
REPORT



과 목 명	시스템분석및설계
-------	----------

담 당 교 수	박용범
---------	-----

학 과	소프트웨어학과
-----	---------

학 번	32193445
-----	----------

성 명	이재희
-----	-----

제 출 일	2020.06.30
-------	------------

시스템 분석 및 설계 기말 과제 보고서

목 차

1. Use case에 따른 UML 다이어그램 설계
2. DB ERD 설계
3. 실행 결과
4. 소스 코드(Github)

1. Use case에 따른 UML 다이어그램 설계

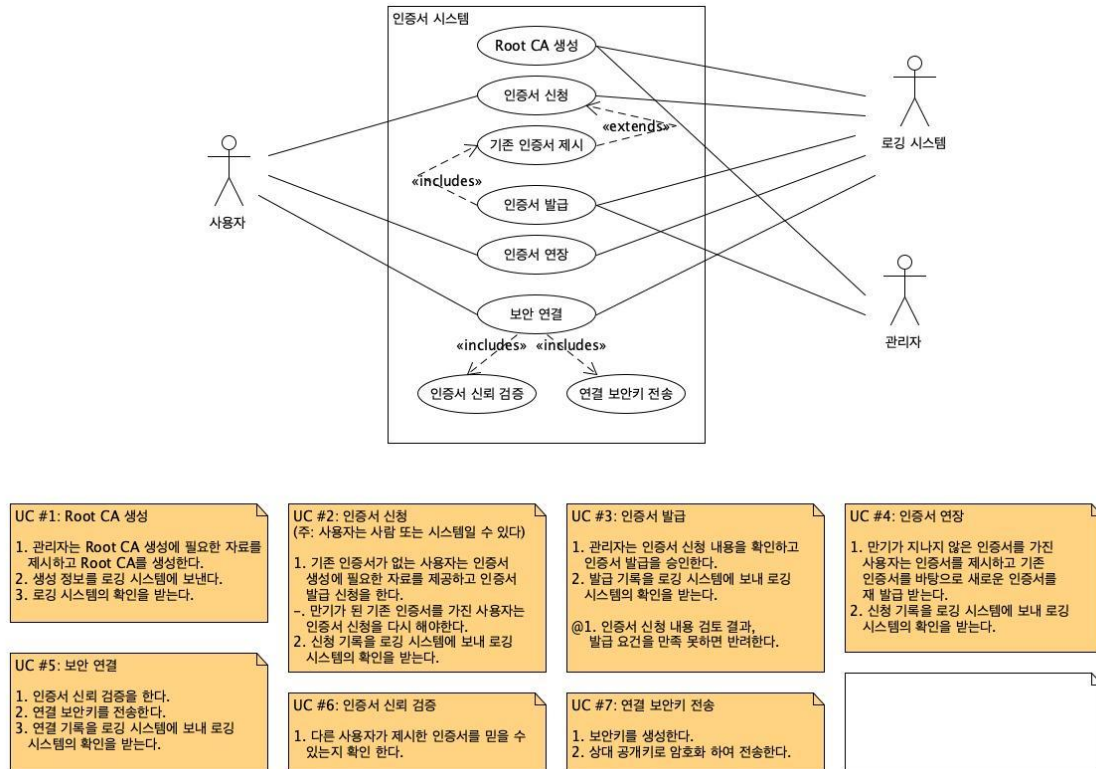


그림 1 - Use Case

Use Case에 나타난 3개의 Actor가 행동함에 따라 총 7개의 Use Case가 나타남을 확인할 수 있습니다.

Use Case에 대해서 구현에 앞서 UML 다이어그램을 그림으로써 시스템의 전반적인 설계와 Use Case에서 시스템 설계에 부족한 부분을 찾고 보완합니다.

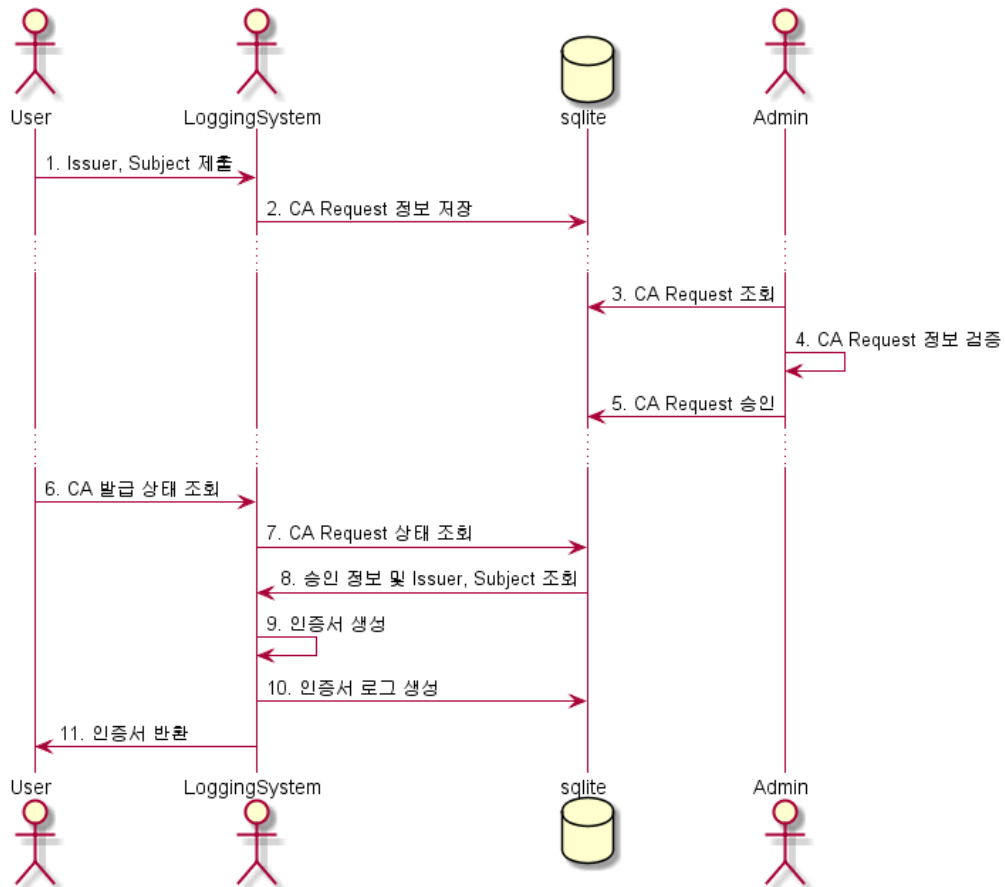


그림 2 - 인증서 요청 및 발급 시퀀스 다이어그램

사용자가 인증서 신청(UC#2)을 하고 관리자가 인증서 신청 내용을 확인하고 인증서 발급을 수행하면 로깅 시스템이 사용자에게 인증서를 발급(UC#2)하는 시퀀스 다이어그램입니다.

다른 Actor와 Database간의 커뮤니케이션에 대한 설계와 순서도가 필요했기에 시퀀스 다이어그램을 선택해 인증서 신청 과정과 최종적으로 사용자에게 인증서가 발급되는 과정을 그렸습니다.

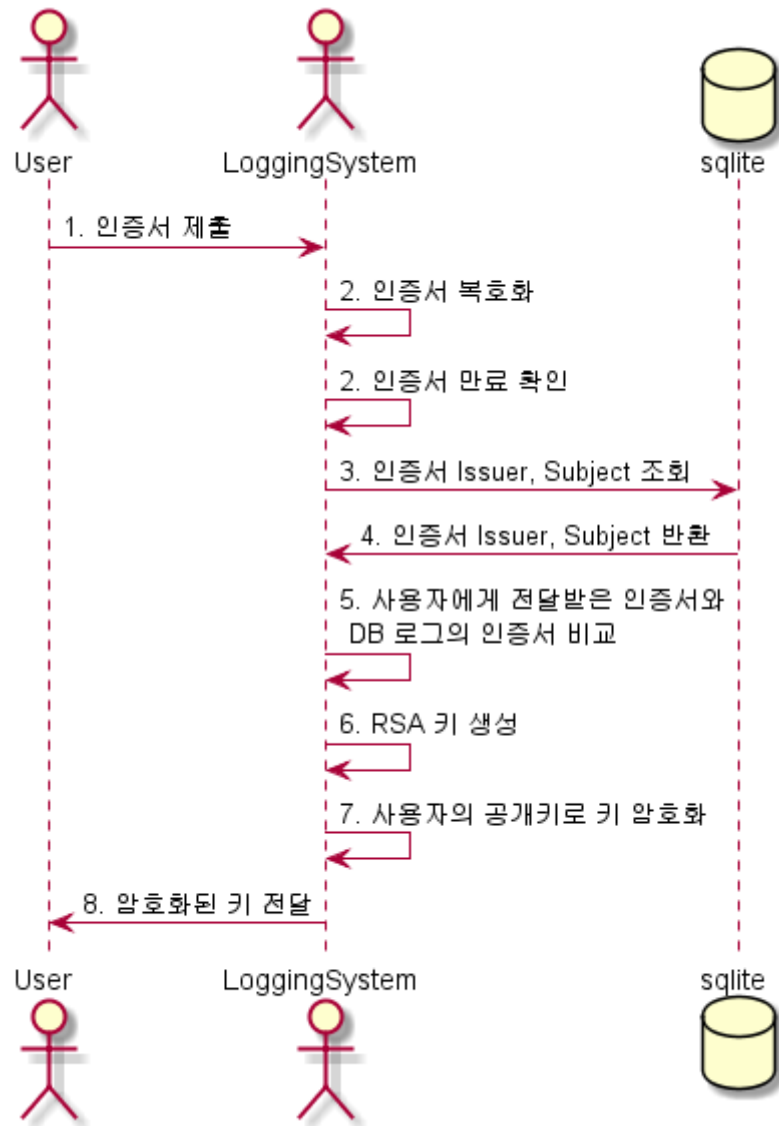


그림 3 - 보안 연결 시퀀스 다이어그램

사용자가 보안 연결(UC#5)을 신청하면 해당 인증서를 제출하고 인증서가 유효한지(UC#6) 검사한 후 보안 연결 키를 생성하여 사용자의 공개키로 암호화한 후 사용자에게 전달합니다(UC#7).

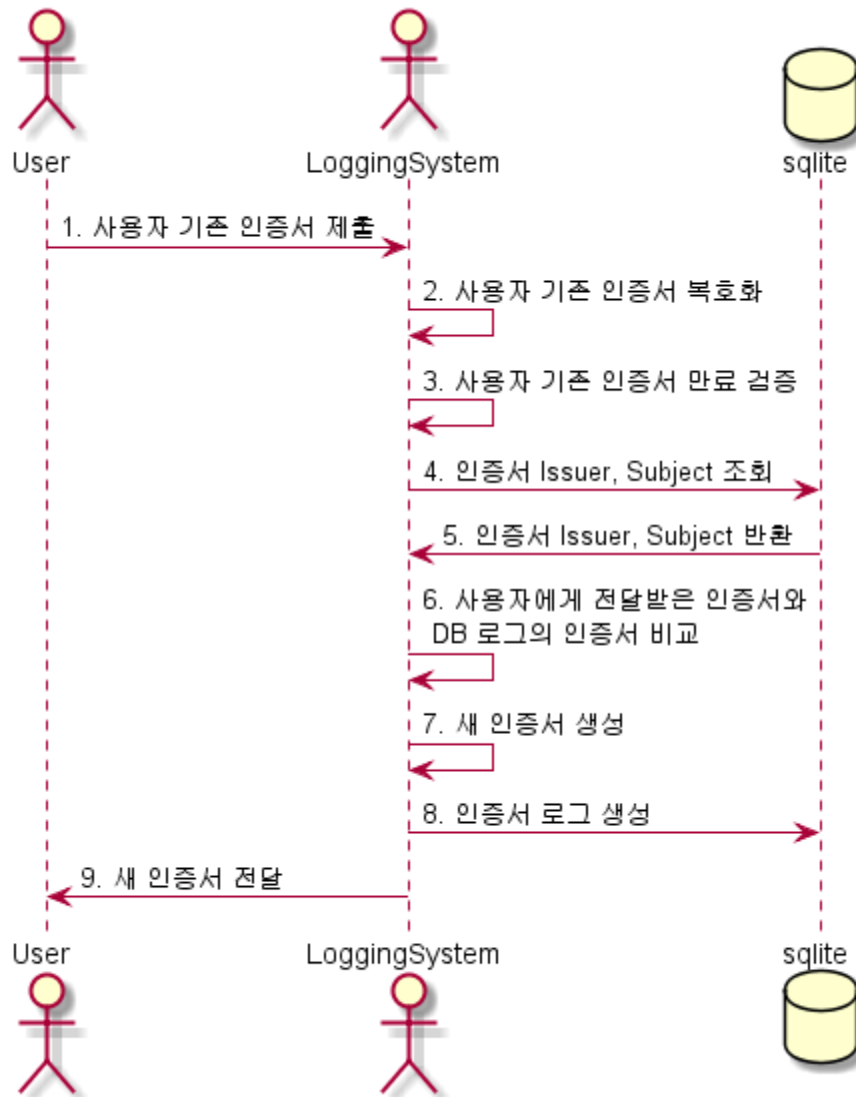


그림 4 - 인증서 연장 시퀀스 다이어그램

사용자가 인증서 연장(UC#4)을 요청하면 인증서의 기간이 유효한지 확인한 후 새로운 인증서를 새로 생성해 반환하는 과정이 그려진 시퀀스 다이어그램입니다.

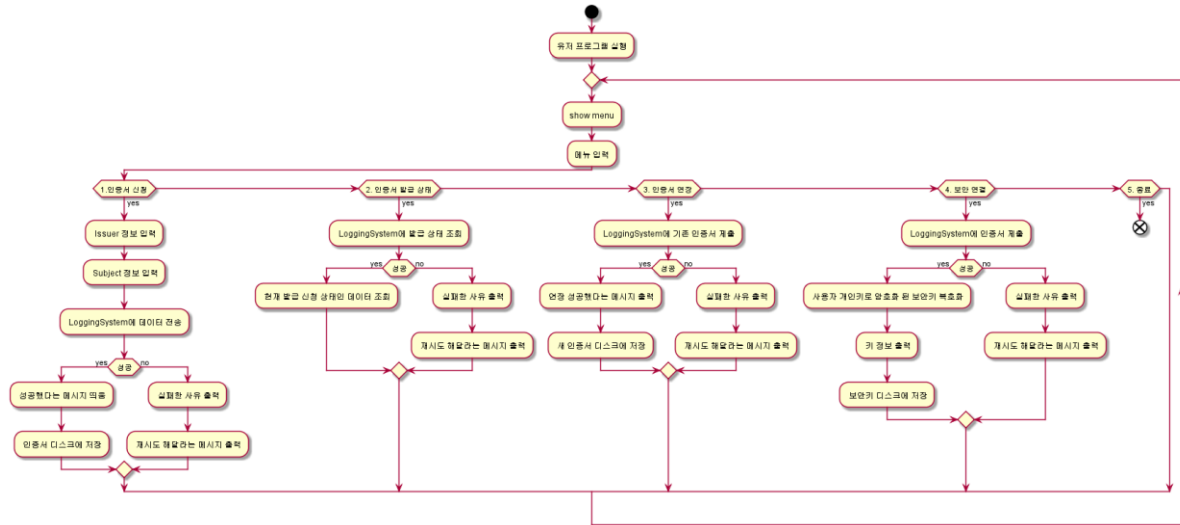


그림 5 - 유저 프로그램 액티비티 다이어그램

Use Case에 대한 과정 및 행동 정의는 시퀀스 다이어그램으로 전반적인 시스템을 설계했으니, 각 Actor인 User, Admin에 대한 동작 설계가 필요해서 액티비티 다이어그램을 사용했습니다.

프로그램 작동 시 행하는 행동과 사용자가 선택할 수 있는 메뉴와, 선택 시 행동 및 나타나는 메시지를 정의했습니다.

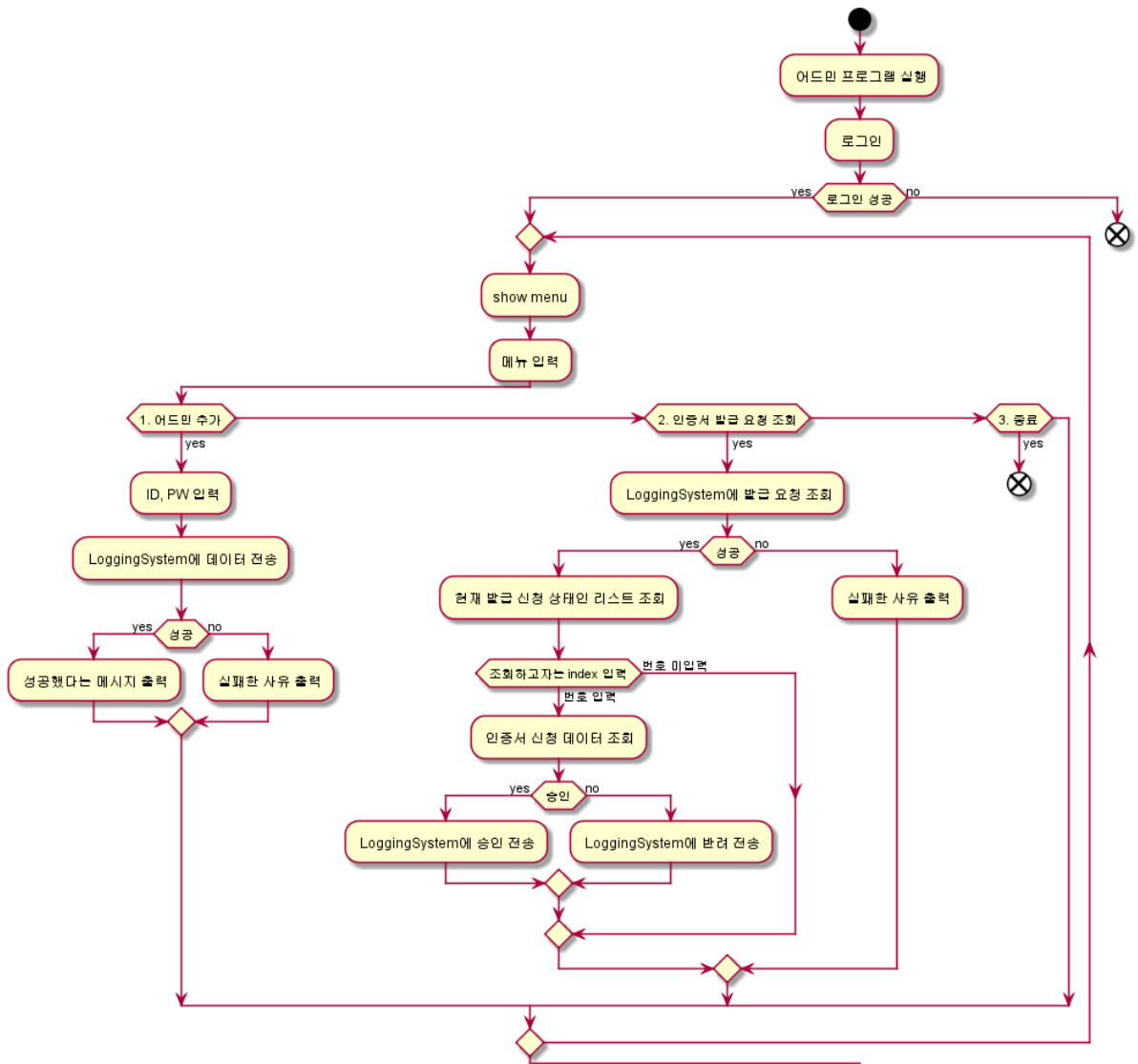


그림 6 - 관리자 프로그램 액티비티 다이어그램

관리자 프로그램 또한 유저 프로그램과 동일하게 작성했습니다.

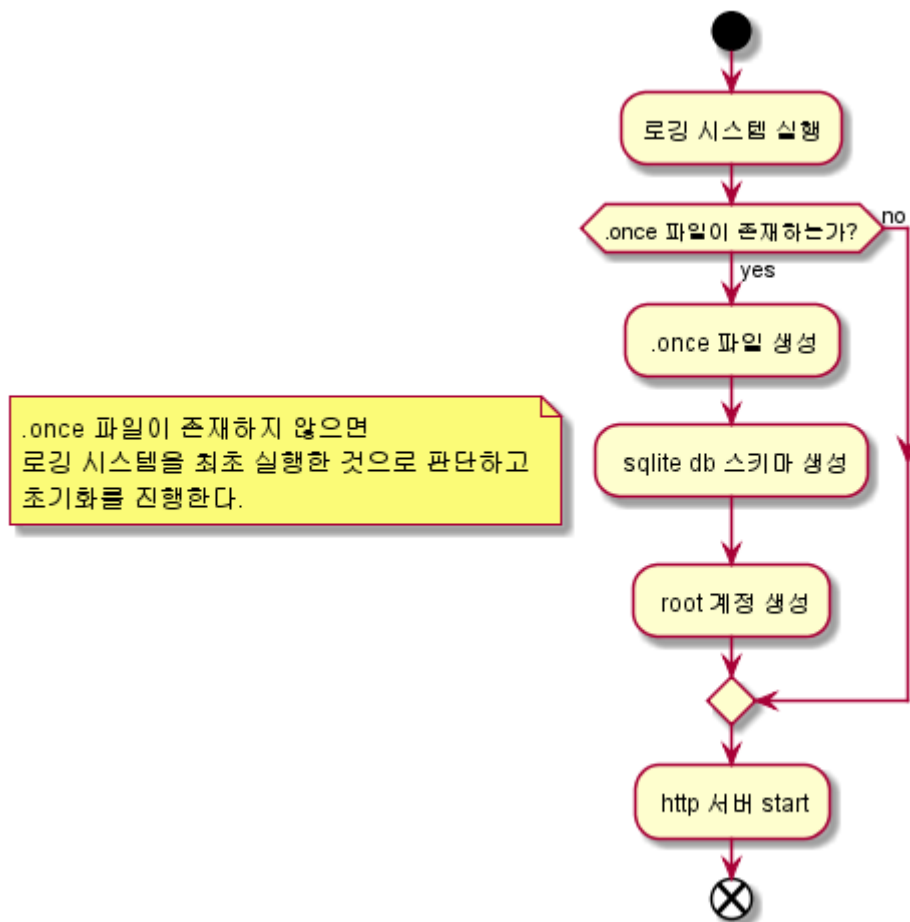


그림 7 - 로깅 시스템 액티비티 다이어그램

로깅 시스템 Actor는 스스로 이벤트를 발생시키는 Act보다는 다른 Actor로부터 상호 작용되는 기능이 더 많기 때문에 로깅 시스템 Actor가 행하는 행동은 시퀀스 다이어그램 위주로 정리하였습니다.

이외의 로깅 시스템 프로그램이 작동될 때 행동들은 액티비티 다이어그램으로 설계했습니다.

2. DB ERD 설계

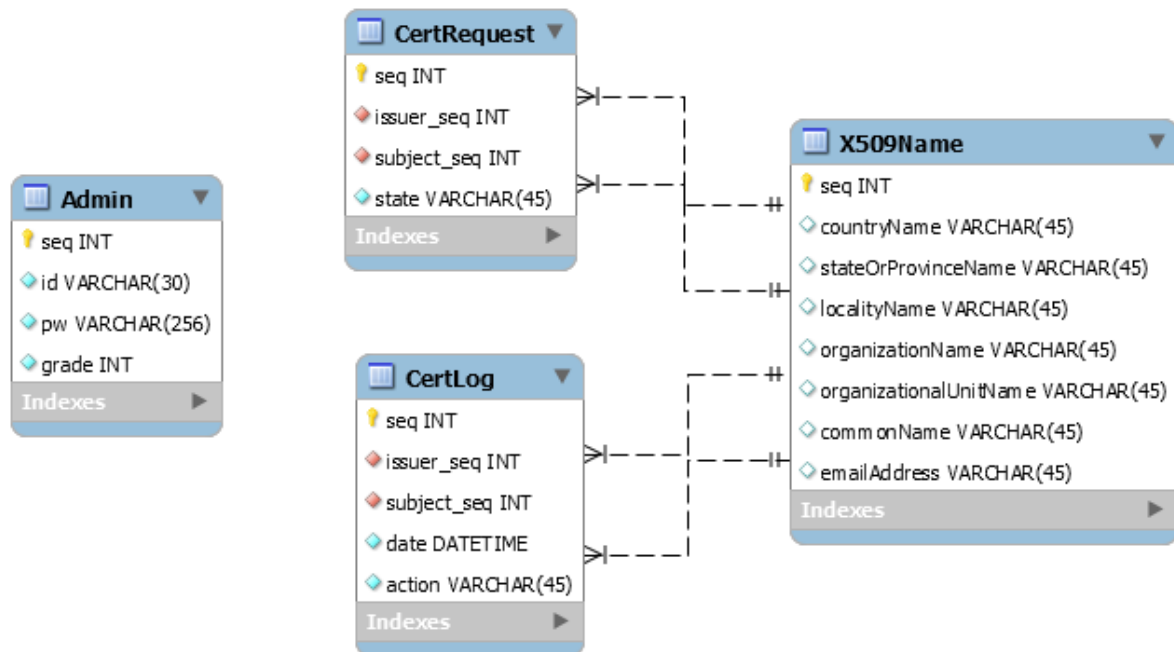


그림 8 - 데이터베이스 E-R 다이어그램

시퀀스 다이어그램의 흐름에서 사용되는 데이터와 pyopenssl 라이브러리에서 인증서를 발급하기 위해 사용되는 데이터에 대해서 E-R 다이어그램을 그렸습니다.

3. 실행 결과

1) 로깅 시스템

```
[C:\git:(master)] flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
First boot up the Logging System
initialize Logging System
initialize Database
Root Account is shown once, if you forgot root pw, delete the .once file and logging system w
ill be reset
Root ID : root
Root PW : 77532b45b36f49d595249fb859417415
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

로깅 시스템은 관리자와 유저 둘 다에게 상호 작용되어야 하기 때문에 네트워크로 연결되어 있다는 가정하에 HTTP 서버로 구성하였습니다.

Python Flask 라이브러리를 사용해 HTTP 서버를 구현하였고, flask run 명령어로 서버를 실행시킬 수 있습니다. 로깅 시스템 프로그램을 켜야 관리자 프로그램과 유저 프로그램이 정상 작동합니다.

최초 실행 시 DB 스키마를 생성하고 비밀번호를 랜덤으로 초기화한 root 계정을 생성하여 보여줍니다.

root 계정은 최초 실행 시에만 보여주며 로깅 시스템을 재 실행시에는 보여주지 않습니다.

2) 관리자 프로그램

```
관리자 ID : root
관리자 PW : 77532b45b36f49d595249fb859417415
0. 관리자 추가
1. Root CA 생성
2. 신청 목록 조회
3. 종료
메뉴를 선택하세요 :
```




관리자 프로그램을 켜면 관리자 id와 pw 입력을 요구합니다. 최초 실행 시 생성된 관리자 계정이 없으므로 로깅 시스템 최초 실행 시 생성되는 루트 계정을 이용해 로그인 할 수 있습니다.

루트 계정으로 로그인 시 4개의 메뉴를 확인할 수 있습니다.

```
메뉴를 선택하세요 : 0
새 관리자 ID : test
새 관리자 PW : test
관리자 생성에 성공했습니다.
└─[$] <git:(master)> python app.py
관리자 ID : test
관리자 PW : test
0. 신청 목록 조회
1. 종료
메뉴를 선택하세요 :
```

관리자 추가 메뉴 선택 시 루트 계정이 아닌 관리자 계정을 생성할 수 있으며, 해당 계정으로 접속 시 인증서 신청 목록을 조회하고 승인 및 거부만 할 수 있는 권한만 주어집니다.

```
메뉴를 선택하세요 : 1
Root CA 생성에 성공했습니다.
```

 root-cert.crt
 root-key.pem
 root-key.pub

Root CA 생성 메뉴 선택 시 로깅 시스템 폴더에 Root CA 파일과 키가 생성됩니다.

```
메뉴를 선택하세요 : 2
인증서 요청 목록
0. commomName : user-ca
1. commomName : user-ca
2. commomName : user-ca
승인 또는 거부할 인증서를 고르시오 :
```

신청 목록 조회 메뉴 선택 시 사용자가 신청한 인증서 요청 목록이 보이며 승인 및 거부할 인증서를 선택 후 사용자가 전송한 세부 정보를 확인할 수 있습니다.

```
승인 또는 거부할 인증서를 고르시오 : 1
countryName : kr
stateOrProvinceName : a
localityName : b
organizationName : c
organizationalUnitName : d
commonName : user-ca
emailAddress : e
1. 승인 2. 거부 : 1
승인에 성공했습니다.
```

```
승인 또는 거부할 인증서를 고르시오 : 1
countryName : kr
stateOrProvinceName : a
localityName : b
organizationName : c
organizationalUnitName : d
commonName : user-ca
emailAddress : e
1. 승인 2. 거부 : 2
거부에 성공했습니다.
```

관리자는 사용자가 보낸 정보를 확인하고 승인 또는 거부를 할 수 있습니다.

3) 사용자 프로그램

```
[L$] <git:(master)> python app.py
countryName(ex : kr) : kr
stateOrProvinceName : a
localityName : b
organizationName : c
organizationalUnitName : d
commonName : user-ca
emailAddress : e
User가 성공적으로 생성되었습니다.
0. 인증서 신청
1. 신청 목록 조회
2. 인증서 연장
3. 보안 연결
4. 종료
메뉴를 선택하세요 :
```

사용자 프로그램은 실행되면 사용자의 정보(subject)를 입력받습니다.

모든 정보를 입력 후 사용자가 선택할 수 있는 5가지의 메뉴를 확인할 수 있습니다.

```
메뉴를 선택하세요 : 0
인증서 요청이 성공했습니다.
```

인증서 신청 메뉴 선택 시 프로그램이 켜질 때 입력한 정보를 바탕으로 인증서 요청을 관리자에게 보냅니다.

```
메뉴를 선택하세요 : 1
인증서 요청 목록
0. 상태 : 승인 대기 중인 인증서
1. 상태 : 승인
2. 상태 : 거부
발행할 승인된 인증서를 고르시오 :
```

신청 목록 조회 메뉴 선택 시 지금까지 신청한 인증서 요청 목록을 확인할 수 있습니다.

상태가 “승인”인 인증서 요청 목록을 선택해 인증서를 발행할 수 있습니다.

```
발행할 승인된 인증서를 고르시오 : 1
인증서 발급에 성공했습니다.
```

```
ca.crt
encrypt_ca.crt
key.pem
key.pub
```

인증서 발행 성공 시 키와 인증서, root private key로 암호화된 인증서를 생성합니다.

```
발행할 승인된 인증서를 고르시오 : 0
인증서 발급에 실패했습니다.
사유 : The request is not confirm
```


잘 못 된 인증서 번호 입력 시 발급에 실패하며 사유를 띄웁니다.


```
메뉴를 선택하세요 : 2
인증서 갱신에 성공했습니다.
```

인증서 갱신 메뉴 선택 시 생성되어 있는 인증서를 전송해 인증서를 갱신합니다.

```
메뉴를 선택하세요 : 3
인증서 검증에 성공했습니다.
보안 연결 키가 생성되었습니다.
```

보안 연결 메뉴 선택 시 생성되어 있는 인증서를 전송해 검증한 후 로깅 시스템에서 새로운 키를 생성해 사용자의 공개키로 암호화 후 전송합니다.

 secure-key.pem

 secure-key.pub

보안 연결에 성공했다면 새로 생성된 키를 확인할 수 있습니다.

4. 소스 코드(Github)

<https://github.com/dku32193445/lecSystem-final-homework>