

데이터베이스 요구사항 분석서

과제명 : JAVASWING 기반 교육생 관리 서비스

2020. 06. 08

스치다(스터디에 미치다)

I. 요구사항 정의서

개발 환경

- A. 강의실 컴퓨터 Java EE java version "1.8.0_251"
- B. 개인 노트북 Java SE java version "14.0.1" 2020-04-14
- C. 변수명은 기능_(목적)_원형 을 기본으로 하여 검색 및 용도 파악을 쉽게 한다.
 - 1. 예) Background_imgLabel
 - 2. 예) ID_TextField
 - 3. 예) Login_Button
- D. 추가 라이브러리
 - 1. ojdbc6
 - 2. jfreechart-1.0.19, jcommon-1.0.23
 - 3. Windowbuilder (Java 버전이 높은곳에서 수정한 파일은 낮은곳에서 사용 불가, 같은 클래스 새로 만들어 코드 복붙 실행 필요)

II. 공통

- A. 현재 view에서 다음 view로 넘어갈 때 로그인 시 사용한 아이디를 넘겨준다.
- B. 콘솔창에서 뜨는 확인 메시지는 개발과정에서 신경 쓰지 않고 최대한 남긴다.
- C. 과정은 '빅데이터 분석서비스 개발자과정'
- D. 과목은 'Java' 'R' 'DB'
- E. 모든 과목의 모든 점수 데이터가 DB에 최종적으로 모두 입력된 상태로 진행한다.

III. 로그인 페이지

- A. All_Login.java
- B. 로그인 창에서 ID와 PW를 입력받을 수 있다.
- C. ID를 입력받는 곳에 안내글을 보이게 한다.
- D. ID를 입력받는 곳에 마우스 오버 시 툴팁을 표시한다.
- E. PW 입력창은 특수문자로 보이게 한다.
- F. PW 입력창 위에 마우스 오버 툴팁을 표시한다. (추가 구현 기능 : 체크시 비밀번호 직접 보기)
- G. 우측 상단에 생성된 닫기 버튼을 통해 뷰를 종료할 수 있다.
- H. 로그인 정보가 DB 정보와 매칭되지 않으면 팝업으로 로그인 실패를 알린다**
- I. 로그인 실패 팝업창에 최대 5번의 시도 횟수를 카운팅하여 보여준다.
- J. 로그인 정보 입력이 연속 5번 이상일 경우 경고 메시지와 함께 로그인 창이 닫힌다.
- K. 로그인 정보가 DB와 매칭되면, "성공" 안내 팝업창이 뜨며 메인 페이지로 이동한다.
- L. DB에서 가져온 아이디를 구분하여 학생 메인 페이지와 연구원 메인 페이지를 구분한다.**

IV. 학생 메인 페이지

- A. Student_1Main.java
- B. 학생 아이디로 로그인 시, 아이디를 식별해 학생 메인 페이지로 이동한다.
- C. 초기 비밀번호가 변경되지 않았다면, 비밀번호 변경을 위해 개인정보 페이지로 이동할것인가를 묻는 안내 팝업창이 뜬다.

- D. 로그인 시 입력한 아이디를 식별해 학생이 수강하는 과정의 타이틀을 DB에서 가져온다
- E. 로그인 시 입력한 아이디를 식별해 DB에서 해당 과목 목록을 불러온다.
- F. 로그인 시 입력한 아이디를 식별해 DB에서 학생 이름을 우측 상단에 불러온다.
- G. 수업 과목 버튼을 누르면 해당 과목의 상세 페이지로 이동한다.
- H. 학생 메인 페이지에서 이벤트 목록을 누르면 이벤트 상세 페이지로 이동한다.
(이벤트 페이지에 무엇을 어떻게 보여줄 것인가에 관한 아이디어 자체가 없어 개선 필요)
- I. 학생 메인 페이지에서 장학금 클릭시 이벤트 상세 페이지로 이동한다.
- J. 학생 메인 페이지에서 우측 상단의 개인 정보를 클릭하면 학생 개인정보 페이지로 이동한다.
- K. 우측 상단에 종료 버튼을 통해 프로그램 창을 종료할 수 있다.

V. 개인 정보 페이지

- A. AccountView.java
- B. 학생 로그인 시, 비밀번호가 초기값과 같다면, 비밀번호 변경을 안내하는 팝업창이 뜬다.
- C. 초기 비밀번호는 초기값과 아이디를 조합한 간단한 알고리즘을 사용한다.
- D. 학생 메인 페이지에서 '나의 정보'를 클릭해서 개인 정보 페이지로 접속이 가능하다.
- E. 수정 가능한 개인 정보 목록은, 비밀번호, 성별, 전공, 최종 경력, 자기 소개가 있다.
- F. 비밀번호 변경만으로 저장 가능하며, 기타 개인정보는 선택사항으로 둔다.

VI. 과목 상세 페이지

- A. Student_2Subject.java
- B. 학생 아이디를 식별해 수강하는 과목의 챕터 목록을 가져온다.
- C. 각각의 챕터 제목을 db에서 가져온다
- D. 각각의 챕터 점수를 db에서 가져온다.
- E. 우측 상단 그래프에 가져온 챕터별 점수를 나타낸다.
- F. 우측 하단 그래프에 지난 3회차에 해당하는 점수 db에서 가져와 나타낸다.
- G. 우측 챕터를 클릭하면 해당 챕터의 테스트 페이지로 이동 할 수 있다.

VII. 테스트 페이지

- A. Student_2Test.java
- B. 미구현
- C. 학생은 연구원(관리자)가 업로드한 문제 정보를 바탕으로 챕터별 시험을 볼 수 있다.
- D. 정답은 라디오 및 정답 번호 입력 등으로 구현한다.
- E. 저장과 함께 해당 view 가 종료된다
- F. 저장 버튼을 누르면 창이 닫히기 전에 점수를 팝업창으로 표시한다.
- G. 점수가 과목 상세 페이지의 챕터별 점수에 연동된다.
- H. 시험 결과는 챕터를 클릭하면 다시 표시된다.
- I. 시험 결과는 수정 불가능하게 표시한다..

VIII. 연구원 메인 페이지

- A. Teacher_1Main.java
- B. 미구현

- C. 로그인 창에서 연구원 아이디로 로그인시, 연구원 메인 페이지로 이동한다.
- D. 관리자 페이지에서 학생의 챕터 점수를 입력할 수 있다.
- E. 관리자 페이지에서 학생의 장학 점수를 입력할 수 있다.
- F. 관리자 페이지에서 그래프로 지난 과목 평균점수를 알 수 있다.
- G. 관리자 페이지에서 시험문제를 출제할 수 있다.
- H. 과목명과 챕터명을 선택하면, 과목의 챕터 점수를 입력하여 입력, 수정 할 수 있다.
- I. 문제 입력을 누르면, 챕터별 문제 입력 페이지로 이동한다.
- J. 이벤트명과 학생 이름을 선택하면, 이벤트 점수를 관리자가 입력, 수정할 수 있다.
- K. 우측 그래프에 지난 과정의 성적 데이터를 불러와서 보여준다.
- L. 학생 정보 수정 페이지를 통해 학생 기본 id와 pw 등 모든 정보를 입력 할 수 있다.

2. 객체 정의서

객체 명	속성 명
챗터	챗터코드(PK), 챗터이름, 연구원코드
학생	학생코드(PK), 학생팀코드, 학생이름, 학생pw, 학생점수, 학생전공여부
이벤트	이벤트코드(PK), 이벤트점수, 이벤트이름 연장 유/무
학생챗터	학생챗터코드(PK), 챗터코드(FK), 학생코드(FK), 학생챗터점수
학생이벤트	학생이벤트코드(PK), 이벤트코드(FK), 학생코드(FK), 학생이벤트점수

3. E-R Diagram

