

LTE Design Specification

소프트웨어공학개론 42class
TEAM #5
김준형 배경률 신아름 이금택 주유흠 천수현

목차

1	Preface.....	6
1.1	Objective	6
1.2	Readership.....	6
1.3	Document Structure.....	6
A.	Preface	6
B.	Introduction	6
C.	System Architecture	6
D.	로그인 및 회원가입 시스템	6
E.	지도 시스템.....	7
F.	타임스탬프 기록 시스템.....	7
G.	사진 등록 시스템.....	7
H.	이미지 분석 시스템 (해시태그 시스템)	7
I.	Protocol Design	7
J.	Database Design.....	7
K.	Testing Plan	7
L.	Development Environment.....	8
M.	Development Plan	8
N.	Index	8
1.4	Version of the Document.....	8
A.	Version Format.....	8
B.	Version Management Policy	8
C.	Version Update History.....	8
2	Introduction	9
2.1	Objective	9
2.2	Applied Diagram	9
A.	UML.....	9
B.	Class Diagram.....	9
C.	Sequence Diagram.....	10
D.	State Diagram.....	11
E.	ER Diagram.....	11
2.3	Applied Tool	12
A.	draw.io	12
2.4	Project Scope.....	13
3	System Architecture	15
3.1	Objective	15
3.2	System Organization	15
A.	로그인 및 회원가입 시스템	15

	B.	지도 시스템	16
	C.	타임스탬프 기록 시스템	17
	D.	사진 등록 시스템	17
	E.	이미지 분석 시스템 (해시태그 시스템).....	17
4		로그인 및 회원가입 시스템	19
	4.1	Objective	19
	4.2	Class Diagram	19
	A.	Sign Up	19
	B.	DB Handler	20
	C.	Login.....	20
	4.3	Sequence Diagram	21
	4.4	State Diagram	22
5		지도 시스템	23
	5.1	Objective	23
	5.2	Class Diagram	23
	A.	Map View	23
	B.	Pin Button.....	24
	C.	Image	24
	D.	Sort View	24
	E.	Record Button.....	24
	F.	DB Handler	24
	5.3	Sequence Diagram	25
	5.4	State Diagram	26
6		타임스탬프 기록 시스템	27
	6.1	Objective	27
	6.2	Class Diagram	27
	A.	Map View	27
	B.	Timestamp Button.....	27
	C.	DB Handler.....	27
	6.3	Sequence Diagram	28
	6.4	State Diagram	28
7		사진 등록 시스템	29
	7.1	Objective	29
	7.2	Class Diagram	29
	A.	Map View	29
	B.	Pin Button	30
	C.	Image	30
	D.	Upload View.....	30

	7.3	Sequence Diagram	31
	7.4	State Diagram	32
8		이미지 분석 시스템 (해시태그 시스템)	33
	8.1	Objective	33
	8.2	Class Diagram	33
	A.	Image Analysis	33
	B.	Hashtag Select View	33
	C.	DB Handler	34
	8.3	Sequence Diagram	34
	8.4	State Diagram	35
9		Protocol Design	36
	9.1	Objective	36
	9.2	JSON	36
	9.3	Protocol Description	36
	A.	Overview	36
	B.	Login Protocol	37
	C.	Sign Up Protocol	37
	D.	Map View Protocol	37
	E.	Route Protocol	38
	F.	Timestamp Protocol	38
	G.	Save, Modify Picture Protocol	38
	H.	Image Analysis Protocol	39
	I.	Hashtag Protocol	39
10		Database Design	40
	10.1	Objective	40
	10.2	ER Diagram	40
	A.	Entity	41
	B.	Relationship	43
	10.3	Relational Schema	44
	A.	User	44
	B.	Picture	44
	C.	Route	45
	10.4	SQL DDL	45
	A.	User	45
	B.	Picture	46
	C.	Route	46
11		Testing Plan	47
	11.1	Objective	47

11.2	Testing Policy.....	47
A.	Developing Testing.....	47
B.	Release Testing.....	47
C.	User Testing.....	47
11.3	Test Case.....	47
A.	Login Test.....	47
B.	Sign Up Test.....	48
C.	Record Test.....	48
D.	Timestamp Test.....	48
E.	Hashtag Test.....	48
F.	Image Save, Modify Test.....	49
12	Development Environment.....	50
12.1	Objective.....	50
12.2	Programming Language & IDE.....	50
A.	Programming Language.....	50
B.	IDE.....	52
12.3	Version Management Tool.....	52
A.	GitHub.....	52
12.4	네이버 지도 API.....	53
12.5	Google Cloud Vision API.....	53
13	Development Plan.....	55
13.1	Objective.....	55
13.2	Schedule (Gantt Chart).....	55
14	Index.....	56
14.1	Objective.....	56
14.2	Table Index.....	56
14.3	Figure Index.....	56
14.4	Diagram Index.....	57

1 Preface

1.1 Objective

Preface 에서는 본 문서의 독자를 정의하고, 본 문서의 구조를 소개한다. 문서의 구조를 소개할 때에는 각 목차의 목적 및 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들과 변경에 대한 근거 등을 기술한다.

1.2 Readership

본 문서의 독자는 시스템을 개발하는 소프트웨어 엔지니어와 시스템 설계 및 개발에 참여하는 모든 구성원으로 정의한다. 만약 해당 시스템 개발을 위해 외부 업체가 참여한다면, 해당 업체의 개발 참여 인원 역시 독자에 포함한다.

1.3 Document Structure

A. Preface

Preface 에서는 본 문서의 독자를 정의하고, 본 문서의 구조를 소개한다. 문서의 구조를 소개할 때에는 각 목차의 목적 및 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들과 변경에 대한 근거 등을 기술한다.

B. Introduction

Introduction 에서는 본 문서의 시스템을 설계하기 위해 사용된 모든 종류의 Diagram 과 Tool 에 대해 설명하고, 본 프로젝트를 통해 개발 중인 'LTE' 시스템의 scope 에 대해 설명한다.

C. System Architecture

System Architecture 에서는 팀에서 개발하고자 하는 LTE 시스템에 대한 전반적인 사항들을 설명한다. LTE 시스템의 전반적인 구조를 block diagram 을 통해 나타낸 후, 이를 class diagram 과 sequence diagram, state diagram 으로 보다 상세히 기술한다.

D. 로그인 및 회원가입 시스템

사용자가 처음으로 본 시스템에 접근할 때 입력하는 사용자 정보를 처리하는 시스템으로, 로그인과 회원가입의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

E. 지도 시스템

로그인한 사용자가 사진과 경로 정보를 입력하면 지도 위에 데이터를 처리하는 프론트 역할의 시스템이다. 지도 시스템의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

F. 타임스탬프 기록 시스템

사용자가 입력하는 시간, 위치 정보를 처리하는 시스템으로, 타임스탬프 기록의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

G. 사진 등록 시스템

사용자가 사진 업로드, 수정, 삭제를 진행하는 시스템으로, 사진 업로드와 수정, 삭제의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

H. 이미지 분석 시스템 (해시태그 시스템)

사용자가 업로드하는 사진 파일을 입력 값으로 받아 해당 이미지를 분석하여 라벨을 추출하고 해당 라벨을 해시태그로 사용자에게 추천하는 시스템으로, 해당 시스템의 설계 구조에 대해 class diagram, sequence diagram, state diagram 을 통해 설명한다.

I. Protocol Design

Protocol Design 에서는 서브시스템 간 상호 통신을 위한 프로토콜에 대해 서술한다. 해당 프로토콜은 원활한 통신을 위해 필수적으로 준수되어야 하며, 이때 전달되는 메시지의 형식과 용도, 의미를 설명한다.

J. Database Design

Database Design 은 시스템에 필요한 데이터베이스를 설계하는 것을 목표로 한다. 우선적으로 데이터베이스의 전체 흐름을 보기 위한 ER Diagram 을 작성하고, 이를 활용하여 Relational Schema 와 SQL DDL 을 작성한다.

K. Testing Plan

시스템이 의도한 대로 실행되는지를 확인하고, 시스템 내부의 결함을 찾기 위한 과정으로, 시스템 설계 단계에 미리 계획하여 시행한다. 본 항목에서는 test policy 에서 testing 의 범위와 접근 방식에 대해 명확히 서술하고, testing 하는 리소스와 항목, testing 기능에 대해 구체적인 입출력 데이터를 바탕으로 test case 를 명시한다.

L. Development Environment

본 항목에서는 개발자의 환경에 대해 설명하며, 사용한 프로그래밍 언어와 IDE 등에 대해 기술한다.

M. Development Plan

본 항목에서는 개발 계획에 대해 서술한다. 개발 계획은 gantt chart 를 통해 구체화하여 표현한다.

N. Index

본 항목에서는 도표, 다이어그램 및 삽입 이미지들의 인덱스를 표기한다. 이를 통해 원하는 정보의 위치를 빠르게 파악할 수 있다.

1.4 Version of the Document**A. Version Format**

해당 문서의 버전은 <major number>.<minor number>의 형태로 표현하며, 본 문서의 버전은 0.1 부터 시작한다.

B. Version Management Policy

본 명세서가 수정될 때마다 버전을 업데이트한다. 이미 완성된 부분에 대한 변경일 경우, 버전 번호를 추가로 업데이트하지 않고 minor number 를 변경하는 것을 원칙으로 한다. 새로운 부분을 추가하거나 문서의 구조가 크게 변경된 경우, major number 를 변경한다.

C. Version Update History

버전	변경 일자	변경 내용
0.1	2019-05-06	초안 작성 및 설계 부분 담당
1.0	2019-05-12	DB 설계 완성
2.0	2019-05-12	로그인 시스템 및 이미지 분석 시스템 설계 완료
2.1	2019-05-16	나머지 서브 시스템 설계 완료
2.2	2019-05-18	서브 시스템 설계 일부 항목 수정
3.0	2019-05-19	Logo 등 Figure 항목 삽입 및 개발 환경 정리 완료
4.0	2019-05-19	목차 및 Index 삽입, 문서 서식 정리 완료
5.0	2019-05-26	DB 설계 일부 수정

Table 1 Version History Table

2 Introduction

2.1 Objective

Introduction 에서는 본 문서의 시스템을 설계하기 위해 사용된 모든 종류의 Diagram 과 Tool 에 대해 설명하고, 본 프로젝트를 통해 개발 중인 'LTE' 시스템의 scope 에 대해 설명한다.

2.2 Applied Diagram

A. UML



Figure 1 UML Logo

The Unified Modeling Language (UML) is a non-patent third-generation modeling and specification language. UML is an open approach to illustrating, visualizing, building, and writing an open approach to an object that is being developed, object-oriented, and software-intensive.

B. Class Diagram

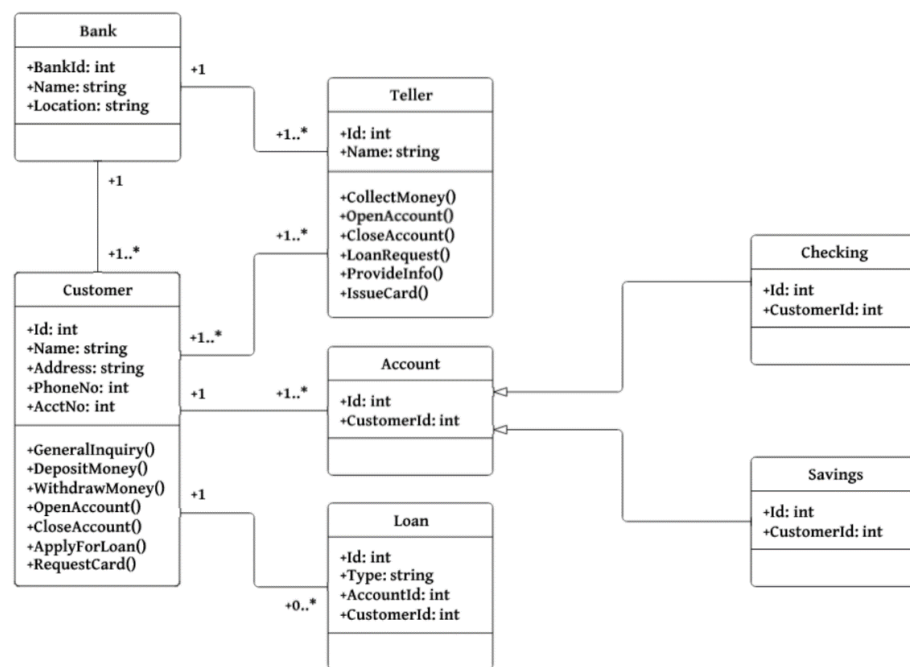


Figure 2 Example of Class Diagram

The class diagram is used for object-oriented modeling. The class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

C. Sequence Diagram

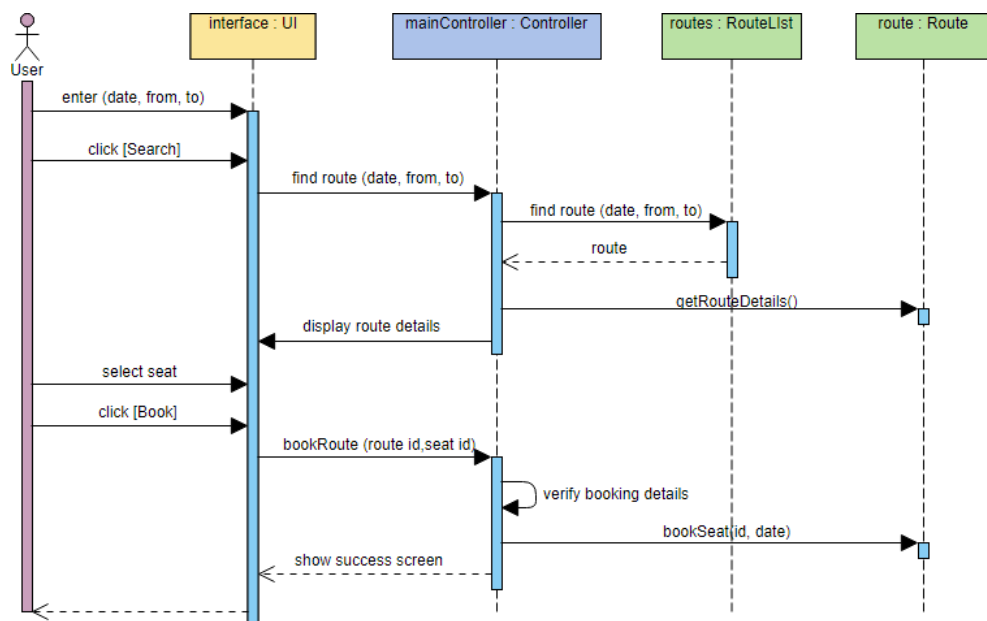


Figure 3 Example of Sequence Diagram

The sequence diagram is a UML interaction diagram. It shows dynamic collaboration between multiple objects by describing the chronological order in which messages are sent between objects.

D. State Diagram

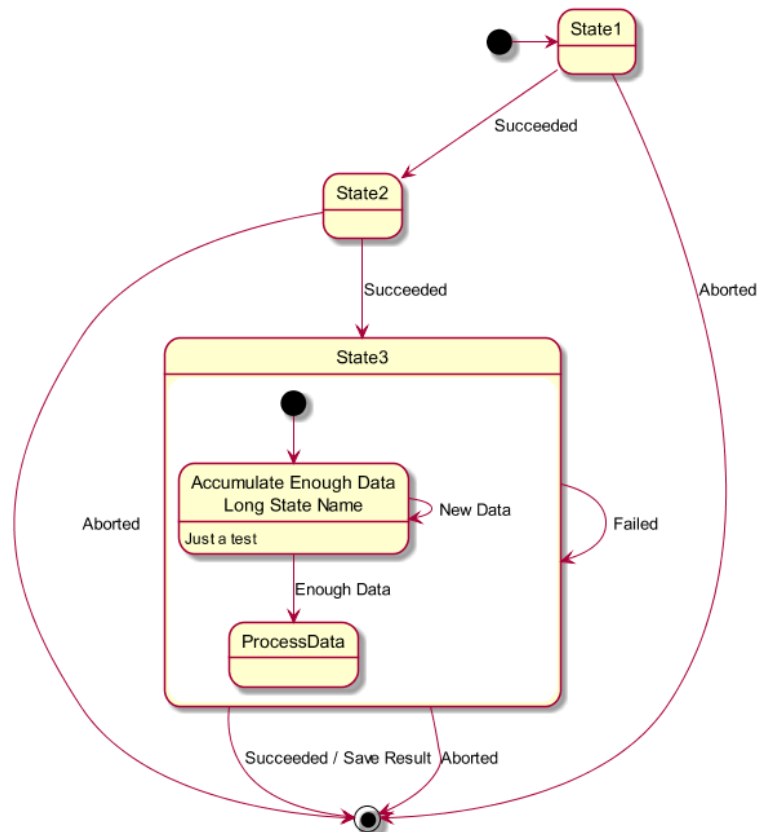


Figure 4 Example of State Diagram

The state diagram is a dynamic behavior that describes an entity's event-based response, showing how the entity reacts to different events based on the current state.

E. ER Diagram

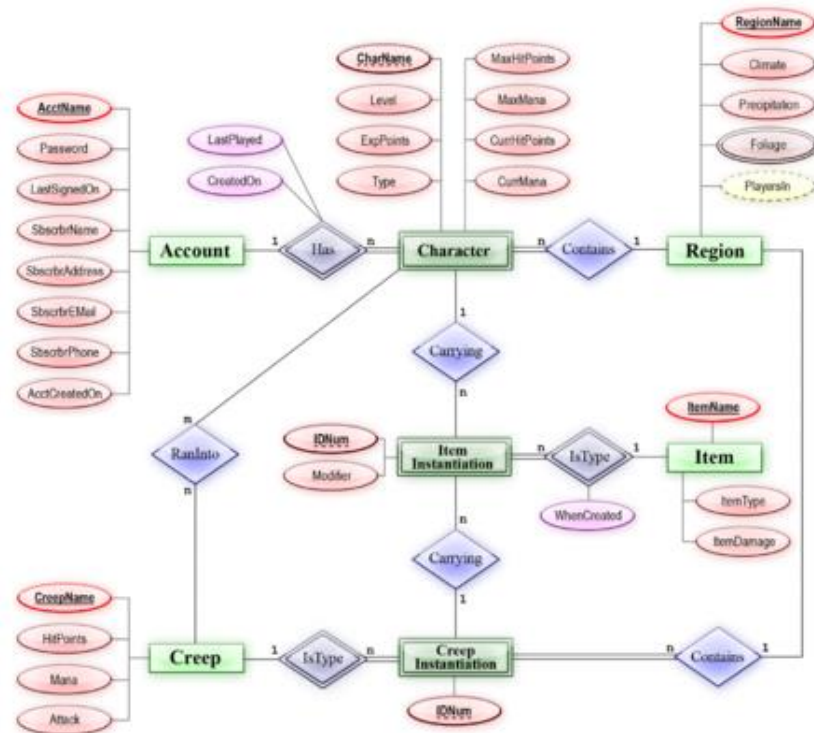


Figure 5 Example of ER Diagram

Typically, a data model is built using an entity-relationship diagram. The entity-contact diagram can be simply referred to as an ER diagram, and the data model depicted by the ER diagram can be referred to as an ER model.

2.3 Applied Tool

A. draw.io



Figure 6 draw.io Logo

draw.io is an open platform where you can create and share diagrams. It's integrated with the tools you already use. By using this platform, you can create UML diagrams, ER models, and other models much easier.

2.4 Project Scope

'LTE'는 사용자가 여행이나 일상 속에서 찍은 사진들의 위치 정보와 해당 여행지에서의 사용자의 실시간 GPS 위치 정보를 이용하여, 사용자의 여행 및 일상 기록들을 지도를 기반으로 하여 정리하는 안드로이드 앱이다. 본 시스템은 총 5 개의 sub-system 으로 구성된다.

로그인 및 회원가입 시스템은 사용자가 해당 시스템에 접근했을 때 발생하는 활동과 관련된 시스템이다. 사용자가 로그인이나 회원가입을 시도할 때, 사용자에게 적합한 정보를 입력 받고, 그 정보는 user DB 를 통해 관리된다.



Figure 7 로그인 및 회원가입 시스템

지도 시스템은 사용자가 본 시스템에 업로드한 사진의 위치가 핀의 형태로 나타나고 실시간으로 기록한 여행 경로가 지도를 기반으로 보여지는 지도 관련 시스템을 의미한다. 지도 위의 정보들은 사진 DB 를 통해 관리한다.

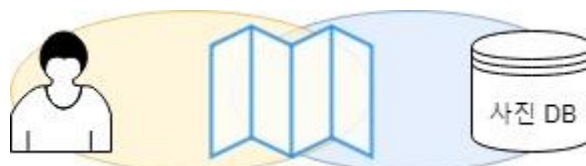


Figure 8 지도 시스템

타임스탬프 기록 시스템은 사용자가 자신의 현재 위치 및 시간 정보를 입력하고 싶을 때, 타임스탬프 기록 버튼을 누르면 사용자의 현재 GPS 위치를 기반으로 시간 및 위치 정보가 지도 위에 핀으로 기록되는 시스템이다. 해당 기록은 사진 DB 를 통해 관리한다.



Figure 9 타임스탬프 기록 시스템

사진 등록 시스템은 사용자가 사진 업로드, 수정, 삭제를 진행하는 시스템으로, 사진 DB 를 기반으로 정보를 추가 및 변경, 삭제를 진행한다.

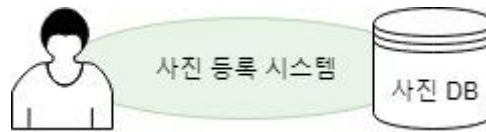


Figure 10 사진 등록 시스템

이미지 분석 시스템은 다른 말로는 '해시태그 시스템'이라고도 부르는데, 사용자가 업로드하는 사진 파일을 입력 값으로 받아 해당 이미지를 분석하여 라벨을 추출하고 해당 라벨을 해시태그로 사용자에게 추천하는 시스템이다. 사용자가 선택한 해시태그는 사진 DB 에 다른 사진 정보들과 함께 관리된다.



Figure 11 해시태그 시스템

3 System Architecture

3.1 Objective

System Architecture 에서는 팀에서 개발하고자 하는 LTE 시스템에 대한 전반적인 사항들을 설명한다. LTE 시스템의 전반적인 구조를 block diagram 을 통해 나타낸 후, 이를 class diagram 과 sequence diagram, state diagram 으로 보다 상세히 기술한다.

3.2 System Organization

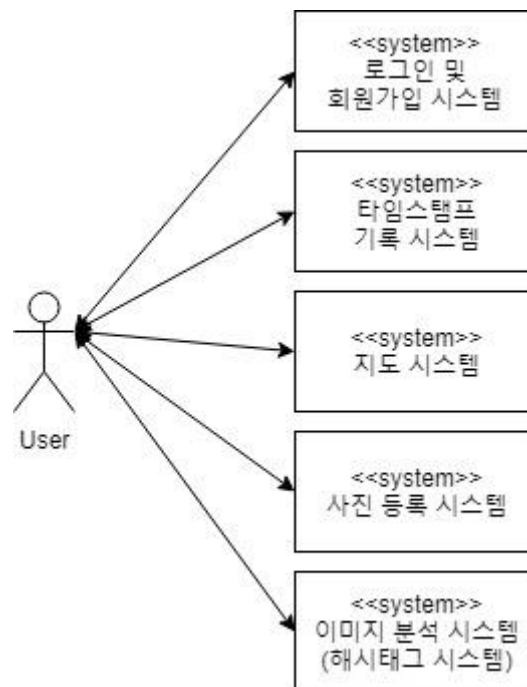


Diagram 1 Block Diagram of LTE System

'LTE' 시스템은 사용자가 안드로이드 앱이라는 UI 를 통해 각각의 sub-system 과 상호작용하고, 해당 상호작용을 통해 주고받은 데이터는 서버에 연결된 DB 에 저장 및 변경, 삭제되는 Client-Server 모델 구조를 바탕으로 하고 있다.

해당 시스템은 로그인 및 회원가입 시스템, 타임스탬프 기록 시스템, 지도 시스템, 사진 등록 시스템, 이미지 분석 시스템(해시태그 시스템), 이렇게 총 5 개의 서브 시스템으로 이루어져 있다.

A. 로그인 및 회원가입 시스템

로그인 및 회원가입 시스템은 본 시스템에 사용자가 접근했을 때 최초 접근인지, 기존 사용자인지를 판별하는 시스템이다. 사용자에게서 디바이스를 통해 로그인

정보를 받으면, 해당 데이터를 서버로 넘겨 User DB 의 데이터와 대조해본다. 존재하는 정보와 일치할 경우, 로그인을 허락한다. 회원가입인 경우, 유저의 회원가입 데이터를 받아 해당 데이터를 서버로 넘겨 User DB 와 확인하여, 해당 가입 정보가 User DB 에 있는 데이터와 중복되지 않는다면 회원가입을 승인한다.

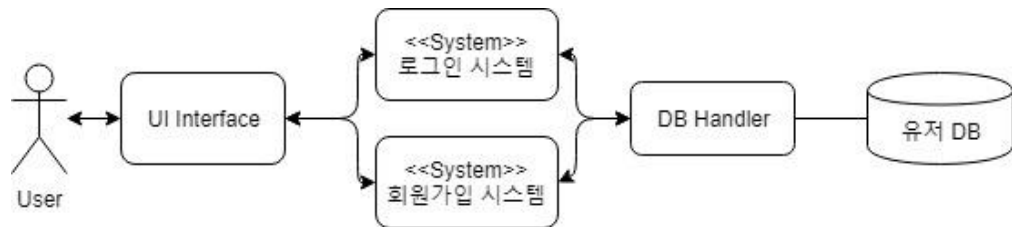


Diagram 2 Block Diagram of Login & Sign Up System

B. 지도 시스템

사용자의 로그인이 완료되고 나면, 핀이나 경로와 같은, 사용자가 저장해 둔 데이터들이 지도 위에 노출된다. 해당 시스템에서 사진과 경로를 기록할 수 있으며, 타 기능들을 호출하는 프론트 역할을 한다.

입력된 사진 기록은 사진 정보로 분류되어 서버에 진입하게 되고 해당 사진 정보는 사진 DB 에 저장된다. 이때 사진 정보를 변경 및 삭제하게 될 경우, 사진 DB 에 접근하여 해당 정보들을 수정 및 삭제할 수 있고 이는 지도 시스템 내에 있는 모든 화면에 반영된다.

경로 데이터는 경로 기록이 시작될 때 Local DB 에 저장되다가 경로 기록이 종료되면 Local DB 에 가지고 있던 정보를 사진 DB 에 업로드한다.

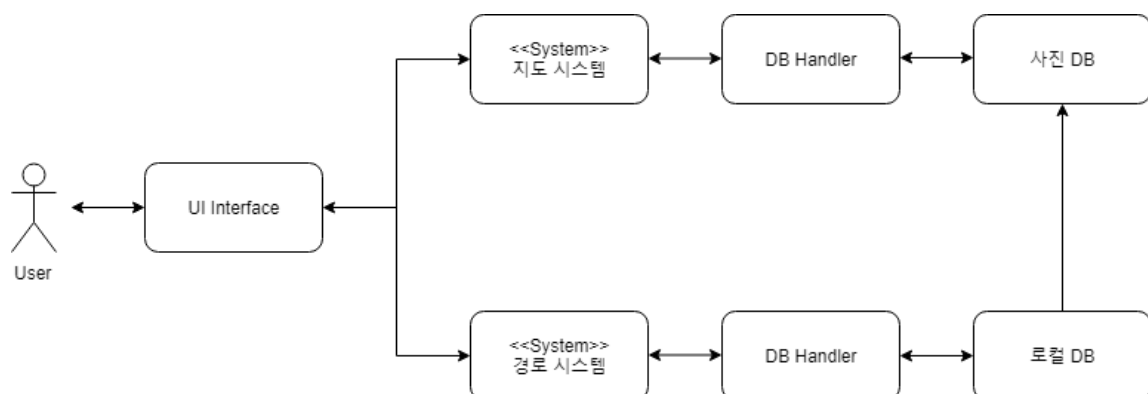
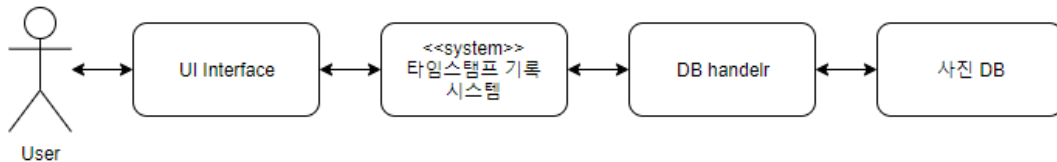


Diagram 3 Block Diagram of Map System

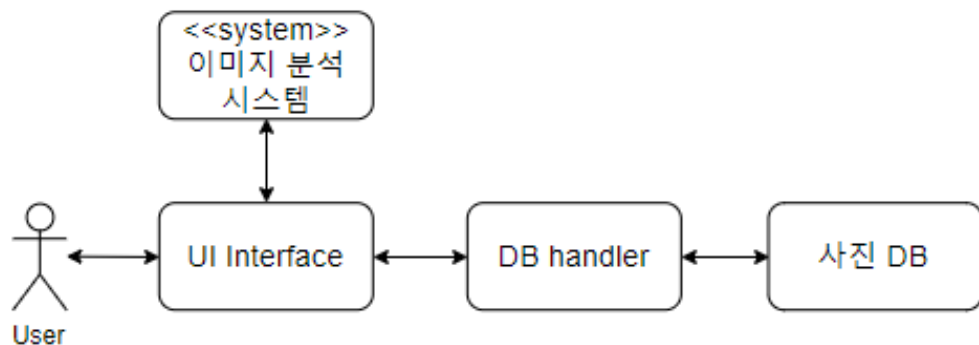
C. 타임스탬프 기록 시스템

사용자의 로그인이 완료되고 난 후, 사용자가 타임스탬프 기록 버튼을 누르면 사용자의 디바이스가 현재 시각, 현재 위치 정보를 서버에 전달하여 해당 정보를 사진 DB 에 저장한다. 그리고 DB 에 저장된 위치 정보 및 시간 정보는 다시 서버에서 디바이스로 전달되어 지도 위에 핀으로 기록된다.

**Diagram 4 Block Diagram of Timestamp System****D. 사진 등록 시스템**

사용자가 등록하고 싶은 사진을 고르면 사진에 저장되어 있는 위치정보를 입력 받아 지도에 핀으로 표현해주고 이미지 분석을 위해 사진 정보를 넘겨준다. 이미지 분석 시스템으로부터 해시태그를 추천 받은 뒤 해시태그, 별점 정보를 입력 받고 서버를 통해 사진 DB 에 입력된다.

또한 사용자가 기존에 등록한 사진, 해시태그, 별점을 변경하거나 삭제할 수 있도록 하여 사용자가 기존 내역을 수정할 수 있도록 한다. 사진을 추가적으로 업로드 시 기존의 사진 업로드 방식과 같이 디바이스의 갤러리를 통해 DB 에 업데이트 하게 된다. 수정 내역은 서버를 통해 사진 DB 에 업데이트 된다.

**Diagram 5 Block Diagram of Uploading Picture System****E. 이미지 분석 시스템 (해시태그 시스템)**

사용자가 사진을 업로드하면, 해당 사진 데이터는 이미지 분석 시스템에 전달되고, 분석을 통해 이미지에 있는 라벨을 감지하여 이를 토대로 해당 사진에 맞는 해시태그를 추천해 준다. 사용자가 추천된 해시태그들 중 해시태그를 선택해

결정하면 사진 데이터는 지정된 해시태그 정보와 함께 사진 DB에 저장되고, 이후 사진 DB에 있는 사진 정보들을 정렬할 때 해시태그 기준으로 정렬하여 볼 수 있다.

사진 정보를 변경 및 삭제하고자 할 때 해시태그 정보 역시 수정 가능하며, 수정된 정보는 다시 서버로 전달되어 사진 DB에 반영된다.

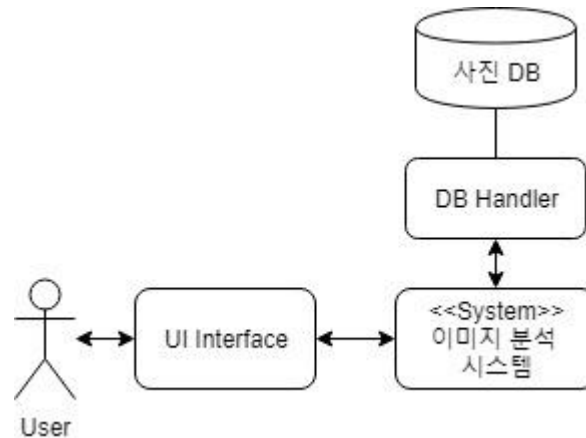


Diagram 6 Block Diagram of Image Analysis System

4 로그인 및 회원가입 시스템

4.1 Objective

사용자가 처음으로 본 시스템에 접근할 때 입력하는 사용자 정보를 처리하는 시스템으로, 로그인과 회원가입의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

4.2 Class Diagram

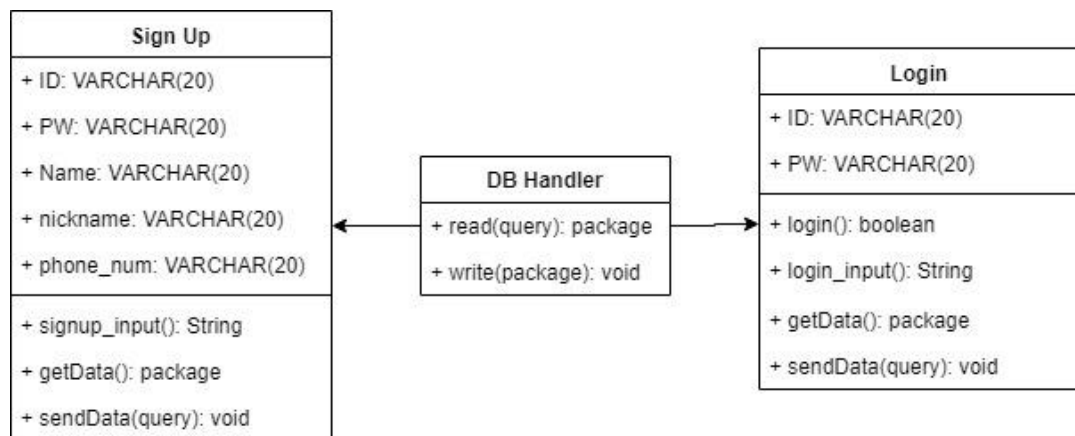


Diagram 7 Class Diagram of Login & Sign Up System

A. Sign Up

1) Attributes

- ID: 로그인 시 사용하게 될 사용자의 ID
- PW: 로그인 시 사용하게 될 사용자의 비밀번호
- Name: 사용자의 실명
- nickname: 사용자가 본 시스템 내에서 사용하게 될 별명
- phone_num: 사용자의 전화번호 (핸드폰 번호)

2) Methods

- String signup_input(): 정해진 폼에 맞추어 회원가입 정보를 입력 받는다.
- package getData(): 입력 받은 회원가입 정보를 가져온다.
- void sendData(query): User DB 에 데이터를 보낸다.

B. DB Handler

1) Attributes

해당 사항 없음

2) Methods

- package read(query): User DB 에서 사용자 데이터를 읽어온다.
- void write(package): User DB 에 사용자 데이터를 저장한다.

C. Login

1) Attributes

- ID: 사용자가 등록한 ID
- PW: 사용자가 등록한 비밀번호

2) Methods

- boolean login(): 읽어 온 사용자 데이터가 입력된 데이터와 일치하는지 확인한다.
- String login_input(): ID 와 PW 값을 사용자로부터 입력 받는다.
- package getData(): 입력 받은 로그인 정보를 가져온다.
- void sendData(query): 입력 받은 로그인 정보를 User DB 에 보낸다.

4.3 Sequence Diagram

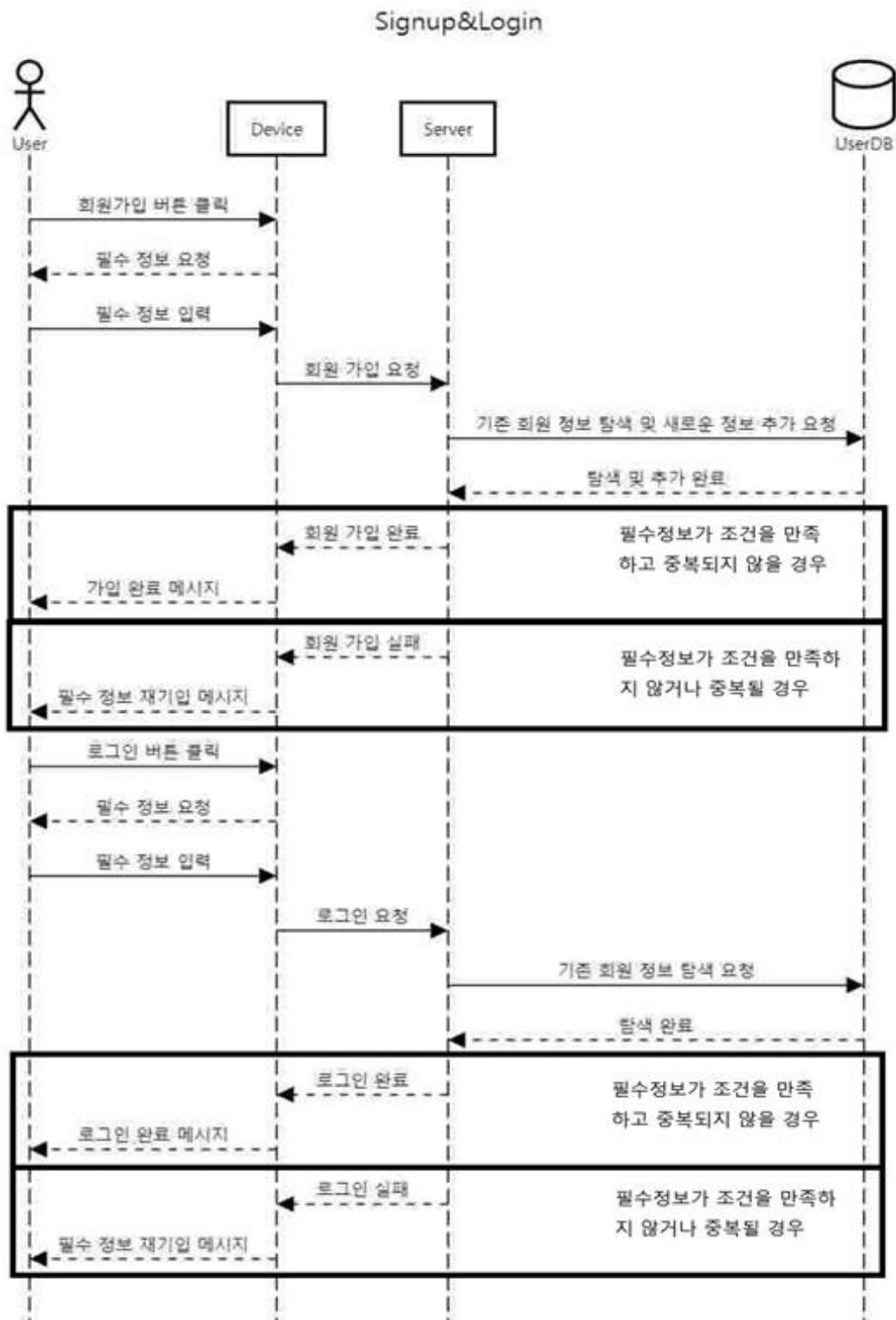


Diagram 8 Sequence Diagram of Login & Sign Up System

4.4 State Diagram

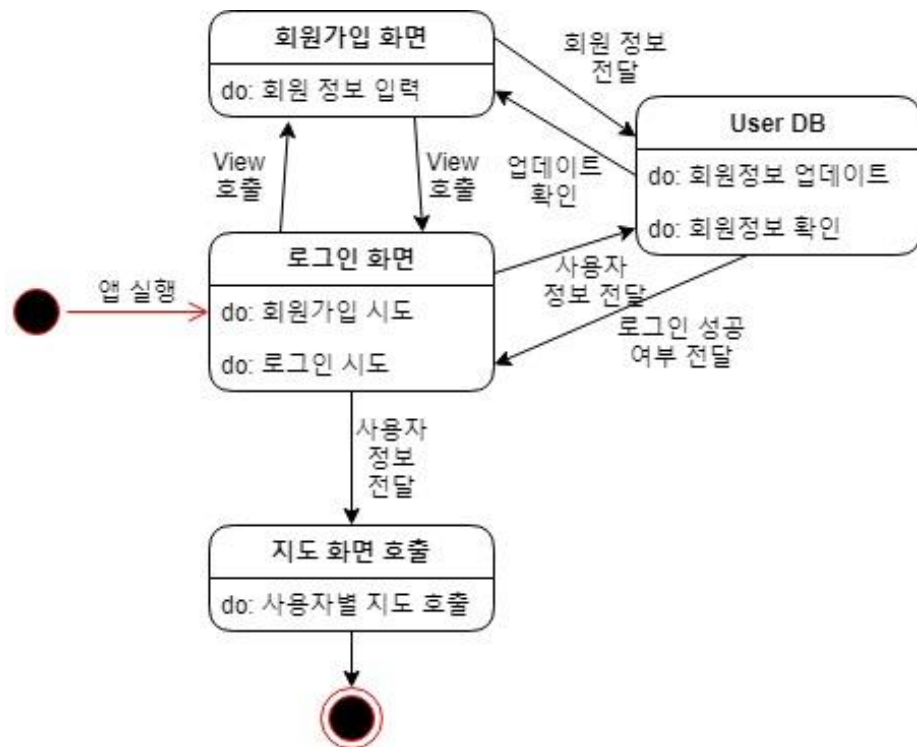


Diagram 9 State Diagram of Login & Sign Up System

5 지도 시스템

5.1 Objective

로그인한 사용자가 사진과 경로 정보를 입력하면 지도 위에 데이터를 처리하는 프론트 역할의 시스템이다. 지도 시스템의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

5.2 Class Diagram

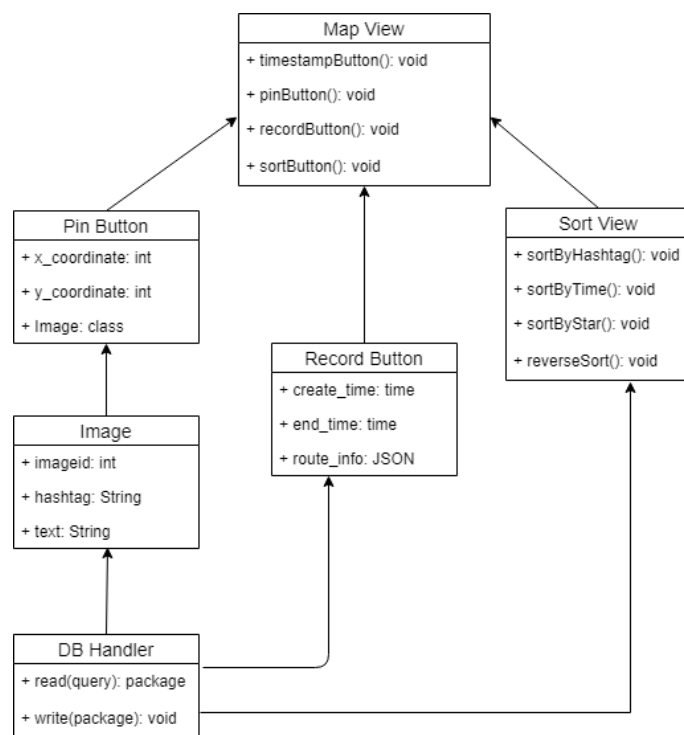


Diagram 10 Class Diagram of Map System

A. Map View

1) Attributes

해당 사항 없음

2) Methods

- + `void timestampButton()`: 지도 화면에서 타임스탬프 버튼을 클릭해 활성화한다.
- + `void pinButton()`: 지도 화면에서 핀 버튼을 클릭해 활성화한다.
- + `void recordButton()`: 지도 화면에서 경로 기록 버튼을 클릭해 활성화한다.
- + `void sortButton()`: 지도 화면에서 정렬 버튼을 클릭해 활성화한다.

B. Pin Button

- 1) Attributes
 - + x_coordinate: 지도 x 좌표값
 - + y_coordinate: 지도 y 좌표값
 - + Image: 사진 데이터
- 2) Methods
 - 해당 사항 없음

C. Image

- 1) Attributes
 - + imageid: 사진 id
 - + hashtag: 사진 해시태그
 - + text: 사진 설명
- 2) Methods
 - 해당 사항 없음

D. Sort View

- 1) Attributes
 - 해당 사항 없음
- 2) Methods
 - + void sortByHashtag(): 데이터를 해시태그로 정렬한다.
 - + void sortByTime(): 데이터를 등록한 시간으로 정렬한다.
 - + void sortByStar(): 데이터를 별점으로 정렬한다.
 - + void reverseSort(): 정렬 순서를 역순으로 전환한다.

E. Record Button

- 1) Attributes
 - + create_time: 경로 기록 시작 시간
 - + end_time: 경로 기록 종료 시간
 - + route_info: 경로 정보 JSON
- 2) Methods
 - 해당 사항 없음

F. DB Handler

- 1) Attributes
 - 해당 사항 없음
- 2) Methods

- + package read(query): 사진 DB 에서 사진 데이터를 읽어온다.
- + void write(package): Local DB 및 사진 DB 에 데이터를 저장한다.

5.3 Sequence Diagram

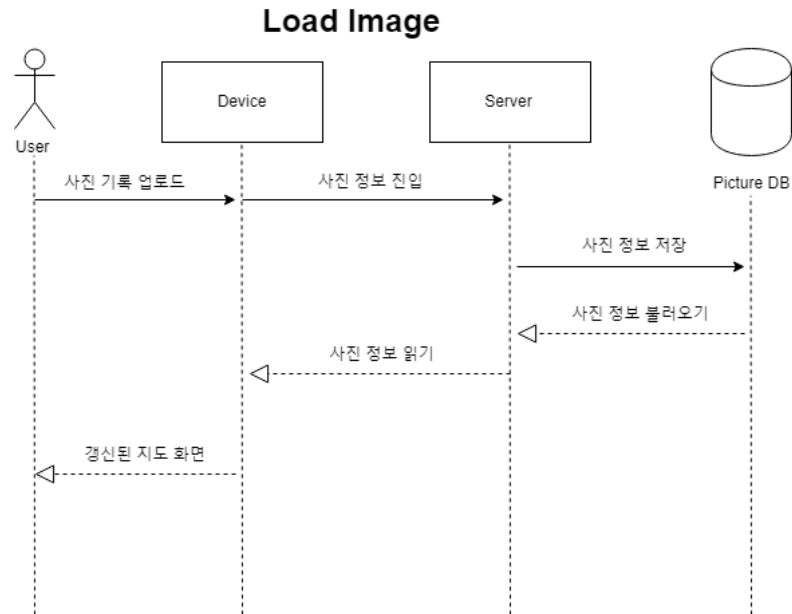


Diagram 11 Sequence Diagram of Loading Image System

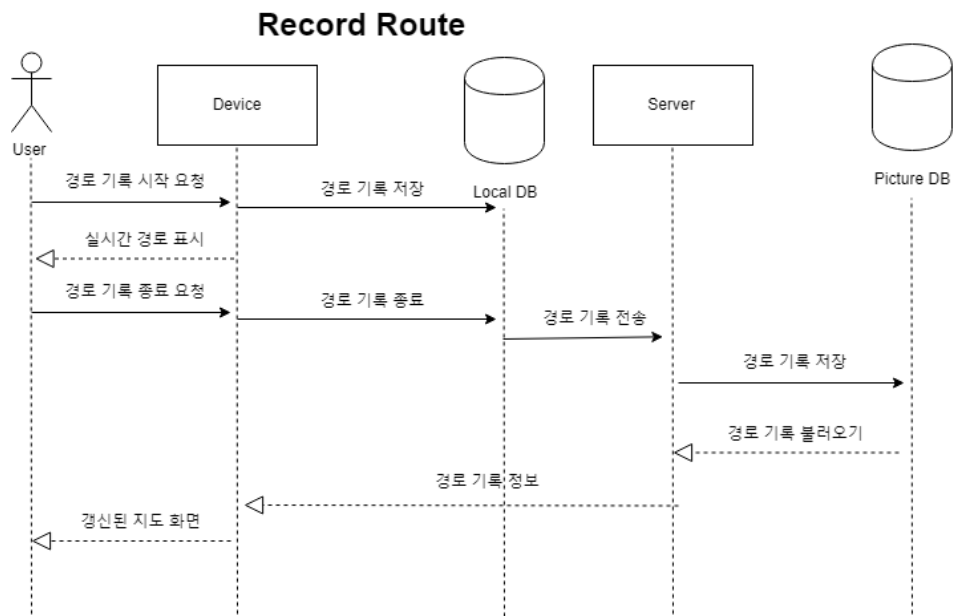


Diagram 12 Sequence Diagram of Recording Route System

5.4 State Diagram

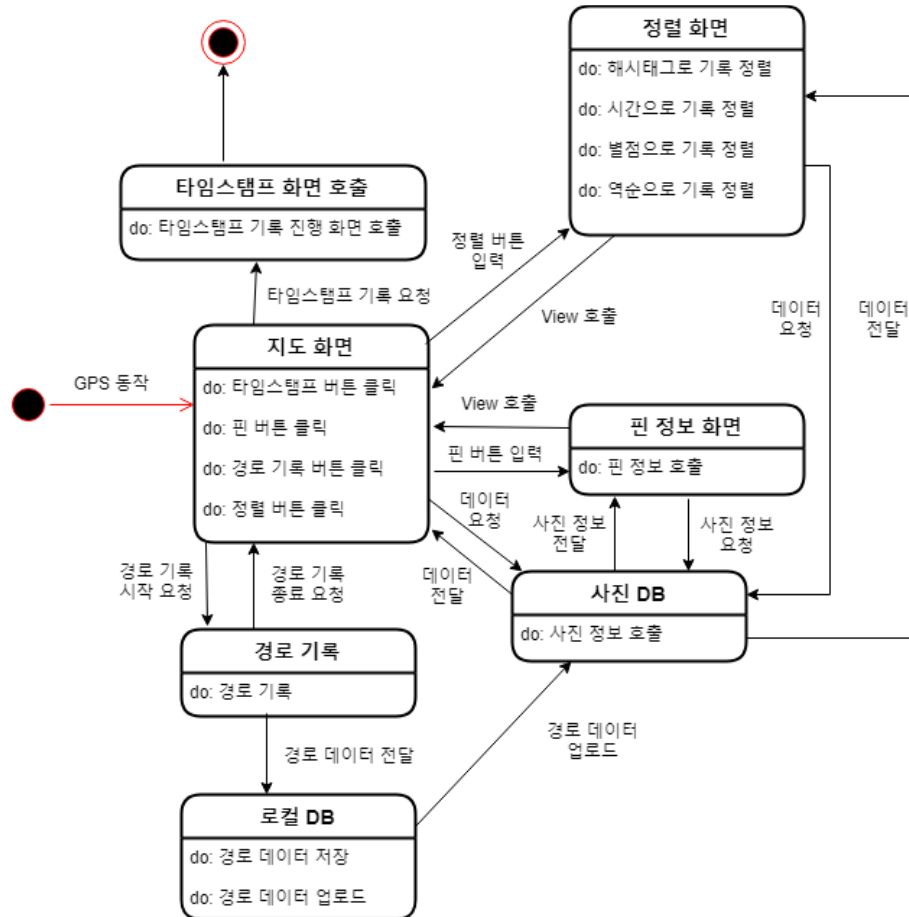


Diagram 13 State Diagram of Map System

6 타임스탬프 기록 시스템

6.1 Objective

사용자가 입력하는 시간, 위치 정보를 처리하는 시스템으로, 타임스탬프 기록의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

6.2 Class Diagram

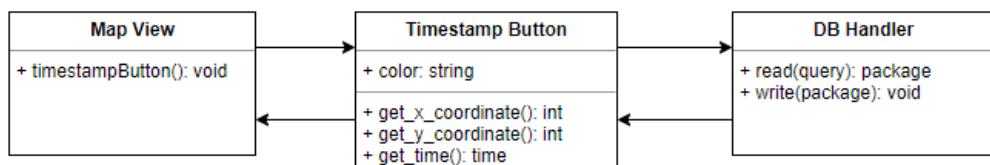


Diagram 14 Class Diagram of Timestamp System

A. Map View

1) Attributes

해당 사항 없음

2) Methods

+timestampButton(): 지도 화면에서 타임스탬프 버튼을 클릭해 활성화한다.

B. Timestamp Button

1) Attributes

+color: 타임스탬프 핀이 가지는 색. 핀이 가지고 있는 정보에 따라 색이 다르다.

2) Methods

+get_x_coordinate(): 위치정보를 가져올 때 x 좌표를 가져온다.

+get_y_coordinate(): 위치정보를 가져올 때 y 좌표를 가져온다.

+get_time(): 시간 정보를 가져온다.

C. DB Handler

1) Attributes

해당 사항 없음

2) Methods

+read(query): 사진 DB 에서 정보를 가져온다.

+write(package): 사진 DB 에 정보를 입력한다.

6.3 Sequence Diagram

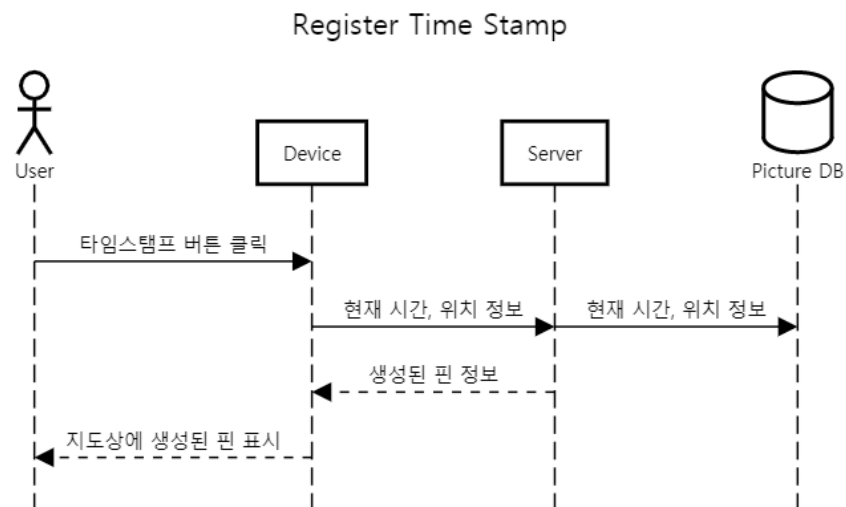


Diagram 15 Sequence Diagram of Registering Timestamp

6.4 State Diagram

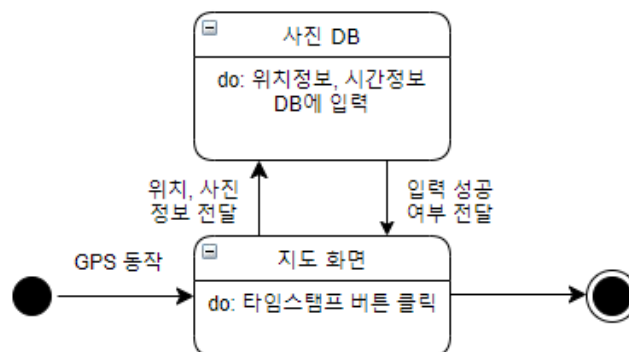


Diagram 16 State Diagram of Registering Timestamp

7 사진 등록 시스템

7.1 Objective

사용자가 사진 업로드, 수정, 삭제를 진행하는 시스템으로, 사진 업로드와 수정, 삭제의 설계 구조를 class diagram, sequence diagram, state diagram 을 통해 설명한다.

7.2 Class Diagram

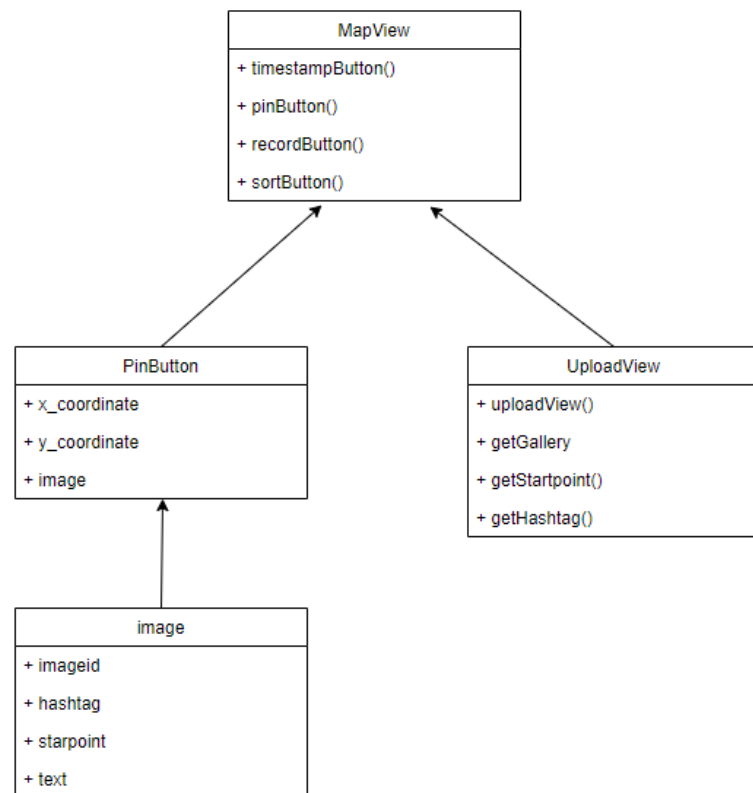


Diagram 17 Class Diagram of Uploading Picture System

A. Map View

1) Attributes

해당 사항 없음

2) Methods

+ void timestampButton(): 지도 화면에서 타임스탬프 버튼을 클릭해 활성화한다.

- + void pinButton(): 지도 화면에서 핀 버튼을 클릭해 활성화한다.
- + void recordButton(): 지도 화면에서 경로 기록 버튼을 클릭해 활성화한다.
- + void sortButton(): 지도 화면에서 정렬 버튼을 클릭해 활성화한다.

B. Pin Button

1) Attributes

- + x_coordinate: 지도 x 좌표값
- + y_coordinate: 지도 y 좌표값
- + Image: 사진 데이터

2) Methods

해당 사항 없음

C. Image

1) Attributes

- + imageid: 사진 id
- + hashtag: 사진 해시태그
- + starpoint: 사진 별점
- + text: 사진 설명

2) Methods

해당 사항 없음

D. Upload View

1) Attributes

해당 사항 없음

2) Methods

- + void uploadView(): Image 정보를 사진 DB 에 저장하기 위해 서버로 전송한다.
- + void getGallery(): 디바이스의 갤러리로부터 사진들을 가져온다.
- + void getStarpoint(): 사용자로부터 별점을 입력 받는다.
- + void getHashtag(): 사용자로부터 해시태그들을 입력 받는다.

7.3 Sequence Diagram

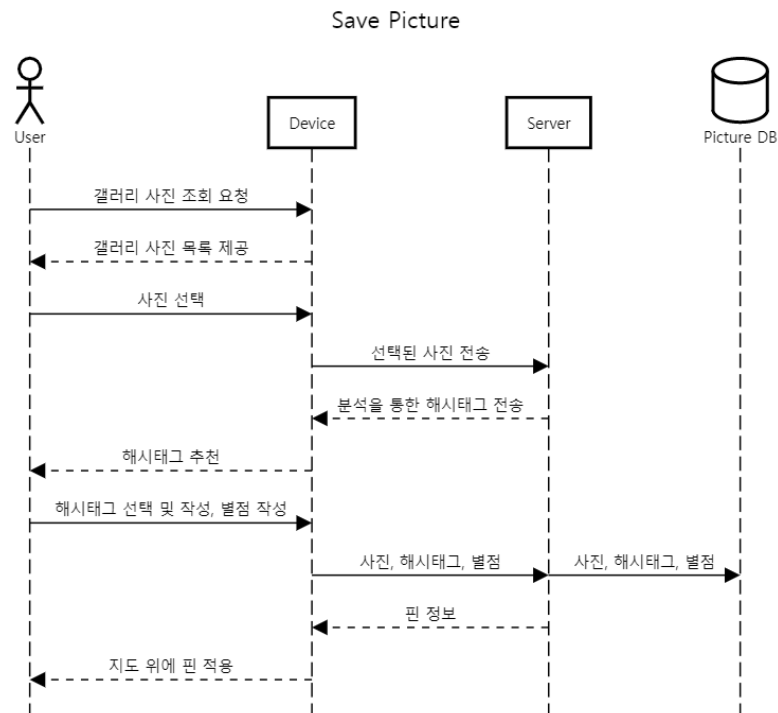


Diagram 18 Sequence Diagram of Saving Picture

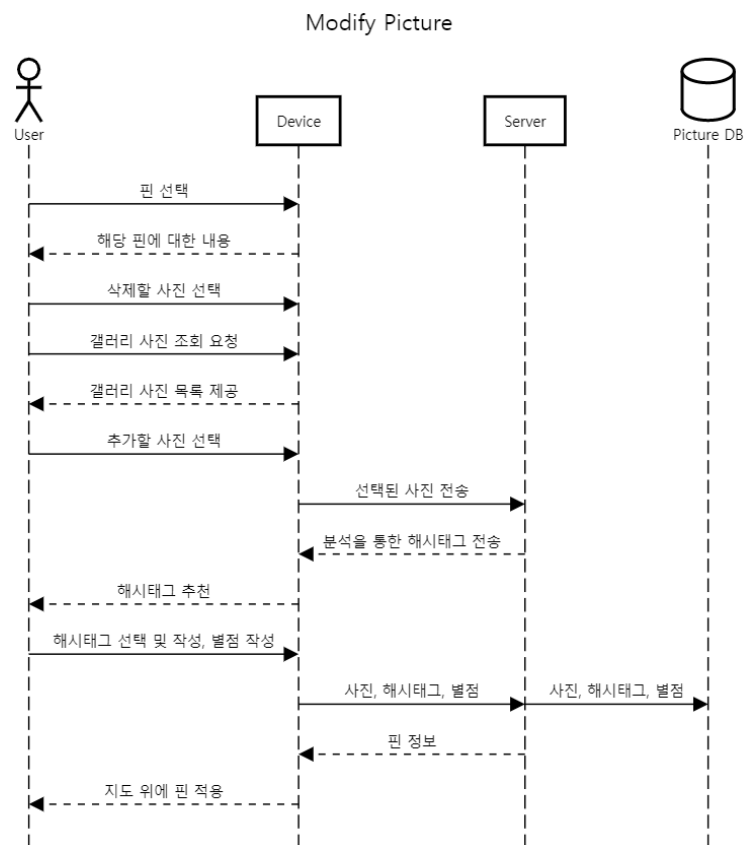


Diagram 19 Sequence Diagram of Modifying Picture

7.4 State Diagram

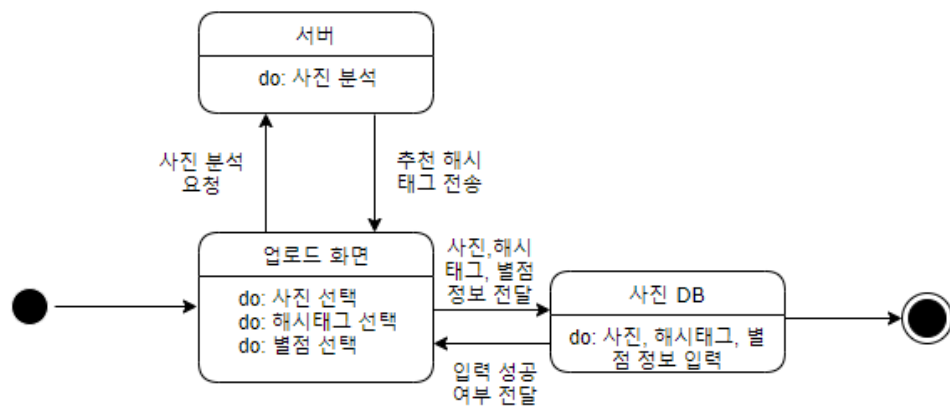


Diagram 20 State Diagram of Uploading Picture System

8 이미지 분석 시스템 (해시태그 시스템)

8.1 Objective

사용자가 업로드하는 사진 파일을 입력 값으로 받아 해당 이미지를 분석하여 라벨을 추출하고 해당 라벨을 해시태그로 사용자에게 추천하는 시스템으로, 해당 시스템의 설계 구조에 대해 class diagram, sequence diagram, state diagram 을 통해 설명한다.

8.2 Class Diagram

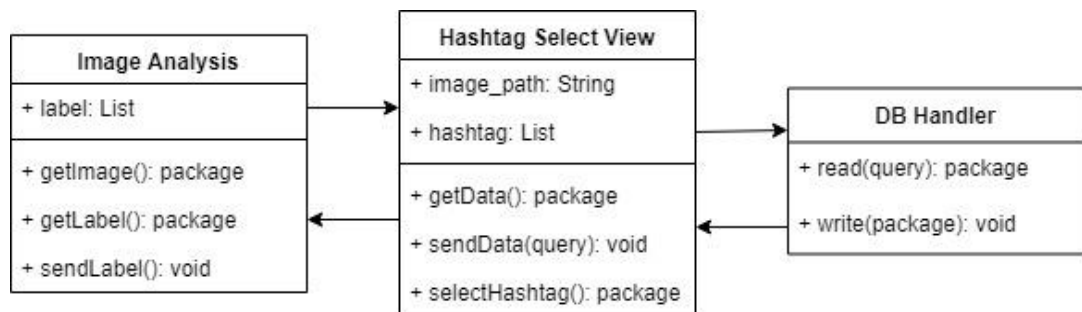


Diagram 21 Class Diagram of Image Analysis System

A. Image Analysis

1) Attributes

- label: 이미지 분석 시스템을 통해 감지된 해당 이미지의 라벨

2) Methods

- package getImage(): 사용자가 입력한 이미지를 받아온다.
- package getLabel(): 분석 시스템을 통해 감지된 라벨을 받아온다.
- void sendLabel(): 감지된 라벨을 사용자에게 해시태그로 추천한다.

B. Hashtag Select View

1) Attributes

- image_path: 사진 DB 에 저장되어 있는 서버에 저장된 이미지의 경로
- hashtag: 추천받은 해시태그

2) Methods

- package getData(): 분석 시스템을 통해 얻은 라벨을 얻어 온다.
- void sendData(query): 선택된 해시태그를 사진 DB 로 전달한다.
- package selectHashtag(): 사용자가 해시태그를 선택한다.

C. DB Handler

1) Attributes

해당 사항 없음

2) Methods

- package read(query): 사진 DB 에서 정보를 가져온다.
- void write(package): 사진 DB 에 정보를 전달한다.

8.3 Sequence Diagram

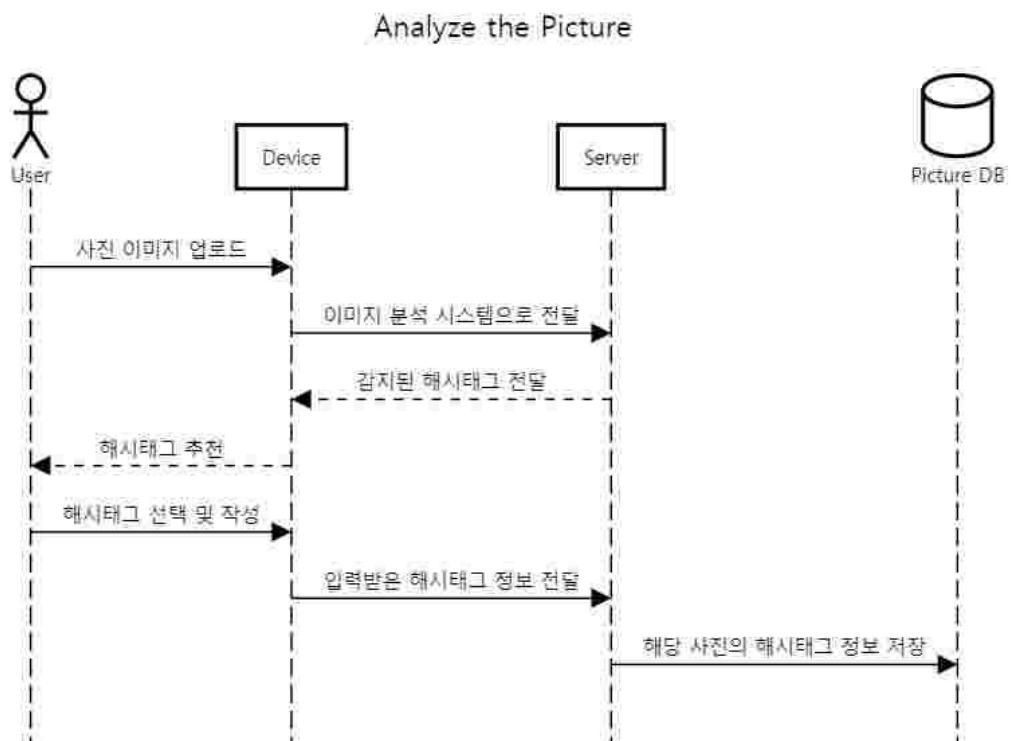


Diagram 22 Sequence Diagram of Analyzing Picture

8.4 State Diagram

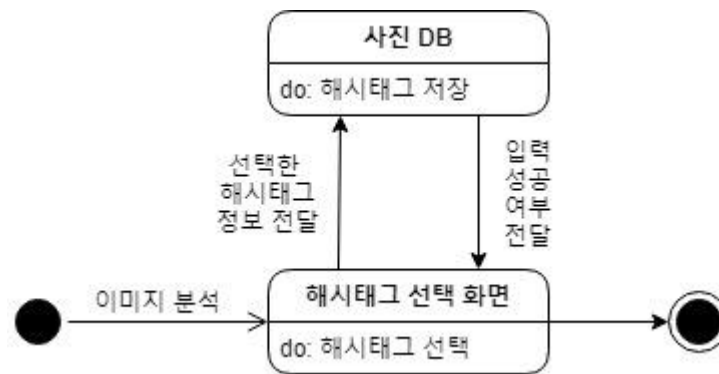


Diagram 23 State Diagram of Image Analysis System

9 Protocol Design

9.1 Objective

Protocol Design에서는 서브시스템 간 상호 통신을 위한 프로토콜에 대해 서술한다. 해당 프로토콜은 원활한 통신을 위해 필수적으로 준수되어야 하며, 이때 전달되는 메시지의 형식과 용도, 의미를 설명한다.

9.2 JSON

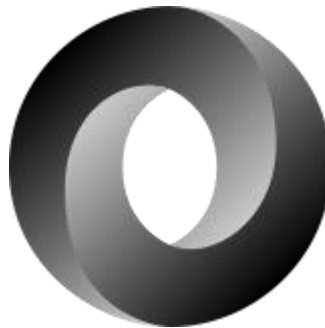


Figure 12 JSON Logo

JavaScript Object Notation(JSON)은 속성과 값의 쌍으로 이루어진 데이터 객체를 사람이 이해가능한 언어로 전달하는 데에 사용하는 표준 형식이다. 인터넷에서 자료를 주고받을 때 그 자료를 표현하는 방법으로 널리 쓰이고 있으며, 자료의 종류에 제한이 없고, 컴퓨터 프로그램의 변수 값을 표현하는 데에 특화되어 있다.

9.3 Protocol Description

A. Overview

Client 와 server 사이에서 오고 가는 요청(Request) - 응답(Response)의 메시지의 형태를 정의하고, 그 내용을 설명한다.

B. Login Protocol

1) Request

Attribute	Value
ID	사용자의 ID
PW	사용자의 비밀번호(password)

Table 2 Login Protocol Request

2) Response

Attribute	Value
login_success	로그인 성공 여부

Table 3 Login Protocol Response

C. Sign Up Protocol

1) Request

Attribute	Value
ID	사용자의 ID
PW	사용자의 비밀번호(password)
Name	사용자의 실명
nickname	사용자의 별명(닉네임)
phone_num	사용자의 전화번호(폰 번호)

Table 4 Sign Up Protocol Request

2) Response

Attribute	Value
signup_success	회원가입 성공 여부

Table 5 Sign Up Protocol Response

D. Map View Protocol

1) Request

Attribute	Value
Image	사진 정보 클래스
imageid	사진 id
hashtag	사진 해시태그
text	사진 설명

Table 6 Map View Protocol Request

2) Response

Attribute	Value
mapload_success	지도 불러오기 성공 여부

Table 7 Map View Protocol Response

E. Route Protocol

1) Request

Attribute	Value
create_time	경로 기록 시작 시간
end_time	경로 기록 종료 시간
route_info	경로 정보 JSON

Table 8 Route Protocol Request

2) Response

Attribute	Value
routeload_success	경로 불러오기 성공 여부

Table 9 Route Protocol Response

F. Timestamp Protocol

1) Request

Attribute	Value
x_coordinate	사용자의 현재 x 좌표
y_coordinate	사용자의 현재 y 좌표
time	사용자의 현재 시간

Table 10 Timestamp Protocol Request

2) Response

Attribute	Value
timestamp_success	타임스탬프 성공 여부
pin_color	타임스탬프 핀 색깔

Table 11 Timestamp Protocol Response

G. Save, Modify Picture Protocol

1) Request

Attribute	Value
-----------	-------

ID	사용자의 ID
x_coordinate	사용자의 현재 x 좌표
y_coordinate	사용자의 현재 y 좌표
Image	사진 데이터
imageid	사진 ID
hashtag	사진 해시태그
starpoint	사진 별점
text	사진 설명

Table 12 Save, Modify Picture Protocol Request

2) Response

Attribute	Value
save_success	저장, 수정 성공여부

Table 13 Save, Modify Picture Protocol Response

H. Image Analysis Protocol

1) Request

Attribute	Value
image_path	서버에 저장된 이미지 경로

Table 14 Image Analysis Protocol Request

2) Response

Attribute	Value
label	이미지에서 감지된 라벨 리스트

Table 15 Image Analysis Protocol Response

I. Hashtag Protocol

1) Request

Attribute	Value
hashtag	사용자가 선택한 해시태그 리스트

Table 16 Hashtag Protocol Request

2) Response

Attribute	Value
hashtag_success	해시태그 저장 성공 여부

Table 17 Hashtag Protocol Response

10 Database Design

10.1 Objective

Database Design 은 시스템에 필요한 데이터베이스를 설계하는 것을 목표로 한다. 우선적으로 데이터베이스의 전체 흐름을 보기 위한 ER Diagram 을 작성하고, 이를 활용하여 Relational Schema 와 SQL DDL 을 작성한다.

10.2 ER Diagram

ER Diagram 에서, 객체(Entity)는 직사각형으로, 속성(Attribute)은 타원으로, 관계(Relation)는 마름모 모양으로 표현된다. 각 Entity 의 Primary Key 에는 밑줄이 쳐져 있으며, Multi-valued Attribute 는 이중 타원으로 표현한다.

본 시스템의 전체 ER Diagram 은 다음과 같다.

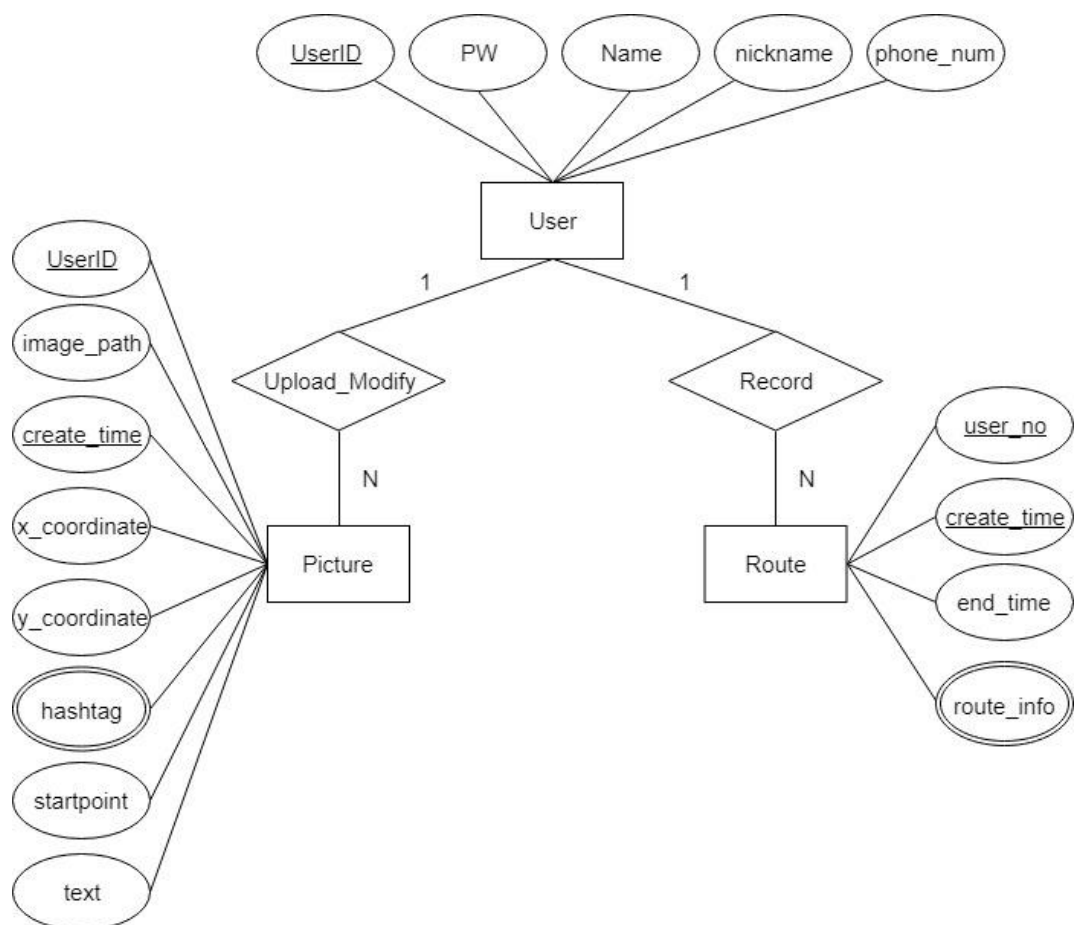


Figure 13 ER Diagram of LTE System

A. Entity

1) User

User 는 사용자 정보를 나타낸다. user_no, ID, PW, Name, nickname, phone_num 을 속성으로 가지고 있으며, Primary Key 는 user_no 이다.

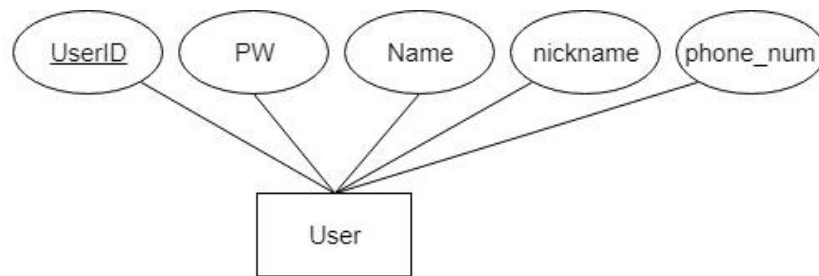


Diagram 24 User ER Diagram

2) Picture

Picture 는 사용자가 업로드한 사진의 정보를 나타낸다. user_no, image_id, image_path, create_time, x_coordinate, y_coordinate, hashtag, starpoint, text 를 속성으로 가지며, Primary Key 는 user_no, image_id 이다.

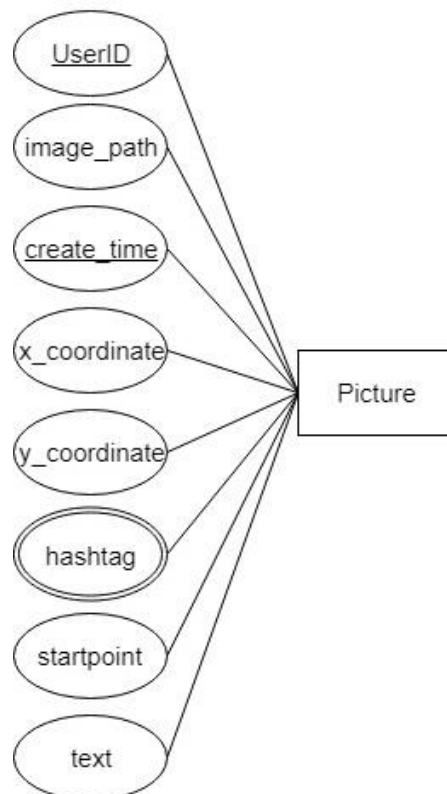


Diagram 25 Picture ER Diagram

3) Route

Route 는 사용자가 기록한 경로 데이터를 의미한다. user_no, route_id, create_time, end_time, route_info 를 속성으로 가지고 있으며, Primary Key 는 user_no, route_id 이다.

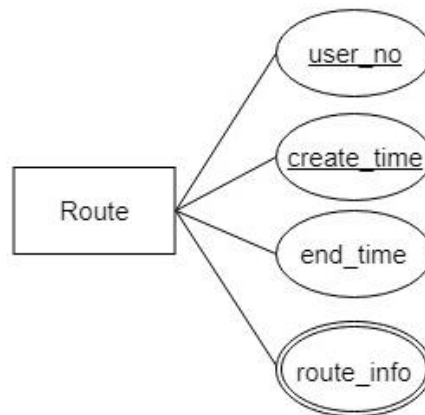


Diagram 26 Route ER Diagram

B. Relationship

1) Upload_Modify

Upload_Modify Relationship 은 사용자(User)가 사진(Picture) 정보를 업로드하거나 수정할 때의 관계로, 각각의 사진은 단 한 명의 사용자에게 포함되며, 한 명의 사용자는 여러 개의 사진을 가질 수 있다.

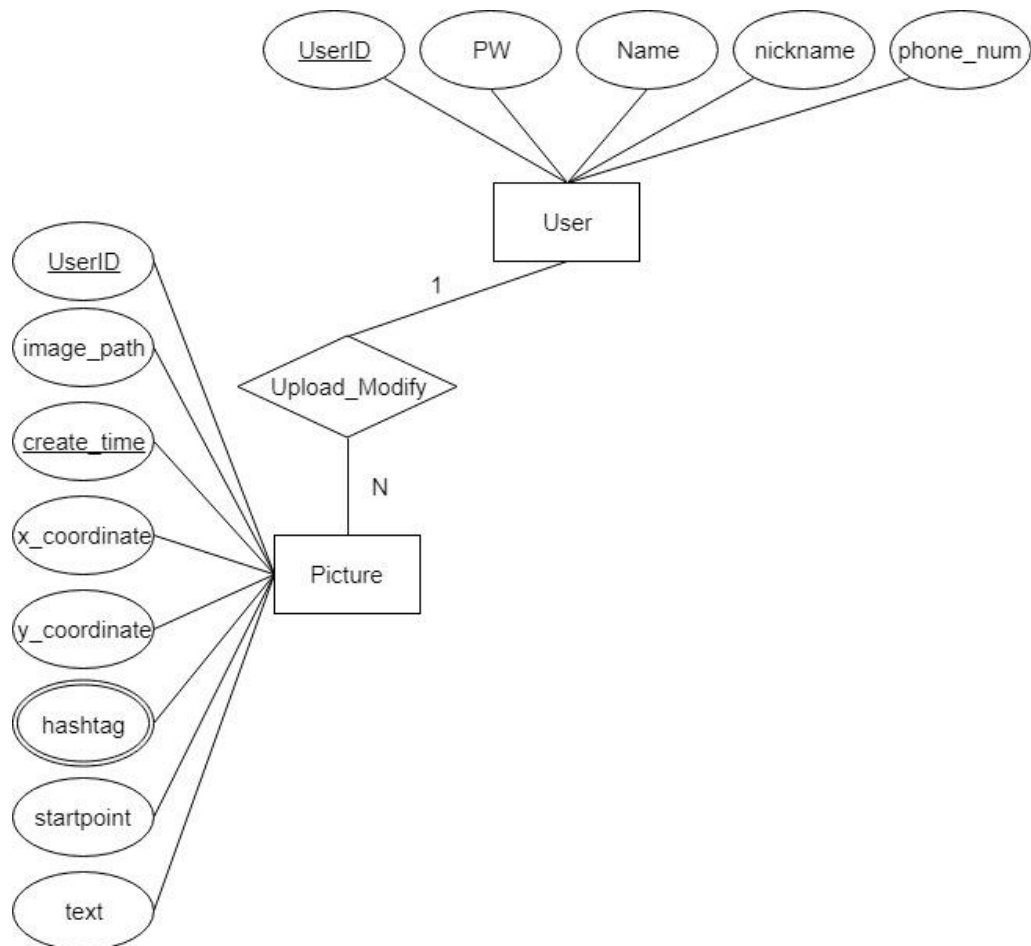


Diagram 27 Modify_Upload ER Diagram

2) Record

Record Relationship 은 사용자가 기록하는 경로 데이터 간의 관계를 나타낸다. 각각의 경로 기록은 한 명의 사용자에게 포함되며, 한 명의 사용자는 여러 개의 경로 기록을 가질 수 있다.

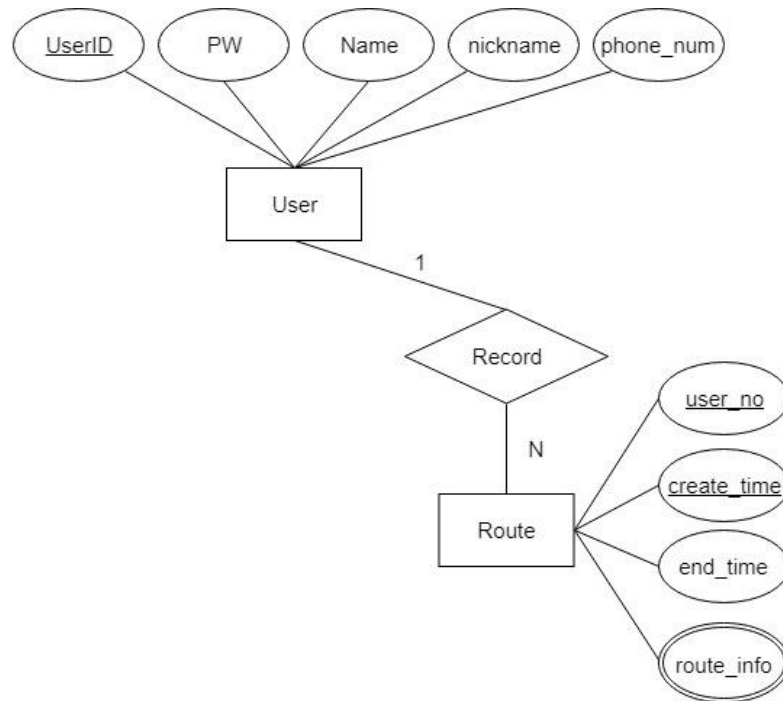


Diagram 28 Record ER Diagram

10.3 Relational Schema

A. User

<u>UserID</u>	PW	Name	nickname	phone_num
---------------	----	------	----------	-----------

Primary Key(PK): UserID

FUNCT DEP(FD): UserID → {UserID, PW, Name, nickname, phone_num}

Description:

사용자에 관한 테이블이다. UserID 를 PK 로 가지며, 해당 항목은 UNIQUE 속성을 가진다. UserID, PW, Name, nickname, phone_num 에 NULL 을 허용하지 않는다.

B. Picture

<u>UserID</u>	image_path	<u>create_time</u>	x_coordinate
---------------	------------	--------------------	--------------

y_coordinate	hashtag	starpoint	Text
--------------	---------	-----------	------

Primary Key(PK): UserID, create_time

Foreign Key(FK): UserID → User(UserID)

FUNCT DEP(FD): {UserID, create_time} → {UserID, image_path, create_time, x_coordinate, y_coordinate, hashtag, starpoint, text}

Description:

사용자가 업로드한 사진과 위치 정보에 대한 테이블이다. UserID, create_time, x_coordinate, y_coordinate 의 값에 NULL 을 허용하지 않으며, User 테이블의 UserID 를 FK 로 사용하며, UserID 와 create_time 를 PK 로 가진다.

C. Route

<u>UserID</u>	<u>create_time</u>	end_time	route_info
---------------	--------------------	----------	------------

Primary Key(PK): UserID, create_time

Foreign Key(FK): UserID → User(UserID)

FUNCT DEP(FD): {UserID, create_time} → {UserID, create_time, end_time, route_info}

Description:

사용자가 기록한 경로 데이터에 대한 테이블이다. UserID, create_time, end_time, route_info 의 값에 NULL 을 허용하지 않으며, User 테이블의 UserID 를 FK 로 가지며, UserID 와 create_time 를 PK 로 가진다.

10.4 SQL DDL

A. User

```
CREATE TABLE User (
    user_id VARCHAR(20) NOT NULL UNIQUE,
    PW VARCHAR(20) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    nickname VARCHAR(20) NOT NULL,
    phone_num VARCHAR(20) NOT NULL,
    PRIMARY KEY(user_id)
)
```

B. Picture

```
CREATE TABLE Picture (  
    user_id VARCHAR(20) NOT NULL,  
    image_path VARCHAR(255),  
    create_time DATETIME NOT NULL,  
    x_coordinate FLOAT NOT NULL,  
    y_coordinate FLOAT NOT NULL,  
    hashtag JSON,  
    starpoint INT,  
    text LONGTEXT,  
    PRIMARY KEY(user_id, create_time),  
    FOREIGN KEY(user_id) REFERENCES User(user_id) ON DELETE CASCADE  
ON UPDATE CASCADE  
)
```

C. Route

```
CREATE TABLE Route (  
    user_id VARCHAR(20) NOT NULL,  
    create_time DATETIME NOT NULL,  
    end_time DATETIME NOT NULL,  
    route_info JSON NOT NULL,  
    PRIMARY KEY(user_id, create_time),  
    FOREIGN KEY(user_id) REFERENCES User(user_id) ON DELETE CASCADE  
ON UPDATE CASCADE  
)
```

11 Testing Plan

11.1 Objective

시스템이 의도한 대로 실행되는지를 확인하고, 시스템 내부의 결함을 찾기 위한 과정으로, 시스템 설계 단계에 미리 계획하여 시행한다. 본 항목에서는 test policy 에서 testing 의 범위와 접근 방식에 대해 명확히 서술하고, testing 하는 리소스와 항목, testing 기능에 대해 구체적인 입출력 데이터를 바탕으로 test case 를 명시한다.

11.2 Testing Policy

A. Developing Testing

본 testing 은 시스템이 아직 개발 단계에 있는 동안 시행된다. 개발 단계가 나아갈 때마다, 만들어진 모든 구성 요소를 테스트하는 것으로, 구성 요소가 제대로 작동하는지를 확인하고자 한다. 만약 구성 요소가 제대로 작동하지 않는다면, 다음 단계로 넘어가지 않는다.

B. Release Testing

본 testing 은 해당 시스템을 시장에 release 하기 이전에 최종적으로 개발자가 확인하는 과정으로, requirement specification 의 요구사항이 시스템에 제대로 반영되었는지를 확인한다.

C. User Testing

본 testing 은 사용자가 직접 제품 및 서비스를 사용하면서 진행하는 과정이다.

11.3 Test Case

A. Login Test

- 1) User: 앱을 실행하면 뜨는 로그인 창에서 ID와 비밀번호를 입력하고 로그인을 시도한다.
- 2) 시스템 동작: 입력 받은 ID와 PW를 User DB의 정보와 비교한다.
 - 2-1) (로그인 성공): 사용자만의 지도 뷰가 보인다.
 - 2-2) (로그인 실패): 시스템 알림으로 '로그인 실패'를 출력하고 다시 로그인 화면을 보여준다.

B. Sign Up Test

- 1) User: 회원가입에 필요한 정보를 폼에 맞추어 작성하여 회원가입 버튼을 눌러 회원가입을 시도한다.
- 2) 시스템 동작: 입력 받은 정보가 빠진 내용은 없는지 확인하고 ID 가 중복되지 않는지 User DB 와 확인하여, 빠진 내용이 없고 중복되지 않는 ID 라면 회원가입에 성공하여 다시 로그인 화면을 띄워준다.
 - 2-1) (ID 중복): '이미 사용되고 있는 ID 입니다.'라는 시스템 알림을 띄운 후, ID 를 새로 입력 받는다.
 - 2-2) (빠진 정보): 빠진 내용이 있다는 시스템 알림을 띄우고, 빠진 내용을 채우기 전까지 회원가입 다음 단계로 넘어갈 수 없도록 한다.

C. Record Test

- 1) User: 로그인을 한 상태에서 지도 화면에 존재하는 Record 버튼을 눌러서 경로 기록을 시작하고 Record 버튼을 다시 눌러서 종료한다.
- 2) 시스템 동작: 경로 기록 시작 시간, 종료 시간, 기록 시간동안 GPS 를 통한 사용자의 현재 위치를 지속적으로 추출하여 사진 DB 에 추가한다.
 - 2-1) (추가 성공): 지도화면 위에 경로를 생성하고 지도화면으로 돌아가 생성된 경로를 보여준다.
 - 2-2) (추가 실패): "추가 실패"를 출력하고 GPS 를 켜야 한다는 알림을 보여주며 지도화면으로 이동한다.

D. Timestamp Test

- 1) User: 로그인을 한 상태에서 지도 화면에 존재하는 Timestamp 버튼을 누른다.
- 2) 시스템 동작: GPS 를 통해 사용자의 현재 위치, 그리고 현재 시각을 추출해 사진 DB 에 추가한다.
 - 2-1) (추가 성공): 지도화면 위에 핀을 생성하고 지도화면으로 돌아가 생성된 핀을 보여준다.
 - 2-2) (추가 실패): "추가 실패" 를 출력하고 GPS 를 켜야 한다는 알림을 보여주며 지도화면으로 이동한다.

E. Hashtag Test

- 1) User: 추천된 해시태그들 중 자신이 원하는 태그를 선택한다.
- 2) 시스템 동작: 사용자가 선택한 내용이 맞는지 사용자가 선택한 해시태그들을 다시 보여주는 시스템 알림을 띄우고, '확인'을 눌러야 다음 단계로 넘어가도록 한다.

F. Image Save, Modify Test

- 1) User: 사용자가 입력, 수정하고자 하는 사진, 해시태그, 텍스트 정보 등에 대해 입력, 수정한다.
- 2) 시스템 동작: 입력, 수정이 된 결과 값을 보여주며 입력, 수정된 결과를 사진 DB에 저장한다.

12 Development Environment

12.1 Objective

본 항목에서는 개발자의 환경에 대해 설명하며, 사용한 프로그래밍 언어와 IDE 등에 대해 기술한다.

12.2 Programming Language & IDE

A. Programming Language

1) PHP



Figure 14 PHP Logo

PHP 는 프로그래밍 언어의 일종이다. 원래는 동적 웹 페이지를 만들기 위해 설계되었으며 이를 구현하기 위해 PHP 로 작성된 코드를 HTML 소스 문서 안에 넣으면 PHP 처리 기능이 있는 웹 서버에서 해당 코드를 인식하여 작성자가 원하는 웹 페이지를 생성한다.

PHP 는 Android 와 Database 를 연동할 때에 필요하다. Android 는 단독으로 Database 에 접근할 수 없기 때문에 PHP 가 Android 와 Database 를 웹서버를 통해서 xml 파일을 주고받는 방식으로 중계함으로써 접근을 가능하게 한다.

2) JAVA



Figure 15 JAVA Logo

JAVA 는 객체 지향 프로그래밍 언어이다. 객체 지향 언어의 특성상 모듈화가 용이하고 재사용이 용이하다.

Android Studio 에서 안드로이드 어플리케이션을 제작하기위해 사용된다. Android 어플리케이션 제작을 위한 유일한 언어는 아니지만 상당수의 오픈소스가 JAVA 로 되어있어 가장 사용하기에 용이하다.

3) MySQL



Figure 16 MySQL Logo

MySQL 은 관계형 데이터베이스(RDB)를 관리하는 시스템으로, 오픈 소스라는 장점을 가지고 있어 데이터베이스를 구축할 때 널리 쓰이고 있다. 다중 스레드, 다중 사용자 형식의 구조를 가지고 있으며, 오라클이 관리 및 지원을 담당하고 있다.

MySQL 은 본 시스템에서 사용되는 서버와 연결되는 데이터베이스를 구축하고 관리하는 데에 사용된다. MySQL 을 통해 생성한 DB 에 본

시스템에서 사용하고 저장해야 하는 데이터들을 사용자 별로 저장하여, 사용자가 저장한 데이터가 보일 수 있도록 편리함을 추구한다.

B. IDE

1) Android Studio



Figure 17 Android Studio Logo

Android Studio is an Android integrated development tool from Google, based on IntelliJ IDEA. Similar to Eclipse ADT, Android Studio provides integrated Android development tools for development and debugging.

12.3 Version Management Tool

A. GitHub



Figure 18 GitHub Logo

Github is a hosting platform for open source and proprietary software projects, because only git is supported as the only repository format for hosting, hence the name Github.

12.4 네이버 지도 API



Figure 19 Naver Map API

네이버 지도 API 는 네이버 클라우드 플랫폼에서 제공하는 지도 API 로, 실시간 업데이트와 사용자의 지도 수정 요청을 실시간으로 반영하며 다양한 위치기반 API 서비스를 제공한다. 제공 서비스 중 Mobile Dynamic Map 을 사용할 예정이며, 이는 Android SDK 로 제공된다.

네이버 지도 API 는 본 시스템에서 지도를 표시하고 지도, 핀, 경로 등 관련 기능을 제공하는데 사용된다.

12.5 Google Cloud Vision API

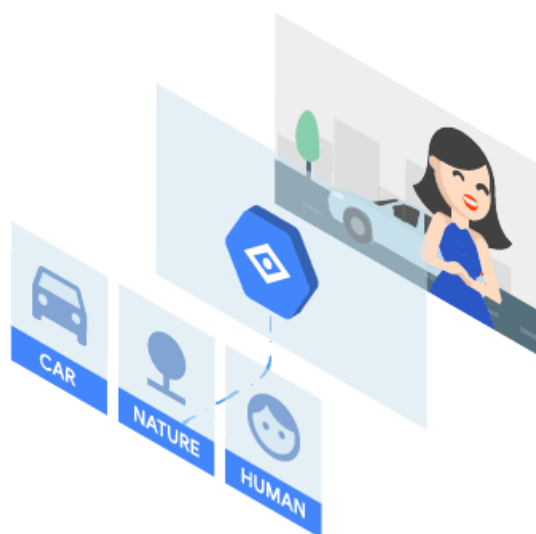


Figure 20 Google Cloud Vision API

Cloud Vision 은 API 를 통해 강력한 머신러닝 모델을 개발자가 쉽게 사용할 수 있도록 제공한다. 해당 API 를 통해 이미지의 내용을 파악할 수 있는데, 수천 가지 카테고리로 빠르게 이미지를 분류하기도 하고, 이미지 안의 개별 객체와 얼굴을 감지하며, 이미지에 인쇄된 단어를 판독한다.

해당 API 는 본 시스템 내에서 이미지 분석 시스템을 통해 해시태그를 추천해 줄 때에, 사용자가 업로드하는 이미지를 분석하여 카테고리를 감지하는 '라벨 감지'에 사용될 예정이다.

13 Development Plan

13.1 Objective

본 항목에서는 개발 계획에 대해 서술한다. 개발 계획은 gantt chart 를 통해 구체화하여 표현한다.

13.2 Schedule (Gantt Chart)

	1주	2주	3주	4주	5주	6주	7주	8주	9주	10주	11주	12주	13주	14주	15주
Idea															
Feasibility Study															
Requirement Analysis															
Design Study															
Design															
Develop															
Integration Testing															
Documentation															

14 Index

14.1 Objective

본 항목에서는 도표, 다이어그램 및 삽입 이미지들의 인덱스를 표기한다. 이를 통해 원하는 정보의 위치를 빠르게 파악할 수 있다.

14.2 Table Index

Table 1 Version History Table.....	8
Table 2 Login Protocol Request.....	37
Table 3 Login Protocol Response.....	37
Table 4 Sign Up Protocol Request.....	37
Table 5 Sign Up Protocol Response.....	37
Table 6 Map View Protocol Request.....	37
Table 7 Map View Protocol Response.....	38
Table 8 Route Protocol Request.....	38
Table 9 Route Protocol Response.....	38
Table 10 Timestamp Protocol Request.....	38
Table 11 Timestamp Protocol Response.....	38
Table 12 Save, Modify Picture Protocol Request.....	39
Table 13 Save, Modify Picture Protocol Response.....	39
Table 14 Image Analysis Protocol Request.....	39
Table 15 Image Analysis Protocol Response.....	39
Table 16 Hashtag Protocol Request.....	39
Table 17 Hashtag Protocol Response.....	39

14.3 Figure Index

Figure 1 UML Logo.....	9
Figure 2 Example of Class Diagram.....	10
Figure 3 Example of Sequence Diagram.....	10
Figure 4 Example of State Diagram.....	11
Figure 5 Example of ER Diagram.....	12
Figure 6 draw.io Logo.....	12
Figure 7 로그인 및 회원가입 시스템.....	13

Figure 8 지도 시스템	13
Figure 9 타임스탬프 기록 시스템	13
Figure 10 사진 등록 시스템	14
Figure 11 해시태그 시스템	14
Figure 12 JSON Logo	36
Figure 13 ER Diagram of LTE System	40
Figure 14 PHP Logo	50
Figure 15 JAVA Logo	51
Figure 16 MySQL Logo	51
Figure 17 Android Studio Logo	52
Figure 18 GitHub Logo	52
Figure 19 Naver Map API	53
Figure 20 Google Cloud Vision API	53

14.4 Diagram Index

Diagram 1 Block Diagram of LTE System	15
Diagram 2 Block Diagram of Login & Sign Up System	16
Diagram 3 Block Diagram of Map System	16
Diagram 4 Block Diagram of Timestamp System	17
Diagram 5 Block Diagram of Uploading Picture System	17
Diagram 6 Block Diagram of Image Analysis System	18
Diagram 7 Class Diagram of Login & Sign Up System	19
Diagram 8 Sequence Diagram of Login & Sign Up System	21
Diagram 9 State Diagram of Login & Sign Up System	22
Diagram 10 Class Diagram of Map System	23
Diagram 11 Sequence Diagram of Loading Image System	25
Diagram 12 Sequence Diagram of Recording Route System	25
Diagram 13 State Diagram of Map System	26
Diagram 14 Class Diagram of Timestamp System	27
Diagram 15 Sequence Diagram of Registering Timestamp	28
Diagram 16 State Diagram of Registering Timestamp	28
Diagram 17 Class Diagram of Uploading Picture System	29
Diagram 18 Sequence Diagram of Saving Picture	31
Diagram 19 Sequence Diagram of Modifying Picture	31
Diagram 20 State Diagram of Uploading Picture System	32
Diagram 21 Class Diagram of Image Analysis System	33

Diagram 22 Sequence Diagram of Analyzing Picture	34
Diagram 23 State Diagram of Image Analysis System.....	35
Diagram 24 User ER Diagram	41
Diagram 25 Picture ER Diagram	41
Diagram 26 Route ER Diagram	42
Diagram 27 Modify_Upload ER Diagram	43
Diagram 28 Record ER Diagram	44

<작성 담당 부분>

이름	작성 부분
김준형	* 타임스탬프 기록 시스템과 관련된 부분 전체 작성 - 3.2.C. 타임스탬프 기록 시스템 - 6. 타임스탬프 기록 시스템 - 9. Protocol Design 부분 중 해당 시스템 관련 부분 작성 - 11. Testing Plan 부분 중 해당 시스템 관련 부분 작성 - 12. Development Environment 부분 중 PHP 등 통신 관련 부분 작성
배경률	* 사진 등록 시스템과 관련된 부분 전체 작성 - 3.2.D. 사진 등록 시스템 - 7. 사진 등록 시스템 - 9. Protocol Design 부분 중 해당 시스템 관련 부분 작성 - 11. Testing Plan 부분 중 해당 시스템 관련 부분 작성 - 12. Development Environment 부분 해당 시스템에서 이용하는 부분 작성
신아름	* 전체 항목 목적 작성 및 문서 서식 정리 등 문서 전반 관리 및 정리 - 1. Preface/2. Introduction 부분 정리 - 목차 작성 - 13. Development Plan 작성 - 14. Index 작성 * 2.4 Project Scope 부분 작성 * 로그인 및 회원가입 시스템/이미지 분석 시스템과 관련된 부분 전체 작성 - 3.2.A. 로그인 및 회원가입 시스템 - 3.2.E. 이미지 분석 시스템 (해시태그 시스템) - 4. 로그인 및 회원가입 시스템 - 8. 이미지 분석 시스템 (해시태그 시스템) - 11. Testing Plan 부분 중 해당 시스템 관련 부분 작성 - 12. Development Environment 부분 해당 시스템에서 이용하는 부분 작성 * DB 설계 - 10 Database Design 부분 작성
이금택	1. Preface 작성 2. Introduction 작성
주유흠	1. Preface 작성 2. Introduction 작성
천수현	* 지도 시스템과 관련된 부분 전체 작성 - 3.2.B. 지도 시스템 - 5. 지도 시스템

	<ul style="list-style-type: none">- 9. Protocol Design 부분 중 해당 시스템 관련 부분 작성- 11. Testing Plan 부분 중 해당 시스템 관련 부분 작성- 12. Development Environment 부분 해당 시스템에서 이용하는 부분 작성
--	--