

---

# Software Design

# Specification

For

# Coloride

문준현 이용우 전덕우 왕보준

소프트웨어공학개론 8 조

2018-05-20

# 목차

---

1. Preface.....	8
1.1 Objective.....	8
1.2 Readership.....	8
1.3 Document structure .....	8
A. Preface.....	8
B. Introduction.....	8
C. System Architecture .....	8
D. User Management System .....	8
E. Public Project System.....	9
F. Private Project System.....	9
G. Custom Request Project System.....	9
H. Protocol Design .....	9
I. Database Design .....	9
J. Testing Plan.....	9
K. Development Environment .....	10
L. Develop Plan .....	10
2. Introduction .....	11
2.1 Objective.....	11
2.2 Applied Diagram.....	11
A. UML .....	11
B. Package Diagram.....	11
C. Deployment Diagram .....	12
D. Class Diagram.....	13
E. State Diagram.....	14

F. Sequence Diagram .....	15
G. ER Diagram .....	15
2.3 Applied Tool .....	16
A. Flow Chart Maker & Online Diagram Software .....	16
2.4 Project Scope .....	17
A. User Management System.....	17
B. Public Project System .....	17
C. Private Project System .....	17
D. Custom Project System .....	17
3. System Architecture .....	18
3.1 Objectives.....	18
3.2 System Organization.....	18
A. User management system.....	19
B. Public project system .....	21
C. Private project system.....	22
D. Custom request system.....	23
3.3 Deployment Diagram .....	24
4. User Management System.....	25
4.1 Objectives.....	25
4.2 Class Diagram .....	25
A. DB Handler.....	25
B. Account.....	25
C. Color history .....	26
4.3 Sequence Diagram.....	27
A. Log in with using 'google' .....	27

B. My page.....	28
4.4 State Diagram .....	29
A. Log in with using 'google' .....	29
5. Public Project System.....	30
5.1. Objectives.....	30
5.2. Class Diagram .....	30
A. DB Handler.....	31
B. Picture.....	31
C. Painting.....	31
D. Account.....	31
5.3. Sequence Diagram.....	32
A. Painting System .....	32
5.4 State Diagram .....	33
A. Painting System .....	33
6. Private Project System .....	33
6.1. Objectives.....	33
6.2. Class Diagram .....	34
A. DB Handler.....	34
B. Picture.....	35
C. Painting.....	35
D. Account.....	35
6.3. Sequence Diagram.....	36
A. Painting System .....	36
6.4 State Diagram .....	37
A. Painting System .....	37

7. Custom Request System.....	38
7.1. Objectives.....	38
7.2. Class Diagram .....	38
A. DB Handler.....	38
B. Custom Request.....	38
C. Account.....	39
7.3. Sequence Diagram.....	40
7.4 State Diagram .....	41
8. Protocol Design .....	42
8.1 Objectives.....	42
8.2 JSON.....	42
8.3 Protocol Description .....	42
A. Overview.....	42
B. Login Protocol.....	42
C. Registration Protocol.....	43
D. Using Project Select/Edit Protocol .....	43
E. Project View Protocol.....	43
F. Comment Post/Edit Protocol.....	44
G. Comment View Protocol.....	44
H. Customized Project Insert/Edit Protocol .....	45
I. Customized Project Delete Protocol .....	45
J. Addressing Customized Project Protocol.....	45
K. Customized Project View Protocol.....	46
L. Project Post/Edit Protocol.....	46
10. Database Design.....	47

10.1 Objectives .....	47
10.2 ER Diagram .....	47
A. Entity .....	49
B. Relation .....	52
10.3 Relational Schema .....	53
A. Member .....	53
B. Comments.....	54
C. Picture.....	54
D. Public Project .....	54
E. Private Project .....	55
F. Custom Request .....	55
G. Colors .....	55
10.4 Normalization .....	56
12. Testing Plan.....	57
12.1 Objectives [1].....	57
12.2 Testing Policy [1].....	57
1) Development Testing .....	57
<i>Release</i> Testing .....	57
<i>User Testing</i> .....	57
12.3 Test <i>case</i> .....	58
A. User Management System.....	58
B. Project Management System .....	59
C. Custom Project Request System .....	61
D. Comments Systems.....	62
12. Development Environment .....	64

12.1 Objectives.....	64
12.2 Bootstrap (front-end framework) .....	64
12.3 PHP+MYSQL(back-end) .....	64
13.3 GitHub (Version Management) .....	65
13.Develop Plan .....	67
13.1 Objectives.....	67
13.2 Schedule.....	67
14. Index.....	68
14.1 Figure Index.....	68
14.2 Diagram Index.....	68
15. References.....	70

# 1. Preface

---

## 1.1 Objective

Preface 에서는 본 문서의 독자를 정의하고, 본 문서를 이루는 세부 사항들과 각 사항들의 목적을 서술한다.

## 1.2 Readership

본 문서의 독자는 다음과 같다.

- 1) 시스템을 설계하는 시스템 엔지니어
- 2) 시스템을 개발하는 소프트웨어 엔지니어
- 3) 그 외 Coloride 시스템 개발에 관련된 엔지니어들

기본적으로 시스템 설계와 개발에 관련된 구성원들을 대상으로 한다.

## 1.3 Document structure

### A. Preface

Preface 에서는 본 문서의 독자를 정의하고, 본 문서는 이루는 세부 사항들과 각 사항들의 목적을 서술한다.

### B. Introduction

Introduction 에서는 본 시스템을 이루는 Architecture 과 각 Subsystem 들에 대한 소개와 이를 구성하는 도구 등을 서술한다.

### C. System Architecture

System Architecture 에서는 우리 팀이 개발하는 시스템에 대해 전반적으로 서술한다. 전체적인 시스템의 구조를 설명하고 시스템을 Block Diagram 으로 나타내서 시스템 간의 관계와 구조를 설명한다. 또한 Deployment Diagram 을 사용해서 시스템의 실제 어떻게 사용되는지를 서명한다.

### D. User Management System

User Management System 에서는 사용자가 시스템을 사용하면서 생기는 데이터와 사건들을 관리하는 시스템을 서술한다. 회원가입, 로그인,



회원정보 관리 등 회원에 대한 관리와, 비회원 사용자에게 대한 관리를 통틀어 서술한다. 이 시스템들의 기능과 설계를 설명하고, Class Diagram, Sequence Diagram, State Diagram 을 통해 구조를 표현한다.

#### E. Public Project System

Public Project System에서는 유저가 다른 유저와 공동으로 한 그림을 어느 파트를 맡아서 그리고 저장하는 시스템에 대해서 설명한다. 이 시스템은 Painting 이 이루어지는 Painting sub-system 을 가지고 있다. Public Project System 의 전체적인 구조를 Class Diagram 으로 설명하고, 핵심 기능이 이루어지는 Painting sub-system 에 대해서 Sequence Diagram 과 State Diagram 을 통해 설명한다.

#### F. Private Project System

Private Project System에서는 사용자가 각자 진행하는 개인 프로젝트에 대한 사항들을 서술한다.

#### G. Custom Request Project System

Custom Request Project System에서는 사용자가 이벤트, 프로모션 등을 위해 Coloride 관리자에게 Custom project 를 신청하는 시스템에 대해 서술한다.

#### H. Protocol Design

Protocol Design에서는 본 시스템에 존재하는 서브시스템들 사이에 일어나는 상호작용의 프로토콜에 대해 서술한다.

#### I. Database Design

Database Design에서는 시스템 개발에 필요한 Database 에 관련된 사항을 서술한다. 전체적인 구조를 ER Diagram 을 통해 표현하고, 이를 통해 상세한 Relation 과 Table 를 구상한다.

#### J. Testing Plan

Testing Plan에서는 프로그램이 의도한 대로 개발되었는지, 그리고 작동하는 과정에서 발생할 수 있는 문제들을 확인하기 위한 테스트에 대해 서술한다.

## K. Development Environment

Development Environment에서는 개발하는 환경에 대해 서술한다. Front-end와 Back-end로 나누어 서술한다.

## L. Develop Plan

Develop Plan에서는 개발 계획에 대해 서술한다.

## 2. Introduction

---

### 2.1 Objective

이 항목에서는 시스템 설계에 사용한 Diagram 과 Tool 에 대하여 설명한다.

### 2.2 Applied Diagram

#### A. UML

Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. Essentially, there are three perspectives that can be represented in the UML

1. A requirements perspective based on use-case diagrams.
2. A static perspective, based on object and composite structure diagrams.
3. A dynamic perspective, based on sequence and state diagrams.



Figure 1. Logo of UML

#### B. Package Diagram

UML 의 package diagram 은 모델을 구성하고 있는 package 들 사이의 의존성을 나타내는 그림이다. Package diagram 은 시스템의 기능을 설명하기 위해 use case 들을 가진 package 를 이용한다. 다층구조의 시스템을 package diagram 으로 표현할 경우

[illegible]

### C. Deployment Diagram

페이지 | 12

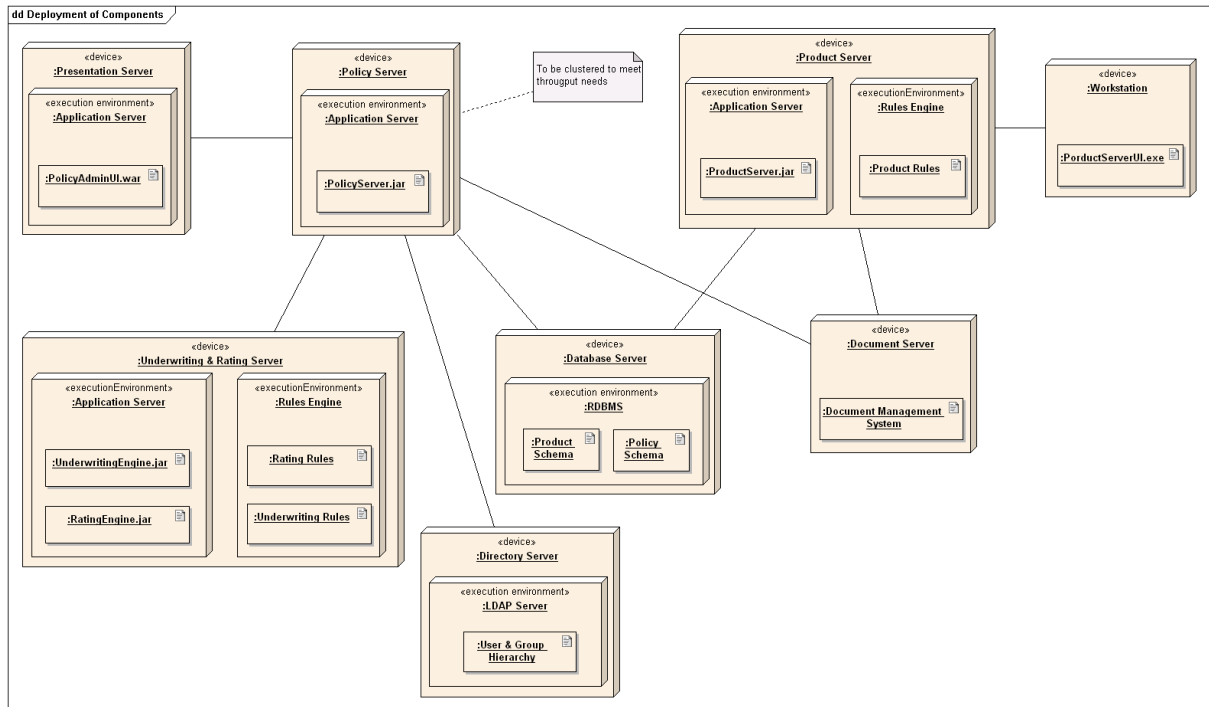


Figure 3. Example of Deployment Diagram

#### D. Class Diagram

Class diagram 은 시스템의 class, attribute, operation, object 사이의 relation 을 diagram 의 형식으로 보여줌으로써 시스템의 구조를 설명한다. Class diagram 은 소프트웨어의 시스템을 일반적으로 개념화할 때나 프로그래밍 코드로 변환하기 쉽도록 시스템의 구조를 자세히 모델링하는데 사용한다. Class diagram 에서 class 는 name, attribute, operation 의 세가지 구성 요소로 나타내어지고 class 를 사용함으로써 객체 지향적인 성격을 가지고 있다.

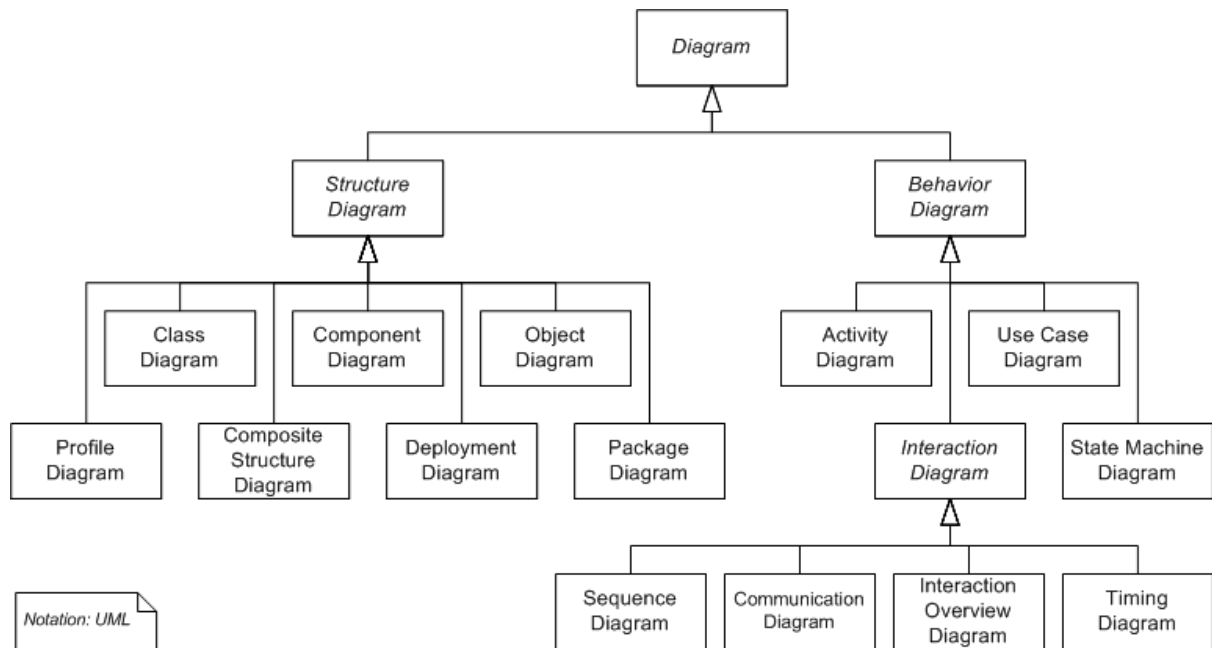


Figure 4. Example of Class Diagram

### E. State Diagram

State diagram 은 device, computer program, process work 들을 하나의 state 로 규정하고 event 의 발생에 의해 state 변화를 그림으로 표현한 것이다. 그래서 state diagram 은 directed graph 로 나타내어진다. State 는 각자의 state name 을 가지고 있다. Transition 은 화살표로 나타내고 발생한 event 에 따라 선택적으로 나타난다. 적은 데이터를 처리하는 real-time 시스템을 표현하기 적절하다.

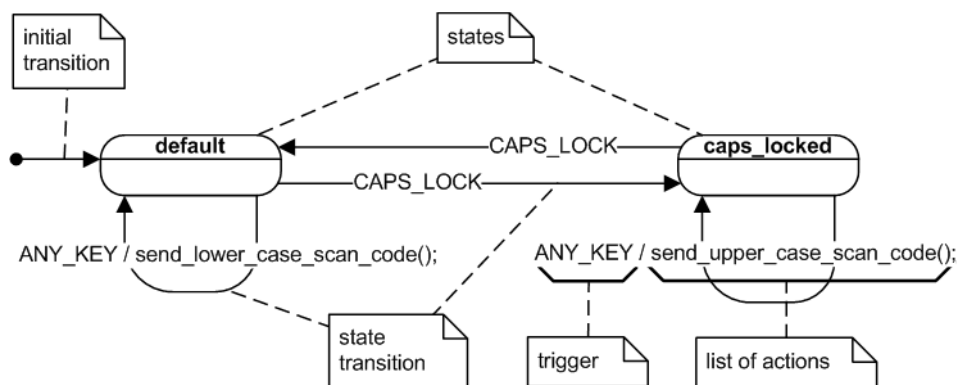


Figure 5. Example of State Diagram

## F. Sequence Diagram

Sequence diagram 은 time sequence 에 따른 object 의 상호작용을 표현해준다. 따라서 interaction modeling 의 대표적인 기법으로 쓰인다. Scenario 로부터 object, class, sequence 들을 추출하여 diagram 에 반영한다. Sequence diagram 은 개발 하에 시스템을 logical view 로 분석하여 use case 를 실체화 한다. Sequence diagram 의 구성요소는 lifeline, process, object, messages 들이 있다. 다음 구성 요소들을 통해 runtime scenario 를 시각적으로 표현할 수 있다.

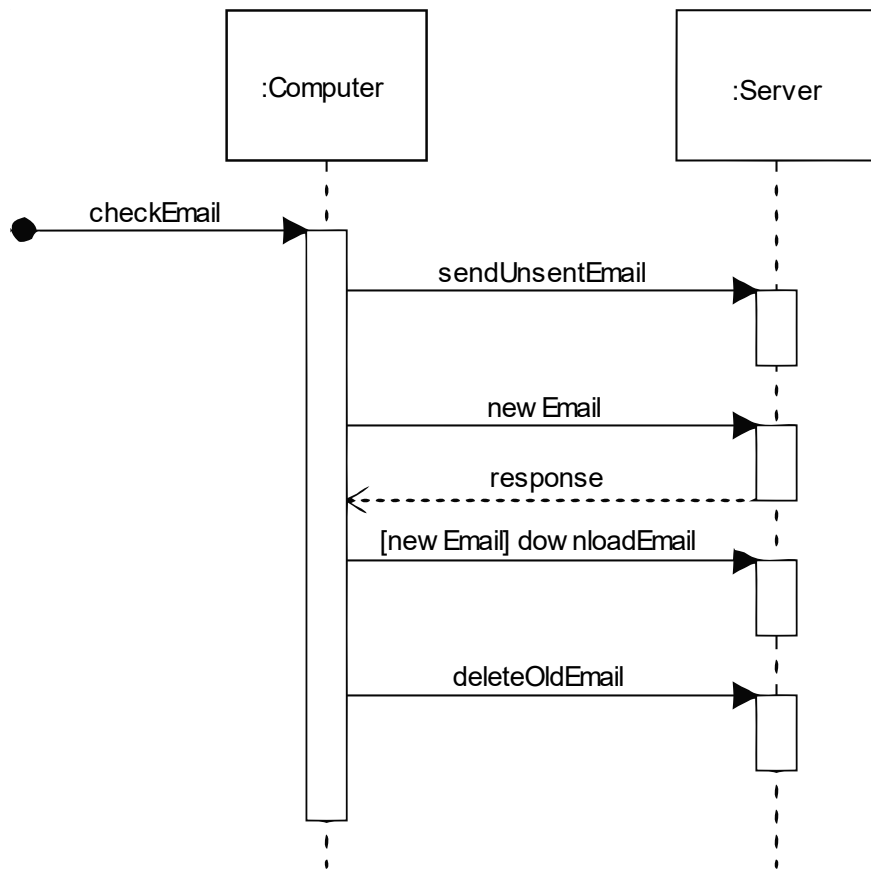


Figure 6. Example of Sequence Diagram

## G. ER Diagram

Entity-relationship model 은 특정 domain 의 지식 속에서 서로 관련이 있는 관심사들을 보여준다. ER model 은 entity type 들과 그들 사이의 relationship 을 나타낸다. ER model 은 사업의 분야에서 무엇을 처리하는 것이 중요한지 설명하고 정의하기 위한 시스템 분석의 결과물이다. 사업의 과정을 설명하지는 않지만 data schema 를 시각적인

형태로 표현한다. 전형적으로 database 를 나타낼 때, 특히 relational database 를 나타낼 때 ER model 을 사용하고 있다.

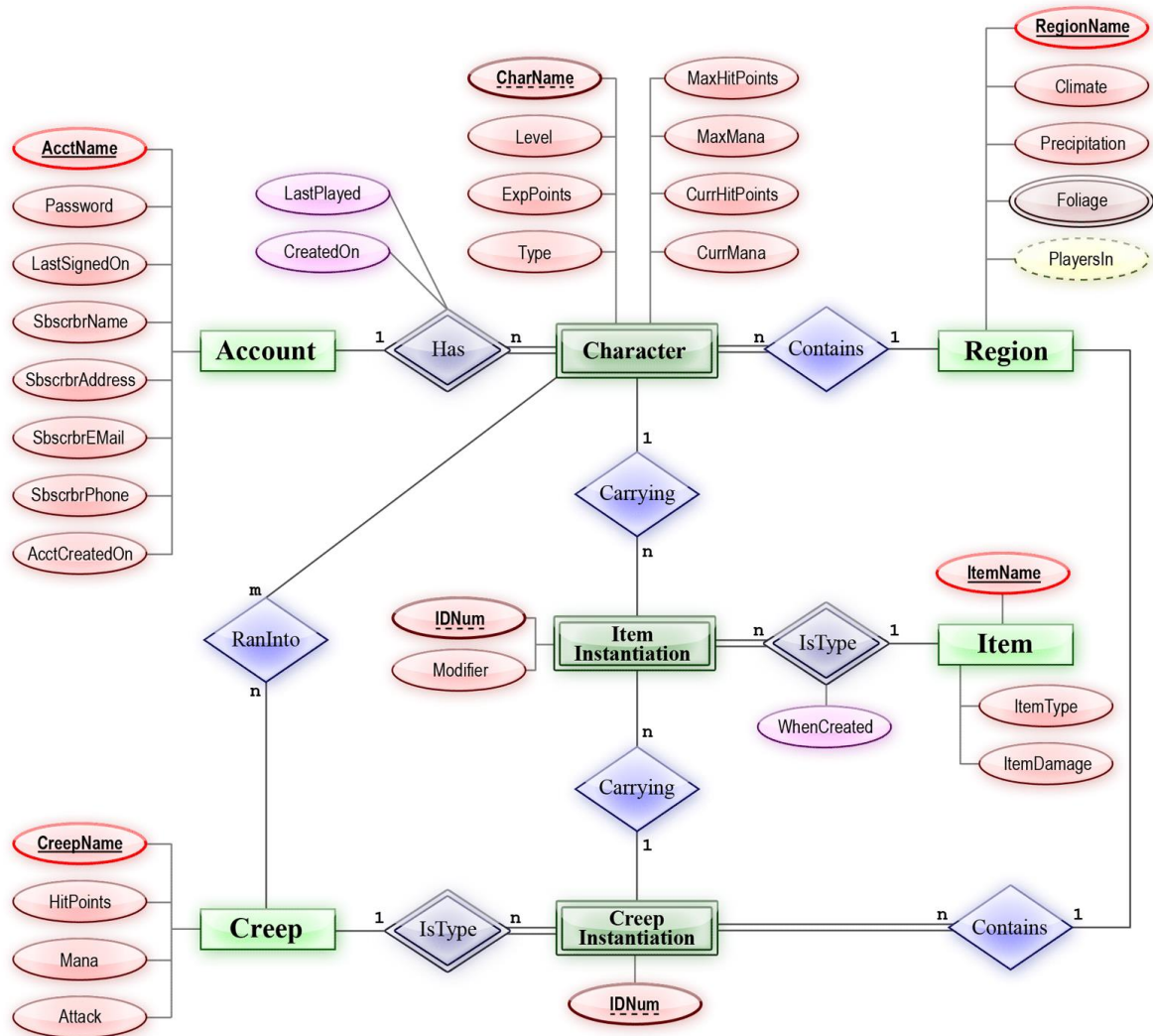


Figure 7. Example of ER Diagram

## 2.3 Applied Tool

### A. Flow Chart Maker & Online Diagram Software

Specification 에 필요한 시스템 설계도와 diagram 을 작성하기 위해 'Flow Chart Maker & Online Diagram Software' 서비스를 이용하였다. 가입 없이 온라인으로 작업할 수 있기 때문에 여러 diagram 과 flow chart 를 만드는데 장점을 갖는다. 그리고 작성한 작업물을 구글 드라이브를 통해 저장하고 공유할 수 있어 편리하다.



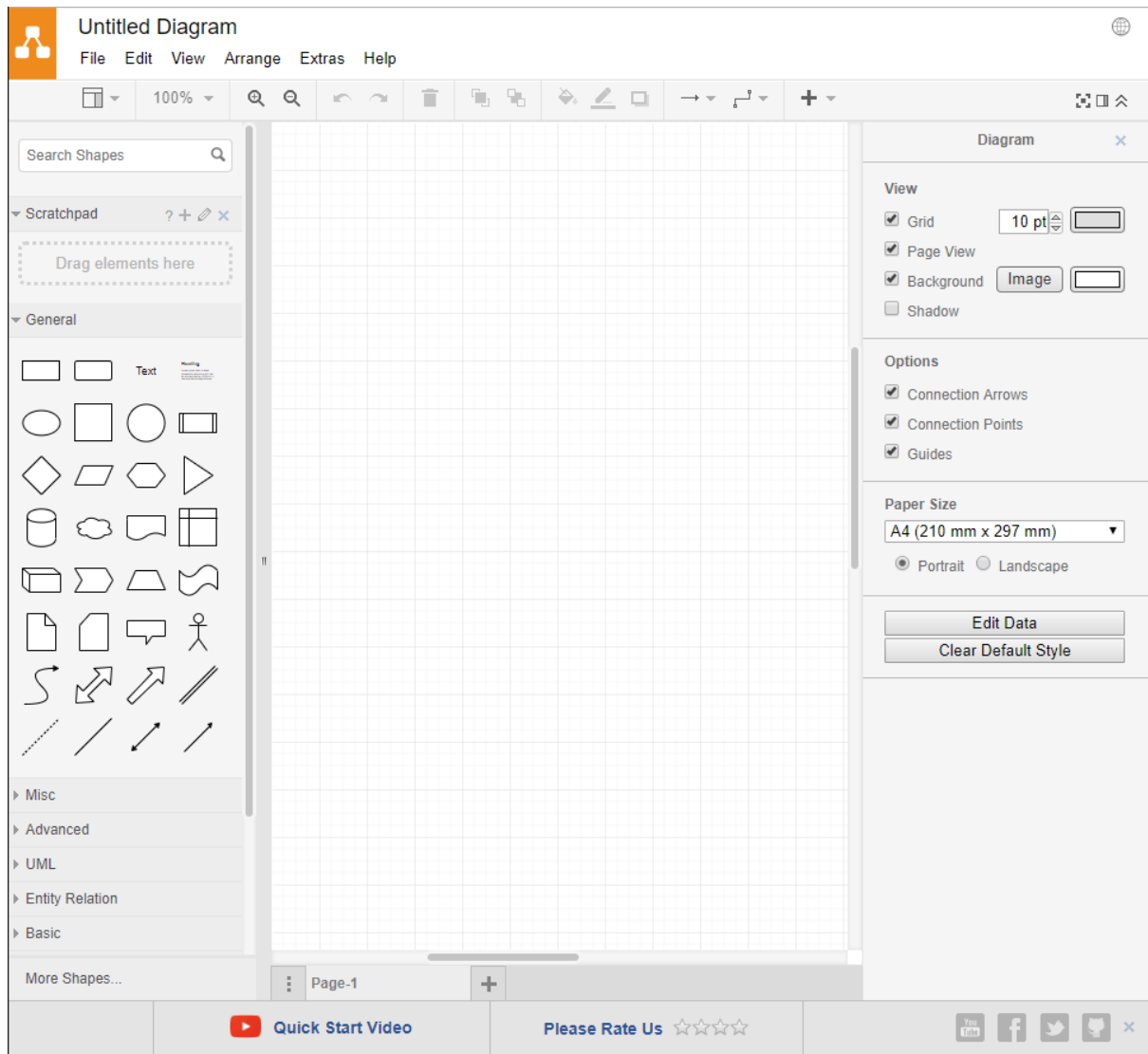


Figure 8. Captured Image of Flow Chart Maker & Online Diagram Software

## 2.4 Project Scope

- A. User Management System
- B. Public Project System
- C. Private Project System
- D. Custom Project System

## 3. System Architecture

### 3.1 Objectives

System Architecture 에서는 우리 팀이 개발하는 시스템에 대해 전반적으로 서술한다. 전체적인 시스템의 구조를 설명하고 시스템을 Block Diagram 으로 나타내서 시스템 간의 관계와 구조를 설명한다. 또한 Deployment Diagram 을 사용해서 시스템의 실제 어떻게 사용되는지를 서명한다.

### 3.2 System Organization

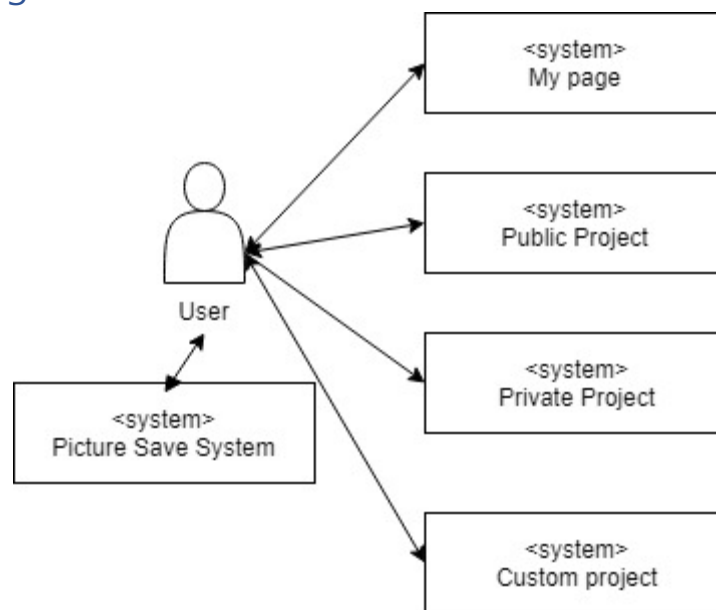
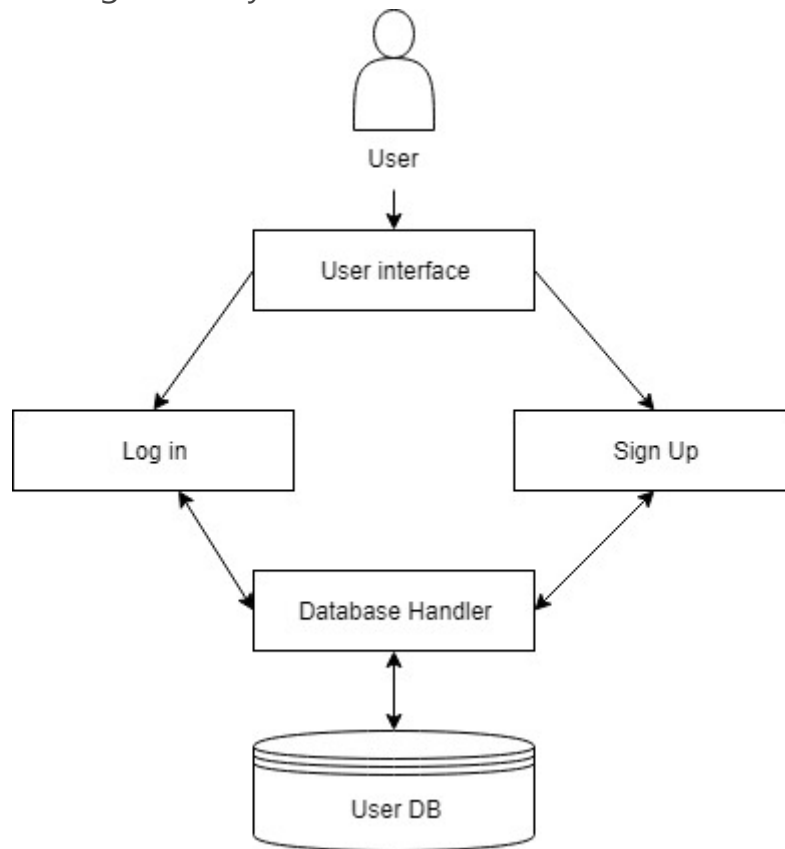


Diagram 1 System Organization Block Diagram

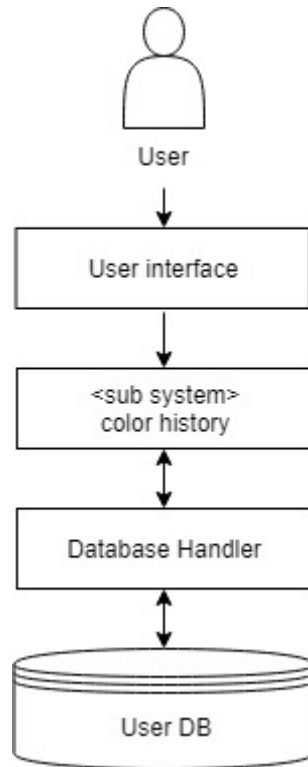
Coloride 시스템은 Client-server model 을 사용하여 구현된다. 유저는 웹의 유저 인터페이스에서 상호작용하고, 그에 따른 입력을 각 시스템들이 받아서 데이터베이스에 접근해서 데이터 수정, 전송이 이루어진다. 각 시스템들은 웹서버에 의해 사용자 클라이언트에서 오는 요청과 데이터를 받는다. 그 데이터 또한 서버를 통해 데이터베이스에 전송되고 적절한 처리가 이루어진 다음 재송신이 이루어진다. 시스템중 My page, Public Project, Private Project, Custom Project 는 유저와 상호작용하는 시스템이다.

## A. User management system



**Diagram 2 User Management System Architecture**

User Management System 은 사용자의 이용과 관련된 시스템이다. 사용자의 회원 가입을 관리하는 Sign Up sub-system 과 로그인 기능을 제공하는 Log in sub-system 으로 이루어진다. 여기서 Log in system 은 google authentication 을 제공할 예정이므로 구글의 시스템 구조를 따른다.



**Diagram 3 Color history sub-system**

또한 회원정보에서 관리할 수 있는 Color history system 이 sub-system 으로 존재한다. 이 시스템은 유저가 자신이 그린 그림의 현황을 체크할 수 있는 시스템이다. 로그인 시스템에 존재하게 되고, 로그인 후 My page 에서 확인할 수 있다.

## B. Public project system

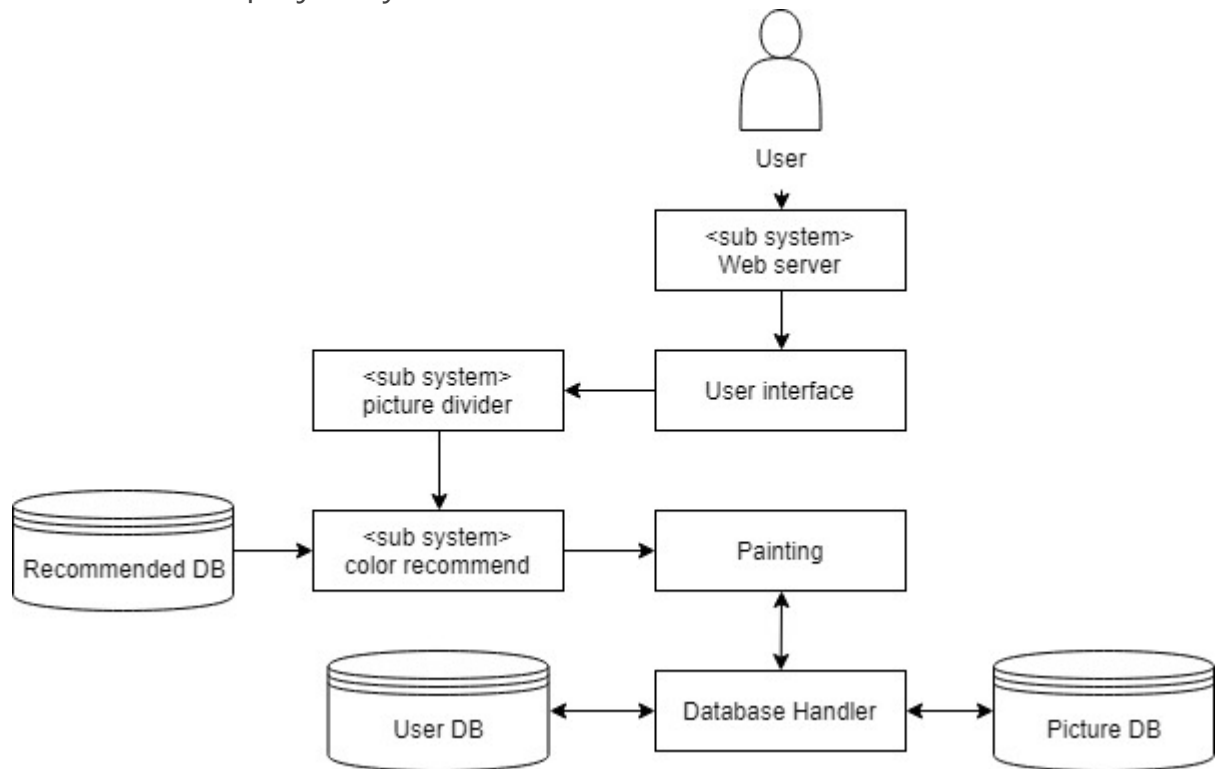


Diagram 4 Public Project System Architecture

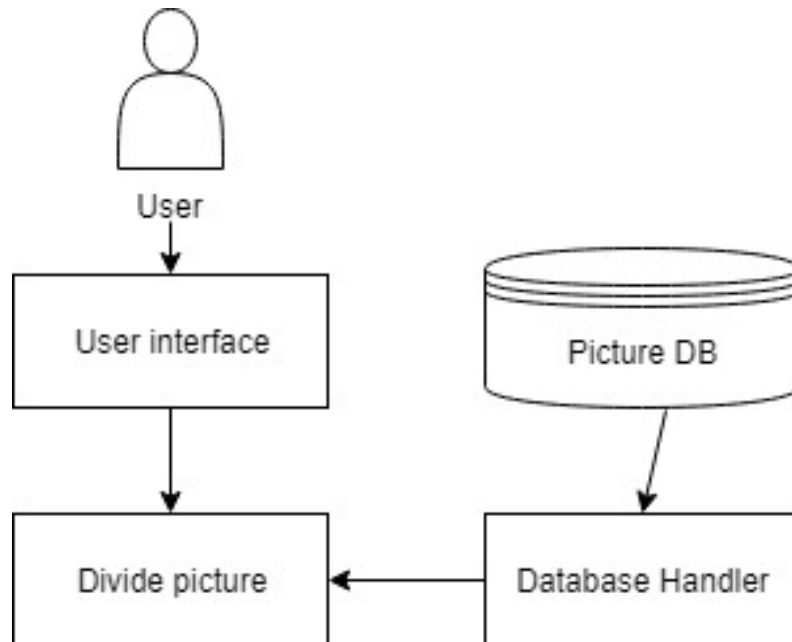
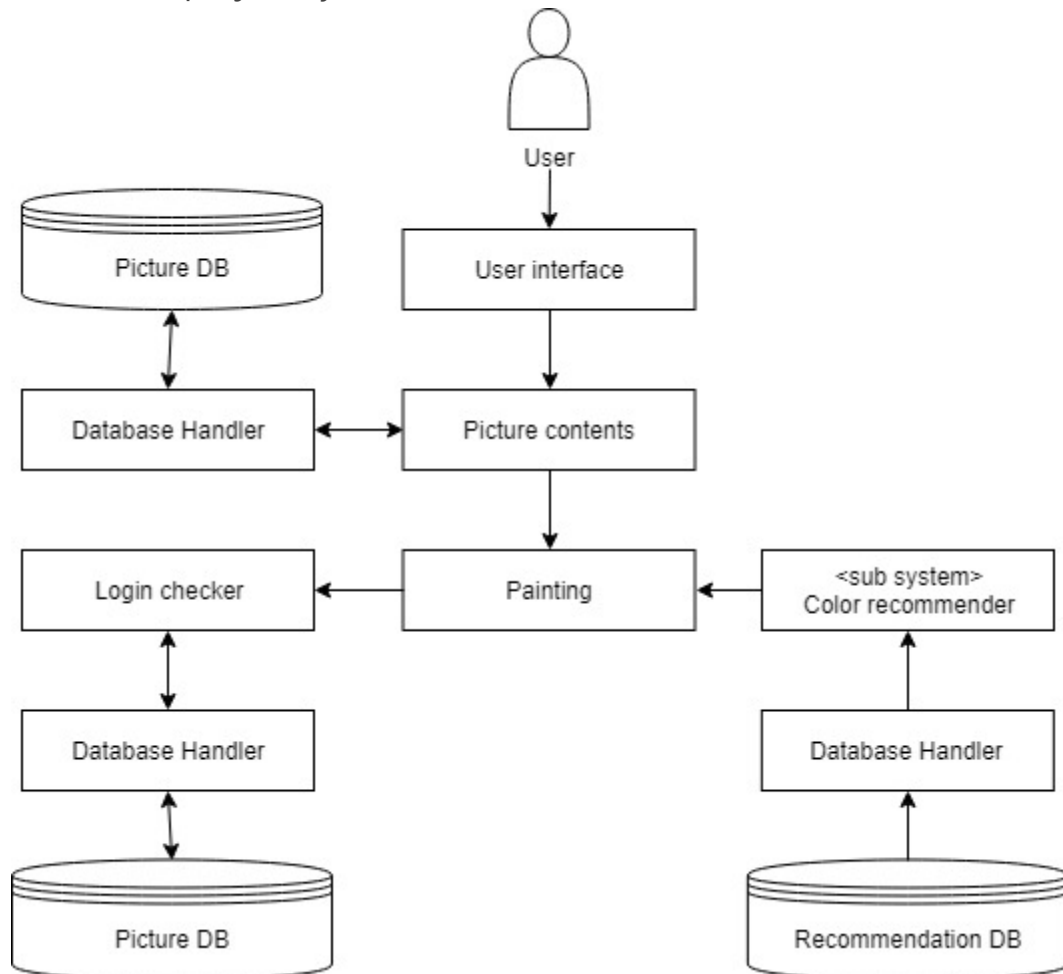


Diagram 5 Public Project Picture Divider Sub-System

Public Project System 은 이용자가 public project 에 등록된 그림을 보고, 선택한 후 그 그림의 어느 한 부분을 맡아서 그리고 그것이 데이터베이스에 반영되는 시스템이다.

Picture divider 와 color recommend sub-system 으로 구성되어 있다. Picture divider 은 그림을 각자 유저에게 분배하는 시스템이고 color recommend 는 유저의 기분을 입력 받아 적절한 색을 추천하는 시스템이다. Painting 에서 웹을 통해서 그림을 그리는 작용이 이루어진다.

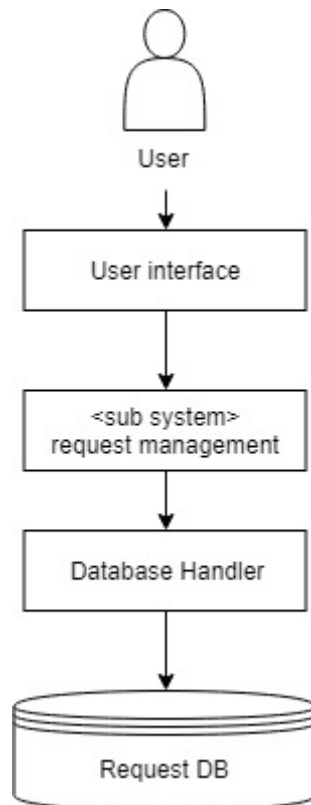
### C. Private project system



**Diagram 6 Private Project System**

Private Project System 은 유저가 혼자서 색칠할 그림을 선택하고 색칠이 완료된 후 저장을 하는 시스템이다. 유저가 그리게 되는 picture contents 는 웹상에서 제공되어지고 color recommender 이라는 sub-system 에 의해 기분에 따른 적절한 색을 제공받는다. Login check 를 한 후 로그인 이 되어 있다면 색칠한 내역을 저장해서 My page 에 color history 를 통해 볼 수 있게 한다.

#### D. Custom request system



**Diagram 7 Custom Request System Architecture**

Custom Request System 은 사용자가 Public Project System 에 등록할 만한 그림을 신청하는 시스템이다. Request sub-system 에 의해서 요청하는 상세 내역을 작성하게 되고 관리자에게 전송하게 된다. 전송한 Request 는 Request DB 에 저장해서 관리된다.

### 3.3 Deployment Diagram

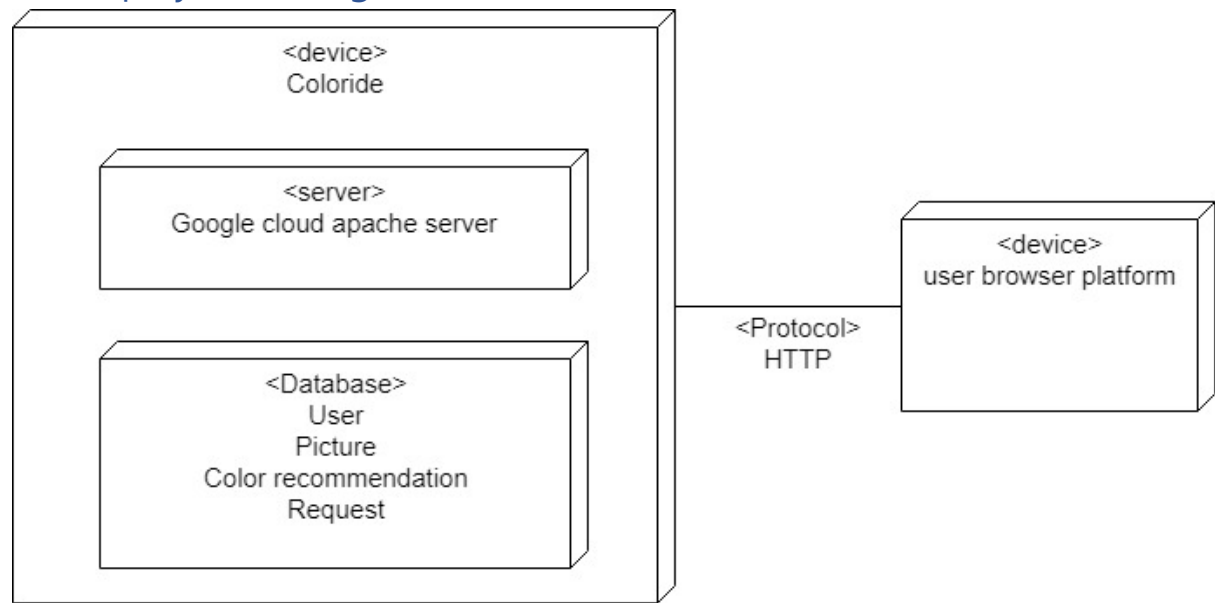


Diagram 8 Overall Deployment Diagram



## 4. User Management System

### 4.1 Objectives

웹 서비스 사용자가 사용하는 시스템이다. 이 시스템은 사용자의 로그인 기능과 시스템을 이용하면서 그린 그림들을 볼 수 있게 하는 시스템이다. 이 시스템들의 기능과 설계를 설명하고, Class Diagram, Sequence Diagram, State Diagram 을 통해 구조를 표현한다.

### 4.2 Class Diagram

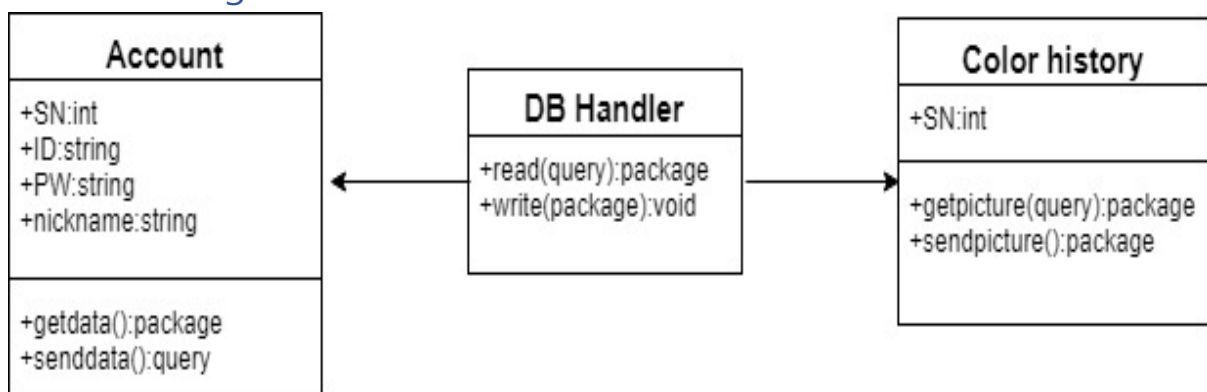


Diagram 9 User Management System Class Diagram

#### A. DB Handler

##### A.1 Attributes

없음

##### A.2 Methods

`+read(query):package`: 해당하는 DB 에서 원하는 data 를 가져옴

`+write(package):void`: 해당하는 DB 에 data 를 저장

#### B. Account

##### B.1 Attributes

`+SN:int`: 식별번호

`+ID:string`: 아이디 정보

+PW:string: 비밀번호 정보

+nickname:string: 닉네임 정보

## B.2 Methods

+getdata():package:DB 에서 data 를 가져온다

+senddata():query:DB 에 data 를 보낸다

## C. Color history

### C.1 Attributes

+SN:int: 그림 식별번호

### C.2 Methods

+getpicture(query):package: DB 에서 해당하는 그림을 가져온다

+sendpicture():package: DB 에 그림을 보낸다

### 4.3 Sequence Diagram

A. Log in with using 'google'

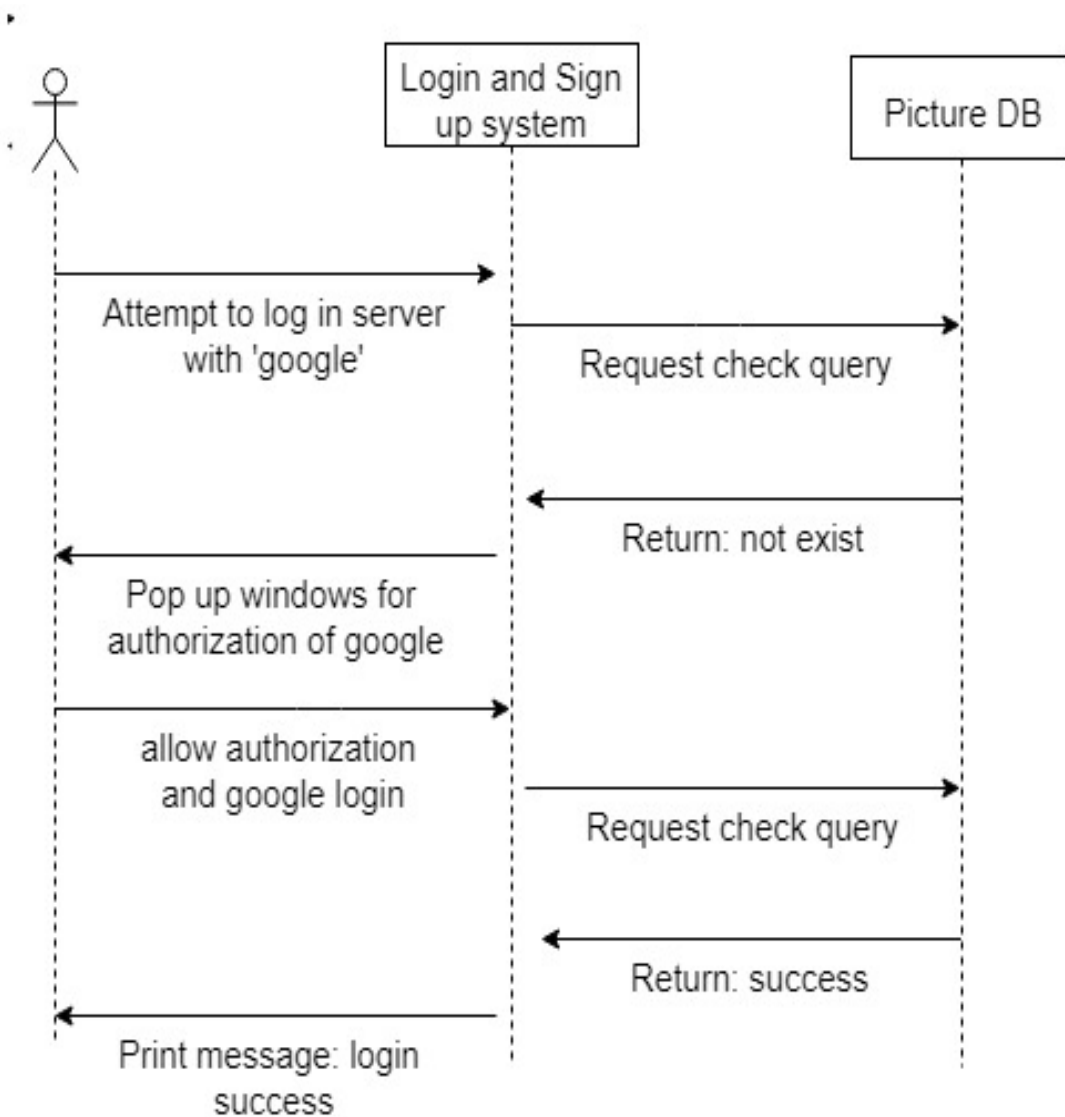


Diagram 10 Log in with Google Sequence Diagram

B. My page

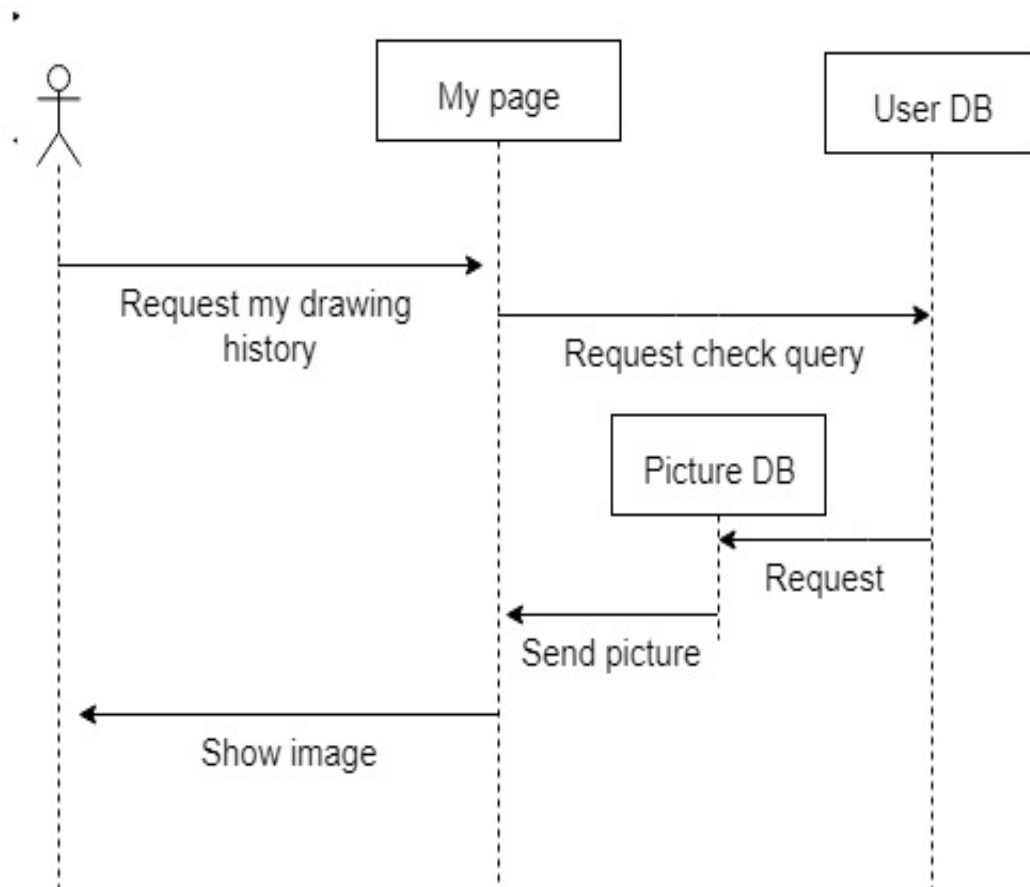


Diagram 11 My page Sequence Diagram

## 4.4 State Diagram

A. Log in with using 'google'

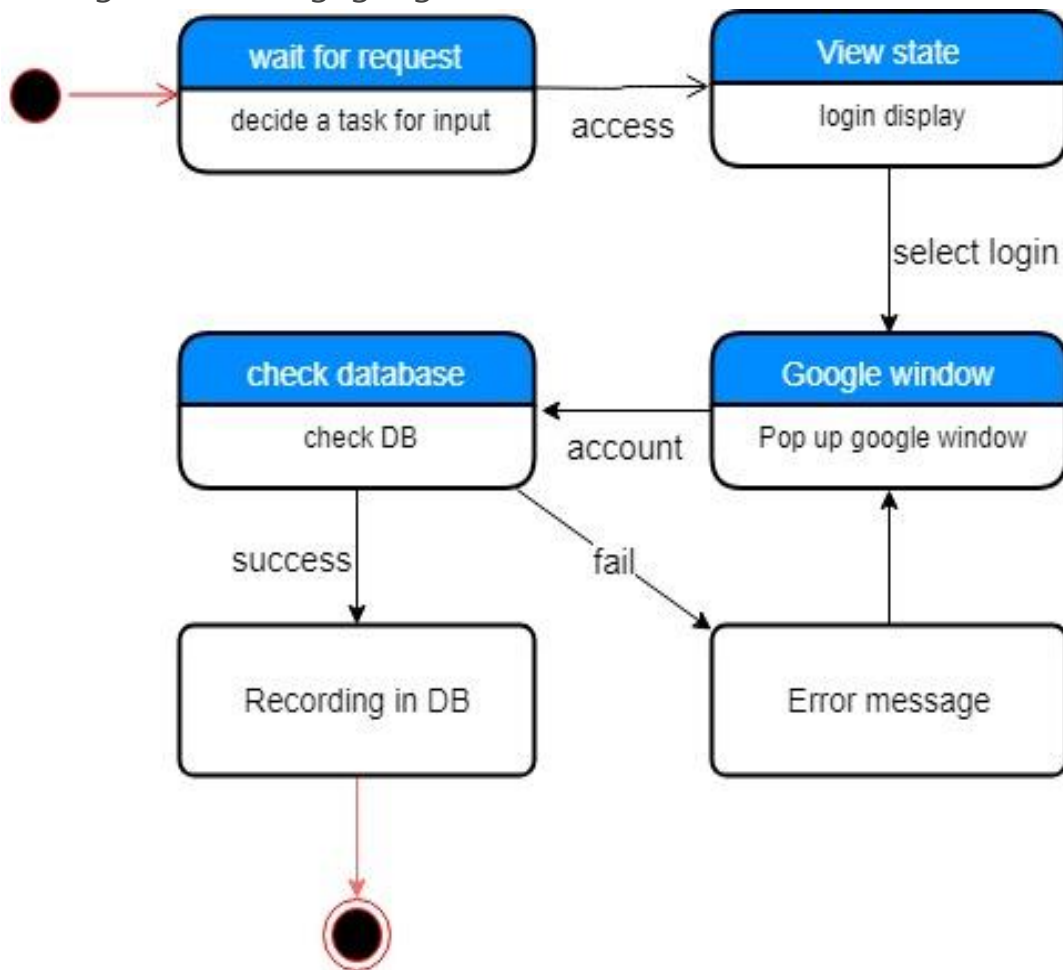


Diagram 12 Log In with Google State Diagram

## 5. Public Project System

### 5.1. Objectives

유저가 다른 유저와 공동으로 한 그림을 어느 파트를 맡아서 그리고 저장하는 System 인 Public Project System 에 대해서 설명한다. 이 시스템은 Painting 이 이루어지는 Painting sub-system 을 가지고 있다. Public Project System 의 전체적인 구조를 Class Diagram 으로 설명하고, 핵심 기능이 이루어지는 Painting sub-system 에 대해서 Sequence Diagram 과 State Diagram 을 통해 설명한다.

### 5.2. Class Diagram

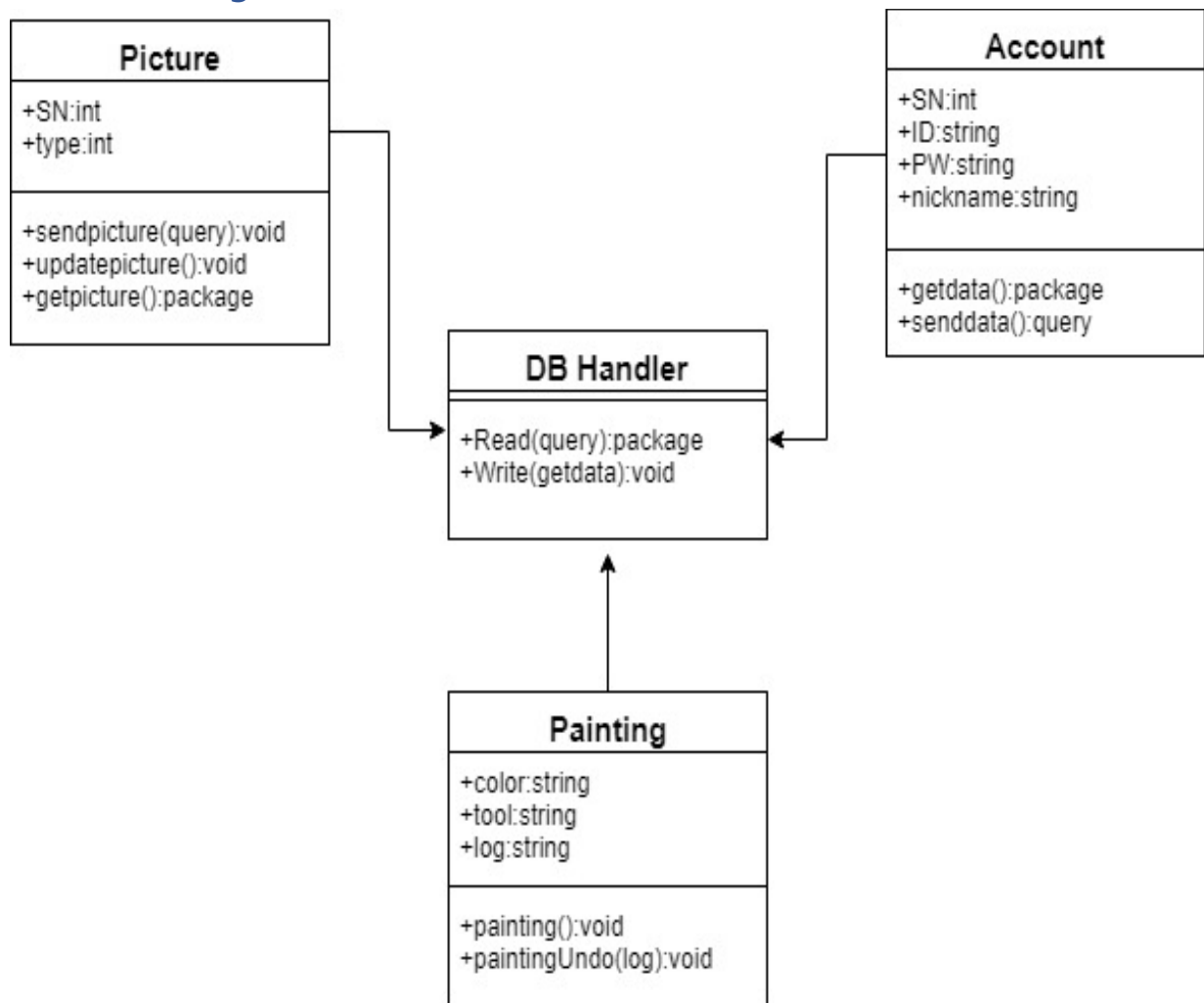


Diagram 13 Public Project Class Diagram

## A. DB Handler

### A.1. Attributes

없음

### A.2. Methods

+read(query):package: 해당하는 DB 에서 원하는 data 를 가져옴

+write(package):void: 해당하는 DB 에 data 를 저장

## B. Picture

### B.1. Attributes

+SN:int: 그림 식별번호

+type:int: 그림이 public 를 위한것인지 private 를 위한것인지 나타냄

### B.2. Methods

+sendpicture(query):void: 그림 전송

+updatepicture():void: 그림 변경부분 업데이트

+getpicture():package: DB 에서 그림을 가져온다

## C. Painting

### B.1. Attributes

+color:string: 색칠할 색을 선택

+tool:string: 색칠 도구 선택

+log:string: painting 현황을 저장

### B.2. Methods

+painting():void: 색칠을 한다

+paintingUndo(log):void: 색칠 되돌리기

## D. Account

### D.1. Attributes

+SN:int: 식별번호

+ID:string: 아이디 정보  
+PW:string: 비밀번호 정보  
+nickname:string: 닉네임 정보

## D.2. Methods

+getdata():package:DB 에서 data 를 가져온다  
+senddata():query:DB 에 data 를 보낸다

## 5.3. Sequence Diagram

### A. Painting System

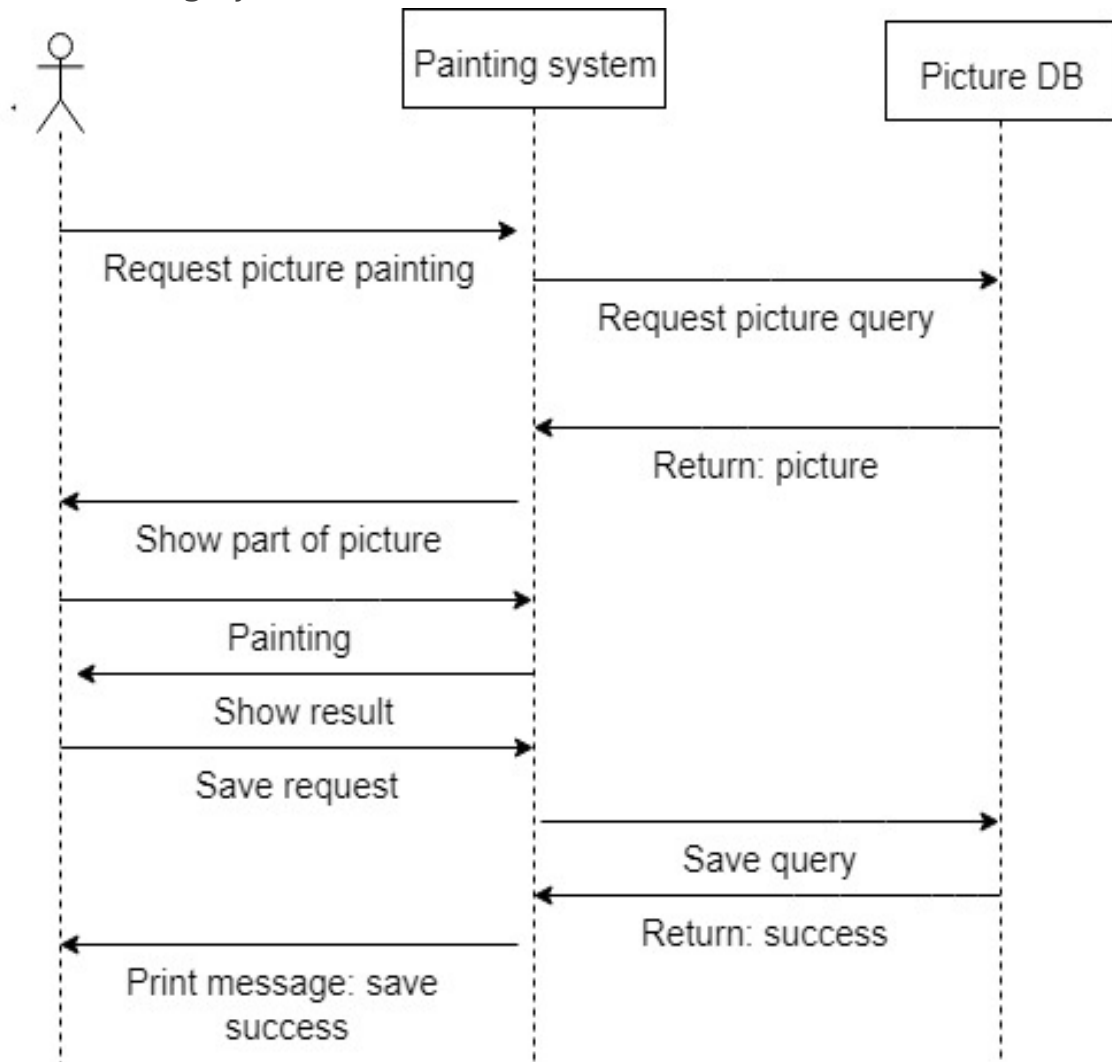


Diagram 14 Public Project Painting System Sequence Diagram



## 5.4 State Diagram

### A. Painting System

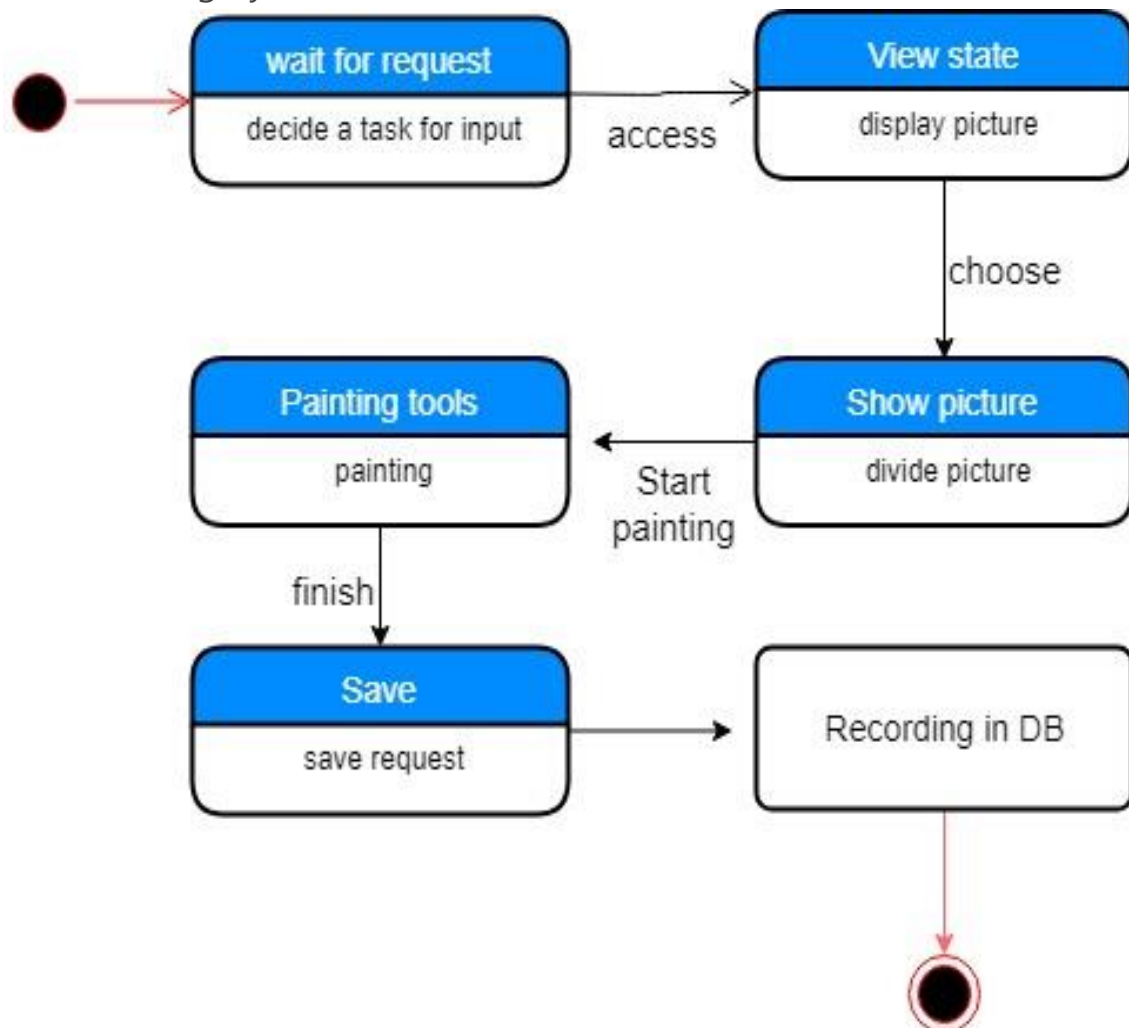


Diagram 15 Public System Painting System State Diagram

## 6. Private Project System

### 6.1. Objectives

유저가 혼자서 그림 하나를 자유롭게 색칠하고 저장하는 System 인 Private Project System 에 대해서 설명한다. 이 시스템은 Painting 이 이루어지는 Painting sub-system 을

가지고 있다. Private Project System 의 전체적인 구조를 Class Diagram 으로 설명하고, 핵심 기능이 이루어지는 Painting sub-system 에 대해서 Sequence Diagram 과 State Diagram 을 통해 설명한다.

## 6.2. Class Diagram

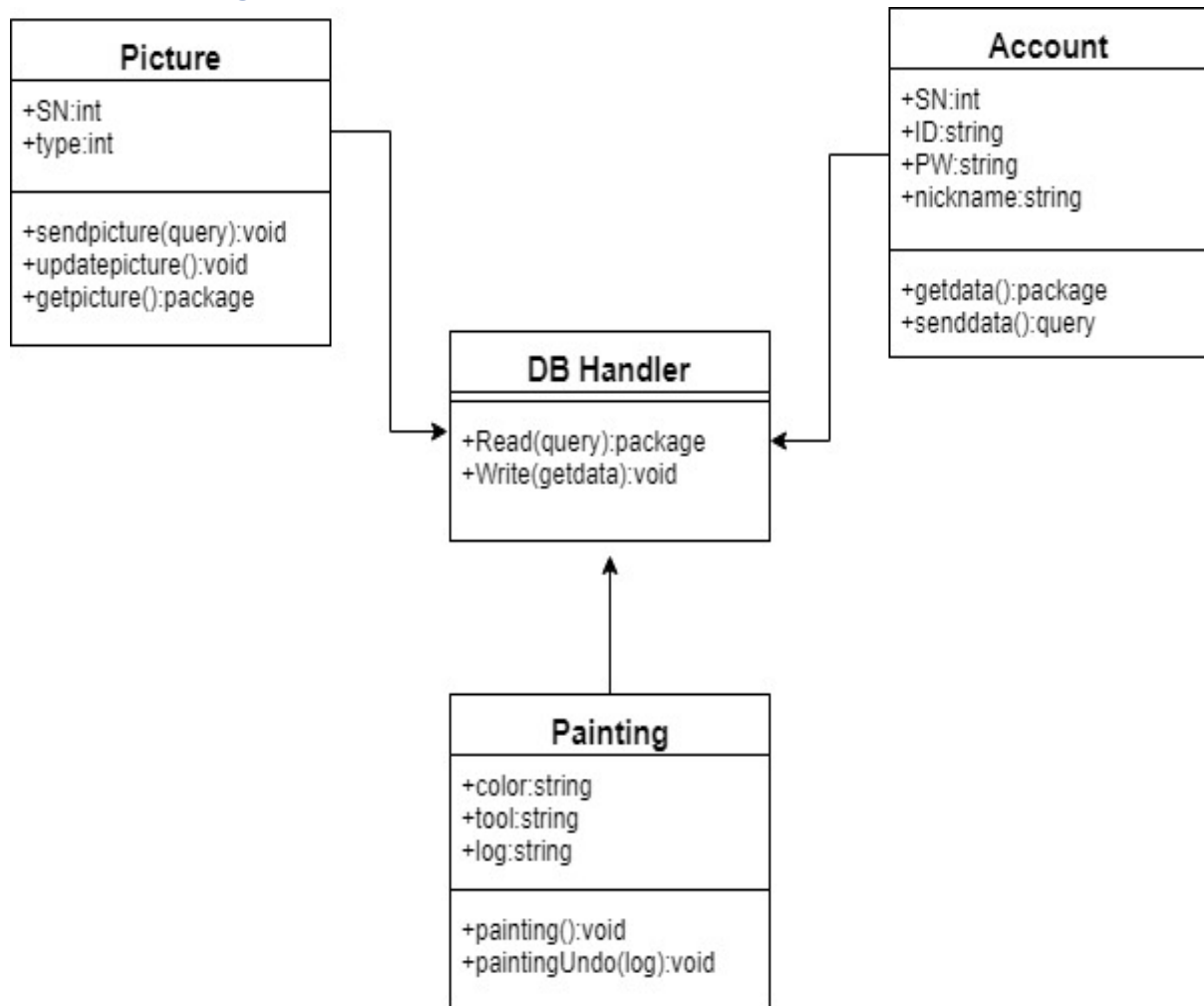


Diagram 16 Private System Class Diagram

### A. DB Handler

#### A.1. Attributes

없음

#### A.2. Methods

`+read(query):package`: 해당하는 DB 에서 원하는 data 를 가져옴

`+write(package):void`: 해당하는 DB 에 data 를 저장

## B. Picture

### B.1. Attributes

+SN:int: 그림 식별번호

+type:int: 그림이 public 를 위한것인지 private 를 위한것인지 나타냄

### B.2. Methods

+sendpicture(query):void: 그림 전송

+updatepicture():void: 그림 변경부분 업데이트

+getpicture():package: DB 에서 그림을 가져온다

## C. Painting

### B.1. Attributes

+color:string: 색칠할 색을 선택

+tool:string: 색칠 도구 선택

+log:string: painting 현황을 저장

### B.2. Methods

+painting():void: 색칠을 한다

+paintingUndo(log):void: 색칠 되돌리기

## D. Account

### D.1. Attributes

+SN:int: 식별번호

+ID:string: 아이디 정보

+PW:string: 비밀번호 정보

+nickname:string: 닉네임 정보

### D.2. Methods

+getdata():package:DB 에서 data 를 가져온다

+senddata():query:DB 에 data 를 보낸다

### 6.3. Sequence Diagram

#### A. Painting System

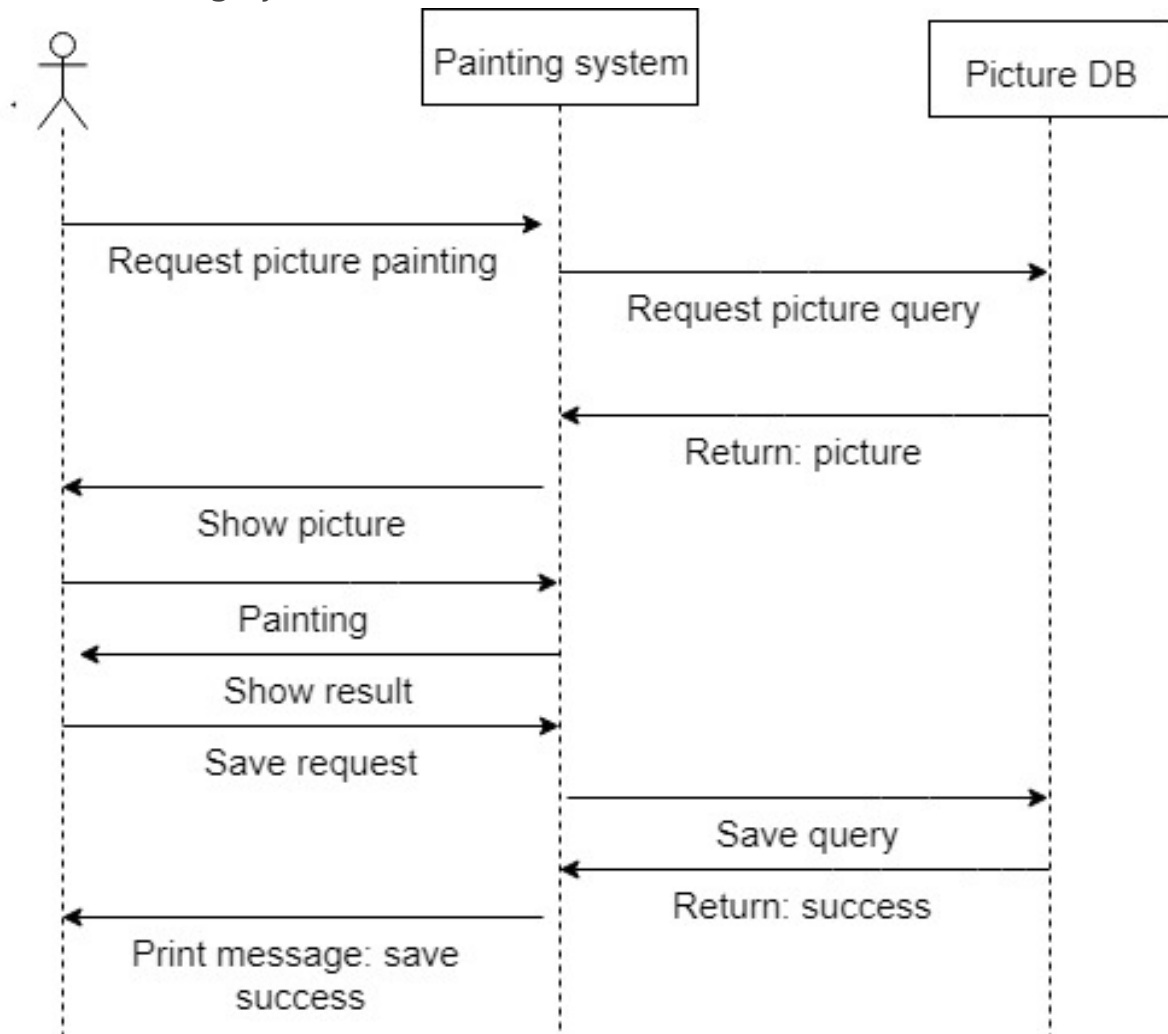


Diagram 17 Private Project Painting System Sequence Diagram

## 6.4 State Diagram

### A. Painting System

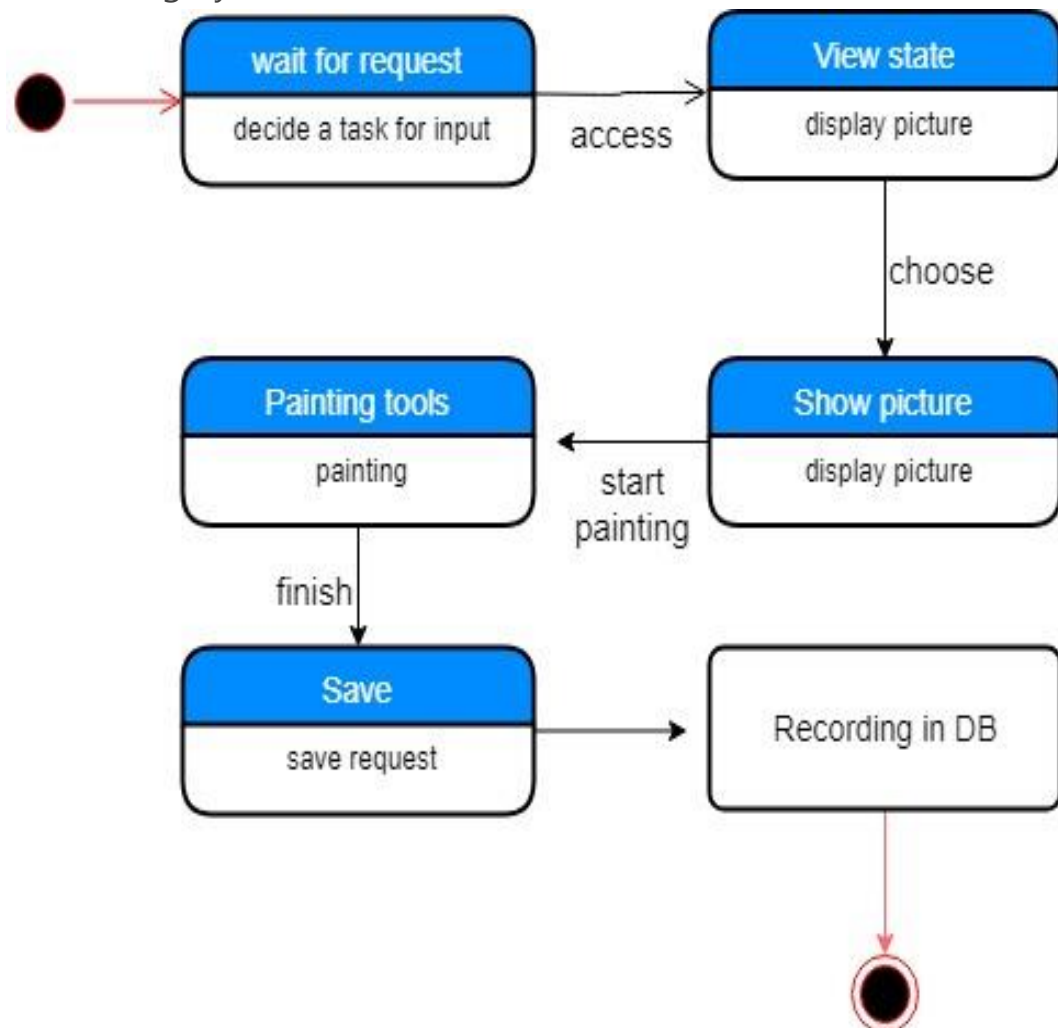


Diagram 18 Private Project Painting System State Diagram

## 7. Custom Request System

### 7.1. Objectives

유저가 혼자서 그림 하나를 자유롭게 색칠하고 저장하는 System 인 Private Project System 에 대해서 설명한다. 이 시스템은 Painting 이 이루어지는 Painting sub-system 을 가지고 있다. Private Project System 의 전체적인 구조를 Class Diagram 으로 설명하고, 핵심 기능이 이루어지는 Painting sub-system 에 대해서 Sequence Diagram 과 State Diagram 을 통해 설명한다.

### 7.2. Class Diagram

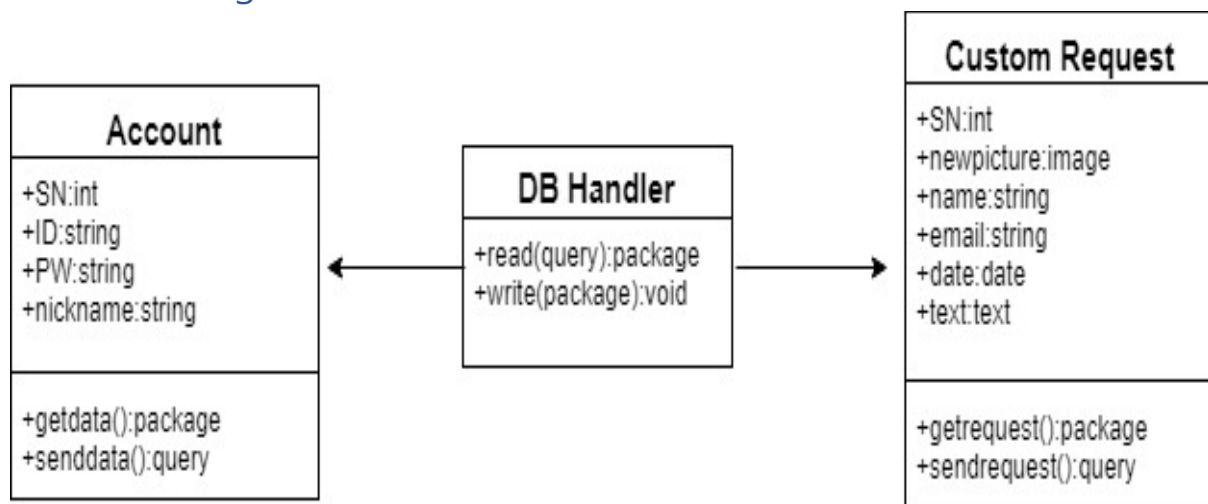


Diagram 19 Custom Request System Class Diagram

#### A. DB Handler

##### A.1. Attributes

없음

##### A.2. Methods

+read(query):package: 해당하는 DB 에서 원하는 data 를 가져옴

+write(package):void: 해당하는 DB 에 data 를 저장

#### B. Custom Request

##### B.1. Attributes

+SN:int: Custom Request 식별번호

+newpicture:image: custom request 에 포함되는 그림 정보

+name:string: custom request 를 하는 유저의 이름

+email:string: custom request 를 하는 유저의 이메일

+date:date: custom request 를 하는 그림의 요청 기간

+text:text: custom request 의 상세 내용 정보

## B.2. Methods

+getrequest():package: custom request 정보를 가져온다

+sendrequest():query: DB 에 custom request 정보를 보낸다

## C. Account

### C.1. Attributes

+SN:int: 식별번호

+ID:string: 아이디 정보

+PW:string: 비밀번호 정보

+nickname:string: 닉네임 정보

### C.2. Methods

+getdata():package:DB 에서 data 를 가져온다

+senddata():query:DB 에 data 를 보낸다

### 7.3. Sequence Diagram

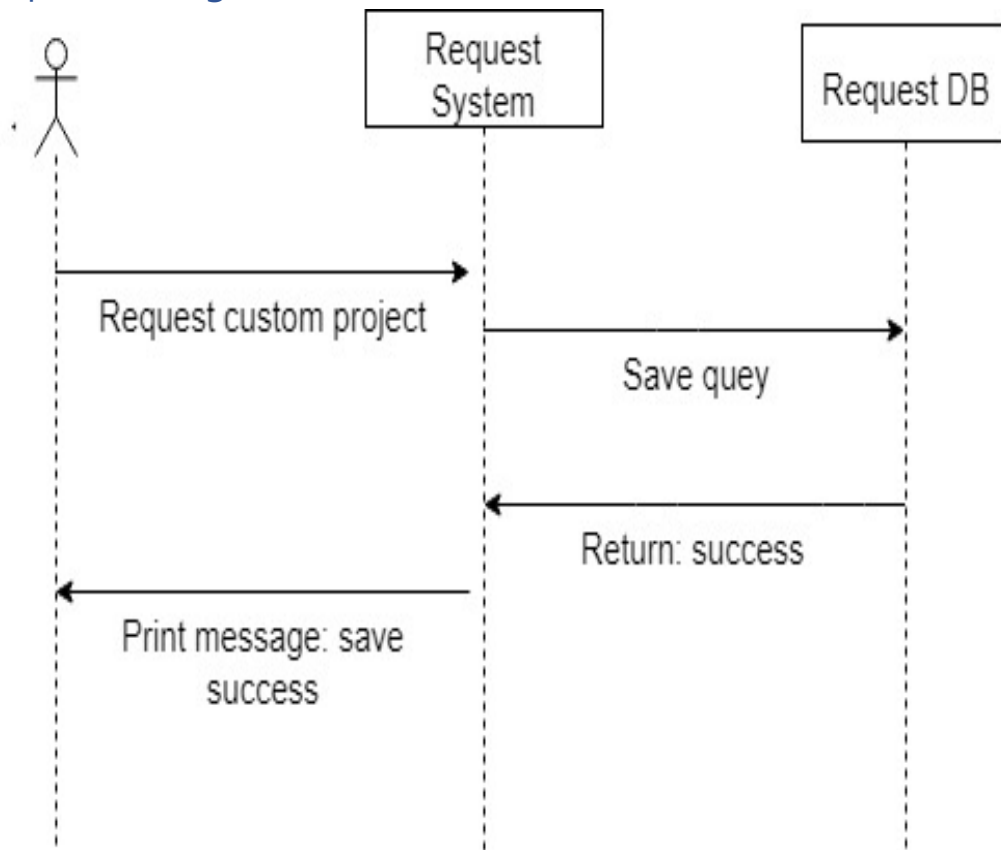


Diagram 20 Custom Request System Sequence Diagram



## 7.4 State Diagram

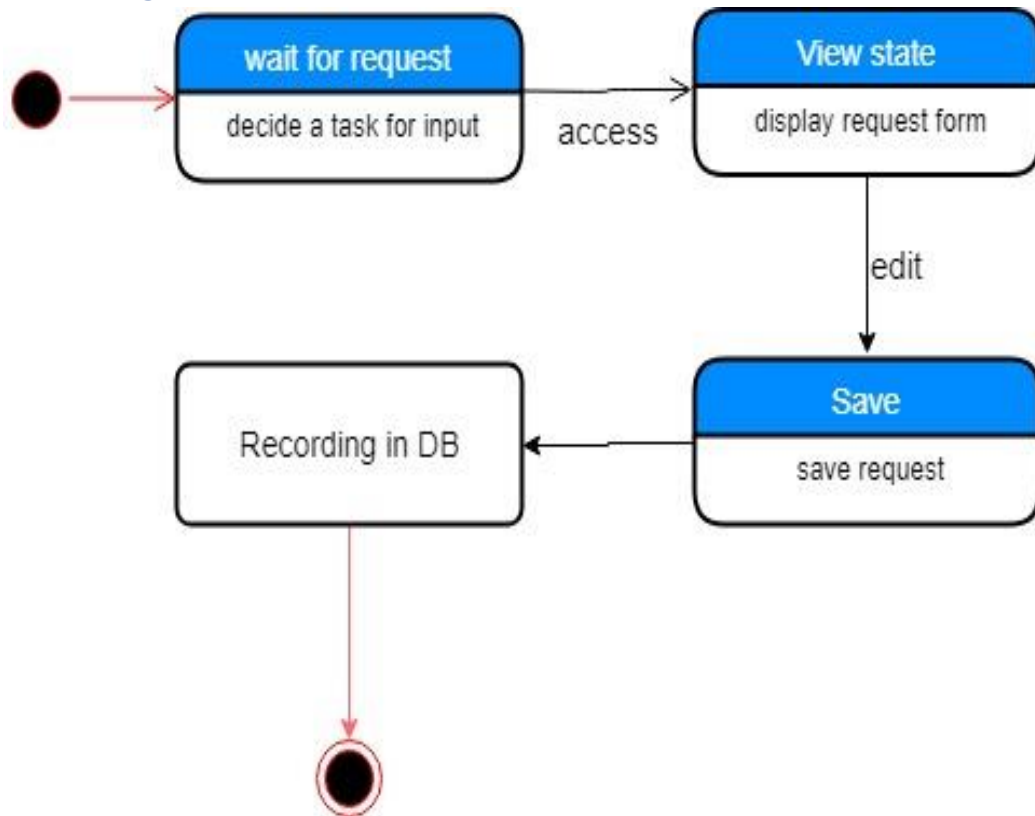


Diagram 21 Custom Request System State Diagram

## 8. Protocol Design

### 8.1 Objectives

이 항목에서는 시스템 상호간에 소통하는 프로토콜들에 대해 설명한다. 프로토콜 작성 형식은 JSON 을 기반으로 하고 요청-반응 형식의 attribute 와 value 에 대해서 설명한다.

### 8.2 JSON

JavaScript object notation(JSON)은 attribute-value pair 나 array data type 의 데이터 객체를 사람이 이해가능한 언어로 전송하는데 사용하는 표준 형식이다. 주로 인터넷에서 자료를 주고 받을 때 자료를 표현하는 방법으로 알려져 있으며 자료의 종류에 제한은 없다. 언어나 플랫폼에 독립적이기 때문에 다양한 언어에서 작업할 수 있는 장점이 있다.

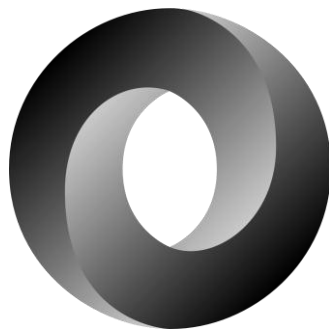


Figure 9. Logo of JSON

### 8.3 Protocol Description

#### A. Overview

Client 와 Server 사이의 요청-응답의 메시지를 protocol 의 구현 기능마다 구분하여 설명하였다. 해당 절의 표는 캡션을 생략한다.

#### B. Login Protocol

##### a. Request

Attribute	Value
Email	사용자의 email
Passwd	사용자의 password

##### b. Response

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

<b>Login_success</b>	로그인 성공 여부
----------------------	-----------

### C. Registration Protocol

#### a. Request

Attribute	Value
<b>Email</b>	사용자의 email
<b>Passwd</b>	사용자의 password
<b>Nickname</b>	사용자의 닉네임

#### b. Response

Attribute	Value
<b>Reg_success</b>	회원가입 성공 여부

### D. Using Project Select/Edit Protocol

#### a. Request

Attribute	Value
<b>category</b>	프로젝트의 종류
<b>Proj_SN</b>	프로젝트 고유번호
<b>Proj_sec</b>	프로젝트의 담당 구역 (private 인 경우 default)
<b>User_SN</b>	사용자의 고유번호

#### b. Response

Attribute	Value
<b>Reg_success</b>	Project 선택/수정 성공 여부

### E. Project View Protocol

#### a. Request

Attribute	Value
<b>category</b>	프로젝트의 종류

b. Response

Attribute	Value	
	해당 종류의 프로젝트 목록	
Project_list	Attribute	Value
	Proj_SN	프로젝트의 고유번호
	Proj_name	프로젝트의 이름
	Proj_image	프로젝트의 도안
	Proj_member	프로젝트 참여가능 인원 수
	Proj_due	프로젝트 작업 기한(private 인 경우 default)

F. Comment Post/Edit Protocol

a. Request

Attribute	Value
User_SN	작성자의 고유번호
Comment	댓글의 내용

b. Response

Attribute	Value
Reg_success	댓글 작성/수정 성공 여부

G. Comment View Protocol

a. Request

Attribute	Value
Comm_SN	댓글의 고유번호

b. Response

Attribute	Value
Writer	댓글 작성자
Comment	댓글의 내용

## H. Customized Project Insert/Edit Protocol

### a. Request

Attribute	Value
User_SN	사용자의 고유번호
Proj_name	프로젝트의 이름
Proj_image	프로젝트에 사용할 도안
Proj_member	프로젝트에 참여할 인원 수
Proj_due	프로젝트 작업 기한
Comment	프로젝트 관련 추가 요청사항

### b. Response

Proj_image	프로젝트에 사용할 도안
Reg_success	커스텀 프로젝트 등록/수정 성공 여부

## I. Customized Project Delete Protocol

### a. Request

Attribute	Value
Customize_SN	커스텀 프로젝트의 고유번호

### b. Response

Attribute	Value
Del_success	커스텀 프로젝트의 삭제 성공 여부

## J. Addressing Customized Project Protocol

### a. Request

Attribute	Value
Customize_SN	커스텀 프로젝트 요청사항의 고유번호
User_SN	사용자의 고유번호

b. Response

Attribute	Value
Reg_success	커스텀 프로젝트 요청사항의 완료 여부

K. Customized Project View Protocol

a. Request

Attribute	Value
Customize_SN	커스텀 프로젝트 요청사항의 고유번호

b. Response

Attribute	Value
User_SN	사용자의 고유번호
Proj_name	프로젝트의 이름
Proj_image	프로젝트에 사용할 도안
Proj_member	프로젝트에 참여할 인원 수
Proj_due	프로젝트 작업 기한
Comment	프로젝트 관련 추가 요청사항

L. Project Post/Edit Protocol

a. Request

Attribute	Value
Proj_SN	프로젝트의 고유번호
Proj_name	프로젝트의 이름
Proj_image	프로젝트에 사용할 도안
Proj_member	프로젝트에 참여할 인원 수(private 인 경우 default 1 로 설정)
Proj_due	프로젝트 작업 기한(private 인 경우 default 무기한으로 설정)

b. Response

Attribute	Value
Reg_success	Project 등록 성공 여부

## 10. Database Design

---

### 10.1 Objectives

Database Design 은 시스템에 필요한 데이터베이스를 설계하는 것을 목표로 한다. 먼저 전체적인 ER Diagram 을 작성하고, 이를 활용하여 Relational Schema 와 SQL DDL 을 작성한다.

### 10.2 ER Diagram

ER Diagram 에 대한 설명은 다음과 같다. Entity 는 직사각형으로, Attribute 는 타원으로, Relation 은 마름모 모양으로 표현한다. 각 Entity 의 Primary key 는 밑줄이 쳐져 있으며, Multi-valued Attribute 는 이중 타원으로 표현한다. 전체적인 ER Diagram 은 다음과 같다.

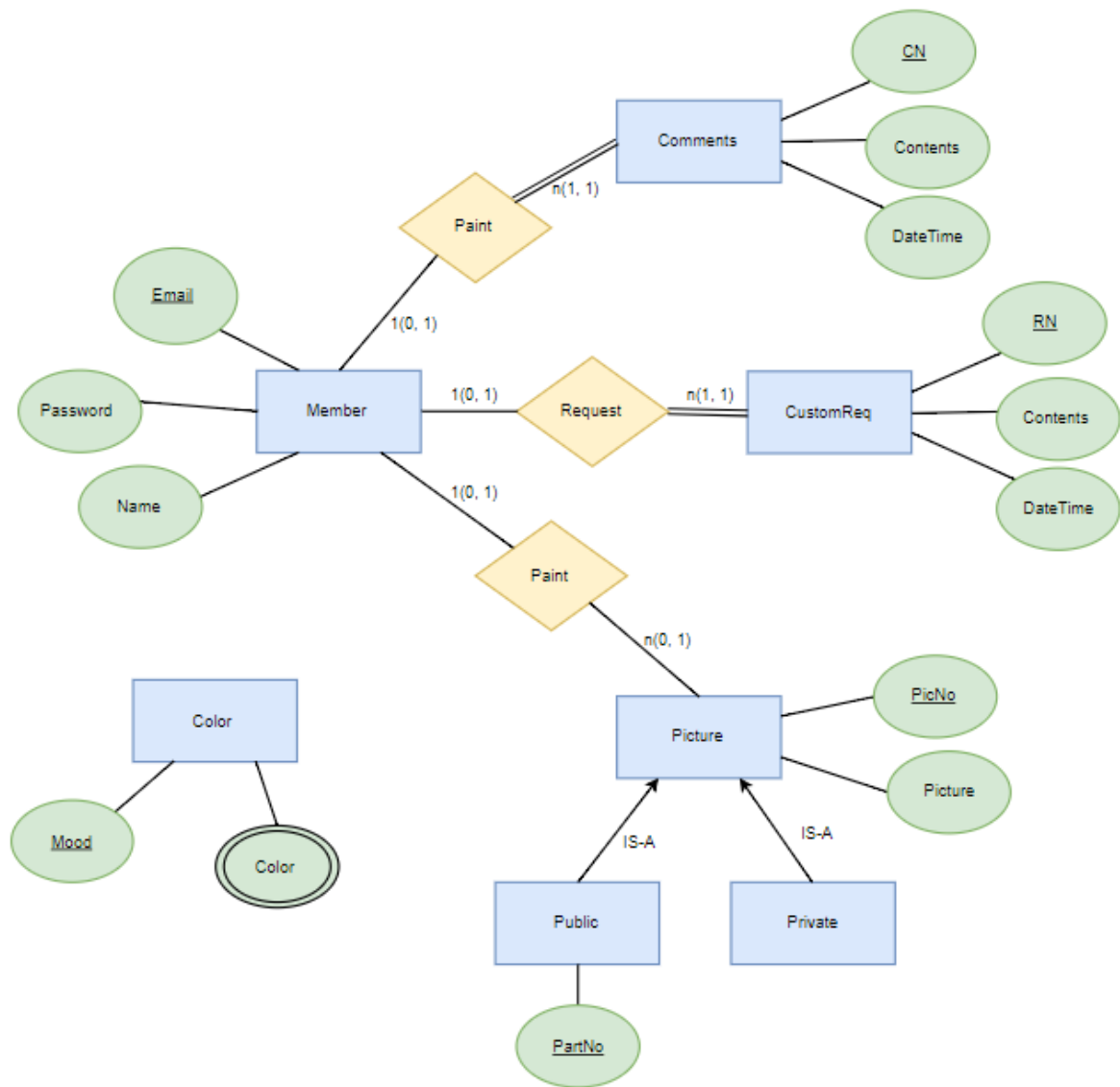
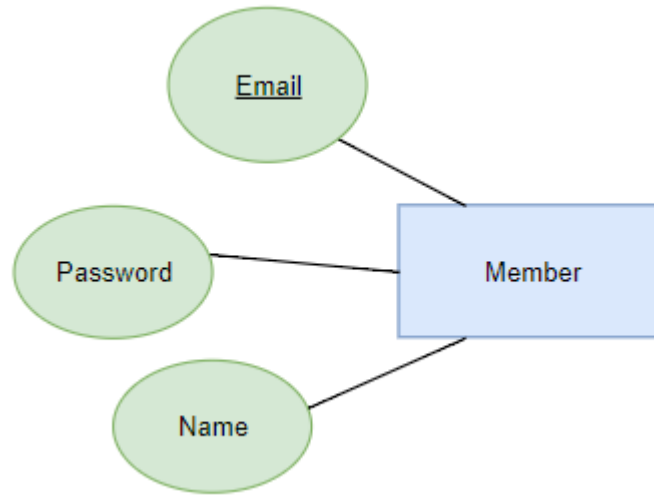


Diagram 22. The General ER Diagram of Coloride



## A. Entity

### A.1. Member



**Diagram 23. ER Diagram of Member**

회원 정보를 나타내는 Member Entity 이다. Coloride 시스템은 사용자의 쉬운 접근을 위해 최소한의 정보만을 수집한다. Email, Password, Name 의 속성을 가지고 있으며, Key 는 Email 이다.

## A.2. Picture

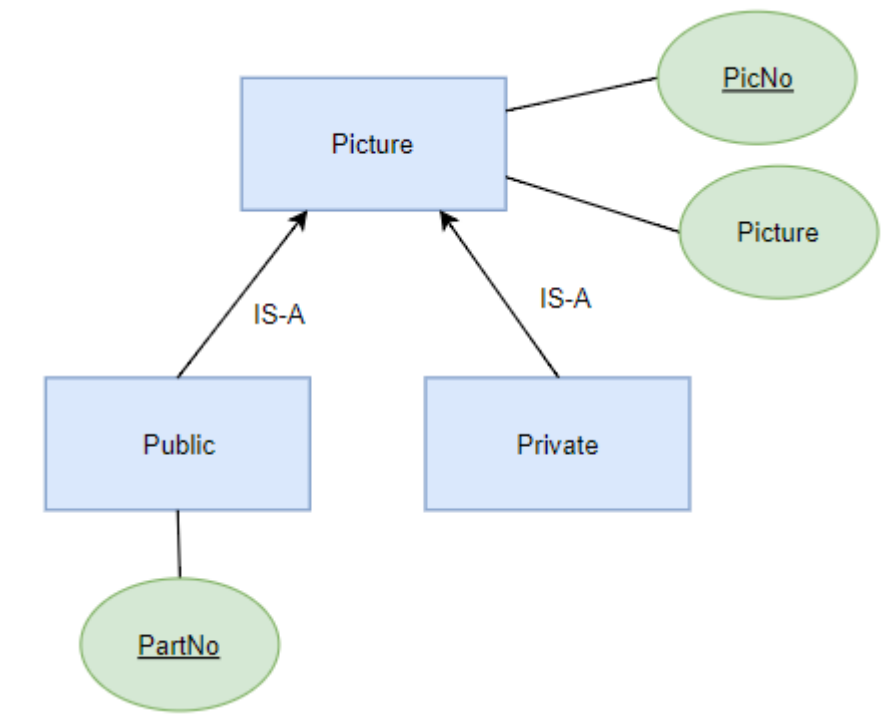


Diagram 24. ER Diagram of Picture

그림, 도안의 정보를 나타내는 picture entity 이다. Public, Private 과 IS-A 관계를 맺고 있다. Picture 는 각 도안의 고유번호를 나타내는 PicNo, 그 도안을 정보로 가지는 Picture 의 속성을 가지고 있으며, Public entity 에는 나뉜 부분의 번호인 PartNo 의 추가적인 속성을 가지고 있다. 이중 Primary Key 는 PicNo, PartNo 이다.

### A.3. Custom Request(CustomReq)

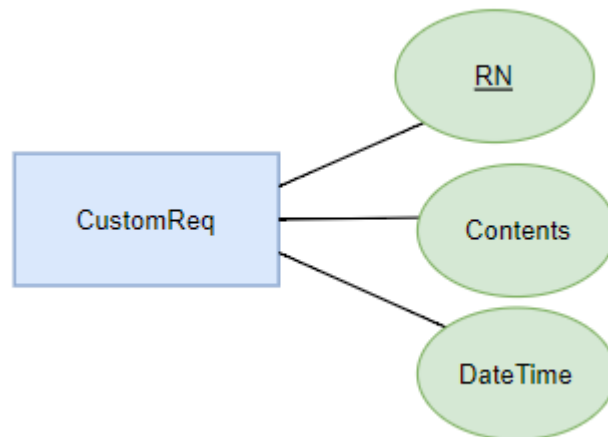


Diagram 25. ER Diagram of Custom Request

Custom Project Request 정보를 나타내는 CustomReq Entity 이다. Request Number 의 앞글자를 딴 RN, 내용을 나타내는 Contents, 신청 시간을 나타내는 DateTime 을 속성으로 갖는다. PK 는 RN 이다.

### A.4. Comments

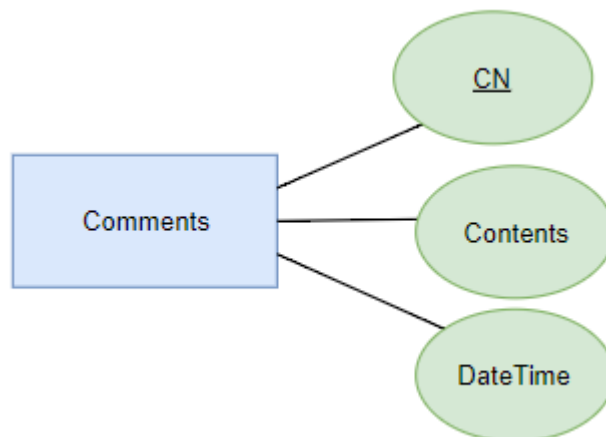


Diagram 26. ER Diagram of Comments

댓글을 나타내는 Comments Entity 이다. Comments Number 의 앞글자를 딴 CN, 내용을 나타내는 Contents, 작성 시간을 나타내는 DateTime 을 속성으로 갖는다. Primary key 는 CN 이다.

## A.5. Colors

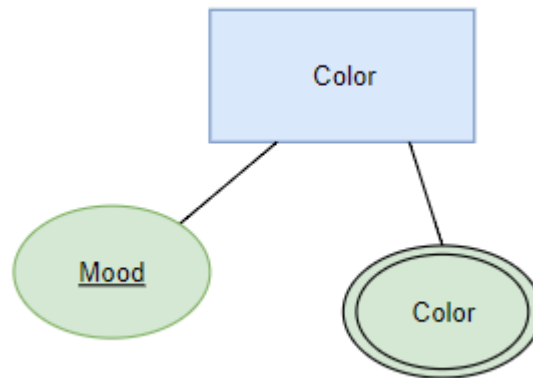


Diagram 27. ER Diagram of Colors

기분에 따른 색상 추천을 위한 색상 정보를 나타내는 Color entity 이다. 기분을 나타내는 Mood, 색상 정보를 Color 를 속성으로 가지며, 이중 Color 는 multi-valued attribute 로 여러 값을 가진다. Primary key 는 mood 이다.

## B. Relation

### B.1. Painting Picture

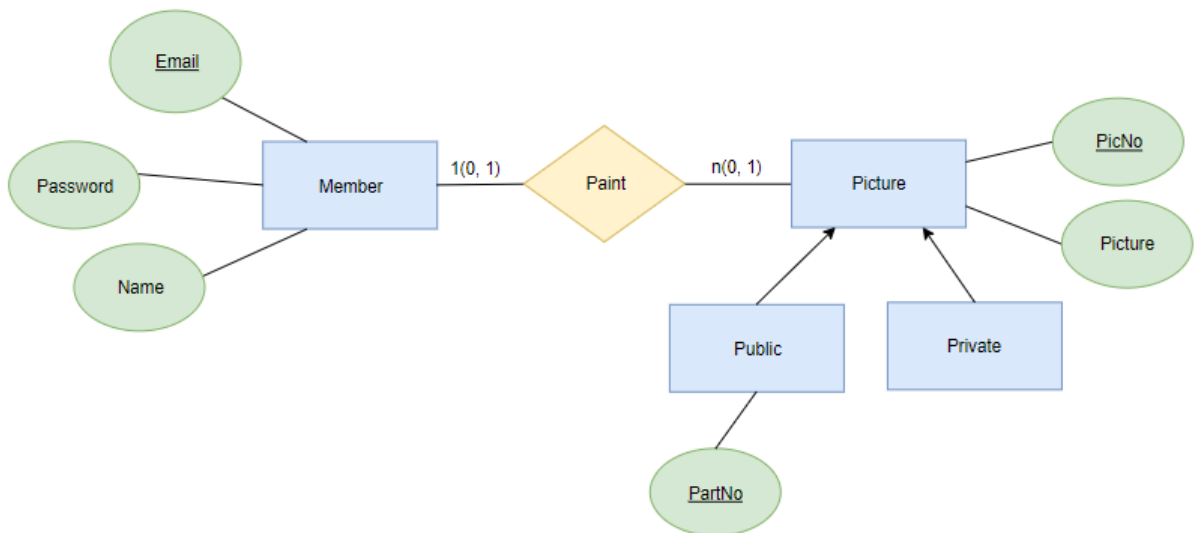


Diagram 28. ER Diagram of Painting Picture

회원이 그림을 그리는 관계를 나타내는 Painting Picture Relation 이다. 한 명의 회원은 여러 그림을 그릴 수 있다.

## B.2. Custom Request

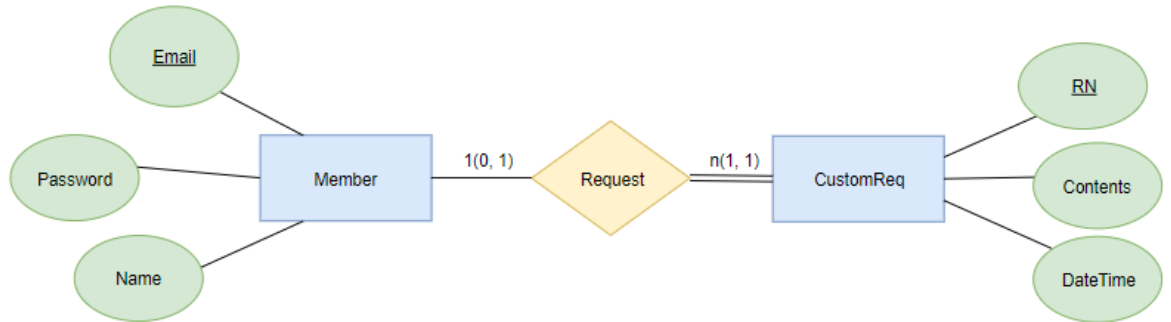


Diagram 29. ER Diagram of Custom Request

회원이 Custom Project Request 를 하는 관계이다. 모든 Custom project request 는 한 명의 회원에 의해 이루어지며, 한 명의 회원이 여러 Request 를 수행할 수 있다.

## B.3. Leave Comments

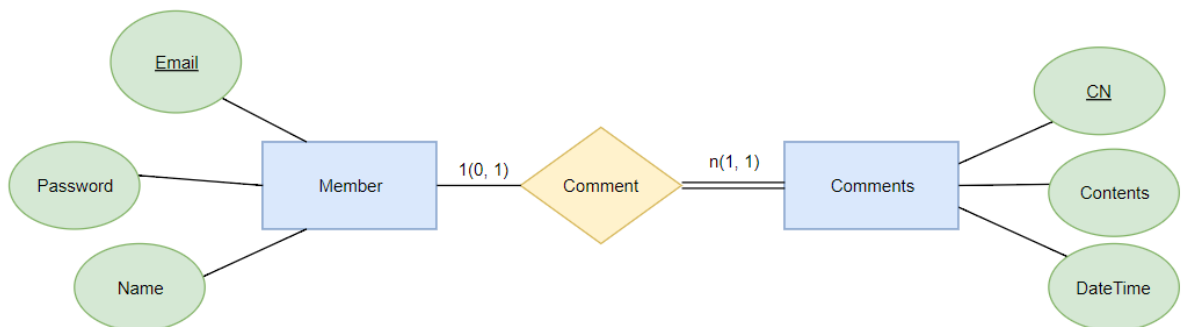


Diagram 30. ER Diagram of Leave Comments

회원이 댓글을 남기는 것을 나타내는 관계이다. 모든 댓글은 한 명의 회원에 의해 작성되며, 한 회원이 여러 댓글을 남길 수 있다.

## 10.3 Relational Schema

### A. Member

<u>Email</u>	Password	Name	
--------------	----------	------	--

Primary Key (PK): Email

Foreign Key (FK): 없음

Function Dependency (FD): Email -> {Password, Name}

Description: 회원에 대한 테이블이다. 모든 속성에 Null 값을 허용하지 않는다.

### B. Comments

<u>CN</u>	Contents	DateTime	Writer_Email
-----------	----------	----------	--------------

Primary Key (PK): CN

Foreign Key (FK): Writer\_Email

Function Dependency (FD): CN -> {Contents, DateTime, Writer\_Email}

Description: 댓글에 대한 테이블이다. 자동 증가되는 CN 값을 PK로 가지며, 모든 값에 Null 값을 허용하지 않는다. Member의 Email을 FK로 가져온다.

### C. Picture

<u>PicNo</u>	Picture
--------------	---------

Primary Key (PK): PicNo

Foreign Key (FK): 없음

Function Dependency (FD): PicNo -> Picture

Description: 다른 테이블의 FD를 해결하기 위한, Picture에 대한 테이블이다. 모든 값에 Null 값을 허용하지 않는다.

### D. Public Project

<u>PicNo</u>	PartNo	Painter_Email
--------------	--------	---------------

Primary Key (PK): PicNo, PartNo

Foreign Key (FK): Painter\_Email

Function Dependency (FD):

PicNo, PartNo -> {Picture, Painter\_Email}

Description: Public project에 대한 테이블이다. PicNo, PartNo를 PK로 가지며, Member의 Email을 FK로 가져온다. 모든 값에 Null 값을 허용하지 않는다.

### E. Private Project

<u>PN</u>	Painter_Email	PicNo
-----------	---------------	-------

Primary Key (PK): PN

Foreign Key (FK): Painter\_Email, PicNo

Function Dependency (FD):

PN -> {Painter\_Email, PicNo}

Description: Private project 에 대한 테이블이다. 모든 값에 Null 값을 허용하지 않으며, Member 의 Email, Picture 의 PicNo 를 FK 로 가져온다.

### F. Custom Request

<u>RN</u>	Contents	DateTime	Requester_Email
-----------	----------	----------	-----------------

Primary Key (PK): RN

Foreign Key (FK): Requester\_Email

Function Dependency (FD): RN -> {Contents, DateTime, Requester\_Email}

Description: Custom Project Request 에 대한 테이블이다. 모든 값에 Null 값을 허용하지 않으며, Member 의 Email 을 FK 로 가져온다.

### G. Colors

<u>Mood</u>	Colors
-------------	--------

Primary Key (PK): Mood

Foreign Key (FK): 없음

Function Dependency (FD): Mood -> Colors

Description: 색상 추천을 위한 색상 테이블이다. 모든 값에 null 값을 허용하지 않으며, Colors 는 Multi-valued Attribute 로 여러 값을 가진다.

## 10.4 Normalization

Public Project, Private Project 에서 초기 테이블에서는 PicNo->Picture 이라는 경우가 있었기 때문에, 이를 Picture 테이블로 분리하여 해결하였다.



## 12. Testing Plan

---

### 12.1 Objectives [1]

Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. When we test software, we execute a program using artificial data called test case. We check the results of the test run for errors, anomalies, or information about the program's non-functional attributes.

### 12.2 Testing Policy [1]

'Coloride' 시스템의 테스트는 development testing, release testing, user testing 으로 크게 3 가지 단계를 거친다. 그 중 development testing 의 과정을 살펴보면 unit testing, component testing, system testing 으로 점점 규모가 증가하는 방향으로 확인한다. 이 단락에선 시스템 테스트 단계에 대한 각각의 정의를 간단하게 설명한다. 이후 Development testing 의 각각 단계에 대하여 'Coloride'에 적용해야 할 test 과제에 대해서 기술하고자 한다.

#### 1) Development Testing

*Development testing, where the system is tested during development to discover bugs and defects. System designers and programmers are likely to be involved in the testing process.*

#### *Release Testing*

*Release testing, where a separate testing team tests a complete version of the system before it is released to users. The aim of release testing is to check that the system meets the requirements of the system stakeholders.*

#### *User Testing*

User testing, where users or potential users of a system test the system in their own environment. For the software products, the "user" may be an internal marketing group that decides if the software can be marketed, released and sold. Acceptance testing is one type of user testing where the

customer formally tests a system to decide if it should be accepted from the system supplier or if further development is required.

## 12.3 Test case

### A. User Management System

#### A.1. Sign up

- 1) User : 회원 가입 기능을 이용하여 회원 가입을 시도한다.
- 2) 시스템 동작 : Member DB 에 해당 Email 이 이미 존재하는지 확인한다.
  - 2-1) (회원가입 성공) : 해당 데이터를 Member DB 에 추가하고, 시스템 알림 "회원가입 완료"를 출력한다.
  - 2-2) (회원가입 실패) : 시스템 알림 "회원가입 실패"를 출력한다.

#### A.2. Log in

- 1) User : 회원가입에 입력한 정보를 통하여 로그인을 시도한다.
- 2) 시스템 동작 : 입력받은 정보와 Member DB 에 저장되어 있는 데이터를 비교하여, 로그인 정보가 일치하는지 확인한다.
  - 2-1) (로그인 성공) : 시스템 알림 "로그인 성공"을 출력하며, 해당 정보로 로그인한다.
  - 2-2) (로그인 실패) : 시스템 알림 "로그인 실패"를 출력한다.
  - 2-3) (관리자 로그인) : 관리자 정보로 로그인 하였을 때이며, 관리자 페이지로 이동한다.

#### A.3. Log out

- 1) User : 로그인을 한 상태에서 로그아웃을 시도한다.
- 2) 시스템 동작 : 시스템 알림 "로그아웃 되었습니다"를 출력 후, 메인 화면으로 이동한다.

#### A.4. Change Password

- 1) User : 로그인 한 상태에서, 회원정보 관리 메뉴에 존재하는 비밀번호 변경 기능을 시도한다.
- 2) 시스템 동작 : 현재 비밀번호와 새 비밀번호, 새 비밀번호 확인을 입력받아, Member DB 에 있는 비밀번호와 현재 비밀번호가 같은지, 그리고 새 비밀번호와 새 비밀번호 확인이 같은지 비교한다.
  - 2-1) 변경 성공 : 시스템 알림 "비밀번호 변경 성공"을 출력하며, Member DB 에 변경사항을 update 한다.
  - 2-2) 변경 실패 : 시스템 알림 "비밀번호 변경 실패"를 출력한다.

#### A.5. Cancel Registration

- 1) User : 로그인한 상태에서, 회원정보 관리메뉴에 존재하는 회원 탈퇴 기능을 시도한다.
- 2) 시스템 동작 : 시스템 알림 "정말 탈퇴하시겠습니까?"를 출력하며, 비밀번호 입력창을 출력한다.
- 3) User : 비밀번호를 입력한다.
- 4) 시스템 동작 : 입력받은 비밀번호와 Member DB 의 정보를 비교한다.
  - 4-1) (회원 탈퇴 성공) : 시스템 알림 "탈퇴 성공"을 출력하며, 로그아웃한 후 Member DB 에 해당 정보를 삭제한다.
  - 4-2) (회원 탈퇴 실패) : 시스템 알림 "탈퇴 실패"를 출력하고, 회원 관리 메뉴로 이동한다.

### B. Project Management System

#### B.1. 비로그인 상태에서 Public Project 기능에 접근할 때

- 1) User : 메인 화면의 Public Project 기능에 접근한다.
- 2) 시스템 동작 : Picture DB 에서 Public Project 에 해당하는 그림의 PicNo 를 조회하고, 아직 그려지지 않은 PartNo 를 임의 선택해서 그 부분의 Painting 화면으로 이동한다. 그 후, Painting DB 의 painter 를 업데이트한다.

## B.2. 로그인 상태에서 Public Project 기능에 접근할 때

- 1) User : 메인 화면의 Public Project 기능에 접근한다.
- 2) 시스템 동작 : Picture DB 에서 Public Project 에 해당하는 그림의 PicNo 를 조회하고 Painter 부분을 확인하여 로그인한 사용자가 Public Project 에 참여한 적이 있는지 확인한다.

2-1) (Public project 를 처음 참여하는 경우) : 아직 Painter 가 없는 PartNo 를 임의 선택하여, 해당 그림의 Painting 화면으로 이동한다. 그 후, Picture DB 의 Painter 를 업데이트한다.

2-2) (이미 Public project 에 참여한 경우) : 해당 그림의 Painting 화면으로 이동한다.

## B.3. Private Project 기능에 접근할 때

- 1) User : 메인 화면에 존재하는 Private Project 기능에 접근한다.
- 2) 시스템 동작 : 도안 선택 페이지로 이동한다. Picture DB 를 참조하여 목록을 출력한다.
- 3) User : 도안들 중에서 하나를 선택한다.
- 4) 시스템 동작 : 해당 도안의 Painting 화면으로 이동한다.

## B.4. Painting 시스템의 기능을 사용할 때

- 1) User : Painting tools 를 이용하여 그리기를 수행한다.
- 2) 시스템 동작 : 해당 화면에 변경사항을 바로 출력해준다.

## B.5. 사용자가 Save 기능을 이용하여 변경사항을 저장하려 할 때

- 1) User : Painting 시스템에 존재하는 저장 기능을 시도한다.

2) 시스템 동작 : 사용자가 로그인한 상태인지 아닌지를 확인한다. 그 후, Picture DB 에 변경사항을 저장한다.

2-1) (비로그인 상태) : 시스템 알림 "로그인을 하십시오"를 출력한다. 그 후, Picture DB 에 변경사항을 저장한 후 로그인 화면으로 넘어가며, 로그인에 성공한 경우 Painter 를 업데이트한다.

2-2) (로그인 상태) : Picture DB 에 변경사항을 업데이트한다.

#### B.6. 사용자가 색상 추천 기능을 이용할 때

1) User : 색상 추천 기능을 이용하여, 현재 기분을 입력하거나, 설문조사 기능으로 이동한다.

2-1) (기분을 입력했을 때) : Color DB 를 참조하여, 해당 Mood 에 맞는 색상을 출력한다.

2-2) (설문조사 기능으로 이동) : 설문조사 페이지를 로드한다.

3) User : 설문조사를 완료한 후 확인 버튼을 누른다.

4) 시스템 동작 : 설문조사 결과에 따라 Mood 를 결정한 다음, Color DB 를 참조하여 해당 Mood 에 맞는 색상을 출력한다.

### C. Custom Project Request System

#### C.1. 비로그인 사용자가 Custom Project Request 를 통해 신청을 하는 경우

1) User : 이름, 이메일, 내용을 입력한 후 신청을 시도한다.

2) 시스템 동작 : 모든 사항이 정상적으로 작성되었는지 확인한 후, CustomReq DB 에 이를 추가한다. 그 후 시스템 알림 "신청 완료"를 출력하고, 메인 화면으로 돌아간다.

#### C.2. 로그인 사용자가 Custom Project Request 를 통해 신청을 하는 경우

1) User : 내용을 작성한 후 신청을 시도한다.

- 2) 시스템 동작 : 모든 사항이 정상적으로 작성되었는지 확인 후, CustomReq DB 에 이를 추가한다. 이 때, Requester\_Email 은 Member DB 의 로그인한 정보를 참조한다. 그 후 시스템 알림 "신청 완료"를 출력하고, 메인 화면으로 돌아간다.

## D. Comments Systems

### D.1. 비로그인 사용자가 댓글을 남길 때

- 1) User : 닉네임, 비밀번호, 내용을 작성한 후 댓글 작성을 시도한다.
- 2) 시스템 동작 : 모든 사항이 정상적으로 작성되었는지 확인한다.
  - 2-1) (작성 성공) : Comments DB 에 이를 추가한다. 그 후, 시스템 알림 "댓글 등록 완료"를 출력한 후 댓글 메인 화면으로 돌아간다.
  - 2-2) (작성 실패) : 시스템 알림 "등록 실패"를 출력한다.

### D.2. 로그인 사용자가 댓글을 남길 때

- 1) User : 내용을 작성한 후 댓글 작성을 시도한다.
- 2) 시스템 동작 : 내용이 정상적으로 작성되었는지를 확인한다.
  - 2-1) (작성 성공) : Comments DB 에 이를 추가한다. 이 때, Comments\_name 은 Member DB 의 로그인 정보를 참조한다. 그 후, 시스템 알림 "댓글 등록 완료"를 출력한 후 댓글 메인 화면으로 돌아간다.
  - 2-2) (작성 실패) : 시스템 알림 "등록 실패"를 출력한다.

### D.3. 비로그인 사용자가 댓글을 삭제할 때

- 1) User : 해당 댓글의 삭제 버튼을 누른다.
- 2) 시스템 동작 : 비밀번호 입력 창을 팝업한다.
- 3) User : 비밀번호를 입력 후, 삭제를 시도한다.

4) 시스템 동작 : 입력받은 비밀번호와 Comments DB 의 정보를 비교하여 비밀번호가 맞는지 확인한다.

4-1) (삭제 성공) : 시스템 알림 "삭제 성공"을 출력하고, 해당 댓글을 Comments DB 에서 삭제한다. 그 후, 댓글 메인 화면으로 돌아간다.

4-2) (삭제 실패) : 시스템 알림 "삭제 실패"를 출력하고, 댓글 메인 화면을 새로고침한다.

#### D.4. 로그인 사용자가 댓글을 삭제할 때

1) User : 댓글 삭제 버튼을 통해 해당 댓글 삭제를 시도한다. 이 때, 본인이 작성한 댓글에만 삭제 버튼이 활성화된다.

2) 시스템 동작 : 시스템 알림 "삭제하시겠습니까?"를 출력하고, 확인 팝업창을 띄운다.

3) User : 확인 버튼을 누른다.

4) 시스템 동작 : 해당 댓글을 Comments DB 에서 삭제하고, 댓글 메인 화면을 새로고침한다.

## 12. Development Environment

---

### 12.1 Objectives

The development environment section describes the environment used by the development, such as the language used and the open source project and the back-end server.

### 12.2 Bootstrap (front-end framework)

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.



Figure 10. Relational Environment of Bootstrap

### 12.3 PHP+MYSQL(back-end)

PHP is server side back end programming language. it executes in server along with maximum all available web servers like Apache, IIS (Internet Information Server) etc., and return the response as required MIME type. it is a Pre Process Hypertext, we could do many things on server by using PHP on server and co-ordinate with DB server for CURD (Create, Update, Read, Delete) actions.





**Figure 11. Logo of PHP**

MySQL is the world's most popular open source database, enabling the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications. It is an integrated transaction safe, ACID-compliant database with full commit, rollback, crash recovery, and row-level locking capabilities. MySQL delivers the ease of use, scalability, and high performance, as well as a full suite of database drivers and visual tools to help developers and DBAs build and manage their business-critical MySQL applications. MySQL is developed, distributed, and supported by Oracle



**Figure 12. Logo of MySQL**

### 13.3 GitHub (Version Management)

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. Through GitHub, our team can easily manage each version of the code, although it is all tested on the server before uploading it to GitHub. However, if something goes wrong with the server, it can still be recovered from GitHub.



Figure 13. Logo of GitHub

## 13. Develop Plan

---

### 13.1 Objectives

The development plan section introduces Gantt charts to the general development process.

### 13.2 Schedule

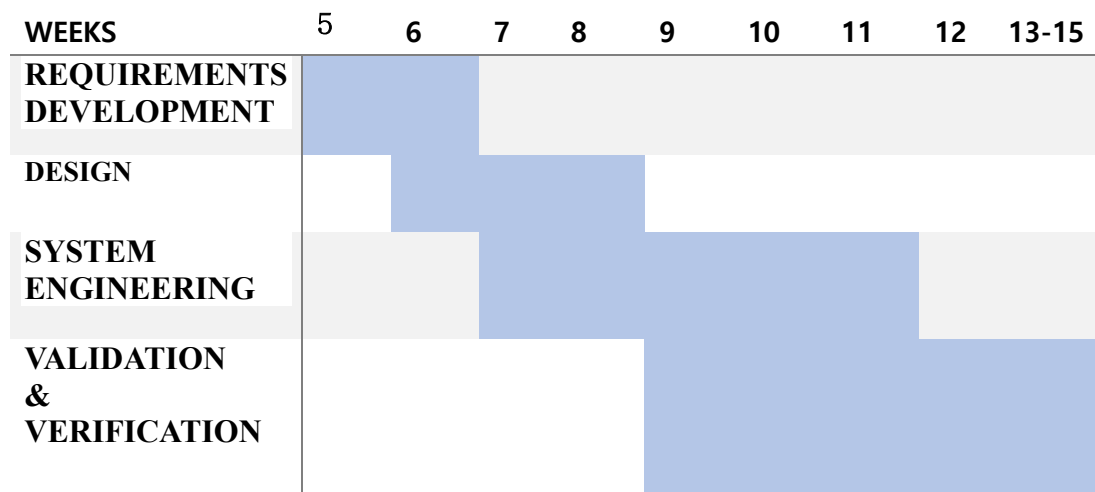


Figure 14. Project Schedule

Our team first analyzed the ideas and came up with the real requirements of the project. Then the project architecture was designed, such as the language to be used and the front and back end of the website. At the same time, various documents for the project need to be produced.

## 14. Index

---

### 14.1 Figure Index

Figure 1. Logo of UML.....	11
Figure 2. Example of Package Diagram.....	12
Figure 3. Example of Deployment Diagram .....	13
Figure 4. Example of Class Diagram .....	14
Figure 5. Example of State Diagram .....	14
Figure 6. Example of Sequence Diagram.....	15
Figure 7. Example of ER Diagram.....	16
Figure 8. Captured Image of Flow Chart Maker & Online Diagram Software.....	17
Figure 9. Logo of JSON.....	42
Figure 10. Relational Environment of Bootstrap .....	64
Figure 11. Logo of PHP.....	65
Figure 12. Logo of MySQL.....	65
Figure 13. Logo of GitHub.....	66
Figure 14. Project Schedule .....	67

### 14.2 Diagram Index

Diagram 1 System Organization Block Diagram.....	18
Diagram 2 User Management System Architecture .....	19
Diagram 3 Color history sub-system.....	20
Diagram 4 Public Project System Architecture .....	21
Diagram 5 Public Project Picture Divider Sub-System.....	21
Diagram 6 Private Project System.....	22
Diagram 7 Custom Request System Architecture .....	23

Diagram 8 Overall Deployment Diagram .....	24
Diagram 9 User Management System Class Diagram.....	25
Diagram 10 Log in with Google Sequence Diagram .....	27
Diagram 11 My page Sequence Diagram.....	28
Diagram 12 Log In with Google State Diagram.....	29
Diagram 13 Public Project Class Diagram.....	30
Diagram 14 Public Project Painting System Sequence Diagram.....	32
Diagram 15 Public System Painting System State Diagram.....	33
Diagram 16 Private System Class Diagram.....	34
Diagram 17 Private Project Painting System Sequence Diagram .....	36
Diagram 18 Private Project Painting System State Diagram.....	37
Diagram 19 Custom Request System Class Diagram .....	38
Diagram 20 Custom Request System Sequence Diagram.....	40
Diagram 21 Custom Request System State Diagram .....	41
Diagram 22. The General ER Diagram of Coloride .....	48
Diagram 23. ER Diagram of Member.....	49
Diagram 24. ER Diagram of Picture .....	50
Diagram 25. ER Diagram of Custom Request .....	51
Diagram 26. ER Diagram of Comments.....	51
Diagram 27. ER Diagram of Colors .....	52
Diagram 28. ER Diagram of Painting Picture.....	52
Diagram 29. ER Diagram of Custom Request .....	53
Diagram 30. ER Diagram of Leave Comments.....	53

## 15. References

---

“What Is UML.” *What Is UML | Unified Modeling Language*, [www.uml.org/what-is-uml.htm](http://www.uml.org/what-is-uml.htm).

“Unified Modeling Language.” *Wikipedia*, Wikimedia Foundation, 12 May 2018, [en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language).

“Package Diagram.” *Wikipedia*, Wikimedia Foundation, 28 Apr. 2018, [en.wikipedia.org/wiki/Package\\_diagram](http://en.wikipedia.org/wiki/Package_diagram).

“Deployment Diagram.” *Wikipedia*, Wikimedia Foundation, 22 Apr. 2018, [en.wikipedia.org/wiki/Deployment\\_diagram](http://en.wikipedia.org/wiki/Deployment_diagram).

“Class Diagram.” *Wikipedia*, Wikimedia Foundation, 17 May 2018, [en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram).

“UML State Machine.” *Wikipedia*, Wikimedia Foundation, 10 Apr. 2018, [en.wikipedia.org/wiki/UML\\_state\\_machine](http://en.wikipedia.org/wiki/UML_state_machine).

“Sequence Diagram.” *Wikipedia*, Wikimedia Foundation, 10 Apr. 2018, [en.wikipedia.org/wiki/Sequence\\_diagram](http://en.wikipedia.org/wiki/Sequence_diagram).

“Entity–Relationship Model.” *Wikipedia*, Wikimedia Foundation, 15 May 2018, [en.wikipedia.org/wiki/Entity%E2%80%93relationship\\_model](http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model).

Sommerville, Ian. *Software Engineering*. Pearson, 2016.

Bootstrap(Wikipedia)< [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) >

PHP < <http://php.net/> >

Github < <https://help.github.com/> >

Mysql <<https://www.oracle.com/mysql/index.html>>