우리말 퀴즈 어플리케이션

Korean Quiz Application

2013156039 조윤선

목차

[1. 개요 3](#_Toc480481283)

[1.1 시스템 목표 3](#_Toc480481284)

[2. 시스템 수행 시나리오 3](#_Toc480481285)

[2.1 스토리보드 3](#_Toc480481286)

[3. 시스템 구성도 7](#_Toc480481287)

[4. 모듈 상세 설계 8](#_Toc480481288)

[4.1 어플리케이션 8](#_Toc480481289)

[4.1.1 사용자 인터페이스 모듈 User Interface 8](#_Toc480481290)

[4.1.2 초성게임 알고리즘 Initial Game Algorithm 8](#_Toc480481291)

[4.1.3 문장게임 알고리즘 Sentence Game Algorithm 10](#_Toc480481292)

[4.1.4 내부 데이터베이스 11](#_Toc480481293)

[4.2 서버 11](#_Toc480481294)

[4.2.1 유저 데이터 관리 Supervise User Data 11](#_Toc480481295)

[4.2.2 유저 매칭 13](#_Toc480481296)

[4.3 데이터베이스 14](#_Toc480481297)

[5. 개발 환경 및 개발 방법 14](#_Toc480481298)

[5.1 개발 환경 14](#_Toc480481299)

[5.2 개발 방법 14](#_Toc480481300)

[6. 데모 환경 설계 15](#_Toc480481301)

[6.1 사전 설정 15](#_Toc480481302)

[6.2 데모 시나리오 15](#_Toc480481303)

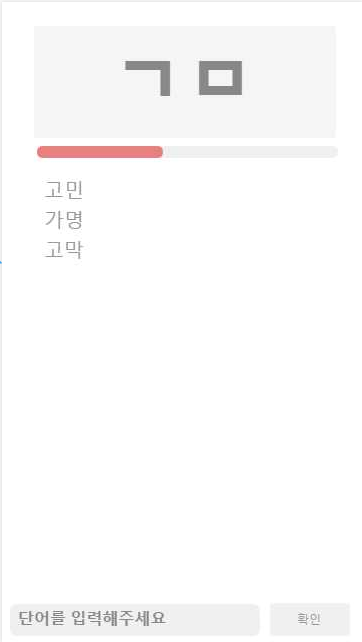
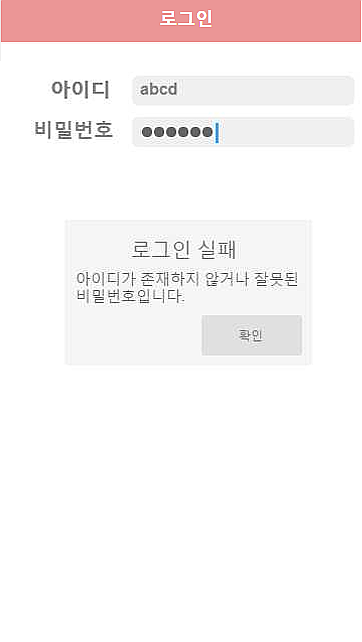
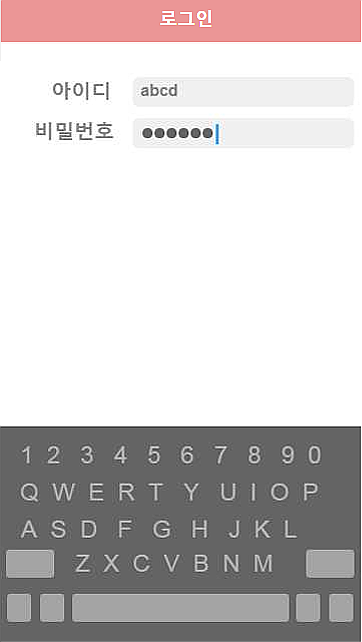
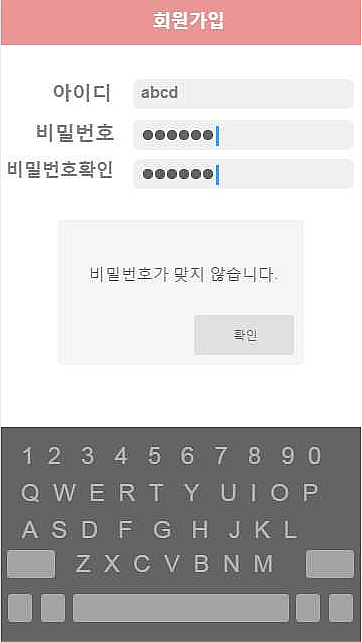
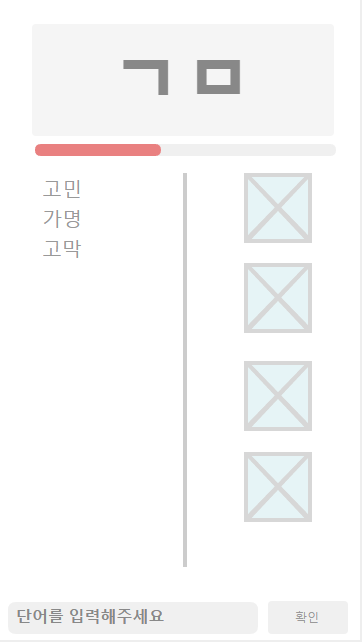
# 1. 개요

## 1.1 시스템 목표

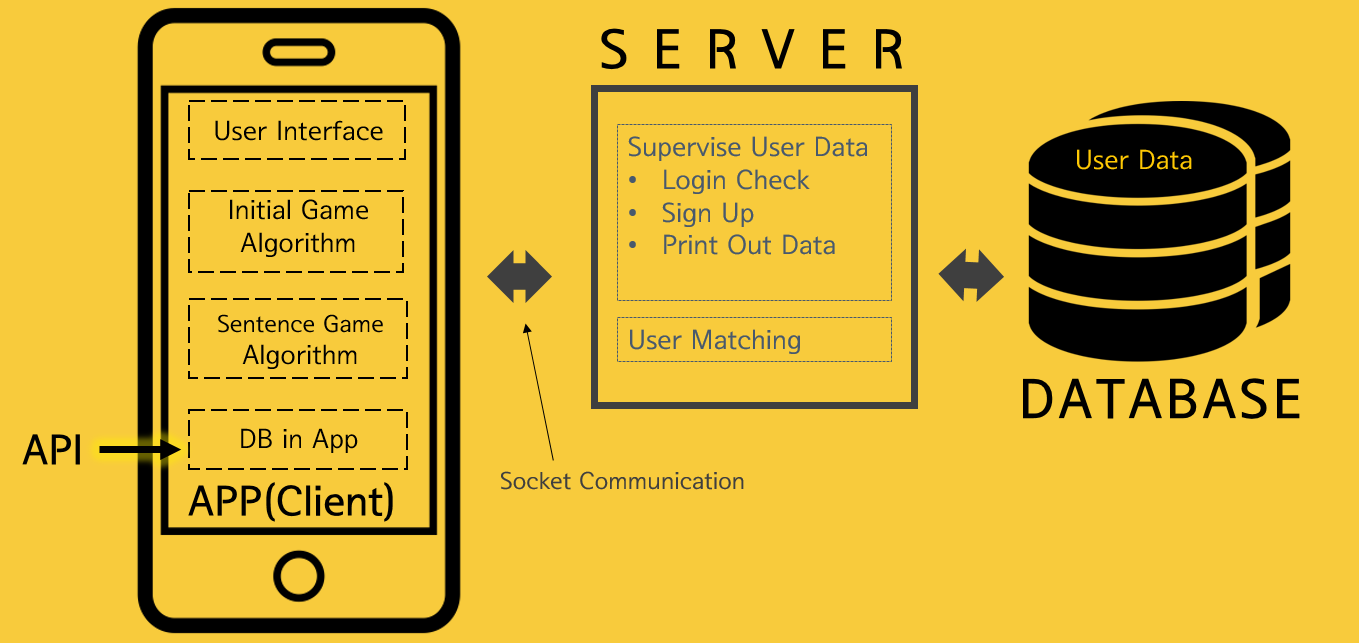
* 게임을 통해서 한글과 맞춤법에 대한 정확한 이해를 돕는다.
* 다중사용자 모드를 지원하여 게임으로써 흥미를 높인다.
* 랭킹을 도출하여 사용자의 성취감을 충족시킨다.

# 2. 시스템 수행 시나리오

## 2.1 스토리보드

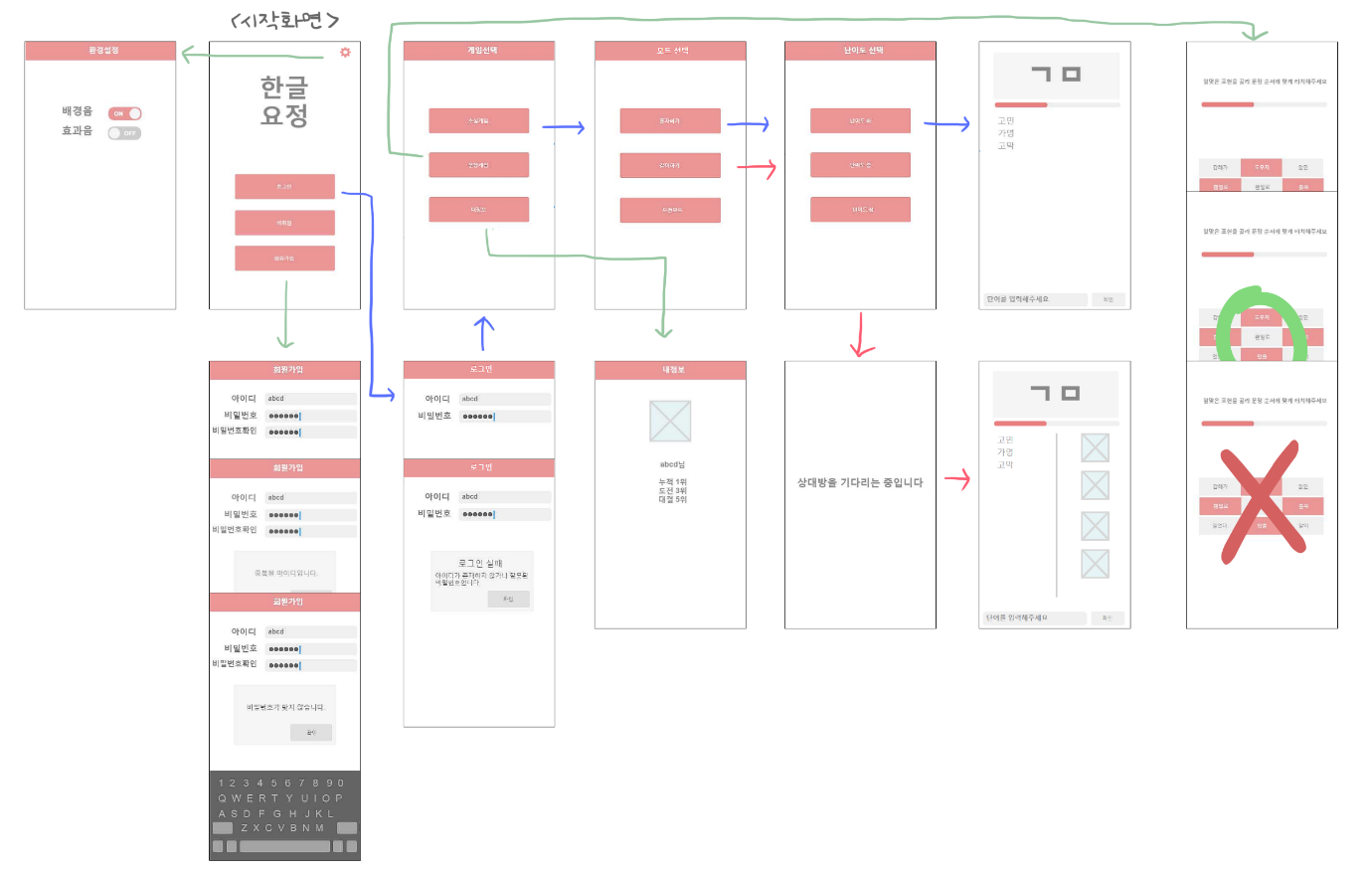
# 3. 시스템 구성도



# 4. 모듈 상세 설계

## 4.1 어플리케이션

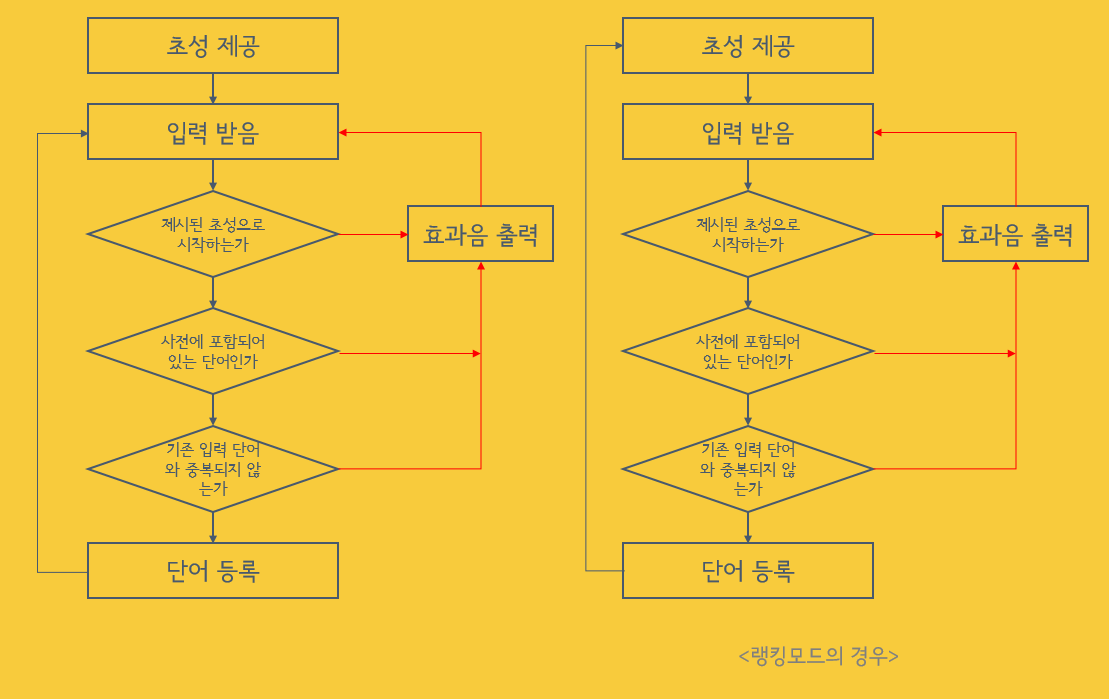
### 4.1.1 사용자 인터페이스 모듈 User Interface



* GUI 제공
* 배경음과 효과음의 On/Off기능 지원

### 4.1.2 초성게임 알고리즘 Initial Game Algorithm

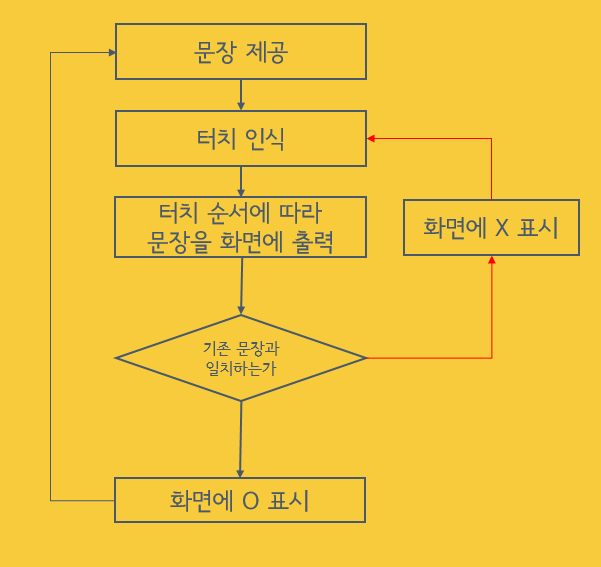
* 데이터
* 입력 단어 - string word
* 입력한 단어 수 - int wordCounting
* 배점 – int scoreCounting
* 초성 – string initialWord
* 내부DB정보
* 함수
* 난이도 조절() - SetLevel(int wordCounting,int scoreCounting)
* 랜덤 초성 제공() – PresentIntitalWord(string initialWord)
  + 제한시간 설정() - CountDownTimer(제한시간,시간간격l)
  + CheckWord(string word,string InitialWord)
  + 입력 값이 제시된 자음으로 시작하는 단어가 맞는지 확인
  + 사전에 포함된 단어인지 확인
  + 기존 입력단어와 중복되지 않는지 확인
* 추가기능
* 단어 입력 받음
* 오류가 나면 효과음 출력
* 오류가 없으면 단어 등록



(정답을 한번 입력 받을 때 마다 새로운 초성을 제공하는 랭킹모드의 경우 오른쪽 순서도 적용)

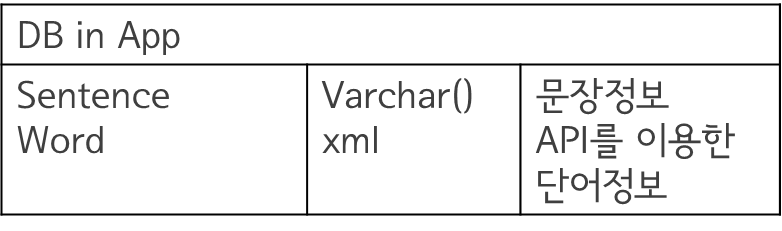
### 4.1.3 문장게임 알고리즘 Sentence Game Algorithm

* 데이터
* String separateWord
* 터치 횟수 int touchNumber
* 내부DB정보
* 함수
  + 헷갈리는 맞춤법 표현이 들어간 문장을 랜덤으로 출력() – PresentSentence()
  + 제한시간 설정() - CountDownTimer(제한시간,시간간격)
  + IsThisCorrectSentence(string separateWord,int touchNumber)
    - 버튼 터치 순서 기억
    - 내부 DB에 저장된 문장과 어순, 맞춤법이 일치하는지 확인
* 추가 기능
  + 사용자가 터치 한 순서대로 문장을 화면에 출력
  + 오답일 경우 X화면 표시
  + 정답일 경우 O화면 표시
  + 정답이거나 시간제한이 끝났을 경우 다음 문장 출력



### 4.1.4 내부 데이터베이스

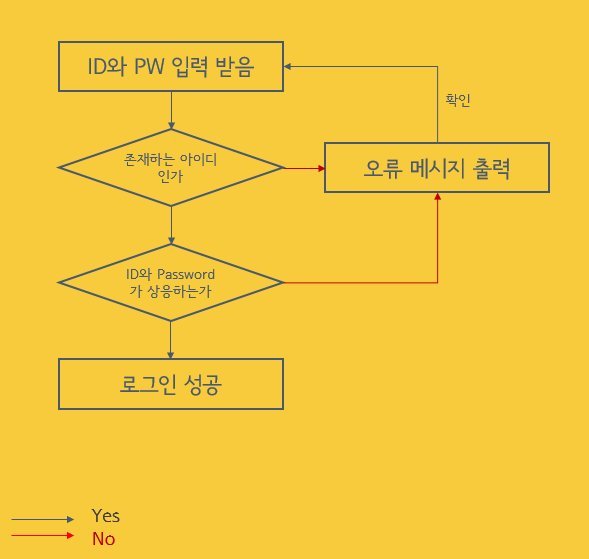
* 사전API의 XML파일을 저장
* 문장정보를 저장



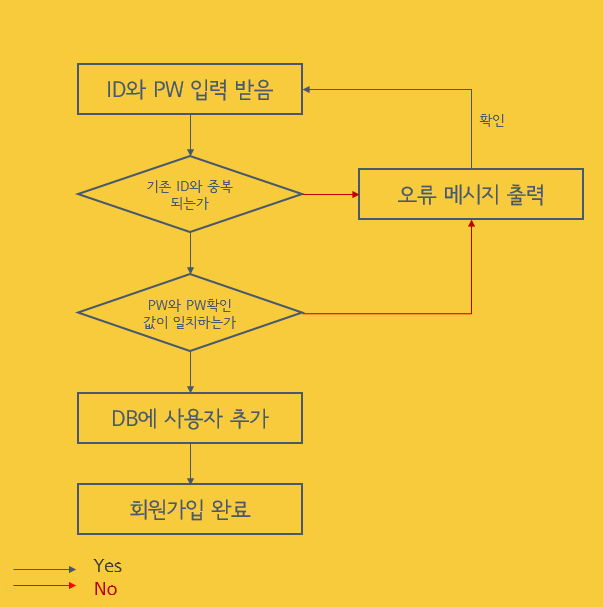
## 4.2 서버

### 4.2.1 유저 데이터 관리 Supervise User Data

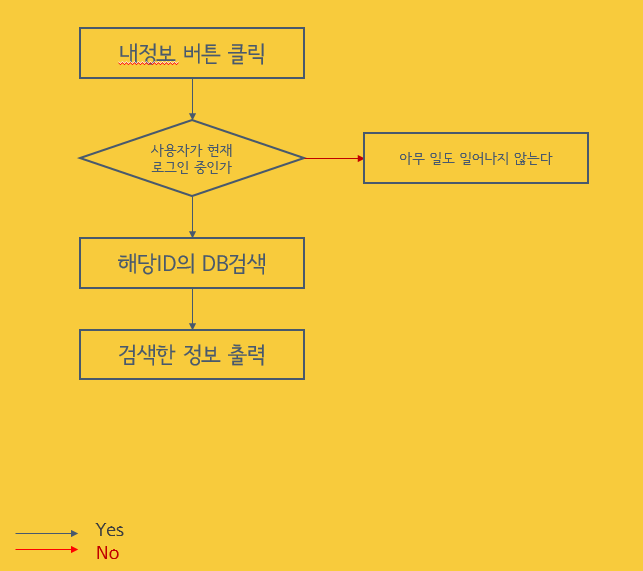
* 데이터
* 로그인 정보 String ID, String Password
* DB정보
  + 로그인 체크 Login Check 함수
* 기존에 존재하는 ID인지 확인()
* ID와 비밀번호 값이 상응하는지 확인()



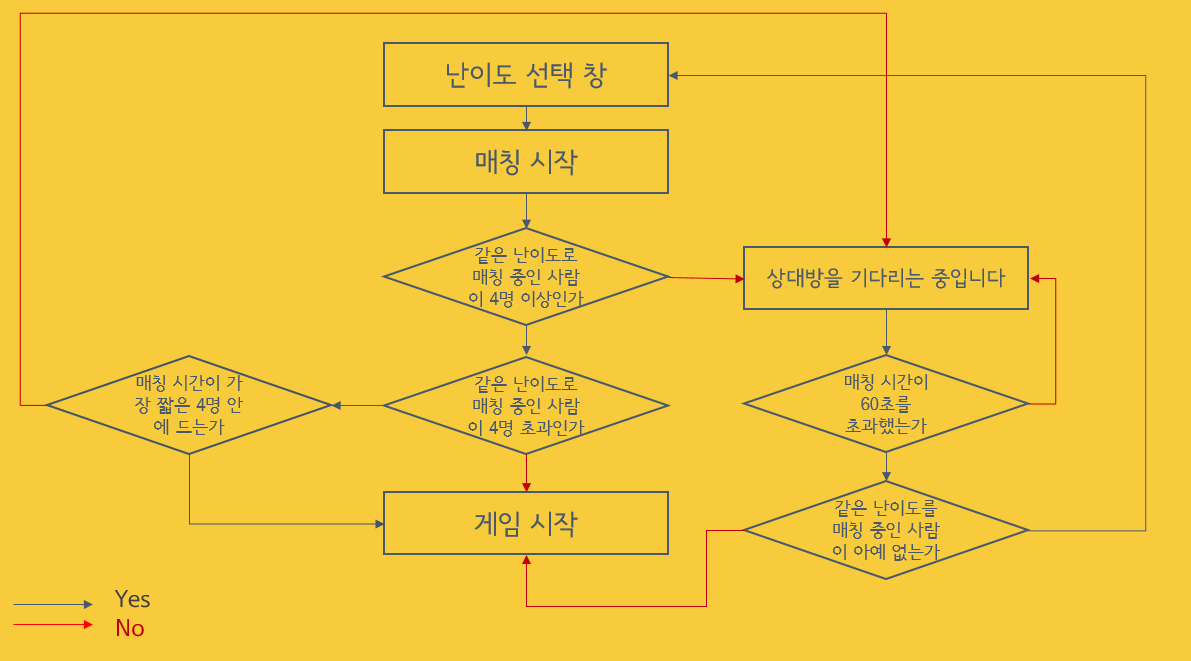
* + 회원가입 Sign Up 함수
* ID가 중복인지 확인()
* DB에 사용자 정보 추가()



* + 회원정보 출력 Print Out Data 함수
* 현재 로그인 상태인지 확인()
* 로그인 하고있는 ID의 정보를 DB에서 검색()
* 해당 랭킹 정보를 출력()



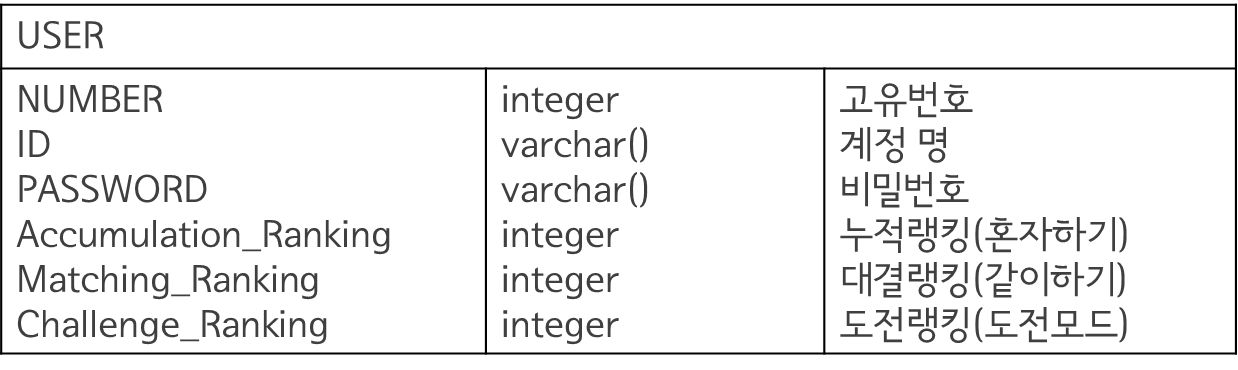
### 4.2.2 유저 매칭



* 데이터
* 매칭 유저 수 – int userNumber
* 매칭 시간 – int matchTimeCounting
* 함수
* 매칭 중인 유저 수 카운트() - CountingUserNumber()
* 매칭 소요시간 카운트() – CountingMatchingTime()
  + - 4명이상이면서 4명을 초과하지 않음 -> 결국 4명인 경우 게임 바로 시작
    - 4명보다 적을 경우 -> 상대 유저를 기다림
    - 매칭 소요시간 카운트()>60sec && 매칭 유저 카운트()>=2이면 게임 시작
    - 소요시간 카운트()를 통해 매칭 우선순위 선별

## 4.3 데이터베이스

* 유저 데이터 저장 User Data



# 5. 개발 환경 및 개발 방법

## 5.1 개발 환경

* + OS : Window
  + DB : MySQL, SQLite
  + 개발언어 : Java
  + 개발 툴 : Android Studio

## 5.2 개발 방법

1. Application
   * Android Studio를 이용한 Android app 구현
   * 안드로이드 5.0부터 6.0버전까지 구현
2. Server 및 DB
   * 서버 구축과 MySQL을 이용한 DB 구축
   * API의 XML데이터 이용
   * 회원가입 기능을 두고 회원사이의 경쟁기능 구현

# 6. 데모 환경 설계

## 6.1 사전 설정

* SERVER PC
* 스마트폰

## 6.2 데모 시나리오

* 회원가입 → 로그인 → 초성게임 다중사용자모드 시연 → 문장게임 시연 → 정보확인