|  |  |  |  |
| --- | --- | --- | --- |
| 프로젝트명 | 인디언포커 개발 | 버전 | 1.0 |
| 문서명 | 소프트웨어 설계 명세서 | 작성일 | 11.09 |
| 단계 | 설계 | 작성자 | 소공프렌즈 |

**소프트웨어 설계**

**-소공프렌즈-**

-목차-

[**1.** **유즈케이스 실현** 3](#_Toc466499227)

[1.1 UI design 3](#_Toc466499228)

[**2.** **DB design** 6](#_Toc466499229)

[**3.** **디테일한 Class Diagram** 7](#_Toc466499230)

[**4.** **Component Diagram** 8](#_Toc466499231)

# **유즈케이스 실현**

## UI design

* + 1. 메인 화면



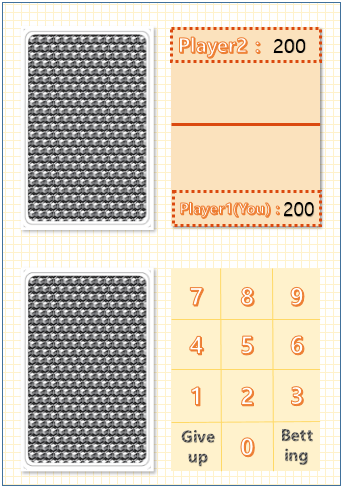
앱을 처음 시작했을 때 보이는 화면이다. start 버튼을 누르면 게임을 시작하는 화면으로 전환하고 Ranking 버튼을 누르면 랭킹을 보여주는 화면으로 전환한다.

1.1.2 Highscore 화면



highscore보기 화면이다. 메인 화면에서 "Ranking"버튼을 누르면 나오는 화면으로 그동안 게임을 한 플레이어들의 점수를 높은 순으로 정렬하여 1위부터 보여주는 화면이다.

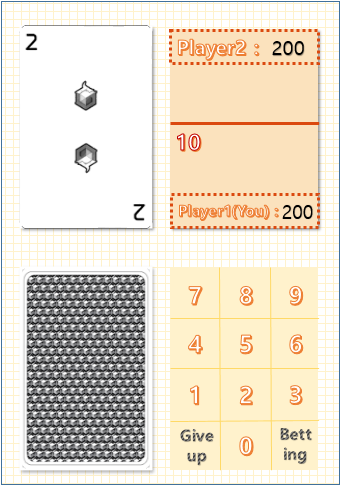
1.1.3 게임 진행 첫 화면

****

Player1과 Player2가 접속한 후 게임이 start될 때의 화면으로, Player1과 Player2는 기본점수 200점을 배당받고 게임을 시작하게된다. 왼쪽 상단은 상대방의 카드, 왼쪽 하단은 내가 뽑은 카드, 오른쪽 상단은 상대와 내가 베팅한 점수판, 오른쪽 하단에는 점수를 베팅을 위한 숫자 버튼, 배당포기를 위한 Give up버튼, 그리고 점수를 배당한 후 제출하는 Betting 버튼이 존재한다.

이 화면은 아직 Player1과 Player2가 자신의 카드를 뽑기 전의 화면이다.

1.1.4 게임 진행 두번째 화면

****

Player1과 Player2가 각자의 카드를 뽑은 후의 화면으로, 상대(Player2)가 뽑은 카드는 무엇인지 나에게 보이지만(왼쪽 상단의 카드), 내가 뽑은 카드는 무엇인지 화면 상에서 보이지 않는다(왼쪽 하단의 카드). 각 Player들은 상대의 카드를 보고 나의 카드숫자를 예측하여 점수를 배당한다. 기본 배당 점수는 10점이고, 이 화면은 Player1이 점수 10점을 배당한 상태의 화면이다.

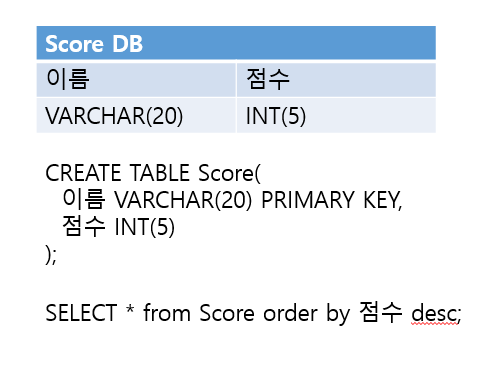
1.1.5게임 종료 화면



게임이 종료되었을 때 나타나는 화면이다. 승자가 누구인지 알려주고 확인을 누르면 메인 화면으로 돌아가게 된다. Player1과 Player2 모두에게 동일하게 나타나는 화면이다.

# **DB design**

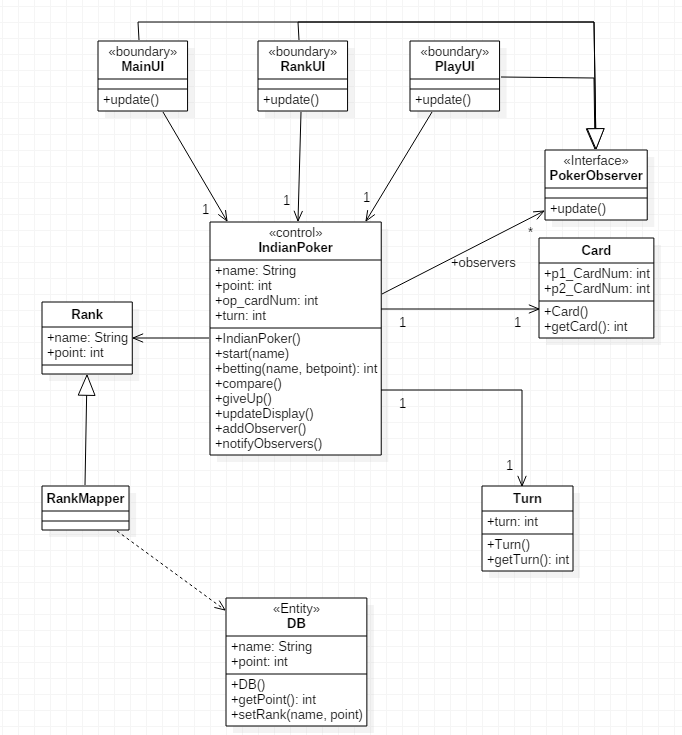
데이터베이스 스키마



게임이 종료된 후 랭킹을 보기 위한 플레이어의 이름과 최종 점수를 기록하는 데이터베이스이다. Score테이블의 속성 중 이름은 varchar(20)타입, 점수는 int(5)타입이며 이름을 기본키로 설정하였다. 또한 랭킹을 확인할 때는 점수가 높은 순으로 정렬하여 확인하도록 select문 설계하였다.

# **디테일한 Class Diagram**

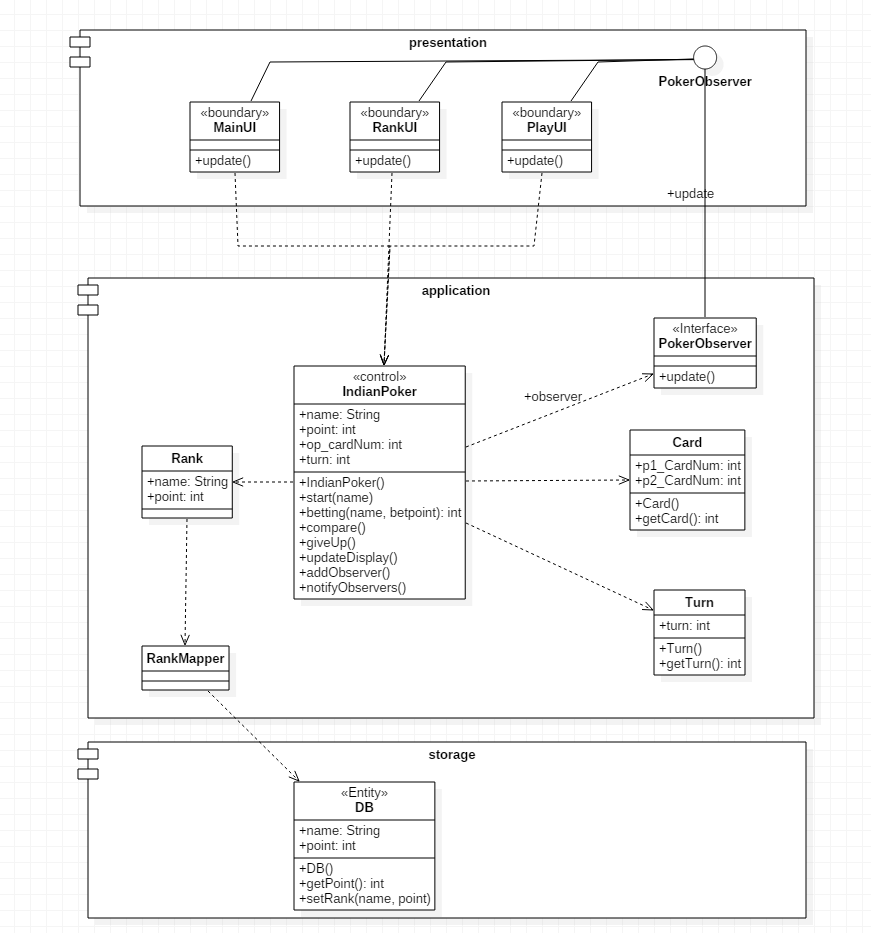
클래스 다이어그램



기존 분석 단계에서 만들었던 클래스다이어그램에서 수정된 부분을 말한다.UI부분을 구현해줄 MainUI, RankUI, PlayUI 클래스가 추가 되었다. Observer 패턴을 이용해서 UI와 Control을 연결해주기 위해 Observer 인터페이스를 만든다. Rank를 저장할 Rank 클래스를 추가하고 DB의 테이블로 매핑해주기 위한 RankMapper클래스도 만든다. Player 클래스는 control과 겹치고 actor이기 때문에 지웠다.

.

# **Component Diagram**



클래스 다이아그램을 바탕으로 MVC패턴을 적용하여 컴포넌트 다이아그램을 작성하였다. 표현은 응용계층의 영향을 받지만 위로 올라가는 형태를 MVC패턴은 나타낼 수 없기 때문에 Observer패턴을 사용하여 인터페이스를 두고 update를 받도록 하였다. 또한 저장계층은 DB스키마를 적용하기 때문에 사용자 정보를 DB와 응용계층이 서로 이어지도록 mapper클래스를 이용하였다.