



국민대학교
소프트웨어융합대학
소프트웨어학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	여기모영
팀 명	꼭꼭
문서 제목	캡스톤디자인 3팀-결과보고서

Version	1.5
Date	2022-05-24

팀원	안 성열 (조장)
	김 상윤
	민 태식

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 "여기모영"을 수행하는 팀 "꼭꼭"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "꼭꼭"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	캡스톤디자인 3팀-결과보고서.doc
원안작성자	김상윤, 민태식, 안성열
수정작업자	김상윤, 민태식, 안성열

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2022-05-04	안성열	1.0	최초 작성	목차 및 1번 내용 작성
2022-05-16	안성열	1.1	초안 작성	2, 3번 내용 작성
2022-05-22	안성열	1.2	추가 작성	4번 내용 작성
2022-05-23	민태식	1.3	추가 작성	4번 내용 작성
2022-05-23	김상윤	1.4	추가 작성	3,4 번 내용 작성
2022-05-24	안성열	1.5	최종 수정	맞춤법 수정 및 마감

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

목차

1	기존 프로젝트	4
1.1	기존 프로젝트 폐기 이유	4
2	프로젝트 개요	5
2.1	프로젝트 추진 배경 및 필요성	5
3	개발 내용 및 결과물	5
3.1	목표	5
3.2	개발 내용 및 결과물	6
3.2.1	앱 어플리케이션(React-Native)	6
3.2.2	웹 어플리케이션 서버(Node.js)	25
3.2.3	AWS EC2	31
3.2.4	데이터베이스	31
3.2.5	블록체인	34
4	자기평가	36
4.1	김상윤	36
4.2	민태식	37
4.3	안성열	37
5	부록	38
5.1	사용자 메뉴얼	38
5.1.1	블록체인 채굴 참여	38
5.2	운영자 메뉴얼	39
5.2.1	블록체인 네트워크 구성	39
5.2.2	서버 개설	41

	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

1 기존 프로젝트

기존 프로젝트명은 '국민을국회로'이다. '국민을국회로'는 입법예고제도를 통해 공시된 발의 법안을 사용자에게 제공하고, 블록체인 기반의 투표를 통해 여론을 형성하는 모바일 어플리케이션 형태의 서비스이다.

입법예고제도는 발의된 법안이 심사를 통해 최종적으로 공포되는 과정 전까지 국민들의 의견을 수렴하기 위해 마련된 제도이다.

기존 프로젝트의 제작 배경은 다음과 같다.


- 새로 발의된 법안은 '국회 웹사이트', '정부 웹사이트' 두 가지의 방법으로 게시된다. 새로 발의된 법안은 하나의 웹 사이트로 정리되지 않았다.
- 각 웹사이트에는 국민의 의견을 수렴하는 기능이 마련되어 있지만, 현재 찬성, 반대 등의 의견이 90% 이상을 차지한다. 기능의 본 목적과 사용자의 요구사항이 일치하지 않는다.

기존 프로젝트에서는 다음과 같은 해결책을 마련했다.

- 국회 홈페이지, 정부 홈페이지의 게시된 법안을 크롤링하여 모바일 어플리케이션의 형태로 사용자에게 제공한다.
- 의견 수렴 기능을 '찬반 투표 기능'과 '세부 의견 작성'으로 기능을 분리한다. 블록체인을 사용하여 찬반 투표 과정에서 투표 결과와 비밀 투표를 가능하게 했다.

1.1 기존 프로젝트 폐기 이유

- 현실적인 문제 : '법'이라는 주제의 진입장벽이 높기 때문에 사람들의 관심을 끌기 어렵고 타겟 사용자의 수가 극히 적다. 또한, 현재 국회나 정부에서 국민들의 의견을 필수적으로 반영해야 한다는 제도가 없어 '입법예고제도'의 실효성에 의문이 있다.
- 기술적인 문제 : 블록체인 기술을 이용하여 투표 조작을 막을 수는 있지만, 부 계정을 이용한 여론 조작을 막을 수는 없다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

2 프로젝트 개요

새로운 프로젝트명은 '모두모영'이다. '모두모영'은 모임의 회비 내역을 공유하는 모바일 어플리케이션이다. 모임의 총무는 결제 내역을 추가하고 영수증 실물을 업로드 한다. 모임의 모든 참가자들은 이를 조회할 수 있다. 또한, 사용자는 모임의 회비 내역을 저장하는 방식을 일반 DB와 블록체인으로 선택할 수 있다.

2.1 프로젝트 추진 배경 및 필요성

매년 모든 대학교의 학생회에서 학생 회비와 관련된 잡음이 지속되고 있다. 학생회의 목적에 맞지 않는 회비를 사용하거나, 회비 사용 내역에 조작이 의심되는 경우가 다수 발견된다.

해당 문제점은 총무가 회비 사용 내역을 분실하거나, 내역 집계 중에 실수를 저지르거나, 회비 사용 내역을 곧바로 공개하지 않거나, 현실적으로 회비 사용 내역을 모두에게 공개하기 어렵기 때문에 발생한다. 본 프로젝트는 총무가 모임의 회비를 편하게 관리하고, 모임 참여자가 회비 내역을 모두 조회하여 서로 간의 갈등을 해결하기 위해 제작되었다.

3 개발 내용 및 결과물

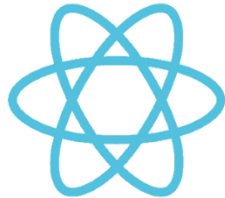
3.1 목표

- 사용자에게 모바일 어플리케이션 형태의 서비스를 제공하기 위해 앱 어플리케이션 서버(React-Native)를 구성한다.
- 클라이언트의 요청을 처리하기 위해 웹 어플리케이션 서버(Node.js)를 구성한다.
- 고정된 IP, PORT 를 사용하기 위해 웹 어플리케이션 서버를 AWS EC2에서 구동한다.
- 사용자에게 모임의 정보 저장 방식을 2가지로 제공하기 위해 일반 DB와 블록체인 네트워크를 구축한다.
- 모임 사용 내역의 신뢰성을 부여하기 위해 실물 영수증의 데이터를 저장한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

아래는 프로젝트에 사용된 도구들을 이미지로 나타낸 것이다.

Front



React-Native

Back



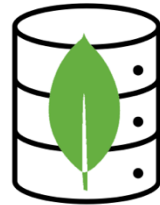
AWS EC2



Blockchain
Network



Node.js



Mongo DB


3.2 개발 내용 및 결과물

3.2.1 앱 어플리케이션(React-Native)

개발 내용을 소개하기에 앞서서, 프로젝트를 진행하였던 개발 환경은 다음과 같다.

```
개발환경 : MacBook Pro : 13-inch, M1, 2020 macOS : Big Sur (11.4.version)
사용언어 : 자바스크립트
사용 프레임워크 : React-Native
개발 툴(IDE) : VS Code
```

본 프로젝트의 목적은 누구나 쉽게 모바일 어플리케이션으로 접속하여 모임을 만들고, 각자가 속한 모임에서 회비 사용내역을 확인 할 수 있도록 하는 것이다. 따라서, 해당 목적을 정상적으로 수행하기 위해서는, OS 가리지 않고 모든 기기에서 해당 어플리케이션이 정상적으로 동작하여야 한다. 약 3달이라는 제한적인 시간과 팀 인원 수를 고려하여, IOS, Android를 각각 개발하는 것 보

 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

다 크로스 플랫폼 앱 개발을 선택하였다.

현재 시중에는 Flutter와 React-native가 크로스 플랫폼 앱 프레임워크로써 가장 많이 사용되고 있다. Flutter의 경우, Dart언어 기반으로 동작하는 프레임워크 이므로, 상대적으로 진입장벽이 높다고 판단했다. 또한, Flutter는 React-native 보다 상대적으로 플러그인과 관련 레퍼런스의 공급이 충분치 않다고 생각하였고, React를 사용한 클라이언트 개발 경험이 있어, React-Native를 선택했다

React-Native를 이용해서 앱을 개발하기 위해서는 다음과 같은 두 가지 방법을 선택할 수 있다.


- Expo Cli 사용
- React Native Cli 사용

첫번째 방법은 휴대폰에 Expo 어플리케이션을 설치해서 개발 PC와 동일한 AP를 사용하는 네트워크망에 연결해서 휴대폰에 자동으로 시뮬레이션하는 방법이다. Expo Cli의 장점은 배포하기 용이하고, 상대적으로 빠르고 간단하게 설치가 가능하다는 점이다. 그러나, 빌드 파일의 크기가 크고, Expo Cli의 일부 라이브러리의 경우 사용하는데 제한사항이 있을 수 있다는 단점이 존재한다.

두번째 방법인 React Native Cli는 직접 안드로이드 스튜디오 에뮬레이터 또는 애플의 Xcode IDE를 이용하는 방법이다. 장점은 개발자가 원하는 프로그래밍 언어로 추가 작성이 가능하며, 빌드파일의 크기가 상대적으로 작다는 점이다.

본 프로젝트에서 사용하는 개발 하드웨어는 MacBook Pro 13-inch M1이다. Expo는 현재 M1에 맞춰 업데이트를 하고, Cli 최적화 작업을 완료하였으나, 사용하지 않은 네이티브 모듈로 인하여, 빌드파일의 크기가 커지고, Expo에서 제공하지 않은 네이티브 모듈을 추가할 때, 불편함이 있어 두번째 방법인 React Native Cli를 사용하여 개발하였다. 또한, 본 프로젝트에서는 React Native Cli를 통하여 앱을 개발하기 위해서 Xcode, Android Studio 외에도, 추가적으로 Nodejs, Watchman, Cocoapods를 사용하였다.

- Nodejs
 - React-native는 javascript 언어를 기반으로 하기 때문에, Nodejs가 필요하다.
 - `brew install node`

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 위의 명령어를 통하여 Node.js를 설치하였고, Node.js를 설치하면, 기본적으로 Node.js 패키지 매니저인 npm도 같이 설치된다. React-Native 에서는 이 npm을 통하여, 라이브러리들을 추가하기 때문에 필수 불가결하다.

● Watchman

- Watchman은 특정 폴더나 파일을 감시하다가 변화가 생기면, 특정 동작을 실행하도록 설정하는 역할을 한다. React-Native에서는 소스코드의 추가, 변경이 발생하면 다시 빌드하기 위해 Watchman을 사용하고 있다.

- `brew install watchman`

- 위의 명령어를 통하여 Watchman을 설치할 수 있다.

● Cocoapods

- Cocoapods는 IOS 개발에 사용되는 의존성 관리자이다. React-Native로 IOS앱을 개발할 때, 라이브러리를 추가할 때마다 pod install을 해주어야 하므로 해당 라이브러리가 필요하다.

- `sudo gem install cocoapods`

- 위의 명령어를 통하여 cocoapods를 설치할 수 있다.

아래는 현재 개발 완료한 앱 어플리케이션의 폴더 구조 다이어그램이다.



```


├── App.js
├── Gemfile
├── Gemfile.lock
├── Screen
│   ├── AddFeeScreen.js
│   ├── AddMemberScreen.js
│   ├── CameraScreen.js
│   ├── ClubScreen.js
│   ├── CreateClubScreen.js
│   ├── JoinClubScreen.js
│   ├── LoginScreen.js
│   ├── MainScreen.js
│   ├── ManualReceiptDetailScreen.js
│   ├── ManualReceiptScreen.js
│   ├── ReceiptImageScreen.js
│   ├── ReceiptInfoScreen.js
│   ├── RegisterScreen.js
│   └── WithdrawScreen.js
├── __test__
├── android
├── app.json
├── babel.config.js
├── config.json
├── index.js
├── ios
├── metro.config.js
├── node_modules
├── package-lock.json
├── package.json
├── src
│   ├── CustomButton.js
│   ├── Router.json
│   ├── Styles.js
│   ├── config.js
│   └── icon
│       ├── BC.png
│       ├── DB.png
│       ├── Login.png
│       ├── Register.png
│       ├── add.png
│       ├── app_icon.png
│       ├── back-btn.png
│       ├── back.png
│       ├── full-star.png
│       ├── home.png
│       ├── logo.png
│       ├── logo_mono.jpg
│       ├── logout.png
│       ├── minus.png
│       ├── plus.png
│       ├── search.png
│       ├── splash.png
│       ├── star.png
│       ├── test.jpeg
│       └── textlogo.png
├── tree.txt
└── yarn.lock
    
```

npx react-native init <프로젝트 이름> 명령어를 통해서 React-Native 프로젝트를 생성할 수 있다. 해당 과정에서, Screen, src 디렉토리를 제외하고는 위 트리 구조 다이어그램과 같이 파일들이 생성된다. 각각의 파일들은 React-Native를 실행하기 위한 중요 파일로써 작동된다.

리액트 네이티브 앱을 실행하기 위해서는 다음 2가지 단계를 따라야 한다.

1. Metro 실행

React-native 앱을 실행하기 위해서는 먼저 metro를 실행해야 한다 Metro는 React-Native와 함께 설치되는 javascript 번들러로써 진입 파일 및 여러 옵션들을 취합하여 모든 코드 및 종속성들을 하나의 자바스크립트 파일로 만들어 반환한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

```
npx react-native start
```

Metro 는 react-native 프로젝트 폴더 내에서 위의 명령어를 통하여 실행 할 수 있다.

2. 어플리케이션 실행


```
npx react-native run-android
```

```
npx react-native run-ios
```

Metro 번들러가 실행되는 터미널은 그대로 두고 별도의 터미널에서 위의 명령어들을 실행한다. 모든 것이 정상적으로 설치 및 설정되었다면 안드로이드 및 Xcode 가상 기기를 통해서 실행된 앱을 확인 할 수 있다

다음은 주요한 javascript 파일을 소개하며 개발 내용을 소개하도록 하겠다.

- App.js
 - App.js 파일은 앱이 머신에서 실행될 때 최초로 실행되는 파일이다. 해당 파일은 직접 화면에 렌더링하는 동작을 하는 것이 아닌 '@react-navigation/native' 과 '@react-navigation/stack' 라이브러리를 통해서 전체 스크린들에 대하여 화면 동작을 명시해주는 파일이다.
 - 위 라이브러리들을 이용하여, 헤더의 유무, 최초 유저에게 보여지는 화면 설정 등과 같은 화면 동작의 세부사항을 명시해 주었다.
 - 또한, 'react-native-splash-screen'라이브러리를 통해서, 아래 그림과 같은 화면을 앱이 실행될 때 스플래쉬 페이지로써 약 4초간 띄워주고, 본격적인 앱 화면으로 넘어가게 해주었다.


 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

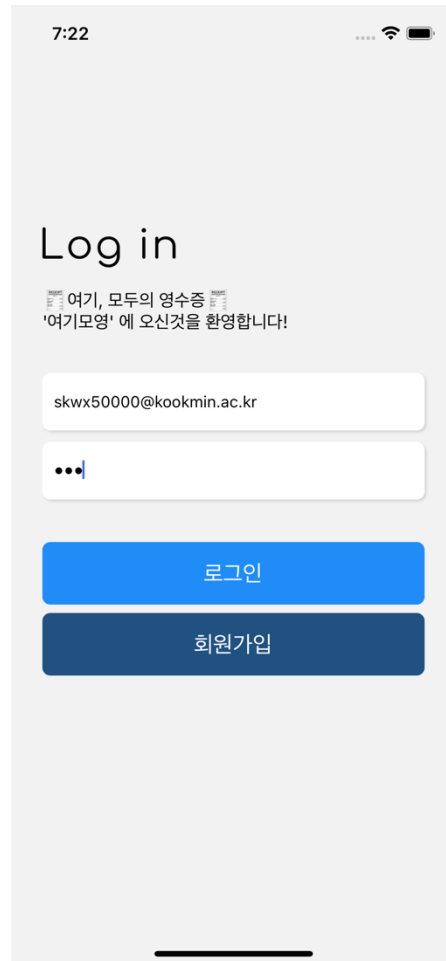
10:52




● LoginScreen.js

- 앱을 최초로 실행하면 사용자에게 보여지는 화면이다. 로그인 화면에는 2개의 TextInput과 2개의 button이 존재한다. 각 TextInput에는 아이디와 비밀번호를 기재하고, button중 상단 버튼은 메인화면으로 이동하는 로그인 버튼, 하단 버튼은 회원가입 페이지로 이동하는 회원가입 버튼이다.
- 만약 아이디, 비밀번호가 정상적으로 입력이 되었다면, 메인페이지로 이동하고, 비정상적으로 입력되면, "아이디 비밀번호를 제대로 입력해주세요" 라는 알림창을 띄워주게 된다.
- 로그인에 성공하면, 서버로 부터 해당 유저에 대한 정보를 받고 그것을 AsyncStorage에 저장하고, 메인페이지로 이동하게 된다.
- 해당 화면은 아래와 같다.

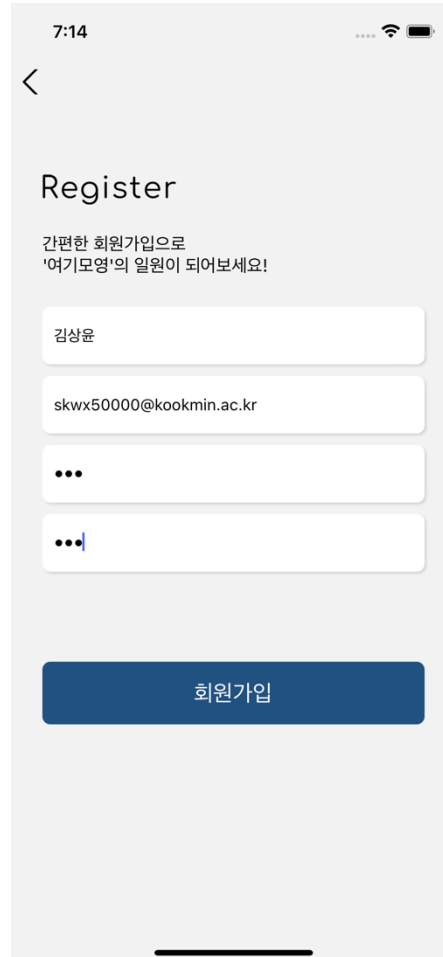
 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24



- RegisterScreen.js
 - 회원가입 페이지이다. 해당 페이지에는 4개의 TextInput과 1개의 button이 존재한다. 각 TextInput에는 이름, 아이디, 비밀번호, 비밀번호 확인을 기재한다. 작성이 완료되었다면, button을 눌러 회원가입을 할 수 있다.
 - 만약, 기재한 아이디가 이미 존재하는 아이디라면, 회원가입이 완료되지 않고 "회원가입에 실패하였습니다."라는 알림 창을 띄워준다. 또한, 기재 사항이 제대로 적혀있지 않다면, "정보를 모두 입력해주세요"라는 알림 창을 띄워주고, 비밀번호와 비밀번호 확인에 기재되어있는 내용이 다르면 "비밀번호를 다시 입력해주세요"라는 알림 창이 띄워지게 된다.


 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 해당 화면은 아래와 같다.



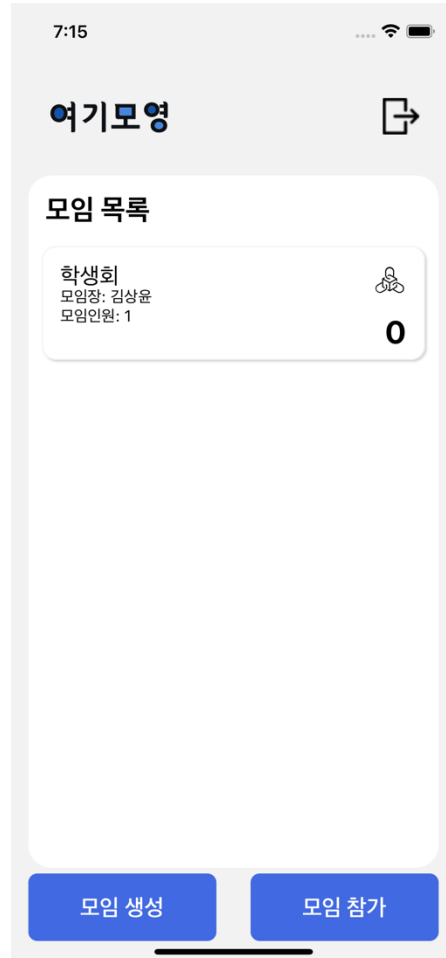
- MainScreen.js

- 메인페이지에는 상단에 로고, 로그아웃 버튼, Flatlist, 하단에 button 2개가 위치한다. Flatlist에는 서버로 부터 받은 데이터를 쿼리하여, 카드 형식으로 자신이 속한 모임을 보여준다. 또한, Flatlist에 포함된 각 카드에는 각 모임의 이름, 모임장, 모임인원, 모임 잔액, 저장방식이 표기된다. 하단의 버튼 2개는 각각 '모임 생성', '모임 참가' 버튼이다. '모임 생성' 버튼을 누르면 모임 생성 페이지로 이동하게 되고, '모임 참가' 버튼을 누르면 모임 참가 페이지로 이동하게 된다.
- Flatlist에는 refresh기능을 적용하였다. isRefreshing, setIsRefreshing 이라는 Boolean useState 변수를 선언해주고, Flatlist를 아래로 당길 때 마다 해당 변수의 값이 바뀌면

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

서 refresh기능이 작동하게 된다. Refresh기능이 작동되면, AsyncStorage에 저장된 유저 정보를 다시 서버로 보내서 최신 데이터를 받아서 화면에 렌더링해주게 된다.

- 해당 화면은 아래와 같다.

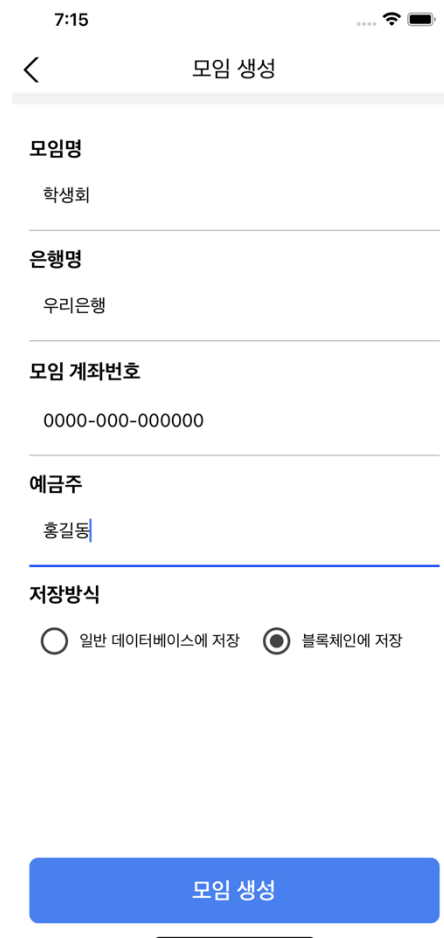


- CreateClubScreen.js

- 모임을 만들 수 있는 페이지이다. 총 4개의 TextInput과 1개의 버튼과 라디오버튼 그룹이 존재한다. 각 TextInput에 모임명, 은행명, 모임 계좌번호, 예금주를 입력하고, 라디오버튼 그룹에서 저장 방식을 선택하게 된다. 일반데이터베이스 저장을 선택하게 되면 mongoDB에 저장하게 되고, 블록체인에 저장을 선택하게 되면 블록체인에 저장하게 된다.


 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 해당 폼을 작성완료하고 모임 생성 버튼을 누르게 되면, 해당 데이터를 서버에 보내게 되고, 서버는 해당 데이터를 토대로 모임을 생성한다. 이때, 같이 현재 폼을 작성한 유저의 user_id를 보내게 되는데, 이를 이용하여 만들어진 모임장을 해당 유저로 설정하게 된다.
- 해당 화면은 아래와 같다.



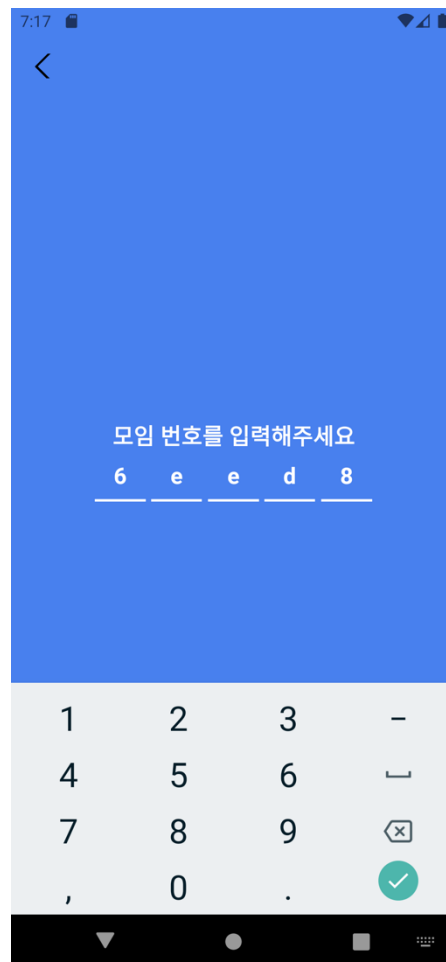
● JoinClubScreen.js

- 모임에 참가하는 페이지이다. 해당 페이지에는 총 5개의 자리를 적을 수 있는 칸이 마련되어있다. 해당 칸에 모임 코드에 해당하는 5자리를 입력하게 되면, 자동적으로 서버에 해당 코드를 보내고 코드에 해당하는 모임이 있다면 유저를 모임에 참가시킨

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24


다.

- 5개의 자리에는 숫자 뿐만 아니라 알파벳도 입력할 수 있다. 해당 칸은 React-native 의 'react-native-smooth-pincode-input' 라이브러리를 이용하여 구성하였다.
- 해당 화면은 아래와 같다.



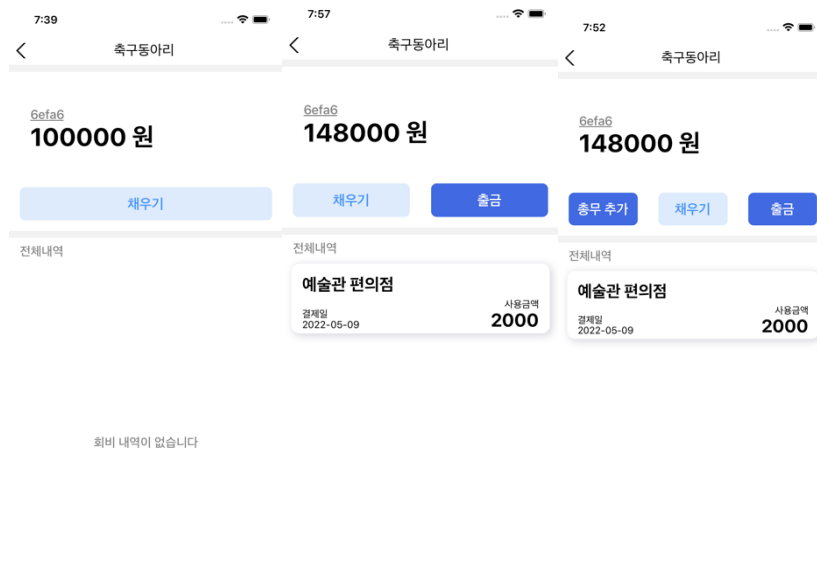
- ClubScreen.js

- 해당 페이지는 각 모임의 전용 페이지 이다. 메인페이지의 Flatlist에 기재된 각 카드를 터치하면, 해당 모임의 전용 페이지로 이동하게 된다. 상단 헤더에는 모임페이지의 이름이 나타나게 되고, 그 아래에는 모임 코드와 현재 모임 잔액을 명시해준다. 가장 하

 <div> 국민대학교 소프트웨어학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	모두모임	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24


단에는 해당 모임의 출금 내역을 표시해주는 Flatlist가 자리잡게 된다. 해당 Flatlist에는 회비의 출금 내역을 카드 형식으로 요약해서 보여주게 되고, 각 카드를 터치할 시 해당 출금 내역의 상세정보 페이지로 이동시킨다.

- 모임페이지 중단에는 버튼이 놓여있게 된다. 일반 참여자일 경우 '채우기' 버튼 1개, 총무의 권한이 있을경우 '채우기', '회비출금' 버튼이 놓여진다. 가장 높은 권한인 모임장의 경우는 '총무 추가', '채우기', '회비 출금'과 같은 3개의 버튼이 놓여져 있다.
- 해당 화면은 아래와 같다.



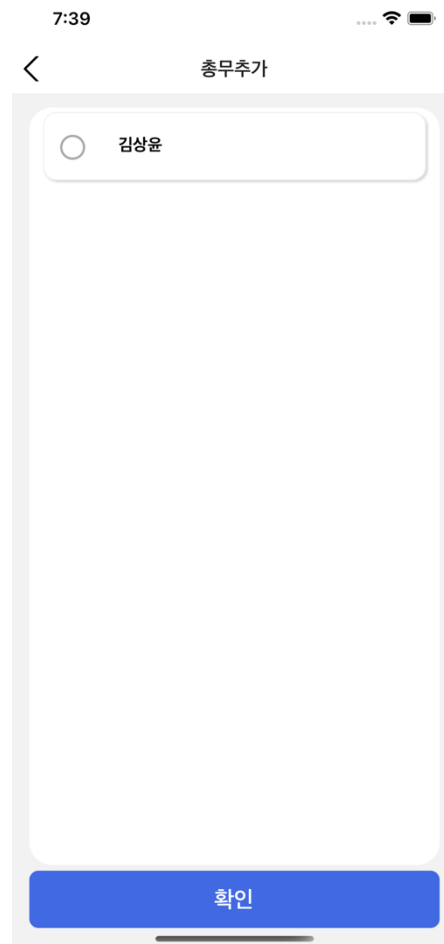
● AddMemberScreen.js

- 총무 권한을 부여할 수 있는 페이지이다. 해당 페이지에 들어가면, Flatlist와 하단에 button 1개가 위치한다. Flatlist에는 현재 모임장인 자신과 현재 총무를 제외한 나머지 모임 참가자가 카드형식으로 떠오르게 된다. 각 카드에는 참가자의 이름과 선택 유무를 표시하는 체크박스가 존재하는데, 카드를 터치할 때 마다 체크박스에 선택 유무가 표시된다.
- 총무 권한을 부여할 유저들을 선택하고, 확인을 누르게 되면, 선택한 유저들이 리스트

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24


형식으로 서버에 전달되고, 해당 유저들은 총무 권한을 가지게 된다.

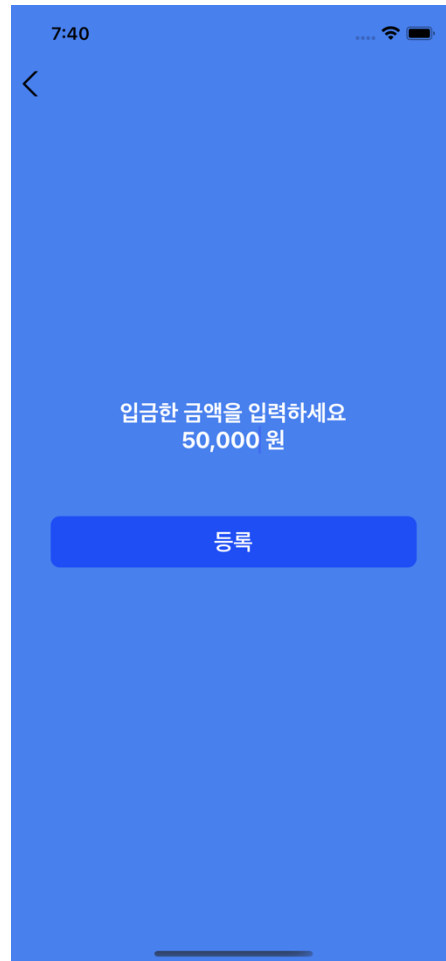
- 해당 화면은 아래와 같다.




- AddFeeScreen.js

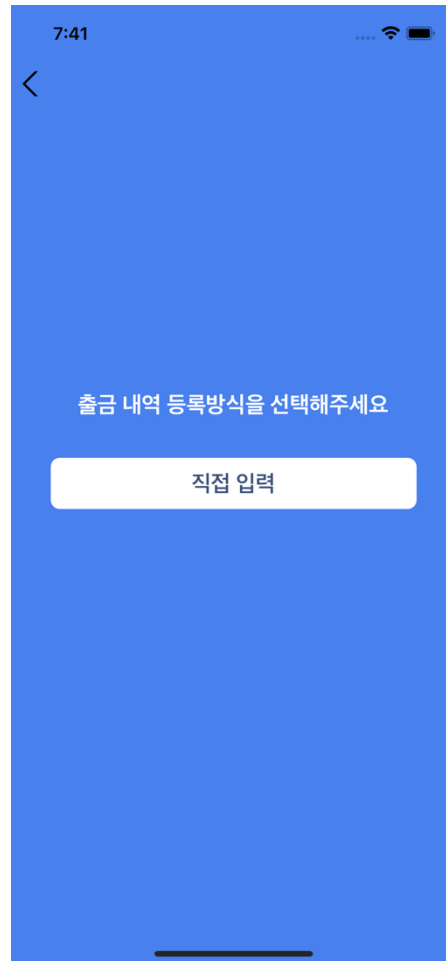
- 회비를 추가할 수 있는 페이지이다. 해당 페이지에는 1개의 TextInput과 버튼이 위치하고 있으며, 원하는 금액을 입력 후 버튼을 누르게 되면, 서버에 해당 모임의 id와 금액을 전달하여, 모임의 잔액에 입력한 금액만큼 더해지게 된다.
- 해당 화면은 아래와 같다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24




- WithDrawScreen.js
 - 해당 페이지는 모임페이지에서 '회비 출금' 버튼을 누르면 이동하게 되는 페이지이다. 해당 페이지에는 버튼 1개가 위치하여 있고, 해당 버튼을 누를 시 회비 내역을 작성하게되는 페이지로 이동한다.
 - OCR 인식등과 같은 추후 확장성을 고려하여 제작한 페이지이다.
 - 해당 화면은 아래와 같다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24



- ManualReceiptScreen.js
 - 회비 내역을 증명할 수 있는 영수증의 기본 정보를 입력하는 페이지이다. 해당 페이지에는 총 4개의 TextInput이 존재하며, 하단에는 1개의 버튼이 존재한다. TextInput 각각에 결제자 이름, 사용처, 결제 일자, 결제 금액을 명시해주는 페이지이다.
 - 각각의 TextInput에 내용을 기재한 뒤 '다음' 버튼을 누르게 되면 영수증의 상세정보를 입력해주는 페이지로 이동하게 된다.
 - 해당 화면은 아래와 같다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24



- ManualReceiptDetailScreen.js

- 영수증의 상세정보를 기입하는 페이지이다. 총 2개의 뷰가 위치해 있으며, 첫번째 뷰에는 2개의 TextInput이 존재한다. 각 TextInput에 상품의 이름과 가격을 적고 우측 상단의 +버튼을 누르게 되면 하단 카드에 존재하는 Flatlist에 카드형식으로 추가가 된다.
- 해당 카드에는 위에서 입력한 상품의 정보가 기재 되어있고 우측 상단에는 -버튼이 존재한다. 해당 '-' 버튼을 누르게 되면 Flatlist에서 삭제된다.
- 정보를 모두 기입하고, '사진 등록하기' 버튼을 누르면 사진을 등록할 수 있는 페이지로 이동하게 된다.


 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 해당 화면은 아래와 같다.



- CameraScreen.js

- 사진을 등록할 수 있는 페이지이다. 해당 페이지에는 큰 사각형의 터치가 가능한 공간과 1개의 버튼이 존재한다. 초기 해당 페이지에는 빈 사각형에 +표시가 있는 공간이 보이게 된다. 해당 사각형을 터치하게 되면 자동적으로 camera 기능이 켜지게 되고, camera를 이용하여, 영수증을 찍게 된다.
- Camera관련 기능을 구현하기 위하여 react-native의 react-native-image-crop-picker 라이브러리를 이용하였다. 해당 라이브러리를 통하여, 사진을 찍고 또한 Crop할 수 있는 기능을 구현하였다.


 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 정상적으로 사진촬영이 완료되었으면 기존 빈 사각형에 촬영하였던 사진이 렌더링 되고, 사진이 정상적으로 렌더링 되었다면, '등록' 버튼을 눌러서 최종 출금 내역 제출을 완료하고 메인페이지로 이동할 수 있다.
- 해당 화면은 아래와 같다.

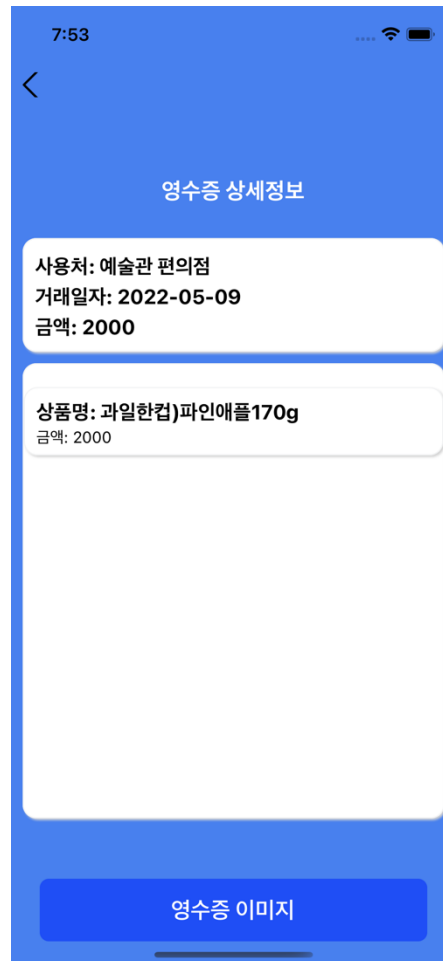


● ReceiptInfoScreen.js

- 각 모임 페이지에 기재된 카드 형식의 출금 내역을 터치하면 이동하게 되는 페이지이다. 총 2개의 View와 1개의 버튼으로 이루어져 있고, 상단에는 해당 출금 내역의 사용처, 거래 일자, 금액을 명시해준다. 하단 View에는 Flatlist로 구성되어있고, 각 출금 내역의 상세정보가 기재 되어있다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

- 하단에는 '영수증 이미지'라는 버튼이 위치하여 있는데, 해당 버튼을 터치하게 되면, 해당 출금 내역에 대한 영수증 이미지를 볼 수 있는 페이지로 이동하게 된다.
- 해당 화면은 아래와 같다.



- ReceiptImageScreen.js
 - 영수증의 사진을 보여주는 페이지이다. 해당 페이지에는 영수증 이미지를 보여주는 공간이 중간에 위치하게 된다.
 - 해당 페이지로 이동하게 되면, 서버와 DB에서 해당 페이지에 렌더링 해줄 base64 코드를 찾아서 클라이언트로 보내주게 되고, 클라이언트에서는 base64 코드를 디코딩하

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24

여 이미지를 사용자에게 제공하게 된다.


- 해당 화면은 아래와 같다.

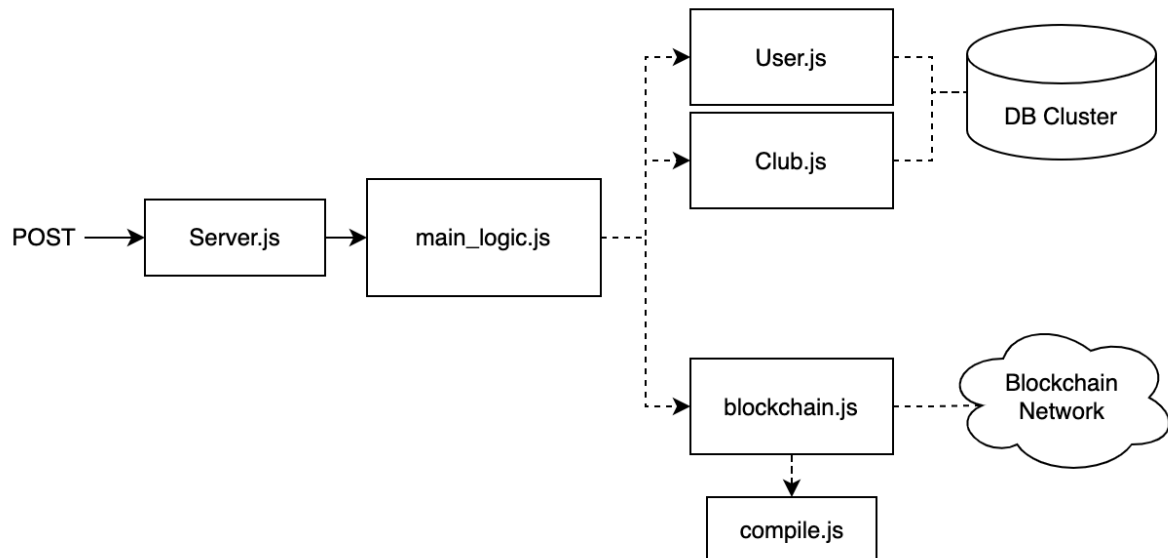


3.2.2 웹 어플리케이션 서버(Node.js)

Node.js는 web3 모듈을 이용하여 블록체인 네트워크(Geth)와 통신하기 위한 표준 플랫폼이다. 또한 본 프로젝트에서는 소규모의 서비스를 제공하기 때문에 Node.js 가 적합하다고 판단했다.

웹 어플리케이션 서버는 총 6개의 컴포넌트로 구성되어 있다.

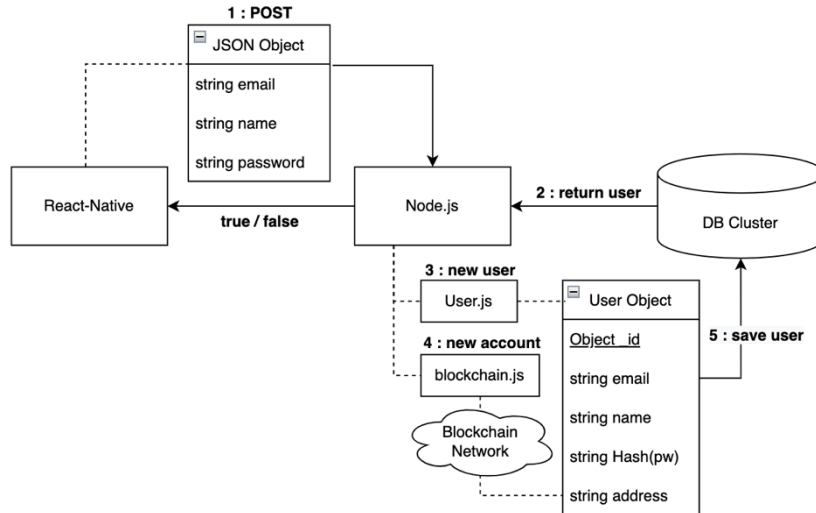
 <div> <p>국민대학교</p> <p>소프트웨어학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.5	2022-MAY-24



- server.js : 앱 서버에서 요청한 기능을 비즈니스 로직(main_logic)으로 연결하는 역할을 수행한다.
- main_logic.js : 정의된 기능에 맞는 데이터 처리 역할을 담당한다. 모임 정보 저장 방식에 따라 두가지 방식으로 분기된다. 다음 페이지에서 자세히 설명한다.
- User.js : DB에 저장할 User의 데이터 구조가 정의되어 있다. 3.2.4(데이터베이스)에서 자세하게 설명한다.
- Club.js : DB에 저장할 Club의 데이터 구조가 정의되어 있다. 3.2.4(데이터베이스)에서 자세하게 설명한다.
- blockchain.js : 블록체인 네트워크와 RPC(Remote Procedure Call) 통신을 수행한다. 3.2.5(블록체인)에서 자세하게 설명한다.
- compile.js : Club 스마트 컨트랙트를 컴파일하고 bytecode, ABI(Application Binary Interface)를 반환하는 역할을 수행한다.

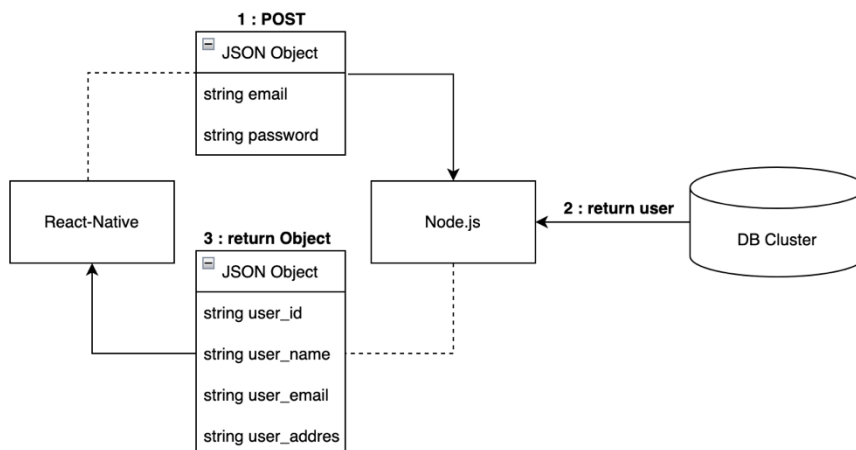
main_logic 컴포넌트에서는 사용자가 요청한 기능에 따라 총 7개의 역할을 수행한다.

● 회원가입 (register)



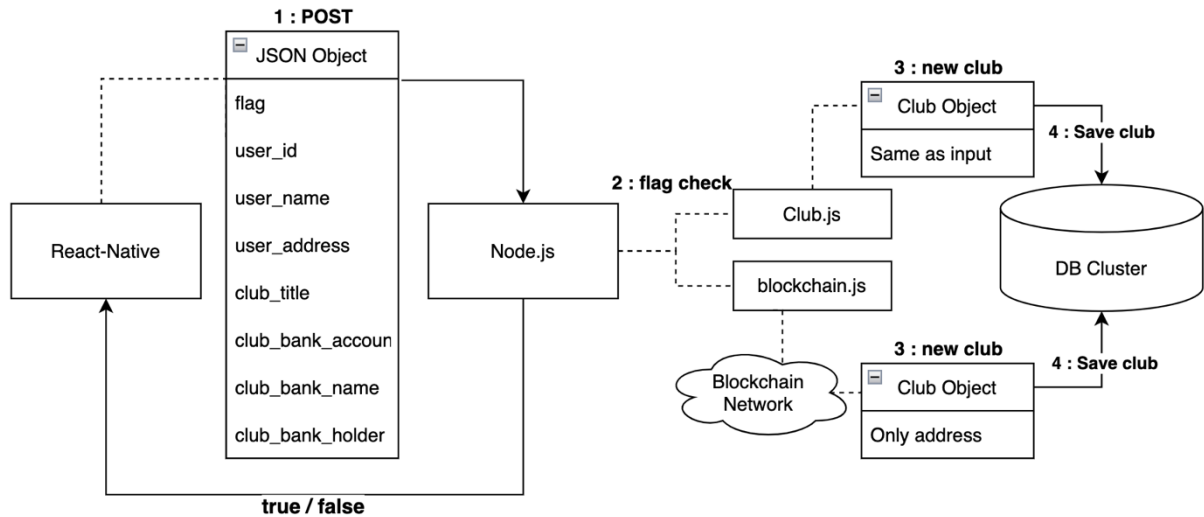
사용자가 입력한 데이터를 받는다. DB를 조회하여 이메일 중복 검사를 진행한다. 새로운 User를 만들고 블록체인 네트워크에서 계정을 생성한다. DB에 User를 저장한다.

● 로그인 (login)




사용자가 입력한 데이터를 받는다. DB를 조회하여 비밀번호 일치 여부를 진행한다. User의 데이터 일부분을 가공하여 응답한다.

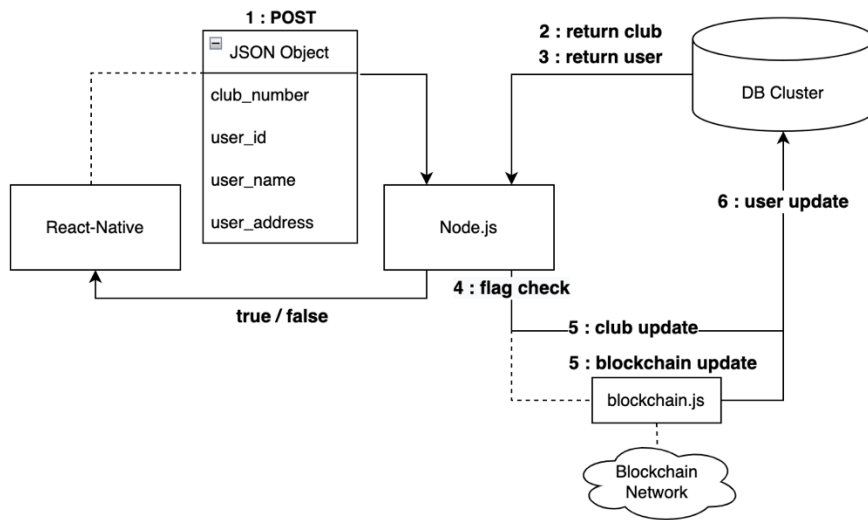
● 모임 생성 (createClub)



사용자가 입력한 데이터를 받는다. 모임의 저장 방식(flag)에 따라 분기한다. 일반 DB 방식의 경우, 새로운 Club을 만들고 사용자가 입력한 데이터를 할당한다. 블록체인 방식의 경우, 입력한 데이터로 스마트 컨트랙트를 발급하고 사용자가 입력한 데이터를 저장한다. 그리고 컨트랙트의 주소를 새로운 Club에 저장한다. DB에 Club을 저장한다.

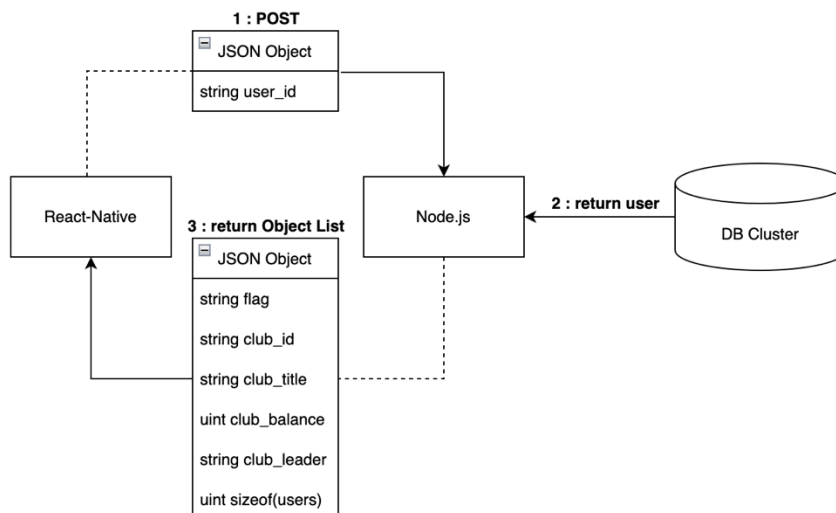
● 모임 참가 (joinClub)

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16



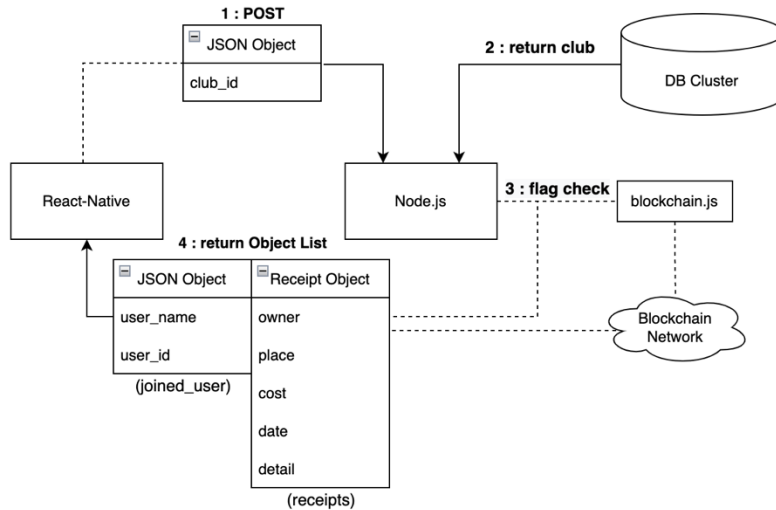
사용자가 입력한 데이터를 받는다. DB에서 일치하는 club과 user를 조회한다. Club의 저장 방식에 따라 분기한다. 일반 DB 방식의 경우, club의 joined_user에 user_id를 추가한다. 블록체인 방식의 경우, 스마트 컨트랙트 변수 joined_user에 user_id를 추가한다. 마지막으로 user의 joined_club에 club_id를 추가한다.

- 사용자 모임 조회 (userClubInfo)



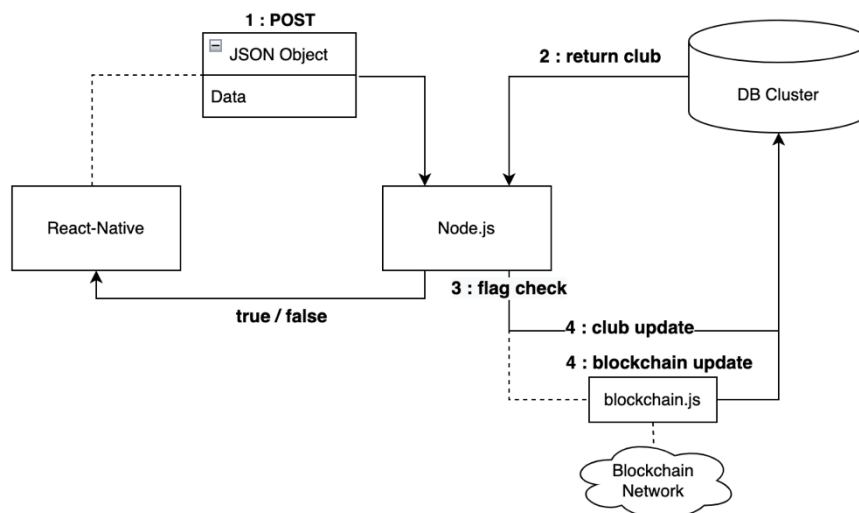
사용자가 입력한 데이터를 받는다. DB에서 user를 조회한다. User의 joined_club 리스트 데이터의 일부분을 가공하여 응답한다.

● 모임 페이지 이동 (gotoClub)



사용자가 입력한 데이터를 받는다. DB에서 club을 조회한다. Club의 저장 방식에 따라 분기한다. 일반 DB 방식의 경우, club의 joined_user와 club의 receipt 데이터 일부분을 가공하여 응답한다. 블록체인 방식의 경우, 스마트 컨트랙트의 변수 joined_user, receipts 일부분을 가공하여 응답한다.

● 총무 추가 (addClubMember), 회비 추가 (addClubFee), 영수증 추가 (addClubReceipt)



 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16

사용자가 입력한 데이터를 받는다. DB에서 club을 조회한다. Club의 저장 방식에 따라 분기한다. 일반 DB 방식의 경우, club의 데이터를 변경한다. 블록체인 방식의 경우, 스마트 컨트랙트의 변수를 변경한다. 세 기능 모두 동일한 흐름이나, 아래 그림과 같이 입력되는 데이터에 차이가 있다.

총무 추가	회비 추가	영수증 추가	
<div>JSON Object</div> <div>club_id</div> <div>user_id</div> <div>deparment</div>	<div>JSON Object</div> <div>club_id</div> <div>fee</div>	<div>JSON Object</div> <div>club_id</div> <div>user_name</div>	<div>Receipt Object</div> <div>owner</div> <div>date</div> <div>cost</div> <div>place</div> <div>detail</div>


3.2.3 AWS EC2

AWS EC2는 서버를 개설하는 가장 일반적인 방법이다. EC2 인스턴스에서 Node.js와 Geth 클라이언트를 구동한다. Geth에서 블록을 채굴할 수 있는 최소 사양 조건과 지원받을 수 있는 금액 조건에 맞추어 t2.medium을 선택했다. React-Native와의 POST 통신 연결 주소를 고정하기 위해 Elastic IP를 배정받았고 Port를 3001로 고정했다.

3.2.4 데이터베이스

데이터 베이스는 mongoDB를 선택하였다. MongoDB는 NoSQL 데이터베이스로써 수평 확장이 가능하고, Schema-less 하며 완화된 ACID가 큰 특징 중 하나이다.

MongoDB를 선택한 이유는 본 프로젝트에서는 지연 시간과, SNS 기반 서비스 이기 때문이다. 누구나 해당 어플리케이션을 통하여 모임에 참여 및 생성을 할 수 있고, 영수증 데이터를 앱에 올려야 하기 때문에, DB에서의 쿼리에 있어서 지연시간이 오래되면 오래될 수록 해당 서비스의 만

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16

속도와 상품성이 현저하게 떨어질 것이라 판단하였다.

또한, 본 프로젝트에서 웹 어플리케이션 서버는 Node.js로 구성을 하였다. Node.js에서는 mongoose 라이브러리를 통하여 MongoDB와의 연결을 더욱 용이하게 해준다. 해당 라이브러리 또한 데이터베이스로써 MongoDB를 선택한 가장 큰 이유였다.

본 프로젝트에서는, MongoDB Atlas를 이용하여, 데이터베이스를 로컬호스트가 아닌 클라우드 저장소에 개설했고, 크게 Info라는 데이터베이스를 개설했고, 그 아래에 2개의 collection을 개설했다.

하나는 유저의 정보를 담기 위한 users라는 collection이고 또 하나는 모임의 정보를 담기 위한 clubs라는 collection이 존재한다.

본 프로젝트에서는 모임의 저장 방식을 DB 또는 블록체인을 선택하게 해주었다. 따라서, 어떤 옵션을 선택하느냐에 따라 club collection에 서로 다른 field를 가진 데이터들이 저장된다.

본 프로젝트에서의 MongoDB 데이터베이스의 데이터 저장 구조는 아래와 같다.

- user


유저 스키마 이다. name필드를 통하여 유저의 이름을 명시하고, email필드를 통하여, 가입한 유저의 이메일을 명시해준다. 비밀번호 같은 경우에는 password필드를 통하여 명시하였는데, 이것은 Node.js 서버 단의 bcrypt 라이브러리를 통하여, 암호화 하여 저장하게 된다. joined_club 필드에는 해당 유저가 참여한 모임의 object_id를 리스트의 형태로 저장하게 된다. 마지막으로 address 필드에는 각 유저의 CA가 저장된다.

```

_id: ObjectId("628b799b2c77db31f765f390")
name: "김상윤"
email: "skwx5000@kookmin.ac.kr"
password: "$2b$10$1STLIb4lcxbnUT2Q2IDDcuajluxl5qLn8E6h8tfya0D8DDo2rJ.e2"
joined_club: Array
  0: "628b748768ebb6898816f42a"
address: "0x3096461dC6fd65B79eE06ffd877f4FADbCA96FB6"
__v: 0

```

- club

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모임	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16

모임의 경우, 저장되는 방식이 2가지 이다. 각 방식마다 저장되는 필드는 상이하다.

저장 방식이 DB인 경우, 아래와 같이 필드가 작성된다. Joined_user에는 현재 해당 모임에 참여중인 유저의 object_id가 리스트형태로 담기게 된다. Club_leader_name 과 club_leader_id에는 모임 개설자의 정보가 기재된다. Club_number는 모임에 참여하기 위하여 필요한 5자리 코드이다. Deployed_time은 모임이 개설된 시점을 나타내는 필드이다. 해당 필드를 이용하여, 클라이언트에 모임에 대한 내용이 정렬되어 사용자에게 표시된다.

```

_id: ObjectId("628c6dd47865fa3b3eb1b05c")
flag: "DB"
club_title: "학생회"
club_bank_account: "0000-000-000000"
club_bank_name: "우리은행"
club_bank_holder: "김상윤"
> joined_user: Array
> joined_member: Array
> receipt: Array
club_balance: 0
club_leader_name: "test"
club_leader_id: "628c6dab7865fa3b3eb1b056"
club_number: "1b05c"
deployed_time: 1653370324
__v: 0


```

저장 방식이 BC인 경우, 아래와 같이 필드가 작성된다. 특별히 별도로 추가된 필드는 없으나, DB저장방식과의 차이점은, 몇가지 필드는 데이터 베이스가 아닌 스마트 컨트랙트에 저장되는 차이점이 있다.

```

_id: ObjectId("628b748768ebb6898816f42a")
> joined_user: Array
> joined_member: Array
> receipt: Array
address: "0x9E9F753DD31078c958DdE2dBdD03ccD0785831Ba"
flag: "BC"
club_number: "6f42a"
__v: 0

```

	국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서	
		프로젝트 명	모두모영
		팀 명	꼭꼭
		Confidential Restricted	Version 1.1 2022-MAY-16

3.2.5 블록체인

블록체인은 P2P 기반의 데이터 저장 방식이다. 데이터는 블록에 기록되고 해당 블록의 해쉬코드가 다음 블록에 저장된다. 누군가 임의적으로 블록의 데이터를 변경하면 이어지는 블록들의 해쉬코드 값이 변경된다. 블록체인 네트워크의 참가자들은 모두 동일한 데이터를 가지고 있기 때문에, 채굴과정에서 이를 승인하지 않는다. 결과적으로, 블록체인에 저장된 데이터를 조작할 수 없다. 본 프로젝트에서는 총무가 회비를 데이터를 고의적으로 조작하는 것을 방지하기 위해 블록체인을 도입했다.

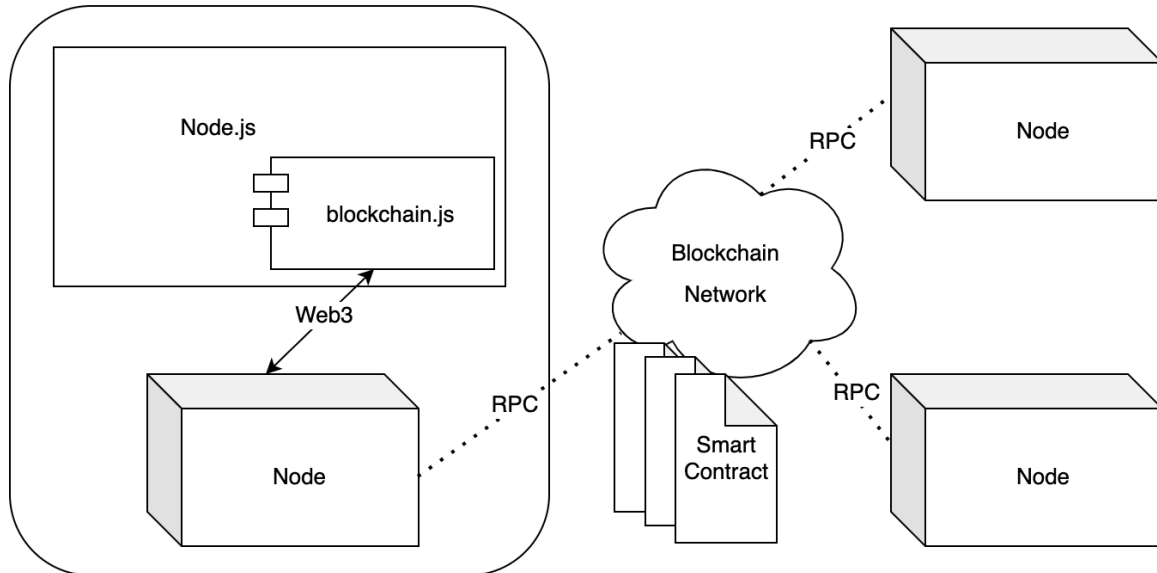
블록체인 네트워크는 이더리움 방식 중 하나인 Go-ethereum(이하 'Geth') 클라이언트를 사용했다. Geth는 다른 도구(HyperLedger, EOS...)보다 많은 정보가 있어 비교적 학습이 빠르다는 이유로 채택했다.

현재 블록체인은 3개의 노드(작성자의 개인 머신 2개, AWS EC2 1개)로 구성되어 있다. 서비스의 특징에 맞게 블록체인의 환경설정을 변경했다.

- Base64 데이터를 저장하기 위해 개별 트랜잭션 처리량을 128KB에서 약 10MB로 증가시켰다.
- Base64 데이터를 저장하는 함수를 실행하는 경우 트랜잭션 처리 비용인 gas를 약 300배 증가시켰다.
- 트랜잭션 처리 비용이 증가함에 따라, 최대 gas 사용량인 gasLimit을 300배 증가시켰다.
- 채굴에 사용되는 컴퓨팅 비용을 줄이기 위해 트랜잭션이 생성될 때 3개의 블록을 채굴하도록 스크립트를 작성했다.

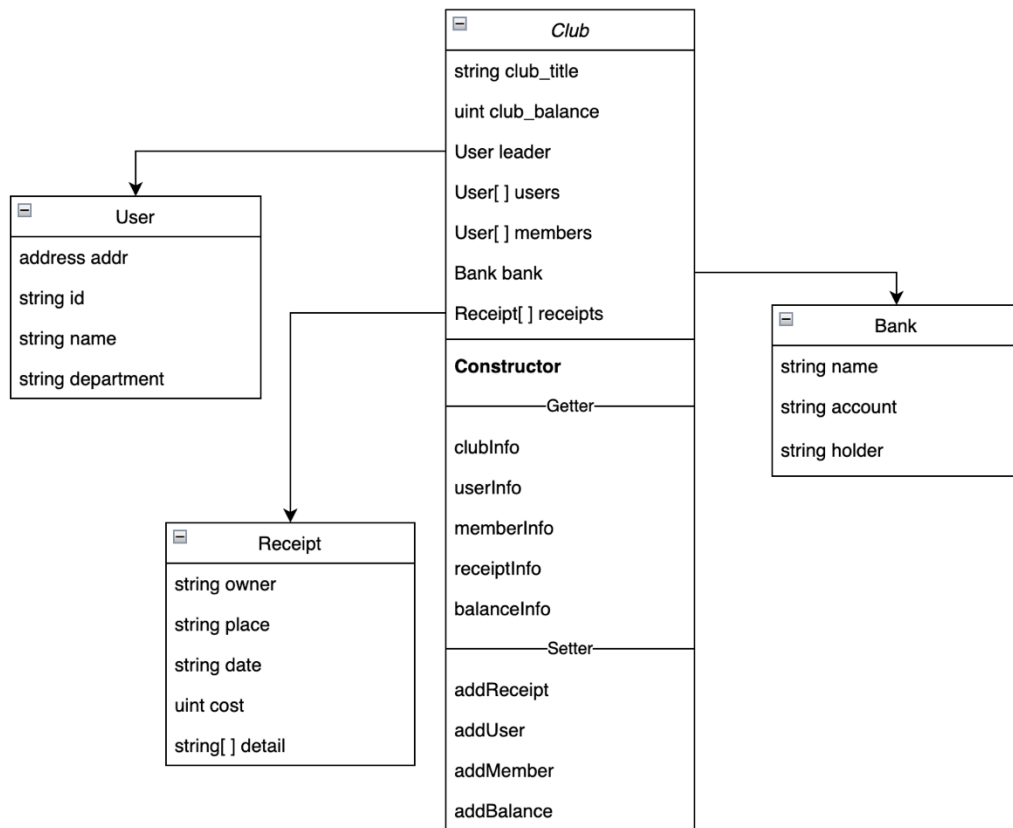
이더리움 기반의 블록체인은 스마트 컨트랙트를 미리 작성하고, 서버에서 함수를 실행 요청할 수 있다. 함수를 실행하는 주체(서버)는 채굴자에게 지급되는 비용인 gas를 지불한다. 이때, 사용자는 채굴을 하지 않기 때문에 gas가 없다. 이를 해결하기 위해 사용자가 계정을 생성하는 경우, 10 Ether(10^{19} , 약 300회 트랜잭션 처리 가능)를 보내는 방법을 사용했다.

AWS EC2



사용자가 모임을 블록체인으로 생성한 경우, AWS EC2에서 참여한 노드를 통해 Club 스마트 컨트랙트가 블록체인 네트워크에 배포된다. 된다. 해당 컨트랙트에서 데이터 변경이 일어날때마다 트랜잭션이 발생되고, 이는 모두 블록체인에 기록된다. 스마트 컨트랙트의 데이터 구조는 일반 DB에 저장되는 데이터 구조와 가능한 동일하게 구성했다.

아래는 스마트 컨트랙트의 데이터 구조이다.




스마트 컨트랙트에는 modifier를 설정하여 Getter / Setter 함수를 실행하는 주체를 Leader, Member, User로 제한했다. 일반 사용자(User)는 내역 조회만 가능하고 모임장(Leader), 총무(Member)는 변수 변경이 가능하도록 구성했다.

4 자기평가

4.1 김상윤

MongoDB 구축, Node.js 서버 개발, Node.js 서버와 MongoDB 통신, React-native 개발 및 서버와의 연결을 담당했다. 3월달에 캡스톤프로젝트가 개시된 이후, 소프트웨어융합대학 자율주행스터디 강의실에 매일 출석하며, 개발에 정진하였고, 매주 월, 수, 일요일마다 팀원들과 회의를 하며 이슈 사항 및 공유사항을 전달하였다. 또한, 이번 프로젝트에서는 Github을 적극적으로 사용하여 개발에 임하였다. 그러나 해당 프로젝트에 아쉬움도 많이 남는다. 기존 앱 기획 단에서 계획한 스케줄

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16


보다, React-native 개발 파트가 지연되는 덕분에 기존에 계획하였던, 영수증 이미지 인식 기능을 탑재하지 못한 것이 아쉬웠다. 조금 더 구체적이고 현실적인 계획을 세웠다면, 개발 범위를 좀더 줄여 핵심적인 기능을 위주로 결과물에 담아 조금 더 완성도가 높은 서비스를 제작할 수 있었을 것이라 기대한다. 해당 서비스는 캡스톤디자인 강의로 한정 짓기 보다는 추후에 계속 개발하여 좀 더 완성도가 높은 서비스로 발전시켜 본격적인 배포 및 출시를 하고자 한다.

4.2 민태식

UI/UX 구성을 맡았다. 본래는 앱 개발까지 맡았으나, 제작 중 어려움을 느꼈고, 김상윤 학생의 도움을 받아 UI/UX 구성을 맡는 것으로 역할을 조정했다. 이전에 모바일 프로그래밍 과목을 수강하면서 앱 제작을 해본 경험이 있기 때문에 문제없이 프로젝트를 진행할 수 있다고 생각했으나, 처음 배워 사용해 보는 언어와, 프레임워크를 비롯한 환경에 예상보다 적응에 어려움을 겪었다. 팀원인 안성열학생과 김상윤학생이 많은 도움을 주었지만, 팀 프로젝트 수행이 익숙하지 않아 소통과정에서 실수도 잦았고, 오해를 많이 만들었다. 프로젝트를 진행하면서 많은 부분에서 부족한 모습을 보였고, 누가 되었다고 생각한다. 개인적으로, 팀 적으로 미안한 마음이 크다. 프로젝트 진행 전 기본적인 준비과정이 부족했다는 것을 많이 느낀다. 이번 프로젝트를 통해 많은 부분을 배웠고, 앞으로 맡게 될 다음 프로젝트부터는 더 나은 모습을 보일 수 있도록 노력하겠다.

4.3 안성열

Geth 클라이언트를 이용한 블록체인 네트워크 구성, 스마트 컨트랙트 코드 개발, Node.js 서버 개발, Node.js 서버와 블록체인 네트워크 통신, AWS EC2 개설 및 작업 환경 구축을 담당했다. 3팀의 팀장으로서 프로젝트 추진 및 진행상황을 관리했고, 팀원 간 적극적인 소통을 위해 노력했다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16

5 부록

5.1 사용자 메뉴얼

5.1.1 블록체인 채굴 참여

본 프로젝트에서 개설한 네트워크에 참여하는 방법을 설명한다.

- Geth 클라이언트 다운로드

<https://geth.ethereum.org/downloads/> 에 접속하여 본인의 환경에 맞는 버전을 설치한다.

- Genesis.json 파일 다운로드

[깃허브 저장소 blockchain folder](#) 에서 파일을 다운로드 받는다.

- 블록체인 네트워크 실행

\$ geth init --datadir <file_name> genesis.json : 네트워크 개설을 위한 초기화 작업이다.

\$ geth --datadir <file_name> --networkid 2022 console : 네트워크를 개설한다.

- 블록체인 네트워크 노드 참가


```
> admin.addPeer("enode://c0a7e4fa2e1e9f23fdcf45e96eb432e251c31725e8756b05578fb9e3d96338e34241f46c32bc335a7aba61d59f57122fc31baef9eef5bd6f973890dc8538ab3b@118.33.68.135:30303?discport=0")
```

```
> admin.peers
```

- 채굴 시작

> personal.newAccount("<password>") : 계정을 생성한다. 처음 실행하는 경우 자동으로 채굴비용을 지급하는 계정으로 설정된다.

> miner.start(<thread_count>) : 채굴을 시작한다. Thread_count 만큼 Thread를 사용한다.

	국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서	
		프로젝트 명	모두모영
		팀 명	꼭꼭
		Confidential Restricted	Version 1.1 2022-MAY-16

Thread_count가 높을 수록 채굴이 빨라진다.

> miner.stop() : 채굴을 중지한다.

> eth.getBalance(eth.coinbase) : 현재까지의 채굴 보상을 확인한다.

5.2 운영자 메뉴얼

5.2.1 블록체인 네트워크 구성

본 프로젝트에서 개설한 네트워크에 참여하지 않고, 다른 사설 네트워크를 개설하는 방법을 설명한다.

- Geth 클라이언트 다운로드

<https://geth.ethereum.org/downloads/> 에 접속하여 본인의 환경에 맞는 버전을 설치한다.

- genesis.json 파일 다운로드

[깃허브 저장소 blockchain folder](#) 에서 파일을 다운로드 받는다. 이때, chainId, difficulty, gasLimit의 값을 변경했다.


chainId : 사설 블록체인 네트워크의 고유 id 값. 참가하기 위해 동일해야 한다.

difficulty : 블록을 채굴하는 난이도이다. 값이 높을수록 블록을 채굴하는 평균 시간이 증가한다.

gasLimit : 트랜잭션을 일으키는 최대 비용이다. 큰 데이터를 사용하면 비용이 증가하고, 이에 따라 gasLimit을 증가시켜야 한다.

- Automine.json 파일 다운로드

[깃허브 저장소 blockchain folder](#) 에서 파일을 다운로드 받는다. 트랜잭션이 발생하면 3개의 블록을 채굴하도록 조정되었다.

	국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서	
		프로젝트 명	모두모영
		팀 명	꼭꼭
		Confidential Restricted	Version 1.1 2022-MAY-16

● 사설 네트워크 개설

\$ geth init --datadir <file_name> genesis.json : 네트워크 개설을 위한 초기화 작업이다.

\$ geth --datadir <file_name> --networkid <chainId> --nodiscover --nat extip:<ip_address>
 --http --http.addr "<ip_address>" --http.api "web3,eth,personal" --allow-insecure-unlock --
 miner.gasprice 0 --preload "<automine.js pwd>" console : 네트워크가 개설된다.

- --nodiscover : 수동으로 노드를 추가한다.
- --nat extip : 데이터 통신 시 지정 IP로 고정한다.
- --http : 블록체인 네트워크 연결 시 http 방식을 사용한다.
- --http.addr : http 방식에서 사용되는 endpoint를 고정한다.
- --http.api : RPC 통신을 통해 외부에서 실행할 수 있는 모듈을 정의한다.
- --allow-insecure-unlock : RPC 통신을 통해 외부에서 계정 해제를 가능하게 한다.
- --miner.gasprice : 채굴자가 트랜잭션을 처리하는 가스 비용을 설정한다.
- --preload : geth 클라이언트에서 동작하는 스크립트를 설정한다.


● 채굴 시작

> personal.newAccount("<password>") : 계정을 생성한다. 처음 실행하는 경우 자동으로 채굴비용을 지급하는 계정으로 설정된다.

> miner.start(<thread_count>) : 채굴을 시작한다. Thread_count 만큼 Thread를 사용한다. Thread_count가 높을 수록 채굴이 빨라진다. (자동채굴 스크립트를 추가한 경우, 해당 명령어를 사용하지 않아도 된다.)

> miner.stop() : 채굴을 중지한다.

> eth.getBalance(eth.coinbase) : 현재까지의 채굴 보상을 확인한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	모두모영	
	팀 명	꼭꼭	
	Confidential Restricted	Version 1.1	2022-MAY-16

5.2.2 웹 서버 개설

- 5.2.1 에 작성된 메뉴얼을 따라 블록체인 네트워크를 개설하고 채굴을 진행한다.
- 블록체인 네트워크 개설 후 endpoint를 저장한다.

```
INFO [05-24|06:13:02.043] HTTP server started          endpoint=172.
31.8.46:8545 auth=false prefix= cors= vhosts=localhost
```

- [GitHub 저장소](#)의 server 폴더를 다운로드 받는다.
- Server.js 소스코드 상단의 server_port 변수를 조정한다.
- Blockchain.js 소스코드 상단의 blockchain_endpoint 변수를 조정한다.
- node server.js 명령어를 입력해 서버를 실행한다.