

# Web Project

Open Banking 웹 서비스

# 목차

01

## 프로젝트 개요

- 어플리케이션 소개
- 사용 소프트웨어
- 개발 환경 및 시스템 아키텍처

02

## 설계

- 기능 명세서
- ERD
- Class Diagram

03

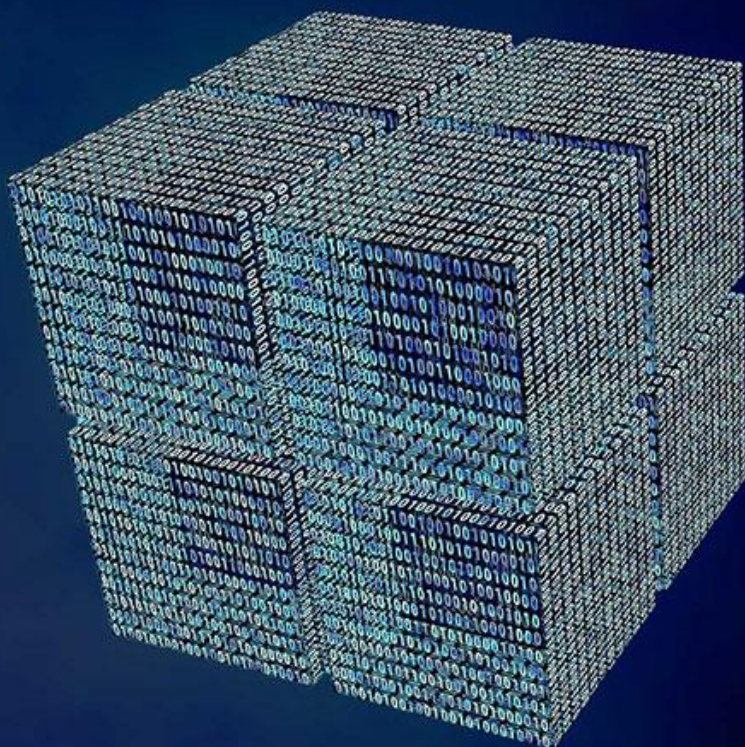
## 상세기능

- 계좌 이체 및 문자 알림 서비스
- 통합 계좌 서비스

04

## 마무리

- 리뷰







# 프로젝트 개요

- 어플리케이션 소개
- 사용 소프트웨어
- 개발 환경 및 시스템 아키텍처

# 소개



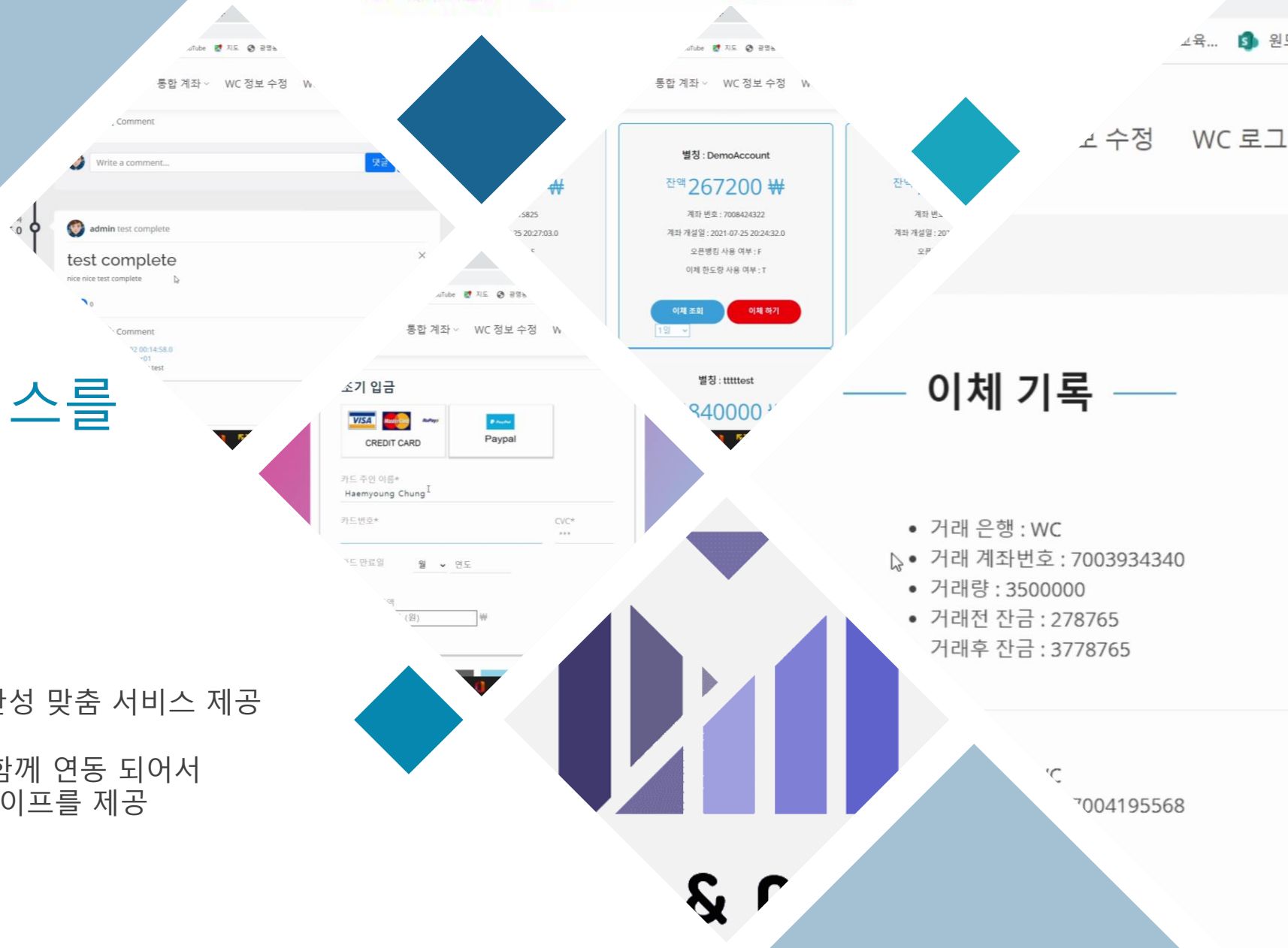
## 최고의 사용자 인터페이스를 자랑하는 온라인 뱅킹 Wells & Clarify

- 보안이 강하지만 간편한 이체 시스템
- 알림 메시지와 다양한 옵션을 통해 사용자 안성 맞춤 서비스 제공
- TVA, 세진저축은행, MG구마을금고 은행과 함께 연동 되어서 통합계정을 구축하고 편리한 온라인 금융 라이프를 제공

TVA

세진저축은행

MG구마을금고



### 이체 기록

- 거래 은행 : WC
- 거래 계좌번호 : 7003934340
- 거래량 : 3500000
- 거래전 잔금 : 278765
- 거래후 잔금 : 3778765

# Using Software

Front-End



Back-End



# 개발 환경

## 개발 HW 사양

Intel® Core™ i9-1098HK CPU  
RAM : 32GB



## 사용 Tool 및 SW

Tool : Eclipse IDE, Erwin, SQLdeveloper  
DB : Oracle  
Server: Tomcat9



## 개발 언어

HTML, CSS, JAVASCRIPT, JSP  
JSTL, Ajax, jQuery  
JAVA (OpenJDK 1.8)  
Oracle SQL

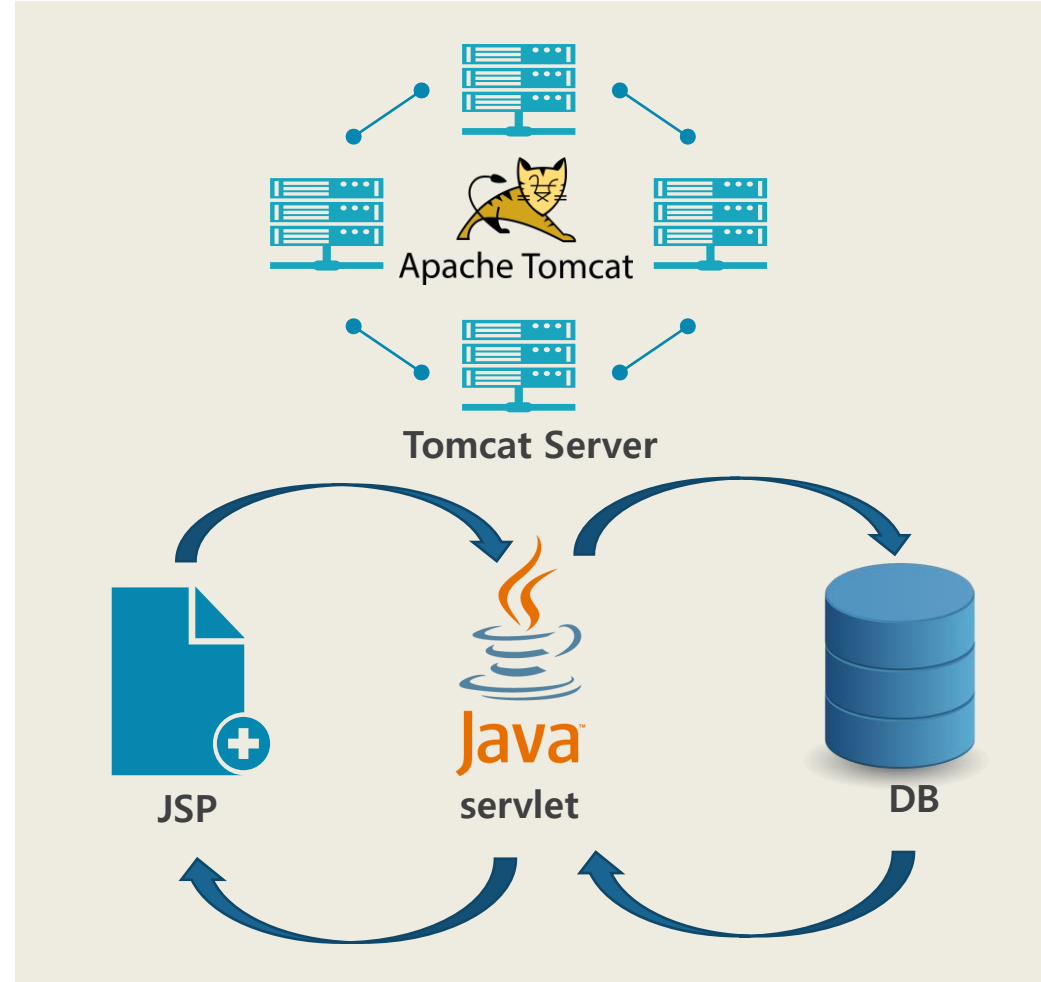
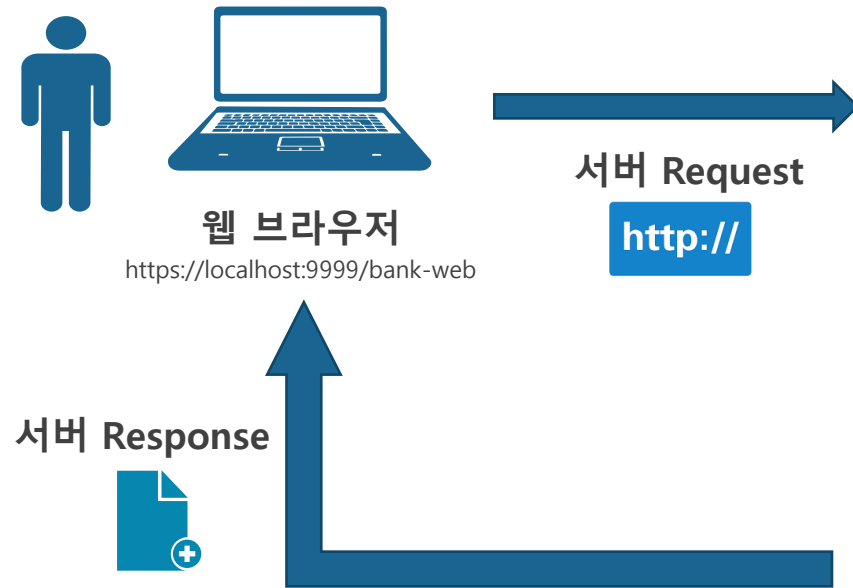


## 개발 인원 및 기간

개발 인원 : 4명  
개발 기간 : 16일



# 시스템 아키텍처





# 설계

- 기능 명세서
- ERD
- Class Diagram



# 기능 명세서 – 회원

업무	1st	2nd	기능설명
사용자	사용자 관리	회원가입	<ul style="list-style-type: none"> <li>• 입력정보 : 아이디, 비밀번호, 이름, 주민등록번호, 휴대전화번호, 이메일</li> <li>- Ajax를 활용하여 아이디 중복 체크</li> <li>- HTML 패턴 속성을 활용하여 전화번호, 이메일 문법 확인</li> <li>- 회원 정보 데이터베이스에 삽입</li> </ul>
		로그인	<ul style="list-style-type: none"> <li>• 카카오 로그인</li> <li>- 카카오 API를 활용해서 간편 로그인</li> <li>- 처음 보는 카카오 사용자는 자동으로 회원 정보 데이터베이스에 삽입</li> <li>• 일반 로그인</li> <li>- 아이디, 비밀번호 둘 다 일치 할 시 로그인</li> </ul>

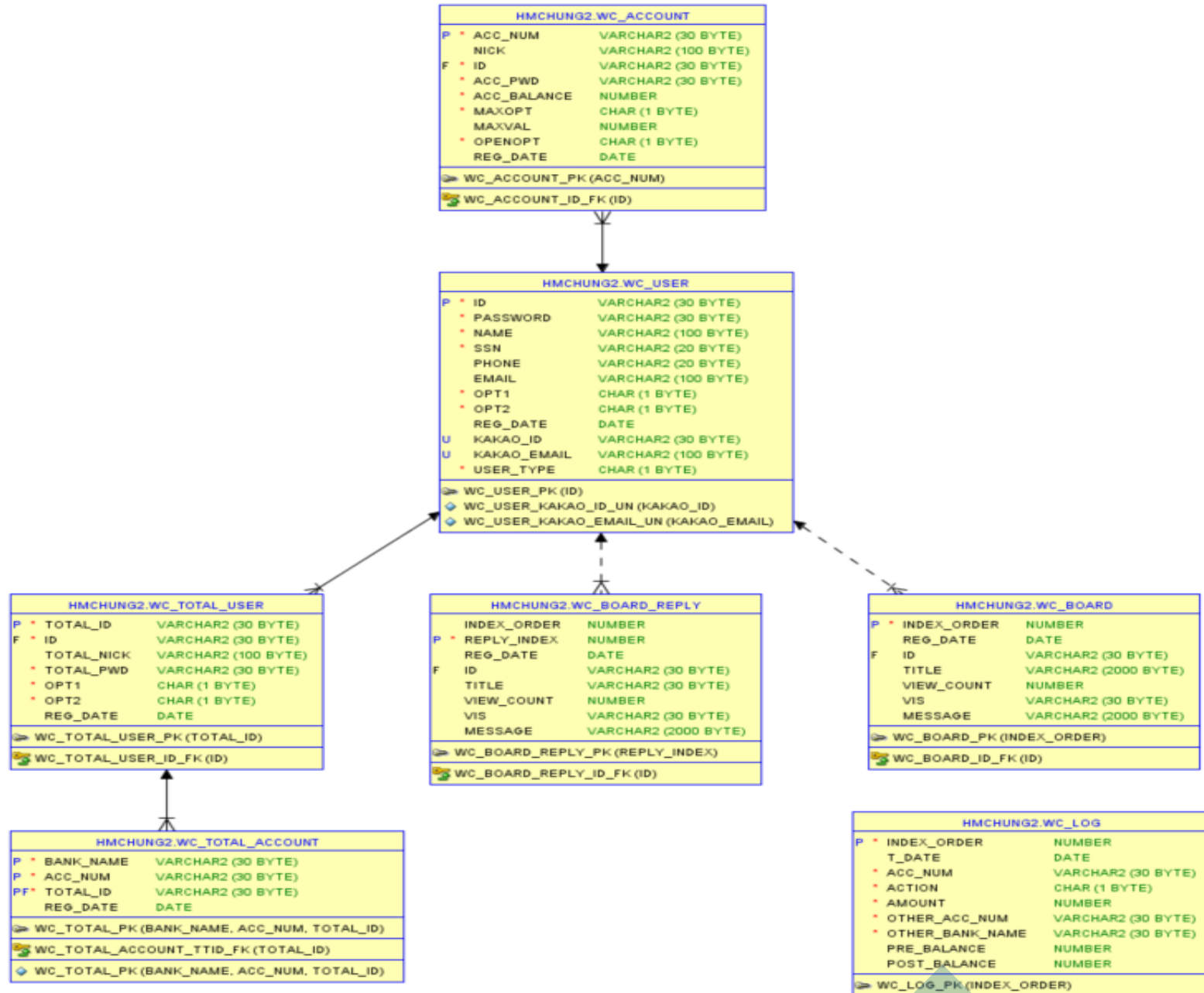
# 기능 명세서 – 개인 금융 업무

업무	1st	2nd	기능설명
개인 금융 업무	계좌 관리	계좌개설	<ul style="list-style-type: none"> <li>• 입력 정보 : 비밀번호, 주민등록번호, 카드 번호, 카드 비밀번호, 초기 입금 금액, 계좌 별칭</li> <li>- HTML 패턴 속성과 자바스크립트를 활용해서 입력 정보 유효성 확인</li> <li>- 입력 정보 이슈 발생시 모달로 이슈 내용 출력</li> <li>- 랜덤 계좌 번호 생성</li> <li>- 최대 이체 금액 설정</li> </ul>
		계좌 조회	<ul style="list-style-type: none"> <li>• 사용자가 보유한 모든 계좌 조회</li> <li>- 계좌 개설일, 현재 잔액, 계좌 설정, 계좌 별칭 정보 출력</li> </ul>
		계좌 이체	<ul style="list-style-type: none"> <li>• 입력 정보 : 계좌 비밀번호, 이체 금액, 보내는 계좌 번호 입력, 은행 이름</li> <li>- AJAX 를 활용해서 계좌 비밀번호 조회</li> <li>- 현재 잔액 또는 최대 이체 금액 설정 값보다 적은 이체 금액인지 확인</li> <li>- AJAX 를 활용해서 보내는 계좌번호의 사용자 정보 출력</li> <li>- AJAX 를 활용해서 문자 알림 서비스 실행</li> <li>- 한 개의 프로세스 이슈 발생시 모든 절차 롤백</li> </ul>
		거래 내역	<ul style="list-style-type: none"> <li>• 거래 시간 및 계좌 번호 정보를 입력해서 거래내역 정보 조회</li> </ul>

# 기능 명세서 – 게시글

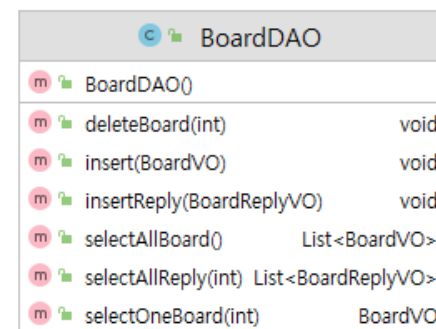
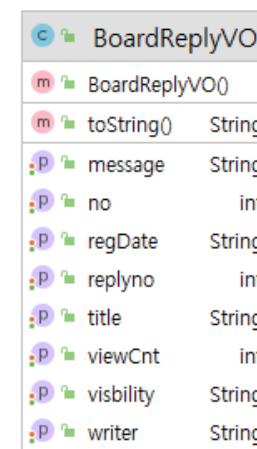
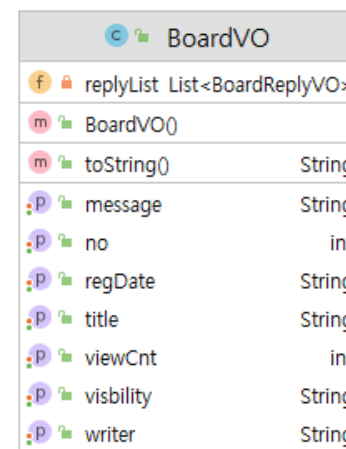
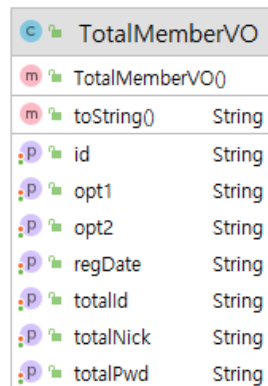
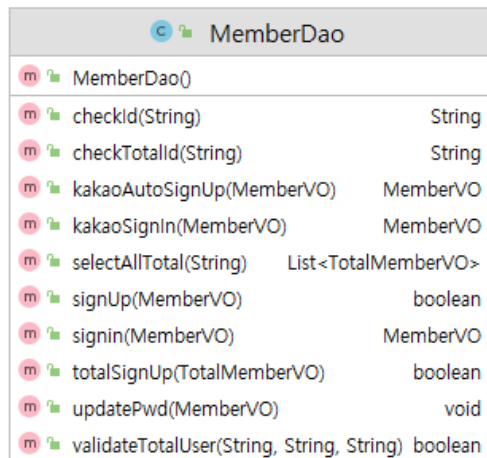
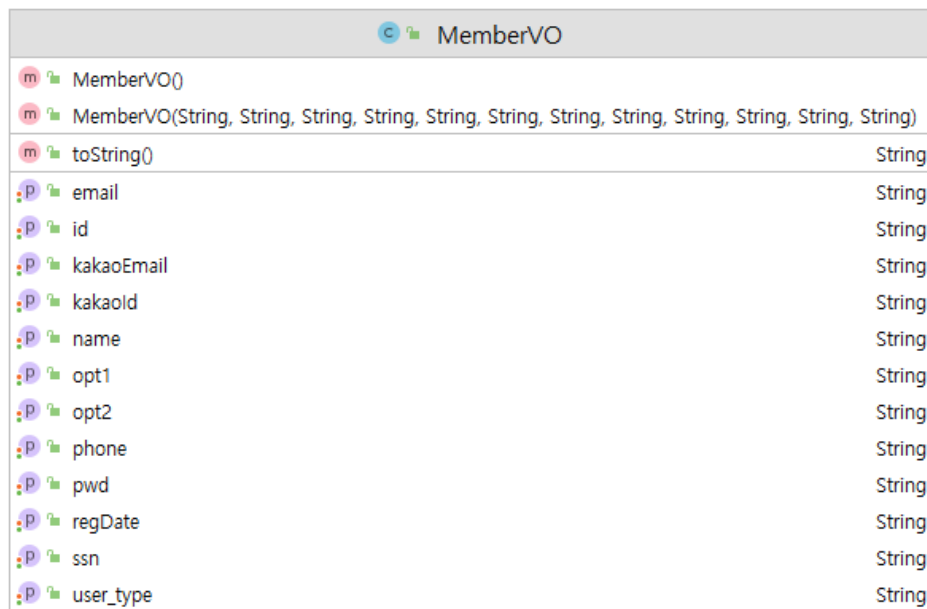
업무	1st	2nd	기능설명
게시판 관리	게시글 관리	게시글 조회	<ul style="list-style-type: none"> <li>게시글 한 페이지별 리스트 형식으로 3개의 게시글 및 게시글의 댓글 조회</li> <li>- 작성 아이디, 작성 시간 정보 출력</li> <li>- AJAX 를 활용해서 게시글 별 댓글 정보 출력</li> </ul>
		새 게시글 작성	<ul style="list-style-type: none"> <li>입력 정보 : 게시글 제목, 게시글 내용</li> <li>- 게시글 정보 데이터베이스에 삽입</li> </ul>
		게시글 댓글 작성	<ul style="list-style-type: none"> <li>입력 정보: 댓글 내용</li> <li>- AJAX 를 활용해서 댓글 정보 데이터베이스에 삽입 및 댓글 해당 페이지에 내용 출력</li> </ul>
		게시글 삭제	<ul style="list-style-type: none"> <li>입력 정보 : 비밀번호</li> <li>- AJAX 를 활용해서 댓글의 대한 사용자 비밀번호와 입력 비밀번호가 동일한지 확인</li> <li>- 올바른 비밀번호 입력 시 데이터베이스에서 게시글 정보 삭제</li> </ul>

# ERD





# Class Diagram – Member , Board Package



Powered by yFiles

Powered by yFiles

# Class Diagram – Account Package

AccountDAO		
m	AccountDAO()	
m	bsDeposit(Connection, String, int)	boolean
m	bsWithdraw(Connection, String, int)	boolean
m	checkSameAcc(String)	boolean
m	createAcc(AccountVO)	boolean
m	deposit(Connection, String, String, int)	boolean
m	getBSUserByID(String)	MemberVO
m	getMGUserByID(String)	MemberVO
m	getUserByID(String, String)	MemberVO
m	getWCUserByID(String)	MemberVO
m	insertBSDepositLog(Connection, String, String, String, int, int)	boolean
m	insertBSWithdrawLog(Connection, String, String, String, int, int)	boolean
m	insertDepositLog(Connection, String, String, String, String, int, int)	boolean
m	insertMGDepositLog(Connection, String, String, String, int, int)	boolean
m	insertMGWithdrawLog(Connection, String, String, String, int, int)	boolean
m	insertTotalAccount(AccountVO, String, String)	boolean
m	insertTotalWCAccount(AccountVO, String, String)	boolean
m	insertWCDepositLog(Connection, String, String, String, int, int)	boolean
m	insertWCWithdrawLog(Connection, String, String, String, int, int)	boolean
m	insertWithdrawLog(Connection, String, String, String, String, int, int)	boolean
m	mgDeposit(Connection, String, int)	boolean
m	mgWithdraw(Connection, String, int)	boolean
m	selectAccount(String, String)	AccountVO
m	selectAllAccounts(String)	List<AccountVO>
m	selectAllTotalAccounts(String)	List<TotalAccountVO>
m	selectBSAccount(String)	AccountVO
m	selectMGAccount(String)	AccountVO
m	selectWCAccount(String)	AccountVO
m	transaction(String, String, String, String, int, int, int)	Map<String, String>
m	wcDeposit(Connection, String, int)	boolean
m	wcWithdraw(Connection, String, int)	boolean
m	withdraw(Connection, String, String, int)	boolean

AccountVO		
m	AccountVO()	
m	AccountVO(String, String, String, String, int, String, int, String, String, String)	
m	toString()	String
p	accBalance	int
p	accNum	String
p	accPwd	String
p	id	String
p	maxOpt	String
p	maxVal	int
p	nick	String
p	openOpt	String
p	regDate	String

LogVO		
m	LogVO()	
m	toString()	String
p	accNum	String
p	action	String
p	amount	int
p	logDate	String
p	otherAccNum	String
p	otherBankNum	String
p	postBalance	int
p	preBalance	int

TotalAccountVO		
m	TotalAccountVO()	
m	toString()	String
p	accNum	String
p	bankName	String
p	regDate	String
p	totalId	String

LogDAO		
m	LogDAO()	
m	selectLogs(String)	List<LogVO>
m	selectLogs(String, String)	List<LogVO>

# Class Diagram – AJAX package

SendTextMsgServlet		
serialVersionUID	long	
SendTextMsgServlet()		
getBankName(String)	String	
service(HttpServletRequest, HttpServletResponse)	void	

CreateAccountProcessServlet		
serialVersionUID	long	
CreateAccountProcessServlet()		
generateDigits()	String	
service(HttpServletRequest, HttpServletResponse)	void	

CheckId		
serialVersionUID	long	
CheckId()		
service(HttpServletRequest, HttpServletResponse)	void	

RegisterAccToTotalServlet		
serialVersionUID	long	
RegisterAccToTotalServlet()		
service(HttpServletRequest, HttpServletResponse)	void	

ReplyMessageServlet		
serialVersionUID	long	
ReplyMessageServlet()		
service(HttpServletRequest, HttpServletResponse)	void	

FindAccountServlet		
serialVersionUID	long	
FindAccountServlet()		
service(HttpServletRequest, HttpServletResponse)	void	

GetUserInfoServlet		
serialVersionUID	long	
GetUserInfoServlet()		
service(HttpServletRequest, HttpServletResponse)	void	









TransactionProcessServlet		
serialVersionUID	long	
TransactionProcessServlet()		
service(HttpServletRequest, HttpServletResponse)	void	







CheckTotalId		
serialVersionUID	long	
CheckTotalId()		
service(HttpServletRequest, HttpServletResponse)	void	







ShowReply		
serialVersionUID	long	
ShowReply()		
service(HttpServletRequest, HttpServletResponse)	void	





DeleteCommentServlet		
serialVersionUID	long	
DeleteCommentServlet()		
service(HttpServletRequest, HttpServletResponse)	void	





# Class Diagram – Controller package





HandlerMapping	
 	mappings Map<String, Controller>
 	HandlerMapping()
 	HandlerMapping(String)
 	getController(String) Controller




SignUpController	
 	SignUpController()
 	checkNull(String) String
 	handleRequest(HttpServletRequest, HttpServletResponse) String





KakaoSignInProcessController	
 	KakaoSignInProcessController()
 	checkNull(String) String
 	handleRequest(HttpServletRequest, HttpServletResponse) String





BoardWriteProcessController	
 	BoardWriteProcessController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





WriteNewBoardController	
 	WriteNewBoardController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





TotalSignUpController	
 	TotalSignUpController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String




ViewLogController	
 	ViewLogController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

UpdateUserInfoProcessController	
 	UpdateUserInfoProcessController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





TotalSignUpProcessController	
 	TotalSignUpProcessController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

ViewBoardController	
 	ViewBoardController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





UpdateUserInfoController	
 	UpdateUserInfoController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





TransactionController	
 	TransactionController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





TotalContractController	
 	TotalContractController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





TotalTransactionController	
 	TotalTransactionController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





SignContractController	
 	SignContractController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





ViewTotalController	
 	ViewTotalController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





ViewEachTotalController	
 	ViewEachTotalController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String





ViewAccountController	
 	ViewAccountController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String


CreateAccountController	
 	CreateAccountController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String



SignUpProcessController	
 	SignUpProcessController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

SignInProcessController	
 	SignInProcessController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

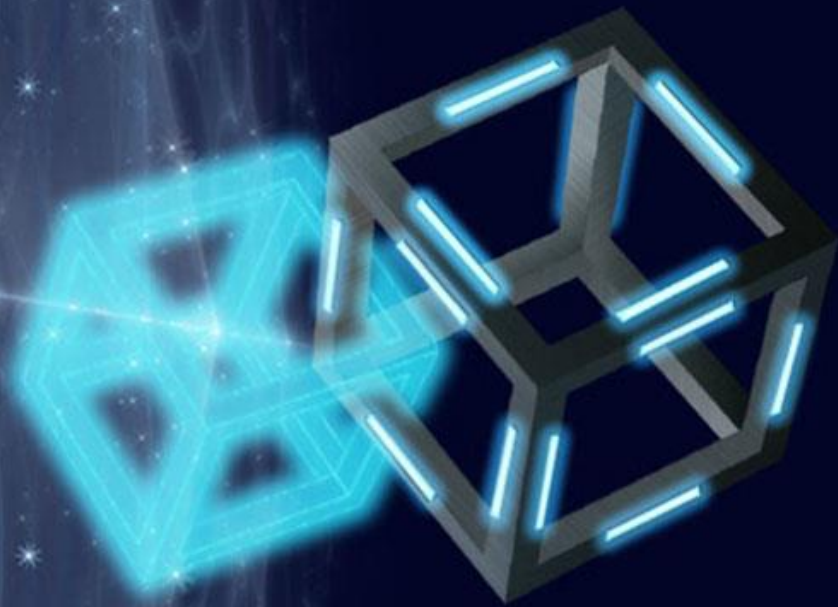
TotalAccountRegisterController	
 	TotalAccountRegisterController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

SignInController	
 	SignInController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

SignOutController	
 	SignOutController()
 	handleRequest(HttpServletRequest, HttpServletResponse) String

Controller	
 	handleRequest(HttpServletRequest, HttpServletResponse) String





# 주요기능

- 서비스 개요
- 주요 서비스 소개
- 이체 프로세스 상세 설명

# 서비스 개요



## 회원가입/ 로그인

회원가입 및 로그인 서비스



## 계좌 개설

새로운 계좌를 생성해주는 서비스



## 계좌 이체

간단한 인증과 보내는 계좌에 대한 정보를 입력 후 계좌 이체를 하는 서비스



## 계좌 조회

사용자가 보유한 모든 계좌들의 정보를 한눈에 볼 수 있는 서비스



## 게시판

문의 혹은 사용자들끼리 자유롭게 교류 할 수 있는 서비스

# 주요 서비스 소개 – 회원가입/ 로그인

Home 게시판 WC 계좌 ▾ 통합 계좌 ▾ 회원가입 로그인 시작하기

Get started with your free account

카카오로 로그인

OR

아이디 중복 확인

중복 확인이 필요합니다.

성

이름

이메일 @ naver.com naver.com ▾

010- 핸드폰 번호 : xxxx-xxxx 형식으로 입력하세요

주민등록번호 : xxxxxx-xxxxxx 형식으로 입력하세요

비밀번호 입력

비밀번호 다시 입력

Create Account

이미 회원 이십니까? 로그인

## 회원가입 서비스

AJAX를 활용하여 아이디 중복 체크/  
HTML 내부 패턴 속성을 활용해서 입력 값 제어

Wells & Clarify

please Sign In

아이디

비밀번호

로그인

카카오로 로그인

아직 회원가입을 안하셨나요? [회원가입 하기](#)

## 로그인 서비스

아이디 비밀번호 입력 혹은 카카오 API로  
로그인

# 주요 서비스 소개 – 계좌 개설

Wells & Clarify

홈 / WC 계좌 개설

### 계좌 개설 하기

다음 절차를 통해 계좌를 생성하세요

인증

계좌정보 생성

초기 입금

완료

#### 계좌 정보 생성

계좌 별칭

계좌 비밀번호 입력

계좌 비밀번호 확인

오픈뱅킹 등록 여부

이체 최대 금액 설정 여부

이체 최대 금액 량

10000000 원

Previous

Next Step

## 계좌 설정 서비스

사용자가 직접 옵션을 선택해서 계좌를 생성해주는 서비스

### 계좌 개설 하기

다음 절차를 통해 계좌를 생성하세요

인증

계좌정보 생성

초기 입금

완료

#### 초기 입금

VISA

Master Card

Amex

CREDIT CARD

Paypal

Paypal

카드 주인 이름\*

카드 번호\*

CVC\*

카드 만료일

월

연도

초기 입금 금액

숫자로만 작성 (원)

₩

Previous

Confirm

## 초기 입금 및 계좌 생성 서비스

(가상의)카드 정보와 금액을 지정해서 계좌에 초기 입금 후 계좌 개설 완료



# 주요 서비스 소개 – 계좌 조회

Wells & Clarify

홈 / 계좌조회 / 계좌이체

— 계좌 이체 —

계좌 비밀번호 확인 -> 금액 입력 -> 보내는 계좌 번호 확인 -> 보내기 순으로 진행주세요

WC 은행

별칭 : ShowingDemo

잔액 3778765 ₩

계좌 주인 : 이지유

계좌 번호 : 7003915825

계좌 개설일 : 2021-07-25 20:27:03.0

오픈뱅킹 사용 여부 : F

이체 한도량 사용 여부 : T

이체 한도량 : 9000000

3500

변경

다른 계좌

계좌인증 완료

보내기

받는 사람

은행 : WC은행

7008424322

받는분 : 이지유

변경

## 이체 서비스

보내는 계좌 비밀번호, 받는 계좌 번호, 금액을 작성하고 계좌 이체를 해주는 서비스

SKT 9:22 80%

정해명  
010-9119-2831

495000원이 이체되었습니다  
오전 6:15

[Web발신]  
7007783649 W&C  
계좌 에서 7004195568  
W&C 은행 계좌로  
3333원이 이체  
되었습니다  
오전 6:25

[Web발신]  
7003934340 W&C  
계좌 에서 7003915825  
W&C 은행 계좌로  
3500000원이 이체  
되었습니다  
오후 2:21

[Web발신]  
7007783649 W&C  
계좌 에서 7003934340  
W&C 은행 계좌로  
6667원이 이체  
되었습니다  
오후 3:11

## 알림 서비스

사용자의 핸드폰 번호로 이체 알림 SMS 메시지 전송 서비스

# 주요 서비스 소개 – 계좌 개설

— 이지유님의 계좌 조회 —

이체 가능 및 이체 역사 조회

별칭 : ShowingDemo

잔액 3778765 ₩

계좌 번호 : 7003915825

계좌 개설일 : 2021-07-25 20:27:03.0

오픈뱅킹 사용 여부 : F

이체 한도량 사용 여부 : T

이체 조회

이체 하기

1월 ▼

별칭 : DemoAccount

잔액 267200 ₩

계좌 번호 : 7008424322

계좌 개설일 : 2021-07-25 20:24:32.0

오픈뱅킹 사용 여부 : F

이체 한도량 사용 여부 : T

이체 조회

이체 하기

1월 ▼

별칭 : lastOne

잔액 725000 ₩

계좌 번호 : 7007472165

계좌 개설일 : 2021-07-22 22:41:57.0

오픈뱅킹 사용 여부 : T

이체 한도량 사용 여부 : F

이체 조회

이체 하기

1월 ▼

별칭 : lastOne

잔액 785000 ₩

계좌 번호 : 7004859778

계좌 개설일 : 2021-07-22 22:41:52.0

오픈뱅킹 사용 여부 : T

이체 한도량 사용 여부 : F

이체 조회

이체 하기

1월 ▼

별칭 : tttttest

잔액 840000 ₩

계좌 번호 : 7005627978

계좌 개설일 : 2021-07-22 22:39:46.0

오픈뱅킹 사용 여부 : T

이체 한도량 사용 여부 : T

이체 조회

이체 하기

1월 ▼

별칭 : test123

잔액 779010 ₩

계좌 번호 : 7003615990

계좌 개설일 : 2021-07-22 22:34:05.0

오픈뱅킹 사용 여부 : T

이체 한도량 사용 여부 : T

이체 조회

이체 하기

1월 ▼

## 잔액 조회 서비스

사용자의 모든 계좌 정보를 한 눈에 조회 할 수 있는 서비스

— 이체 기록 —

입금

이체일 : 2021-08-01 23:21:02.0

- 거래 은행 : WC
- 거래 계좌번호 : 7003934340
- 거래량 : 3500000
- 거래전 잔금 : 278765
- 거래후 잔금 : 3778765

출금

이체일 : 2021-07-26 10:36:32.0

- 거래 은행 : WC
- 거래 계좌번호 : 7004195568
- 거래량 : 1235
- 거래전 잔금 : 280000
- 거래후 잔금 : 278765

출금

이체일 : 2021-07-25 21:22:34.0

- 거래 은행 : WC
- 거래 계좌번호 : 7004195568
- 거래량 : 3000
- 거래전 잔금 : 283000
- 거래후 잔금 : 280000

출금

이체일 : 2021-07-25 21:21:31.0

- 거래 은행 : WC
- 거래 계좌번호 : 7004195568
- 거래량 : 5000
- 거래전 잔금 : 288000
- 거래후 잔금 : 283000


## 거래 내역 조회 서비스

한번에 혹은 선택한 날짜만큼 거래내역을 조회하는 서비스


# 주요 서비스 소개 – 통합계좌

### 통합 계정

새 통합계정 만들기를 클릭해서 새로운 계정을 만드세요



새 통합계정 만들기



별칭 client01Total

- 아이디 : client01Total
- 개설일 : 2021-08-02 00:09:19.0
- 약관 설정1 : T
- 약관 설정2 : T

조회

### 이지유님의 CLIENT01TOTAL계정

이체하기 눌러서 이체하러 가기

새로운 계좌 등록

☒ WC

돌아가기

등록하기

WC 은행

계좌 번호 : 7007783649

계좌 개설일 : 2021-08-02 00:09:50.0

이체 하기

WC 은행

계좌 번호 : 7003915825

계좌 개설일 : 2021-08-02 00:09:37.0

이체 하기

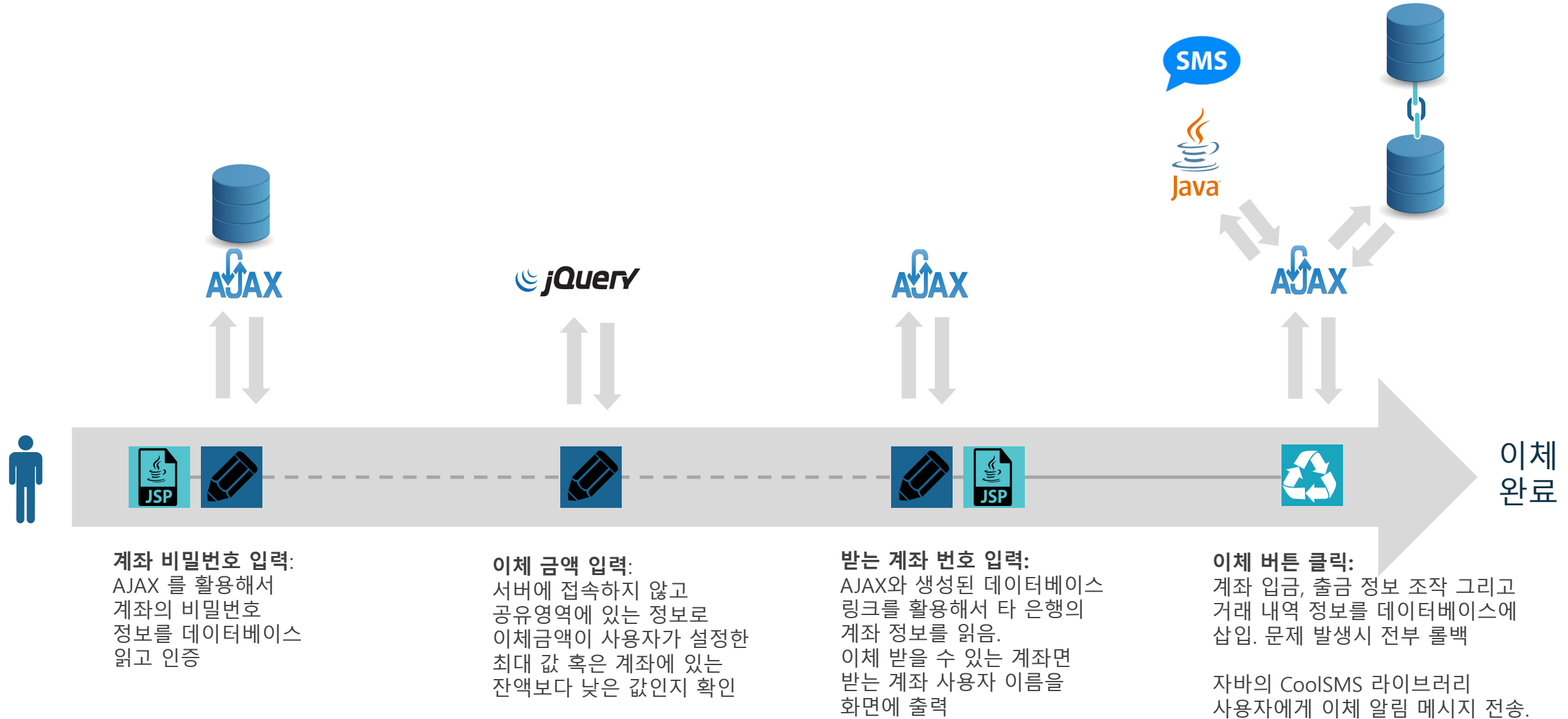


## 통합계좌 서비스

W&C 은행, 세진저축은행, MG구마을금고를 한곳에 모아서 편리하게 금융 관리를 해주는 서비스.

다른 은행 계좌도 W&C 에서 편하게 계좌 이체 등 잔액 조회를 편리 하게 할 수 있는 기능을 제공.

# 이체 프로세스 상세 설명



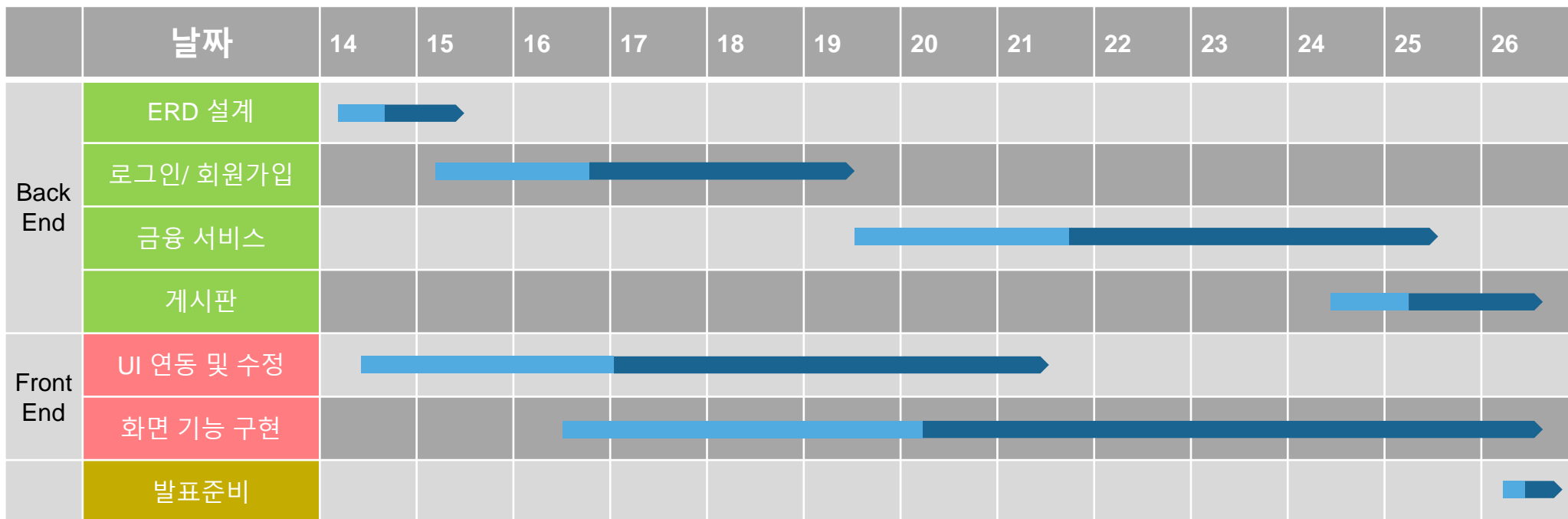




# 마무리

- 간트차트
- 리뷰

## Gantt Chart



# 프로젝트 리뷰



오픈 banking 웹프로젝트를 마무리 하며...

아쉬움



만족감



## Things to Say

전반적으로 아쉬움이 많은 프로젝트였다. 하지만 아쉬움이 많은 만큼 문제점을 정확히 파악 했고 다음 프로젝트때는 잘 할 자신이 생겼다.

배운 점 1:

**개발하기 앞서 과연 기능이 구현 난이도 만큼이나 구현 가치가 있는가를 명확히 구분해야 한다.** 예를 들어서 DB link를 위해 VPN 서버를 구축하겠다고 이들을 사용했다. 처음부터 리모트 데스크탑을 사용해서 네트워크 연동해도 전혀 프로젝트에 지장이 없다. 기술 구현에 대한 욕망을 줄이고 구현 난이도와 가치를 비교해서 이성적인 판단을 해야한다.

배운 점2:

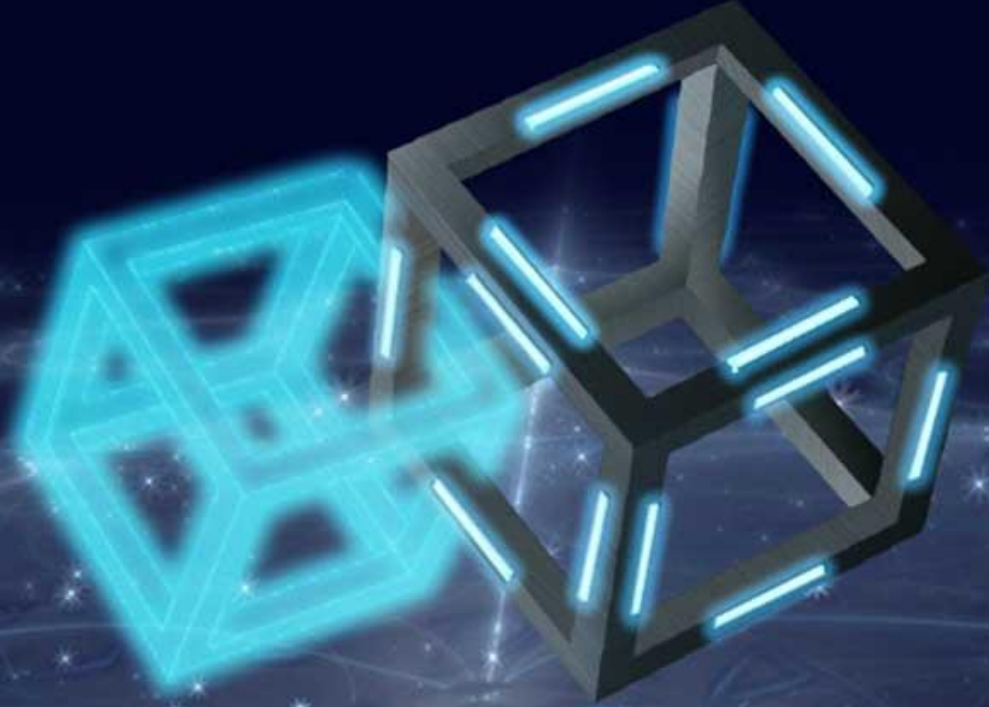
**Back-End 기능 개발을 제일 우선적으로 해야한다. Front-end 는 개발은 끝이 없다.** 동영상 편집이 끝이 없는 것처럼 웹페이지 디자인 개선도 끝이 없다. 결국 프로젝트 막바지에 너무 급하게 기능을 구현을 하게 되어서 너무 아쉬웠고 발표 준비 할 시간도 생기지 않았다. 다음 프로젝트에서는 먼저 웹 플랫폼 전체 뼈대를 먼저 구축하고 마지막에 디자인에 신경을 쓸 예정이다.

배운 점3:

**개발에 있어서 가장 중요한점은 시간관리다.** 세부적인 계획을 마련하지 않고 추상적으로 개발 플랜을 하면 굉장히 위험하다. "오늘 이정도 했으면 많이 했다." 라고 느끼고 하루의 작업을 멈추는 게 아니라 전체 프로젝트에서 몇 퍼센트가 완료 되었고 현재 몇일이 남았다고 매일 숫자로 계산을 해야한다. 느낌에 의존을 하게 되면 모든 기능을 완료 못할 수 있다.

결론:

어려움만큼 배운 점도 많고 개발 능력도 많이 향상 되었다. 하지만 이 프로젝트로 가장 크게 얻은 것은 개발 능력 향상이 아니라 위에 언급한 문제점들에 대한 인지와 개선 방향이다. 이 부분이 가장 만족스럽고 다음 프로젝트때에는 훨씬 완성도가 높은 결과물을 만들 자신이 생겼다.



감사합니다

정해명