

2022 전기 졸업과제 중간 보고서

블록체인을 활용한 포트폴리오 관리 플랫폼 개발



AOMD 팀

담당교수 : 권동현

201624586 조병우

201712159 조현우

201724490 서지원

목차

요구조건 및 제약 사항 분석.....	3
요구 조건.....	3
제약 사항에 대한 수정사항.....	4
보고 시점까지의 과제 수행 내용 및 중간 결과.....	5
백엔드(Back-End).....	5
블록체인(Block-Chain).....	6
프론트엔드(Front-End).....	12
사용자 시나리오.....	20
갱신된 과제 추진 계획.....	22
개발 일정 변경.....	22
구성원별 진척도.....	23
현재까지 진행 상황.....	23
앞으로의 추진 계획.....	23

1. 요구조건 및 제약 사항 분석

1.1. 요구 조건

본 졸업 과제는 프라이빗 블록체인 기반의 개인 포트폴리오 관리 플랫폼 개발을 목표로 한다. 제안하는 시스템은 개인정보 위, 변조 문제를 해결하기 위해 자기 정보 통제 및 데이터 무결성 측면에서 장점을 가지는 블록체인을 활용하고 블록체인 네트워크는 프라이빗 네트워크의 구현체인 하이퍼레저 패브릭을 기반으로 동작한다.

1) 블록 체인 (Block Chain)

- a) 블록체인의 공유 원장에는 개인의 학력과 경력, 성적 등의 개인 포트폴리오에 기입할 수 있는 전반적인 내용을 기록한다. 이는 내부 합의 알고리즘을 통해 검증된 데이터만 블록으로 생성하는 형태로 동작한다.
- b) 블록체인에는 포트폴리오에 기록할 개인정보들을 저장한다. 체인코드를 사용해 블록체인 네트워크로부터 데이터를 가져올 수 있다. 가져온 데이터는 읽기 전용 데이터이다.

2) 백엔드 (Back-End)

- a) ID/PW 또는 OAuth2.0 기반의 사용자 인증을 통해 이용자를 식별한다.
- b) 식별된 사용자에게 따라 개인 포트폴리오 정보를 블록체인과 통신하여 불러온다.
- c) 자기소개서 등의 블록체인 네트워크에 저장되지 않는 정보는 DB 를 통해 저장/관리한다.
- d) 사용자가 자신이 만든 포트폴리오의 읽기전용 url 을 만들 수 있으며, 이를 입사희망 기업에 제출할 수 있다. 또한 언제든지 이 url 을 사용자가 파기할 수 있게 한다.
- e) 링크를 받은 기업은 링크를 통해 해당 웹 어플리케이션에 접속하여 읽기만 가능한 상태의 포트폴리오를 제공받는다.

3) 프론트엔드 (Front-End)

- a) 리액트 라이브러리를 통해 DOM 관리와 상태 변화 관리를 최소화하고 개발자는 오직 기능 개발과 사용자 인터페이스에 집중할 수 있게 한다. 또한 SPA 방식으로 부분 갱신을 통해 전체적인 트래픽 감소와 페이지 렌더링에서의 효율을 보장한다.

- b) 리덕스 상태 관리 라이브러리를 사용하여 단방향 데이터 흐름을 구현한다. 이를 통해 웹 사이트에서 현재 필요한 정보들을 체계적으로 관리하고 코드의 유지 보수성을 높인다.
- c) 트래픽의 감소와 속도, 반응성의 향상으로 인해 앱처럼 자연스러운 최상의 사용자 경험(UX)을 제공한다. 이는 추후 사용자 테스트를 진행하고 수치화된 자료를 토대로 인사이트를 내어 증명한다.

1.2. 제약 사항에 대한 수정사항

(1) 블록체인 네트워크를 단순하게 변경하였다.

- 블록체인 네트워크에 많은 시간을 투자하게 될수록 서버 개발이 늦어지게 된다. 이는 다른 팀원들의 진행 상황에 영향을 끼칠 수 있다. 따라서, 블록체인 네트워크를 단순하게 구현한 뒤 서버 개발 착수 이후 다시 빌드업 하기로 하였다.

(2) 체인 코드 언어를 생소한 Go 에서 익숙한 Java 로 변경하였다.

- 기존에 나와 있는 레퍼런스가 Go 언어로 되어 있기 때문에 Go 로 개발할 예정이었으나, 코드 이해가 어려운 점과 언어 통일을 고려하여 체인 코드의 언어를 Java 로 변경하였다.

(3) Fabric SDK 의 사용을 보류하였다.

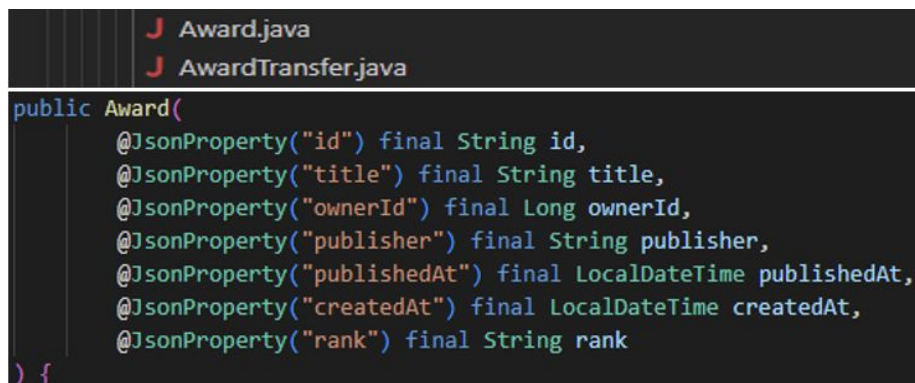
- Java 어플리케이션에서 동작하는 SDK 를 제공해주는 것은 확인하였으나 실제 동작 시 grpc 통신에서 문제가 발생하였다. 같은 명령어를 쉘 스크립트로 발생시킬 경우 정상 동작하기 때문에 SDK 의 사용은 추후 서버개발의 진척이 생기면 다시 진행해보기로 하였다.

2. 보고 시점까지의 과제 수행 내용 및 중간 결과

2.1. 백엔드(Back-End)

1) 원장 Asset 과 체인코드 정의

1-1) 수상내역(award), 학력(education), 자격증(license)의 3 가지 Asset 과 체인코드를 정의한다.



```

Award.java
AwardTransfer.java

public Award(
    @JsonProperty("id") final String id,
    @JsonProperty("title") final String title,
    @JsonProperty("ownerId") final Long ownerId,
    @JsonProperty("publisher") final String publisher,
    @JsonProperty("publishedAt") final LocalDateTime publishedAt,
    @JsonProperty("createdAt") final LocalDateTime createdAt,
    @JsonProperty("rank") final String rank
) {
```

그림 1. 원장에 기록될 Award Asset 정의

1-2) 수상 내역(Award)는 id, title, ownerId, publisher, publishedAt, createdAt, rank 의 member 를 갖고 있다.



```

Education.java
EducationTransfer.java

public Education(
    @JsonProperty("id") String id,
    @JsonProperty("title") String title,
    @JsonProperty("ownerId") Long ownerId,
    @JsonProperty("publisher") String publisher,
    @JsonProperty("publishedAt") LocalDateTime publishedAt,
    @JsonProperty("createdAt") LocalDateTime createdAt,
    @JsonProperty("state") String state,
    @JsonProperty("departmentInfo") String departmentInfo
) {
```

그림 2. 원장에 기록될 Education Asset 정의

1-3) 학력(Education)은 id, title, ownerId, publisher, publishedAt, createdAt, state, departmentInfo 의 member 를 갖고 있다. State 는 입학 예정, 재학 중, 휴학, 졸업 예정, 졸업의 상태를 갖게 된다.

```

License.java
LicenseTransfer.java

public License(
    @JsonProperty("id") String id,
    @JsonProperty("title") String title,
    @JsonProperty("ownerId") Long ownerId,
    @JsonProperty("publisher") String publisher,
    @JsonProperty("publishedAt") LocalDateTime publishedAt,
    @JsonProperty("createdAt") LocalDateTime createdAt,
    @JsonProperty("description") String description,
    @JsonProperty("expireDate") LocalDateTime expireDate,
    @JsonProperty("qualificationNumber") String qualificationNumber
) {

```

그림 3. 원장에 기록될 License Asset 정의

1-4) 자격증(License)는 id, title, ownerId, publisher, publishedAt, createdAt, description, expireDate, qualificationNumber 의 member 를 갖고 있다.

2.2. 블록체인(Block-Chain)

1) 블록체인 네트워크 생성

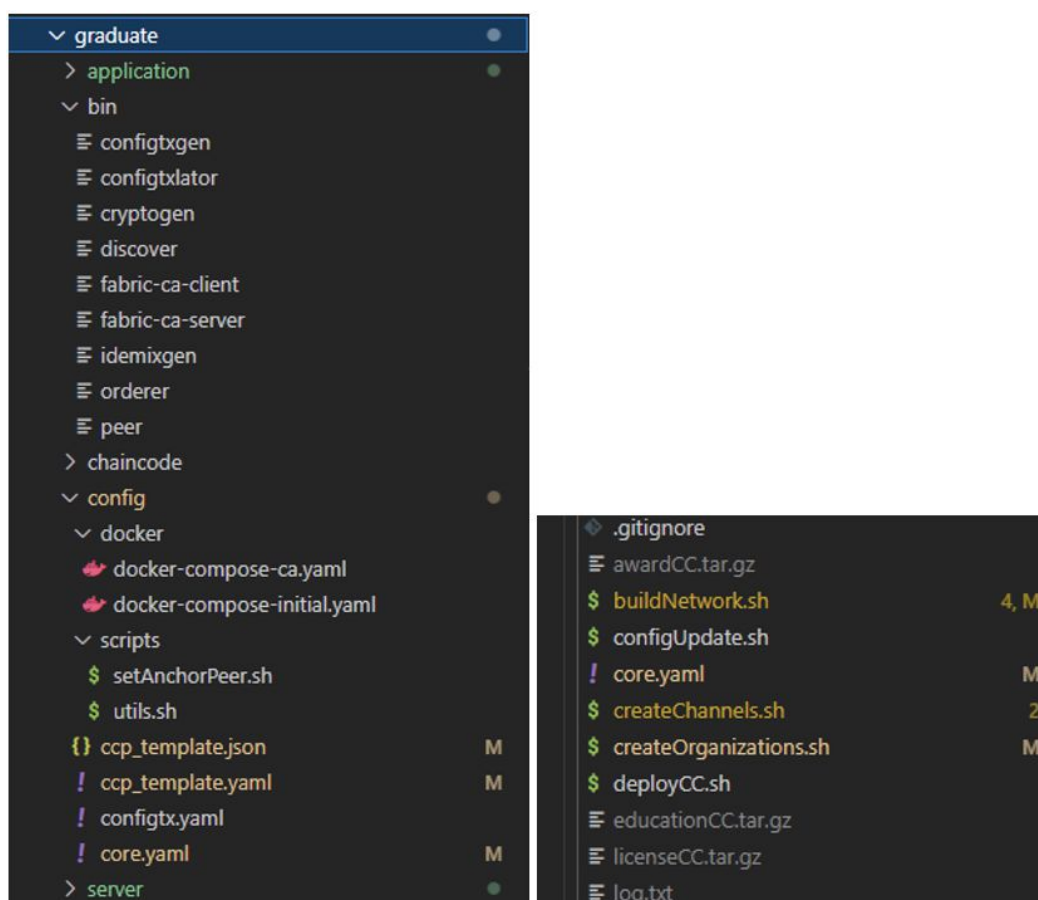
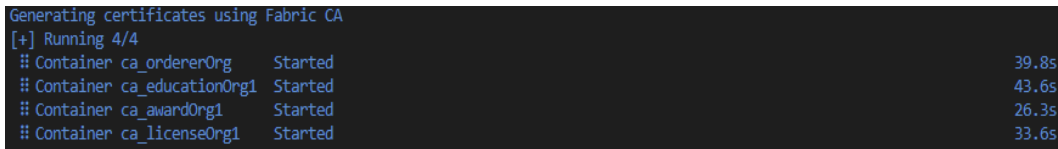


그림 4. 블록체인 네트워크 프로젝트 내부 파일

- application 폴더: SDK 사용을 테스트하기 위한 프로젝트가 들어있는 폴더
- bin 폴더: 블록체인 구성을 위한 기본 바이너리 파일들이 들어있는 폴더
- chaincode 폴더: chaincode 와 asset 을 정의하는 java 프로젝트 파일들이 들어있는 폴더
- config 폴더: docker-compose 구성 파일들과 디버깅을 위한 utils 스크립트, 네트워크 구성원(노드)들의 초기 구성을 위한 template 파일, 네트워크 구성을 위한 config 과 core 파일이 들어있는 폴더
- server 폴더: 서버(SpringBoot)프로젝트가 들어있는 폴더, 추후 폴더 구성을 변경할 예정

buildNetwork.sh up 명령어를 사용하여 네트워크 구성을 시작한다.

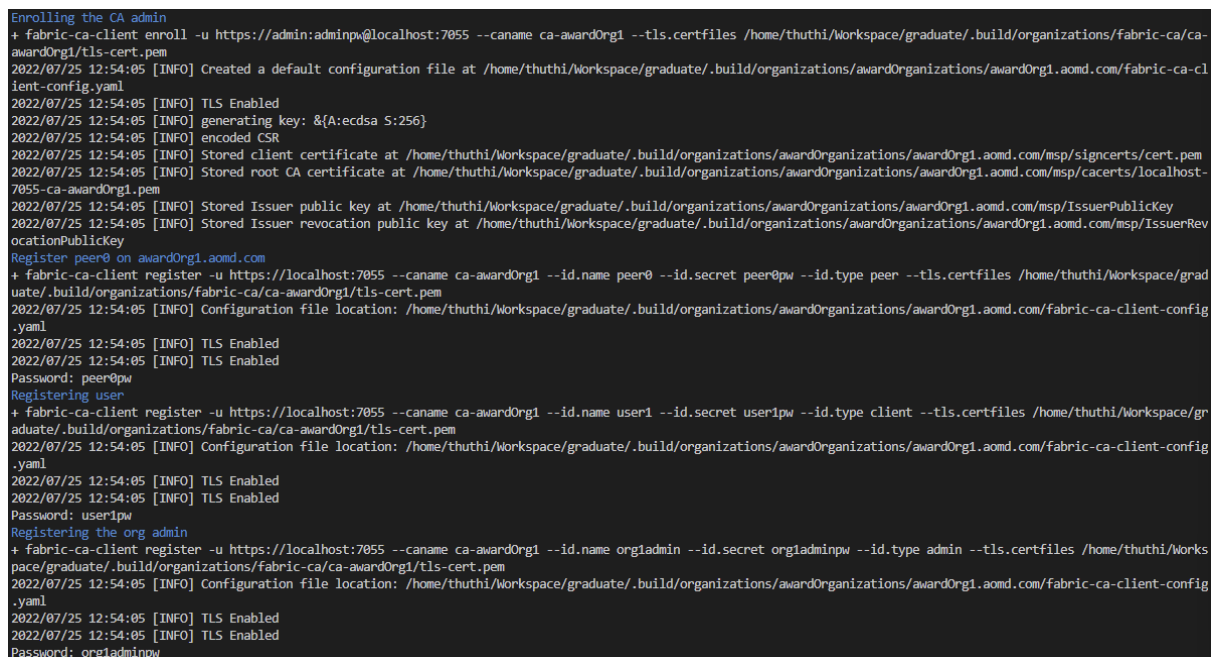
1-1) 각 Organization 의 CA 선언(order, award, education, license)



```
Generating certificates using Fabric CA
[+] Running 4/4
# Container ca_ordererOrg Started 39.8s
# Container ca_educationOrg1 Started 43.6s
# Container ca_awardOrg1 Started 26.3s
# Container ca_licenseOrg1 Started 33.6s
```

그림 5. 앞으로 등록할 Organization 의 CA 를 docker 에 선언 및 등록한 log

1-2) 각 Organisation 들의 CA admin, 기본 peer, user, org admin 등록



```
Enrolling the CA admin
+ fabric-ca-client enroll -u https://admin:adminpw@localhost:7055 --caname ca-awardOrg1 --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] Created a default configuration file at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/fabric-ca-client-config.yaml
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] generating key: &{A:ecdsa S:256}
2022/07/25 12:54:05 [INFO] encoded CSR
2022/07/25 12:54:05 [INFO] Stored client certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/msp/signcerts/cert.pem
2022/07/25 12:54:05 [INFO] Stored root CA certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/msp/cacerts/localhost-7055-ca-awardOrg1.pem
2022/07/25 12:54:05 [INFO] Stored Issuer public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/msp/IssuerPublicKey
2022/07/25 12:54:05 [INFO] Stored Issuer revocation public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/msp/IssuerRevocationPublicKey
Register peer0 on awardOrg1.aomd.com
+ fabric-ca-client register -u https://localhost:7055 --caname ca-awardOrg1 --id.name peer0 --id.secret peer0pw --id.type peer --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] Configuration file location: /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/fabric-ca-client-config.yaml
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] TLS Enabled
Password: peer0pw
Registering user
+ fabric-ca-client register -u https://localhost:7055 --caname ca-awardOrg1 --id.name user1 --id.secret user1pw --id.type client --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] Configuration file location: /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/fabric-ca-client-config.yaml
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] TLS Enabled
Password: user1pw
Registering the org admin
+ fabric-ca-client register -u https://localhost:7055 --caname ca-awardOrg1 --id.name orgladmin --id.secret orgladminpw --id.type admin --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] Configuration file location: /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/fabric-ca-client-config.yaml
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] TLS Enabled
Password: orgladminpw
```


그림 6. Organization 의 기본 구성원들을 등록한 log

1-3) 등록된 구성원들의 인증서 발급

```

Generating the peer0 msp
+ fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7055 --caname ca-awardOrg1 -M /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/msp --csr.hosts peer0.awardOrg1.aomd.com --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] generating key: &{A:ecdsa S:256}
2022/07/25 12:54:05 [INFO] encoded CSR
2022/07/25 12:54:05 [INFO] Stored client certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/msp/signcerts/cert.pem
2022/07/25 12:54:05 [INFO] Stored root CA certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/msp/cacerts/localhost-7055-ca-awardOrg1.pem
2022/07/25 12:54:05 [INFO] Stored Issuer public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/msp/IssuerPublicKey
2022/07/25 12:54:05 [INFO] Stored Issuer revocation public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/msp/IssuerRevocationPublicKey
Generating the peer0-tls certificates
+ fabric-ca-client enroll -u https://peer0:peer0pw@localhost:7055 --caname ca-awardOrg1 -M /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/tls --enrollment.profile tls --csr.hosts peer0.awardOrg1.aomd.com --csr.localhost --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] generating key: &{A:ecdsa S:256}
2022/07/25 12:54:05 [INFO] encoded CSR
2022/07/25 12:54:05 [INFO] Stored client certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/tls/signcerts/cert.pem
2022/07/25 12:54:05 [INFO] Stored TLS root CA certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/tls/tlscacerts/tls-localhost-7055-ca-awardOrg1.pem
2022/07/25 12:54:05 [INFO] Stored Issuer public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/tls/IssuerPublicKey
2022/07/25 12:54:05 [INFO] Stored Issuer revocation public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/peers/peer0.awardOrg1.aomd.com/tls/IssuerRevocationPublicKey
Generating the user msp
+ fabric-ca-client enroll -u https://user1:user1pw@localhost:7055 --caname ca-awardOrg1 -M /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/User1@awardOrg1.aomd.com/msp --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] generating key: &{A:ecdsa S:256}
2022/07/25 12:54:05 [INFO] encoded CSR
2022/07/25 12:54:05 [INFO] Stored client certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/User1@awardOrg1.aomd.com/msp/signcerts/cert.pem
2022/07/25 12:54:05 [INFO] Stored root CA certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/User1@awardOrg1.aomd.com/msp/cacerts/localhost-7055-ca-awardOrg1.pem
2022/07/25 12:54:05 [INFO] Stored Issuer public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/User1@awardOrg1.aomd.com/msp/IssuerPublicKey
2022/07/25 12:54:05 [INFO] Stored Issuer revocation public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/User1@awardOrg1.aomd.com/msp/IssuerRevocationPublicKey
Generating the org admin msp
+ fabric-ca-client enroll -u https://org1admin:org1adminpw@localhost:7055 --caname ca-awardOrg1 -M /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/Admin@awardOrg1.aomd.com/msp --tls.certfiles /home/thuthi/Workspace/graduate/.build/organizations/fabric-ca/ca-awardOrg1/tls-cert.pem
2022/07/25 12:54:05 [INFO] TLS Enabled
2022/07/25 12:54:05 [INFO] generating key: &{A:ecdsa S:256}
2022/07/25 12:54:05 [INFO] encoded CSR
2022/07/25 12:54:06 [INFO] Stored client certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/Admin@awardOrg1.aomd.com/msp/signcerts/cert.pem
2022/07/25 12:54:06 [INFO] Stored root CA certificate at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/Admin@awardOrg1.aomd.com/msp/cacerts/localhost-7055-ca-awardOrg1.pem
2022/07/25 12:54:06 [INFO] Stored Issuer public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/Admin@awardOrg1.aomd.com/msp/IssuerPublicKey
2022/07/25 12:54:06 [INFO] Stored Issuer revocation public key at /home/thuthi/Workspace/graduate/.build/organizations/awardOrganizations/awardOrg1.aomd.com/users/Admin@awardOrg1.aomd.com/msp/IssuerRevocationPublicKey

```

그림 7. 등록된 조직의 구성원들의 tls 인증서 발급 log

1-4) Organization 등록

```

[+] Running 9/9
  # Volume "docker_orderer.aomd.com"          Created           0.0s
  # Volume "docker_peer0.educationOrg1.aomd.com" Created           0.0s
  # Volume "docker_peer0.awardOrg1.aomd.com"    Created           0.0s
  # Volume "docker_peer0.licenseOrg1.aomd.com"   Created           0.0s
  # Container peer0.licenseOrg1.aomd.com         Started          30.9s
  # Container orderer.aomd.com                   Started          48.4s
  # Container peer0.educationOrg1.aomd.com        Started          44.8s
  # Container cli                                 Started          10.4s
  # Container peer0.awardOrg1.aomd.com           Started          38.6s

```

그림 8. 조직을 docker 에 등록한 log

1-5) Channel 생성을 위한 각 Channel 당 초기 artifacts 정의

```

Generating channel create transaction 'educationchannel.tx'
+ configtxgen -profile EducationChannel -outputCreateChannelTx .build/channel-artifacts/educationchannel.tx -channelID educationchannel
2022-07-25 12:54:56.004 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-07-25 12:54:56.039 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/thuthi/Workspace/graduate/config/configtx.yaml
2022-07-25 12:54:56.039 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2022-07-25 12:54:56.040 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Generating channel create transaction 'awardchannel.tx'
+ configtxgen -profile AwardChannel -outputCreateChannelTx .build/channel-artifacts/awardchannel.tx -channelID awardchannel
2022-07-25 12:54:56.055 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-07-25 12:54:56.087 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/thuthi/Workspace/graduate/config/configtx.yaml
2022-07-25 12:54:56.087 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2022-07-25 12:54:56.088 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0
Generating channel create transaction 'licensechannel.tx'
+ configtxgen -profile LicenseChannel -outputCreateChannelTx .build/channel-artifacts/licensechannel.tx -channelID licensechannel
2022-07-25 12:54:56.103 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-07-25 12:54:56.135 KST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/thuthi/Workspace/graduate/config/configtx.yaml
2022-07-25 12:54:56.135 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2022-07-25 12:54:56.136 KST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
+ res=0

```

그림 9. 각 Channel 생성을 위한 초기 artifacts 를 정의한 log

1-6) 각 Channel 생성

```

educationchannel을 만드는 중입니다.
+ peer channel create -o localhost:9050 -c educationchannel --ordererTLSTLSHostnameOverride orderer.AOMD.com -f .build/channel-artifacts/educationchannel.tx --outputBlock .build/channel-artifacts/educationchannel.block --tls --cafile .build/organizations/ordererOrganizations/orderer.aomd.com/orderers/orderer.aomd.com/msp/tlscacerts/tlsca.aomd.com-cert.pem
+ res=0
2022-07-25 13:18:44.676 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-07-25 13:18:44.683 KST [cli.common] readBlock -> INFO 002 Expect block, but got status: &{NOT_FOUND}
2022-07-25 13:18:44.685 KST [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2022-07-25 13:18:44.889 KST [cli.common] readBlock -> INFO 004 Expect block, but got status: &{SERVICE_UNAVAILABLE}
2022-07-25 13:18:44.897 KST [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2022-07-25 13:18:45.102 KST [cli.common] readBlock -> INFO 006 Expect block, but got status: &{SERVICE_UNAVAILABLE}
2022-07-25 13:18:45.110 KST [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2022-07-25 13:18:45.314 KST [cli.common] readBlock -> INFO 008 Expect block, but got status: &{SERVICE_UNAVAILABLE}
2022-07-25 13:18:45.321 KST [channelCmd] InitCmdFactory -> INFO 009 Endorser and orderer connections initialized
2022-07-25 13:18:45.524 KST [cli.common] readBlock -> INFO 00a Expect block, but got status: &{SERVICE_UNAVAILABLE}
2022-07-25 13:18:45.532 KST [channelCmd] InitCmdFactory -> INFO 00b Endorser and orderer connections initialized
2022-07-25 13:18:45.735 KST [cli.common] readBlock -> INFO 00c Received block: 0
/home/thuthi/Workspace/graduate/.build/channel-artifacts/educationchannel.block
+ peer channel join -b /home/thuthi/Workspace/graduate/.build/channel-artifacts/educationchannel.block
+ res=0
2022-07-25 13:18:48.769 KST [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-07-25 13:18:48.782 KST [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
채널 educationchannel을 생성하고, Organization들이 참여했습니다.
educationchannel의 configuration을 가져옵니다.
채널 educationchannel의 config block의 최신 내용을 불러옵니다.
+ peer channel fetch config config_block.pb -o orderer.aomd.com:9050 --ordererTLSTLSHostnameOverride orderer.aomd.com -c educationchannel --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/orderer.aomd.com/orderers/orderer.aomd.com/msp/tlscacerts/tlsca.aomd.com-cert.pem
2022-07-25 04:18:48.918 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-07-25 04:18:48.919 UTC [cli.common] readBlock -> INFO 002 Received block: 0
2022-07-25 04:18:48.919 UTC [channelCmd] fetch -> INFO 003 Retrieving last config block: 0
2022-07-25 04:18:48.920 UTC [cli.common] readBlock -> INFO 004 Received block: 0

```

그림 10. 각 Channel 을 생성한 log

1-7) 각 Channel 에 Organization 참가 및 초기 Configuration 업데이트

```

채널 educationchannel을 생성하고, Organization들이 참여했습니다.
educationchannel의 configuration을 가져옵니다.
채널 educationchannel의 config block의 최신 내용을 불러옵니다.
+ peer channel fetch config config_block.pb -o orderer.aomd.com:9050 --ordererTLSHostnameOverride orderer.aomd.com -c educationchannel --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/organizations/ordererOrganizations/orderer.aomd.com/orderers/orderer.aomd.com/msp/tlsacerts/tlsca.aomd.com-cert.pem
2022-07-25 04:18:48.918 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-07-25 04:18:48.919 UTC [cli.common] readBlock -> INFO 002 Received block: 0
2022-07-25 04:18:48.919 UTC [channelCmd] fetch -> INFO 003 Retrieving last config block: 0
2022-07-25 04:18:48.920 UTC [cli.common] readBlock -> INFO 004 Received block: 0
config_block을 json형태로 바꾸는 중입니다.
+ configtxlator proto_decode --input config_block.pb --type common.Block
+ jq '.data.data[0].payload.data.config'
educationchannel의 educationOrg1에 anchor peer update tx를 생성합니다.
+ jq '.channel_group.groups.Application.groups.EducationOrg1MSP.values += [{"AnchorPeers":{"mod_policy": "Admins","value":{"anchor_peers": [{"host": "peer0.educationOrg1.aomd.com","port": 6051}],"version": "0"}}}]' EducationOrg1MSPconfig.json
+ configtxlator proto_encode --input EducationOrg1MSPmodified_config.json --type common.Config
+ configtxlator proto_encode --input EducationOrg1MSPmodified_config.json --type common.Config
+ configtxlator compute_update --channel_id educationchannel --original original_config.pb --updated modified_config.pb
+ configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate
+ jq .
++ cat config_update.json
+ echo '{"payload":{"header":{"channel_header":{"channel_id":"educationchannel", "type":2}}, "data":{"config_update":{"channel_id":"educationchannel", "isolated_data": {}, "read_set": {"groups": {"Application": {"groups": {"EducationOrg1MSP": {"groups": {}, "mod_policy": "", "policies": {"Admins": {"mod_policy": "", "policy": null, "version": "0"}, "Readers": {"mod_policy": "", "policy": null, "version": "0"}, "Writers": {"mod_policy": "", "policy": null, "version": "0"}, "values": {"MSP": {"mod_policy": "", "value": null, "version": "0"}, "version": "0"}, "mod_policy": "", "policies": {}, "values": {}, "version": "1"}, "write_set": {"groups": {"Application": {"groups": {"EducationOrg1MSP": {"groups": {}, "mod_policy": "Admins", "policies": {"Admins": {"mod_policy": "", "policy": null, "version": "0"}, "Endorsement": {"mod_policy": "", "policy": null, "version": "0"}, "Readers": {"mod_policy": "", "policy": null, "version": "0"}, "Writers": {"mod_policy": "", "policy": null, "version": "0"}, "values": {"AnchorPeers": {"mod_policy": "Admins", "value": {"anchor_peers": [{"host": "peer0.educationOrg1.aomd.com", "port": 6051}]}], "version": "0"}, "MSP": {"mod_policy": "", "value": null, "version": "0"}, "version": "1"}, "mod_policy": "", "policies": {}, "values": {}, "version": "1"}, "mod_policy": "", "policies": {}, "values": {}, "version": "0"}}}}}}}}'
+ configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope

```

그림 11. 등록된 Channel 에 Organization 참가 및 초기 config 업데이트 log

1-8) 블록체인 네트워크 최종 구성 docker

```

thuthi@thuthi-server:~/Workspace/graduate$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
18f75284eba1	hyperledger/fabric-peer:latest	"peer node start"	5 minutes ago	Up 5 minutes
25bb5beea4db	hyperledger/fabric-peer:latest	"peer node start"	5 minutes ago	Up 4 minutes
b9068ee6d211	hyperledger/fabric-peer:latest	"peer node start"	5 minutes ago	Up 5 minutes
46faf6a1fd65	hyperledger/fabric-tools:latest	"/bin/bash"	5 minutes ago	Up 5 minutes
2578140eb85c	hyperledger/fabric-orderer:latest	"orderer"	5 minutes ago	Up 4 minutes
87d93c839644	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se..."	6 minutes ago	Up 5 minutes
9a8f21bae7b7	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se..."	6 minutes ago	Up 6 minutes
360289bb2894	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se..."	6 minutes ago	Up 5 minutes
3e4a223ff49f	hyperledger/fabric-ca:latest	"sh -c 'fabric-ca-se..."	6 minutes ago	Up 5 minutes

그림 12. 최종적으로 블록체인 네트워크에 등록된 구성원들의 docker 상태 log

2) 체인코드

2-1) Award 체인코드

```

cc 배포를 위해 패키지와 합니다.

BUILD SUCCESSFUL in 677ms
4 actionable tasks: 4 up-to-date
CC 패키지를 awardOrg1의 peer0에게 설치합니다.
2022-07-25 13:26:02.337 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nLawardCC_1.0:12b3cc84177307bc7dd339313c1d1699f3b730ba1e9bc98517559e467bfc1a9d\022\013awardCC_1.0" >
2022-07-25 13:26:02.337 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: awardCC_1.0:12b3cc84177307bc7dd339313c1d1699f3b730ba1e9bc98517559e467bfc1a9d
정상 설치되어 있는지 확인합니다
awardCC_1.0:12b3cc84177307bc7dd339313c1d1699f3b730ba1e9bc98517559e467bfc1a9d
awardOrg1의 peer0에게 cc 패키지를 승인 받습니다.
2022-07-25 13:26:04.417 KST [chaincodeCmd] ClientWait -> INFO 001 txid [20994406cff4a81a023f628a2070b724eacf5b23e203e35c3db04eafceefb09] committed with status (VALID) at localhost:7051
{
  "approvals": {
    "AwardOrg1MSP": true
  }
}
CC 패키지를 commit 합니다.
2022-07-25 13:26:06.968 KST [chaincodeCmd] ClientWait -> INFO 001 txid [62a8e4b44b7f2f64925738043ddaad3b57933f312df0ff4cfff019527d4c72b00] committed with status (VALID) at localhost:7051
Committed chaincode definition for chaincode 'awardCC' on channel 'awardchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [AwardOrg1MSP: true]

```

그림 13. Award 체인코드 패키지 생성, 설치, 승인, 커밋 작업 log

2-2) Education 체인코드

```

cc 배포를 위해 패키지와 합니다.
Starting a Gradle Daemon (subsequent builds will be faster)

BUILD SUCCESSFUL in 6s
4 actionable tasks: 4 up-to-date
CC 패키지를 educationOrg1의 peer0에게 설치합니다.
2022-07-25 13:25:54.236 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nPeducationCC_1.0:d7a625ec9323c28740b396fe7c17de25f0de850642bccebed940c6e397eb2db1\022\017educationCC_1.0" >
2022-07-25 13:25:54.236 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: educationCC_1.0:d7a625ec9323c28740b396fe7c17de25f0de850642bccebed940c6e397eb2db1
정상 설치되어 있는지 확인합니다
educationCC_1.0:d7a625ec9323c28740b396fe7c17de25f0de850642bccebed940c6e397eb2db1
educationOrg1의 peer0에게 cc 패키지를 승인 받습니다.
2022-07-25 13:25:56.324 KST [chaincodeCmd] ClientWait -> INFO 001 txid [8f4204a9f28872372b5fceb9799d8944ec8f005b8cb700568887625974d52a] committed with status (VALID) at localhost:6051
{
  "approvals": {
    "EducationOrg1MSP": true
  }
}
CC 패키지를 commit 합니다.
2022-07-25 13:25:58.783 KST [chaincodeCmd] ClientWait -> INFO 001 txid [e0b00b3b7e29b761286ee89b2fdfa2cbdc680d20c484c3cfd83bdad94a5f7264] committed with status (VALID) at localhost:6051
Committed chaincode definition for chaincode 'educationCC' on channel 'educationchannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [EducationOrg1MSP: true]

```

그림 14. Education 체인코드 패키지 생성, 설치, 승인, 커밋 작업 log

2-3) License 체인 코드

```

cc 배포를 위해 패키지와 합니다.

BUILD SUCCESSFUL in 631ms
4 actionable tasks: 4 up-to-date
CC 패키지를 licenseOrg1의 peer0에게 설치합니다.
2022-07-25 13:26:10.521 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 001 Installed remotely: response:<status:200 payload:"\nLicenseCC_1.0:14279cf25924b39cd037cffbacd9b237e6fbce9314106536ffcb758da9ad0fb\022\rlicenseCC_1.0" >
2022-07-25 13:26:10.521 KST [cli.lifecycle.chaincode] submitInstallProposal -> INFO 002 Chaincode code package identifier: licenseCC_1.0:14279cf25924b39cd037cffbacd9b237e6fbce9314106536ffcb758da9ad0fb
정상 설치되어 있는지 확인합니다
licenseCC_1.0:14279cf25924b39cd037cffbacd9b237e6fbce9314106536ffcb758da9ad0fb
licenseOrg1의 peer0에게 cc 패키지를 승인 받습니다.
2022-07-25 13:26:12.600 KST [chaincodeCmd] ClientWait -> INFO 001 txid [f529a4d20f42181f9f21704a2aa6b02211493c7a1cc97a4ed19a8e0eb9a35a4] committed with status (VALID) at localhost:8051
{
  "approvals": {
    "LicenseOrg1MSP": true
  }
}
CC 패키지를 commit 합니다.
2022-07-25 13:26:15.049 KST [chaincodeCmd] ClientWait -> INFO 001 txid [8e6358b8acf9068fd0f28f39023afd8a96241147d537d56709d20ebd789c7838] committed with status (VALID) at localhost:8051
Committed chaincode definition for chaincode 'licenseCC' on channel 'licensechannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [LicenseOrg1MSP: true]

```

그림 15. License 체인코드 패키지 생성, 설치, 승인, 커밋 작업 log

3) 최종 docker 상태

```
thuthi@thuthi-server:~/workspace/graduate$ docker ps --format "table {{.ID}}\t{{.Names}}\t{{.Status}}"
```

CONTAINER ID	NAMES	STATUS
18b7f7a6dee4	dev-peer0.licenseOrg1.aomd.com-licenseCC_1.0-14279cf25924b39cd037cfffba9b237e6fbce9314106536ffc0b758da9ad0fb	Up 5 minutes
6c8d5295b10e	dev-peer0.awardOrg1.aomd.com-awardCC_1.0-12b3cc84177307bc7dd339313c1d1699f3b730ba1e9bc98517559e467bfc1a9d	Up 5 minutes
79ddcd02783c	dev-peer0.educationOrg1.aomd.com-educationCC_1.0-d7a625ec9323c28740b396fe7c17de25f0de850642bccebed940c6e397eb2db1	Up 5 minutes
18f75284eba1	peer0.licenseOrg1.aomd.com	Up 13 minutes
25bb5beea4db	peer0.awardOrg1.aomd.com	Up 13 minutes
b9068ee6d211	peer0.educationOrg1.aomd.com	Up 13 minutes
46faf6a1fd65	cli	Up 13 minutes
2578140eb85c	orderer.aomd.com	Up 13 minutes
87d93c839644	ca_ordererOrg	Up 14 minutes
9a8f21hae7b7	ca_educationOrg1	Up 14 minutes
360289bb2894	ca_licenseOrg1	Up 14 minutes
3e4a223ff49f	ca_awardOrg1	Up 14 minutes

그림 16. 체인코드까지 모두 설치한 docker 상태 log

2.3. 프론트엔드(Front-End)

1) 페이지 라우팅 구조

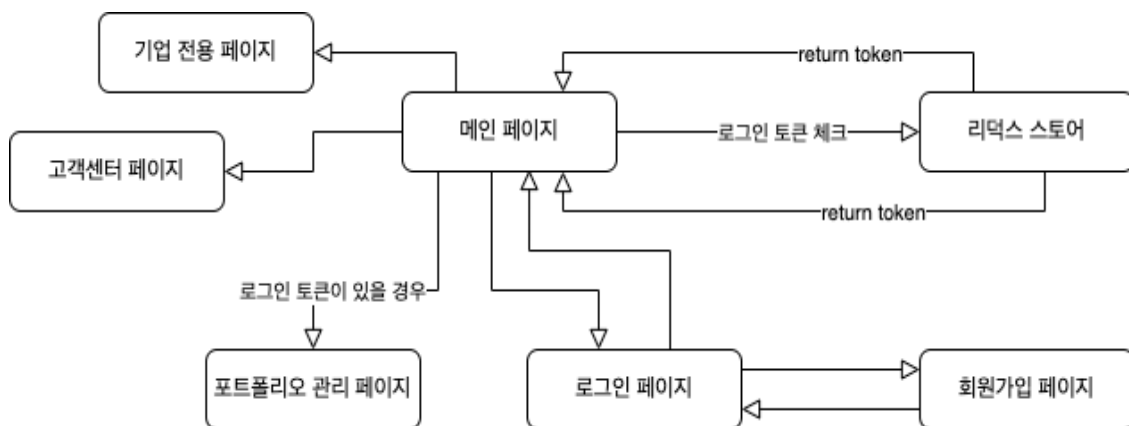


그림 17. 페이지 라우팅 구조

2) 포트폴리오 페이지 드래그 앤 드랍 컨텍스트 구조

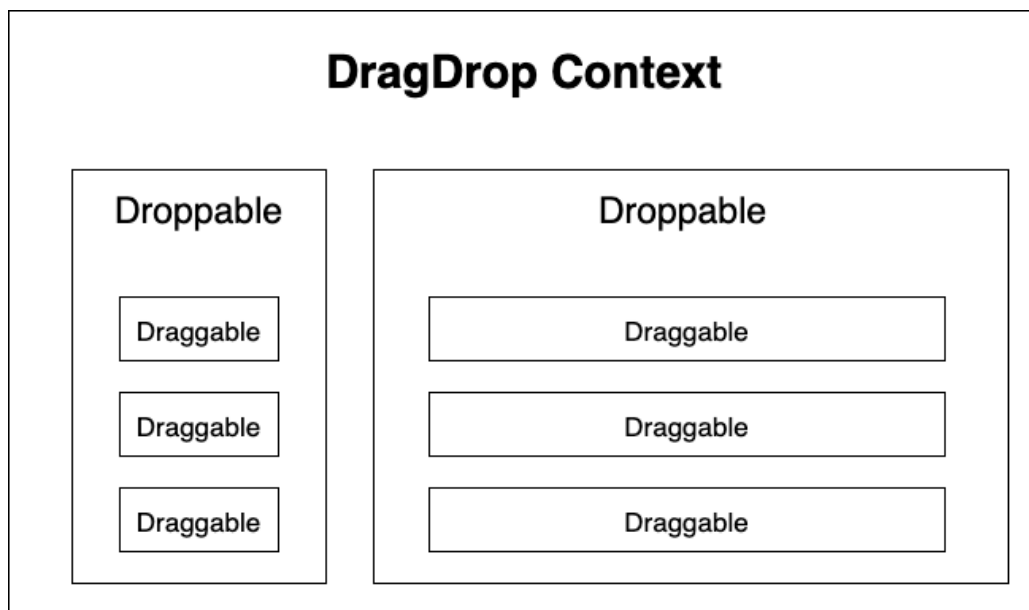


그림 18. 드래그 앤 드랍 컨텍스트 구조 (DND Context)

3) 데이터 상태 관리

핵심 로직 및 개념들만 뽑아서 나열하였다.

```
import { configureStore } from '@reduxjs/toolkit';
import loginDataReducer from './LoginCheck';
import rawDataReducer from './RawdataSlice';
import portDataReducer from './PortdataSlice';

export default configureStore({
  reducer: {
    loginData: loginDataReducer,
    rawData: rawDataReducer,
    portData: portDataReducer,
  },
});
```

코드 1. 리덕스 스토어(Store.js)

```
import { createSlice } from '@reduxjs/toolkit';

export const LoginCheck = createSlice({
  name: 'loginData',
  initialState: {
    loginState: false,
  },
  reducers: {
    SuccessLogin: state => {
      state.loginState = true;
    },
    SuccessLogout: state => {
      state.loginState = false;
    },
  },
});
```

```

    },
  },
});

export const { SuccessLogin, SuccessLogout } = LoginCheck.actions;
export const selectLoginData = state => state.loginData;
export default LoginCheck.reducer;

```

코드 2. 로그인 토큰 체크 리듀서 (LoginCheck.js)

```

export const RawdataSlice = createSlice({
  name: 'rawData',
  initialState: {
    origin: {
      title: '나의 데이터',
      items: data,
    },
  },
  reducers: {
    DragdataChange: (state, action) => {
      state.origin.items = action.payload;
    },
  },
});

export const { DragdataChange } = RawdataSlice.actions;
export const selectRawData = state => state.rawData;
export default RawdataSlice.reducer;

```

코드 3. 개인 정보 데이터 전용 리듀서 (RawdataSlice.js)

```

export const PortdataSlice = createSlice({
  name: 'portData',
  initialState: {
    port1: {
      title: '포트폴리오 1',
      items: data1,
      introductions: introdata1,
    },
  },
  reducers: {
    DragdataChangePort: (state, action) => {
      state.port1.items = action.payload;
    },
    DeletedataChangePort: (state, action) => {
      state.port1.items = action.payload;
    },
  },
});

export const { DragdataChangePort, DeletedataChangePort } =
PortdataSlice.actions;
export const selectPortData = state => state.portData;
export default PortdataSlice.reducer;

```

코드 4. 포트폴리오 데이터 전용 리듀서 (PortdataSlice.js)

```

export function ChangeData() {
  const dispatch = useDispatch();
  const originData = useSelector(selectRawData);
  const portData = useSelector(selectPortData);

  const onDragEnd = result => {

```



```

    if (!result.destination) return;

    const { source, destination } = result;

    if (source.droppableId !== destination.droppableId) {
        const sourceColumn = originData['origin'];
        const destColumn = portData['port1'];
        const sourceItems = [...sourceColumn.items];
        const destItems = [...destColumn.items];

        for (var index_11 = 0; index_11 < 3; index_11++) {
            ...
        }

        ...

        dispatch(DragdataChangePort(destItems));
    } else { return; }
  }
};

...

```

코드 5. 데이터 가져오기와 디스패치 및 드래그 앤 드랍 구현(ChangeData.js)

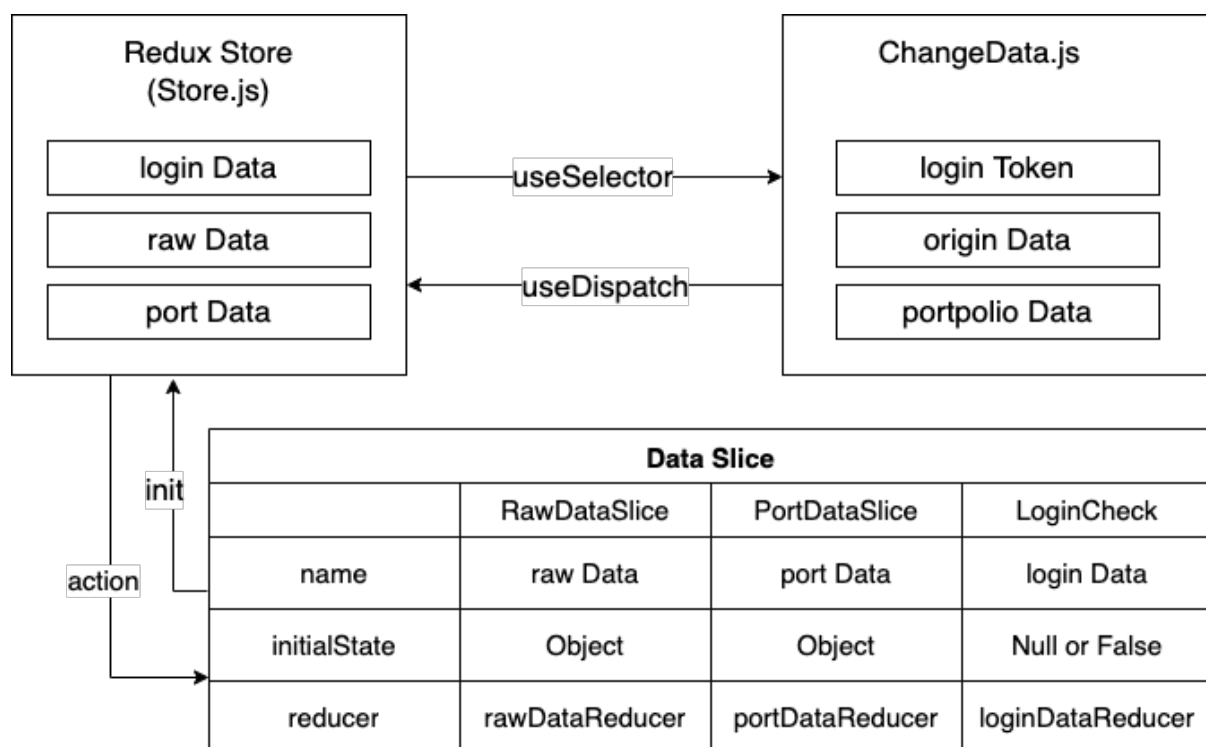


그림 19. 리덕스 통신 구조

4) UX/UI 및 페이지 디자인

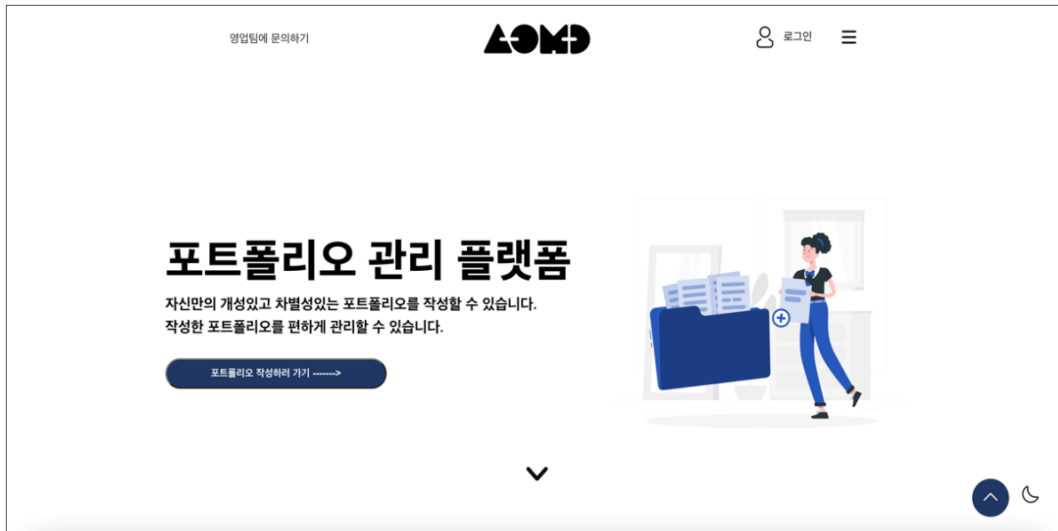


그림 20-1. 메인페이지



그림 20-2. 메인페이지 (카테고리 클릭 했을 경우)

본 플랫폼에 처음 들어왔을 경우 보일 페이지이다. 아래로 스크롤 하면 애니메이션 효과로 해당 플랫폼을 소개하는 문구와 그림이 차례대로 나온다. 오른쪽 상단의 카테고리 버튼(햄버거 버튼이라고도 불린다)을 누르면 카테고리 바가 나오고 각각의 텍스트를 클릭하면 해당 페이지로 이동한다. 오른쪽 하단에는 최상의 사용자 경험(UX)을 제공하기 위해 페이지 끌어올리기 버튼과 다크모드 버튼을 구현하였다.

The login page features a clean, minimalist design. At the top left is the 'AOMD' logo, and at the top right are links for '로그인' (Login) and a menu icon. The central form is a light blue box with rounded corners, containing the following elements from top to bottom: an 'ID' input field, a 'PASSWORD' input field, a dark blue '로그인' (Login) button, a link '아이디가 없으신가요?' (Don't have an ID?), a dark blue '회원가입' (Sign Up) button, a note '소셜 로그인을 통한 간편 회원가입' (Easy sign-up via social login), and two social login buttons: a yellow '카카오 로그인' (Kakao Login) button and a white '구글 로그인' (Google Login) button with a Google logo. A small circular logo is in the bottom right corner of the page.

그림 21-1. 로그인 페이지

The sign-up page has a similar layout to the login page. It includes the 'AOMD' logo and '로그인' (Login) link at the top. The central form is a light blue box with rounded corners, containing: a 'name' input field, a 'password' input field, an 'email address' input field, a dark blue '아이디 생성' (Create ID) button, a link '이미 회원가입하셨나요?' (Already signed up?), and a dark blue '로그인하러 가기' (Go to login) button. A small circular logo is in the bottom right corner of the page.

그림 22-2. 회원가입 페이지

메인 페이지에서 오른쪽 상단에 있는 로그인 버튼을 클릭했을 경우 이동하는 로그인 페이지와 회원가입 페이지이다. 상업 목적이 아니므로 간단한 개인 정보만을 통해 회원가입이 가능하고 추가로 카카오와 구글을 통한 소셜 로그인도 가능하다.

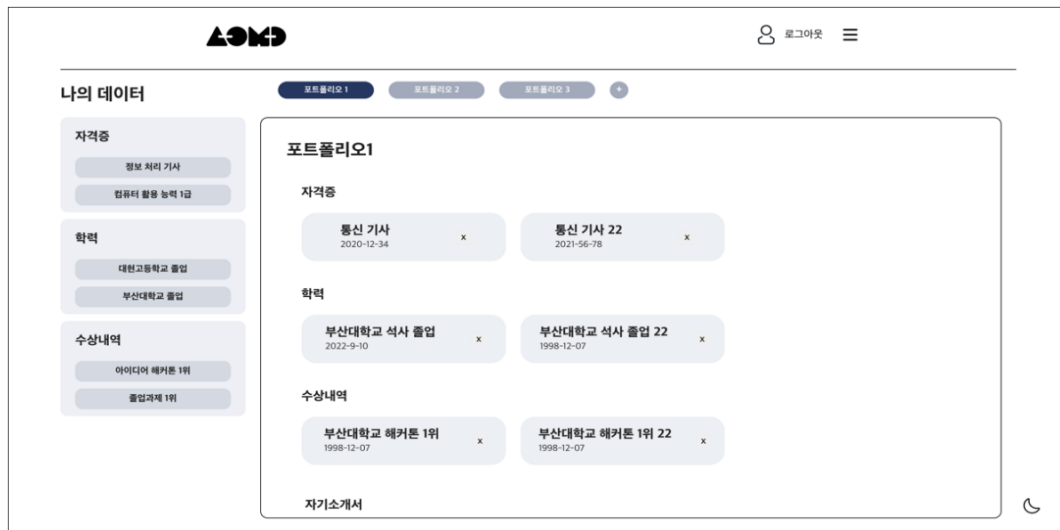


그림 23-1. 포트폴리오 관리 페이지



그림 23-2. 포트폴리오 관리 페이지(아래로 스크롤 했을 경우)

최상의 사용자 경험을 제공하기 위하여 드래그 앤 드랍(Drag and Drop) 기능을 구현하였다. 페이지의 왼쪽에는 블록체인에서 가져온 나의 개인정보 데이터를 배치하고 오른쪽에는 사용자가 만든 포트폴리오를 배치하였다. 왼쪽에 있는 데이터는 드래그(Drag)할 수 있으며 오른쪽에 놓을 경우(Drop) 자동으로 데이터가 추가된다.



그림 23-3. 포트폴리오 관리 페이지 (자동 저장 진행시)

사용자가 자기소개서 문항 및 내용을 수정할 경우 내부 알고리즘을 통해 몇 초를 간격으로 두고 자동으로 서버에 저장 요청을 보낸다. 이는 네트워크 불안정과 같은 갑자기 일어날 수 있는 상황에 대비한 것으로 작성 중인 내용을 잃어버리지 않고 안전하게 저장할 수 있다. 또한 저장하기 버튼을 통해서도 내용을 저장할 수 있다.

3. 사용자 시나리오

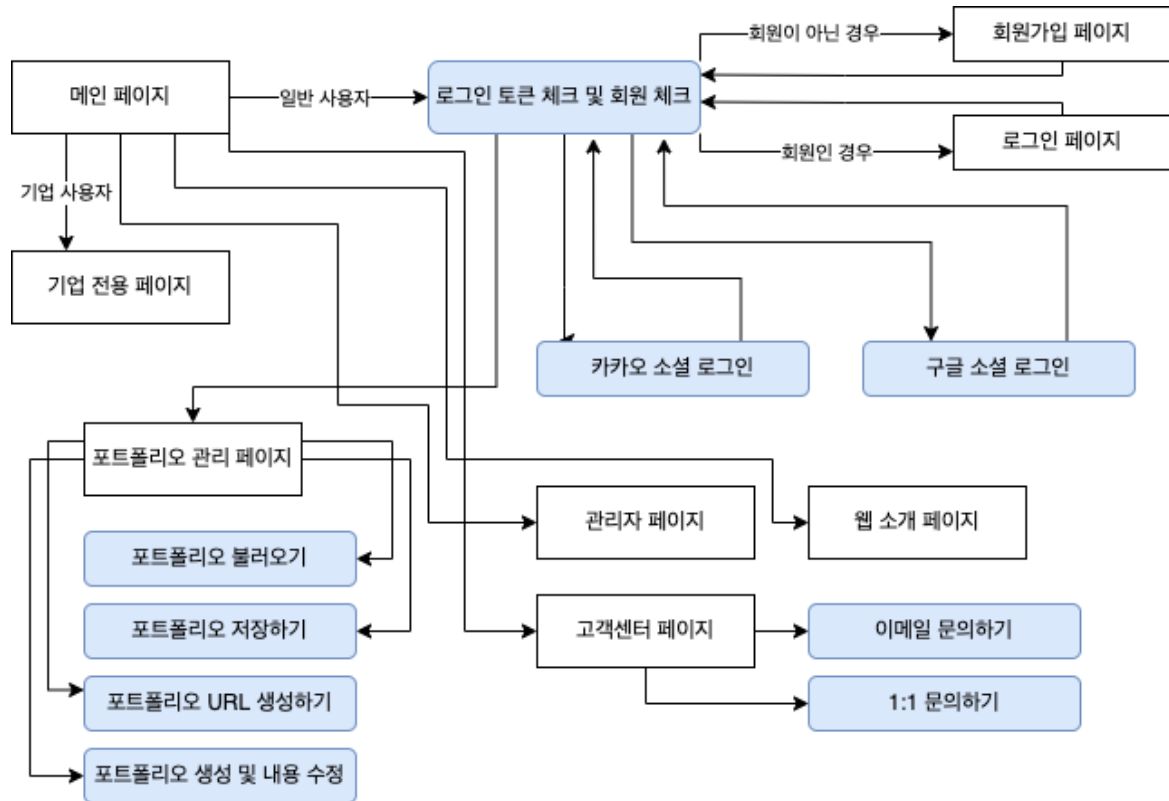


그림 24. 프로세스 다이어그램(Process Diagram)

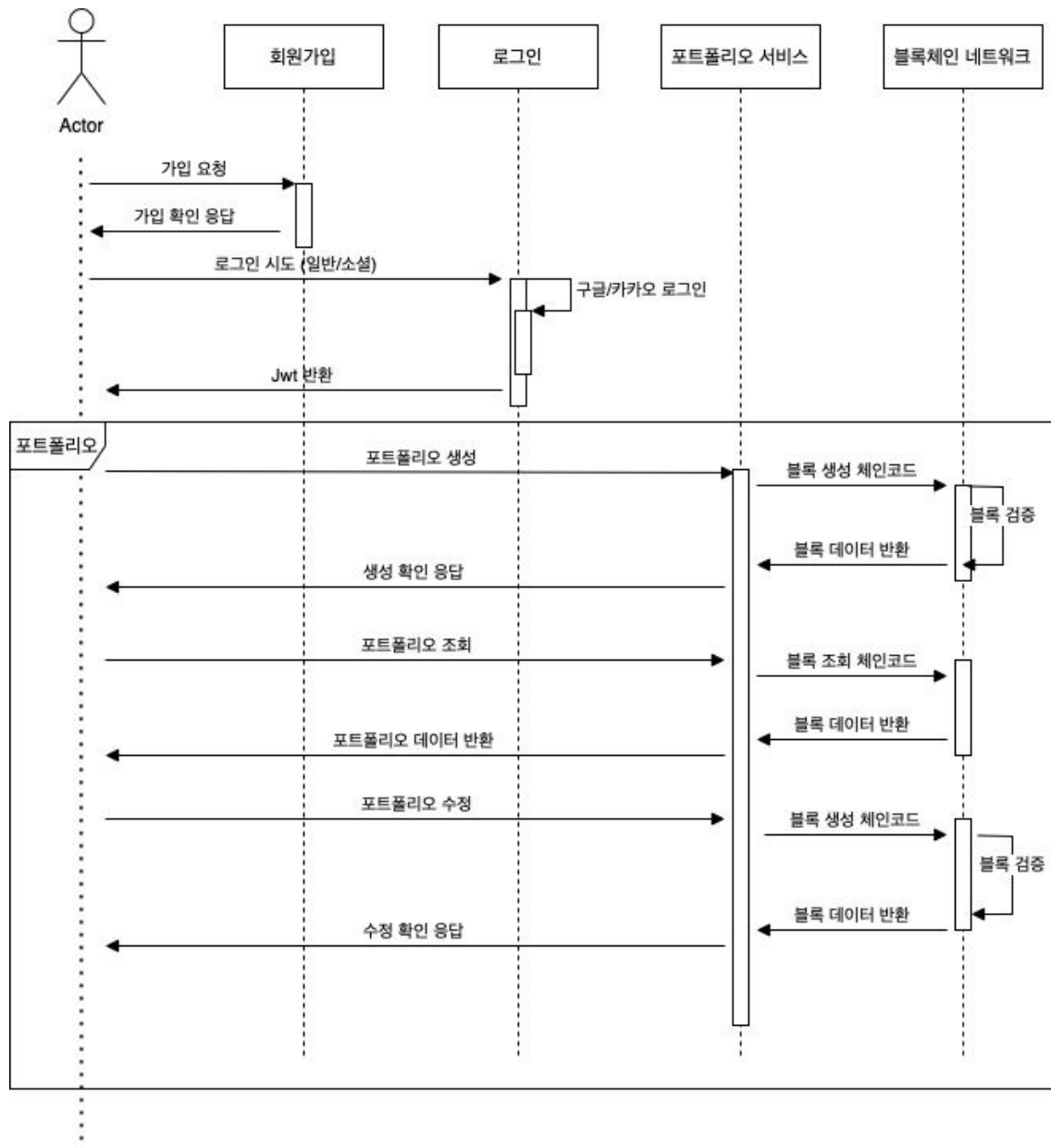


그림 25. 시퀀스 다이어그램(Sequence Diagram)

4. 갱신된 과제 추진 계획

4.1. 개발 일정 변경

a) 초기 개발 일정

5 월					6 월					7 월					8 월					9 월			
2주	3주	4주	5주		1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주
블록체인 스터디																							
					개발 프레임워크 스터디																		
								블록체인 네트워크 구축															
								체인코드 개발															
												서버 개발											
												웹 UI 및 개발											
															테스트 및 수정								
																				최종 점검 및 최종 발표 준비			

표 1. 초기 개발 일정

b) 변경된 개발 일정

5 월					6 월					7 월					8 월					9 월			
2주	3주	4주	5주		1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주
블록체인 스터디																							
					개발 프레임워크 스터디																		
								블록체인 네트워크 구축															
								체인코드 개발															
												서버 개발											
												웹 UI 및 개발											
															테스트 및 수정								
																				최종 점검 및 최종 발표 준비			

표 2. 변경된 개발 일정

5. 구성원별 진척도

5.1. 현재까지 진행 상황

(a) 조병우

- 1) 블록체인 네트워크 채널, 구성원 정의
- 2) 블록체인 네트워크 생성 스크립트 작성
- 3) 체인코드 정의
- 4) 체인코드 패키지 생성, 설치, 동의 스크립트 작성
- 5) 체인코드 테스트 및 서버 연결 테스트

(b) 조현우

- 1) DB 스키마 정의
- 2) Spring ORM JPA 엔티티 정의
- 3) Spring Security 를 활용한 자체 회원가입/로그인 기능 개발
- 4) Oracle Cloud 에 개발서버 배포 및 CROS 문제 해결

(c) 서지원

- 1) 페이지 라우팅 구조 설계 및 각 컴포넌트 구성
- 2) 리덕스 통신 구조 설계
- 3) 최상의 UI/UX 를 위한 추가적인 요소 구현 (다크모드, Move Top, 저장 중 모션)
- 4) AWS-EC2 인스턴스를 통한 웹 사이트 배포

5.2. 앞으로의 추진 계획

(a) 조병우

- 1) 블록체인 네트워크 고도화
- 2) 블록체인 네트워크 Java SDK grpc 문제 해결
- 3) 백엔드 서버 비즈니스 로직 개발

(b) 조현우

- 1) OAuth2.0 을 이용한 소셜로그인 구현
- 2) API 정의 및 API Doc 제작
- 3) 백엔드 서버 비즈니스 로직 개발
- 4) 백엔드 서버 테스트 코드 작성

(c) 서지원

- 1) 사용자 테스트를 통한 평가 및 인사이트 도출
- 2) 페이지 리렌더링 최소화를 통한 웹 페이지 최적화
- 3) 백엔드 API 를 통한 기능 개발