIN

Window1




Metro Bank


ENTER YOUR NEW PIN

OK

Screen 02: Display screen require customer enter new PIN

Item	Type	Description
	button	Submit enter new PIN


Window1

 Metro Bank

CONFIRM YOUR NEW PIN

OK

Screen 03: Display screen require customer re-enter new PIN

Item	Type	Description
	button	Submit re-enter new PIN

Activities Flow

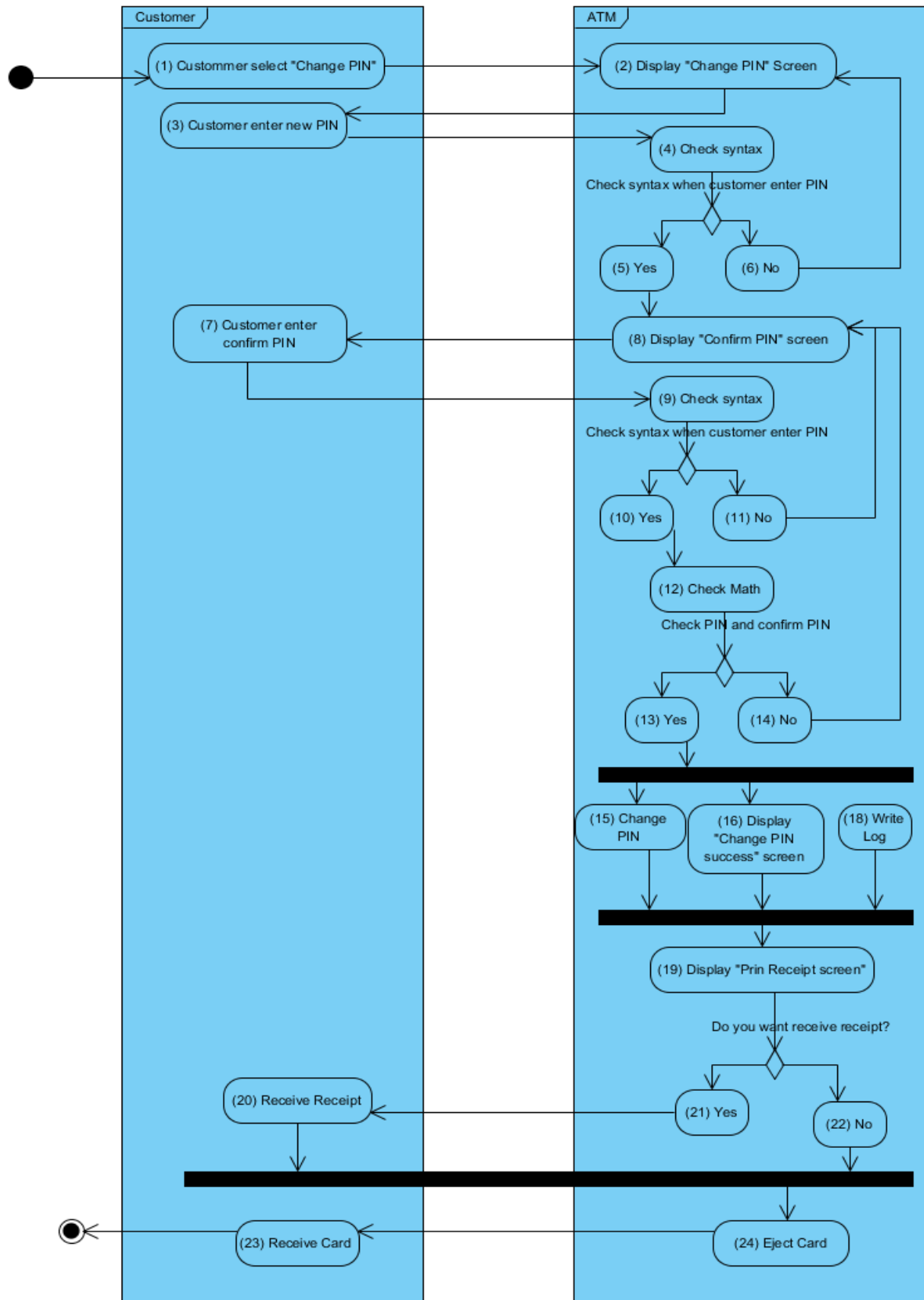
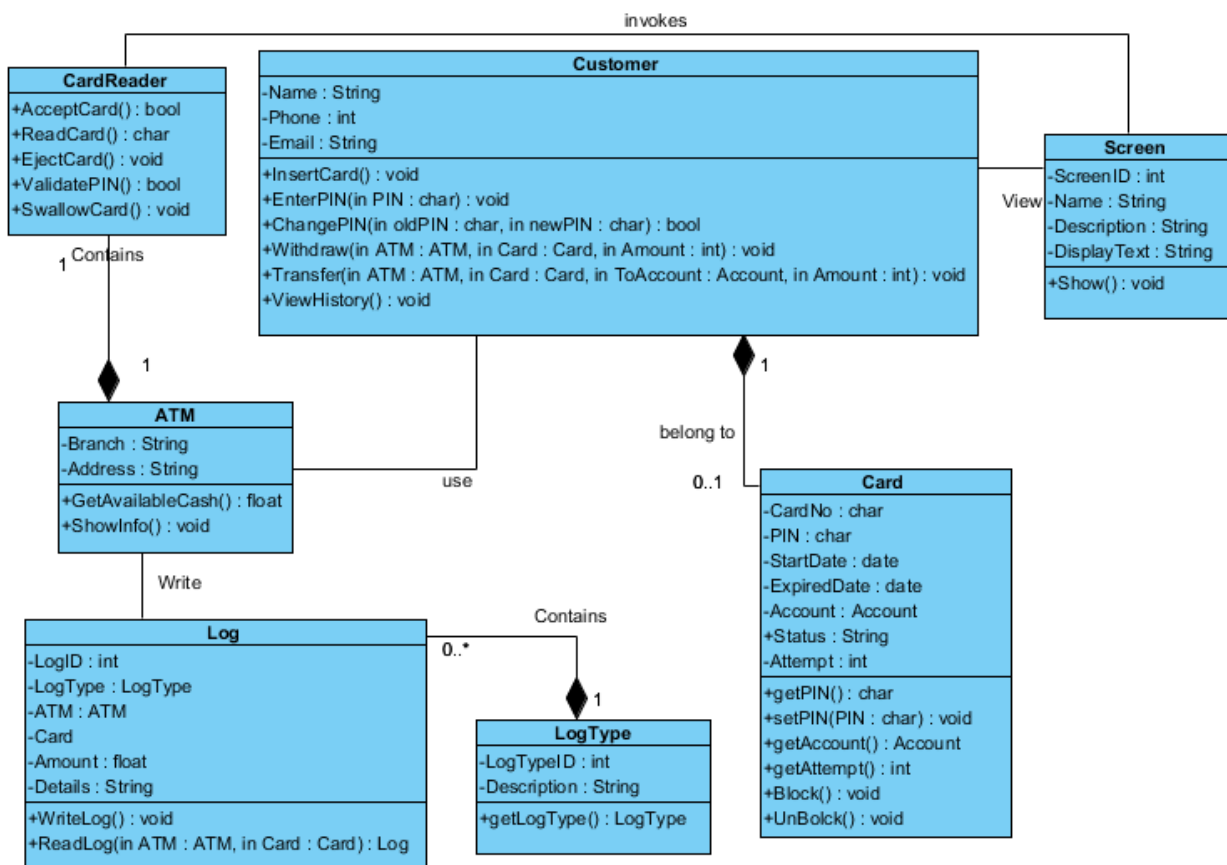


Figure 4.7:Activities flow of Change PIN

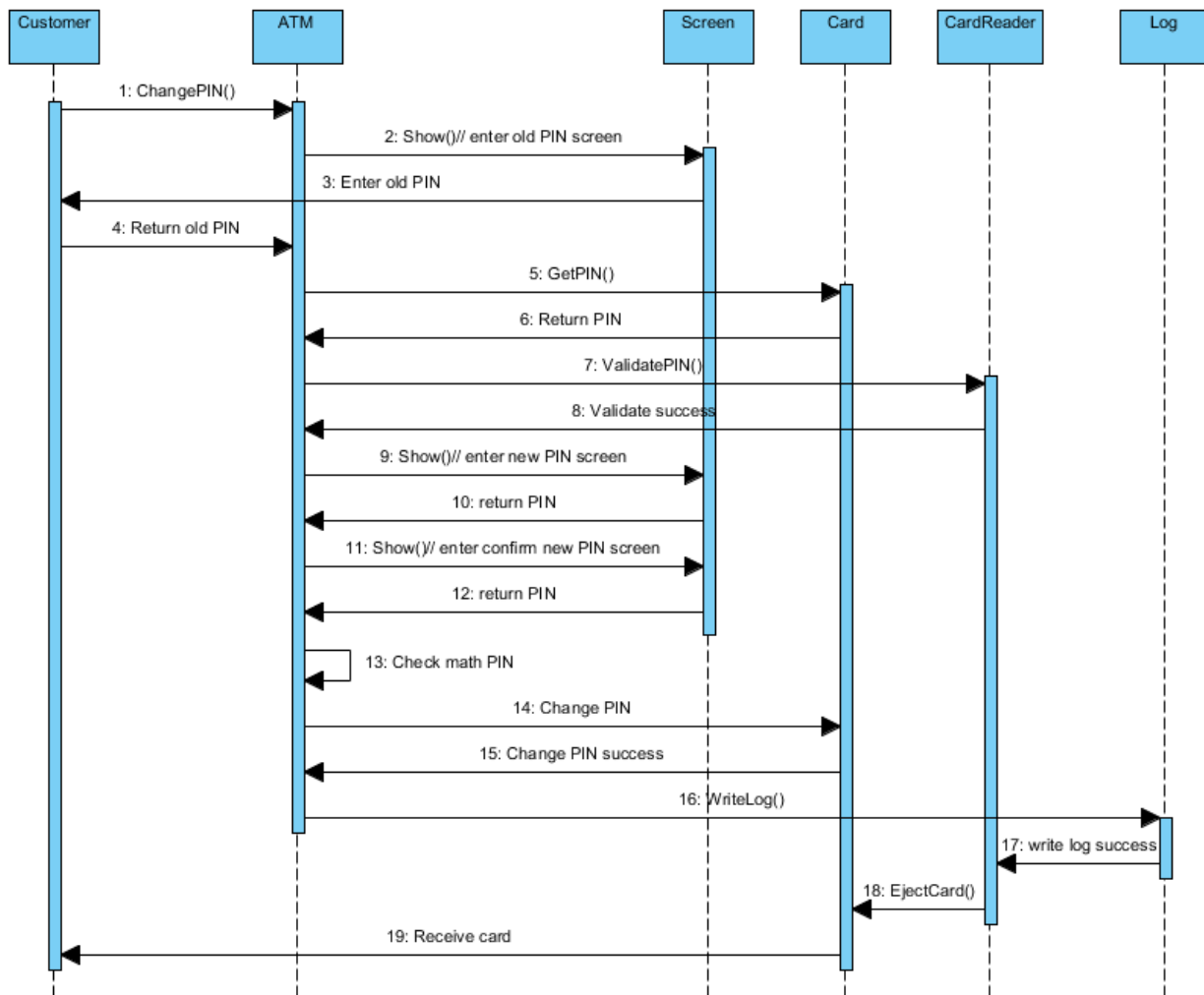
Detail Processing

Activity	BR Code	Description
(4) & (9)	BR01	Check Syntax <ul style="list-style-type: none"> ❖ Get old PIN of customer and compare with new PIN he/she have just entered ❖ IF old PIN match with new PIN THEN <ul style="list-style-type: none"> ➢ DISPLAY 'New pin not allow! Please enter again' ➢ Show change PIN screen again.
(15)	BR02	Change PIN <ul style="list-style-type: none"> ❖ Get new PIN and compare with new PIN which he/she re-enter ❖ IF 2 new PINs are match THEN <ul style="list-style-type: none"> ➢ Display "Your PIN changed" ➢ Eject Card and stop transaction

Class diagram



Sequence diagram



7.7 Use case 07: Logging – LuyenNV + HuyVQ4

7.7.1 Logging

Name	Logging
Description	This use case allows ATM system to log all transaction has been executed by customer
Actor	ATM System
Trigger	When user finish any transaction with ATM system
Pre-condition	A transaction has finished.
Post-condition	New record will be inserted to table Log in database stored information about customer transaction: date, transaction type, amount ...

Screen Design & Data Description

No screen.

Activities Flow

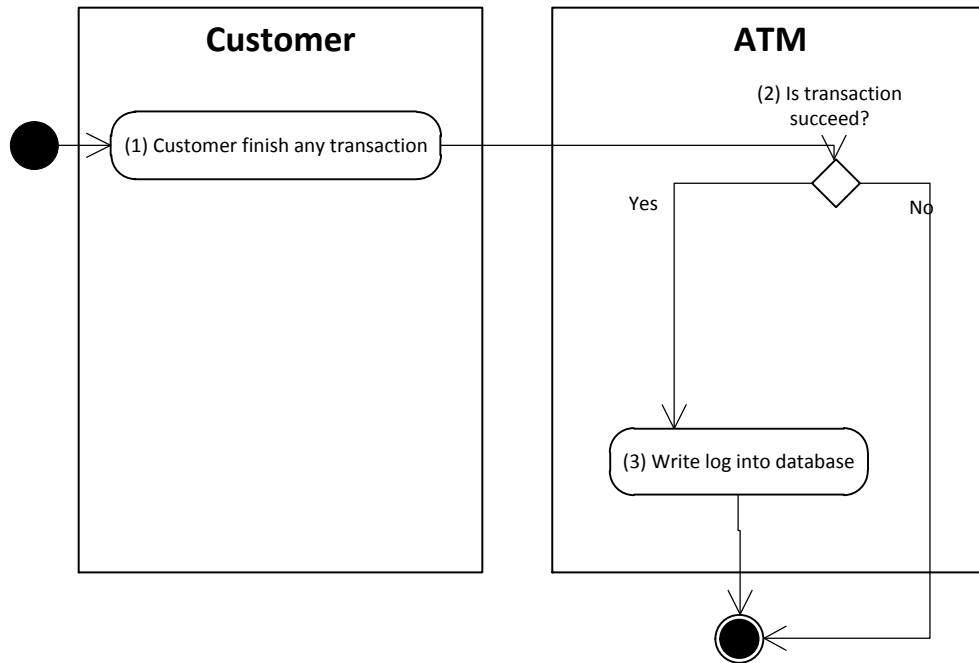
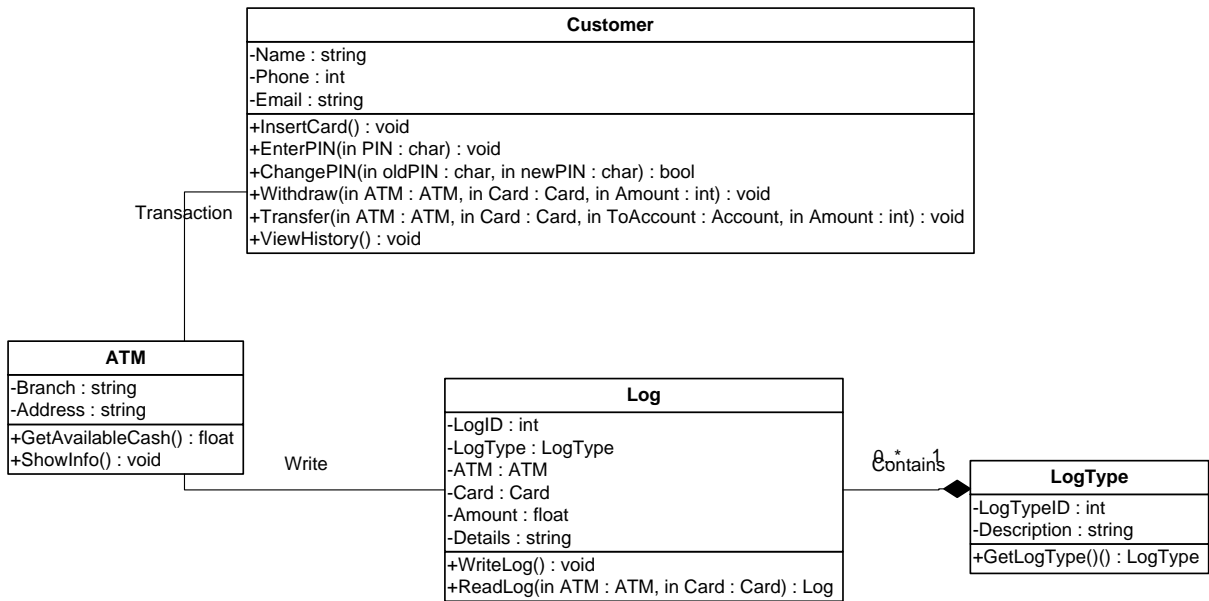


Figure 4.8:Activities flow of Logging

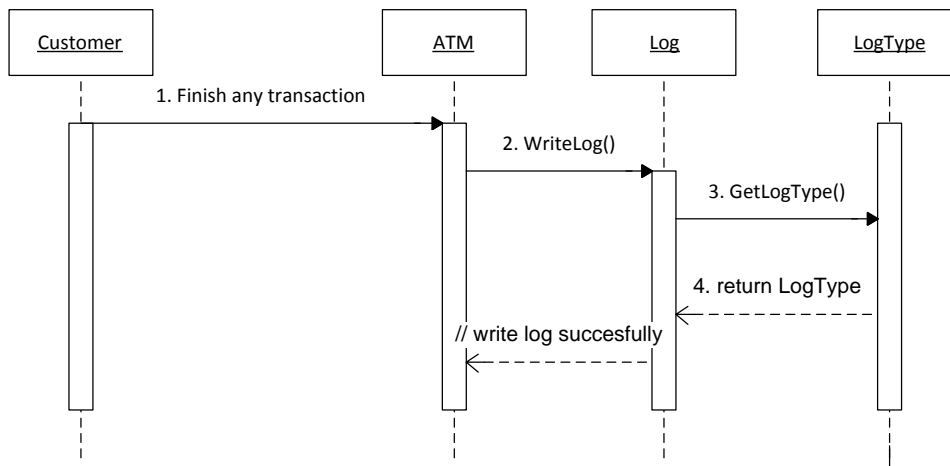
Detail Processing

Activity	BR Code	Description
(3)	BR01	❖ Insert new Log <ul style="list-style-type: none"> ➤ Set <<LogDate>> = [Current Time] ➤ Set <<LogType>> = [Transaction Type] ➤ Set <<ATM>> = [Current ATM Machine] ➤ Set <<Card>> = [Current Card] ➤ Set <<Amount>> = [Amount]

Class diagram



Sequence diagram



8 Configuration

8.1 Application Configuration

Field	Values	Remark
MinWithdraw	50.000	
MaxWithdraw	10.000.000	
Number Record per Page	5	

8.2 System Configuration

Field	Values	Remark
Date format	dd/MM/yyyy	
Time format	hh:mm:ss	
Format money	##,###,###.00	

9 Packaging and Deployment – HuyVQ4

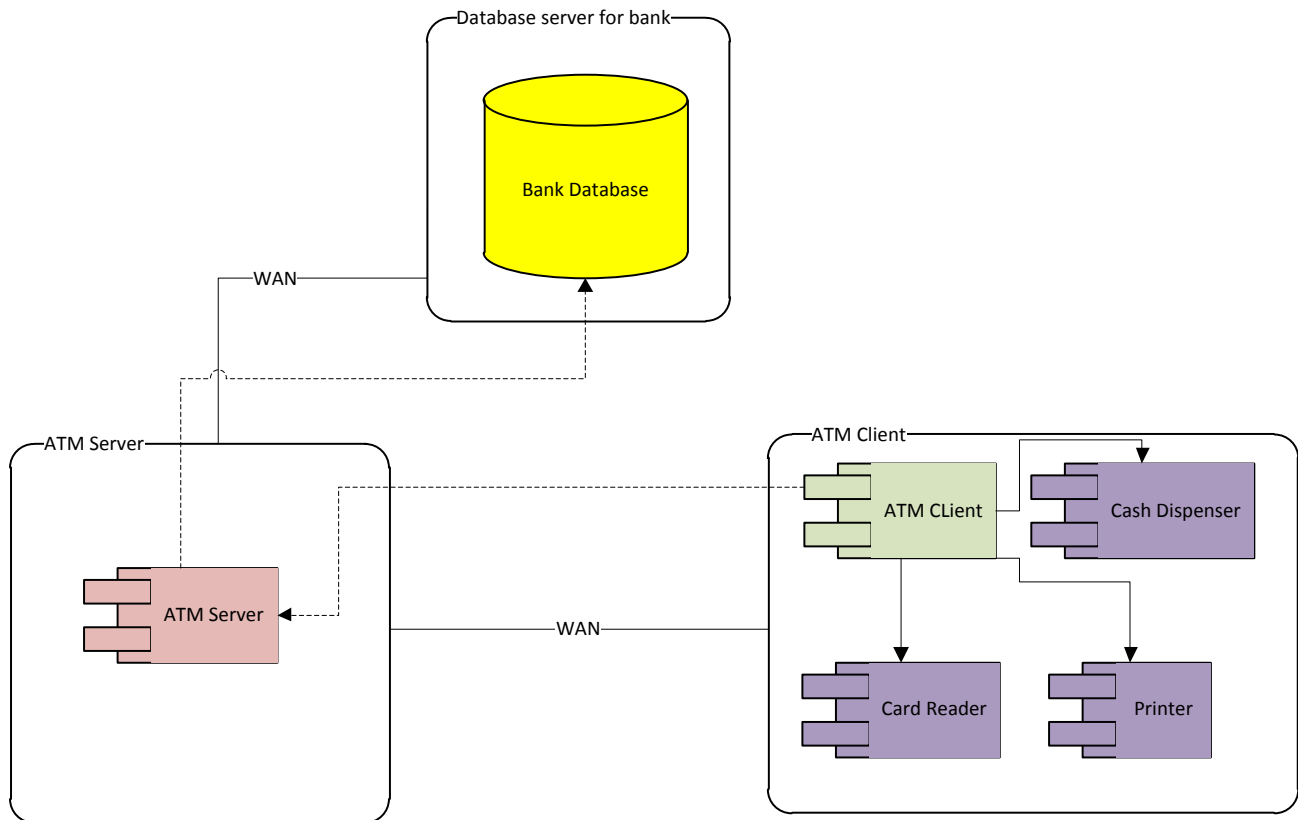


Figure 5.1:Deployment diagram

10 Appendix

10.1 Microsoft Enterprise library