



국민대학교  
전자정보통신대학  
컴퓨터공학부

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	Online algorithm battle service
팀 명	알고리즘 서비스 연구소
문서 제목	중간보고서

Version	1.2
Date	2020-04-20

팀원	최명서 (조장)
	최지욱
	박호준
	강수련
	우현웅
	칼리드
지도교수	최 준수 교수



### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "Online algorithm battle service"를 수행하는 팀 "알고리즘 서비스 연구소"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "알고리즘 서비스 연구소"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


Filename	중간보고서-"Online algorithm battle service.doc"
원안작성자	최명서
수정작성자	최지욱, 강수련, 박호준, 우현웅

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-04-18	최명서	1.0	최초 작성	
2020-04-21	우현웅	1.1	내용 추가	API 수행 내용 추가
2020-04-21	박호준	1.2	내용 추가	클라이언트 수행 내용 추가
2020-04-22	최지욱	1.3	내용 추가	클라이언트 수행 내용 추가
2020-04-23	우현웅	1.4	내용 추가	API 수정된 연구내용 및 향후 계획 추가
2020-04-23	최지욱	1.5	내용 추가	클라이언트 수정사항 추가 및 향후 추진 계획 작성
2020-04-23	최지욱	1.6	내용 수정	클라이언트 수행 내용 수정
2020-04-23	강수련	1.7	내용 추가	클라이언트 수행 내용 추가

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>		
	<b>팀 명</b>		
	Confidential Restricted	Version 1.7	2020-APR-23

## 목 차

1	프로젝트 목표 .....	4
2	수행 내용 및 중간결과 .....	4
2.1	계획서 상의 연구내용 .....	4
2.2	수행내용 .....	4
3	수정된 연구내용 및 추진 방향 .....	6
3.1	수정사항 .....	10
4	향후 추진계획 .....	10
4.1	향후 계획의 세부 내용 .....	12
5	고충 및 건의사항 .....	12

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>		
	<b>팀 명</b>		
	Confidential Restricted	Version 1.7	2020-APR-23

## 1 프로젝트 목표

본 프로젝트는 알고리즘 트레이닝에 지속가능한 동력을 제공하기 위한 웹사이트 제작을 목표로 한다. 상기 목적을 달성하기 위해 다음과 같은 기능을 갖는다.

- 유저와 유저가 한 게임을 두고 알고리즘 제출한 후에 대전하는 기능
- 유저가 대전 기록의 리플레이를 보는 기능
- 유저가 자신의 코드와 대전할 수 있는 기능
- 관리자가 문제를 쉽게 추가할 수 있는 기능

**대전 기능은 아래와 같은 방법을 통해 구현한다.**

기존에 존재하던 게임들의 규칙들을 리스트업 한다. 그리고 게임은 이런 규칙들의 조합으로 이루어진다. 대전에 앞서 게임의 규칙에 따라 유저는 코드를 작성한다. 제출한 코드는 테스트 케이스를 통해 검사하고 통과하면 대전을 진행할 수 있다. 유저의 코드는 보드판을 입력으로 받는다. 유저의 코드가 보드판에 대해 최선의 수라고 판단한 착수를 출력으로 낸다. Gamemanager가 규칙검사를 통해 알맞은 착수면 보드판을 업데이트 한다. 다른 유저의 코드는 업데이트된 보드판을 입력으로 받고 같은 루틴을 반복하여 승자를 정한다.

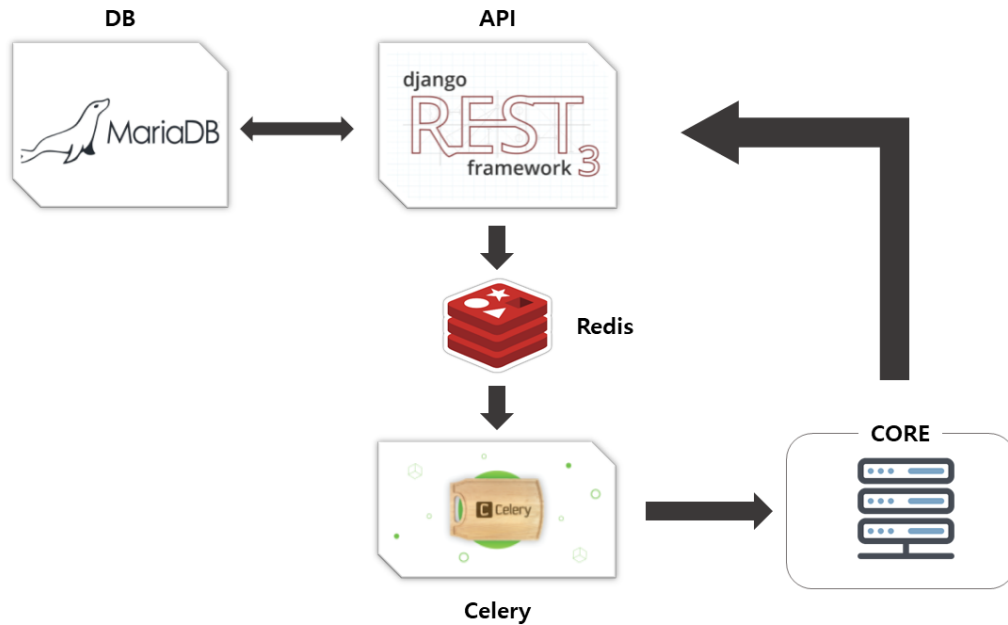
## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

#### 2.1.1 클라이언트 서버

- React/Redux를 사용해 개발 진행.
- 다음과 같은 이유로 디자인 프레임워크로는 Material UI를 채택.
  1. 공식 사이트에 잘 정리된 문서
  2. 참고 가능한 방대한 양의 데이터 존재
  3. React환경에 특화
- 리플레이 페이지에서는 생동감 있는 화면을 위해 phaser3 프레임워크를 이용.

## 2.1.2 API 서버



- Django 프레임워크 사용.
- Django Rest Framework 모듈 사용
- Celery를 사용하여 Core서버와 통신
- Celery의 브로커로 Redis 사용

## 2.1.3 Core 서버

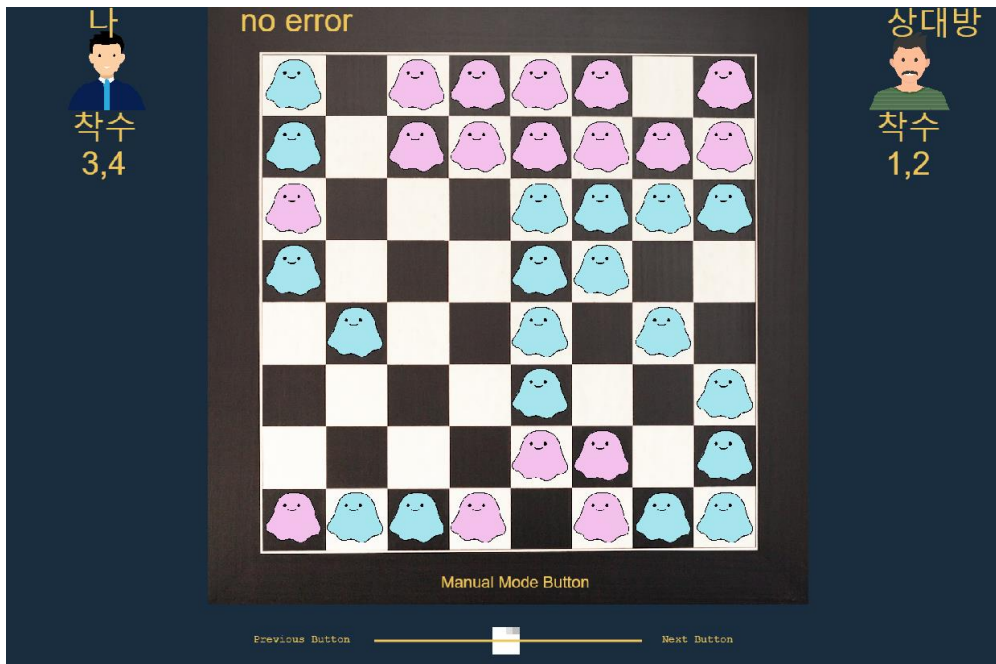
- 유저의 코드를 컴파일하고 실행하여 대전을 하는 과정이 오래 걸릴 것이라 예상되어 비 동기적으로 처리하기 위해 분리
- 실제 로직은 유저의 코드가 예상치 못한 오류를 일으켰을 때를 대비해 Docker 위에서 실행
- 유저의 코드를 보드판을 입력으로 실행. 착수를 출력으로 받아서 보드판을 갱신한 후에 상대 유저에게 같은 로직 반복.

## 2.2 수행내용

### 2.2.1 클라이언트 서버

계획서 상에서 목표로 했던 구현사항 중 유저가 자신의 알고리즘으로 타인과 대전하는 기능, 유저가 대전 기록의 리플레이를 보는 기능까지 구현한 상태이다.

현재 API, Core 서버와 연동하여 클라이언트 서버에서 구현한 부분까지 시나리오대로 진행될 수 있도록 구현하였다.




<그림 1> 리플레이 팝업창

#### - 리플레이 구현 부분

8x8 보드판에서 게임이 어떻게 진행되었는지를 확인시켜준다. 리플레이 화면을 modal 창을 통해서 보여준다. 화면 왼쪽편이 도전자이고 오른쪽 편이 도전을 받는 유저이다. 유저 본인이 도전자일 때는 파랑색 계열의 돌을 사용하고, 도전을 받는 입장이면 핑크색 계열의 말을 사용한다.

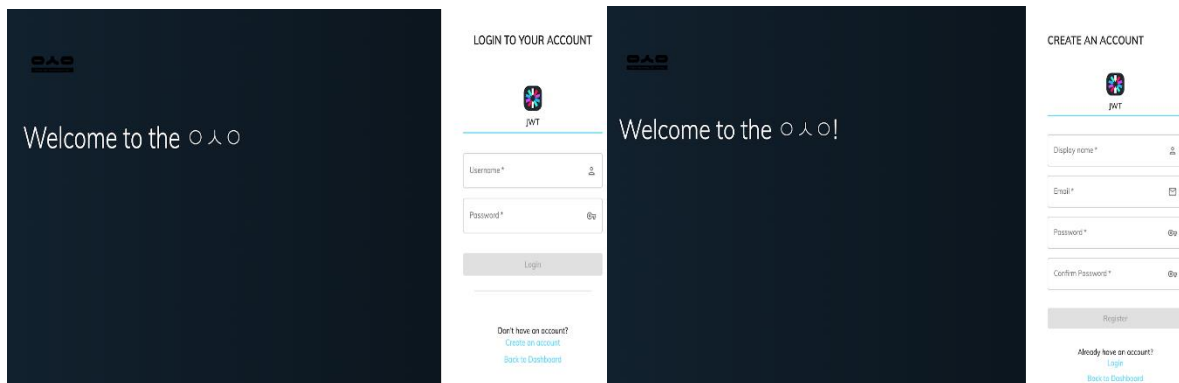
리플레이 게임 정보를 가져오면 게임에 대한 정보를 렌더링 할 수 있다. 자동으로 게임을 보여주는 Auto mode 와 수동으로 게임을 한수 씩 볼 수 있는 Manual mode 를 버튼 클릭으로 전환해볼 수 있다.

Auto mode 인 경우 0.5 초마다 한 수씩 착수를 하는 모습을 볼 수 있으며, 본인 혹은 상대방이 어느 위치에 착수를 하였는지 게임판에서 보여주고 텍스트로도 알려준다. 또한 이동에 대한 수를 보여줄 때는 착수가 아닌 이동으로 텍스트를 보여주고 출발 위치와 도착 위치를 텍스트로 알려준다.

 <div> <b>국민대학교</b>  <b>컴퓨터공학부</b>  <b>캡스톤 디자인 I</b> </div>	<b>중간보고서</b>		
	<b>프로젝트 명</b>		
	<b>팀 명</b>		
	Confidential Restricted	Version 1.7	2020-APR-23

Manual mode 인 경우 슬라이드 바의 버튼과 next, previous 버튼이 활성화가 된다. 0.5 초에 한 수씩 두는 것이 느리다고 판단이 드는 경우 혹은 이전 액션 결과를 재확인 해보고 싶은 경우 등 수동으로 액션 결과를 확인할 수 있는 모드이다. next 와 previous 버튼은 클릭 한 번에 액션 한번의 결과를 보여주고, 슬라이드 바는 마우스로 드래그 하여 빠르게 보드판의 상황을 확인할 수 있다. Manual mode 일 때도 마찬가지로 현 액션의 상황을 텍스트로 확인할 수 있다.

## - 회원가입 및 로그인 구현 부분




### <그림2> 로그인 / 회원가입 화면

회원가입 화면은 로그인시 사용될 Display name과 Email, Password와 Password를 검사할 Confirm Password로 구성되어 있다. 이메일 형식이 올바르지 않은 경우 회원가입 버튼이 비활성화 되며, 이메일 형식은 올바르나 이미 가입된 이메일의 경우 회원가입 버튼을 클릭했을 때 '이미 이 이메일 주소로 등록된 사용자가 있습니다.'의 경고창이 뜨고 가입이 진행되지 않는다. Password를 다시 한 번 확인하기 위해 Password와 Confirm Password를 거쳐 가입을 진행하는데, 두 비밀번호가 같지 않으면 회원가입 버튼이 비활성화되고, 두 비밀번호가 동일하더라도 비밀번호가 너무 짧은 경우는 '비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.'의 경고를 띄우고, 비밀번호가 모두 숫자로 이루어진 경우에는 '비밀번호가 전부 숫자로 되어 있습니다.' 등의 경고창과 함께 가입이 진행되지 않는다.

올바른 회원가입 진행 시 회원가입 완료와 동시에 자동로그인이 진행된다. 로그인은 JWT 방식을 이용하여 로그인한 회원에게 signed 토큰을 지급한다. 유저는 서버에 요청이 필요할 때 localStorage에 저장해둔 토큰을 함께 서버에 전달한다. 토큰의 유효기간은 2일로, 2일후에 자동으로 무효화된다. 이후에 토큰의 유효기간을 줄일 예정이다.

로그인 화면은 회원가입시 입력한 Displayname과 Password를 입력으로 받고, 기존 가입된 정보와 일치하지 않을 때 'ID 나 Password를 확인하세요.'의 알림창이 뜬다.

	중간보고서		
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	프로젝트 명	
		팀 명	
		Confidential Restricted	Version 1.7 2020-APR-23

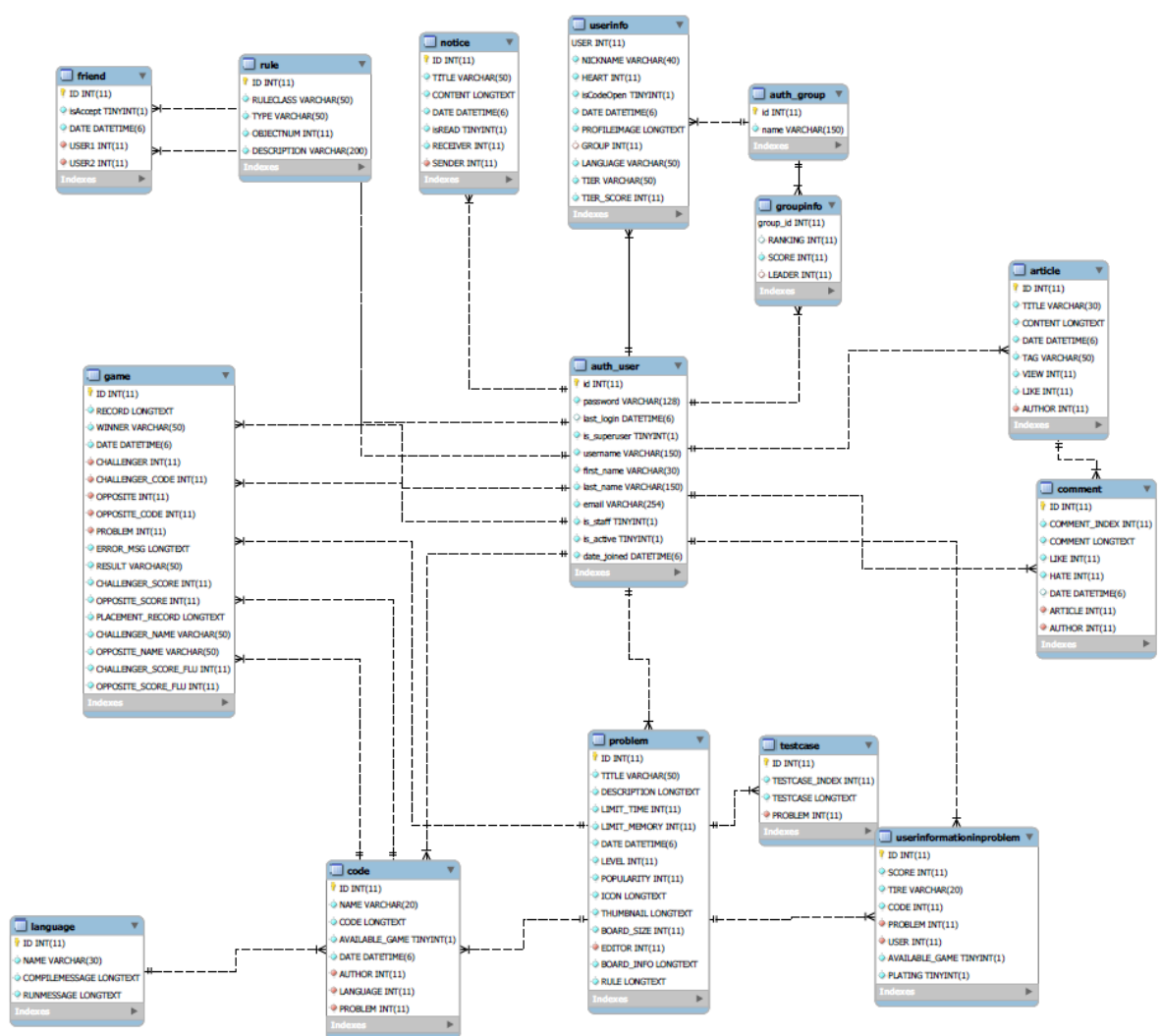
정확한 정보로 로그인 했을 때에는 회원가입 후 자동로그인의 경우와 같이 토큰을 발급받는다.

## 2.2.2 API 서버


계획서에서 목표하였던 구현 사항들은 모두 구현했으나, 테스트 하는 도중 발견되는 오류를 수정하고 있다. 추가적으로 팀원들과의 소통에서 추가해줬으면 하는 기능들 또한 추가 중이다.

설계된 DB 를 바탕으로 ORM 모델들을 구현하였고, 이를 통해 기본적으로 CRUD 역할을 수행할 수 있는 API 서버를 구현하였다.

DB 의 설계 내용은 다음과 같다.





	중간보고서		
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	프로젝트 명	
		팀 명	
		Confidential Restricted	Version 1.7 2020-APR-23

user와 group 모델은 django에 존재하는 기본 모델을 사용하였고, 나머지 테이블은 직접 설계 및 구현하였다.

설계된 DB를 바탕으로 django의 ORM 모델을 구현하고, 이를 통해 기본적으로 CRUD 역할을 수행할 수 있는 API 서버를 구현하였다.

로그인 부분은 rest-auth 모듈의 JWT 토큰을 사용하였고, 이를 통해 앞으로의 request에서 권한과 인증 부분을 검사한다.

그 후 추가적으로 매칭 알고리즘과 점수 변동 알고리즘을 구현하였고, 게임 종료시 유저들의 점수, 등급 업데이트 기능과 그룹 랭킹 업데이트 기능을 구현하였다.

### 2.2.3 Core 서버

계획서 상의 진도는 대전기능과 테스트케이스 검사 기능을 개발 완료했어야 했지만 테스트케이스 검사 기능은 아직 구현하지 못했다.

Api서버가 Celery를 통해 task를 redis에 던지면 Core서버에서 Celery를 통해 해당 task를 받아 docker container를 실행한다. Container상에서 다음의 매칭 로직이 실행된다.

1. 보드판을 input으로 유저의 코드를 실행
2. output으로 착수 값을 받아 착수규칙에 맞는지 확인.
3. 착수 규칙에 맞다면 보드판을 갱신 후 액션규칙 실행. 액션규칙이 없으면 pass
4. 엔딩규칙을 통해 게임이 끝났는지 검사.
5. 갱신된 보드판을 상대 유저에게 input으로 넣어주고 위의 로직 반복

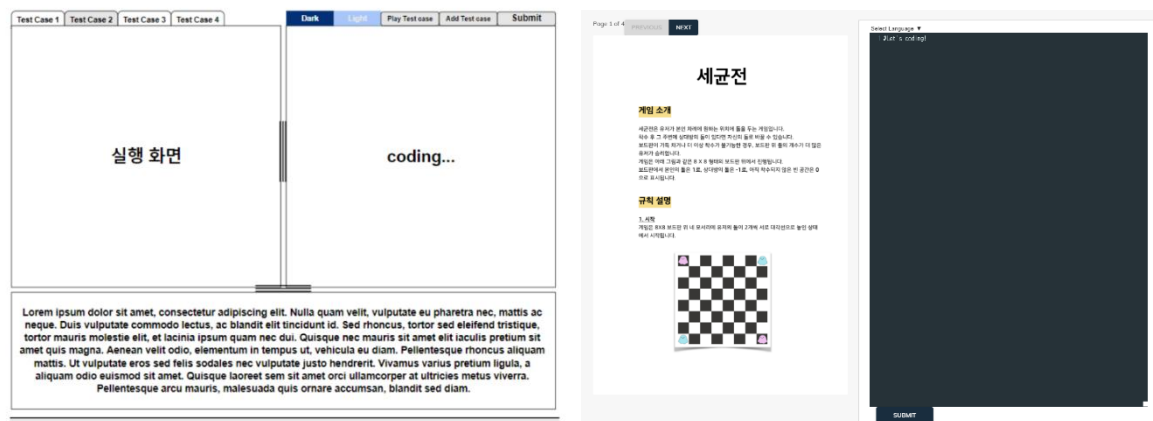
### 2.2.4 테스트

### 3 수정된 연구내용 및 추진 방향

#### 3.1 수정사항

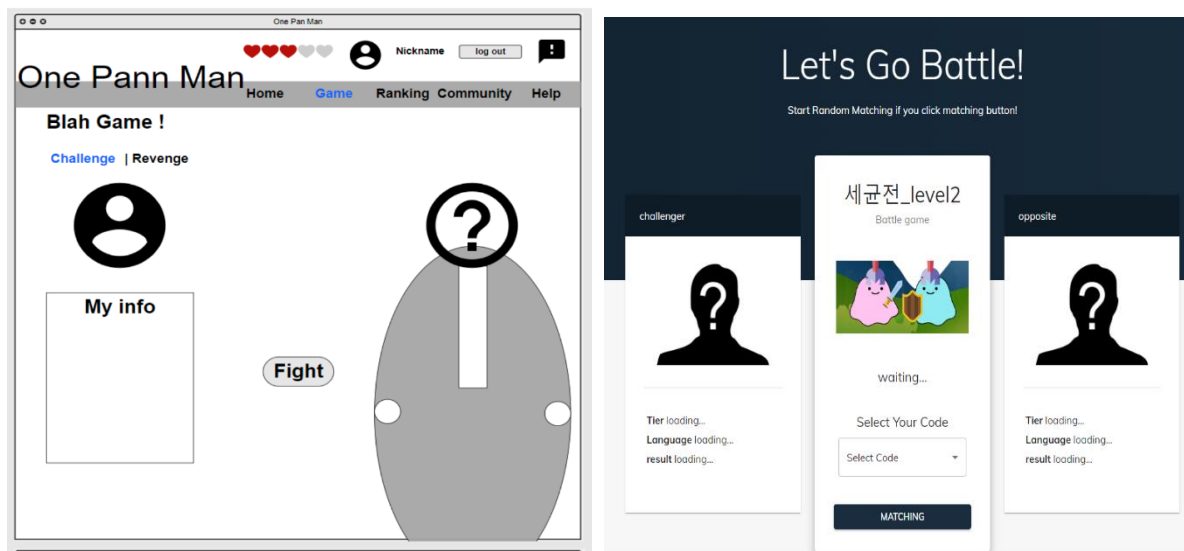
##### 3.1.1 클라이언트 서버

수행계획에서 예상했던 클라이언트 서버의 UI는 다음과 같이 구현되었다.



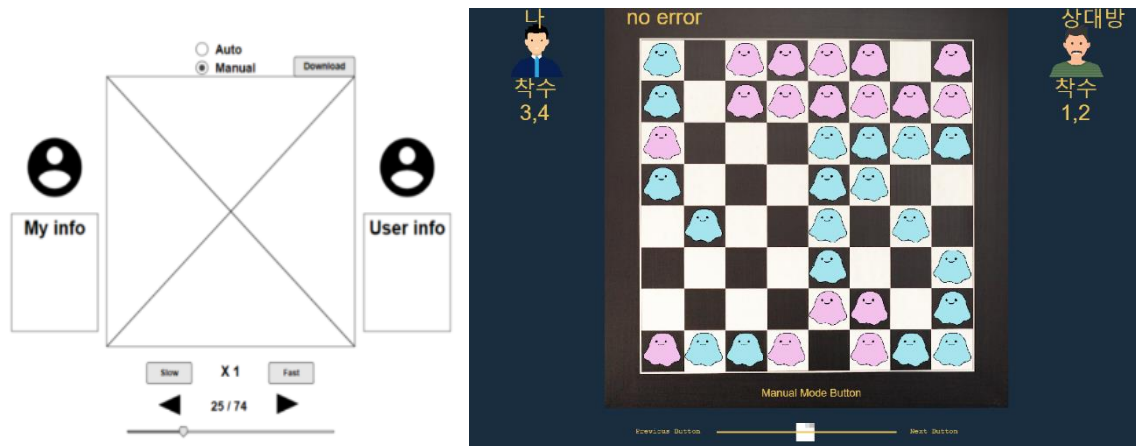
<그림 2> 코드 제출 화면

이전 기획은 왼쪽 사진과 같이 실행화면 UI를 제공할 계획이었으나 기획 회의 중 문제를 읽고 코드를 작성하는 것이 맞다고 판단하여 오른쪽 화면과 같이 문제를 읽고, 코드를 제출할 수 있도록 오른쪽 그림과 같이 UI를 구성하였다.



### <그림 3> 대전 매칭 화면

이전 기획은 왼쪽 사진과 같이 기획하려 했으나 기획 회의 중 게임에 해당하는 코드 선택과 같은 세부적인 구성이 없었기 때문에 오른쪽 사진과 같이 코드선택, 대전 게임 정보 등 더 세부적인 UI를 보완하여 구성하였다.



### <그림 4> 리플레이 화면

이전 기획은 리플레이 기록을 그림 파일로 다운로드 할 수 있도록 기획하였으나 구현 중 실제 리플레이 화면이 생각보다 많은 과정이 있어 사용자가 기록을 다운로드 받아도 제대로 읽지 못하는 상황이 발생할 것이라고 생각하여 다운로드 기능을 추가하지 않았다.

Auto mode 와 Manual mode 버튼을 하단으로 배치하여 슬라이더 바와 함께 사용이 편리하도록 배치를 바꿨다.


리플레이 화면에서 에러로 인해 제대로 게임이 진행되지 않는 경우에, 상단에 에러 메시지를 출력한다. 에러가 없이 잘 수행되었으면 no error 라는 메시지를 출력해준다.

## 3.1.2 API 서버

API 의 기능으로 데이터의 기본적인 CRUD 만을 생각했었는데, 클라이언트의 요구에 따라 여러 데이터를 JOIN 을 이용해서 묶어 보내줄 필요성이 생겼다.

따라서 기존 CRUD api 뿐만 아니라 상황에 맞게 데이터를 묶어 보내는 api 를 제작했다. 예를 들어 user 모델과 userInfo 모델을 함께 묶어 보내주는 userfullInfo API 를 제작하였다.

후에도 필요성에 따라 추가적으로 API 를 구현할 계획이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.7	2020-APR-23

### 3.1.3 Core 서버

수정사항 없음

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### 4.1.1 클라이언트 서버

이동과 착수 모션에 대한 이펙트를 추가할 것이다.

현재 리플레이 화면은 실제로 보드판에서 게임을 하는 듯한 생동감을 줄 수 없다. 이를 보완하기 위해서는 착수와 이동 모션에서 이펙트를 추가하는 방법을 고안하였다. 또한 보드판 위의 말의 종류가 다양해짐에 따라 추가 작업을 해야한다.

유저의 모습을 렌더링 할 때 유저가 올린 프로필 사진으로 렌더링 할 것이다.

추후에 유저 정보를 수정하는 페이지가 생기면 유저 프로필 사진을 등록할 수 있게 될 것이다. 그에 따라 리플레이 화면에서 유저의 프로필등 유저 정보를 리플레이 화면에 렌더링 할 것이다.

슬라이더 바에 보드판의 상황이 대략적으로나마 나올 수 있도록 할 것이다.

기존에 계획한 슬라이더 바는 해당 액션에 슬라이더 바가 찾아갈 수 있는 형태였다. 그래서 슬라이더 바에서 보드판의 상황을 대략적으로 알 수 있어서 유저가 원하는 액션에 손쉽게 갈 수 있도록 보완할것이다.


Auto mode 시 배속 기능을 추가할 것이다.

기획한 것에서는 Auto mode 시 배속 기능이 있다. 수동으로 빠른 탐색이 가능하지만 Auto mode 에서는 조작할 필요가 없는 편리함 때문에 Auto mode 를 사용하는 유저가 있을 것이다. 따라서 편리함을 증대시키고자 배속 모드를 추가할 것이다.

텍스트와 이미지의 정렬을 맞추어 줄 것이다. 현재 정렬이 말끔하지 못하다. 이미지와 텍스트의 정렬을 맞춰서 깔끔한 UI 를 보여줄 것이다.

현재 구현되어 있지 않은 유저 정보 페이지, 코드 리스트 페이지, 그룹 기능, 랭킹 기능을 구현할 예정이다.

유저 정보 페이지에서는 닉네임과 프로필 사진 및 대표 언어를 설정할 수 있다. 그리고 자신이 제출한 코드에 대해 공개 여부를 설정할 수 있다. 코드 리스트 페이지에서는 제출된 코드들을 확인할 수 있다. 전체 코드 리스트에서는 코드 공개를 허용한 유저들의 코드와 내 코드가 같이 나오고, 내 코드 리스트에서는 내가 제출한 코드들만 나온다. 각 코드는 유저가 설정한 코드 이름으로 구분할 수 있다. 전체 코드 리스트에서는 유저이름\_코드네임으로 구성된다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>		
	<b>팀 명</b>		
	Confidential Restricted	Version 1.7	2020-APR-23

그룹 페이지는 그룹 리스트를 보여주고, 그룹에 가입된 유저 수를 표시해 준다. 랭킹 기능에서는 문제별 유저랭킹, 전체 유저랭킹, 그룹 랭킹으로 구성될 예정이다. 랭킹 리스트에서는 등수, 이름, 점수가 표시될 예정이다.

핵심 기능 중 하나인 내 코드와 대전하기 페이지를 구현할 예정이다. 코어 서버와 웹 소켓을 열어 내 코드와 실시간 대전을 할 수 있도록 구현할 것이다.

마지막으로 문제 추가 페이지를 구현할 예정이다. 문제 추가 페이지는 각 문제의 규칙들을 불러온다. 이 규칙들은 착수, 액션, 엔딩 규칙으로 나누어져 있다. 해당 규칙들을 서로 위반되지 않는 선에서 조합하여 새로운 게임을 만들 수 있게 한다.

#### 4.1.2 API 서버

테스트 케이스 검사 기능을 구현할 예정이다. 유저의 코드가 게임의 기본적인 진행 조건을 충족하는지 테스트한다. 해당 테스트 케이스를 통과한 코드는 게임 가능 코드가 되며, 게임 진행에 사용되는 `userInformationInProblem` 모델이 자동 생성된다. 게임 진행과 마찬가지로 `celery` 를 이용해서 코어서버와 통신을 통해 해당 코드가 테스트케이스를 통과하는지 검사한다.

또한 유저 프로필 사진, 문제 썸네일 사진 등을 업로드하고, 저장하는 기능도 구현할 예정이다.

#### 4.1.3 Core 서버

- 진입장벽이 높을 것을 생각해 쉬운 난이도의 게임 추가를 계획하고 있다.
- 현재 유저가 코드 제출시에 실행 가능한 코드인지, 착수 규칙에 맞는 착수를 출력으로 내는지 검사하는 기능인 테스트케이스 기능을 구현하고 있다.
- 유저가 자신이 낸 알고리즘과 대결하는 기능을 계획하고 있다. 이 부분에서 Core서버의 리소스 사용량을 어떻게 조절할지 고민중이다.
- 테스트 중에 보안 이슈가 발생하였다. 유저가 악의적으로 코드에 서버의 시스템을 건드리는 코드를 작성할 경우 게임이 원활히 진행되지 않는다. 코드 검사를 통해 따라서 시스템에 관련된 모듈을 사용하지 못하게 할 예정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명		
	팀 명		
	Confidential Restricted	Version 1.7	2020-APR-23

## 5 고충 및 건의사항

코로나로 인해 오프라인 만남이 불가능한 상황이라 프로젝트 진행이 매우  
힘듭니다.