

# A New Reliable ATM

OOPT Stage 1000

Plan and Elaboration

---

## Project Team T6

201411140 권성완

201511247 김선정

201510436 허윤아

201510285 조수빈

## Date

2018-04-01

## Activity 1001. Define Draft Plan

## 1. Motivation

기술의 발전으로 인하여 현금보다 카드를 많이 사용하게 되어 사람들이 현금을 많이 갖고 다니지 않게 되었다. 하지만 급하게 현금이 필요한 경우가 여전히 종종 남아있어 ATM 기기의 필요성은 더욱 증대되어 이 프로그램을 개발하게 되었다. 이 시스템을 통하여 고객들은 현금 입출금 서비스 뿐만 아니라 대출, 공과금 납부, 환전 등 다양한 업무들을 은행을 가지 않고도 이용할 수 있게 될 것이다.

## 2. Project Objectives &amp; Scope

- 은행에 가지않고 간단한 은행 업무를 볼 수 있다.
- 현금서비스를 이용할 수 있다.

## 3. Functional Requirements

- Deposit
- Withdraw
- Transfer
- Check balance
- Exchange
- Loan
- Input card / bankbook
- Take charge
- Check customer's data from server
- Check password from server
- Update server information
- Print error
- Pay utility bill

## 4. Non-functional Requirements

- 24시간 동안 구동될 시스템이므로 안정적으로 설계한다.
- 돈과 관련된 정보를 다루므로 정보가 정확해야 한다.

- 환전 시 환율은 국가마다 다르다.
- 프로그램 시뮬레이션이므로 모든 입력은 키보드를 통해 이루어진다.

#### 5. Resource Estimation

- Human Efforts(Man-Month) : 4 – 3months
- Project Duration : 12 weeks
- Other Efforts : 좋은 학점을 향한 집념, 조원들의 협동심

#### 6. Other Information

N/A

## Activity 1002. Preliminary Investigation Report

## 1. Alternative Solutions

- 기존의 잘 짜여진 시스템을 구매한다
- 다른 소프트웨어 개발 회사에 의뢰한다.

## 2. Project Justification (Business Demands)

- Cost : 시중의 제품보다 저렴하게 개발할 수 있다(인력비 감소).
- Duration : 회사에 의뢰했을 때보다 짧은 시간 내에 개발이 가능하다.
- Risk : 숙달되지 않은 개발자들의 개발로 인해 에러가 발생할 수 있고, 이로 인해 requirement를 충족시키지 못할 수 있다.
- Effect : requirement를 직접 짜 구현할 수 있고, 노력으로 비용을 감소시킬 수 있다.

## 3. Risk Management

Risk	Probability	Significance	Weight
Lack of JAVA Experience	3	4	12
Lack of UML Experience	3	4	12
Lack of domain knowledge	3	4	12
Lack of GUI Experience	3	4	12
Lack of Time	2	2	4
Lack of Sleep	2	2	4

## 4. Risk Reduction plan

- Lack of Java Experience (12) : JAVA 언어에 관한 책을 통해 공부한다.
- Lack of UML Experience (12) : 강의시간에 배운 내용을 복습하고 팀원끼리 서로 알려준다.
- Lack of domain knowledge (12) : domain diagram을 통해 domain에 대한 이해도를 늘린다. 평소 ATM기를 사용했던 경험을 떠올리며 작업한다.
- Lack of GUI Experience (12) : 책과 우리의 친구 구글에게 도움을 요청한다.
- Lack of Time (4) : 고학년들의 고질적인 문제로 부족한 시간은 여가시간 등을 줄여서 채운다.
- Lack of sleep (4) : 어느정도 감수할 부분으로 개인이 체력관리에 힘쓴다.

## 5. Market Analysis

- 이미 상용화되어 있는 ATM들이 많이 있으므로 이를 참조할 수 있으나, 개발의 필요성이 줄어든다.
- 상용화되어 있는 ATM들의 기능이 다양하므로, 이를 차용한다.

## 6. Other Managerial Issues

- 홈페이지에 기재된 개발 일정에 맞추어 작업한다.

## Activity 1003. Define Requirements

## 1. Functional Requirements (Version 1.0)

- Deposit

고객은 카드 또는 통장을 이용해 자신의 계좌에 현금을 입금한다. ATM이 처리할 수 있는 금액의 단위는 만원 단위이다.

- Withdraw

고객은 카드 또는 통장을 이용해 자신의 계좌에서 현금을 인출한다. 출금을 할 시 고객은 계좌의 비밀번호를 알고 있어야 한다. ATM이 처리할 수 있는 금액의 단위는 만원 단위이다.

- Transfer

고객은 카드 또는 통장을 이용해 자신의 계좌에서 돈을 송금하거나, 무통장으로 돈을 송금할 수 있다. 송금 시에는 정확한 상대방의 계좌정보가 필요하다. 자신의 계좌에서 송금을 할 경우 계좌의 비밀번호를 알고 있어야 한다.

- Check Balance

고객은 카드 또는 통장을 이용해 거래 내역과 그에 따른 잔고를 조회한다. 이후 고객이 확인 버튼을 누르면 초기 화면으로 돌아간다.

- Exchange

환율에 따라 고객이 원하는 국가의 현금으로 환전해준다. 환율은 국가마다 상이하다. 환전 시 현금이 아닌, 고객 본인의 카드 또는 통장이 있어야 한다. ATM이 처리할 수 있는 금액의 단위는 만원 단위이다.

- Loan

고객의 신용카드 한도 내에서 현금을 인출할 수 있는 기능으로, 고객은 신용카드의 비밀번호를 알고 있어야 한다. ATM이 처리할 수 있는 금액의 단위는 만원 단위이다.

- Check Validation

ATM은 고객의 카드 정보 혹은 통장 정보를 입력 받는다.

- Take Charge

환전, 타행 이체, 신용카드를 이용한 현금 출금을 할 때 발생하는 수수료를 받는 행위로 등급에 따라 면제될 수 있다.

- Check customer's data

ATM은 고객이 입력한 통장 정보 혹은 카드 번호를 통해 Offer 서버에 고객 정보를 요청한다.

- Check password

ATM은 Offer 서버에 정보를 요청해, 비밀번호가 기존에 등록된 것과 일치하는지 확인한다.

- Upload

ATM은 변경된 정보를 갱신할 것을 Offer 서버에 요청한다.

- Print Error

ATM은 Error가 발생한 상황에서 Error 메시지를 고객에게 알린다. Error가 발생하는 상황은 고객의 정보가 서버에 등록된 정보와 일치하지 않을 때이다.

- Pay Utility Bill

고객은 지로 고지서를 입력해, 카드 혹은 계좌를 통해 공과금을 납부한다.

## 2. System Functions (Table)

Ref. #	Function	Category
R1.1	Deposit	Evident
R1.2	Withdraw	Evident
R1.3	Transfer	Evident
R2.1	Check Balance	Evident
R2.2	Exchange	Evident
R2.3	Loan	Evident
R3.1	Check Validation	Evident
R4.1	Take charge	Evident
R5.1	Check Customer Data	Hidden
R5.2	Check Password	Hidden
R5.3	Update	Hidden
R6	Print Error	Hidden
R6.1	Check Error	Hidden
R7	Pay Utility Bill	Evident

## 3. Performance Requirements

- 현금 거래에 관한 정보를 신속하고 정확하게 처리하여 서비스를 제공한다.

#### 4. Operating Environments

- OS : Microsoft Windows 10 (64bit)
- CPU : i5-7200U
- Memory : 8GB

#### 5. Development Environments

- OS : Microsoft Windows 10 pro (64bit)
- CPU : i5-7200U
- Memory : 8GB
- Program Language : JAVA
- Editor : eclipse

#### 6. Interface Requirements

- 사용자의 메뉴 선택에 의해 각 기능들을 수행할 수 있는 화면으로 전환된다.
- 다음 버전의 업데이트를 통해 좀 더 정교한 프로그램을 설계한다.

#### 7. Other Requirements

(N/A)



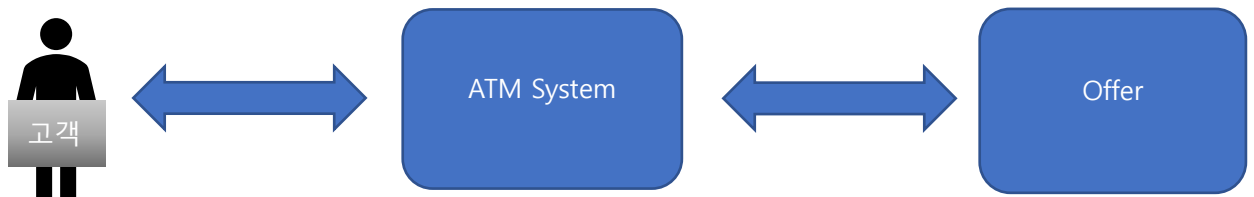
Activity 1004. Record terms in Glossary

Term	Description	Remarks
ATM	Automated Teller Machine의 준말로 현금 자동 입출금기를 의미한다	
Offer	ATM 서비스를 제공하는 회사로 은행과 카드사가 이에 해당한다.	
Bankbook	고객의 계좌 정보를 담고있는 통장	
Upload	변경된 정보를 갱신하는 것으로 ATM이 Offer에게 요구한다.	

Activity 1005. Implement Prototype  
(생략)

Activity 1006. Define Business Use-Cases

1. Define System Boundary



2. Identify and Describe Actors

- Customer : ATM을 이용하는 고객
- ATM : Offer와 고객의 중간다리 역할을 해주는 시스템. 고객이 더 편하게 금융거래를 할 수 있도록 돕는다.
- Offer(Hidden Actor) : 실제로 고객의 금융거래 정보가 저장되는 hidden actor

3. Identify Use-Case

Use Cases by Actor – Customer

Deposit	Withdraw	transfer	check balance
Exchange	input card/bankbook	loan	pay utility bill

Use Cases by Actor – ATM

Deposit	Withdraw	Transfer	check balance	pay utility bill
---------	----------	----------	---------------	------------------

Exchange	take charge	Loan	check password	
----------	-------------	------	----------------	--

## Use Cases by Event

Check Customer Data	Print Error Message	Update Offer Server	Check Error
---------------------	---------------------	---------------------	-------------

## 4. Allocate system functions into Related Use Cases

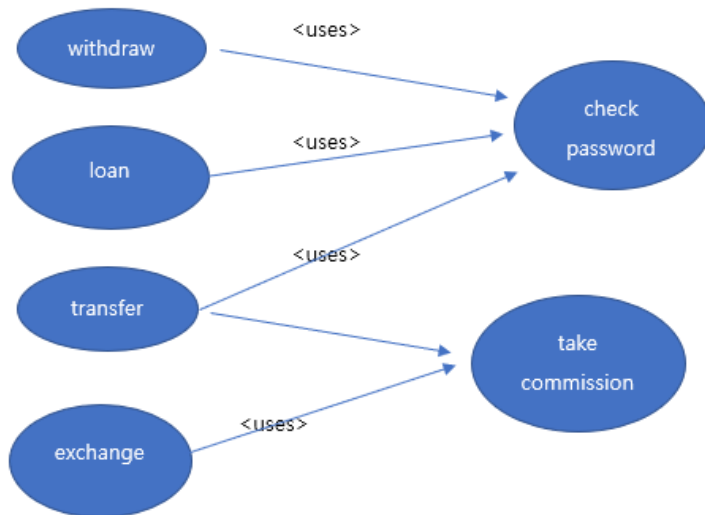
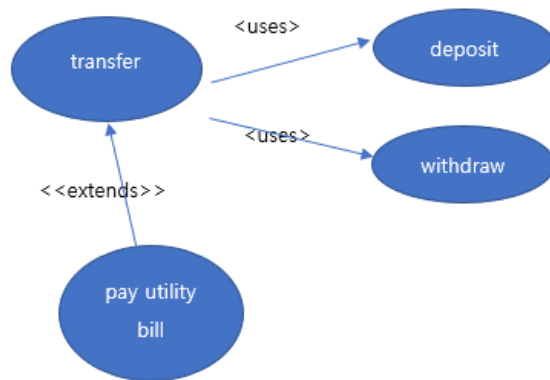
Ref. #	Function	Use Case Number & Name	Category
R1.1	Deposit	1.Deposit	
R1.2	Withdraw	2.Withdraw	
R1.3	Transfer	3.Transfer	
R2.1	Check Balance	4.Check Balance	
R2.2	Exchange	5.Exchange	
R2.3	Loan	6.Loan	
R3.1	Check Validation	7.Input card/bankbook	
R4.1	Take charge	8.Take Charge	
R5.1	Check Customer Data	9.Check Customer Data	
R5.2	Check Password	10. Check Password	
R5.3	Update	11. Update Offer Server	
R6.1	Print Error	12. Print Error Message	
R6.2	Check Error	13. Check Error	
R7	Pay Utility Bill	14. Pay Utility Bill	

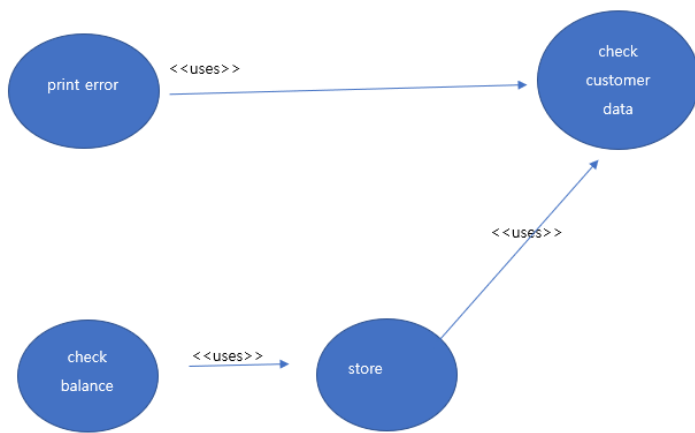
## 5. Categorize Use Cases

Ref. #	Function	Use Case Number & Name	Category
R1.1	Deposit	1.Deposit	Primary
R1.2	Withdraw	2.Withdraw	Primary
R1.3	Transfer	3.Transfer	Primary
R2.1	Check Balance	4.Check Balance	Primary
R2.2	Exchange	5.Exchange	Primary
R2.3	Loan	6.Loan	Primary
R3.1	Check Validation	7.Input card/bankbook	Primary
R4.1	Take charge	8.Take Charge	Secondary
R5.1	Check Customer Data	9.Check Customer Data	Secondary
R5.2	Check Password	10. Check Password	Secondary

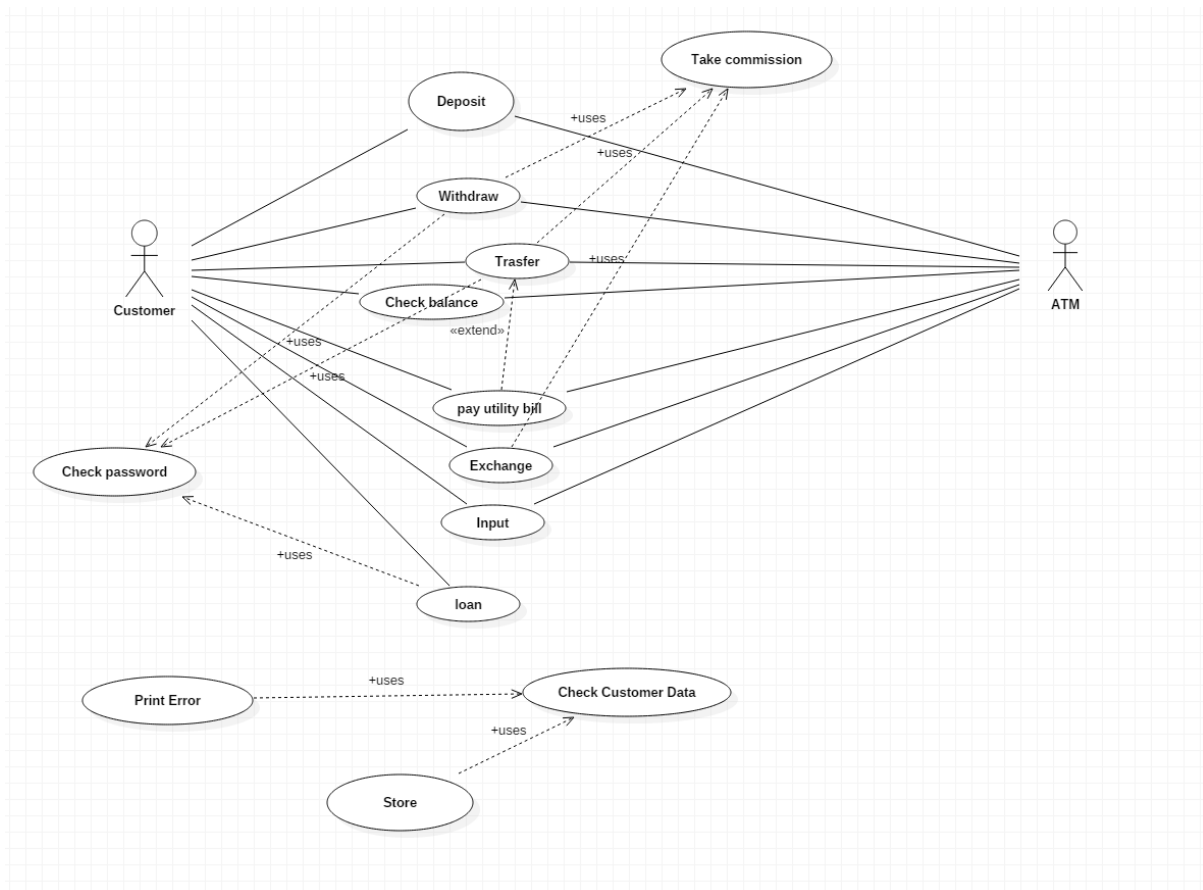
R5.3	Update	11. Update Offer Server	Secondary
R6.1	Print Error	12. Print Error Message	Secondary
R6.2	Check Error	13. Check Error	Secondary
R7	Pay Utility Bill	14. Pay Utility Bill	Secondary

6. Identify the relationships between Use-Case





7. Draw a Use-Case diagram



8. Describe Use-Case

<b>Use Case</b>	1. Deposit
<b>Actors</b>	Customer, ATM
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can deposit cash into account by using card or bankbook.</li> <li>-Customer can deposit money into credit card.</li> <li>-Customer can only deposit money in unit of 10000₩.</li> <li>-If customer successfully deposits, ATM gives updated information to Offer's server.(Hidden)</li> <li>-If customer deposits cash into credit card, ATM gives updated information to Offer's server.(Hidden)</li> </ul>

<b>Use Case</b>	2. Withdraw
<b>Actors</b>	Customer, ATM
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can withdraw cash from account by using card or bankbook.</li> <li>-Customer can only withdraw money in unit of 10000₩ under balance.</li> <li>-Customer has withdrawing limit at a day.</li> <li>-Customer needs to choose the number of 10000₩ and 50000₩.</li> <li>-Customer needs to know password of an account.</li> <li>-If customer successfully withdraws, ATM gives updated information of to Offer's server.(Hidden)</li> </ul>

<b>Use Case</b>	3. Transfer
<b>Actors</b>	Customer, ATM
<b>Description</b>	<ul style="list-style-type: none"> <li>-Customer can transfer money to other's account by using card or bankbook, or by paying cash.</li> <li>-If customer uses card or bankbook, ATM requests password and balance to Offer's server.</li> <li>-If customer enters wrong password, ATM prints out error message.</li> <li>-Customer needs to input receiver's account number correctly.</li> <li>-If customer did it wrong, ATM prints out the error message.</li> <li>-If customer is transferring money to other bank's account, ATM takes charge of certain percentage.</li> <li>-If customer successfully transfers money from account, ATM gives updated information to Offer's server.(Hidden)</li> </ul>

<b>Use Case</b>	4. Check balance
<b>Actors</b>	Customer, ATM
<b>Description</b>	-Customer can check balance of own account. -ATM prints out customer's recent transaction history.

<b>Use Case</b>	5. Exchange
<b>Actors</b>	Customer, ATM
<b>Description</b>	-Customer can exchange money into other country's cash. -When customer exchanges money, ATM takes charge of certain percentage. -Customer can only use card or bankbook to exchange money. -If customer successfully exchanges money, ATM gives updated information to Offer's server.(Hidden)

<b>Use Case</b>	6.Loan
<b>Actors</b>	Customer, ATM
<b>Description</b>	-Customer can withdraw cash with credit card, and we call it Loan. -Each customer has loan limits. -Credit Card Company(Hidden actor) lends money, and requests ATM to take charge. -If customer wants to repay, customer can deposit money into credit card.

<b>Use Case</b>	7.Input Card or Bankbook
<b>Actors</b>	Customer, ATM
<b>Description</b>	-Customer can choose when to input card or bankbook, before or after selecting the menu. -When customer inputs card or bankbook, ATM requests customer data to the Offer's server to check validation.

<b>Use Case</b>	8.Take Charge
<b>Actors</b>	ATM
<b>Description</b>	-Offer(hidden actor) requests to ATM to take charge from customer when transferring to other bank, loaning, exchanging. -If customer's grade is high, ATM doesn't take charge.

<b>Use Case</b>	9.Check customer data
<b>Actors</b>	ATM
<b>Description</b>	-This use case begins when customer inputs card or bankbook. -ATM requests customer data to Offer's server, and compares to check validation.

<b>Use Case</b>	10.Check password
<b>Actors</b>	ATM
<b>Description</b>	-This use case begins when customer inputs password of own account while withdrawing, transferring, loaning. -ATM compares data requested to Offer's server, and check validation.

<b>Use Case</b>	11.Update Offer Server
<b>Actors</b>	ATM
<b>Description</b>	-This use case begins when customer successfully deposits, withdraws, transfers from account, and loans, and exchanges, and pays utility bills. -ATM requests to renew Offer's server data by sending updated data(recent transaction record).

<b>Use Case</b>	12.Print Error Message
<b>Actors</b>	ATM
<b>Description</b>	-This use case begins when customer input wrong account number while transferring money, or wrong password. -ATM prints out error message to customer.

<b>Use Case</b>	13.Check Error
<b>Actors</b>	ATM
<b>Description</b>	-This use case is triggered by every input customer makes. -Before this use case, ATM requests information to Offer's server to check error. -ATM checks if there are enough balance, or if account number that customer is trying to transfer is correct, or if there are limits when loaning, withdrawing, and transferring, or if customer inputs money not in the unit of 10000₩.

<b>Use Case</b>	14.Pay Utility Bill
<b>Actors</b>	Customer, ATM
<b>Description</b>	-Customer can only use card or bankbook to pay utility bill. -Customer needs to input notice of payment before paying bill. -Customer can only deposit money into virtual payment account.

## 9. Rank Use-Cases

<b>Rank</b>	<b>Function</b>
High	1.Deposit
High	2.Withdraw
High	3.Transfer
High	4.Check Balance
Medium	5.Exchange
Medium	6.Loan
High	7.Input card/bankbook
Medium	8.Take Charge
High	9.Check Customer Data
Medium	10.Check Password
High	11.Update Offer Server
High	12.Print Error Message
High	13.Check Error
Medium	14.Pay Utility Bill



## Activity 1007. Define Business Concept Model

Customer(User)	ATM	Credit Card Company	Bank	Credit Card	Check Card	Cash
Name	Password	Utility Bill	Exchange rate(USD)	Exchange rate(JPY)	Exchange rate(EUR)	Exchange rate(CNY)
Deposit	Withdraw	Transfer	Check Balance	Exchange	Loan	Input
Commission	Store	Check Error	Print Error	Bank book	Limit	Dept
Account Number	Account Owner	10000₩	50000₩	Grade	Customer's Bank Data	Customer's Grade Data

## Activity 1008. Define Draft System Architecture

(생략)

## Activity 1009. Refine Plan

## 1. Project Scope &amp; Objectives

- 은행에 가지않고 간단한 은행 업무를 볼 수 있다.
- 현금서비스를 이용할 수 있다.

## 2. Functional Requirements

Ref. #	Function	Category
R1.1	Deposit	Evident
R1.2	Withdraw	Evident
R1.3	Transfer	Evident
R2.1	Check Balance	Evident
R2.2	Exchange	Evident
R2.3	Loan	Evident
R3.1	Check Validation	Evident
R4.1	Take charge	Evident
R5.1	Check Customer Data	Hidden
R5.2	Check Password	Hidden
R5.3	Update	Hidden

R6	Print Error	Hidden
R6.1	Check Error	Hidden
R7	Pay Utility Bill	Evident

### 3. Performance Requirements

- 현금 거래에 관한 정보를 신속하고 정확하게 처리하여 서비스를 제공한다.

### 4. Operating Environments

- OS : Microsoft Windows 10 (64bit)
- CPU : i5-7200U
- Memory : 8GB

### 5. User Interface Requirements

- 사용자의 메뉴 선택에 의해 각 기능들을 수행할 수 있는 화면으로 전환된다.
- 다음 버전의 업데이트를 통해 좀 더 정교한 프로그램을 설계한다.

### 6. Other Requirements (N/A)

### 7. Resources

- Human Efforts(Man-Month) : 4 – 3months
- Project Duration : 12 weeks
- Other Efforts : 좋은 학점을 향한 집념, 조원들의 협동심

### 8. Scheduling

Stage	Phase(00x0)/Activity(000x)	Schedule(Day)				
		1	2	3	4	5
1000. Plan & Elaborate	1001. Define Draft Plan	█				
	1002. Create Preliminary Investigation Report	█				
	1003. Define Requirements	█				
	1004. Record Terms in Glossary	█				
	1005. Implement Prototype	█				
	1006. Define Use Cases	█				
	1007. Define Draft Conceptual Model	█				
	1008. Define Draft System Architecture	█				
	1009. Refine Plan	█				
2000. Build	2010. Revise Plan					
	2020. Synchronize Artifacts					
	2030. Analyze					
	2031. Define Essential Use Case	█				
	2032. Refine Use Case Diagrams	█				
	2033. Refine Conceptual Model	█				
	2034. Refine Glossary	█				
	2035. Define System Sequence Diagrams	█				
	2036. Define Operation Contracts	█				
	2037. Define State Diagrams	█				
	2040. Design					
	2041. Define Real Use Cases					
	2042. Define Reports, UI and Storyboards					
	2043. Refine System Architecture					
	2044. Define Interaction Diagrams					
	2045. Define Design Class Diagrams					
	2046. Define Database Schema					
	2050. Construct					
	2051. Implement Class & Interface Definition					
	2052. Implement Methods.					
	2053. Implement Windows					
	2054. Implement Reports					
	2055. Implement DB Schema					
	2056. Write Test Code					
	2060. Test					
	2061. Unit Testing					
2062. Integration Testing						
2063. System Testing						
2064. Performance Testing						
2065. Acceptance Testing						
2066. Documentation Testing						
3000. Deploy -ment	3001. Complete Technical Documents					
	3002. Complete User Documents					
	3003. System Testing					
	3004. Acceptance Testing					
	3005. Documentation Testing					
	3006. Train					
	3007. Establish Parallel Runs and Crossover					
	3008. Establish Support					
	3009. Install					

(SMA Case Study : Plan 마지막 도표 참조)