



PROJECT 2. 암기 학습을 위한 암기용



소개

- 학습관리와 게임을 더한 암기 웹 & 앱

기간 & 인원

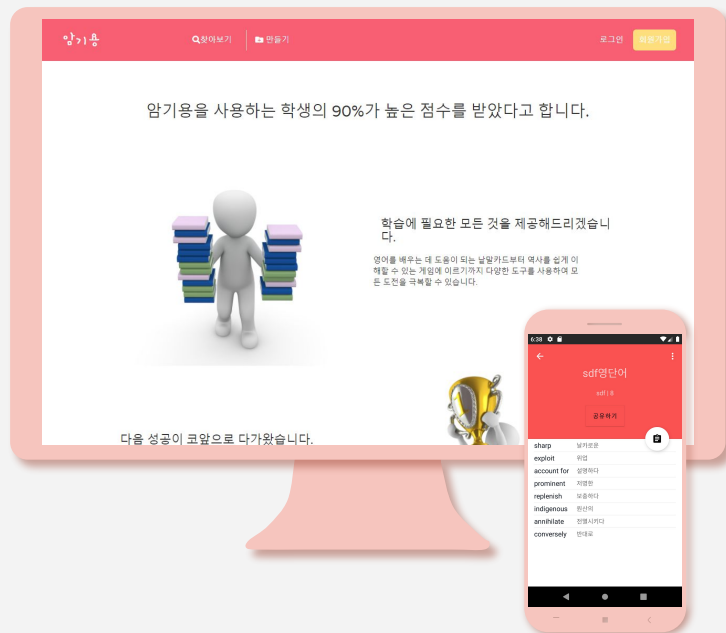
- 7.13 ~ 10.29, 4인

개발환경 & 사용기술

- JAVA, JavaScript, jQuery, JSP, CSS
- Spring Framework, MyBatis, Bootstrap
- Oracle
- Apache Tomcat v9.0
- 형상관리 : GitHub

Git 주소

- <https://github.com/chuseok/Final.git>



소개

전체기능

상세기능

문제해결

느낀점

PROJECT 2. 암기 학습을 위한 암기용

기획

- kakao oven을 사용해 모형 제작



소개

전체기능

상세기능

문제해결

느낀점

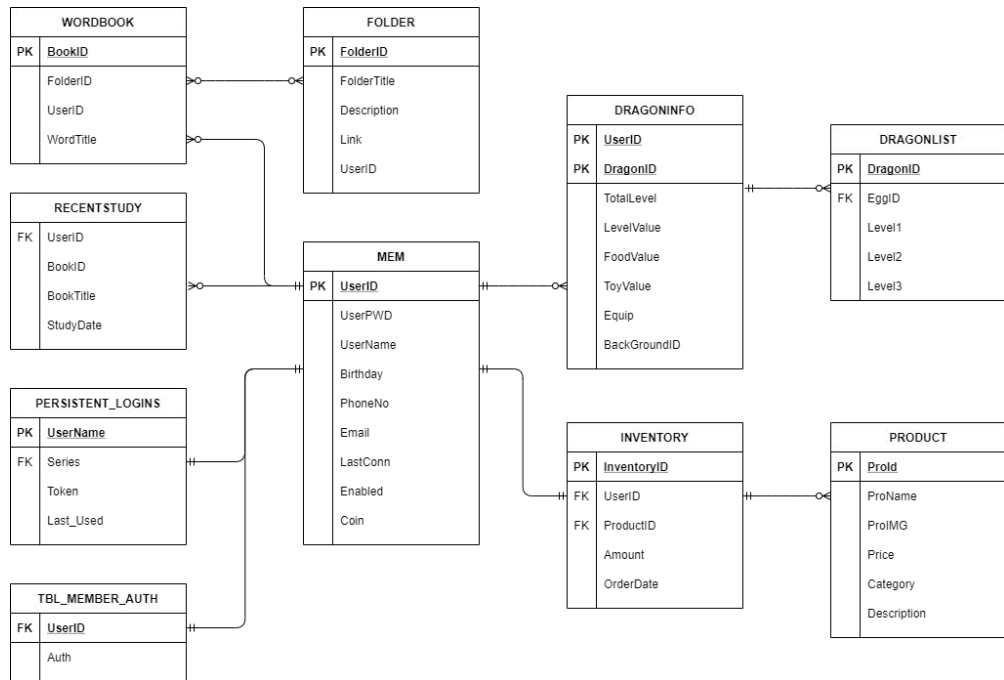


PROJECT

2. 영어 학습을 위한 암기용



DB & Directory



```
src/main/java
├── com.test.tst
├── org.android.controller
├── org.dragon.android.controller
├── org.dragon.controller.game
├── org.dragon.controller.login
├── org.dragon.controller.study
├── org.dragon.controller.word
├── org.dragon.domain.game
├── org.dragon.domain.login
├── org.dragon.domain.study
├── org.dragon.domain.word
├── org.dragon.listener.game
├── org.dragon.mapper.game
├── org.dragon.mapper.login
├── org.dragon.mapper.study
├── org.dragon.mapper.word
├── org.dragon.security
├── org.dragon.security.domain
├── org.dragon.service.game
├── org.dragon.service.login
├── org.dragon.service.study
├── org.dragon.service.word
├── org.dragon.web.controller
└── src/main/resources
    ├── org
    │   ├── dragon
    │   │   ├── mapper
    │   │   │   ├── game
    │   │   │   ├── login
    │   │   │   ├── study
    │   │   │   └── word
```

```
src
├── main
│   └── webapp
│       ├── resources
│       │   ├── css
│       │   ├── dist
│       │   ├── images
│       │   ├── img
│       │   ├── jquery
│       │   ├── js
│       │   └── vendor
│       ├── WEB-INF
│       │   ├── spring
│       │   └── views
│       │       ├── dragon
│       │       ├── folder
│       │       ├── includes
│       │       ├── learn
│       │       ├── main
│       │       ├── sample
│       │       ├── shop
│       │       └── word
│       ├── accessError.jsp
│       ├── home.jsp
│       ├── main.jsp
│       ├── naverSuccess.jsp
│       ├── search.jsp
│       └── withdrawal.jsp
└── web.xml
```



소개

전체기능

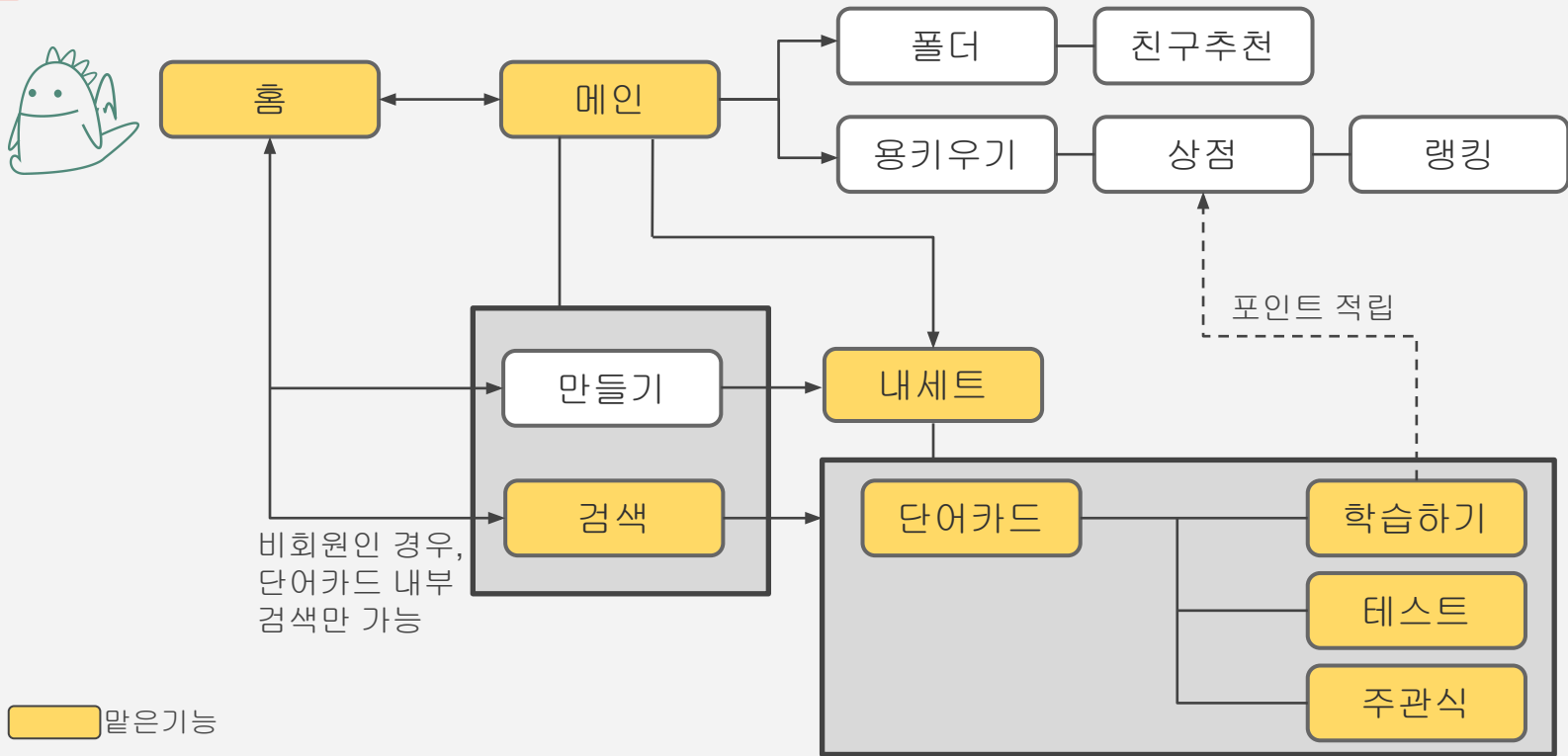
상세기능

문제해결

느낀점

PROJECT 2. 영어 학습을 위한 암기용

전체구성 - 웹



소개

전체기능

상세기능

문제해결

느낀점

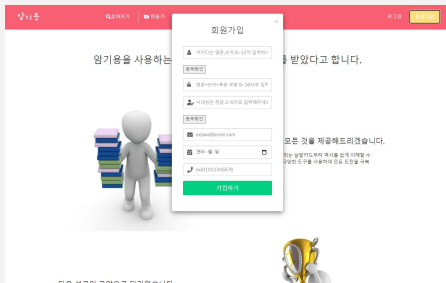


PROJECT

2. 영어 학습을 위한 암기용



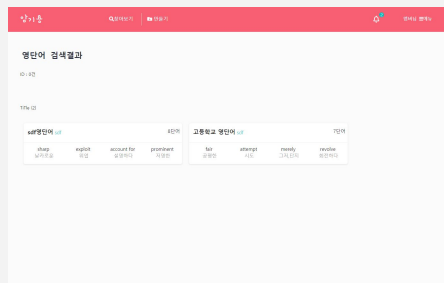
전체 화면구성 - 웹



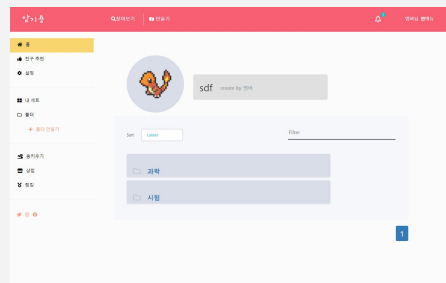
회원가입



메인화면



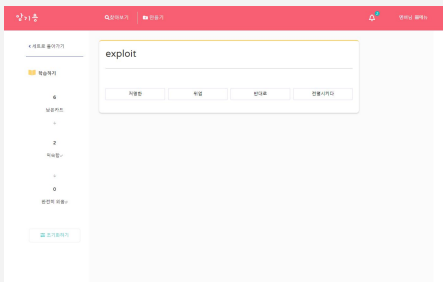
검색



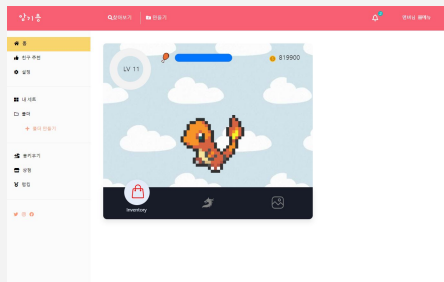
폴더



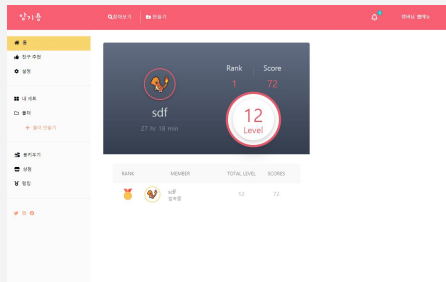
내 세트



학습하기



용키우기



랭킹



소개

전체기능

상세기능

문제해결

느낀점

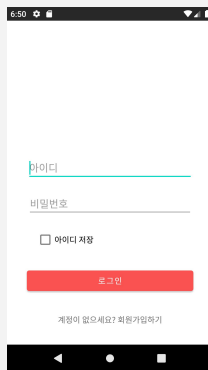
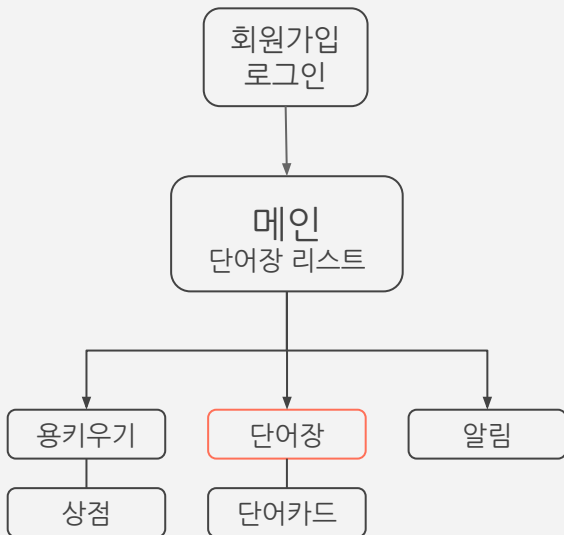


PROJECT

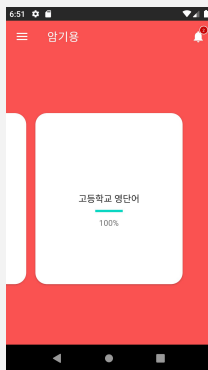
2. 영어 학습을 위한 암기용



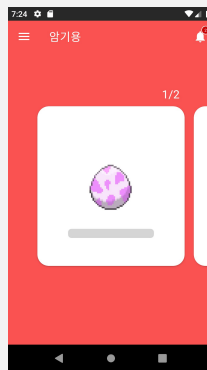
전체구성 - 앱



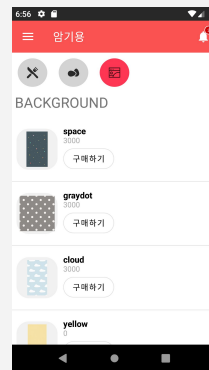
<로그인>



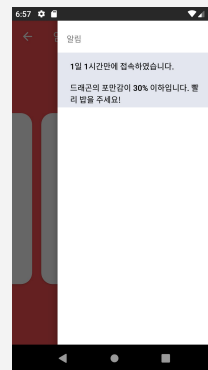
<단어장리스트>



<용키우기>



<상점>



<알림>



소개

전체기능

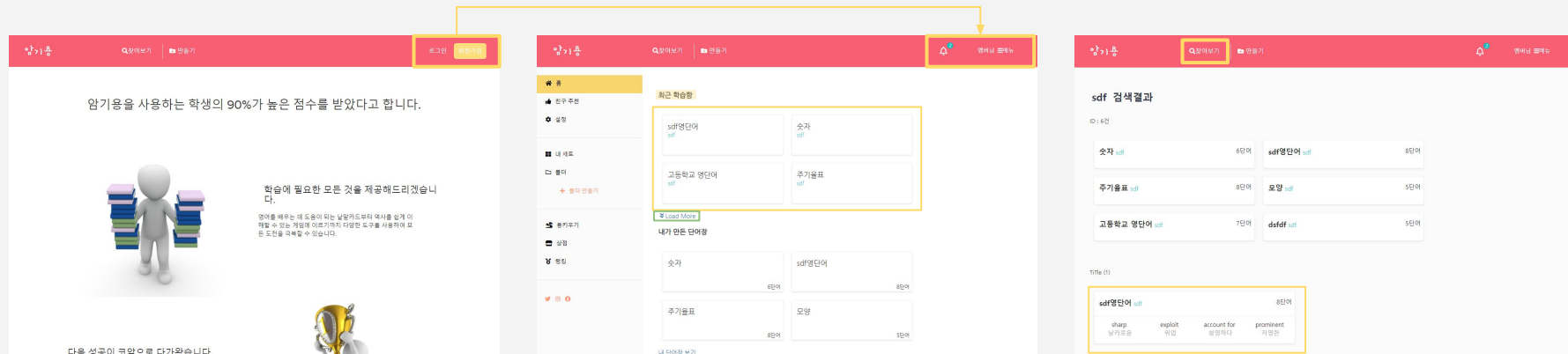
상세기능

문제해결

느낀점

PROJECT 2. 영어 학습을 위한 암기용

상세기능 - 메인



- Spring Security로 권한에 따라 보여지는 홈화면(비로그인)과 메인화면(로그인)을 구분
- 메인화면에서 최근학습 단어장들과 내가 만든 단어장 확인 가능
- 최근학습은 최근에 학습한 단어장이 최상단으로 노출되며, 기본 4개 출력.
Load More버튼으로 4개씩 더 확인할 수 있도록 구현

- 클릭 시 나오는 검색창으로 검색
- ID와 TITLE로 검색 결과 분류
- TITLE 검색결과는 단어장의 일부 단어를 보여주어, 학습할 단어장을 미리 보고 선택할 수 있도록 구현



소개

전체기능

상세기능

문제해결

느낀점

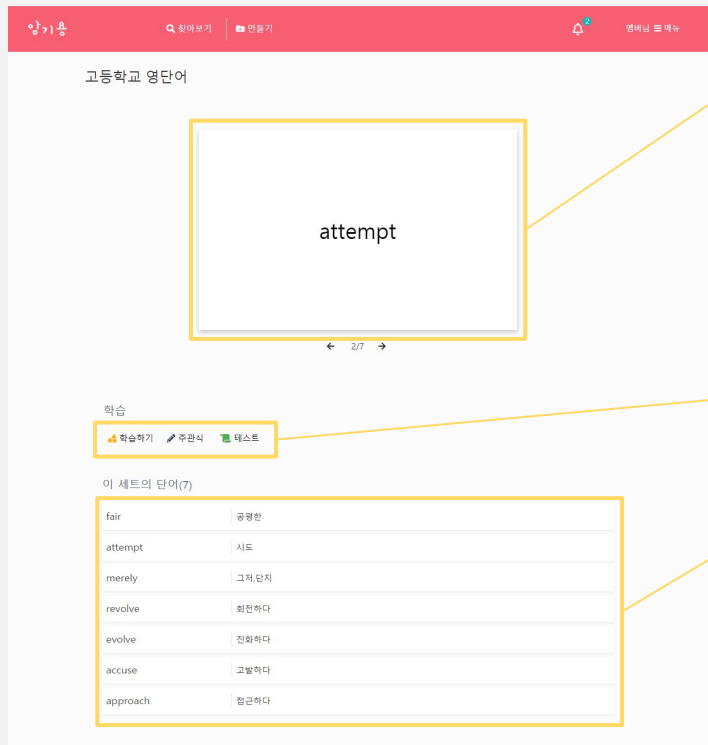


PROJECT

2. 영어 학습을 위한 암기용



상세기능 - 단어카드



카드 클릭 시, 카드가 뒤집어지며 뜻 확인

```
//card click - flip animation
var card = document.querySelector('.flip-card-inner');
card.addEventListener('click', function()
{
    card.classList.toggle('is-flipped');
});
```

```
.flip-card-inner.is-flipped {
    transform: rotateY(-180deg);
}
```

현재 열린 단어장의 학습하기, 주관식, 테스트로 이동하는 링크메뉴

단어장의 단어들을 모두 출력

```
<c:forEach items="${list.item}" var="item">
  <div class="oneOfWords">
    <div class="item-word"><c:out value='${item.word}' /></div>
    <div class="item-meaning"><c:out value='${item.meaning}' /></div>
  </div>
</c:forEach>
```



소개

전체기능

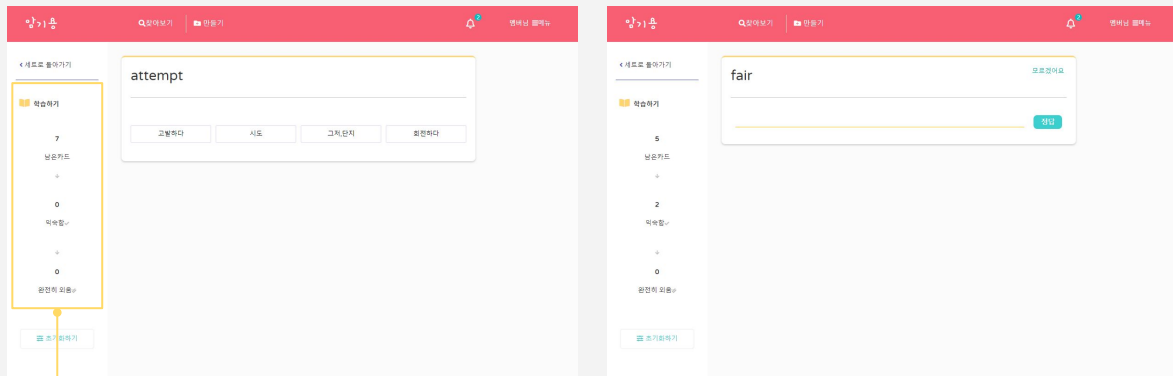
상세기능

문제해결

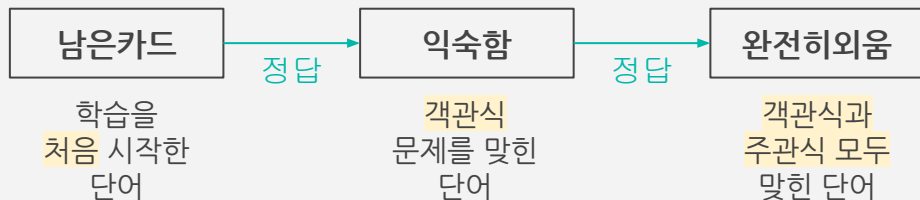
느낀점

PROJECT 2. 영어 학습을 위한 암기용

상세기능 - 학습하기



남은카드 / 익숙함 / 완전히 외움 세가지로 학습률에 따라 분류



- 객관식, 주관식 두가지로 출제.
- 모든 단어는 객관식이 먼저 출제되며, 객관식을 맞춘 단어만 주관식으로 출제
- 학습중인 과정이 동적으로 저장되어 종료 후 재접속 시 이어서 학습 가능



소개

전체기능

상세기능

문제해결

느낀점



PROJECT

2. 영어 학습을 위한 암기용



상세기능 - 학습하기

객관식

주관식

- 객관식과 주관식이 **조건**에 따라 최대 5번씩 번갈아가며 출제

=> 지루함을 줄이기 위한 선택

```
//문제출력 control
function selectQuestion()
{
  showProgress();
  if(lastCardLen == 0 && knownCardLen == 0){//더 이상 낼 문제가 없을경우
    showFinalResult();
  }
  else{
    if(isMatched()){//객관식 출력
      showMultipleQuestion();
    }
    else{//주관식 출력
      showSubjQuestion();
    }
  }
};
```

```
function isMatched()
{
  if(lastCardLen == 0){ return false; }
  if(multiCount == 5){ multiCount =0; return false; }
  if(subjCount>=1 && subjCount <5){ return false; }
  if(subjCount == 6){subjCount =0;}

  return true;
};
```

- 객관식 or 주관식을 결정하는 과정

: 연속으로 5번이상 출제되었거나
문제로 낼 단어가 없는경우인지 체크

```
if(rate == 1){
  moveWord(word, meaning, KnownCardArray);
}
else if(rate >=2){
  moveWord(word, meaning, WellKnownCardArray);
}
else{
  moveWord(word, meaning, LastCardArray);
}
```

데이터를 불러올 때 **학습률**에 따라 분류
=> 이전학습을 이어서 학습가능



소개

전체기능

상세기능

문제해결

느낀점



PROJECT

2. 영어 학습을 위한 암기용



상세기능 - 학습하기

암기용

로그인하기

회원가입

스터디로 돌아가기

학습하기

5

남은카드

2

익숙함

0

완전히 외움

스터디로 돌아가기

스터디가 필요합니다!

문제

accuse

답변하신 답

그치, 단지

정답

고발하다

정답 확인하기

- 오답 시,
입력한 답과 정답을 볼 수 있는 오답
피드백 화면 출력

문제 / 내가 입력한 답 / 정답 표시

```
//학습률 올리기
function upRate(answer){
    var obj = new Object();
    var wordId = WordJsonArray[0]["id"];
    var wordTitle = WordJsonArray[0]["title"];
    var upWord = answer;
    obj.id = wordId;
    obj.title = wordTitle;
    obj.word = upWord;

    var jsonData = JSON.stringify(obj);

    $.ajax({
        type: "POST",
        dataType: "json",
        url: "/study/uprate",
        contentType: 'application/json; charset=utf-8',
        data: jsonData
    });
};
```

- 정답 시,
Ajax를 통해 해당 단어의 학습률
상승.

암기용

로그인하기

회원가입

스터디로 돌아가기

학습하기

0

남은카드

0

익숙함

7

완전히 외움

스터디로 돌아가기

fair

attempt

evolve

manely

accuse

revolve

approach

공정한

시도

진화하다

그치, 단지

고발하다

회전하다

접근하다

처음부터 다시하기

- 학습이 끝나면 모든 단어를 출력하여
복습 유도

- 처음부터 다시하기 버튼으로
초기화하여 학습 가능



소개

전체기능

상세기능

문제해결

느낀점



PROJECT

2. 영어 학습을 위한 암기용



상세기능 - 테스트

암기용 로그아웃 회원가입 로그인 회원정보

4. 테스트 보기

테스트

주관식 문제 3개

1. 공평한
2. 진화하다
3. 시도

객관식 문제 2개

1. revolve
 - ☐ 회전하다
 - ☐ 접근하다
 - ☐ 그쳐, 단지
 - ☐ 시도
2. approach
 - ☐ 회전하다
 - ☐ 접근하다
 - ☐ 그쳐, 단지
 - ☐ 시도

단답형 문제 2개

1. merely → 그쳐, 단지
 - ☐ 오답
 - ☐ 옳음
2. accuse → 고발하다
 - ☐ 오답
 - ☐ 옳음

정답 확인



정답확인

1. 회전하다

정답

✓ revolve

2. 접근하다

오답

✗ approach

정답

approach

1. merely

☐ 고발하다

☐ 시도

☐ 공평한

☒ 그쳐, 단지 ✓

2. attempt

☒ 진화하다 ✗

☐ 그쳐, 단지

☐ 접근하다

☐ 시도 ✓

1. evolve → 진화하다

정답

✓ true

2. accuse → 시도

오답

✗ true

정답

false

정답 : 고발하다

테스트

점수

57%

- 학습했던 단어들을 객관식, 주관식, 단답형으로 종합 평가

```
function showScore()
{
  var allQuestionNum = 0;
  var score = 0;
  allQuestionNum = correctNum + incorrectNum;

  score = (Math.floor((correctNum / allQuestionNum) * 100)) + "%";
  str = '<div class="sideBar-resultBox"><div class="result-label">';
  str += '<span>점수</span></div><div class="result-score">';
  str += '<span>'+score+'</span></div></div>';

  $('<div class="sideBar">').append(str);
};
```



소개

전체기능

상세기능

문제해결

느낀점

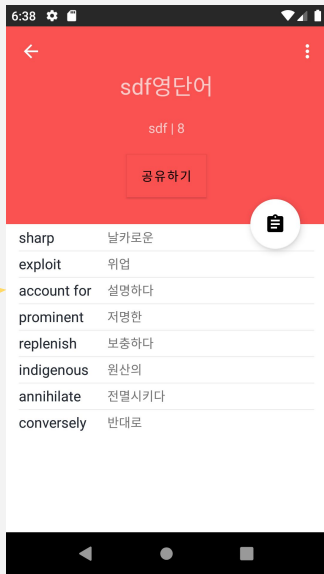
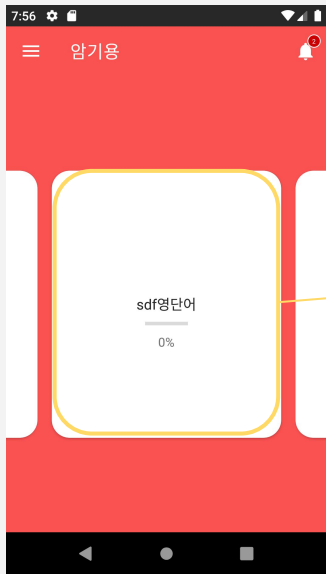


PROJECT

2. 영어 학습을 위한 암기용



상세기능(앱) - 단어장



Activity

```
wordList = new ArrayList<>();
String url = URLs.URL_STUDY_GETWORDLIST+"?bookId="+ bookId +"&bookTitle="+bookTitle;
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d( tag, "result",  msg: "[" + response + "]" );
            try {
                JSONArray array = new JSONArray(response);
                for(int i =0; i<array.length(); i++){
                    Word words;
                    String word = array.getJSONObject(i).getString( name: "word");
                    String meaning = array.getJSONObject(i).getString( name: "meaning");
                    words = new Word(word, meaning);
                    wordList.add(words);
                    mAdapter.notifyDataSetChanged();
                }
                bookNum = array.length();
                Log.d( tag, "bookNum",  msg: "[" + bookNum + "]" );

                final String tv_subtitle = bookId+" | "+bookNum;

                activity_word_detail_tv_title.setText(bookTitle);
                activity_word_detail_tv_subtitle.setText(tv_subtitle);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }
);
```

```
VolleySingleton.getInstance(getApplicationContext()).addRequestQueue(stringRequest);
```

```
mAdapter = new WordDetailAdapter(wordList);
mRecyclerView.setAdapter(mAdapter);
```

Adapter

```
View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.content_word_item, parent, false);
CustomViewHolder holder = new CustomViewHolder(view);

return holder;

public void onBindViewHolder(@NonNull WordDetailAdapter.CustomViewHolder holder, int position) {
    holder.content_word_tv_word.setText(wordList.get(position).getWord());
    holder.content_word_tv_meaning.setText(wordList.get(position).getMeaning());
}
```

xml

```
<TextView
    android:id="@+id/content_word_tv_word"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="3"
    android:gravity="left|center_vertical"
    android:textSize="16dp"
    android:textColor="#222831"
    android:text="@{word}" />

<TextView
    android:id="@+id/content_word_tv_meaning"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="7"
    android:gravity="left|center_vertical"
    android:textSize="14dp"
    android:text="@{meaning}" />
```



소개

전체기능

상세기능

문제해결

느낀점



PROJECT

2. 영어 학습을 위한 암기용



문제해결

- 특정 단어장을 학습할 때 무한 로딩에 걸리는 오류
 - 원인
 - 객관식 문제의 보기를 단어의 뜻이 담겨있는 배열에서 random하게 가져오는데, 이 때 중복검사에서 오류가 나는 것을 확인
 - 한 단어장에 뜻이 같은 단어가 여러개 존재 할 경우를 대비해 문제해결 필요
 - 해결
 - 단어들의 뜻을 담은 meanArray를 만들 때 Set 객체와 Array.from 을 이용해 unique한 array를 얻음
 - filter나 reduce등 다른 방법도 있지만 Set객체의 처리 속도가 가장 빠르기 때문에 Set을 선택

```
//meaning만 들어있는 array생성
function getMeaningArray(){
  for(i in WordCardArray)
  {
    x= WordCardArray[i].meaning;
    meanArray.push(x);
  }
  meanArray = Array.from(new Set(meanArray));
};
```





PROJECT

2. 영어 학습을 위한 암기용

느낀점

- 후기
 - Git & Github, SourceTree를 이용해 협업능력을 향상시킬 수 있었습니다.
 - 혼자 구현할 때에는 알지못했던 오류를 팀원의 피드백을 받으며 찾아내 수정하며 테스트와 피드백의 중요성을 알게되었습니다.
 - 안드로이드 단어장리스트 화면을 구현하면서 단시간 내에 낯선 환경에서 개발해보는 경험을 하였습니다.
- 보완점
 - Commit, push, pull 외에 issue, pull request등의 기능도 사용해 프로젝트를 좀 더 효율적으로 관리해보고 싶습니다.
 - Refactoring을 공부해 조금 더 가독성있는 코드를 구현하고 싶습니다.

