
Software Requirements Specification

for

HY Bank ATM,

Release 1.0

Version 1.0 approved

Prepared by 윤상현

October 13, 2022

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction(서론)	1
1.1 Purpose(목적)	1
1.2 Project Scope and Product Features(범위와 특징)	1
1.3 Stakeholders (이해당사자).....	1
1.4 References(참고문헌)	2
1.5 Glossary (용어정의).....	2
2. Overall Description(개요)	3
2.1 Product Perspective(제품 개요).....	3
2.2 User Characteristics(사용자 특성)	3
2.3 Operating Environment(동작환경).....	3
2.4 General Constraints(제약사항).....	3
2.5 Assumptions and Dependencies(가정 및 의존사항).....	4
3. Functional Requirements(기능적 요구사항)	4
3.1 FR-1: 입금	4
3.2 FR-2: 출금	5
3.3 FR-3: 이체	5
4. Interface Requirements(인터페이스 요구사항).....	6
4.1 User Interfaces(사용자 인터페이스).....	6
4.2 Hardware Interfaces(하드웨어 인터페이스).....	7
4.3 Software Interfaces(소프트웨어 인터페이스)	9
5. Nonfunctional Requirements(비기능적 요구사항).....	10
5.1 Performance Requirements(성능)	10
5.2 Safety Requirements(안전성)	10
5.3 Security Requirements(보안).....	11
5.4 Software Quality Attributes(품질)	12

Revision History

Name	Date	Reason For Changes	Version
윤상현	10/9/22	initial draft	1.0 draft 1
윤상현	11/15/22	baseline following changes after inspection	1.0 approved

1. Introduction(서론)

1.1 Purpose(목적)

목적: 이 문서의 목적은 은행 시스템에서 사용할 Atm 기반 입금, 출금, 이체 서비스를 구현하기 위함이다.

용도: 해당 문서를 통해 시스템을 설계, 구현을 해보고 검증하여 더 나은 설계를 하기 위함이다.

1.2 Project Scope and Product Features(범위와 특징)

제품의 용도는 ATM을 사용하는 사용자가 입금, 출금, 이체를 통하여 자신의 자산을 관리하는 것이다. 이 시스템은 단순 ATM에서의 입출금 작업 뿐만이 아니라, 이후 인터넷 뱅킹 시스템을 지원하게 된다면 인터넷 뱅킹의 입, 출금, 이체 시스템에도 적용될 수 있다.

1.3 Stakeholders (이해당사자)

- **PM**
 - 1) 프로젝트 이해관계자 식별 및 교류 관리를 책임진다.
 - 2) 중요한 이해관계자들을 식별하고 그들과 교류한다.
 - 3) 팀원들의 지원을 받고, 해결 불가능한 문제에 직면하면 후원자에게 의뢰한다.
- **후원자**
 - 1) 프로젝트의 소유자로서 프로젝트 성공의 궁극적인 책임자.
 - 2) 프로젝트 보호자 및 옹호자로 필요한 지원, 자원을 공급한다.
 - 3) 프로젝트의 존속 여부를 결정한다.
- **프로젝트 팀**
 - 1) 교류 전략을 수립하는 것을 지원하거나 이해관계자 정보 수집 및 평가 수행한다.
 - 2) PM 이 프로젝트의 이해관계자를 식별하고 requirement 를 판단하는 것을 돕는다.
 - 3) 실질적인 프로젝트를 수행한다.

1.4 References(참고문헌)

https://itwiki.kr/w/%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8_%EC%9D%B4%ED%95%B4%EA%B4%80%EA%B3%84%EC%9E%90

<https://hanamon.kr/%EC%BB%B4%ED%93%A8%ED%84%B0%EC%9D%98-io-%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4-gui-cli/#:~:text=CLI%EB%8A%94%20Command%20Line%20Interface,%EC%82%AC%EC%9A%A9%20%EB%90%9C%20%EC%A0%84%ED%86%B5%EC%A0%81%EC%9D%B8%20%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4%EC%9D%B4%EB%8B%A4.&text=GUI%EA%B0%80%20%ED%95%A0%20%EC%88%98%20%EC%9E%88%EB%8A%94,CLI%EB%A1%9C%20%ED%95%A0%20%EC%88%98%20%EC%9E%88%EB%8B%A4.>

https://ko.wikipedia.org/wiki/%EB%B9%84%EA%B8%B0%EB%8A%A5_%EC%9A%94%EA%B1%B4#:~:text=%EC%8B%9C%EC%8A%A4%ED%85%9C%20%EA%B3%B5%ED%95%99%EA%B3%BC%20%EC%9A%94%EA%B5%AC%EA%B3%B5%ED%95%99,%EC%9A%94%EA%B1%B4%EA%B3%BC%EB%8A%94%20%EB%8C%80%EC%A1%B0%EB%90%9C%EB%8B%A4.

<https://m.cafe.daum.net/smbitpro/TFEn/7?listURI=%2Fsmbitpro%2FTFEn>

1.5 Glossary (용어정의)

- **Database** : 저장할 정보들을 저장해주는 공간
- **영속성 컨텍스트**: 데이터를 db 에 저장하기 전에 담아두는 공간
- **GUI**: 이미지, 아이콘, 텍스트 상자와 같은 시각 보조 도구로 사용자가 컴퓨터와 상호작용하게 만들어지는 시스템
- **Transaction**: 데이터를 실제 database 에 반영하는 의미 단위.
- **Commit**: 영속성 컨텍스트의 데이터를 실제 db 에 반영시키는 것
- **Rollback**: 에러 발생시 해당 부분을 다시 시작하는 것
- **Inconsistency**: 정보가 일관되지 못하는 것
- **Overflow**: 용량초과
- **LRU**: 가장 오랫동안 쓰이지 않은 데이터. 캐시 메모리에서 가장 먼저 쫓아낼 대상.

2. Overall Description(개요)

2.1 Product Perspective(제품 개요)

- 제품의 핵심 기능 소개
- 제품 개발의 주요 고려 사항
- 해당 제품은 ATM 기기를 통하여 사용자가 입금, 출금, 이체 등 자산을 관리하는 것에 편의를 주기 위해 고안되었다.
- 자산에 관련된 영역은 보안과 오류 처리가 최우선으로 고려되어야 한다.

2.2 User Characteristics(사용자 특성)

- 이체자 : 자신이 가지고 있는 자산을 타인에게 이체한다.
- 출금자: 자신이 가지고 있는 자산을 현금으로 인출한다.
- 입금자: 자신이 가지고 있는 자산을 입금한다.
- 수령자: 타인으로부터 일정 금액을 입금 받는다.

2.3 Operating Environment(동작환경)

- 사용자 환경: ATM 기기를 통해 다양한 은행 작업을 수행한다.
- 하드웨어 환경: ATM 기기를 통해 사용자가 입력한 정보들을 처리한 뒤 소프트웨어 환경에 넘긴다.
- 소프트웨어 환경: 서버에서는 ATM 기기에서 보낸 정보들을 처리,반환하고, 보존해야 할 데이터는 DB 에 넘겨 보존한다.

2.4 General Constraints(제약사항)

- Java 언어를 사용하여 구현한다.
- 보존해야 할 데이터를 내장할 데이터베이스를 위해 MariaDb 등의 데이터 베이스를 이용한다.
- 최신버전의 자바언어로 개발한다.
- 자산에 실질적인 변동이 발생하는 입금, 출금, 이체 등 핵심 기능에서 오류를 최소화하고, 예기치 않은 오류 발생시 그것을 감지할 수 있게 한다.

2.5 Assumptions and Dependencies(가정 및 의존사항)

- 사용자가 ATM 기기, 혹은 다른 디지털 장치를 어느정도 사용할 수 있다고 가정한다.
- 자산의 변동은 critical 한 영역이므로 모든 기능들을 수행할 때, 에러가 발생할 수 있다는 가정하에 기능을 구현한다.

3. Functional Requirements(기능적 요구사항)

3.1 FR-1: 입금

Use case 형태로 핵심 functional requirements(FR) 작성

ID	UC-1-FR-1
Name	입금
Primary actor	입금자
Goal in context	자신의 돈을 자신의 통장에 넣는 것
Preconditions	1. 입금할 곳이 정해져 있어야 한다. 2. 입금할 수 있는 상황이어야 한다. 3. 사용자가 하는 입력은 완료된 상황이다.
Scenario	1. 입금 정보를 입력 받는다. (입금 위치, 금액 등) 2. 입금 받을 곳에 입금이 들어왔다는 요청을 보낸다. 3. ATM에서 요청을 받으면 돈을 받고 해당 위치에 들어온 돈만큼의 데이터를 쌓아둔다.
Exception	1. Data가 중간에 손실되는 경우 2. 사용자 잔액 부족 3. 외적 요인으로 정보가 전달되지 못하는 상황인 경우
Postconditions	Exception 들에 대한 대처, 송금하는 사람과 받는 사람에 대한 확실한 구별. 입금 성공시에만 출금이 발생하게 처리.

FR-2: 출금

ID	UC-2-FR-1
Name	출금
Primary actor	출금자
Goal in context	어떠한 오류 없이 출금이 올바른 상황에서만 일어나게 하는 것.
Preconditions	<ol style="list-style-type: none"> 1. 출금 외의 상황들에 대한 구현의 완료. (돈이 사라지는 일은 없다. 출금이 필요한 상황들에 대한 구현이 완료되어 있어야 한다.) 2. 돈을 출금하는 것에 대한 데이터 처리는 완벽하게 구현되어 있어야 한다. (돈만 빠져나가고 그에 대한 처리가 이루어지지 않는 일은 없어야 한다.) 3. 사용자 피아식별. (로그인, 비밀번호, 사용자 식별 정보) 4. 사용자가 하는 입력은 완료된 상황이다.
Scenario	<ol style="list-style-type: none"> 1. 출금자 가 출금 요청 2. 출금할 은행에 요청 전달 후, 다시 출금자에게 전달 3. 요청 데이터가 성공적으로 반환 시 출금.
Exception	<ol style="list-style-type: none"> 1. data의 손실 2. 사용자 잔액 부족 3. Atm 기기의 잔액 부족 4. 외적 요인으로 데이터가 전달되지 못하는 상황
Postconditions	실질적인 출금은 그에 대한 데이터가 완벽하게 처리된 상황 이후에서만 발생해야 한다. 어디서 (from) , 어디로 (to) 에 대한 정보가 명확하게 저장되어야 한다.

3.2 FR-3: 이체

ID	UC-3-FR-1
Name	이체
Primary actor	입금자, 수령자
Goal in context	송금자가 돈을 송금하면, 그 돈이 받는 사람에게 전해지는 것
Preconditions	<ol style="list-style-type: none"> 1. 송금하는 사람의 피아 식별. (로그인, 비밀번호, 사용자 식별 정보) 2. 받는 사람의 피아 식별. (로그인, 비밀번호, 사용자 식별 정보) 3. 자산의 이동과정에서 누락이 없음.
Scenario	<ol style="list-style-type: none"> 1. 이체하는 사람을 식별한 뒤, 이체 받을 사람에게 요청 전달 2. 입금 받는 사람 쪽의 결과 (실패 or 성공)를 다시 입금한 사람에게 전달한다. 3. 만약 요청이 실패하게 되면, 출금하지 않고, 성공 시 출금한다.

Exception	<ol style="list-style-type: none"> 1. Data의 손실 2. 외적인 문제(네트워크 등) 로 정보가 손실 된 경우 3. 송금한 사람 혹은 송금 받을 사람 측에서 변경, 혹은 추가된 데이터를 올바르게 반영하지 못하는 경우 4. 사용자가 하는 입력은 완료된 상황이다.
Postconditions	Exception 들에 대한 대처, 송금하는 사람과 받는 사람에 대한 확실한 구별. 입금 성공시에만 출금이 발생하게 처리.

4. Interface Requirements(인터페이스 요구사항)

4.1 User Interfaces(사용자 인터페이스)

ID	UI-1
Name	로그인 인터페이스
Goal in context	사용자가 자신의 개인정보를 통해 로그인할 수 있게 한다.
Description	<ol style="list-style-type: none"> 1. 사용자가 자신이 아이디 계정 인터페이스에서 생성한 아이디와 비밀번호를 통하여 로그인할 수 있게 한다. 2. 다른 사이트의 로그인 정보를 가져와서 자신을 식별한 뒤 로그인 할 수 있다. 3. 로그인에 성공하면, 개인정보를 DB에 보내며 관리 인터페이스로 이동시킨다.

ID	UI-2
Name	관리 인터페이스 (GUI)
Goal in context	다양한 기능들을 구현한 인터페이스를 클릭 한 번으로 사용하게 모아 놓아 편리함을 제공.
Description	<ol style="list-style-type: none"> 1. 로그인 한 사용자가 계좌 등록, 입출금, 잔액 확인 등의 부가적인 기능들을 손쉽게 사용하도록 만든 중앙 인터페이스. 2. 각 기능들을 카테고리 별로 분류해 놓고, 원하는 기능을 클릭하면 해당 기능이 구현된 인터페이스로 이동하여 해당 작업에 최적화된 인터페이스에서 작업을 수행하도록 처리한다. 3. GUI 시스템을 사용하여, 사용자가 작업을 수행하기 쉽게 구현한다.

ID	UI-3
Name	입출금 인터페이스
Goal in context	이체 대상, 출금될 계좌 등 사전 정보를 입력하여 입금, 출금, 이체 서비스를 제공한다.
Description	<ol style="list-style-type: none"> 1. 관리 인터페이스에서 입출금 기능을 선택하면 해당 창으로 넘어오게 구현이 되어있기에, 입출금 시스템에 최적화하여 구현한다. 2. 이체, 출금, 입금 등의 작업을 진행할 대상 (본인 계좌, 상대 계좌 등)에 대한 정보를 전부 수집한 뒤, 유저가 선택한 서비스에 맞는 기능을 제공한다. 3. 실제 기능을 수행하기 전, 사용자에게 대한 추가적인 보안 인증을 진행하기 위해 보안 인터페이스로 이동하여 2차 인증 (계좌 비밀번호 등)을 진행한 뒤 다시 돌아와 요청한 기능을 수행한다.

ID	UI-4
Name	보안 인터페이스
Goal in context	인증이 필요한 기능을 수행하기 전, 마지막 인증을 수행하여 보안 강화
Description	<ol style="list-style-type: none"> 1. 인증이 필요한 특정한 요청 (이체, 출금 등)을 수행하기 전, 마지막으로 사용자 인증을 진행하는 인터페이스. 2. 계좌 비밀번호, 혹은 인터넷 뱅킹 비밀번호를 통하여 사용자를 식별한다. 3. 성공적으로 보안을 통과했다면, 보안 인증을 요청한 인터페이스에 보안인증에 성공했다는 정보를 다시 전달하며 보안 인터페이스를 요청한 인터페이스로 돌아간다.

4.2 Hardware Interfaces(하드웨어 인터페이스)

ID	HI-1
Name	네트워크 송, 수신 장치
Goal in context	어디서든 데이터를 송, 수신 받을 수 있게 만든다.
Description	<ol style="list-style-type: none"> 1. 데이터를 전달받고 전해줄 수 있는 송 수신 장치를 통하여 어느 위치 에서나 atm을 통해 자산에 변동이 일어나게 되면 웹 서버에 전달할 수 있게 만든다. 2. 이때 atm 서비스 외의 외부정보들(새로운 광고, 혹은 날씨 등 실제 기능과 관계되진 않지만, 유익한 정보)도 함께 가져와 다양한 데이터를 atm 기기에 전달해 준다.

ID	HI-2
Name	터치 스크린
Goal in context	유저와 최상단에서 상호작용할 수 있는 디바이스 구축.
Description	<ol style="list-style-type: none"> 1. 터치스크린을 통하여 유저가 로그인을 할 수 있게 만들어준다. 2. 입금, 출금, 이체 등 핵심적인 작업 및 UI를 스크린으로써 상호작용하게 만들어준다.

ID	HI-3
Name	<i>Memory buffer</i>
Goal in context	동시에 여러가지 요청이 들어와도 모든 요청을 차례로 수행할 수 있어야한다.
Description	<ol style="list-style-type: none"> 1. 버퍼에 모든 요청들을 쌓아 놓고, 들어온 순서대로 요청을 처리하여 여러 요청이 쌓여 있더라도 모든 요청들에 대한 처리를 진행할 수 있어야 한다. 2. 버퍼 overflow, 혹은 buffer와 Memory 간의 inconsistency 와 같은 critical한 문제에 대한 대비가 되어있어야 한다.

ID	HI-4
Name	Cache memory
Goal in context	중요한 기능 들에 대한 처리 시간을 최소화해야 한다.
Description	<ol style="list-style-type: none"> 1. 가장 주요한 기능 (입금, 출금, 이체 등) 들에서 일어나는 반복적이고 용량이 큰 작업들은 Cache 메모리에 저장해 놓고 빠른 시간 안에 처리하게 만들어 줌으로써 반복적인 작업의 부하를 최소화 한다. 2. LRU 와 같은 효율적인 캐시 전략을 활용하여 한정된 캐시 용량에서 최대의 효율을 뽑아낸다.

4.3 Software Interfaces(소프트웨어 인터페이스)

ID	SI-1
Name	서버
Goal in context	사용자 정보를 주고받고, 필요한 정보는 데이터 베이스에 저장한다.
Description	<ol style="list-style-type: none"> 1. UI 파트에서 전달한 요청들을 받아 처리한 뒤, 올바른 곳에 전달한 뒤, 수정 혹은 새롭게 추가된 데이터들을 찾는다. 2. 매 데이터 transfer 과정 마다, 보내기 전에 요청을 보낸다. 요청을 받아 처리해야할 목적지에서 데이터를 온전히 받을 수 있는 상황이라면, 해당 요청을 돌려주어 데이터를 요청할 수 있는 상황임을 알려 준다. 3. 만약 유저 정보 수정(인증 이메일, id, password, 계좌 정보 수정) 등 변화가 보존되어야 할 변화라면, 영속성 컨텍스트에 변경된 정보를 저장한다.

ID	SI-2
Name	데이터 베이스
Goal in context	새로운 정보, 혹은 수정된 정보를 저장한다.
Description	<ol style="list-style-type: none"> 1. 기존에 데이터베이스에 담긴 정보 중 수정된 정보, 혹은 새롭게 추가된 정보들 중, 저장되어야 할 정보들을 반영한다. 2. 데이터 베이스의 경우 서버에서 변경되거나 추가된 정보를 database에 반영하기 전에, 영속성 컨텍스트에 저장하고, transaction 단위로 commit을 한다. 3. 만약 어떤 정보가 database에 commit 하는 중 오류가 생기면, 해당 데이터를 포함하는 transaction 전체를 rollback 한 뒤, 다시 등록하는 작업을 수행하는 것으로, 잘못된 정보가 database에 바로 반영되는 것을 방지한다.

5. Nonfunctional Requirements(비기능적 요구사항)

5.1 Performance Requirements(성능)

각 성능/안전성/보안/품질 별로 중요한 비기능적 요구사항 작성

ID	PR -1
Name	Response time
Goal in context	어떤 요청에 대한 응답시간은 2 초를 넘어가면 안 된다.
Description	<ol style="list-style-type: none">1. 유저가 UI를 통하여 어떠한 입력을 주면 그에 대한 응답은 요청을 모두 처리한 것과 별개로 빠른 시간안에 수행 되어야한다.

ID	PR -2
Name	throughput
Goal in context	동시에 여러 일을 처리할 수 있어야한다.
Description	<ol style="list-style-type: none">1. 동시에 1000가지이상의 업무가 들어와도 처리할 수 있어야한다.2. 만약 과부하가 걸린다면 기존의 정보들을 손실없이 유지할 수 있어야 한다.3. 시스템 복구에 소요되는 시간은 3시간을 넘어서 안된다.

5.2 Safety Requirements(안전성)

ID	SFT -1
Name	데이터 failure detection
Goal in context	데이터가 이동, 저장될 때 의도치 않게 손실되거나 변경되었다면 감지할 수 있어야 한다.
Description	<ol style="list-style-type: none">1. 서버간 데이터 transfer 중, 데이터의 손실을 감지할 수 있어야한다.2. 데이터 베이스에 정보를 추가, 수정하는 중 잘못된 정보가 있다면 감지할 수 있어야 한다3. 감지한 오류를 failure handling 하는 영역에 전달해 줄 수 있어야 한다.

ID	SFT -2
Name	Failure handling
Goal in context	잘못된 정보를 감지했을 때, 올바른 정보를 다시 반환해줄 수 있어야 한다.
Description	<ol style="list-style-type: none"> 1. 데이터베이스에서 손상된 정보는 <i>rollback</i> 하여 다시 올바른 정보를 반영할 수 있게 한다. 2. 서버간 데이터 교환시에 손상된 정보를 <i>rollback</i> 후 다시 전달할 수 있어야 한다.

5.3 Security Requirements(보안)

ID	SEC -1
Name	보안 장치
Goal in context	의도적인 해킹을 하드웨어 적으로 방지한다.
Description	<ol style="list-style-type: none"> 1. 유저 사용 하드웨어에서 지문 인식, 얼굴인식, 혹은 비밀번호 가림막 등의 추가적인 장치를 추가하여 보안을 강화한다.

ID	SEC-2
Name	유저 식별
Goal in context	각 유저를 식별할 수 있는 방법을 만든다.
Description	<ol style="list-style-type: none"> 1. 계정, 혹은 계좌, 주민번호 등 각 유저들이 자신만이 알 수 있는 개인 정보를 통해서 자신을 식별하게 함으로써 해당 정보를 알지 못하는 타인들이 접근하는 것을 방지한다.

ID	SEC-3
Name	가상 키보드 보안
Goal in context	
Description	<ol style="list-style-type: none"> 1. 가상 키보드를 사용하는 하드웨어의 경우, 터치 시 버튼이 눌리는 이펙트를 동시에 여러가지 자판에 주는 것으로 다른 사람이 사용자의 중요한 정보를 눈으로 읽어드리는 것을 방지한다. 2. 가상 키보드의 배열을 랜덤하게 재배치하여 해킹 프로그램에 혼선을 준다.

ID	SEC-4
Name	계정 lock
Goal in context	해킹 의심 상황이 발생하면, 계정을 lock 하여 추가적인 시도를 미연에 방지.
Description	<ol style="list-style-type: none"> 1. 로그인, 혹은 중요 작업 전 인증 페이지에서 일정 횟수 이상 잘못된 입력을 수행한다면 해킹 시도가 있는 것으로 판단하여 계정에 lock을 걸어 놓고 사용자가 등록한 이메일과 핸드폰에 해킹 감지를 알린다. 2. 만약 사용자가 사용하지 않았던 기기나 등록하지 않았던 기기에서 중요한 요청이 발생한다면, 기존 인증 프로세스를 다시 한번 수행하게 하여 해킹을 사전에 방지한다.

5.4 Software Quality Attributes(품질)

ID	SQ-1
Name	Availability
Goal in context	
Description	<ol style="list-style-type: none"> 1. 입금, 송금, 출금과 같이 실질적 자산에 변동이 있는 중요한 기능들은 시스템이 필요한 시간에 잘 작동되어야 한다. 2. 실패 후 다시 동작할 때의 시간은 최소화되어야 한다.

ID	SQ-2
Name	Integrability
Goal in context	개별적인 시스템 들이 한가지 기능을 위해 올바르게 동작해야 한다.
Description	<ol style="list-style-type: none"> 1. 유저가 사용하는 ATM에서 어떠한 처리를 수행하면 서버에 그 정보가 넘겨지고, 서버에서 올바르게 처리한 뒤 저장되어야 할 정보를 데이터베이스에 저장하는 과정이 합일화 되어 구상한대로 잘 동작하여야 한다.