

# 보고서



# 차 례

## 1. 팀 소개

- 1.1 팀원 소개
- 1.2 역할 분담
- 1.3 프로젝트 활동 간 규칙

## 2. 개요

- 2.1 주제 선정 배경
- 2.2 프로그램의 목적

## 3. 데이터베이스 설계

- 3.1 사용자 정보 요구사항
- 3.2 개념적 설계
- 3.3 논리적 설계
- 3.4 정규화

## 4. 프로그램 설명

- 4.1 GUI

## 5. 결과 화면

## 6. 결론

- 6.1 소감

## 7. 부록

- 7.1 사용자 가이드 북

# 1. 팀 소개

## 1.1. 팀원 소개



최 민 석



남 태 우

## 1.2. 역할 분담

남 태 우

- ① 데이터베이스 설계
- ② Back-end

최 민 석

- ① 데이터베이스 설계
- ② Front-end

## 1.3. 프로젝트 활동 간 규칙

- ① GitHub로 협업 능력 키우기
- ② 불만 사항 피드백 실시간 진행하기
- ③ 프로젝트 완성 시켜 프로그래밍 실력과 뿌듯함 얻기
- ④ 포기하지 않고 최선을 다하기

## 2. 개요

### 2.1. 주제 선정 배경

학생인 신분은 명함이라는 것을 관리하지 않기 때문에 사회에 나가 사람들을 만나 주고 받는 물건이 명함.

최근, 많은 사람들도 명함관리 어플리케이션을 자주 사용하는 것으로 파악이 되어, 명함관리 프로젝트를 웹앱 형태로 개발해서 사람들이 원하는 정보를 보다 빠르게 얻을 수 있는 서비스를 제공

### 2.2. 프로그램의 목적

데이터베이스 프로그래밍인 과목인 만큼, 현실 세계에서 종이 명함으로 관리하기 힘들 속에서 데이터베이스를 활용하여 필요한 정보를 보다 빠르게 찾을 수 있게 할 수 있다.

---

## 3. 데이터베이스 설계

### 3.1. 사용자 정보 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.

### 3.2. 개념적 설계

- 요구사항의 명사 표시

사용자의 정보 요구사항을 토대로 명사를 찾아서 분석 엔티티 풀을 이용하여 분석

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

- 엔티티 기술서

분석을 하여 최종 엔티티 도출 및 속성 등을 정의합니다.

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	
		휴대전화번호	
		이메일 주소	
		저장 일자	
회사 정보	명함의 회사에 대한 정보	회사 번호	
		회사명	
		회사 주소	
		회사 팩스 번호	
		회사 전화번호	다중 값

## • 엔티티 도출 결과

분석을 하여 최종 엔티티 도출 및 속성 등을 정의합니다.

사용자	명함	회사 정보
◆ 사용자 번호	◆ 개인 고유번호	◆ 회사 번호
이름	성명	회사명
아이디	부서	회사 주소
비밀번호	직급	회사 팩스 번호
	휴대전화번호	회사 전화번호
	이메일	
	저장일자	

사용자 요구사항에는 존재하지 않지만, 개인 정보가 중요 시 되고 있는 추세에, 주민등록번호, 아이디(유니크 값으로 한다고 하더라도) 노출을 안 시키기 위해서 의미로 고유 번호를 주어 식별자로 정하였습니다.

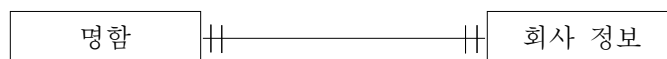
## • 엔티티 간의 관계성 정의

엔티티를 토대로 엔티티 간의 관계를 정의합니다.

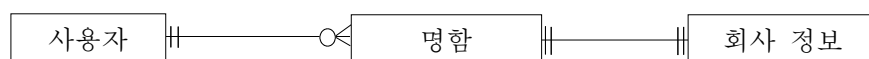
- 각 사용자는 명함을 등록할 수도 있다.
- 각 사용자는 여러 개의 명함을 등록 할 수 있다.



- 명함에는 회사 정보가 관리된다.
- 명함에는 하나의 회사 정보가 적혀 있다.



## • 최종 ERD



### 3.3. 논리적 설계

#### · 엔티티와 단순 속성 변환

위의 작성한 엔티티를 통해 단순 속성 변환을 진행합니다.

릴레이션 이름	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

• 관계 변환

엔티티의 관계를 토대로 관계 변환을 진행합니다.

1:N 관계인 경우,

ERD에서 일측 엔티티에 대응하는 릴레이션의 기본 키를 다측 엔티티에 대응하는 릴레이션의 속성으로 복사한 다음에 이 속성을 외래 키로 지정합니다.

1:1 관계인 경우,

어느 곳에 외래 키를 지정하든 상관 없지만, 자주 사용되거나 안되나 등을 따져서 지정하면 됩니다.

명함과 회사정보에서 회사정보 릴레이션에 외래키로 지정합니다.

릴레이션 이름	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	사용자 번호	
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	



### 3.4. 정규화

#### • 제 1정규화

제 1정규화는 엔티티에서 하나의 속성이 복수 개의 값을 갖도록 설계되어 있을 때, 하나의 속성이 단일 값(atomic value)을 갖도록 설계를 변경하는 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	

회사 정보 릴레이션에서 회사 전화번호는 다중 값입니다.

제 1정규화를 만족하기 위해서는 원자 값이 되어야 합니다.

그래서, 릴레이션을 분리를 통해 해소해야 합니다.

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		FK	회사 정보	회사 번호	

릴레이션 분리할 때 회사 정보의 기본 키를 회사 전화번호 릴레이션의 외래키로 가져옵니다.

회사 번호로만 회사 전화번호 릴레이션에서 식별을 할 수 없기에 회사 전화번호와 같이 기본 키로 설정합니다.

- 제 2정규화

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

사용자 릴레이션을 보면 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

명함 릴레이션을 보면 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

회사정보 릴레이션을 보면 복합 키가 아니므로 부분 함수 종속이 일어날 수 없음.

회사 전화번호 릴레이션을 보면 기본 키가 아닌 속성이 존재하지 않으므로 부분 함수 종속이 일어날 수 없음.

- 제 3정규화

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

회사 주소를 상세히 나눠서 분리하게 되면, 이행 함수 종속 제거를 진행해야 하지만, 실제로 명함 프로그램에서 사람을 찾을 때 그사람의 상세 정보까지 다 봐야하기 때문에 분리하지 않음.

• 정규화 결과

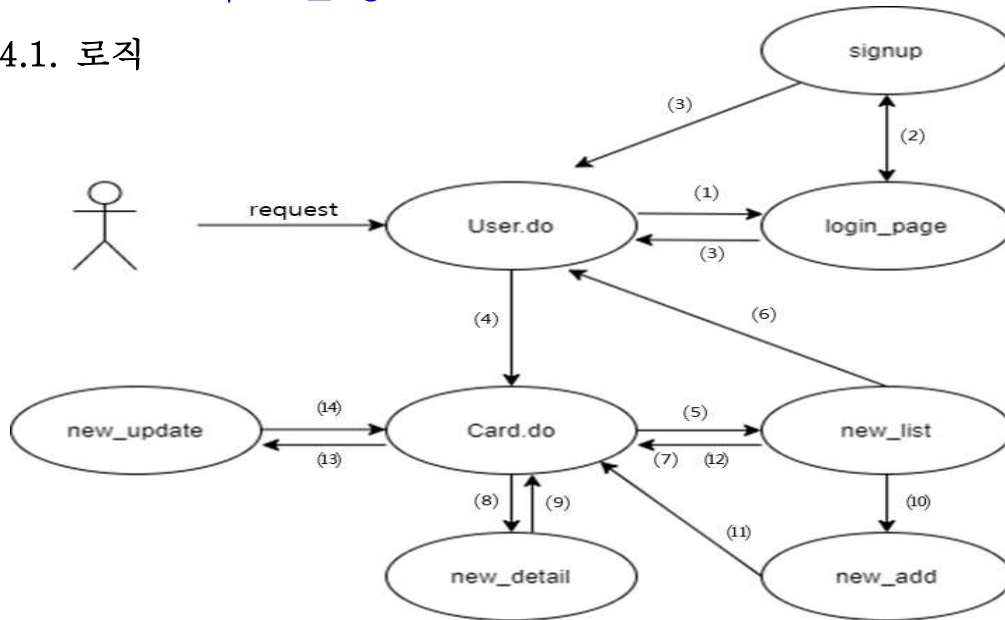
릴레이션 이름	속성 이름	NULL 유무	기본 키	외래 키	FK 릴레이션 이름	FK 속성 이름	비고
사용자	사용자 번호	NOT NULL	PK				
	이름	NOT NULL					
	아이디	NOT NULL					
	비밀번호	NOT NULL					
명함	개인 고유번호	NOT NULL	PK				
	성명	NOT NULL					
	부서						
	직급						
	휴대전화번호	NOT NULL					
	이메일						
	저장 일자	NOT NULL					
	사용자 번호	NOT NULL		FK	사용자	사용자 번호	
회사 정보	회사 번호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

• 최종 테이블 명세서

테이블 명	열 이름	데이터 형식	NULL 유무	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	

## 4. 프로그램 설명

### 4.1. 로직

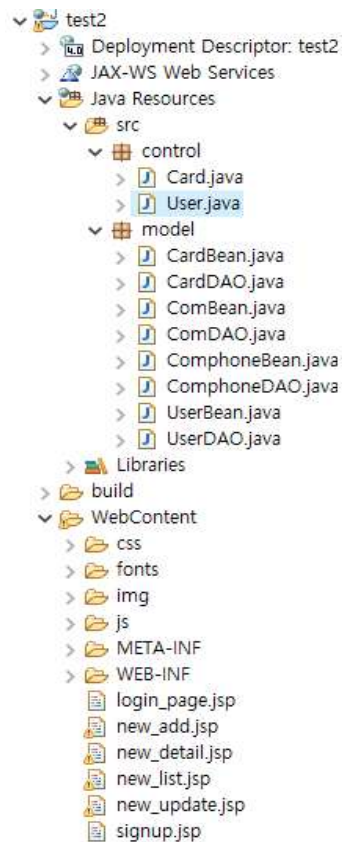


사용자		명함	
User.do	사용자 로직 처리	Card.do	명함 로직 처리
		new_list	명함 리스트 페이지
signup	회원가입 페이지	new_add	명함 등록 페이지
		new_detail	명함 상세 페이지
login_page	페이지	new_update	명함 수정 페이지
		리스트 삭제 버튼 클릭 시	Card.do에서 처리

(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)

## 4.2. Graphic User Interface

### 4.2.1 프로젝트 구조



#### control → 내부 로직 처리

Card.java → 명함 로직 처리

User.java → 사용자 로직 처리

#### model → 데이터베이스 처리

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 데이터 처리

UserDAO.java → 사용자 데이터 처리

#### view → 사용자에게 보여 지는 뷰

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

## 5. 결과 화면

로그인	회원가입
<p>여러분의 계정을 등록해보세요. 여러 명함 관리를 할 수 있습니다!</p> <p>아이디를 입력해주세요.</p> <p>비밀번호를 입력해주세요.</p> <p><b>로그인</b></p> <p>아직 <b>회원</b>이 아니신가요?</p> <p>지금 회원에 가입하시면 최상의 명함 관리를 할 수 있습니다.</p> <p><b>회원가입</b></p>	<p>여러분의 계정을 등록해보세요. 여러 명함 관리를 할 수 있습니다!</p> <p>이름</p> <p>아이디(5~11자)</p> <p>비밀번호(최소 8자)</p> <p>비밀번호 확인(최소 8자)</p> <p><b>회원가입</b></p> <p><b>로그인 하러가기</b></p> <p>회원 가입을 하게 되면 다양한 혜택이 쏟아진다!</p>

<div><div>&lt;</div><div>명함 상세</div></div> <div><div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div>최민석</div><div>소프트웨어공학</div><div>순천향대학교</div><div>도우미</div></div></div></div> <div><div>연락처</div><div>휴대전화</div><div>010-1234-5678</div><div><div></div><div></div></div></div> <div><div>이메일</div><div>chlaistjr@naver.com</div><div><div></div></div></div> <div><div>회사 주소</div><div>충남 아산시 신창면</div><div><div></div></div></div>	<div><div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div>명함 목록</div><div>로그아웃   명함등록   남태우님이 접속하였습니다.</div></div><div><div>이름을 검색하십시오</div><div>Q</div></div><div><div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div>순천향대학교</div></div></div><div><div>최민석</div><div>010-1234-5678</div><div>충남 아산시 신창면</div><div>소프트웨어공학</div><div>도우미</div><div>수정 삭제</div></div></div><div><div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div>순천향대학교 공대</div></div></div><div><div>남태우</div><div>010-3310-9305</div><div>대전광역시 서구</div><div>데이터베이스</div><div>영상도우미</div><div>수정 삭제</div></div></div></div></div>
<div><div>&lt;</div><div>명함 생성</div></div> <div><div>이름</div><div>이름 입력</div></div> <div><div>휴대폰번호</div><div>휴대폰번호 입력</div></div> <div><div>직책</div><div>직책 입력</div></div> <div><div>부서</div><div>부서명 입력</div></div> <div><div>회사</div><div>회사명 입력</div></div> <div><div>이메일</div><div>이메일 입력</div></div>	<div><div>&lt;</div><div>명함 수정</div></div> <div><div>이름</div><div>남태우</div></div> <div><div>휴대폰번호</div><div>010-3310-9305</div></div> <div><div>직책</div><div>영상도우미</div></div> <div><div>부서</div><div>데이터베이스</div></div> <div><div>회사</div><div>순천향대학교 공대</div></div> <div><div>이메일</div><div>skaxodn97@gmail.com</div></div>

## 6. 결론

### 6.1. 소감

**남 태 우** : 예제를 통해서 공부하는 것에 익숙해져 있었습니다. 직접 팀을 꾸려 프로젝트 기획부터 디자인 및 개발까지 같이 협업을 해보니 혼자서 하는 것보다 색다른 경험을 느낄 수 있었습니다. 실제 실무에서도 이렇게 하는지 궁금증이 생겼습니다.

깃허브도 혼자만의 저장소로 이용하였지만, 깃허브로 협업하는 것도 배울 수 있는 계기가 되어서 또 하나의 개발 역량이 향상되었다고 생각합니다.

**최 민 석** : 그동안 저학년부터 여러 팀 프로젝트를 했지만, 이런 식으로 프론트 및 백두 분야로 나누어서 처음 진행해보았습니다. 프로젝트 진행이 효율적으로 진행되는 경험을 하여서 재밌었습니다.

## 7. 부록

### 7.1. 사용자 가이드 북

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 로그아웃합니다.

### 7.2. 소스 코드

[https://github.com/bn-tw2020/BusinessCard\\_Service](https://github.com/bn-tw2020/BusinessCard_Service)

### 7.3. 참고 자료

JSP 2.3 & Servlet 3.1 입문부터 모델2 MVC 패턴까지| 자바 분야 최고의 멘토가 설명하는 완벽한 모델 2 개발 방법의 해설서

<http://www.kyobobook.co.kr/product/detailViewKor.laf?ejkGb=KOR&mallGb=KOR&barcode=9788983799180>