

TEAM PROJECT

민해경 손호현 윤하경 조용훈





놀이동산 이용객이 놀이동산의 서비스를 이용하는 프로그램

작업방식

주제선정 및
요구사항분석

사용자는 누구
필요한 기능 정리
의미론적 / 기술론적

설계

Use Diagram작성
DB 모델링
Prototype UI설계

≡
기능 구체화

구현

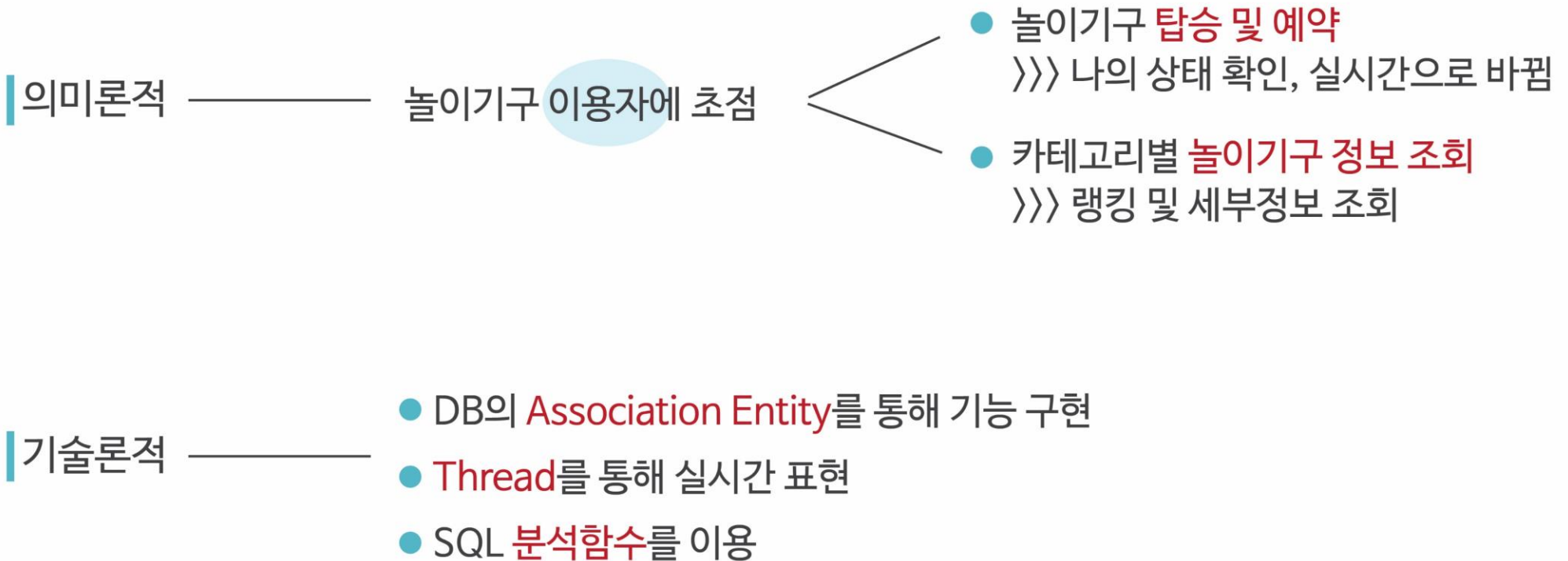
테이블 작성
VO 작성
인터페이스 작성
오버라이딩 후 구현
Thread
Transaction

예외처리

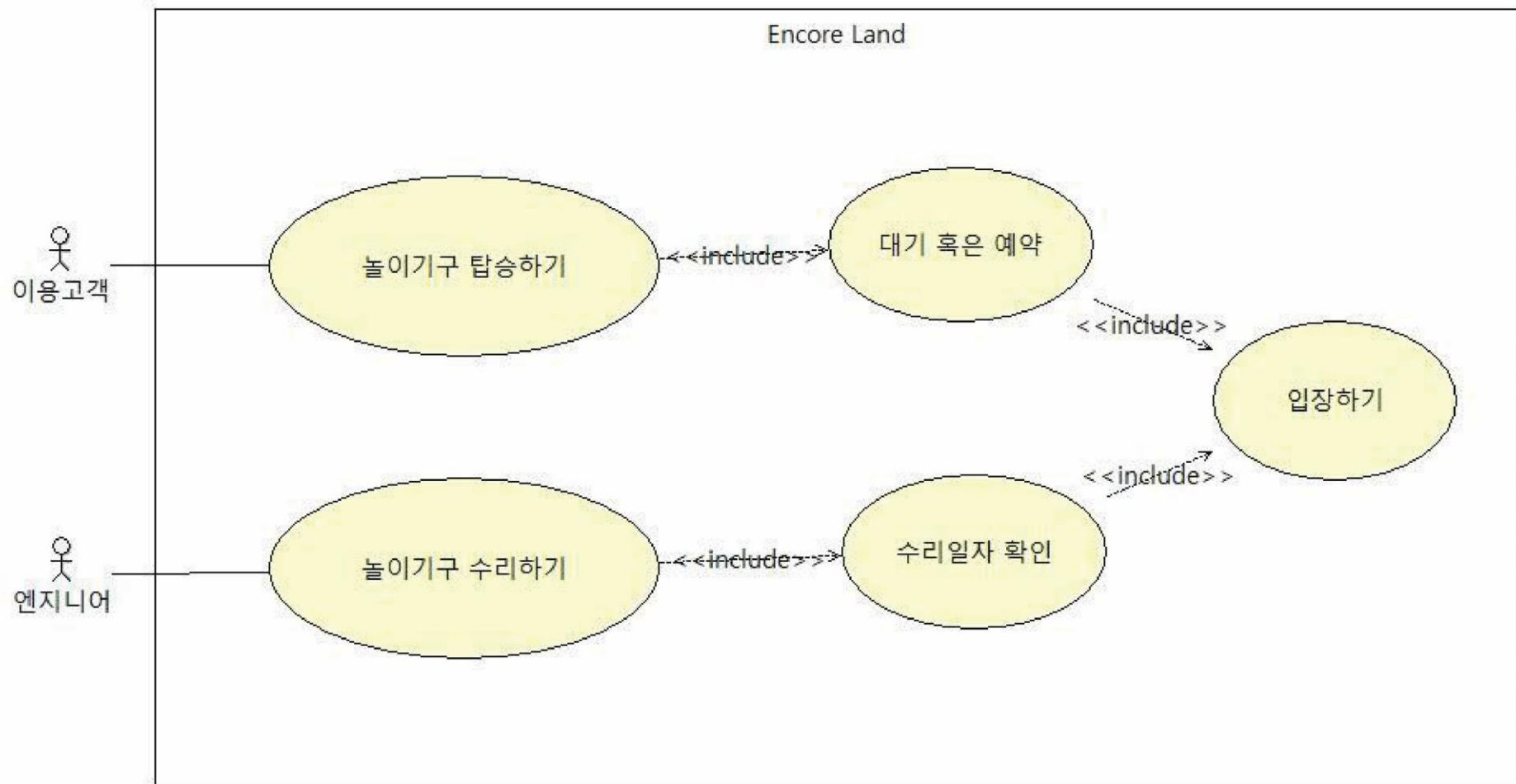
나올 수 있는 예외케이스
수집후 처리

■ TEST

DB에 가상 정보들을 넣고 계속해서 Test함
(random함수를 이용하여 일일이 db를 넣지 않고 자동으로 넣음)

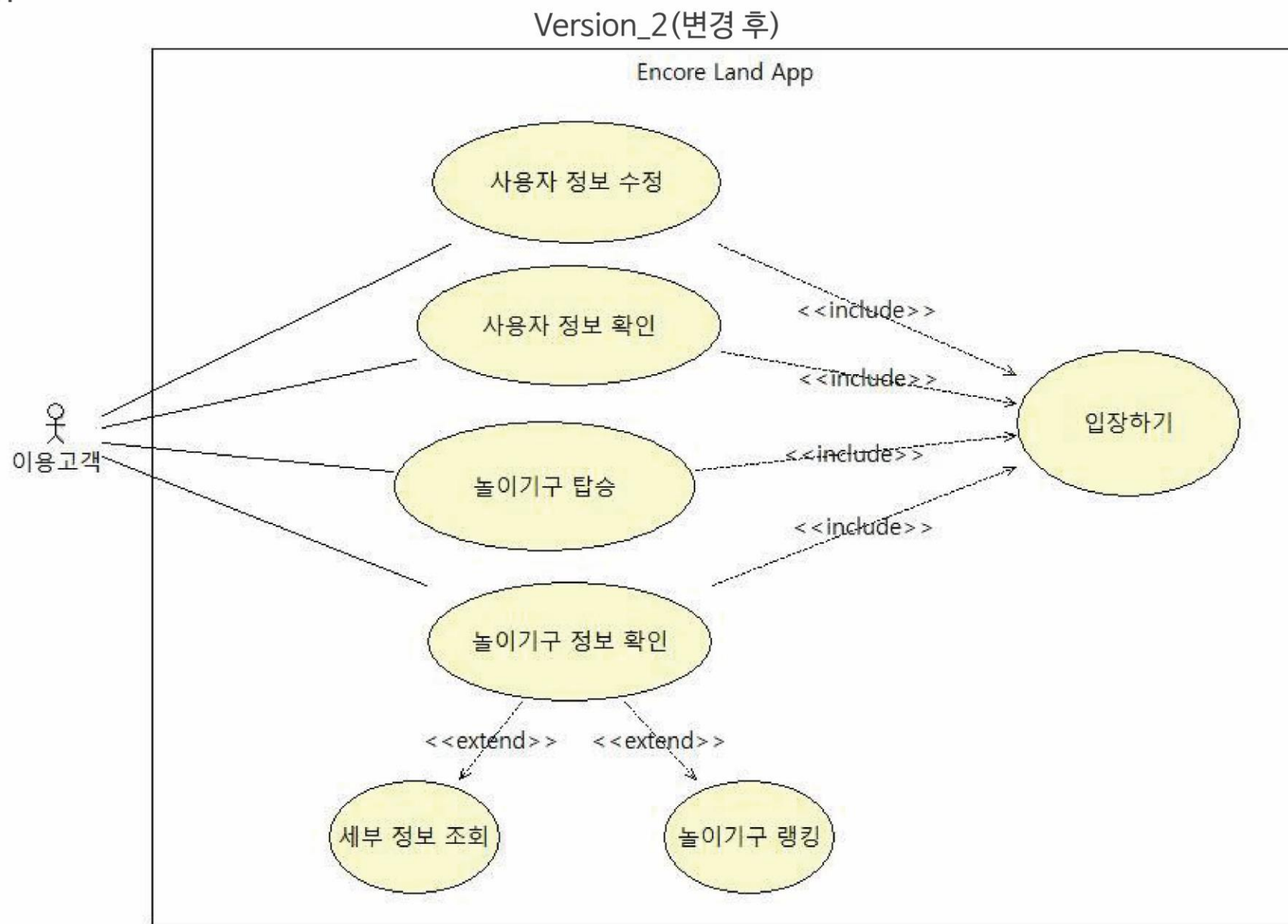


설계_use diagram

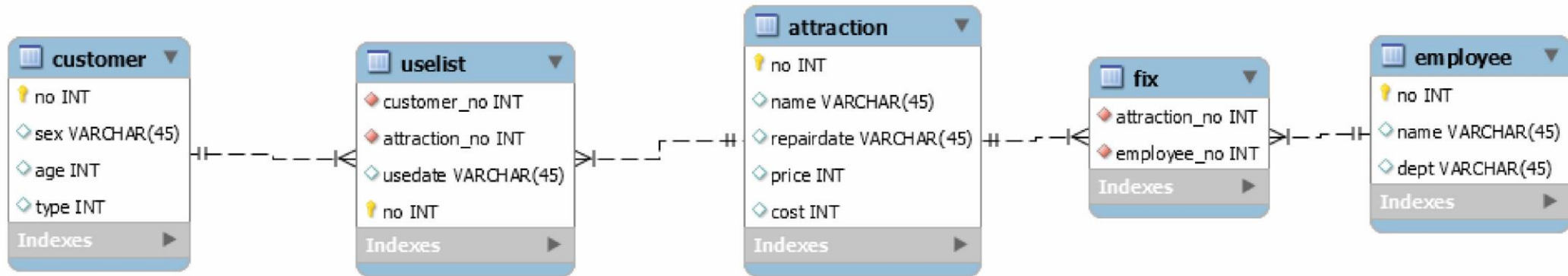


Version_1 (변경 전)

설계_use diagram

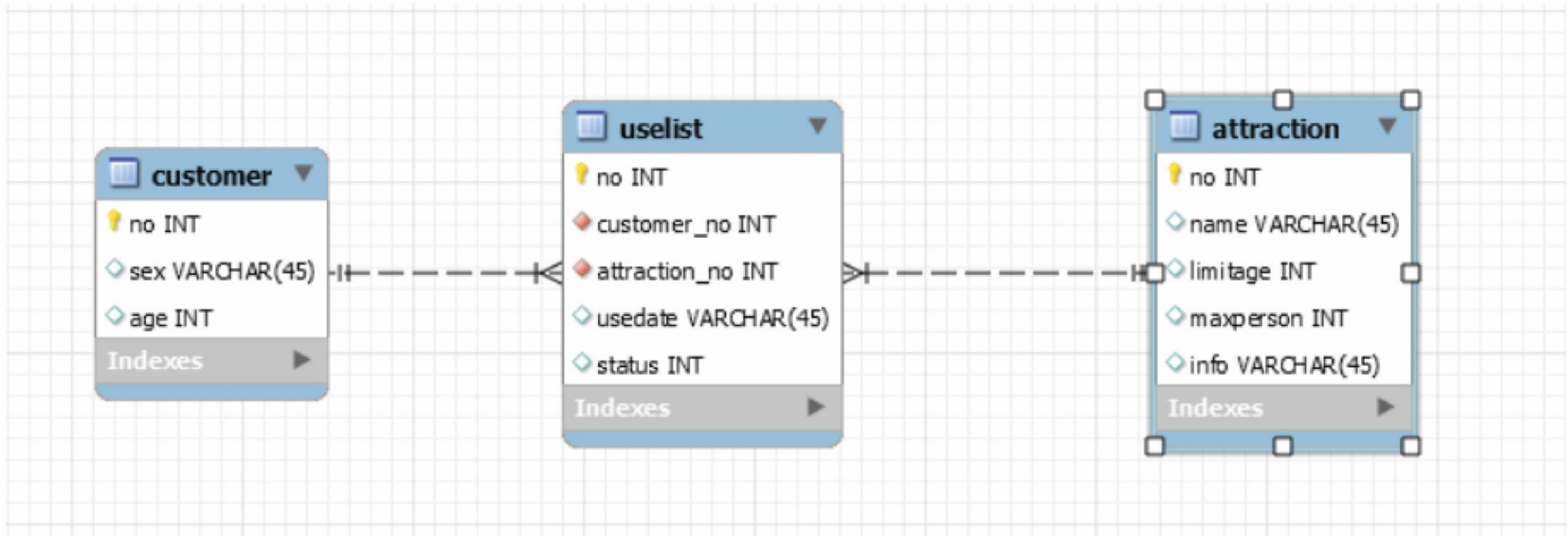


설계_DB Modeling



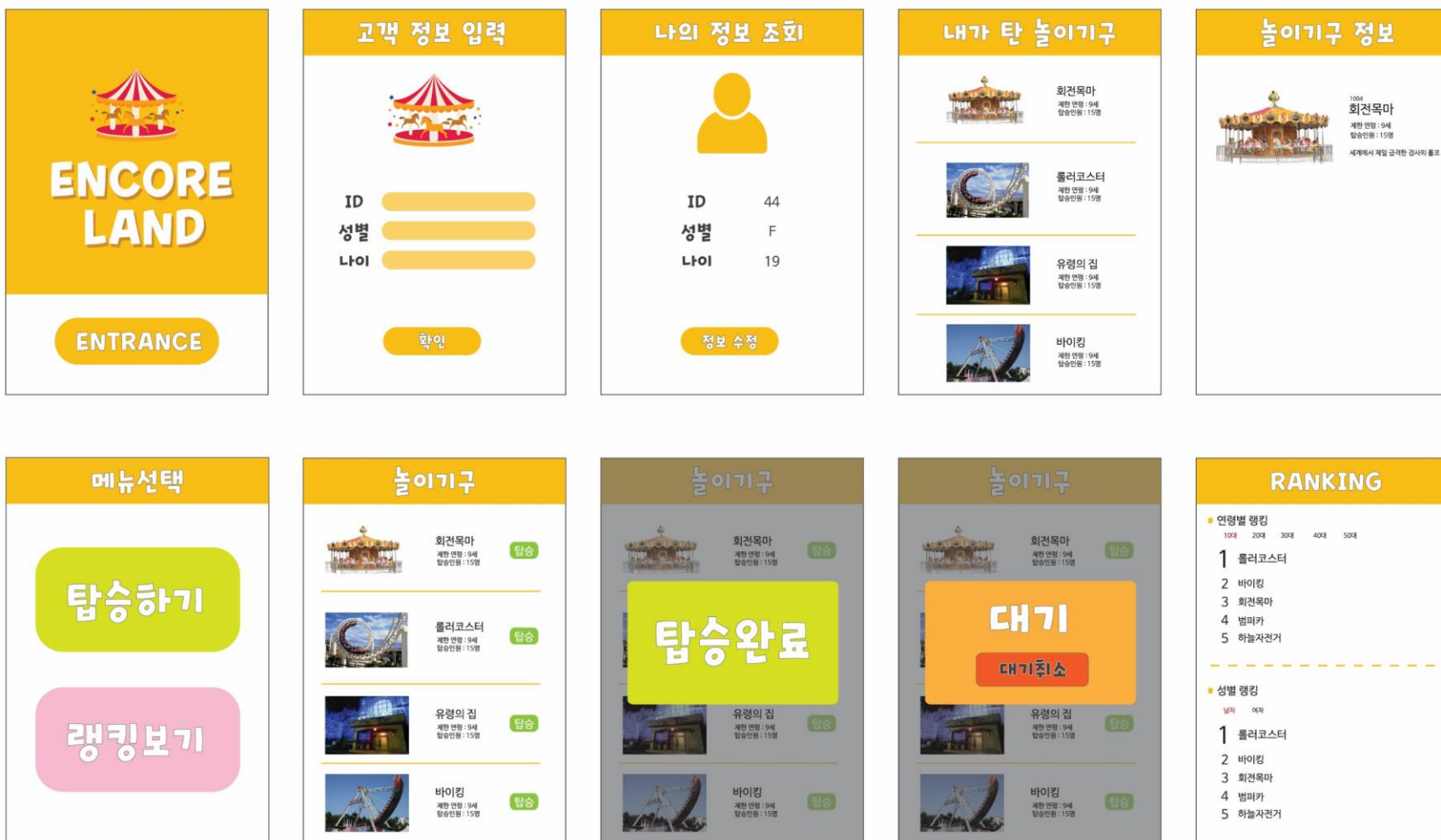
Version_1 (변경 전)

설계_DB Modeling



놀이기구 **이용자**에 초점을 맞추어 구체화

GUI를 그려보면서 비즈니스 로직 구체화



구현_인터페이스

```
public interface EncoreLandDao {  
    //커넥션관련  
    public Connection getConnect() throws SQLException;  
    public void closeAll(PreparedStatement ps, Connection conn) throws SQLException;  
    public void closeAll(ResultSet rs, PreparedStatement ps, Connection conn) throws SQLException;  
  
    public void entranceCustomer(Customer c) throws SQLException; //고객이 놀이공원에 입장하는 메소드  
  
    public ArrayList<UseList> getMyUseList(int custNo) throws SQLException; //내가 탄 놀이기구들 조회  
    public Customer getMyInfo(int custNo) throws SQLException; //내 정보조회  
    public Attraction getAttrInfo(int attrNo) throws SQLException; //놀이기구 정보조회  
    public Attraction getAttrInfo(int attrNo, Connection conn) throws SQLException; //놀이기구 정보 조회(conn포함)  
    public ArrayList<Attraction> getAllAttr() throws SQLException; //모든 놀이기구 조회  
  
    public void updateCustomer(Customer c) throws SQLException; //내정보 수정  
    public void cancelAttr(int custNo, int attrNo) throws SQLException; //대기 취소  
  
    public ArrayList<Attraction> totalRankAttr() throws SQLException; //놀이기구 전체랭킹  
    public ArrayList<Attraction> ageRankAttr(int n, int m) throws SQLException; //연령별 놀이기구 랭킹  
    public ArrayList<Attraction> sexRankAttr(String sex) throws SQLException; //성별 놀이기구 랭킹  
  
    public void useAttraction(int custNo, int attrNo) throws SQLException; //놀이기구 이용  
    public int getCurrentStatus(int attrNo, Connection conn) throws SQLException; //고객상태(실시간)  
    public int getOrder(int custNo, int attrNo); //몇번째 대기인지 알아보는 함수
```

분석

비즈니스

구현_구조 및 예외처리

- ▼ project01-
 - > JRE System Library [JavaSE-1.8]
 - ▼ src
 - ▼ config
 - jdbc.properties
 - ▼ park.dao
 - > EncoreLandDao.java
 - > EncoreLandDaoImpl.java
 - > UpdateThread.java
 - ▼ park.exception
 - > InvalidTransactionException.java
 - > RecordNotFoundException.java
 - ▼ park.test
 - > EncoreLandDaoTest.java
 - ▼ park.vo
 - > Attraction.java
 - > Customer.java
 - > UseList.java

InvalidTransactionException

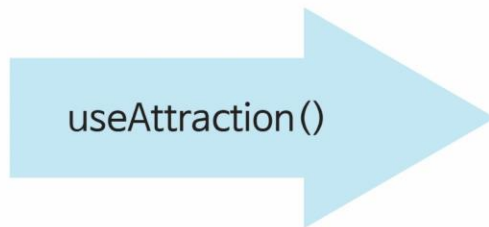
놀이기구의 제한연령보다 사용자의 나이가 더 적을 때
사용자가 이미 대기중인데, 또 대기를 하고자 할 때

RecordNotFoundException

대기상태가 아닌데, 대기취소를 하려고 할 때
사용자가 놀이기구를 아무것도 타지 않았는데,
탑승내역을 조회하려 할 때



사용자



놀이기구

탑승 또는 대기



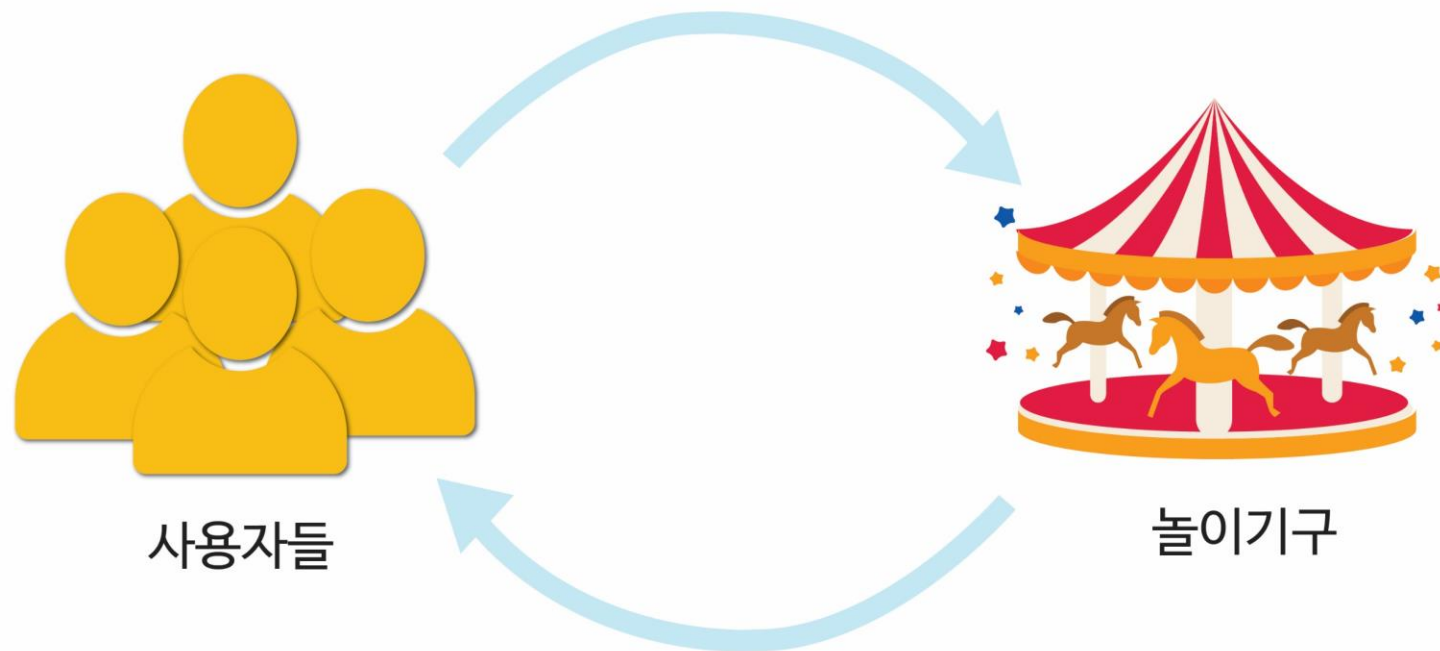
useList에서 Insert

기구의 제한연령과 사용자의 나이 비교

연령이 맞지 않으면 다시 rollback
맞으면 commit



useList에서 Insert 전으로



사용자들의 **상태**(탑승중, 대기, 아무상태)가
놀이기구가 운영을 종료할 때 마다 바뀐다

**ENCORE
LAND**

ENTRANCE

고객 정보 입력



ID


49

성별

나이

확인


나의 정보 조회




ID	44
성별	F
나이	19

정보 수정

나의 탑승권



ID	44
----	----



메뉴선택

탐승하기

랭킹보기

놀이기구 목록



회전목마

제한 연령 : 9세
탐승인원 : 15명

탐승가능



롤러코스터

제한 연령 : 9세
탐승인원 : 15명

예약



유령의 집

제한 연령 : 9세
탐승인원 : 15명

탐승가능



바이킹

제한 연령 : 9세
탐승인원 : 15명

예약

놀이기구



회전목마

제한 연령 : 9세
탐승인원 : 15명

탐승

탐승완료



유령의 집

제한 연령 : 9세
탐승인원 : 15명

탐승



바이킹

제한 연령 : 9세
탐승인원 : 15명

탐승

놀이기구



회전목마

제한 연령 : 9세
탐승인원 : 15명

탐승

대기 대기 : 6번

대기취소



유령의 집

제한 연령 : 9세
탐승인원 : 15명

탐승



바이킹

제한 연령 : 9세
탐승인원 : 15명

탐승



내가 탄 놀이기구



회전목마
제한 연령 : 9세
탑승인원 : 15명



롤러코스터
제한 연령 : 9세
탑승인원 : 15명



유령의 집
제한 연령 : 9세
탑승인원 : 15명



바이킹
제한 연령 : 9세
탑승인원 : 15명

놀이기구 정보



1004
회전목마
제한 연령 : 9세
탑승인원 : 15명

세계에서 제일 급격한 경사의 롤코

느낀점

- 사용자와 프로그램의 목적을 명확히 해야한다는 것을 다시 깨달았다
- 분석단계의 중요성을 다시 한번 체감했다 >>> 시간이 걸려도 자세하게 하는게 좋다
- GUI를 그려봄으로써 비즈니스 로직이 좀더 구체화 되는 것을 느꼈다
- commit 구조를 좀 더 알게 되었다
- 규모가 작아보이는 프로젝트여도 꼼꼼하게 하다보면 할일이 정말 많았다

감사합니다

