**Hamobee**

**개인별 보고서**

**B611132 윤지원**

**내가 수행한 것들**

**1. 프로젝트 기획**

**2. 회원관리 system 구현**

**3. 데이터베이스 설계**

**4. E-mail 기능 구현했지만 사용하지 않음**

**5. 모바일 코드 추가**

**6. 머신러닝**

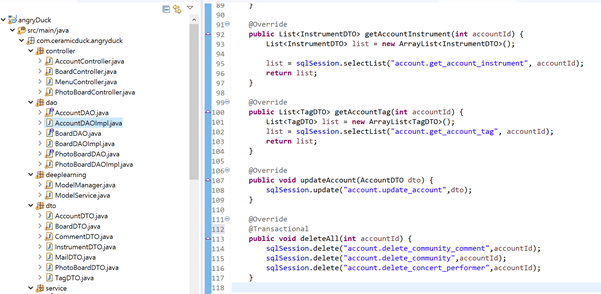
**1. 프로젝트 기획**

대학생활 중, 학업을 제외하고 가장 열심히 했던 활동이 오케스트라 활동이었다. 교내 오케스트라와 대학연합 오케스트라에서 활동하며, 다른 오케스트라 공연에 객원연주자로 참여하기도 하고, 객원연주자를 직접 모집하기도 했다. 오케스트라 공연 준비과정에서 일대일로 연락하는 방식의 객원 모집과정이 번거롭고 비효율적이라고 느꼈다. 그래서 이 과정을 더 효율적으로 개선할 수 있는 서비스를 구상하게 되었다. 공연 주최자와 연주자 모두 이용할 수 있는 웹사이트를 만들어보고자 하모비를 제작하게 되었다.

**2. 회원관리 system 구현**

회원관리 system 코드 내에서 동시에 여러 개의 sql문을 실행하여야 할 때는 transaction을 사용하여 구현했다.

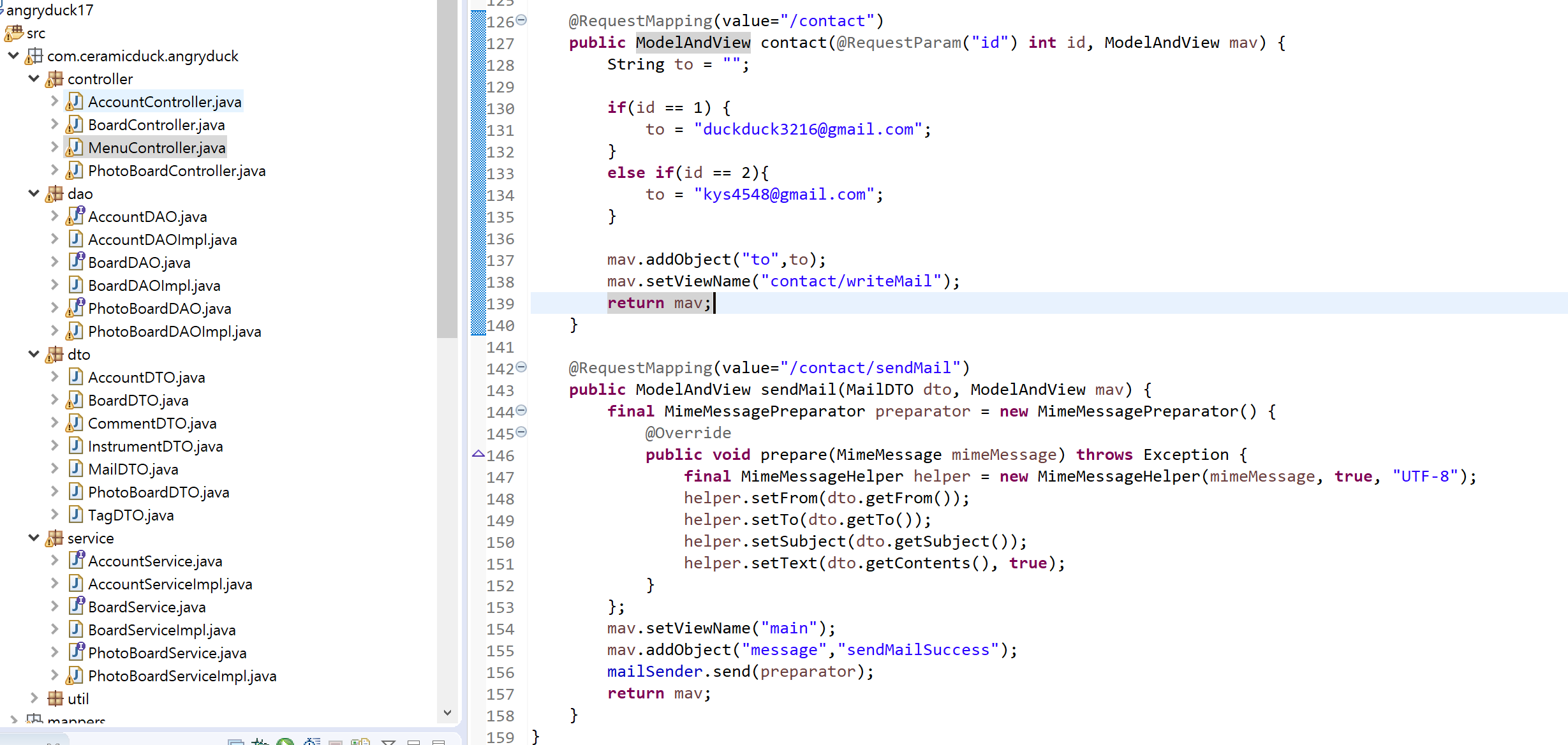
(@Transactional 이 그 표시)



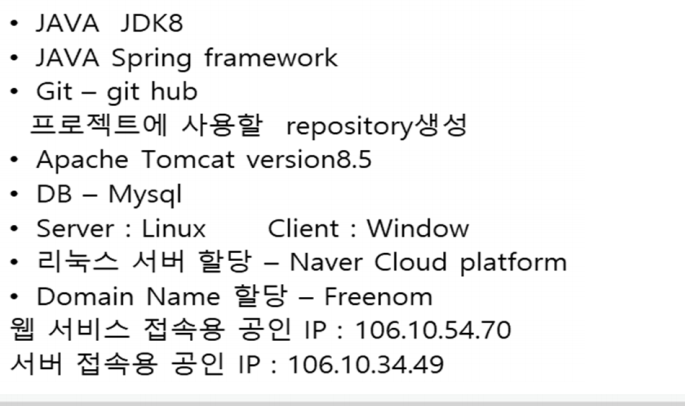
**3. 데이터 베이스 설계**

무결성 제약 조건을 지키면서 제 3정규형으로 설계하려 하였다. 초기에는 Oracle DB로 작성하였으나, 현재는 mysql DB으로 수정하였다. 서버를 네이버 클라우드 플랫폼을 사용하였는데 Oracle DB는 지원하지 않고, Mysql DB만 지원했기 때문이다.

**4.email 기능 구현 했지만 사용하지 않음**



E-mail 기능을 사용하지 않은 것은 보안 상의 이유이다.



졸업프로젝트 포스터에서 사용한 글

우리는 편의 상, 서버 접근 ID와 비밀번호를 가장 기본적인 것을 사용했다.

(ex : id : root, pw : root)

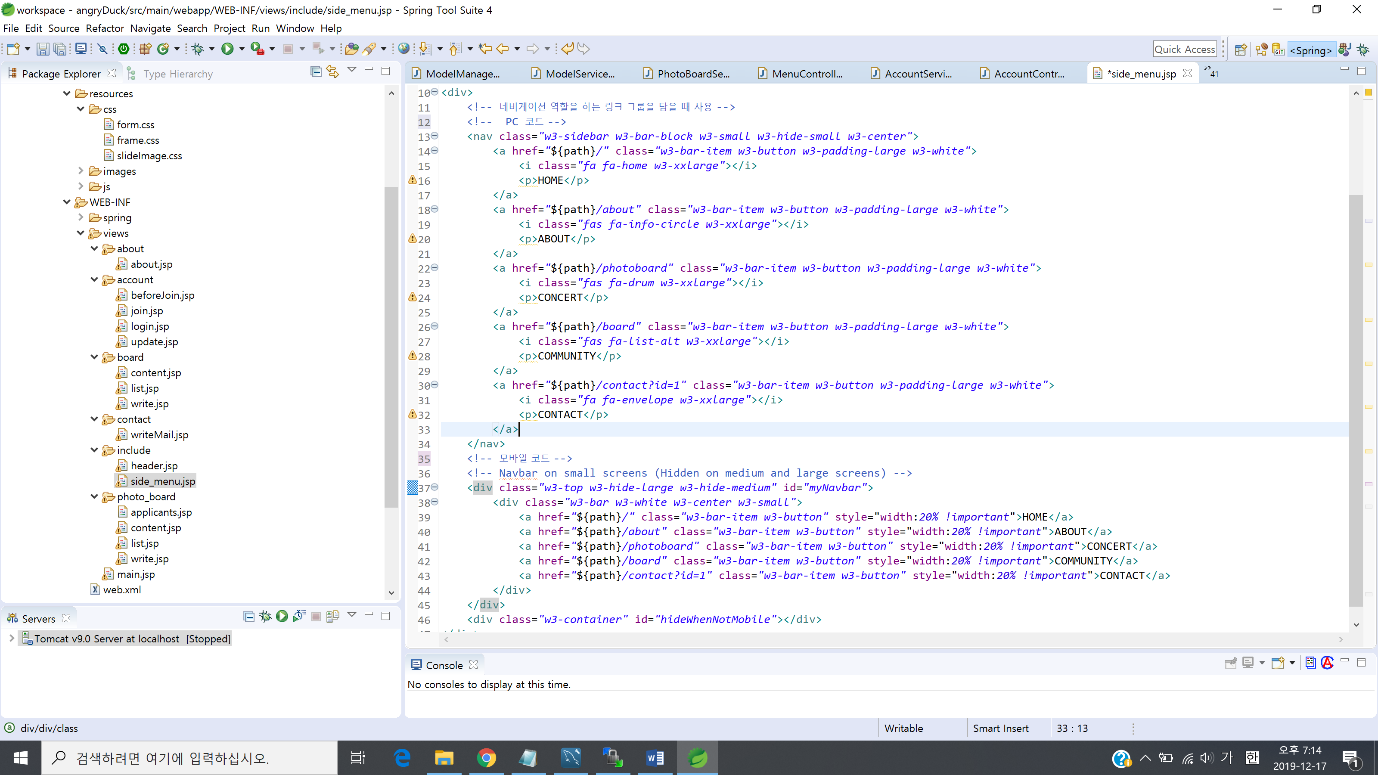
누군가 서버에 접근해 메일 구현 코드 내에 있는 email주소와 비밀번호를 이용하여 개발자들의 gmail에 로그인을 시도했던 것을 알게 되었다. 또한, 누군가의 접근에 의해 DB에 모든 table 정보들이 사라져 재입력했다.

다음의 사건을 통해 가장 기본으로 사용하는 비밀번호 사용해서는 안된다는 것을 느꼈고, 생각보다 해커는 가까이에 있다는 것을 알았다.

위험을 감수하며 이메일 기능을 구현하는 것은 무리인 것 같아 사용하지 않기로 결정했다.

**5. 모바일 코드 추가**

졸업전시회에서 모바일 환경에서도 사이트를 편리하게 이용할 수 있도록 개선해볼 것을 권유받았다. 따라서 그 부분을 보완했다.



하나의 예시로 pc환경에서는 메뉴가 왼쪽 옆에 존재하지만,

모바일에서는 메뉴가 나타나지 않아 모바일환경에서 작동하는 코드를

추가했다.



(모바일과 pc 화면 차이)

**6. 머신러닝 관련**

기획 당시에는 머하웃을 활용한 머신러닝을 사용하려고 했으나, softmax를 활용했다. 우리가 주로 java를 사용하여 개발했으므로, dl4j라이브러리를 사용했다.

|  |
| --- |
| MultiLayerConfiguration multiLayerConfiguration = new NeuralNetConfiguration.Builder()  .seed(RandomUtils.nextInt())  .updater(new Sgd(LEARNING\_RATE))  .list()  .layer(new DenseLayer.Builder()  .nIn(20)  .nOut(100)  .activation(Activation.TANH)  .weightInit(WeightInit.XAVIER)  .build())  .layer(new DenseLayer.Builder()  .nIn(100)  .nOut(100)  .activation(Activation.TANH)  .weightInit(WeightInit.XAVIER)  .build())  .layer(new DenseLayer.Builder()  .nIn(100)  .nOut(50)  .activation(Activation.TANH)  .weightInit(WeightInit.XAVIER)  .build())  .layer(new DenseLayer.Builder()  .nIn(50)  .nOut(20)  .activation(Activation.TANH)  .weightInit(WeightInit.XAVIER)  .build())  .layer(new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)  .activation(Activation.SOFTMAX)  .weightInit(WeightInit.XAVIER)  .nIn(20)  .nOut(11).build())  .build(); |

위의 코드를 통해 5개의 레이어를 만들었다. 레이어 신경망은 심층 신경망을 사용했다. 이미지처리나 순차데이터가 아니므로, 가장 보편적으로 사용되는 레이어 신경망을 사용했다.

현재 학습된 인공지능 모델은 다음과 같은 파라미터로 학습되어진 모델이다.

Data size = 1024

batch size = 32

epoch = 32736

모델 튜닝 과정

모델 튜닝을 통해 다음과 같은 내용들이 바뀌었다.

학습률 : 0.0005 -> 0.1 (얼마나 꼼꼼히 문제를 검토할 것인가? 낮을수록 꼼꼼히 본다는 의미)

레이어별 활성화 함수 : ReLU -> Tanh

1) 처음에는 학습률을 0.005 또는 0.0005를 사용했었다.

이 수치는 일반적으로 이미지 데이터를 학습 시킬 때 많이 사용되는 값이다. 그래서 이를 사용하려했었다.

2) 처음에 ReLU를 사용하려 했었다.

마찬가지의 이유로 DCGAN 논문이 ReLU를 사용하여 성공적인 결과를 얻었다고 쓰여있기 때문에 이를 그대로 사용하려했었다.

처음의 목적은 충분히 낮은 숫자로 천천히 오래 학습을 시킬 예정이었으나, 모델이 제대로 학습되지 않았다.

이유를 추적했더니 도중 레이어 별 뉴런들이 내뱉는 값을 보니 모두 하나의 값만 가리키는 현상이 발견됬다. 우리는 이 현상을 보고 ‘학습하던 도중 잠정적으로 ax + b에서 a 값이 0 으로 유도되는 바람에 이런 문제가 발생하고 있구나’ 라는 생각을 했다. 그럼 왜 a 가 0으로 유도 됬는가?

다음과 같은 가설을 세웠다.

1. 학습률이 너무 낮아서 그럴 수 도 있을거라 판단했다. 학습하는 과정중에서 학습률을 곱하는 과정이 나오는데, 학습률이 너무 낮은데다 레이어가 중첩되어있으면 학습률 값이 제곱수로 누적되면서 double 이 표현할 수 있는 유효 숫자 한계를 벗어나게 된다고 판단했다. 그리고 이로 인해 a 가 0 으로 값이 유도 될 수 있겠다.라는 생각이 들었다.

2. ReLU 활성화 함수는 값이 한번 0이하로 떨어지게 되면 경사를 잃어버리게 된다. 즉 그 뉴런은 더 이상 학습이 불가능한 죽은 뉴런이 된다. 그래서 그냥 ReLU를 사용하지 않고 Tanh를 사용하기로 했다.

위 가설을 바탕으로 모델을 튜닝하고 다시 학습시켰더니 원하는 결과를 얻을 수 있었다.