

데이터베이스 프로그래밍 명함 관리 시스템

최민석 남태우

2020 - 11 - 23

목차



1. 팀 소개 및 협업 과정, 팀 규칙

2. 프로젝트 개요

3. 데이터베이스 설계

4. 프로그램 설명

5. 시연 영상

팀 소개



3학년16학번 최민석
Front-end



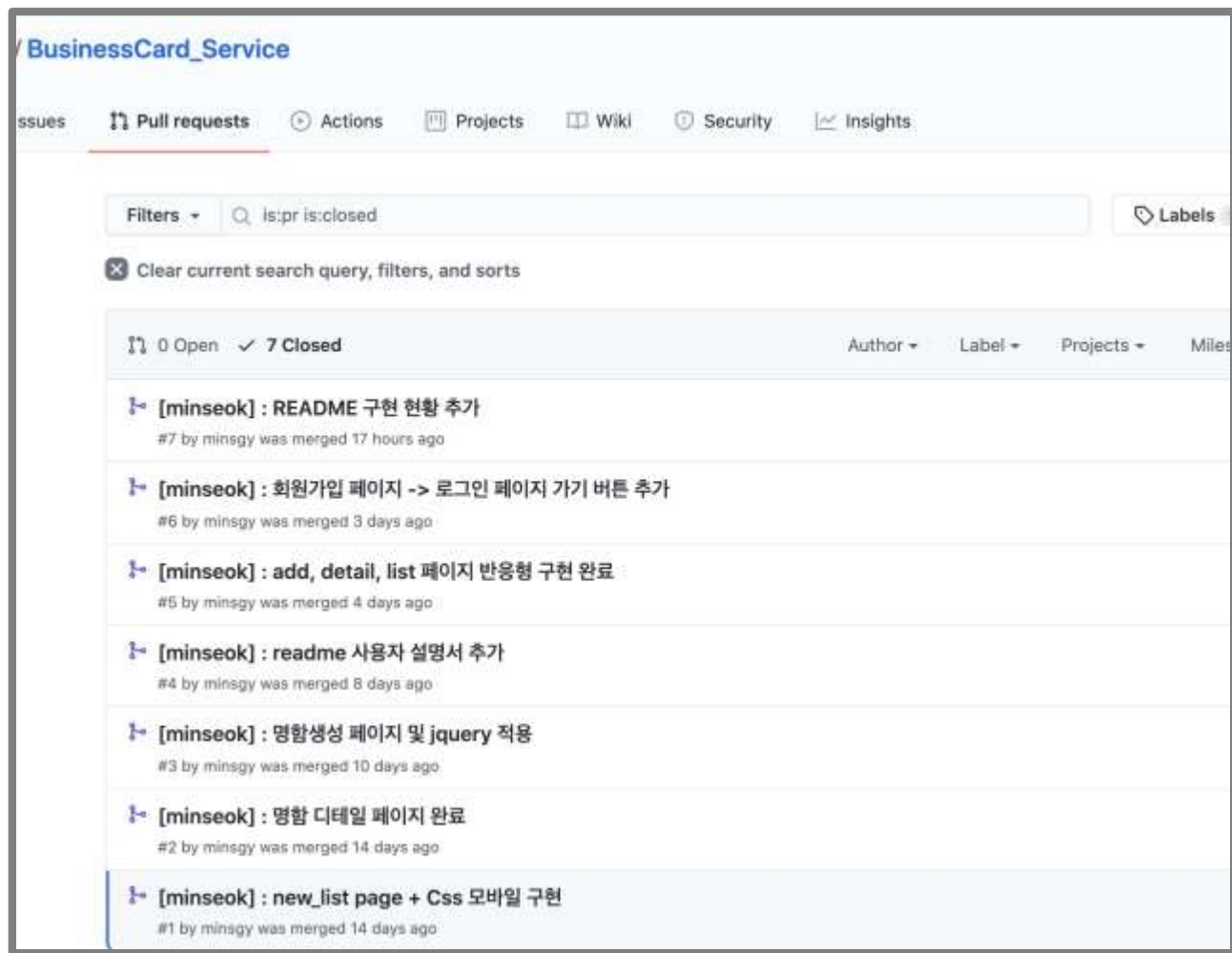
3학년16학번 남태우
Back-end

협업 과정



Github

Pull Request 방식을 사용하여
온라인 상황에서도 규칙에 따라 협업.



팀 규칙

1. 서로에 대한 피드백 확실히 전달 하기.
2. 서로의 개발 코드 설명 해주기
3. 개발 툴 통일 하기.
 - Eclipse

프로젝트 개요

주제 선정 배경



당신들은 순천향 정보 시스템 응용 S/W 개발1팀 이다.
당신들의 고객인 홍길동교수는 명함관리를 편하게 하기 위해
웹기반의 '명함관리 S/W 시스템'을 원한다.
홍길동교수가 원하는 '명함관리 S/W 시스템'을 납품하시오.

납품 목록

- 실행가능한 웹기반의 '명함관리 S/W 시스템'
- 사용자 설명서가 포함된 결과 보고서

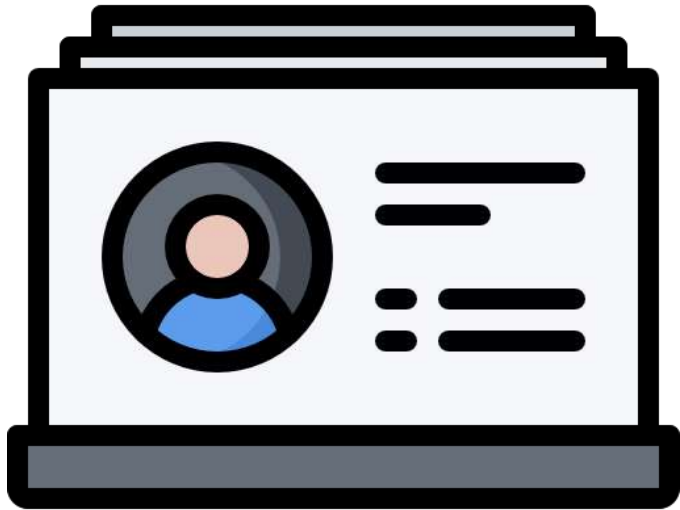
프로젝트 개요

프로그램의 목적

- 명함 관리 시스템을 구현하여 사용자에게 제공함.
- 사용자가 인터넷이 연결 된 “어디서든 사용 할 수 있는 환경 제공”
- 사용자가 “휴대폰을 이용하여 사용 할 수 있는 환경 제공”

프로젝트 개요

프로그램 기능 명세서



명함 관리 S/W

기본 기능 요구 사항

- 명함 생성 기능(CREATE) : 명함 생성 페이지
- 명함 수정 기능(UPDATE) : 명함 상세 수정 페이지
- 명함 삭제 기능(DELETE) : 명함 삭제 버튼
- 명함 읽기 기능(READ) : 명함 리스트 출력 및 상세 페이지

추가 한 요구 사항

- 명함 이름 검색 기능 : 검색bar 추가
- 모바일 / 웹 반응형 구현

데이터베이스 설계

사용자 요구사항

1. 명함에는 개인 정보가 적혀 있다.
2. 명함에는 회사 정보가 적혀 있다.
3. 명함의 정보가 저장될 때, 저장 일자가 관리된다.
4. 회사 정보가 있을 때, 회사명은 반드시 관리되어야 한다.
5. 회사 정보에는 회사 주소, 회사 팩스 번호, 회사전화번호가 있다.
6. 회사 팩스 번호는 없을 수도 있다.
7. 회사전화번호는 없을 수도 있고 여러 개일 수도 있다.
8. 개인 정보에는 성명, 부서, 직급, 휴대전화번호, 이메일 주소가 있다.
9. 명함에는 반드시 성명과 휴대전화번호가 있어야 한다.

데이터베이스 설계

개념적 설계

1. 명함에는 **개인 정보**가 적혀 있다.
2. 명함에는 **회사 정보**가 적혀 있다.
3. 명함의 정보가 저장될 때, **저장 일자**가 관리된다.
4. 회사 정보가 있을 때, **회사명**은 반드시 관리되어야 한다.
5. 회사 정보에는 **회사 주소, 회사 팩스 번호, 회사전화번호**가 있다.
6. 회사 팩스 번호는 없을 수도 있다.
7. 회사전화번호는 없을 수도 있고 여러 개일 수도 있다.
8. 개인 정보에는 **성명, 부서, 직급, 휴대전화번호, 이메일 주소**가 있다.
9. 명함에는 반드시 성명과 휴대전화번호가 있어야 한다.



“명사 도출”

데이터베이스 설계

엔티티 도출 및 속성 정의

엔티티	엔티티 설명	관련 속성	비고
사용자	사용자에 대한 정보	사용자 번호	
		이름	
		아이디	
		비밀번호	
명함	명함의 개인에 대한 정보	개인 고유번호	
		성명	
		부서	
		직급	
		휴대 전화번호	
		이메일 주소	
		저장 일자	
회사 정보	명함의 회사에 대한 정보	회사 번호	
		회사명	
		회사 주소	
		회사 팩스 번호	
		회사 전화번호	다중 값

데이터베이스 설계

엔티티 도출 및 속성 정의

사용자
◆ 사용자 번호
이름
아이디
비밀번호

명함
◆ 개인 고유번호
성명
부서
직급
휴대전화번호
이메일
저장일자

회사 정보
◆ 회사 번호
회사명
회사 주소
회사 팩스 번호
회사 전화번호

- 사용자 요구사항에는 존재하지 않는 **고유 번호**를 ‘주민등록번호’, ‘ID’ 노출을 피하기 위해, 지정했습니다.

데이터베이스 설계

엔티티 간 관계성 정의



1(사용자) : N(명함)

- 각 사용자는 명함을 등록 할 수 있다.
- 각 사용자는 여러 개의 명함을 등록 할 수 있다.

데이터베이스 설계

엔티티 간 관계성 정의



1(명함) : 1(회사정보)

- 명함에는 회사 정보가 관리 된다.
- 명함에는 하나의 회사 정보가 적혀 있다.

데이터베이스 설계

최종 ERD



데이터베이스

논리적 설계

- 단순 속성

릴레이션이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
사용자	사용자 번호	NOT NULL	PK				
	이름	NOT NULL					
	아이디	NOT NULL					
	비밀번호	NOT NULL					
명함	개인 고유번호	NOT NULL	PK				
	성명	NOT NULL					
	부서						
	직급						
	휴대전화번호	NOT NULL					
	이메일						
	저장 일자	NOT NULL					
회사 정보	회사 번호	NOT NULL	PK				
	회사명	NOT NULL					
	회사 주소						
	회사 팩스 번호						
	회사 전화번호						다중 값

데이터베이스 설계

관계 변환

1:N 관계,

- ERD에서 일측 엔티티에 대응하는 릴레이션의 기본 키를
다측 엔티티에 대응하는 릴레이션의 속성으로 복사한 다음에 이 속성을 **외래 키**로 지정.

1:1 관계,

- 어느 곳에 외래 키를 지정하든 상관이 없지만, 자주 사용되나 안되나 등을 따져서 지정.
- 명함과 회사정보에서 **회사정보 릴레이션에 외래키로 지정합니다.**

데이터베이스

논리적 설계

- 단순 속성

릴레이션이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
사용자	사용자 번호	NOT NULL	PK				
	이름	NOT NULL					
	아이디	NOT NULL					
	비밀번호	NOT NULL					
명함	개인 고유번호	NOT NULL	PK				
	성명	NOT NULL					
	부서						
	직급						
	휴대전화번호	NOT NULL					
	이메일						
	저장 일자	NOT NULL					
	사용자 번호	NOT NULL		FK	사용자	사용자번호	
회사 정보	회사 번호	NOT NULL	PK				
	회사명	NOT NULL					
	회사 주소						
	회사 팩스 번호						
	회사 전화번호						다중 값
	개인 고유 번호	NOT NULL		FK	명함	개인 고유번호	

데이터베이스 설계

정규화 - 제1 정규화

제 1정규화는 엔티티에서 하나의 속성이 복수 개의 값을 갖도록 설계 되어 있을 경우,
하나의 속성이 단일 값(Atomic value)를 갖도록 설계하는 과정

릴레이션 이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
회사 정보	회사 번호	NOT NULL	PK				
	회사명	NOT NULL					
	회사 주소						
	회사 팩스 번호						
	회사 전화번호						다중 값
	개인 고유 번호	NOT NULL		FK	명함	개인 고유번호	

- 회사정보 릴레이션에서 회사 전화번호는 다중 값입니다.
- 제 1정규화를 만족하기 위해, 원자 값이 되어야 합니다.
- 릴레이션 분리를 통해 해소합니다.

데이터베이스 설계

정규화 - 제1 정규화

릴레이션 이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
회사 정보	회사 번호	NOT NULL	PK				
	회사명	NOT NULL					
	회사 주소						
	회사 팩스 번호						
	개인 고유 번호	NOT NULL		FK	명함	개인 고유번호	
릴레이션 이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
회사 전화번호	회사 전화번호	NOT NULL	PK				
	회사 번호	NOT NULL	PK	FK	회사 정보	회사 번호	

- 릴레이션 분리 시, **회사 정보의 기본 키(회사번호)**를 회사 전화번호 릴레이션의 **외래키**로 가져오게 됩니다.
- 회사 번호로만 회사 전화번호 릴레이션에서 식별을 할 수 없으므로, **회사 전화 번호와 같이 기본키로 설정합니다.**

데이터베이스 설계

정규화 - 제 2 정규화

제 2 정규화는 주 식별자가 아닌 속성들 중에서 주 식별자 전체가 아닌 일부 속성에 종속된 속성을 찾아 제거하는 과정 (부분 함수 종속 제거)

사용자 릴레이션

- 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

명함 릴레이션

- 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

회사 정보 릴레이션

- 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

회사 전화번호 릴레이션

- 기본 키가 아닌 속성이 존재하지 않으므로, 부분 함수 종속이 일어날 수 없음.

데이터베이스 설계

정규화
- 제 3 정규화

제 3 정규화는 주 식별자가 아닌 속성들 중에 종속 관계에 있는 속성을 찾아 제거하는 과정
(이행 함수 종속 제거)

회사 주소를 상세하게 나눠 분리 하게 되면,
이행 함수 종속 제거를 진행 해야함.

그러나, 우리가 실제로 명함 프로그램에서 사람을 찾을 때는
“사람의 상세 정보 까지 다 봐야 하므로, 분리하지 않습니다.”

정규화 결과

릴레이션이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
사용자	사용자 번호	NOT NULL	PK				
	이름	NOT NULL					
	아이디	NOT NULL					
	비밀번호	NOT NULL					
명함	개인 고유번호	NOT NULL	PK				
	성명	NOT NULL					
	부서						
	직급						
	휴대전화번호	NOT NULL					
	이메일						
	저장 일자	NOT NULL					
	사용자 번호	NOT NULL		FK	사용자	사용자번호	
회사 정보	회사 번호	NOT NULL	PK				
	회사명	NOT NULL					
	회사 주소						
	회사 팩스 번호						
	회사 전화번호						다중 값
	개인 고유 번호	NOT NULL		FK	명함	개인 고유번호	
릴레이션 이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
회사 전화번호	회사 전화번호	NOT NULL	PK				
	회사 번호	NOT NULL	PK	FK	회사 정보	회사 번호	

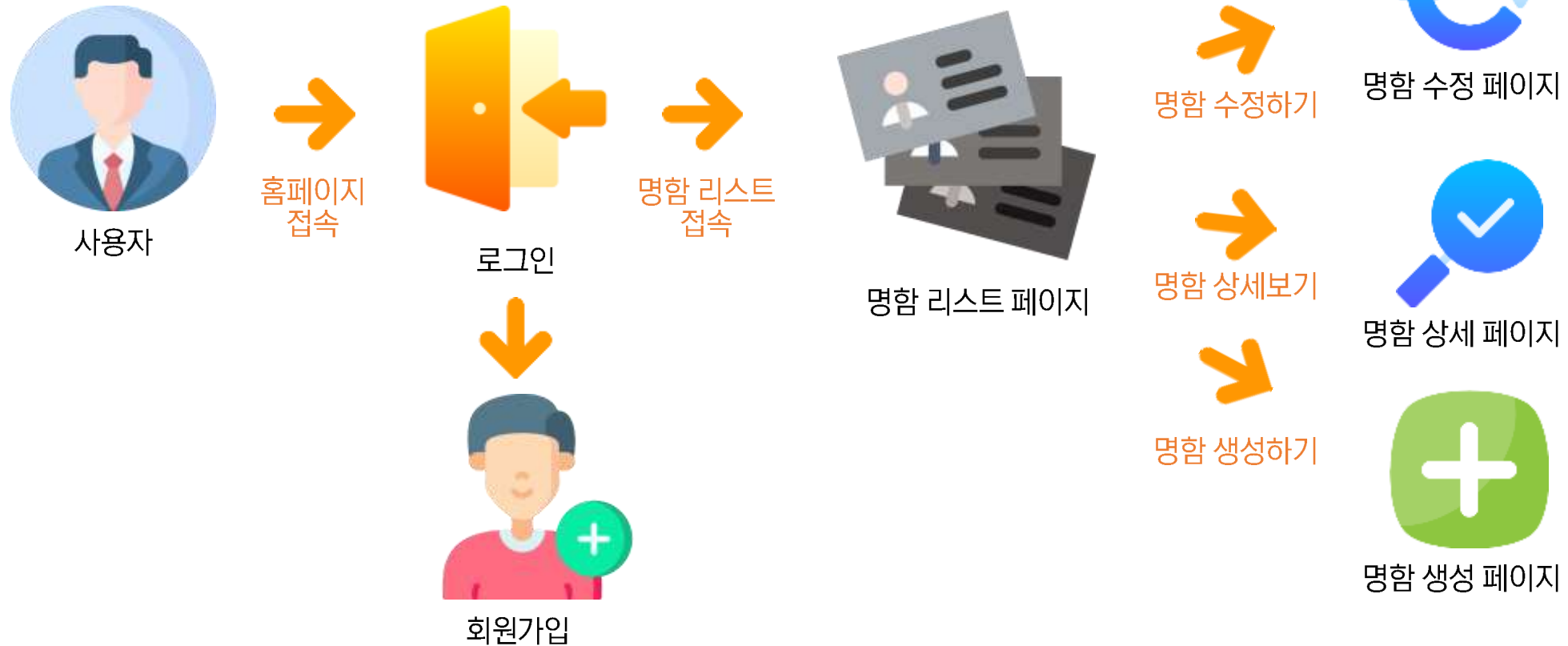
데이터베이스 설계

최종 테이블 명세서

테이블 명	열 이름	데이터 형식	NULL유무	PK	FK	FK 테이블명	비고
사용자	사용자 번호	INTEGER	NOT NULL	PK			AI
	이름	VARCHAR(10)	NOT NULL				
	아이디	VARCHAR(11)	NOT NULL				
	비밀번호	VARCHAR(8)	NOT NULL				
명함	개인 고유번호	INTEGER	NOT NULL	PK			AI
	성명	VARCHAR(10)	NOT NULL				
	부서	VARCHAR(10)					
	직급	VARCHAR(5)					
	휴대전화번호	VARCHAR(15)	NOT NULL				
	이메일	VARCHAR(45)					
	저장 일자	DATETIME	NOT NULL				
	사용자 번호	INTEGER	NOT NULL		FK	사용자	
회사 정보	회사 번호	INTEGER	NOT NULL	PK			AI
	회사명	VARCHAR(10)	NOT NULL				
	회사 주소	VARCHAR(45)					
	회사 팩스 번호	VARCHAR(15)					
	개인 고유번호	INTEGER	NOT NULL		FK	명함	
회사 전화번호	회사 전화번호	VARCHAR(15)	NOT NULL	PK			
	회사 번호	INTEGER	NOT NULL		FK	회사 정보	

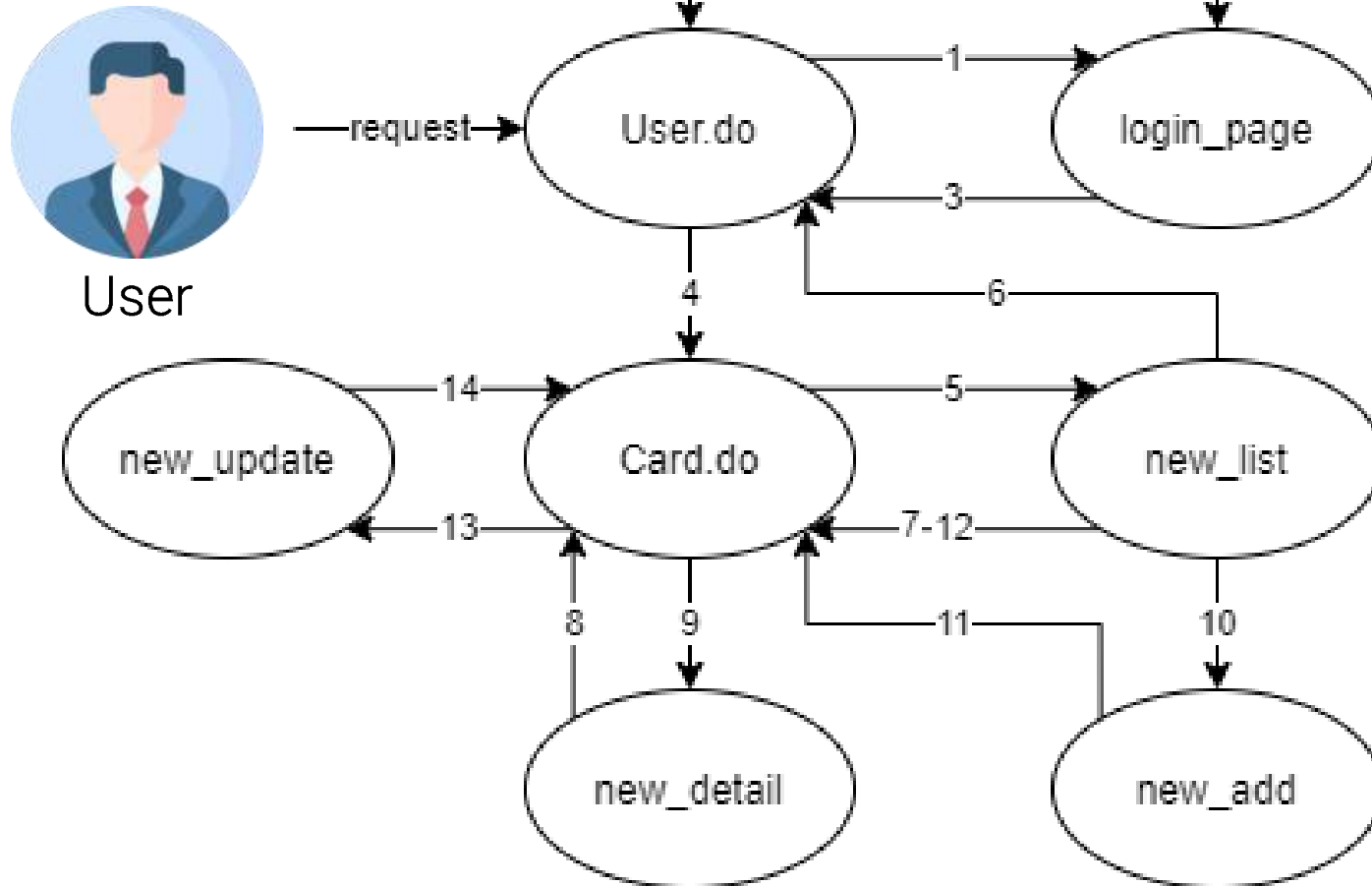
프로그램 설명

시나리오



프로그램 설명

LOGIC



(1)	사용자 request를 확인 후 로그인 페이지로 이동합니다.
(2)	로그인 페이지에서 회원가입 페이지로 이동 / 회원가입 페이지에서 로그인 페이지로 이동 회원가입 완료 시 (3) 회원가입 등록 후 (1) 로그인 페이지 이동
(3)	login_page 입력해서 로그인 여부를 확인 / signup 입력하여 회원가입을 등록
(4)	로그인 여부가 승인되면, Card.do로 이동
(5)	Card.do에서 new_list로 이동하면서 명함 리스트로 이동
(6)	로그아웃 시 User.do로 이동 후 (1) 이동
(7)	상세 버튼 클릭 시 Card.do로 이동 후 ActionName 확인
(8)	ActionName에 맞게 new_detail로 이동
(9)	뒤로가기 버튼을 누르면 Card.do로 이동 후 (5) new_list로 이동
(10)	명함 리스트에서 명함 등록을 누르면 new_add 페이지로 이동
(11)	new_add 페이지에서 등록을 하면 Card.do에서 명함 등록 처리
(12)	명함 삭제를 누르면 Card.do 이동 후 삭제 처리
(13)	명함 리스트에서 수정을 누르면 (12) Card.do에서 ActionName을 확인 후 (13) new_update 이동
(14)	수정 완료를 누르면 Card.do로 처리 후 명함 리스트로 이동 (5)

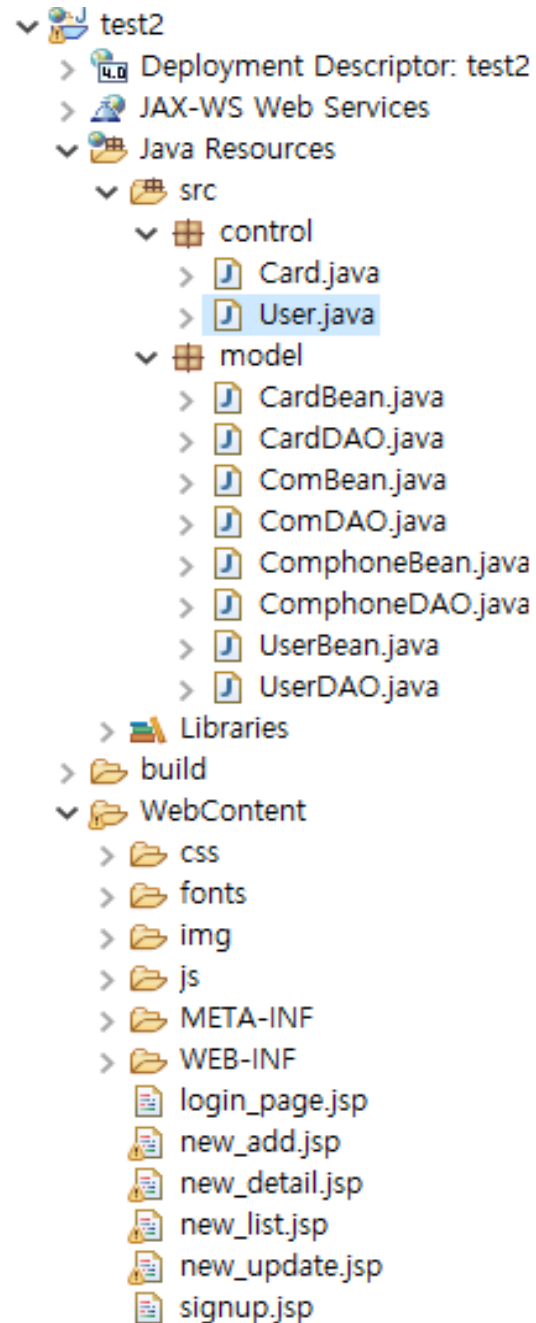
프로그램 설명

Page 및 Logic 표

사용자		명함	
User.do	사용자 로직 처리	Card.do	명함 로직 처리
		new_list	명함 리스트 페이지
signup	회원가입 페이지	new_add	명함 등록 페이지
		new_detail	명함 상세 페이지
login_page	로그인 페이지	new_update	명함 수정 페이지
		리스트 삭제 버튼 클릭 시	Card.do에서 처리

프로그램 설명

MVC 및 프로젝트 구조



Control (Inner Logic)

Card.java - 명함 로직 처리
User.java - 사용자 로직 처리

Model (Database)

CardBean.java - 명함 자바빈
ComBean.java - 회사 자바빈
ComphoneBean.java - 회사번호 자바빈
UserBean.java - 사용자 자바빈
CardDAO.java - 명함 데이터 처리
ComDAO.java - 회사 데이터 처리
ComphoneDAO.java - 회사번호 데이터처리
UserDAO.java - 사용자 데이터 처리

View (User Interface)

login_page.jsp - 로그인 페이지
new_add.jsp - 명함 등록 페이지
new_detail.jsp - 명함 상세 페이지
new_list.jsp - 명함 리스트 페이지
new_update.jsp - 명함 수정 페이지
signup.jsp - 회원가입 페이지

프로그램 설명

결과 화면

로그인

여러분의 계정을 등록해보세요.
여러 명함 관리를 할 수 있습니다!

아이디를 입력해주세요.

비밀번호를 입력해주세요.

로그인

아직 **회원**이 아니신가요?

지금 회원에 가입하시면
최상의 명함 관리를 할 수 있습니다.

회원가입

회원가입

여러분의 계정을 등록해보세요.
여러 명함 관리를 할 수 있습니다!

이름

아이디(5~11자)

비밀번호(최소 8자)

비밀번호 확인(최소 8자)

회원가입

로그인 하러가기

회원 가입을 하게 되면 다양한 혜택이 쏟아진다!



명함 목록

로그아웃 | 명함등록 | 남태우님이 접속하였습니다.

이름을 검색하시오.



순천향대학교

최민석

010-1234-5678
충남 아산시 신창면
소프트웨어공학
도우미

수정 삭제



순천향대학교 공대

남태우


010-3310-9305
대전광역시 서구
데이터베이스
영상도우미

수정 삭제

프로그램 설명

결과 화면

< 명함 상세






최민석
소프트웨어공학
순천향대학교
도우미

연락처

휴대전화
010-1234-5678

이메일
chlaistjr@naver.com

회사 주소
충남 아산시 신창면



< 명함 생성

이름

이름 입력

휴대폰번호

휴대폰번호 입력

직책

직책 입력

부서

부서명 입력

회사

회사명 입력

이메일

이메일 입력

< 명함 수정

이름

남태우

휴대폰번호

010-3310-9305

직책

영상도우미

부서

데이터베이스

회사

순천향대학교 공대

이메일

skaxodn97@gmail.com

프로그램 설명

사용자 가이드북

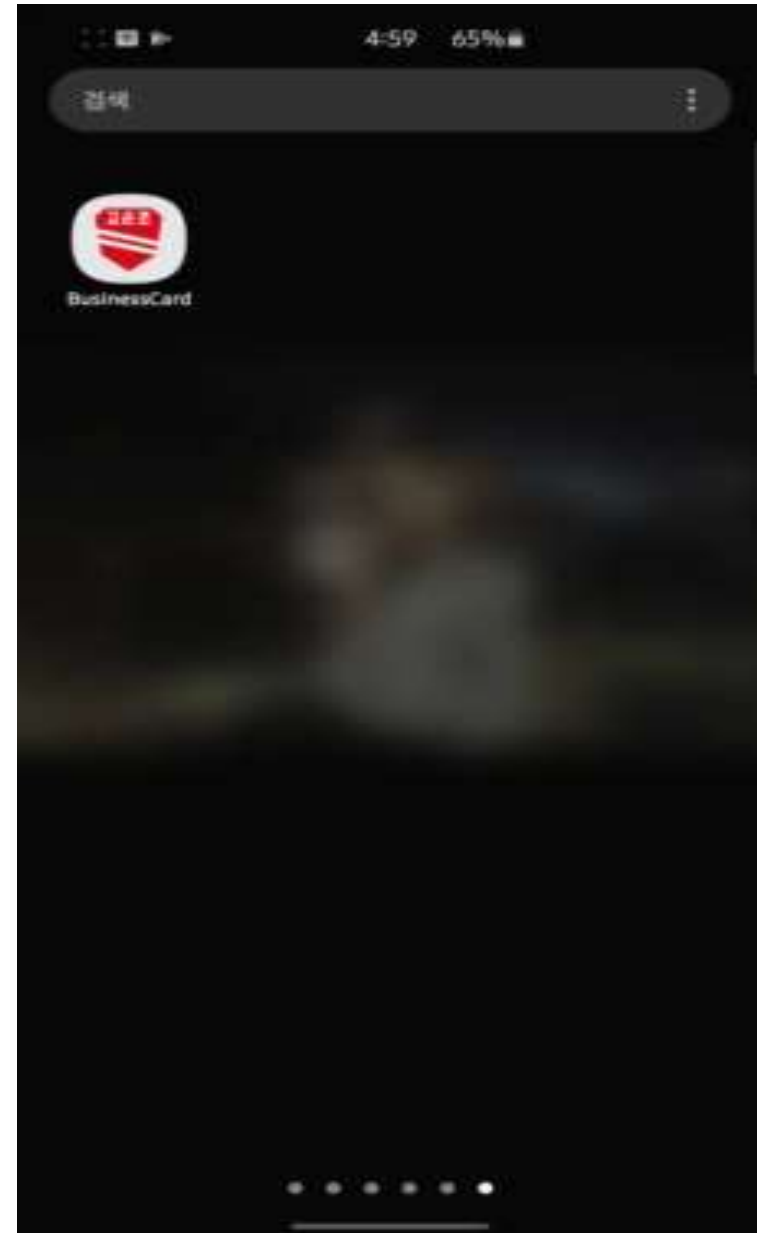
1. 로그인 페이지에 들어갑니다.
2. 회원가입 페이지에 들어가며, 계정 생성을 합니다.
3. 메인 리스트 화면(명함 리스트)에 들어갑니다.
4. 명함 등록 버튼을 클릭하여 명함 등록 페이지로 이동합니다.
5. 관리하고 싶은 명함을 토대로 명함 폼을 작성하며 생성합니다.
6. 등록한 명함을 클릭하면 명함에 대한 상세 내역이 나옵니다.
7. 수정하고 싶으면 폼에 맞게 수정을 진행합니다.
8. 삭제를 하고 싶으면 명함의 아래 삭제 버튼을 클릭합니다.
9. 사용을 종료하고 싶으면 로그아웃 버튼을 통해 로그아웃 합니다.

시연 영상 및 Github 공유



Github
[프로젝트 Repo 가기](#)

youtube 시연영상
https://www.youtube.com/watch?v=703DDaHX5_Q



질문