소프트웨어 상세 설계서

App

> 이 프로그램의 메인. 최초실행

- **init**()
  - 메인

View

MainWidget(QWidget)

> 제일 처음 ui 리턴

- (DBManager) dbmanager
  - 영단어를 가져오기 위해 가지고있음
- **init**()
- init_ui()
- daily_clicked()
- words_clicked()
- quiz_clicked()
- repeat_clicked()
- keyPressEvent() : esc가 눌렸을 때 꺼짐

DailyWidget(QWidget)

> Daily widget 리턴

- (DBManager) dbmanager
  - 영단어를 가져오기 위해 가지고있음
- **init**()
- init_ui()
- word_clicked()

WordsWidget(Qwidget)

> Words widget 리턴

- **init**()
- init_ui()
- set_table_widget()
    - all tab, known tab, unknown tab의 각 테이블에 영어단어를 set해줌
    - 디자인적 요소를 고려한 속성 추가
- add_btn_clicked()
- word_clicked()

QuizWidget(Qwidget)

> Study Widget 리턴

- (DBManager) db_manager
- (QuizMaker) quiz_maker
- (QPushButton) answer_btn1
- (QPushButton) answer_btn2
- (QPushButton) answer_btn3
- (QLabel) problem_label
- (QPushButton) next_btn
- (boolean) has_answer
    - 객관식 답을 클릭했는지 안 했는지 가지고있음
- (QLabel) text_label
- **init**()
- init_ui()
- answer_btns_clicked()
- next_btn_clicked()

RepeatWidget(QWidget)

> Repeat Widget 리턴

- **init**()
- init_ui()
- answer_btns_clicked()
- next_btn_clicked()
- play_btn_clicked()

AddWidget(QWidget)

> Add Widget 리턴. 단어 추가하는 화면

- (DBManager) dbmanager
    - 영단어를 가져오기 위해 가지고있음
- (QLineEdit) kor_edit
    - 영단어중 뜻이 입력되는 lineedit
- (QLineEdit) eng_edit
    - 영단어중 영어단어이 입력되는 lineedit
- **init**()
- init_ui()
- ok_btn_clicked()
- import_csv_btn_clicked()

Model

Word

> 영어단어, 뜻, 맞은 갯수를 담고 있는 클래스

- **init**()
- **init**(eng,kor,count)
- (string) eng
    - 영어단어
- (string) kor
    - 뜻
- (int) count
    - 맞은 개수

DBManager

> Word 관련된 데이터를 관리하는 클래스(싱글톤)

- **init**()
    - 파일에서 불러와서 __all_words, __known_words, __unknown_words, __right_words_count 초기화
- (dic) __all_words
    - {eng, Word}
    - 추가한 모든 단어
- (dic) __known_words
    - {key: eng, value: Word}

    - 사용자가 알고있는 단어
- (dic) __unknown_words
    - {key: eng, value: Word}
    - 사용자가 모르는 단어
- (dic) __daily_words
    - {key: eng, value: Word}
    - unknown에서 무작위로 가져온 20개의 단어
    - 매일매일 리스트가 바뀌어야 함
- (int) GRADE_CUT
    - 모르는 단어에서 아는 단어로 전환되는 기준
- _get_instance()
    - 싱글톤을 구현하기 위한 메쏘드
- instance()
    - 싱글톤을 구현하기 위한 메쏘드
- add_word(word, mean)
    - 이미 있는 단어는 false, 없는 단어는 true 리턴
    - __all_words, __unknow_words에 추가
- add_words_from_csv(path)
    - csv파일은 선택하여 여러개 한번에 추가
    - csv형식
영어	한글
banana	바나나
apple	사과
paper	종이
- solve_quiz(eng)
    - 사용자가 문제를 맞췄을 때
    - __right_words_count + 1
    - n번 이상 맞으면 unknown으로 넘어감
- get_all_words()
    - __all_words 리턴
- get_know_words()
    - __get_know_words 리턴

- get_unknown_words()
    - __unknown_words 리턴
- get_daily_words()
    - __daily_words 리턴
- load()
    - __all_words,__right_words_count, 파일에서 불러옴(pickle 이용)
    - __daily_words, 마지막으로 저장한 시간 불러옴
- save()
    - __all_words,__right_words_count, daily_words를 파일에 저장(pickle이용)
    - __daily_words, 마지막으로 저장한 시간 파일에 저장

QuizMaker

> 문제내는 것을 도와주는 클래스

- (string) __problem
    - 1개 문제
- (list) __example
    - 사용자가 선택할 수 있는 보기 3개
- (dic) __words
    - 출제 범위에 들어가는 단어들 저장
- (dic) __answer
    - 정답
- **init**(words)
    - __answer, __example 새로운 값으로 초기화
    - 어떤 영어단어중에서 문제를 낼 것인지 words인자를 통해 받음
- new_problem()
    - 리턴값 없음
    - __answer, __example 새로운 값으로 초기화
    - random 모듈 이용
- get_problem()
    - return __answer 리턴
- get_example()

    - return __example 리턴

- get_answer()
    - return __answer 리턴