

# 증강현실 기반 언어-이미지 번역 어플리케이션



## 설계 명세서

과목명	소프트웨어 공학
담당교수	박외진 교수님
학과명	컴퓨터 공학과
이름	2013312448 이수영
	2011312488 박성우
	2011312662 박주홍
	2011311259 이근성
	2011312373 이학상

# 목 차

<b>1. Preface .....</b>	<b>4</b>
1.1 Objectives .....	4
1.2 Readership .....	4
1.3 Document Structure .....	4
1.4 Version of the Document .....	5
<b>2. Introduction .....</b>	<b>7</b>
2.1 Objectives .....	7
2.2 Applied Model .....	7
2.3 Applied Tool .....	9
<b>3. System Architecture .....</b>	<b>10</b>
3.1 Objectives .....	10
3.2 System Organization .....	10
<b>4. User Requirements Definition .....</b>	<b>11</b>
4.1 Objectives .....	11
4.2 Class Diagram .....	11
4.3 Sequence Diagram .....	13
4.4 State Diagram .....	16
<b>5. Database Design .....</b>	<b>20</b>
5.1 Objectives .....	20
5.2 ER Diagram .....	20
5.3 Relational Schema .....	22
5.4 SQL DDL .....	23
<b>6. Development Environment .....</b>	<b>25</b>
6.1 Objectives .....	25
6.2 Programming language & IDE .....	25
6.3 Coding Rule .....	25
6.4 Version Management Tool and Policy .....	26

<b>7. Index .....</b>	<b>27</b>
7.1 Image Index .....	27
7.2 Diagram Index .....	27
 <b>8. 기타 .....</b>	 <b>28</b>
8.1 회의록 .....	28
A. 5월 24일 .....	29
B. 6월 2일 .....	32
C. 6월 12일 .....	35
8.2 Trello 기록 .....	37

# 1. Preface

## 1.1 Objectives

Preface 에서는 본 문서의 대상 독자들과, 문서의 전반적인 구조, 각 부분의 역할에 대하여 제시한다. 그리고 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경 사항들을 서술한다.

## 1.2 Readership

해당 설계 명세서는 시스템의 기능과 서비스, 그리고 운영적인 제약사항을 구조적인 문서로 나타낸 것이다. 대상 독자는 System end-users, client engineers, system architects, software developers이며, 시스템의 개발과 유지 보수에 있어 외주 업체가 참여한다면 해당 업체도 대상이 된다. 즉 시스템의 개발과 유지 보수에 관련된 모든 사람을 대상 독자로 한다.

## 1.3 Document Structure

이 문서는 총 9개의 부분으로 구성되어 있다. Preface, Introduction, System Architecture, 증강현실 번역 시스템, Database Design, Development Environment, Index, 기타로 구성된다. 각 부분의 역할은 다음과 같다.

### A. Preface

본 문서의 대상 독자들과, 문서의 전반적인 구조, 각 부분의 역할에 대하여 제시한다. 그리고 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들을 서술한다.

### B. Introduction

Introduction 에서는 시스템 설계에 사용한 다이어그램과 틀에 대해 소개하고 설명한다.

### C. System Architecture

System Architecture를 통해 목표 시스템에 대한 자세한 개요를 설명한다. 본 어플리케이션의 전체적 분포를 파악할 수 있으며 System의 구조를 Block diagram으로 나타낸 뒤, 그것들의 관계와 실제 사용을 Package diagram, Deployment diagram으로 표현한다. 시스템의 Modular decomposition 방식은 대제목 번호 4번에서 설명한다.

### D. 증강현실 언어-이미지 번역 시스템

증강현실 언어-이미지 번역 어플리케이션의 시스템 설계를 Class diagram, Sequence diagram, State diagram을 통해 설명한다.

### E. Database Design

Database Design에서는 요구사항 명세서에 서술된 데이터베이스에 대한 요구사

항을 기반으로 ER diagram과 Relational schema를 작성한다. 작성한 내용을 바탕으로 SQL DDL을 구성한다.

#### **F. Development Environment**

Development Environment에서는 증강현실 번역 시스템의 개발 환경과 코딩 규칙에 대해 설명한다. 사용한 버전 관리 도구의 소개와 사용 규칙에 대해 서술한다.

#### **G. Index**

Index에서는 문서의 인덱스들이 포함된다. 알파벳 순서의 인덱스, 다이어그램의 인덱스, 기능의 인덱스 등이 포함된다.

#### **H. 기타**

기타에는 회의록과 Trello를 통한 협업 기록이 포함된다. 효율적으로 Waterfall 모델을 적용하고자 매주 팀 모임을 가졌으며 모임에서 이뤄진 회의의 내용을 회의록으로 정리하고 기록하였다. 또한 오프라인에서 만나지 않는 기간에도 온라인 협업 도구인 Trello를 이용하여 과제를 수행하였다.

### **1.4 Version of the Document**

#### **A. Version Format**

버전 번호는 major number와 minor number로 이루어져 있으며, (major number).(minor number)의 형태로 표현한다. 문서의 버전은 0.1 부터 시작한다.

#### **B. Version Management Policy**

설계 명세서를 수정할 때 마다 버전을 업데이트한다. 다만 변경간의 간격이 1 시간 이내일 때에는 버전 번호를 업데이트하지 않고, 하나의 업데이트로 간주한다. 이미 완성된 파트를 변경할 때에는 minor number를 변경하며, 새로운 부분을 추가하거나 문서의 구성이 예전에 비해 괄목할 변화가 있을 경우 major number를 변경한다.

#### **C. Version Update History**

Version	Modified Date	Explanation
0.1	2017. 6. 8	문서의 초안을 작성함. 각 항목에 들어갈 내용을 간략하게 정리함.
1.0	2017. 6. 9	Preface, Introduction, Development Environment를 작성함.
2.0	2017. 6. 12	System Architecture, 증강현실 언어-이미지 번역 시스템, Database Design을 작성함.
2.1	2017. 6. 17	진행된 내용까지의 Diagram을 추가함.
2.2	2017. 6. 19	증강현실 언어-이미지 번역 시스템의 State Diagram을 세분화함.
3.0	2017. 6. 20	기타에 회의록을 추가함.
3.1	2017. 6. 21	기타에 Trello 기록을 추가함.
4.0	2017. 6. 21	Index를 작성함.
4.1	2017. 6. 22	문서 전체의 형식을 통일함.

## 2. Introduction

### 2.1 Objectives

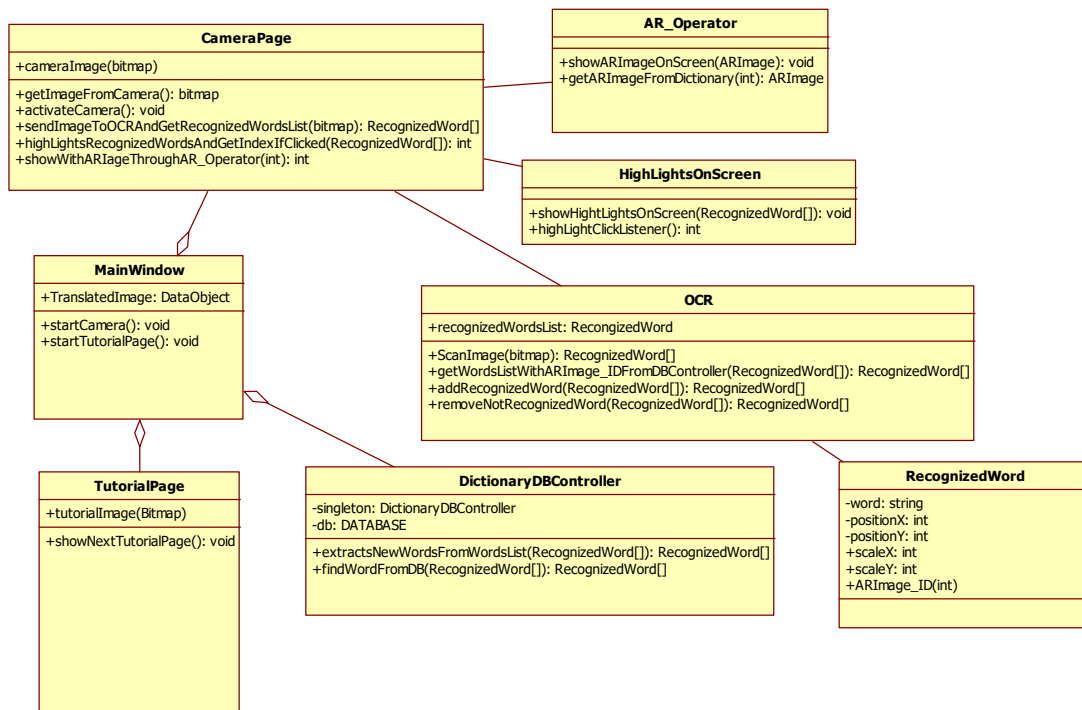
Introduction 에서는 시스템 설계에 사용한 다이어그램과 틀에 대해 소개하고 설명한다.

### 2.2 Applied Model

#### A. UML

Unified Model Language, UML은 객체지향 소프트웨어 설계를 위해 사용되던 여러 종류의 다이어그램들을 통합하여 만든 모델링 표기법이다. 본 문서에서는 다음과 같은 다이어그램을 활용하여 본 시스템 및 시스템에 포함된 서브 시스템의 다양한 설계를 표현하였다.

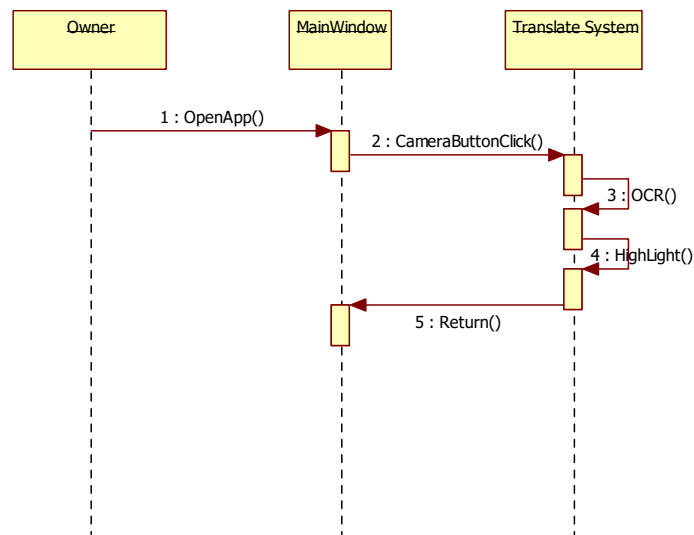
#### B. Class Diagram



(다이어그램 1 : Class Diagram)

Class Diagram은 객체지향 지향 설계에서 가장 널리 사용되는 다이어그램이다. 클래스 단위로 그 관계를 표현하며, 각 클래스는 고유한 속성과 행동을 갖는다. 객체지향의 상속 등의 관계를 명확하게 표현할 수 있고, 인터페이스, 추상 클래스 등을 표현하기에도 유리하다.

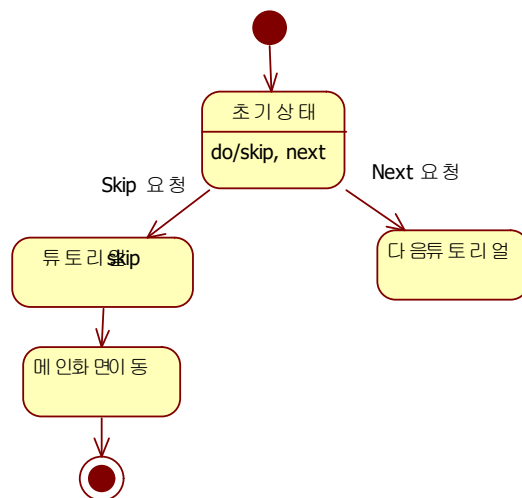
#### C. Sequence Diagram



(다이어그램 2 : Sequence Diagram)

Sequence Diagram은 각 컴포넌트들이 주고 받는 메시지의 흐름을 시간 순차적으로 표현한다. Sequence Diagram은 Use-Case Diagram과 자주 같이 사용되는데, 이는 각 Use-Case를 상세히 표현하는데 Sequence Diagram을 사용하는 것이 유리하기 때문이다. 각 컴포넌트들의 상호 작용 명확히 보이기 때문에 각 컴포넌트간의 관계와 각 컴포넌트들이 갖고 있는 속성, 행동들을 더욱 명확히 할 수 있다.

#### D. State Diagram



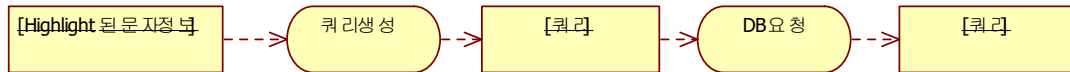
(다이어그램 3 : State diagram)

State Diagram은 각 세부 컴포넌트 혹은 전체 시스템의 행동을 표현한다. 컴포넌트의 상태를 기준으로 메시지에 따라 변화되는 상태를 표현한다. 이러한 표기는 실제 구현에서 UI를 정의



하는데 도움을 준다.

## E. Data-flow Diagram



(다이어그램 4 : Data-Flow Diagram)

Data-flow Diagram은 UML에 포함된 표준 다이어그램은 아니지만 여전히 많은 소프트웨어 설계에서 사용되고 있다. Function단위로 노드를 표현하여 각 Function이 주고받는 데이터의 형태 및 흐름을 나타낼 수 있다.

## 2.2 Applied Model



(그림 1 : StarUML 로고)



(그림 2 : MS Visio 로고)

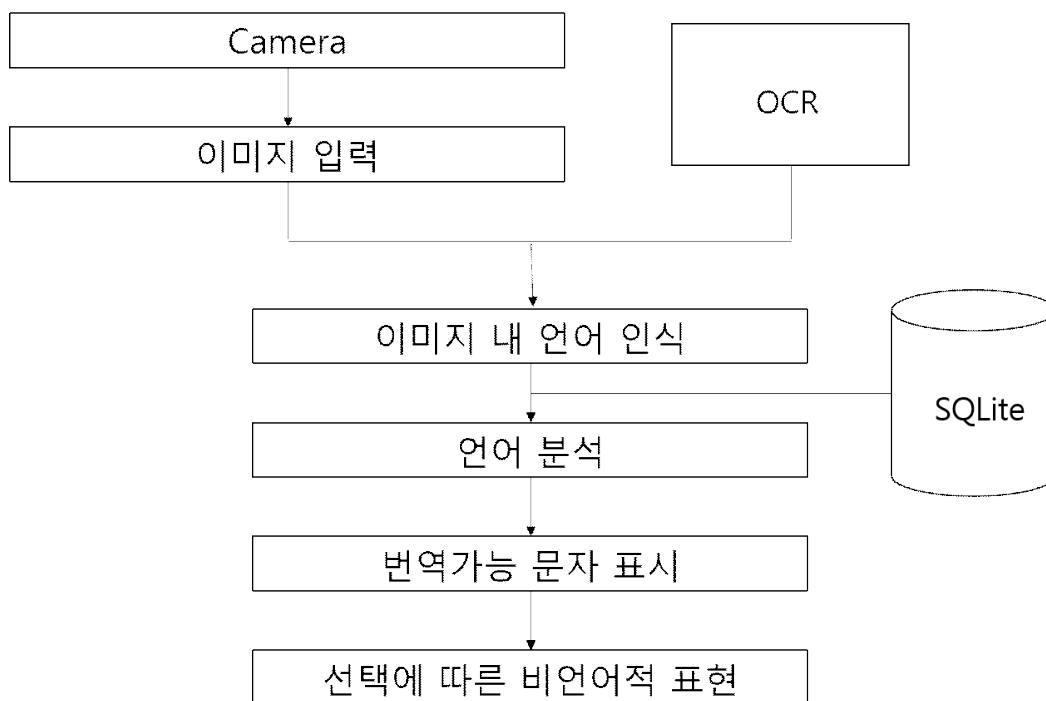
우리 팀에서는 본 시스템 설계를 표현하기 위해 StarUML과 MS Visio를 사용하였다. StarUML은 오픈 소스 UML 플랫폼으로 별도의 기간제한 없이 평가판을 계속 사용할 수 있다. UML에 포함된 대부분의 다이어그램을 지원하고 있어, 설계 명세서에 포함되는 다이어그램을 쉽고 빠르게 표현할 수 있다. MS Visio는 다이어그램이나 도표를 작성을 위한 프로그램으로 UML의 다양한 최신규격을 모두 지원하고 있다. 여러 종류의 스텐실을 지원하고 있어 간편하게 다이어그램을 표현할 수 있다.

### 3. System Architecture

#### 3.1 Objectives

System Architecture를 통해 목표 시스템에 대한 자세한 개요를 설명한다. 본 어플리케이션의 전체적 분포를 파악할 수 있으며 System의 구조를 Block diagram으로 나타낸 뒤, 그것들의 관계와 실제 사용을 Package diagram, Deployment diagram으로 표현한다. 시스템의 Modular decomposition 방식은 대제목 번호 4번에서 설명한다.

#### 3.2 System Organization



(다이어그램 5. 증강현실 번역 시스템 System Architecture)

증강현실 언어-이미지 번역 시스템은 클라이언트와 데이터베이스로 구성된다. 번역 시스템 자체가 클라이언트에 해당되며 해당 클라이언트가 서비스에 접근하여 이용한다.

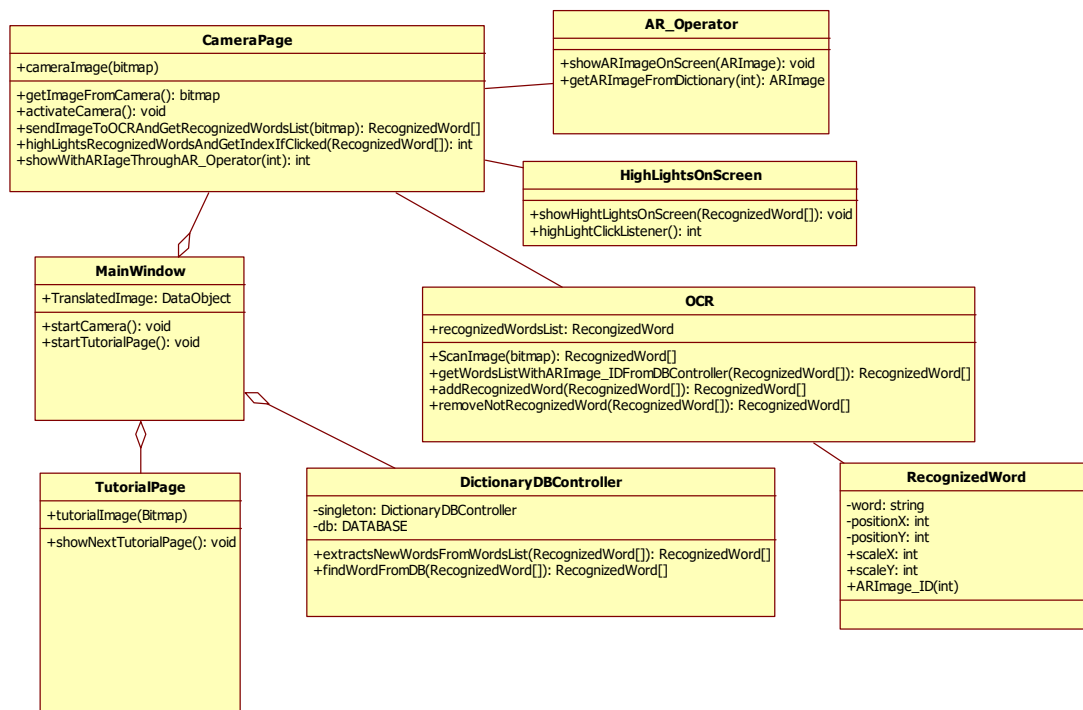
본 어플리케이션은 카메라를 통해서 이미지를 입력하고 그 속에서 언어들은 확인 이를 Database와 함께 언어를 분석하게 된다. 여기서 분석된 문자들 중에 번역이 가능한 문자들을 나타내고 선택에 따라 비언어적 표현을 제공한다.

## 4. 증강현실 언어-이미지 번역 시스템

### 4.1 Objectives

증강현실 언어-이미지 번역 어플리케이션의 시스템 설계를 Class diagram, Sequence diagram, State diagram을 통해 설명한다.

### 4.2 Class Diagram



(다이어그램 6 : 사용자 어플리케이션 Class Diagram)

#### A. MainPage

##### A.1 Methods

- void startCamera(): 화면을 CameraPage로 이동시킨다.
- void startTutorialPage(): 최초사용의 경우 TutorialPage로 이동시킨다.

#### B. TutorialPage

##### B.1 Attributes

- tutorialImage(Bitmap): 튜토리얼용 이미지이다.

##### B.2 Methods

- void showNextTutorialPage(): 다음 튜토리얼 페이지를 보여준다.

## C. CameraPage

### C.1 Attributes

- cameraImage(bitmap): 카메라 현재 이미지를 전달하기 위해 저장한다.

### C.2 Methods

- void activateCamera(): 페이지에 들어와서 카메라를 작동시킨다.
- bitmap getImageFromCamera(): 실시간으로 카메라에서 이미지를 가져와 cameraImage attribute에 저장한다.
- RecognizedWord[] sendImageToOCRAndGetRecognizedWordsList(bitmap): getImageFromCamera()로 가져온 이미지를 OCR객체로 보내고 인식된 단어들의 리스트를 가져온다.
- int highLightsRecognizedWordsAndGetIndexIfClicked(RecognizedWord[]): HighLightsOnScreen클래스를 이용하여 인식된 단어 리스트를 하이라이트 시키고 만약 눌리면 그에 해당하는 index를 가져온다.
- showWithARImageThroughAR\_Operator(int): 선택된 단어에 해당하는 index를 AR\_Operator에 보내 ARImage를 보여준다.

## D. HighLightsOnScreen

### D.1 Methods

- void showHighLightsOnScreen(RecognizedWord[]): 인식된 단어 리스트의 x, y좌표를 이용하여 누를 수 있게 강조 한다.
- int highLightClickListener(): 하이라이트된 단어를 선택하면 리스트의 몇 번째 단어가 선택되었는지 반환한다.

## E. DictionaryDBController

### E.1 Methods

- RecognizedWord[] extractsNewWordsFromWordsList(RecognizedWord[]): 인식된 단어 리스트 중 검색안된 새로운 단어들만 뽑아낸다.
- RecognizedWord[] findWordFromDB(RecognizedWord[]): 새로운 단어들을 데이터베이스에서 찾아 다음에 해당하는 이미지를 찾기 쉽게 ARImage\_ID를 저장해 반환한다.

## F. OCR

### F.1 Attributes

- recognizedWordsList(RecognizedWord[]): 인식된 단어들의 리스트

### F.2 Methods

- RecognizedWord[] scanImage(bitmap): 받은 이미지를 스캔하여 단어들을 뽑아낸다.
- RecognizedWord[] getWordsListWithARImage\_IDFromDBController(RecognizedWord[]): recognizedWordList를 검색을 위해 DictionaryDBController로 보낸 후 데이터베이스에 있는 단어들만 ARImage\_ID 정보를 담아서 가져온다.

- RecognizedWord[] addRecognizedWord(RecognizedWord): 뽑아낸 단어들중 새로운 단어를 추가한다.
- RecognizedWord[] removeNotRecognizedWord(RecognizedWord): 오랫동안 보이지 않는 단어는 리스트에서 제거한다.

## G. AR\_Operator

### G.1 Methods

- ARImage getARImageFromDictionaryDB(int): DictionaryDB에서 ARImage\_ID를 이용하여 ARImage를 가져온다.
- void showARImageOnScreen(ARImage): 가져온 ARImage를 이용하여 화면에 보여준다.

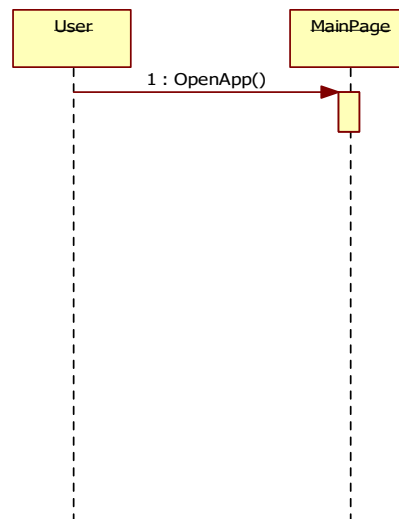
## H. RecognizedWord

### H.1 Attributes

- word(string): 인식된 단어
- positionX(int): 단어의 screen상의 x좌표
- positionY(int): 단어의 screen상의 y좌표
- scaleX(int): 단어의 screen상의 가로길이
- scaleY(int): 단어의 screen상의 세로길이
- ARImage\_ID(int): 데이터베이스에 ID로 빠르게 접근하기 위한 image\_ID

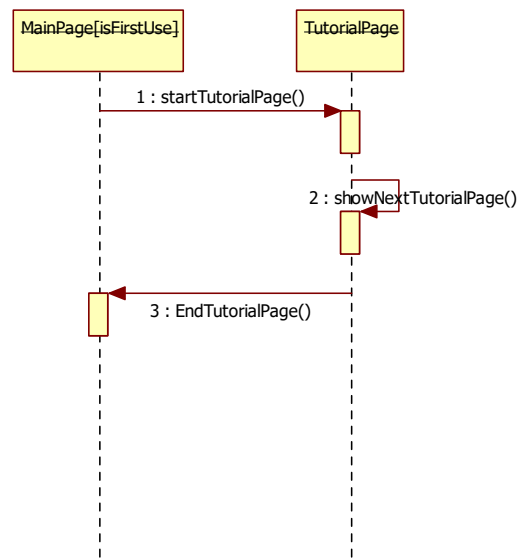
## 4.3 Sequence Diagram

### A. 어플리케이션 실행



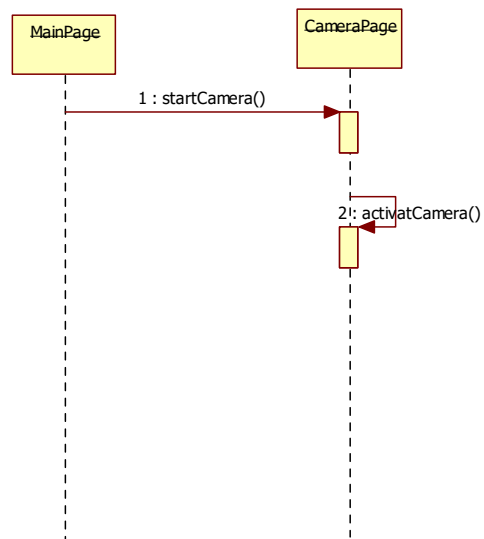
(다이어그램 7 : 어플리케이션 실행  
Sequence Diagram)

### B. 번역하기



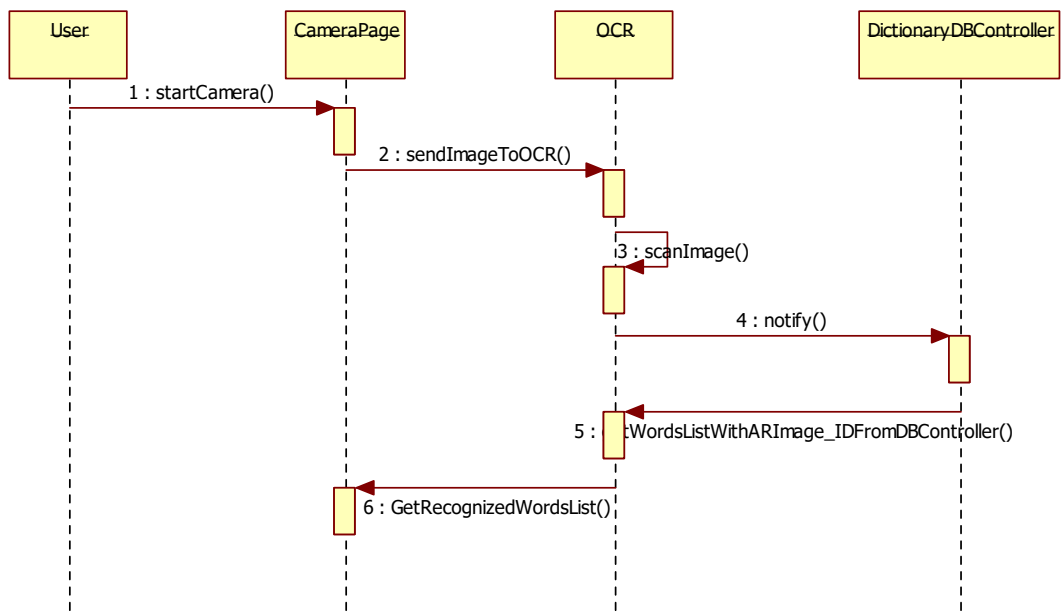
(다이어그램 8 : 튜토리얼 실행 Sequence Diagram)

### C. 번역 버튼 클릭



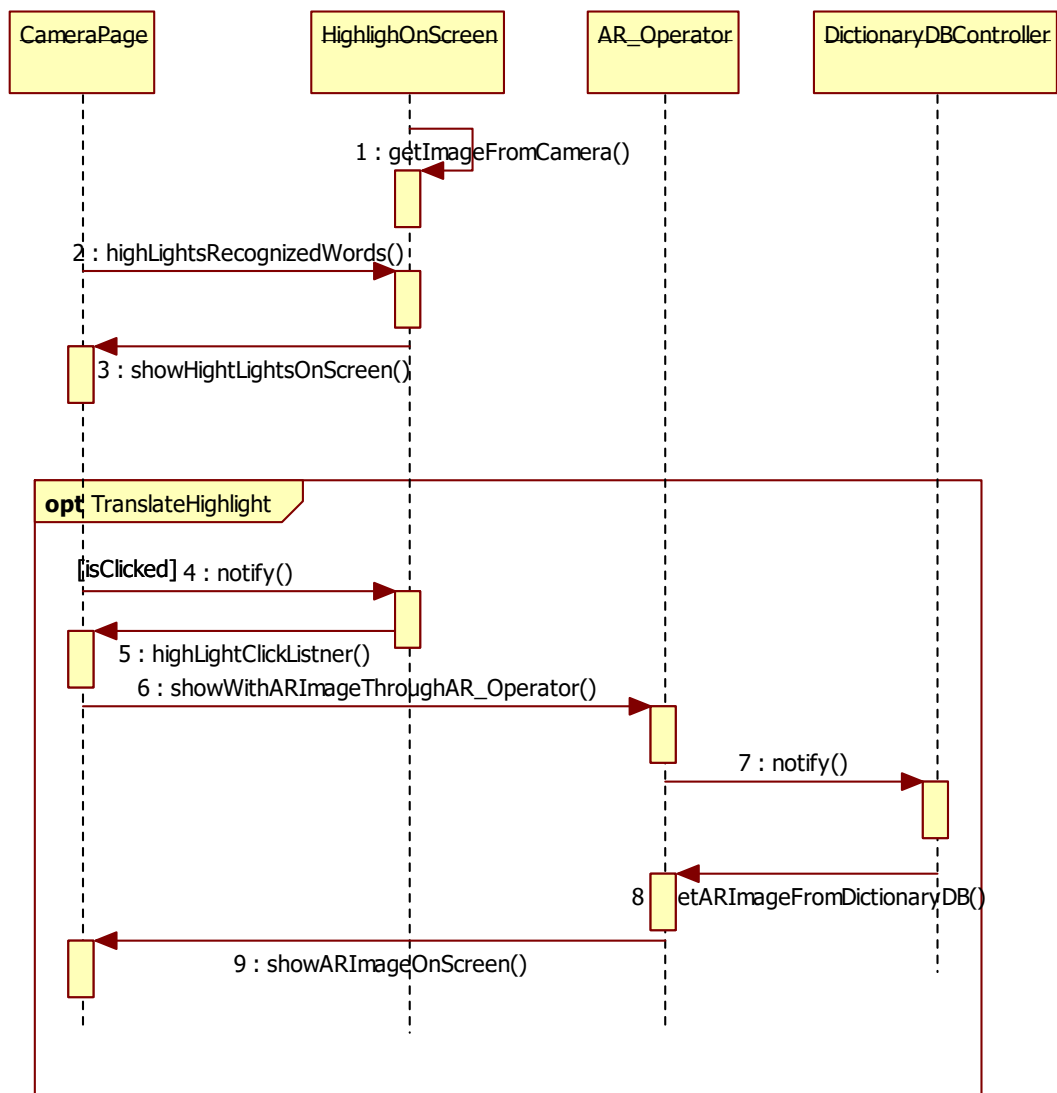
(다이어그램 9 : 번역 버튼 클릭 Sequence Diagram)

### D. 번역 가능한 문자 인식하기



(다이어그램 10 : 번역 가능한 문자 인식하기 Sequence Diagram)

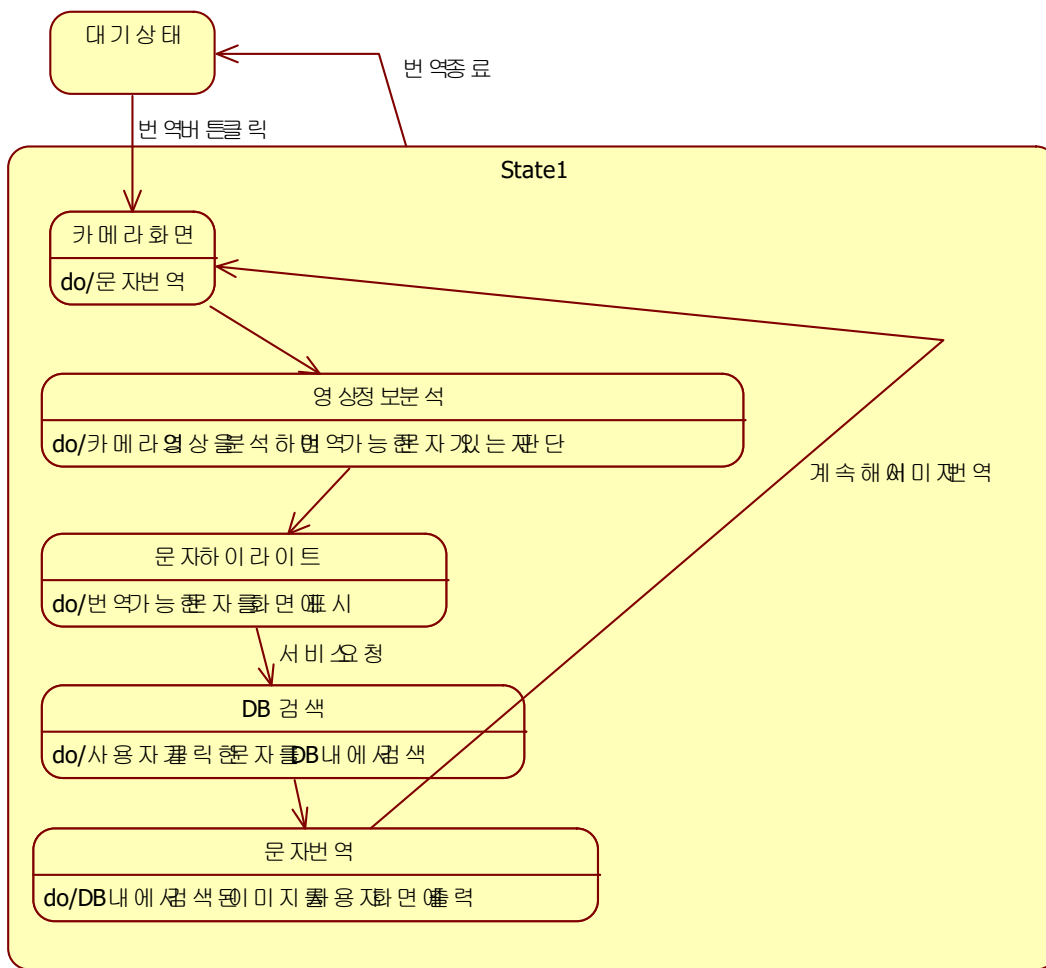
#### E. 강조(하이라이트)된 문자 번역하기



(다이어그램 11 : 하이라이트 된 문자 번역하기 Sequence Diagram)

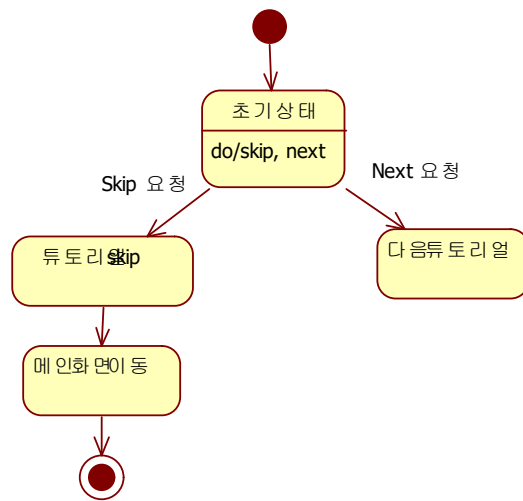
#### 4.4 State Diagram



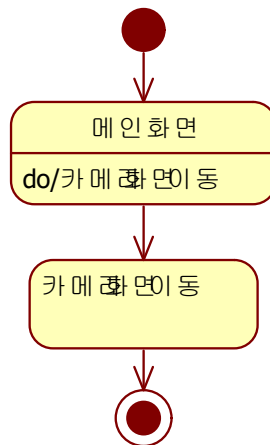


(다이어그램 12 : 전체 어플리케이션 State Diagram)

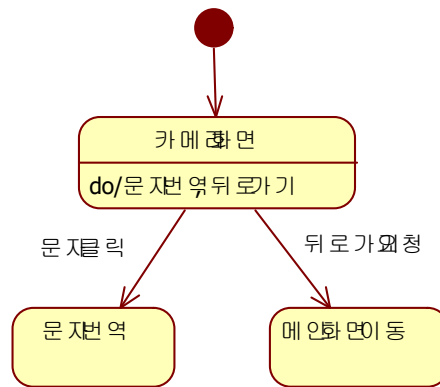
## A. 튜토리얼



## B. 메인 화면

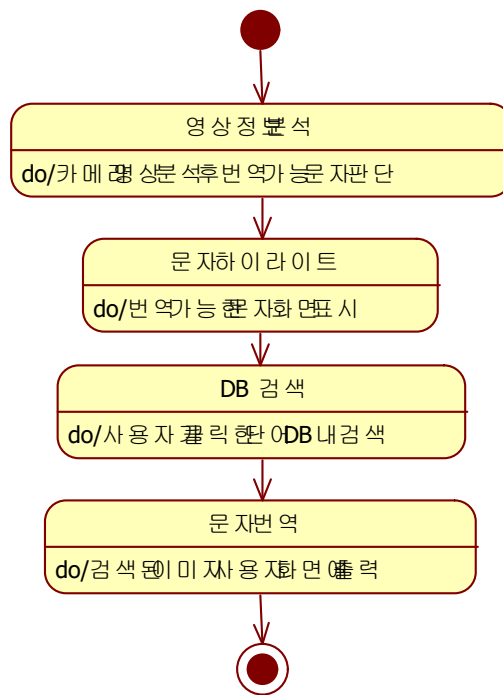


### C. 카메라 화면



(다이어그램 15 : 카메라 화면 State Diagram)

#### D. 문자 번역



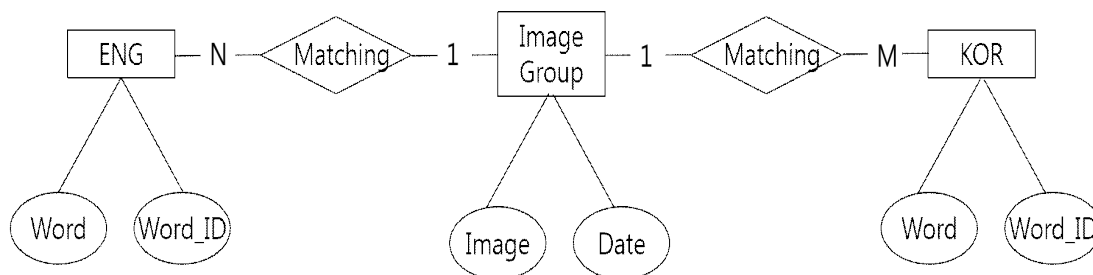
(다이어그램 16 : 문자 번역 State Diagram)

## 5. Database Design

### 5.1 Objectives

Database Design에서는 요구사항 명세서에 서술된 데이터베이스에 대한 요구사항을 기반으로 ER diagram과 Relational schema를 작성한다. 작성한 내용을 바탕으로 SQL DDL을 구성한다.

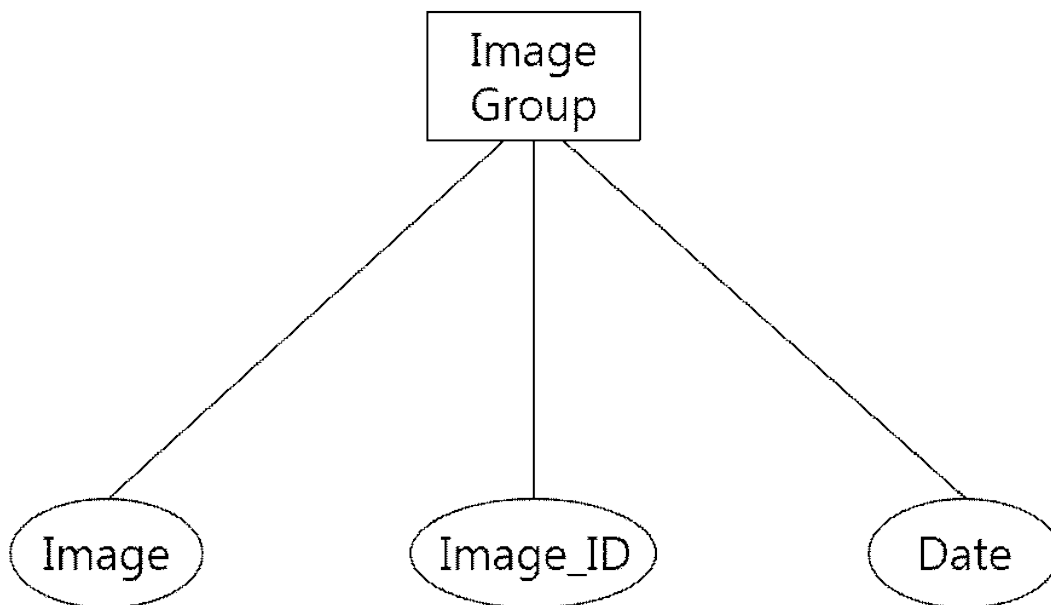
### 5.2 ER Diagram



(다이어그램 17 : 증강현실 번역 시스템 ER Diagram)

#### A. Entity

##### A.1 ImageGroup

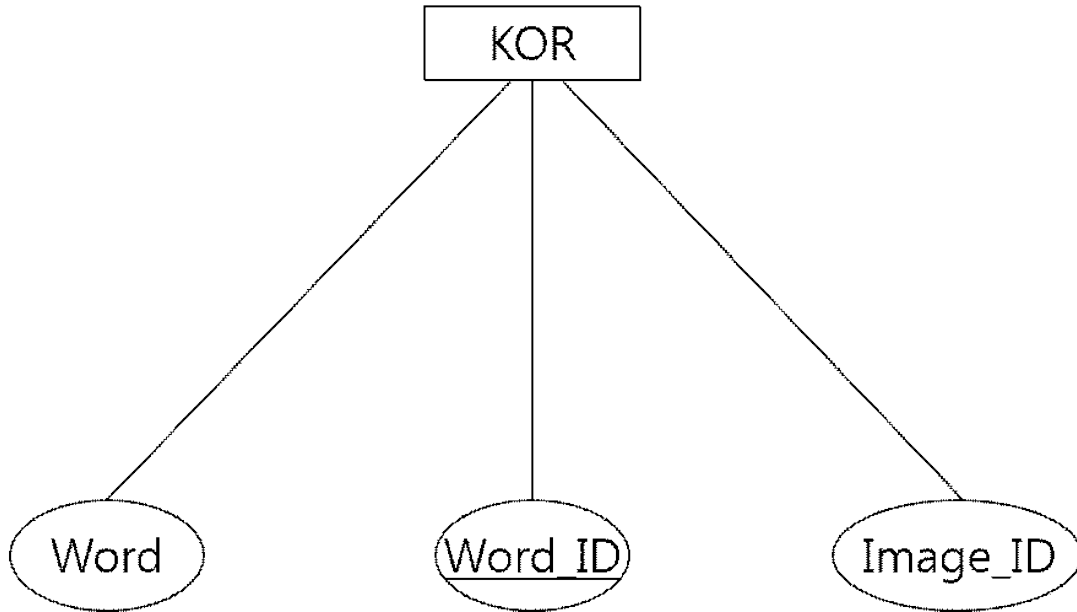


(다이어그램 18 : ImageGroup Entity)

Image를 나타내는 정보이다. Image, Image\_ID, Date 는 각각 이미지, 이미지의 ID, 날짜

를 나타낸다. Key는 Image\_ID 이다.

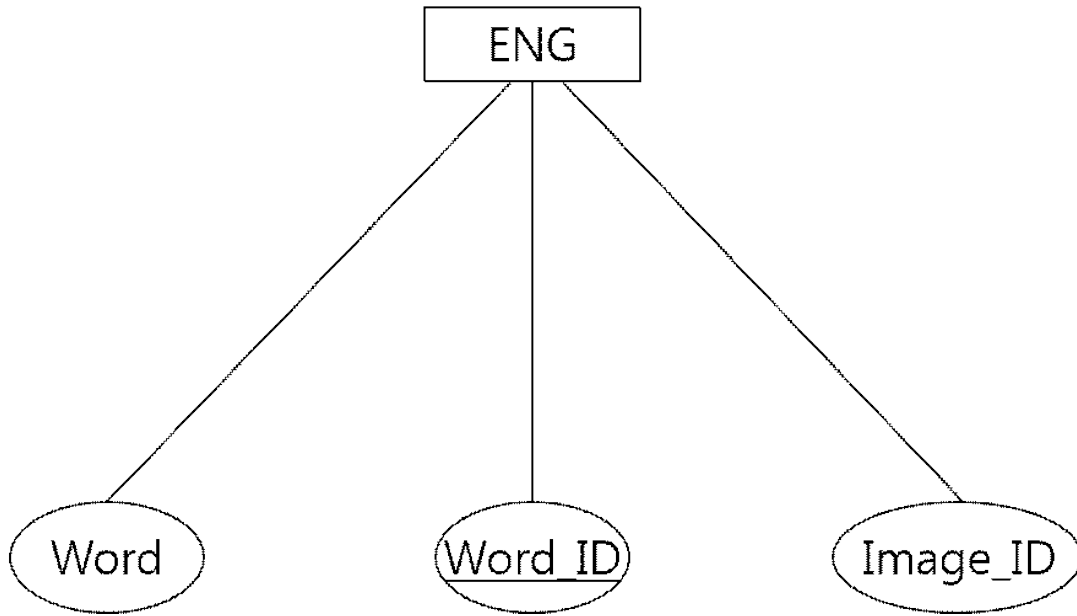
## A.2 KOR



(다이어그램 19 : KOR Entity)

표현 가능한 한국어 단어를 나타낸다. Word, Word\_ID, Image\_ID는 각각 단어, 단어의 ID, 이미지 ID를 나타낸다. 같은 단어이지만 다른 뜻을 나타낼 수 있어서 뜻에 따라 ID를 부여하도록 한다. Key는 Word\_ID이다.

## A.3 ENG



(다이어그램 20 : ENG Entity)

표현 가능한 영어 단어를 나타낸다. Word, Word\_ID, Image\_ID는 각각 단어, 단어의 ID, 이미지 ID를 나타낸다. 같은 단어이지만 다른 뜻을 나타낼 수 있어서 뜻에 따라 ID를 부여하도록 한다. Key는 Word\_ID이다.

### 9.3 Relational Schema

#### A. ImageGroup

ImageGroup		
<u>Image_ID</u>	Image	Date

**Primary Key :** Image\_ID

**Functional Dependencies :** Image\_ID → { Image, Date }

Image 의 속성을 가지는 항목들을 하나로 묶었다. 각 이미지마다 ID가 할당되고 수정된 날짜를 기록하도록 한다. Image\_ID에 따라서 Image와 날짜를 결정한다.

#### B. KOR

KOR		
<u>Word_ID</u>	Word	Image_ID

**Primary Key :** Word\_ID

**Foreign Key :** Image\_ID

**Functional Dependencies :** Word\_ID → { Word, Image\_ID }

한국어 단어들을 하나의 테이블로 관리하도록 한다. 각 단어가 의미하는 뜻에 따라서 ID

를 부여하도록 하고 그에 따라 가리키는 Image\_ID는 다르게 된다. 그리고 같은 단어라도 다른 뜻을 나타내면 Word\_ID가 다르게 된다. 따라서 Word\_ID에 따라서 Word 와 Image\_ID 가 결정된다. 외래키는 Image\_ID로 ImageGroup을 참조한다.

### C. ENG

ENG		
<b>Word_ID</b>	Word	Image_ID

**Primary Key** : Word\_ID

**Foreign Key** : Image\_ID

**Functional Dependencies** : Word\_ID → { Word, Image\_ID }

영어 단어들을 하나의 테이블로 관리하도록 한다. 각 단어가 의미하는 뜻에 따라서 ID를 부여하도록 하고 그에 따라 가리키는 Image\_ID는 다르게 된다. 그리고 같은 단어라도 다른 뜻을 나타내면 Word\_ID가 가다르게 된다. 따라서 Word\_ID에 따라서 Word 와 Image\_ID 가 결정된다. 외래키는 Image\_ID로 ImageGroup을 참조한다.

## 9.4 SQL DDL

### A. ImageGroup

ImageGroup	
<pre>Create ImageGroup(     Image_ID      Image_ID_TYPE NOT NULL,     Image         Image_TYPE NOT NULL,     DATE          DATE_TYPE NOT NULL,     PRIMARY KEY(Image_ID) );</pre>	
<p>Image_ID 는 Primary Key 이므로 NOT NULL이다. Image의 경우 반드시 나타내야하는 항목이므로 NOT NULL로 한다. Date의 경우 업데이트 날짜이므로 NOT NULL이어야 한다.</p>	

### B. KOR

KOR
<pre> Create KOR(     Word_ID Word_ID_TYPE NOT NULL,     Word      Word_TYPE NOT NULL,     Image_ID   Image_ID_TYPE NOT NULL,     PRIMARY KEY(Word_ID),     FOREIGN KEY(Image_ID) REFERENCES ImageGroup ); </pre>
<p>Word_ID 의 경우 Primary Key 이므로 NOT NULL이다. Word의 경우 단어를 나타내므로 반드시 값이 존재한다. Image_ID의 경우 단어가 항상 이미지를 가리켜야 하므로 NOT NULL이다.</p>

### C. ENG

KOR
<pre> Create KOR(     Word_ID Word_ID_TYPE NOT NULL,     Word      Word_TYPE NOT NULL,     Image_ID   Image_ID_TYPE NOT NULL,     PRIMARY KEY(Word_ID),     FOREIGN KEY(Image_ID) REFERENCES ImageGroup ); </pre>
<p>Word_ID 의 경우 Primary Key 이므로 NOT NULL이다. Word의 경우 단어를 나타내므로 반드시 값이 존재한다. Image_ID의 경우 단어가 항상 이미지를 가리켜야 하므로 NOT NULL이다.</p>



## 6. Development Environment

### 6.1 Objectives

Development Environment 에서는 증강현실 번역 시스템의 개발 환경과 코딩 규칙에 대해 설명한다. 사용한 버전 관리 도구의 소개와 사용 규칙에 대해 서술한다.

### 6.2 Programming language & IDE



Developer Tools  
Xcode 4

(그림 3 : Xcode 로고)



(그림 4 : eclipse 로고)

본 제품은 안드로이드와 iOS 모두 지원하도록 한다. iOS 어플리케이션은 Xcode를 사용하여 Object C 언어를 이용하여 개발하도록 하고, 안드로이드 어플리케이션은 이클립스로 자바를 이용하여 제작하도록 한다. OCR 엔진으로 구글의 오픈소스인 Tesseract를 사용하도록 한다. 한글 자연어 처리는 관련 오픈 소스인 KoNLP를 이용한다.

### 6.3 Coding Rule

우리 팀에서는 시스템을 개발하는데 있어서 다음과 같은 Coding rule을 적용한다. 헤더 파일이 여러 번 포함되는 것을 막기 위해 #define을 사용한다. include문에서는 가독성을 높이고 의존성을 낮추기 위해 다음과 같은 순서를 사용한다. 1) Objective C라이브러리, 2) JAVA 라이브러리, 3) C, C++ 라이브러리 4) 프로젝트에 속한 헤더파일의 순서를 사용한다.

함수에서 변수는 가능한 한 좁은 범위에서 활성화되도록 한다. 변수를 선언할 때 초기화한다. 생성자는 멤버 변수를 초기화만 하도록 한다. 상속은 public으로 한다. 멤버 변수는 private로 하고, setter, getter 등의 함수를 통해 접근하도록 한다. 함수 이름, 변수 이름, 그리고 파일 이름은 보고 알아보기 쉽게 표현한다. 변수는 접두사를 붙여 Static 변수는 s\_ , global은 g\_를 사용하도록 한다.

주석은 /\* \*/만을 사용하도록 한다. 모든 파일의 저작권을 잘 표시하도록 한다. 본인이 처음 작성한 경우에는 저작자에 본인을 추가하여 기재하도록 한다. 모든 클래스 정의에 목적과 사용 방법을 설명하는 주석을 추가하여 모두 알아 볼 수 있도록 한다. 구현 내용에서 기술적이고 분명하지 않고 중요한 부분에는 주석문이 있어야 한다. 구문의 정렬은 공백이 아니라 탭을 사용한다. 함수 반환 타입과 매개 변수는 함수 이름과 같은 줄에 쓴다. 변수의 초기화는 등호(=)를 사용한다. public, protected와 private 순으로 두며 각각은 한 칸 들여 쓴다.

## 6.4 Version Management Tool and Policy



(그림 5 : SVN 로고)

Version Management Tool으로 SVN(Subversion)을 사용하도록 한다. CVS의 단점을 개선하고 CVS를 대체할 목적으로 개발된 SVN을 사용하도록 한다. CVS에 비해 빠른 업데이트, 브랜칭, 태깅 시간을 지원한다. 익숙한 CVS와 거의 동일한 사용법으로 쉽게 배울 수 있고, 파일 이름변경, 이동 디렉터리 버전 관리도 지원한다.

개발팀은 할당된 task를 수정하고 완성해 갈 때마다 커밋을 하여 버전을 관리하여야 한다. 커밋 시에는 commit log 기능을 이용하여 어느 부분이 수정되었는지 팀원에게 알려줘야 할 의무가 있다. 이는 자신이 다음 작업 시에도 활용된다.

## 7. INDEX

### 7.1 Image Index

그림 1 StarUML 로고 .....	9
그림 2 MS Visio 로고 .....	9
그림 3 Xcode 로고 .....	25
그림 4 eclipse 로고 .....	25
그림 5 SVN 로고 .....	26

### 7.2 Diagram Index

다이어그램 1 Class Diagram .....	7
다이어그램 2 Sequence Diagram .....	8
다이어그램 3 State diagram .....	8
다이어그램 4 Data-Flow Diagram .....	9
다이어그램 5 증강현실 번역 시스템 System Architecture .....	10
다이어그램 6 사용자 어플리케이션 Class Diagram .....	11
다이어그램 7 어플리케이션 실행 Sequence Diagram .....	13
다이어그램 8 튜토리얼 실행 Sequence Diagram .....	14
다이어그램 9 번역 버튼 클릭 Sequence Diagram .....	14
다이어그램 10 번역 가능한 문자 인식하기 Sequence Diagram .....	15
다이어그램 11 하이라이트 된 문자 번역하기 Sequence Diagram .....	16
다이어그램 12 전체 어플리케이션 State Diagram .....	17
다이어그램 13 튜토리얼 State Diagram .....	18
다이어그램 14 메인 화면 State Diagram .....	18
다이어그램 15 카메라 화면 State Diagram .....	19
다이어그램 16 문자 번역 State Diagram .....	19
다이어그램 17 증강현실 번역 시스템 ER Diagram .....	20

다이어그램 18 ImageGroup Entity .....	20
다이어그램 19 KOR Entity .....	21
다이어그램 20 ENG Entity .....	22

## **8. 기타**

### **8.1 회의록**

A. 5월 24일 회의록

	<h1 style="text-align: center;">팀 프로젝트 회의록</h1>
---	---

2017학년도 봄학기 소프트웨어공학개론(SWE3002)				
팀명	5조			
과제명	증강현실 기반 언어-이미지 번역 어플리케이션			
회의 일시	2017. 5. 24 (수) 오후 6시			
회의 장소	산학협력관 러닝팩토리 85113			
참여자 팀원	성 명	이수영	학 과	컴퓨터공학과
	성 명	박성우	학 과	컴퓨터공학과
	성 명	박주홍	학 과	컴퓨터공학과
	성 명	이근성	학 과	컴퓨터공학과
	성 명	이학상	학 과	컴퓨터공학과

## 1. 이번 회의의 내용

### (1) 개발 방법론 정하기

- Waterfall model(Appendices에 내용 추가), 회의록 작성(팀플 증빙 자료 첨부)



사진 1. 산학 협력관 오프라인 팀플

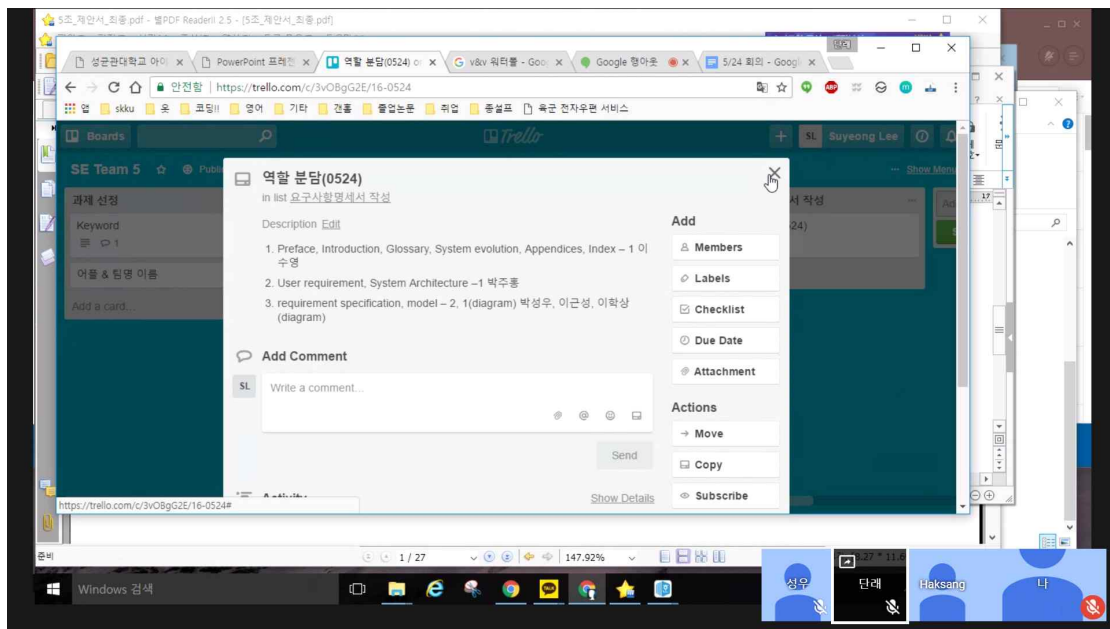


사진 2. 행아웃 화면공유를 이용한 온라인 팀플

### (2) 보고서 작성 시, “~이다.”로 어미 통일

### (3) 요구사항 명세서 작성 역할 분담

이름	역할
이수영	Preface, Introduction, Glossary, System evolution, Appendices, Index
박주홍	User requirement, System Architecture
박성우, 이근성	System requirement specification, System model
이학상	전체 Diagram 작성

#### (4) User requirement 의논

Functional	글자 인식 기능
	글자-이미지 번역 기능(데이터 사용x-이 경우 데이터 용량 너무 크면 안 됨), 데이터 사용
	번역된 이미지 표현 기능
Non-functional (수정 필요)	Performance requirement
	Usability
	Readability

#### (5) System Architecture

- 1) 중앙 서버
- 2) 유저 시스템

## 2. 다음 회의까지 할 것

- 역할 분담한 부분 완료하기(~5/31)

## 3. 그 외 기록할 사항

- 간트 차트의 사용 고려해 보기

B. 5월 24일 회의록

	<h1>팀 프로젝트 회의록</h1>
---	---------------------

2017학년도 봄학기 소프트웨어공학개론(SWE3002)				
팀명	5조			
과제명	증강현실 기반 언어-이미지 번역 어플리케이션			
회의 일시	2017. 6. 2 (금) 오후 6시			
회의 장소	산학 협력관 가든브릿지			
참여자 팀원	성 명	이수영	학 과	컴퓨터공학과
	성 명	박성우	학 과	컴퓨터공학과
	성 명	박주홍	학 과	컴퓨터공학과
	성 명	이근성	학 과	컴퓨터공학과
	성 명	이학상	학 과	컴퓨터공학과



## 1. 이번 회의의 내용

### (1) 요구사항 명세서 말은 부분 피드백

- 미흡한 부분이나 보완이 필요한 부분 피드백, 회의록 작성(팀플 증빙 자료 첨부)



사진 1. 산학 협력관 오프라인 팀플

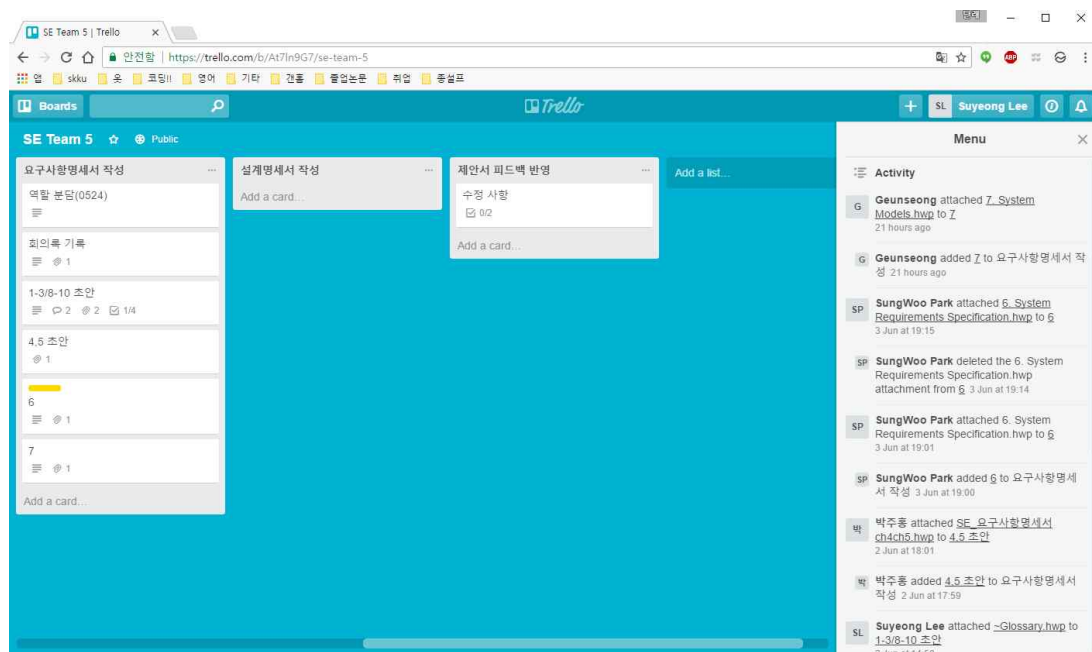


사진 2. 트렐로를 이용한 온라인 팀플

### (2) 피드백 내용

- System architecture와 User requirement 내용 조율
- System requirements specification 피드백 : 구체화 (ex. 이미지 → Stereo type 이미지, dependability requirements 내용 추가)

- 오류가 날 만한 이슈 고민 : text가 아닌 부분 text로 인식할 경우, 세로로 읽을 경우 등

### (3) 설계 명세서 작성 역할 분담

이름	역할
이수영	통합 후 최종 PPT 작성 및 발표
박주홍	요구사항 명세서 appendices의 데이터베이스 요구사항 작성 후 그 내용을 바탕으로 설계 명세서 DB Design
박성우	시스템 Class diagram 내용
이근성	시스템 Sequence diagram 내용
이학상	전체 Diagram (이미지) 작성

## 2. 다음 회의까지 할 것

- 역할 분담한 부분 업로드 및 피드백
- 시나리오 부분 발전

## 3. 그 외 기록할 사항

C. 5월 24일 회의록

	<h1 style="text-align: center;">팀 프로젝트 회의록</h1>
---	---

2017학년도 봄학기 소프트웨어공학개론(SWE3002)				
팀명	5조			
과제명	증강현실 기반 언어-이미지 번역 어플리케이션			
회의 일시	2017. 6. 12 (금) 오후 6시			
회의 장소	산학 협력관 가든브릿지			
참여자 팀원	성 명	이수영	학 과	컴퓨터공학과
	성 명	박성우	학 과	컴퓨터공학과
	성 명	박주홍	학 과	컴퓨터공학과
	성 명	이근성	학 과	컴퓨터공학과
	성 명	이학상	학 과	컴퓨터공학과

## 1. 이번 회의의 내용

### (1) 요구사항 명세서 말은 부분 피드백

- 미흡한 부분이나 보완이 필요한 부분 피드백, 회의록 작성(팀플 증빙 자료 첨부)



사진 1. 산학 협력관 오프라인 팀플

### (2) 피드백 내용

- 언어팩 설계 내용 추가(Structure 등)

### (3) 설계 명세서 작성 역할 분담(변경)

이름	역할
이수영	통합
박주홍	요구사항 명세서 appendices의 데이터베이스 요구사항 작성 후 그 내용을 바탕으로 설계 명세서 DB Design
박성우	시스템 Class diagram 내용
이근성	시스템 Sequence diagram 내용
이학상	전체 Diagram (이미지) 작성

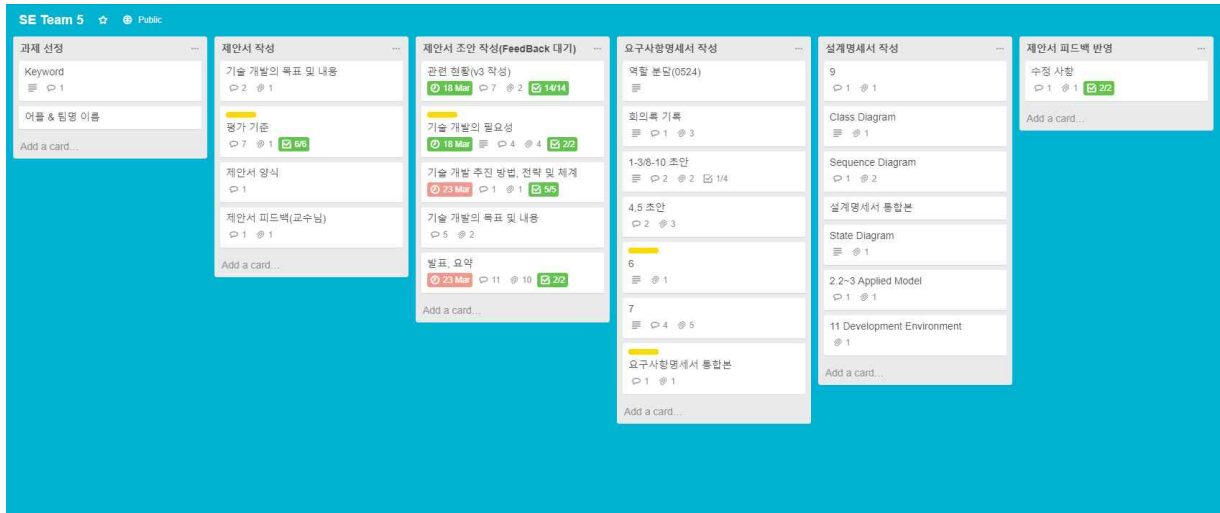
### (4) 역할 분담 변경 : 이수영 - 제안서 피드백 반영 및 회의록 작성(추가)

## 2. 다음 회의까지 할 것

- 역할 분담한 부분 업로드 및 피드백(~21일)

## 3. 그 외 기록할 사항

## 8.2 Trello 기록



본 제품을 개발하는 과정에서 채택한 개발 방법론은 Waterfall 모델이다. Waterfall 모델은 1970년 등장한 이래로 오랜 기간 수많은 프로젝트에 실제로 사용되어 온 기법이므로 적용 사례가 많고, 단계별로 정형화된 접근 방식을 사용하기 때문에 기술적인 위험 요소가 적다는 장점을 가진다. 또한 정해진 순서대로 각 부분의 업무가 분리되고, 문서로 정리되어 다음 단계로 전달되기 때문에 규모가 큰 프로젝트의 경우에도 관리가 편한 방법론이다.

실제로 개발 과정에서도 챗터별로 역할을 나누어 각 파트의 정리된 문서를 받아 다음 파트를 작성하는 식으로 단계별로 진행하였다. Trello의 Board 구성에서 이를 확인할 수 있다.