소프트웨어 공학

- report 1 -

MRD, SRS를 이용한 요구 분석

헬스장 관리 프로그램

팀 명 ME

팀원

2011105108 홍진경

2011105059 우병준

2012105076 이현지

- 목차 -

Ⅰ. MRD ------------------------------------------------------------------------------------------------- 팀원 모두

Ⅱ. SRS

1. 로그인화면 -------------------------------------------------------------------------------------- 우병준
2. 회원관리 패널 ---------------------------------------------------------------------------------- 이현지
3. 운동 기구 패널 --------------------------------------------------------------------------------- 우병준
4. 운동프로그램 패널 ---------------------------------------------------------------------------- 홍진경
5. 트레이너 패널 ---------------------------------------------------------------------------------- 이현지
6. FileManager ------------------------------------------------------------------------------------ 홍진경
7. 회원 모드 ---------------------------------------------------------------------------------------- 이현지
8. 로그아웃 ----------------------------------------------------------------------------------------- 이현지
9. DB다이어그램 --------------------------------------------------------------------------------- 홍진경

**MRD**

1. 헬스장 관리 프로그램이 필요하다.
2. 회원정보, 운동 프로그램 정보, 운동 기구 정보, 트레이너 정보는 모두 DB로 관리한다.
3. 권한
   1. 관리자는 트레이너, 회원, 운동기구, 운동프로그램을 모두 열람 및 관리할 수 있다.
   2. 트레이너는 자신에게 배정된 회원, 자신의 정보, 운동 기구, 운동 프로그램을 열람 및 관리 할 수 있다.
   3. 일반회원은 자신의 정보, 배정된 트레이너 정보, 배정된 운동 프로그램을 열람 할 수 있다.
4. 등록 정보
   1. 트레이너 정보는 이름, 휴대폰번호, 입사일, 주소, 월급, 특이사항으로 구성된다.
   2. 회원 정보는 이름, 휴대폰번호, 키, 몸무게, 체지방량, 근육량, 등록 날짜, 등록 기간, 생년월일, 주소, 특이사항으로 구성된다.
   3. 운동 기구 정보는 운동기구명, 구입날짜, 수량, 특이사항으로 구성된다.
   4. 운동 프로그램 정보는 운동 프로그램 이름, 운동 부위, 난이도, 특이사항으로 구성된다.
5. 로그인
   1. 관리자 로그인 - 관리자의 id는 root이다.
   2. 트레이너, 일반회원 로그인 - id로 자신의 휴대폰 번호를 사용한다.
6. 회원 관리 패널
   1. 회원관리 패널에서는 회원을 추가, 삭제 및 검색 할 수 있다.
   2. 회원관리 패널에서는 회원의 이름과 휴대폰 번호, 배정 트레이너, 배정 프로그램을 볼 수 있다.
   3. 회원 칼럼 상세보기 버튼을 누르면 정보를 상세히 볼 수 있고, 수정이 가능하다. 또, 회원에게 트레이너를 부여하고 운동 프로그램을 배정할 수 있다.
   4. 회원은 관리자 혹은 트레이너가 등록 할 수 있다. 회원 이름, 등록 날짜, 등록 기간, 생년월일, 신체정보(키, 몸무게, 체지방, 근육량), 전화번호, 주소, 특이사항을 입력할 수 있고 나이와 만료일자는 자동으로 계산한다.
   5. 회원 수정은 관리자 혹은 트레이너가 할 수 있다. 관리자는 모든 회원 수정이 가능하지만 트레이너는 본인에게 배정된 회원만 수정이 가능하다.
7. 운동 프로그램 패널
   1. 운동 프로그램 패널에서는 프로그램을 추가, 삭제 및 검색을 할 수 있다.
   2. 운동 프로그램 패널에서는 운동 프로그램의 이름, 운동 부위, 난이도, 비고란을 볼 수 있다.
   3. 운동 프로그램 추가 시 관리자는 운동 이름, 운동 부위, 난이도, 비고를 입력할 수 있다.
   4. 운동 프로그램 상세보기 버튼을 누르면 정보를 상세히 볼 수 있고, 수정이 가능하다.
8. 운동기구패널
   1. 운동기구 관리페이지에서는 운동기구 추가, 삭제 및 검색을 할 수 있다.
   2. 운동기구 관리페이지에서 운동기구명, 구입날짜, 수량, 비고란을 볼 수 있다.
   3. 운동기구 추가 시 관리자는 id, 운동기구명, 구입날짜, 가격, 수량, 비고를 입력할 수 있다.
   4. 운동기구 상세보기 클릭 시 그 운동기구에 대한 상세 내용이 나오며 관리자는 수정 버튼을 통해 수정이 가능하고 파손유무를 체크박스를 통해 관리할 수 있다.
9. 트레이너 관리 패널
   1. 관리자는 트레이너 등록, 삭제, 검색 및 트레이너 정보 확인이 가능하다.
   2. 트레이너 등록 시 이름, id, 입사일, 전화번호, 주소를 입력받는다.
   3. 트레이너 이름 검색을 통해 해당 트레이너의 정보만을 확인 할 수 있다.
   4. 트레이너 선택 후 상세정보 버튼을 클릭하면 해당 트레이너의 상세정보 열람이 가능하다. 상세정보 열람 시 트레이너 정보 수정 및 주소와 월급 확인이 가능하다.
10. 회원 전용 패널
    1. 회원은 자신의 상세 정보를 열람할 수 있다.
    2. 회원은 자신에게 배정된 트레이너 정보 확인이 가능하다.
    3. 회원은 자신에게 배정된 운동 프로그램의 정보 확인이 가능하다.
11. Data의 관리에 대한 자세한 과정은 개발 과정 중에서 의뢰자와의 충분한 대화를 나누며 구현해 나가도록 한다.

**SRS**

개발언어는 JAVA를 이용하고 database는 mysql을 사용한다.

**A. 로그인 화면**

**Function** 로그인

**Description**

로그인 아이디를 이용하여 접속이 가능하다. 로그인 정보에 따라 사용자화면, 관리자 혹은 트레이너 화면을 띄운다.

**Inputs** ID

**Source** 사용자 입력

**Outputs** 새로운 화면

**Action**

관리자 ID ----- 관리자용 화면

트레이너 ID ----- 트레이너용 화면

일반사용자 ID ----- 일반사용자용 화면

**Requirements**

사용자의 입력

**Pre-condition**

데이터베이스에 트레이너 테이블과 일반 사용자 테이블이 존재해야함

**B. 회원관리 패널**

**Function** 회원정보 등록

**Description**

회원 정보를 회원관리 목록에 추가한다.

**Inputs** 회원 이름, 등록 날짜, 등록 기간, 생년월일, 신체정보(키, 몸무게, 체지방, 근육량), 전화번호, 주소, 특이사항

**Source** 사용자로부터 입력받는 정보

**Outputs** client 객체

**Destination**

데이터베이스의 Client table

**Action**

등록버튼을 클릭후, 사용자의 입력을 받게되면 client객체가 생성되고 생성된 client객체는

FileManager로 보내지게 된다. 이후 FileManager에서 객체의 정보를 검토한 후 이상이

없을 시 데이터베이스에 저장을 한다. 이상이 있을시 사용자에게 실패를 알리는 메세지를

보낸 후 객체를 저장하지 않는다.

**Requirements**

사용자의 다이얼 입력

**Pre-condition**

데이터베이스의 client테이블이 필요.

**Post-condition**

client테이블의 Column이 추가되고 회원관리 목록에 추가.

**Function**  회원정보 상세보기

**Description**

선택된 회원의 정보를 상세히 볼 수 있다.

**Inputs** 선택된 회원의 ID, table명

**Source** 해당 객체의 ID

**Outputs** 데이터베이스로부터 읽은 회원의 상세정보

**Destination** 회원관리목록

**Action**

사용자(관리자)가 상세보기를 원하는 회원정보를 선택후 상세보기를 클릭,

그러면 FileManager에게 선택된 회원의 ID정보와 저장되어있는 table명이

전달되고 FileManager는 전달받은 정보로 데이터베이스에 데이터를 요청한다.

그 후 데이터베이스로부터 읽은 데이터를 화면에 출력한다.

**Requirements**

데이터베이스에 저장된 회원 정보

**Pre-condition**

데이터베이스에 client테이블이 필요

**Function** 회원정보 삭제

**Description**

회원정보를 회원관리 목록에서 삭제한다.

**inputs** 삭제할 회원의 ID, table명

**Source** 객체의 ID

**Outputs** 회원정보 삭제.

**Destination**

데이터베이스의 table

**Action**

관리자가 삭제하고자 하는 회원을 선택후 삭제버튼을 클릭힌다.

그러면 FileManager에게 선택된 회원의 ID정보와 table명이 전달되고

FileManager는 전달받은 정보를 통해 데이터베이스에 저장되어있는

회원 정보를 삭제한다.

**Requirements**

데이터베이스에 저장된 회원정보

**Pre-condition**

데이터베이스의 client 테이블 필요

**Post-condition**

client 테이블에서 저장된 회원정보 삭제

**Function**  회원정보 검색

**Description**

회원관리목록에서 회원의 이름을 이용하여 회원정보를 검색할 수 있다.

**inputs** 검색할 회원의 이름 정보.

**Source** 사용자로부터 입력되는 회원 이름

**Outputs** 동일한 이름의 회원정보

**Destination**

회원 목록

**Action**

검색할 회원의 이름을 입력한뒤 검색을 클릭한다. 그러면 입력된 회원정보와 검색범위의 table명이 FileManager에게 전달되고 FileManager은 전달받은 테이블명과 이름을 통해

데이터베이스 내에서 정보를 읽어와 동일한 이름의 회원을 검색한다. 그 후 일치하는 회원이 있으면 회원목록에 출력하고 동일한 이름이 없는 경우 사용자에게 정보가 없음을 알리는 메세지를 보낸다.

**Requirements**

검색할 회원의 이름 정보 입력

**Pre-condition**

데이터베이스의 client테이블이 필요

**Function**  프로그램 배정

**Description**

등록된 회원중 선택한 회원에게 등록된 프로그램을 배정한다.

**inputs** 선택된 회원의 ID정보와 table명, 프로그램 이름

**Source** 사용자로부터 지정되는 회원 및 프로그램

**Outputs** 프로그램이 배정된 회원정보

**Destination**

데이터베이스의 client table

**Action**

프로그램 배정 클릭시, 프로그램명을 선택하는 창이 나타나고 프로그램을 선택한후 배정을

클릭하게 되면 프로그램명과 선택된 회원의 ID정보 및 table명이 FileManager에게 전달된다.

그러면 FileManager는 전달받은 정보를 통해 데이터베이스에 회원정보를 업데이트한다.

**Requirements**

사용자가 선택한 프로그램명

**Pre-condition**

데이터베이스의 client 테이블이 필요

**Post-condition**

데이터베이스내 client 의 정보가 수정됨

**Function** 트레이너 배정

**Description**

등록한 회원중 선택한 회원에게 등록된 트레이너를 배정한다.

**inputs**

선택된 회원의 ID정보와 table명, 트레이너 이름

**Source**

사용자로부터 지정되는 회원 및 트레이너

**Output**

트레이너가 배정된 회원정보

**Destination**

데이터베이스의 client table

**Action**

트레이너 배정 클릭시, 트레이너 명을 선택하는 창이 나타나고 트레이너를 선택한후 배정을

클릭하게 되면 트레이너명과 선택된 회원의 ID정보 및 table명이 FileManager에게 전달된다.

그러면 FileManager는 전달받은 정보를 통해 데이터베이스에 회원정보를 업데이트한다.

**Requirements**

사용자가 선택한 트레이너명

**Pre-condition**

데이터베이스의 client테이블이 필요

**Post-condition**

데이터베이스내 client의 정보가 수정됨

**Function**  연장하기

**Description**

회원의 등록기간을 연장해준다.

**inputs**

선택된 회원의 ID정보와 table명, 연장일수

**Source**

사용자로부터 입력되는 연장일수

**Output**

등록기간이 수정된 회원정보

**Destination**

데이터베이스의 table

**Action**

연장을 원하는 회원을 선택한 후 연장하기버튼을 클릭한다. 그 후 연장일수를 입력받는다 그러면 선택된 회원 ID정보와 table명, 연장일수가 전달되고 FileManager는 받은 연장일수를 통해 만료일을 새롭게 계산하여 회원 정보를 수정하고 데이터베이스에 업데이트 한다.

**Requirements**

사용자의 입력

**Pre-condition**

데이터베이스의 client테이블이 필요

**Post-condition**

만료일이 수정된 회원 정보

**C. 운동기구 패널**

**Function** 운동기구 추가

**Description**

운동기구를 운동기구 관리 목록에 추가를 한다.

**Inputs** 운동기구명, 수량, 가격, 구입날짜, 비고

**Source** 사용자로부터 입력받는 정보(운동기구명, 수량, 가격, 구입날짜, 비고)

**Outputs** Commodity 객체

**Destination**

데이터베이스의 Commodity table

**Action**

추가버튼을 클릭후 사용자의 입력을 받게되면 Commodity객체가 생성되고 생성된 Commodity객체는 FileManager로 보내지게 된다. 이후 FileManager에서 객체의 정보를 검토한 후 이상이 없을시 DataBase에 저장을 한다. 이상이 있을시 사용자에게 실패를 알리는

메세지를 보낸후 객체를 저장하지 않는다.

**Requirements**

사용자의 다이얼 입력

**Pre-condition**

데이터베이스의 Commodity테이블이 필요.

**Post-condition**

Commodity테이블의 Column이 추가되고 운동기구 목록에 추가.

**Function** 운동기구 삭제

**Description**

운동기구를 운동기구 관리 목록에서 삭제한다.

**Inputs** 삭제할 운동기구의 ID정보와 table명.

**Source** 객체의 ID

**Outputs** 데이터의 삭제

**Destination**

데이터베이스의table

**Action**

사용자가 삭제하고자 하는 운동기구를 선택후 삭제버튼을 클릭하게 되면

FileManager에게 선택된 운동기구의 ID정보와 저장되어있는 table명이

전달되고 FileManager는 ID정보와 table명을 받아 데이터베이스에 저장되어있는

정보를 삭제한다.

**Requirements**

데이터베이스에 저장된 운동기구 데이터

**Pre-condition**

데이터베이스의 Commodity테이블이 필요.

**Post-condition**

Commodity테이블에서 저장된 정보 삭제.

**Function** 운동기구 검색

**Description**

운동기구관리목록에서 운동기구의 이름을 이용하여 운동기구정보를 검색할 수 있다

**Input**

검색할 운동기구의 운동기구명

**Source**

사용자로부터 입력되는 운동기구명

**Output**

동일한 이름의 운동기구 정보

**Destination**

운동기구관리목록

**Action**

사용자가 검색하고자 하는 운동기구의 이름을 입력하면 입력된 이름과 검색범위의 table명이

FileManager에게 전달되고 FileManager는 전달받은 정보를 이용하여 데이터베이스에 저장되어있는 정보들 중 동일한 이름의 운동기구를 검색한다. 동일한 이름이 존재하면 운동기구목록에 출력하고 존재하지 않으면 사용자에게 존재하지 않음을 알린다.

**Requirements**

검색할 운동기구의 이름 정보 입력

**Pre-condition**

데이터베이스의 commodity테이블이 필요

**Function**  운동기구 상세보기

**Description**

운동기구 관리목록에 있는 운동기구의 상세정보를 볼 수 있다.

**Input**

선택된 운동기구의 ID정보와 table명.

**Source**

해당 객체의 ID

**Output**

데이터베이스로부터 읽은 운동기구의 정보

**Destination**

운동기구 목록

**Action**

사용자가 상세보기를 원하는 운동기구를 선택후 상세보기를 클릭.

그러면 FileManager에게 선택된 운동기구의 ID정보와 저장되어있는 table명이

전달되고 FileManager는 전달받은 정보로 데이터베이스에 데이터를 요청한다.

그 후 데이터베이스로부터 읽은 데이터를 화면에 출력한다.

**Requirements**

데이터베이스에 저장된 운동기구 데이터

**Pre-condition**

데이터베이스의 Commodity테이블이 필요.

**Function** 운동기구 수정

**Description**

운동기구 관리목록에 있는 운동기구의 상세정보를 수정한다.

**Inputs** 선택된 운동기구의 상세정보.

**Source** 데이터베이스로부터 읽은 운동기구의 상세정보

**Outputs** 수정된 운동기구의 상세정보

**Destination**

데이터베이스의table

**Action**

수정버튼 클릭시, 상세보기창의 텍스트박스들이 활성화 되어 수정이 가능한 상태가 된다.

이후 사용자가 입력을 통해 정보들을 수정하고 수정이 완료되면 저장버튼을 통해

수정된 데이터를 FileManager에게 전달후 FileManager는 데이터베이스에 수정한데이터를

업데이트 한다.

**Requirements**

데이터베이스에 저장된 운동기구 데이터

**Pre-condition**

데이터베이스의 Commodity테이블이 필요.

**Post-condition**

데이터베이스의 Commodity테이블의 해당 튜플이 수정.

**Function** 파손된 운동기구만 보기

**Description**

운동기구관리 목록중에서 파손유무에 체크가 되어있는

운동기구만 모아서 볼수있다.

**Inputs** 운동기구관리목록에 있는 운동기구정보

**Source** 데이터베이스에 있는 운동기구정보

**Outputs** 파손유무에 체크가 되어있는 운동기구 정보

**Destination** 운동기구관리목록

**Action**

파손기구정렬 버튼 클릭시, 파손유무 체크박스가 체크되어있는지 확인하는 메소드가

실행되며 목록이 뜨는 화면에 파손유무 체크박스에 체크가 된 운동기구들만 나타나게된다.

**Requirements**

데이터베이스로부터 읽은 운동기구정보

**Pre-condition**

운동기구관리전체목록

**Post-condition**

운동기구관리목록중 파손유무에 체크된 목록

**D. 운동프로그램 패널**

**Function** 운동 프로그램 열람

**Description**

데이터 베이스에 있는 운동 프로그램을 열람하고 관리 할 수 있다.

**Destination** 운동 프로그램 목록

**Function** 운동 프로그램 추가

**Description**

운동 프로그램 패널에서 추가하기 버튼을 누르면 다이얼로그가 생성되고, 다이얼로그에서 정보를 입력하여 데이터 베이스에 운동 프로그램을 추가한다.

**Inputs** 현재 날짜와 시간(id), 이름(name), 운동 부위(partOfBody), 난이도(Difficulty), 설명(comment).

**Source** 날짜와 시간 - 시스템, 이름, 운동 부위, 난이도 - 다이얼로그에 입력한 내용

**Outputs** Program객체

**Destination** 데이터 베이스의 program 테이블, 운동 프로그램 목록

**Action**

입력 받은 정보를 바탕으로 Program객체를 생성하고 생성한 객체를 FileManager에 포함된 addProgram()에게 전달하고 데이터 베이스에 입력한다. 성공과 실패 시 각각 메시지를 띄워준다.

**Requirements**

사용자의 입력

**Pre-condition**

데이터 베이스에 program(id, name, partofbody, difficulty, comment) 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 새로운 튜플이 추가된다.

**Function** 운동 프로그램 삭제

**Description**

운동 프로그램 목록에서 선택한 운동 프로그램을 삭제한다.

**Inputs** 해당 운동프로그램의 아이디(id), 테이블(program) 이름

**Source** 해당 객체의 아이디

**Outputs** 데이터의 삭제

**Destination** 데이터 베이스의 program 테이블, 운동 프로그램 목록

**Action**

사용자가 삭제하고 싶은 칼럼을 클릭하고 삭제버튼을 클릭하면 해당 칼럼의 아이디와 테이블 이름(program)이 FileManager의 delete()로 전달되어 삭제된다.

**Requirements**

저장된 운동 프로그램 데이터

**Pre-condition**

데이터 베이스에 program(id, name, partofbody, difficulty, comment) 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 해당 튜플이 삭제된다.

**Function** 운동 프로그램 상세보기

**Description**

운동 프로그램 목록에서 선택한 운동 프로그램을 상세히 본다.

**Destination** 상세보기 다이얼로그

**Inputs** 해당 운동프로그램의 아이디(id), 테이블(program) 이름

**Source** 해당 객체의 아이디

**Action**

사용자가 상세히 보고 싶은 칼럼을 클릭하고 상세 보기 버튼을 클릭하면 해당 칼럼의 아이디와 테이블 이름(program)이 FileManager의 getProgram()로 전달되고 데이터 베이스에서 받아온 정보를 다시 운동 프로그램 상세보기 다이얼로그로 전달한다..

**Requirements**

저장된 운동 프로그램 데이터

**Pre-condition**

데이터 베이스에 program(id, name, partofbody, difficulty, comment) 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 해당 튜플이 삭제된다.

**Function** 운동 프로그램 검색

**Description**

운동 프로그램 패널에서 운동 프로그램의 이름을 검색하면 해당 운동기구를 찾을 수 있다.

**Inputs** 이름(name)

**Source** 사용자의 입력

**Outputs** Program객체

**Destination** 데이터 베이스의 program 테이블, 운동 프로그램 목록

**Action**

name is null ------ 모든 튜플을 검색하고 반환

name is not null ------ 해당 이름의 튜플을 검색함

**Requirements**

사용자의 입력

**Pre-condition**

데이터 베이스에 program(id, name, partofbody, difficulty, comment) 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 새로운 튜플이 추가된다.

**Function** 운동 프로그램 수정

**Description**

운동 프로그램 패널에서 추가하기 버튼을 누르면 다이얼로그가 생성되고, 다이얼로그에서 정보를 입력하여 데이터 베이스에 운동 프로그램을 추가한다.

**Inputs** 현재 날짜와 시간(id), 이름(name), 운동 부위(partOfBody), 난이도(Difficulty), 설명(comment).

**Source** 날짜와 시간 - 시스템, 이름, 운동 부위, 난이도 - 다이얼로그에 입력한 내용

**Outputs** Program객체

**Destination** 데이터 베이스의 program 테이블, 운동 프로그램 목록

**Action**

입력 받은 정보를 FileManager에 getProgram()으로 넘겨 결과값을 받는다.

**Requirements**

사용자의 입력

**Pre-condition**

데이터 베이스에 program(id, name, partofbody, difficulty, comment) 테이블이 존재해야 한다.

**E. Trainer 패널**

**function** 트레이너 열람

**description**

데이터 베이스에 있는 트레이너의 이름, 전화번호, 입사일, 비고를 열람할 수 있다.

**source** 트레이너 데이터

**destination** 트레이너 목록

**action**

다수의 트레이너가 있을 시 스크롤로 이동하며 열람.

**functio** 트레이너 등록

**description**

트레이너의 정보를 입력받아 트레이너를 등록한다.

**inputs** 트레이너 이름, id, 입사일, 전화번호, 주소, 월급, 비고

**source** 사용자가 입력한 정보

**outputs** Trainer 객체

**destination** 트레이너 목록, 데이터베이스 Trainer 테이블

**action**

inputs를 입력받아 등록버튼을 누르면 outputs 생성. outputs이 데이터베이스의 Trainer 테이블 tuple 형식과 일치하면 데이터베이스에 값 저장. 월급은 초기 값 100만원으로 자동 생성. tuple 형식과 일치하지 않으면 메시지를 띄어주고 저장되지 않음.

**requirements** 사용자의 입력

**pre-condition**

데이터베이스에 Trainer 테이블 존재. id, name, regi\_date, salary, addr, phone column 존재.

**post-condition**

데이터베이스 Trainer 테이블에 입력받은 tuple이 생성

**function** 트레이너 삭제

**description**

선택된 트레이너를 목록에서 삭제한다.

**inputs** 해당 트레이너의 id, 테이블 이름

**source** 해당 객체의 id

**outputs** 데이터의 삭제

**destination** 트레이너 목록, 데이터베이스 Trainer 테이블

**action**

트레이너 목록에서 원하는 트레이너를 선택 후 삭제버튼을 누르면 삭제 확인 메시지 창 팝업. 취소 버튼을 누르면 원래 목록으로 복귀, 확인 버튼을 누르면 해당 트레이너의 id를 DB에 전달하여 해당 tuple 삭제.

**requirements** 저장된 트레이너 데이터

**pre-condition**

데이터베이스에 Trainer 테이블 존재.

**post-condition**

데이터 베이스에 해당 튜플 삭제.

**function** 트레이너 검색

**description**

원하는 트레이너를 검색한다.

**inputs** 트레이너 이름

**source** 사용자의 입력

**outputs** Trainer 객체

**destination** 트레이너 목록

**action**

검색란에 원하는 트레이너의 이름을 입력 후 검색 버튼을 누르면 해당 트레이너만 목록에서 확인 가능. 해당 트레이너가 존재하지 않으면 메시지 팝업.

**requirements** 사용자의 입력

**pre-condition**

데이터베이스에 Trainer 테이블 존재.

**post-condition**

트레이너 목록에 해당 트레이너만 나타남.

**function** 트레이너 상세보기

**description**

선택된 트레이너의 상세정보를 본다.

**inputs** 해당 트레이너의 id, 테이블 이름

**source** 해당 객체의 id

**destination** 상세보기 창

**action**

트레이너 목록에서 원하는 트레이너를 선택 후 상세보기버튼을 누르면 해당 트레이너의 id를 통해 DB에서 받아온 트레이너 정보를 상세보기 창을 통해 보여줌. 이 창에서 주소, 월급 확인 가능

**requirements** 저장된 트레이너 데이터

**pre-condition**

데이터베이스에 Trainer 테이블 존재.

**function** 트레이너 정보 수정

**description**

선택된 트레이너의 정보를 수정한다.

**inputs** 해당 트레이너의 id, 테이블 이름, 전화번호, 주소

**source** 해당 객체의 id, 사용자가 수정하여 입력한 내용

**outputs** 수정된 트레이너 객체

**destination** 상세보기 창, 트레이너 목록, 데이터베이스 Trainer 테이블

**action**

트레이너 목록에서 원하는 트레이너를 선택 후 상세보기버튼을 눌러 상세보기 창 띄움. 수정 내용을 입력 후 수정 버튼을 누르면 수정된 내용 저장.

**requirements** 사용자의 입력

**pre-condition**

데이터베이스에 Trainer 테이블 존재.

**post-condition**

데이터베이스 Trainer 테이블에 수정된 tuple 생성

**F. FileManager**

**Function** getTable(String id) [getProgram, getClient, …]

**Description**

각 테이블로 부터 튜플을 받아 온다.

**Inputs** Id

**Outputs** 각 테이블에 해당하는 객체(Client, Trainer, Program, Commodity)

**Destination** 각 개체의 목록

**Action**

if (id.equals(“all”)) ------ 테이블의 모든 튜플을 불러온다

else ------ 해당 id의 튜플을 불러온다.

**Requirements**

사용자의 입력

**Pre-condition**

데이터 베이스에 해당 테이블들이 존재해야한다.

**Function** addTable(String id) [addProgram, getClient, …]

**Description**

각 패널에서 추가하기 다이얼로그를 통해 각 테이블에 새로운 튜플을 추가할 수 있다.

**Inputs**

addCommodity - id, 운동기구명, 수량, 가격, 구입날짜, 비고

addProgram - 현재 날짜와 시간(id), 이름(name), 운동 부위(partOfBody), 난이도(Difficulty), 설명(comment).

addClient - 이름, 휴대폰번호, 키, 몸무게, 체지방 량

addTrainer - 트레이너 이름, id, 입사일, 전화번호, 주소

**Source** 각 해당하는 객체

**Destination** Database

**Action**

각각의 추가 다이얼로그에서 입력받은 내용을 객체로 만들어 진 내용을 받아 데이터베이스에 저장한다.

**Pre-condition**

데이터 베이스에 각각의 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 새로운 튜플이 추가된다.

**Function** delete(String id, String tablename)

**Description**

삭제하고 싶은 튜플의 아이디를 받아와 해당 테이블에서 삭제한다.

**Inputs** id, tablename

**Source** 각각의 패널

**Destination** 데이터 베이스의 테이블

**Action**

해당 테이블이름과 삭제하고 싶은 튜플의 아이디를 이용하여 선택된 튜플을 삭제한다.

**Requirements**

사용자의 입력

**Pre-condition**

데이터 베이스에 해당 테이블과 튜플이 존재해야 한다.

**Post-condition**

데이터 베이스에 해당 튜플이 삭제된다.

**Function** updateTable(객체) [updateProgram, updateClient, …]

**Description**

각 패널에서 상세보기 다이얼로그를 통해 각 튜플의 내용을 수정할 수 있다. 이 때 수정할 내용을 담을 객체를 updateTable에서 받아 내용을 갱신한다.

**Inputs**

addCommodity - id, 운동기구명, 수량, 가격, 구입날짜, 비고

addProgram - 현재 날짜와 시간(id), 이름(name), 운동 부위(partOfBody), 난이도(Difficulty), 설명(comment).

addClient - 이름, 휴대폰번호, 키, 몸무게, 체지방 량

addTrainer - 트레이너 이름, id, 입사일, 전화번호, 주소

**Source** 각 해당하는 객체

**Destination** Database

**Action**

각각의 추가 다이얼로그에서 입력받은 내용을 객체로 만들어 진 내용을 받아 해당 튜플을 수정한다.

**Pre-condition**

데이터 베이스에 각각의 테이블이 존재해야 한다.

**Post-condition**

데이터 베이스에 해당 튜플이 수정된다.

**G. 회원모드**

**function** 운동 프로그램 열람

**description**

회원은 데이터베이스에 있는 운동 프로그램의 이름, 운동 부위, 난이도, 설명을 열람할 수 있다.

**source** 운동 프로그램 데이터

**destination** 운동 프로그램 목록 패널

**action**

다수의 운동 프로그램이 있을 시 스크롤로 이동하며 열람.

**requirements** 저장된 운동 프로그램 데이터

**function** 개인정보 확인

**description**

회원은 본인의 등록일, 만료일, 배정 트레이너 및 프로그램을 확인할 수 있다.

**source** 회원정보 데이터

**destination** 개인정보 패널

**requirements** 저장된 회원 데이터

**H. 로그아웃**

**function** 로그아웃

**description** 관리자와 회원은 접속을 종료하고 로그인 화면으로 돌아간다.

**inputs** 클릭

**destination**

로그인 화면

**action**

로그아웃 버튼을 클릭하면 접속이 종료되고 로그인 화면으로 돌아감.

**pre-condition**

접속 중.

**post-condition**

접속 종료.

**I. DB 다이어그램**

