

ICE4016 데이터베이스 설계

<Term Project>

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2023년 12 월 16 일

학부 정보통신공학

학년 4

성명 이재영

학번 12181816



1. 개요

* 아래의 웹 페이지 구성
  + Doctor, nurse의 정보를 관리하는 관리자 페이지
  + Doctor, nurse는 각각 examination treatment를 관리하고 조건에 맞는 patient를 검색할 수 있는 직원 페이지
  + Patient의 예약 조회 및 예약, 예약 확인을 위한 환자 페이지
* 아래 기능을 구현하기 위한 요구사항 추가 후, ER-Diagram 완성하여 제시
* 아래 기능을 구현하기 위한 제약조건, 인덱스, 트랜잭션 등을 제시
  + 관리자 페이지
    - 관리자로 로그인 한 경우 doctor, nurse의 정보 입력/수정/삭제
  + 직원 페이지
    - Doctor로 로그인 한 경우 examination의 정보 입력/수정/삭제
    - Nurse로 로그인 한 경우 treatment의 정보 입력/수정/삭제
    - Doctor 또는 nurse가 로그인 한 경우 조건에 맞는 patient 검색
  + 환자 페이지
    - Patient가 로그인 한 경우 예약하기, 예약 조회, 예약 취소

1. 상세 설계 내용

* 텍스트, 스크린샷, 평행, 폰트이(가) 표시된 사진

  자동 생성된 설명ER-Diagram 제작
  + 텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

    자동 생성된 설명제공되는 필요조건을 고려하여 각 엔티티를 구성하고 필요한 릴레이션을 구성한다.
  + 구성한 엔티티는 manager, doctor, nurse, patient, manager, user, medical\_specialty이다.
    - Manager은 id, name, password를 속성으로 가진다.
    - Doctor는 id, name, address, phone number, password, department id를 속성으로 가진다. Department id는 medical\_specialty의 department id를 참조한다.
    - Nurse는 id, name, address, phone number, password, department id를 속성으로 가진다. Department id는 medical\_specialty의 department id를 참조한다.
    - Patient는 id, name, socialsecuritynumber, gender, address, bloodtype, height, weight, phone number, password, doctorid, nurseid를 속성으로 가진다. Doctor id와 nurse id는 각각 doctor, nurse의 doctor id와 nurse id를 참조한다.
    - User는 id, password, role을 속성으로 가지며, 관리자에 의해 doctor나 nurse의 정보가 추가/삭제 될 경우 user 엔티티에서도 추가/삭제가 되어 로그인이 가능 또는 불가능 해진다.
    - Medical\_specialty는 id, name, phone number를 속성으로 가진다.
  + 구성한 릴레이션은 examination, treatment, reservation이다.
    - Examination은 id, datetime, details, doctorid, patientid를 속성으로 가진다. Doctorid는 doctor 엔티티의 id를 참조하고, patientid는 patient 엔티티의 id를 참조한다.
    - Treatment는 id, datetime, details, nurseid, patientid를 속성을 가진다. Nurseid는 nurse 엔티티의 id를 참조하고 patientid는 patient 엔티티의 id를 참조한다.
    - Reservation은 id, datetime, departmentid, patientid를 속성으로 가진다. Departmentid는 medical\_specialty의 id를 참조하고 patientid는 patient 엔티티의 id를 참조한다.
* Login.js와 login.hbs
  + 텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

    자동 생성된 설명Id, password 입력 시 User 엔티티의 속성 중 role에 따라 각 페이지로 이동한다.
    - Login 시 전달되는 id, password가 User 엔티티 사의 정보와 일치 한지 비교하교 동일한 정보 시에 로그인 여부와 해당 사용자의 role을 저장한다. Role에 따라 관리자, 의사, 간호사, 환자는 각각의 페이지로 이동한다.
* 텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명Manager.js
  + - 텍스트, 스크린샷, 폰트이(가) 표시된 사진

      자동 생성된 설명텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

      자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

      자동 생성된 설명텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

      자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

      자동 생성된 설명role이 manager(관리자)이면, 관리자 페이지로 이동한다. 페이지 하단에는 getDoctor, getNurse 함수로 doctor와 nurse의 정보를 불러온다.
    - 해당 불러온 정보들은 수정 및 삭제가 가능하며, 해당 기능 실행 시 User 엔티티의 정보도 수정 및 삭제가 진행된다. 수정 및 삭제는 doctor의 기본 키인 doctorid 속성에 의해 진행된다. Nurse의 수정 및 삭제 매커니즘은 doctor와 동일하기 때문에 생략한다.
    - 텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

      자동 생성된 설명페이지 상단부에는 doctor와 nurse를 입력할 수 있는 부분이 존재하며, 각 엔티티의 속성에 맞춰 데이터를 입력하고, doctor 또는 nurse 엔티티에 추가한다. 또한 User 엔티티에 추가하여 로그인이 가능해지도록 한다.
* 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

  자동 생성된 설명Doctor.js
  + Doctor가 로그인 하면 해당 doctor의 id를 활용하여 examination 정보와 담당 patient의 정보를 확인할 수 있다. Examination을 수정/삭제/입력이 가능하며 이는 manager의 doctor 또는 nurse 정보의 수정/삭제/입력 메커니즘과 동일하기 때문에 생략한다.
* Nurse.js
  + Doctor가 examination 정보를 다룬다면 nurse는 treatment 정보를 다루기 때문에 보이는 이름만 바뀌었다고 볼 수 있다. 그래서 doctor 페이지의 메커니즘과 동일하며, nurse 역시 담당 patient의 정보를 확인할 수 있다.
* Patient.js
  + 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

    자동 생성된 설명Patient가 로그인 되면, 예약하기, 예약 조회, 예약 취소가 가능하다.
  + 해당 patient의 id를 활용하여 Reservation 릴레이션에서 불러오며 예약 정보를 확인할 수 있다. 해당 예약 정보는 삭제가 가능하다. 예약 삭제 또는 예약하기 기능 은 이전 엔티티들의 삭제/입력 메커니즘과 동일하다.
* Index 최적화
  + 텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명Index를 사용하면 데이터 조회는 빨라지지만, 데이터 삽입, 수정, 삭제 작업이 느려지게 된다. 그렇기 때문에 조회가 많이 일어나는 데이터를 중점적으로 삽입, 수정, 삭제의 정도를 확인해서 적절히 설정해주어야 한다.
  + 기본키 및 외래키는 자동적으로 index 설정이 되어서 해당 키를 복합적으로 묶어 index로 설정해주었다.
* 텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

  자동 생성된 설명텍스트, 폰트, 스크린샷이(가) 표시된 사진

  자동 생성된 설명트랜잭션 설계
  + Insert.sql에서 Start Transaction문으로 트랜잭션을 시작하고, 여러 테이블에 필요한 데이터를 삽입 후에 Commit 구문으로 트랜잭션을 종료했다. 삽입 과정 사이에 에러가 발생하면 모든 변경 사항이 롤백 될 것이며, 성공적으로 완료되면 모든 변경사항이 저장된다. 일단 트랜잭션은 연관된 데이터를 함께 삽입하여 데이터의 일관성을 유지하도록 한다. User 엔티티의 id, password는 doctor, nurse, patient, manager의 id, password와 연관, examination은 doctor, patient와 연관, treatment는 nurse, patient와 연관되어 있기 때문에 결국 거의 대부분의 엔티티와 릴레이션은 연관되어 있다고 판단하였다. 그래서 모든 삽입 문을 하나의 트랜잭션으로 묶어서 활용했다.

1. 실행 화면

* 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

  자동 생성된 설명홈페이지 실행
  + 초기 접속 시에는 로그인 창을 확인할 수 있다.
* 텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

  자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

  자동 생성된 설명관리자 접속
  + 의사, 간호사의 정보 추가하는 부분이 상단에 존재하며, 하단에는 존재하는 의사와 간호사의 정보를 확인할 수 있고 수정 및 삭제가 가능하다.
  + 텍스트, 스크린샷, 라인, 번호이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

    자동 생성된 설명예시로 의사 정보를 위와 같이 추가하면 아래의 의사 정보에 바로 나타나는 것을 확인할 수 있다. 해당 추가된 의사 정보의 id와 password를 로그인 창에 입력하면 해당 의사의 페이지가 나타난다.
* 의사 접속
  + 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

    자동 생성된 설명위 관리자가 추가한 1005의 id를 가진 의사로 접속해본다.
  + 해당 의사는 방금 추가되었기 때문에 examination 및 담당 patient 정보가 아무 것도 나타나지 않는다.
  + 텍스트, 스크린샷, 라인, 폰트이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

    자동 생성된 설명이전에 DB 상에 존재하는 환자 id 10000을 이용해서 위와 같이 검사 정보를 입력하면 검사 정보 목록이 바로 뜨는 것을 확인할 수 있다. 우측의 modify, delete 버튼을 이용해 수정 및 삭제가 가능하다.
  + 기존에 DB상에 존재하는 1000의 id를 가진 doctor로 로그인하고 환자 목록을 보면 위와 같은 것을 확인할 수 있다. Patient 엔티티의 속성으로 doctor id가 존재하기 때문에 해당 doctor id를 가진 patient들을 확인할 수 있다.
* 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

  자동 생성된 설명간호사 접속
  + 간호사 역시 의사 페이지와 유사한 정보를 나타낸다. Doctor는 examination, nurse는 treatment에 대한 입력/수정/삭제가 가능하다. 또한 patient 엔티티의 속성 중에는 nurse id도 존재하기 때문에 해당하는 id를 가진 patient가 존재하면 아래의 환자 목록에서 patient를 확인할 수 있다.
* 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

  자동 생성된 설명환자 접속
  + Patient의 id, password로 접속 시에는 위와 같은 창을 확인할 수 있다. 예약날짜, 진료받기 위한 과, 해당 환자의 id를 이용해여 예약 버튼을 누르면 아래의 예약 조회 표에 추가된다. 예약 ID은 환자 ID + 부서 ID의 형태로 저장된다. 해당 예약을 취소하고 싶을 때는 delete 버튼을 이용해 취소가 가능하다.

1. 결론

* 기존에 진행하던 실습과 과제에서 배운 것을 모두 활용하여 진행해보는 시간이었습니다. 단순히 어느 부분만을 중점적으로 보는 것을 아닌 데이터베이스에서의 엔티티와 릴레이션 설계를 바탕으로 웹 앱 상에서 데이터를 수정, 입력, 삭제에 대한 기능을 구현해볼 수 있었습니다.
* 우선, 구현 못한 기능이 있어서 아쉽습니다. 먼저 의사나 간호사가 로그인 시 해당 사용자는 조건에 맞는 patient를 검색할 수 있는 기능이 있는데, 이는 patient의 이름을 이용해서 LIKE 키워드를 이용해 특정 문자가 포함되면 해당 patient들을 조회할 수 있을 것 같다고 판단했습니다. 하지만 웹 앱 상에서 문제가 발생하는 지 제대로 진행되지 않았습니다.
* 두 번째로는 설계한 엔티티와 릴레이션의 활용을 극대화하여 충분한 기능을 적용해보지 못했다는게 굉장히 아쉽습니다. 생각해본 것으로는 patient가 예약을 하면 해당 예약 내용에는 Department Id가 존재하기 때문에 해당하는 medical\_specialty의 의사들에게 해당 예약 정보를 보이게 했으면 어떨까 등의 아쉬움이 많이 남습니다.
* 세 번째로는 프론트 개발의 부족함입니다. 눈으로 보이는 웹 페이지도 그럴 듯하게 제작해보고 싶었지만, 아직까지 Node.js에 대한 이해가 부족하다고 느껴집니다.
* 마지막으로는 트랜잭션, index 설계에 대한 이해도가 크게 부족하다고 생각합니다. 필요한 엔티티, 릴레이션이나 기능을 만들었다 쳐도, 실제 사용시의 성능과 데이터의 무결성 및 일관성이 보장되어야 하는 것은 매우 중요한 문제입니다. Faker 모듈을 이용하여 대량의 데이터를 이용한 데이터 삽입, 수정, 삭제로 성능 평가를 직접해 보았어야 하는데 그러지 못해 아쉽습니다.