Software Engineering Final Project Team D

**Clothes Shopping Mall using Deep Learning-설계명세서**

아래와 같은 수정사항이 있었습니다.

수정일: 2016년 12월 7일

수정 내용:

1. 기존에 제공하고자 했던 Collection 기능 본 문서에서 삭제.
2. 클릭 수 분석을 통한 고객맞춤형 상품 추천 functional requirements에서 삭제 됨.
3. 위 1, 2번 기능 삭제에 따른 Introduction 수정.
4. 1, 2번 수정에 따른,문서 이음새 수정.
5. Class Diagram에 Image Analysis Class Diagram 추가.
6. Clothes shopping mall 에서 모든 shopping mall solution으로 확장함.

수정본 제출일: 2016년 12월 7일

**Version 7.0.0**

**Shopping MallSolution**

**Using Deep Learning**

**설계명세서**

Software Engineering(ICE3041(41))

D – 임동욱, 신재민,송종훈,장정수

Content

[**1** **Preface** 6](#_Toc468788193)

[1.1 Objectives 6](#_Toc468788194)

[1.2 Readership 6](#_Toc468788195)

[1.3 Document Structure 6](#_Toc468788196)

[A. Preface 6](#_Toc468788197)

[B. Introduction 6](#_Toc468788198)

[C. System Architecture 6](#_Toc468788199)

[D. Class Diagram 6](#_Toc468788200)

[E. Sequence Diagram 6](#_Toc468788201)

[F. State Diagram 6](#_Toc468788202)

[G. Database Design 6](#_Toc468788203)

[H. Testing Plan 7](#_Toc468788204)

[I. Development Environment 7](#_Toc468788205)

[J. Index 7](#_Toc468788206)

[1.4 Version of the Document 7](#_Toc468788207)

[A. Version Format 7](#_Toc468788208)

[B. Version Management Policy 7](#_Toc468788209)

[C. Version Update History 7](#_Toc468788210)

[**2** **Introduction** 8](#_Toc468788211)

[2.1 Objectives 8](#_Toc468788212)

[2.2 Clothes Shopping Mall using Deep Learning 8](#_Toc468788213)

[2.3 Applied Tool 9](#_Toc468788214)

[**3** **System Architecture** 10](#_Toc468788215)

[3.1 Objectives 10](#_Toc468788216)

[3.2 Overview 10](#_Toc468788217)

[3.3 상세 시스템 구조 10](#_Toc468788218)

[3.5 Deployment Diagram 12](#_Toc468788219)

[**4** **Class Diagram** 13](#_Toc468788220)

[4.1 Objectives 13](#_Toc468788221)

[4.2 Overview 13](#_Toc468788222)

[4.3 Sign\_in Diagram 15](#_Toc468788223)

[4.4 Search Diagram 17](#_Toc468788224)

[4.5 Board Diagram 20](#_Toc468788225)

[4.6 ADM Diagram 22](#_Toc468788226)

[4.7 Image Analysis Diagram 24](#_Toc468788227)

[**5** **Sequential Diagram** 24](#_Toc468788228)

[5.1 Objectives 24](#_Toc468788229)

[5.2 Sign\_up Diagram 25](#_Toc468788230)

[5.3 Sign\_in Diagram 25](#_Toc468788231)

[5.4 Search Diagram 26](#_Toc468788232)

[5.5 Cart Diagram 27](#_Toc468788233)

[5.6 Purchasing Cart Diagram 27](#_Toc468788234)

[5.7 Purchasing Item Diagram 28](#_Toc468788235)

[5.8 Purchasing and Payment Diagram 28](#_Toc468788236)

[5.9 Board Diagram 29](#_Toc468788237)

[5.10 Mypage Diagram 30](#_Toc468788238)

[5.11 ADM Diagram 30](#_Toc468788239)

[5.12 Item Add and Correction Diagram 31](#_Toc468788240)

[5.13 User Management Diagram 31](#_Toc468788241)

[5.14 Order Management Diagram 32](#_Toc468788242)

[**6** **State Diagram** 33](#_Toc468788243)

[6.1 Objectives 33](#_Toc468788244)

[6.2 Overall Diagram 33](#_Toc468788245)

[6.3 Sign\_up Diagram 33](#_Toc468788246)

[6.4 Sign\_in Diagram 34](#_Toc468788247)

[6.5 Search Diagram 34](#_Toc468788248)

[6.6 Choose\_Item Diagram 34](#_Toc468788249)

[6.7 Cart Diagram 35](#_Toc468788250)

[6.8 Purchasing Diagram 35](#_Toc468788251)

[6.9 Purchasing Cart Diagram 36](#_Toc468788252)

[6.10 Payment Diagram 36](#_Toc468788253)

[6.11 Shipment Diagram 37](#_Toc468788254)

[6.12 Mypage Diagram 37](#_Toc468788255)

[6.13 Board Diagram 38](#_Toc468788256)

[6.14 ADM Diagram 39](#_Toc468788257)

[A. Overall Diagram 39](#_Toc468788258)

[B. Item Management 39](#_Toc468788259)

[C. User Management Diagram 40](#_Toc468788260)

[**7** **Database Design** 41](#_Toc468788261)

[7.1 User Database 41](#_Toc468788262)

[7.2 Item Database 42](#_Toc468788267)

[7.3 Cart database 42](#_Toc468788271)

[7.4 Order database 44](#_Toc468788284)

[7.5 Board Database 45](#_Toc468788303)

[**8** **Testing Plan** 45](#_Toc468788306)

[8.1 Objectives 45](#_Toc468788307)

[8.2 Testing Policy 45](#_Toc468788308)

[**9** **Development Environment** 49](#_Toc468788309)

[9.1 Objectives 49](#_Toc468788310)

[9.2 Programming Language 49](#_Toc468788311)

[**10** **Index** 52](#_Toc468788312)

[10.1 Objectives 52](#_Toc468788313)

[10.2 Alphabet Index 52](#_Toc468788314)

[10.3 Figure Index 53](#_Toc468788315)

[10.4 Diagram Index 54](#_Toc468788316)

[10.5 Table Index 56](#_Toc468788317)

1. **Preface**
   1. Objectives

Preface에서는 본 문서의 독자를 밝히고,문서의 구조와 파트 별 역할에 대해 소개하며,문서의 작성 및 수정 과정에 대해 서술한다.

* 1. Readership

본 시스템의 개발 및 유지를 하는 시스템 아키텍처(System Architecture)와 소프트웨어 엔지니어(Software Engineer)를 돕고자 작성됨.

* 1. Document Structure
     1. Preface

본 문서의 독자를 밝히고,본 문서가 어떤 형식으로 구성되어 있는지,각 파트가 어떤 역할을 하는지 소개한다.또한,문서의 제작 및 수정 과정에 대해 서술한다.

* + 1. Introduction

목표 시스템에 대한 간단한 소개와 함께,시스템 설계에 사용한 도구를 간략히 소개한다.

* + 1. System Architecture

전체 시스템의 개요와 각 서브시스템(Sub system)에 대한 내용을 다이어그램(Diagram)을 통해 보여주며,시스템과 컴포넌트들, 그리고 그 사이의 관계에 대해 설명한다.

* + 1. Class Diagram

시스템의 object class들과,그 클래스 사이의 관계에 대해 보여준다.

* + 1. Sequence Diagram

각 Actor들과 시스템,그리고 시스템 컴포넌트 사이의 관계와 활동(Interaction)에 대해 보여준다.

* + 1. State Diagram

외부, 내부의 상황(event)에 대해서,시스템이 어떻게 반응하는지 보여준다.

* + 1. Database Design

데이터 관리에 필요한 관리도구,데이터의 저장 방법과 그 규칙들에 대해 설명한다.

* + 1. Testing Plan

시나리오를 바탕으로 목표 시스템이 어떤 방식으로 작동이 되는지 그 과정과 Test case와 Testing policy에 대해 설명한다.

* + 1. Development Environment

목표 시스템 개발환경,언어,도구, 실행 환경을 밝힌다.

* + 1. Index

본 문서에 등장하는 인덱스(Index)를 정리한다.

* 1. Version of the Document
     1. Version Format

버전 번호는 [major.minor.maintenance]로 표현되며,본 문서는 0.1.0부터 시작한다.

* + 1. Version Management Policy

본 문서가 수정될 때 마다, 버전을 업데이트(update)한다.단, 1시간 이내로 수정을 할 경우, 버전 번호를 업데이트 하지 않고,하나의 업데이트로 간주한다.완성된 파트를 수정할 때에는 minor number를 이용하며,새로운 부분을 추가하거나,이 전 버전에 비해 괄목할 만한 변화가 있는 경우 major number을 이용한다.

* + 1. Version Update History

|  |  |  |
| --- | --- | --- |
| Version | Modified Date | Explanation |
| 0.1.0 | 2016.11.10 | Requirement Specification 초안 작성 |
| 0.1.1 | 2016.11.15 | Page Layout Design |
| 1.0.0 | 2016.11.15 | System Architecture 기본 도안 |
| 2.0.0 | 2016.11.16 | System Architecture 상세화 |
| 2.0.1 | 2016.11.16 | 문서 오탈자 수정 |
| 2.1.1 | 2016.11.16 | 문서 제목 수정 |
| 3.0.0 | 2016.11.18 | System Architecture설명 추가 |
| 3.1.0 | 2016.11.20 | 그림(Figure) 추가 |
| 3.1.1 | 2016.11.20 | 그림(Figure) 수정 |
| 4.0.0 | 2016.11.21 | Class diagram 수정 및 추가 |
| 5.0.0 | 2016.11.21 | State diagram 수정 및 추가 |
| 6.0.0 | 2016.11.21 | Sequence diagram 수정 및 추가 |
| 6.1.0 | 2016.11.22 | Index및 목차 정리 |
| 6.1.1 | 2016.11.22 | Page Layout 정리 |
| 6.2.0 | 2016.12.05 | Collection 기능 삭제 및 맞춤형 추천 기능 삭제 |
| 7.0.0 | 2016.12.07 | Clothes Shopping Mall using Deep Learning->Shopping Mall Solution using Deep Learning |

1. **Introduction**
   1. Objectives

목표 시스템에 대한 간단한 소개와 함께,시스템 설계에 사용한 도구를 간략히 소개한다.

* 1. Clothes Shopping Mall using Deep Learning

다른 인터넷 쇼핑몰과의 차별화를 위해,사용자에게 ‘편리함’과 ‘개인화’를 제공하고자 한다.딥러닝(Deep Learning)과 이미지 분석을 이용하여 태그 및 카테고리화를 자동화할 것이며, 구매자에게는 구매 상품의 태그 분석을 통해 관련 상품 추천 기능을 제공할 것이다.

본 문서에서는, ‘의류쇼핑몰’을 예시로,만들고자 하는 시스템에 대해 설명할 것이다.

* 1. Applied Tool

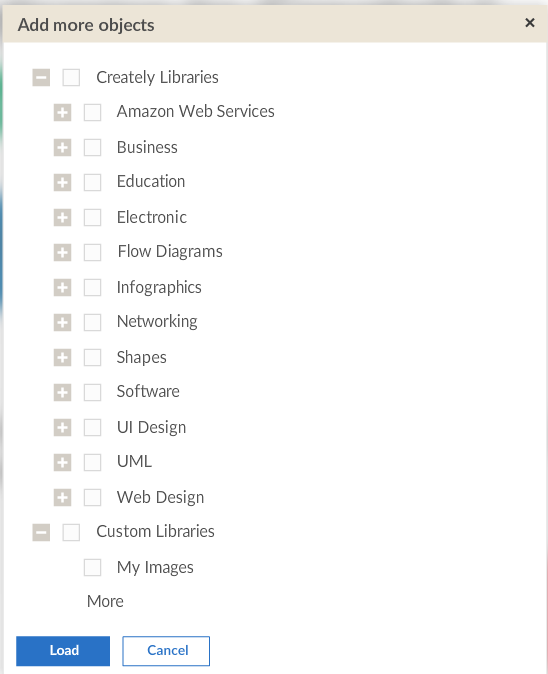
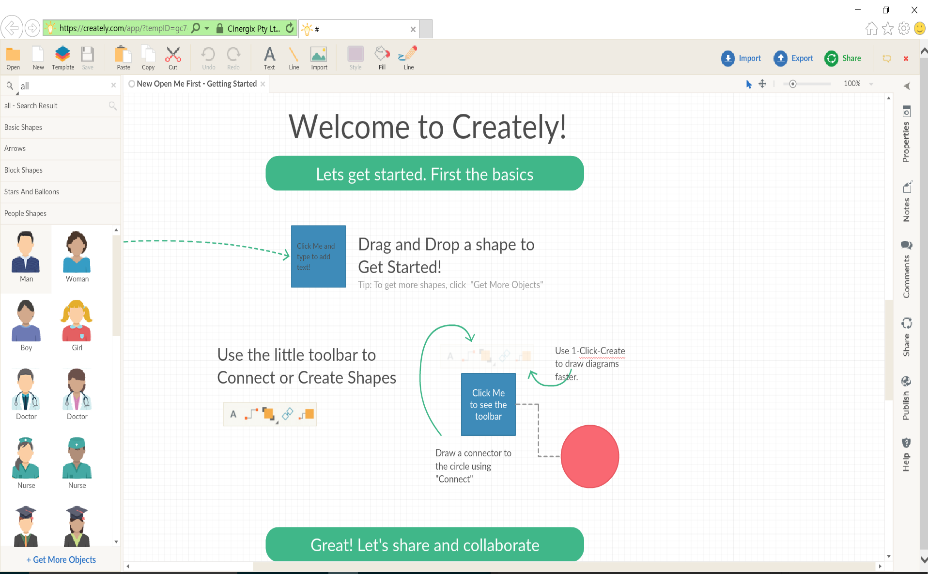
본 문서(설계명세서)와 요구사항 명세서의 그림(Diagram)은 Creately사에서 제공하는 “Online Diagram Demo Version”을 이용해 그렸다. “Creately는 UML Diagram뿐만 아니라,각종 Diagram와 Flow chart의 기본 틀과 모양을 제시해줌으로써,사용자가 손쉽게 Diagram을 그릴 수 있도록 도와준다.만든 Diagram은 Export기능을 통해,컴퓨터 혹은 클라우드(Cloud)에 저장할 수 있다. 간단한 가입절차를 통해 무료 Demo version을 사용할 수 있다.

Figure Creately.com

Figure Creatly objects

1. **System Architecture**
   1. Objectives

전체 시스템의 개요와 각 서브시스템(Sub system)에 대한 내용을 다이어그램(Diagram)을 통해 보여주며,시스템과 컴포넌트들, 그리고 그 사이의 관계에 대해 설명한다.

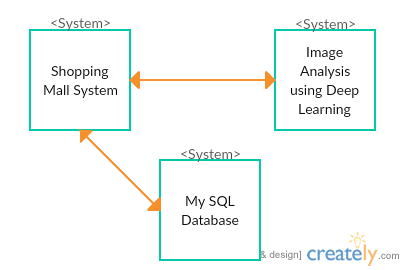
* 1. Overview

Diagram Overview System Architecture

* 1. 상세 시스템 구조

목표 시스템은 크게 Item 판매자가 접근하는 Shopping Mall Management level과 구매자가 접근하는 Shopping Mall level이 있다.

Shopping Mall Management level에서, 판매자는 고객관리, 아이템 관리, 주문 내역 관리를 할 수 있으며, 이러한 관리를 위해서는 Admin 관리자 권한을 가지고 있는지 확인이 필요하다. Admin 권한의 확인 여부는 Login ID와 데이터베이스 내의 관리자 ID의 일치 여부를 통해 확인한다. 또한, 각종 관리를 위한 고객 정보와 아이템 정보, 주문 정보에 관한 것은 관리자 권한으로 데이터베이스에서 가져오도록 한다.

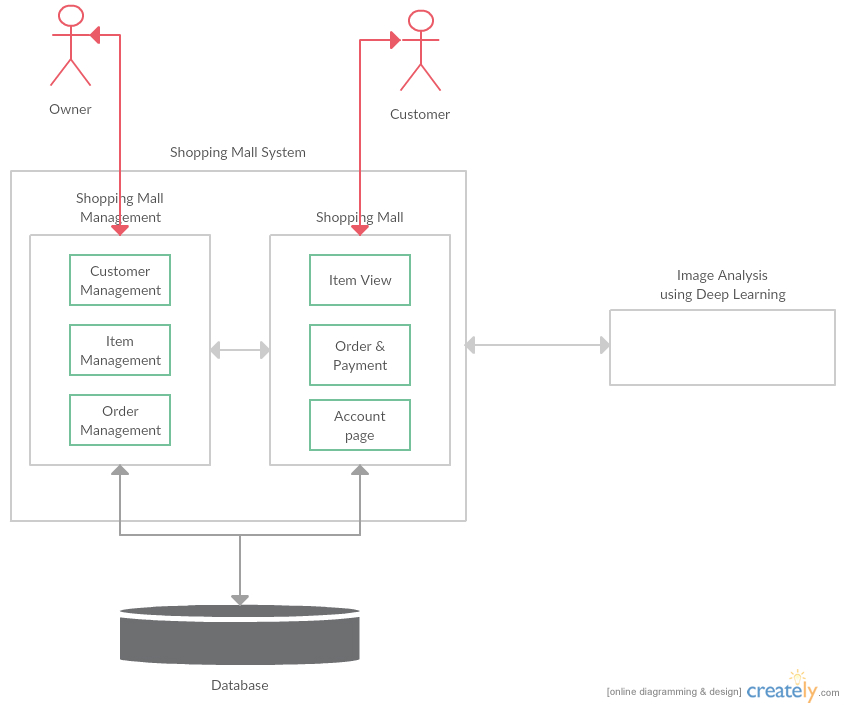
Shopping Mall level에서, 고객은 아이템을 보고, 주문을 하고, 본인의 계정을 관리할 수 있다. 본인의 계정관리 시에는 본인 권한이 필요하며, 이는 Login ID와 데이터베이스 내의 ID와의 일치 여부를 통해 확인한다. 아이템에 대한 정보와 계정에 대한 정보는 데이터베이스에서 가져오며, 새로운 주문 내역은 직접 작성하고, 이를 데이터베이스에 저장하기로 한다.

Diagram System Architecture

* 1. Deployment Diagram

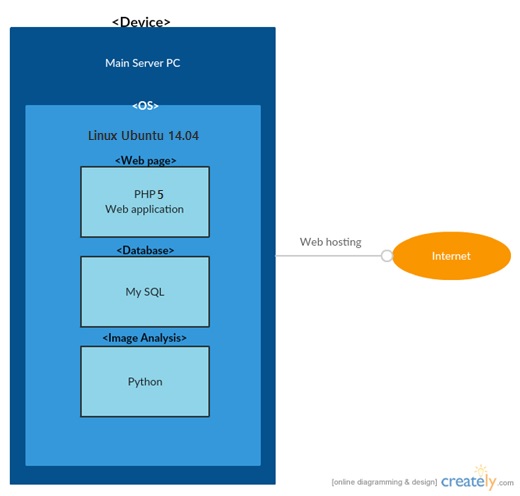


Diagram Deployment Diagram

1. **Class Diagram**
   1. Objectives

시스템의 object class들과,그 클래스 사이의 관계에 대해 보여준다.

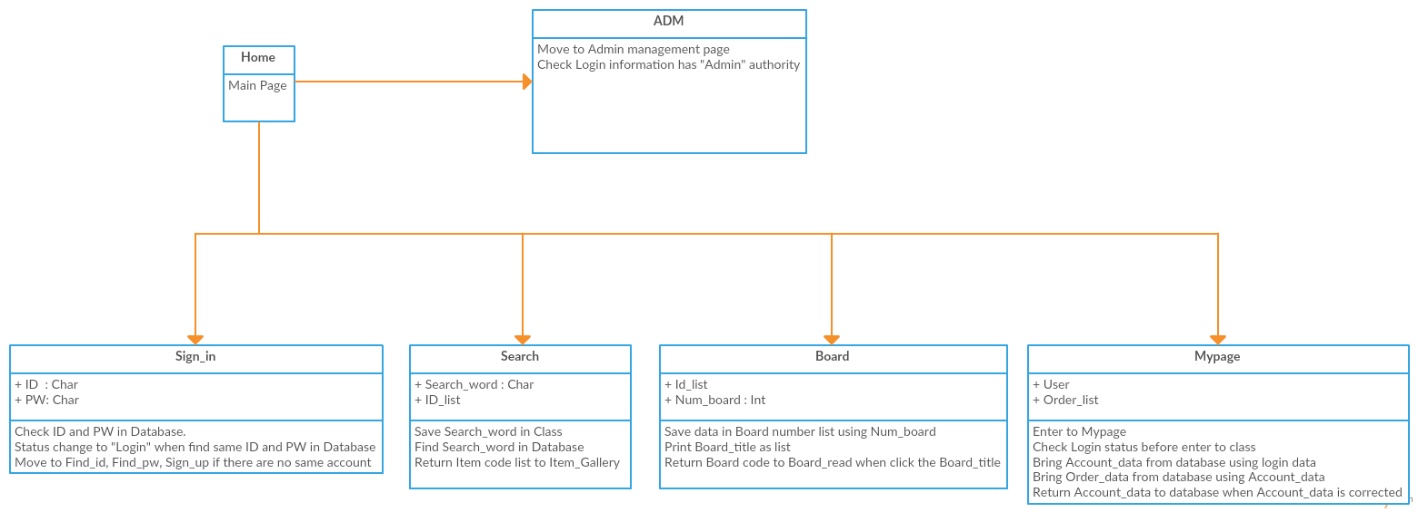
* 1. Overview

Diagram System Overview Class Diagram

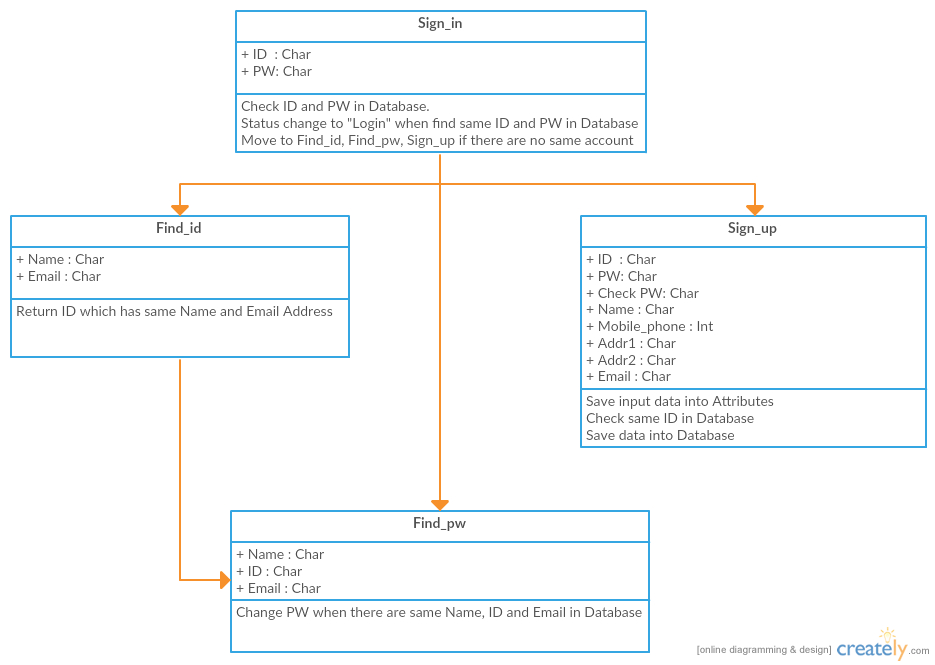
* + 1. Home : 쇼핑몰을 켰을 때 맨 처음 보이는 메인 페이지
    2. Sign\_in
       1. Attributes
          1. ID-입력 받은 아이디
          2. PW-입력 받은 비밀번호
       2. Description
          1. 입력 받은 아이디, 비밀번호와 일치하는 계정이 있는지 데이터베이스에서 확인 후, 일치 하는 것이 있으면 Login 상태로 변경.
          2. Find\_id, Find\_pw, Sign\_up 페이지로 이동 가능
    3. Search
       1. Attributes
          1. Search\_word: 입력 받은 검색 단어
          2. Id\_list: 검색된 상품 코드 리스트
       2. Description
          1. 입력받은 검색문을 클래스에 저장
          2. 저장된 search\_word를 통해 데이터베이스에서 상품 검색 실행
          3. 검색된 상품 코드 리스트를 Item\_gallery에 넘겨줌
    4. Board
       1. Attributes
          1. Id\_list: 게시글 코드 번호 리스트
          2. Num\_board: 게시글 목차 번호
       2. Description
          1. 게시글 목차 번호를 이용하여 최신부터 일부 데이터를 가져와 게시글 번호 리스트에 저장
          2. 게시글 제목을 리스트 형태로 출력
    5. Mypage
       1. Attributes
          1. User: 계정 정보
          2. Order\_list: 주문 정보 리스트
       2. Description
          1. 계정 Mypage로 들어가는 클래스
          2. 클래스로 들어가기 전 로그인 상태 확인
          3. 현 로그인 정보로 데이터베이스에서 해당 계정 데이터를 받아옴.
          4. 해당 계정 데이터에 부합하는 주문 정보들을 데이터베이스에서 가져옴.
          5. 계정 정보 수정 시 해당 수정 내용을 데이터베이스에 반환.
    6. ADM: admin 관리 페이지로 이동하는 클래스로 해당 로그인 계정이 admin인지 아닌지 확인.
  1. Sign\_in Diagram

Diagram Sign\_in Class Diagram

* + 1. Sign\_in
       1. Attributes

ID: 입력 받은 ID

PW: 입력 받은 비밀번호

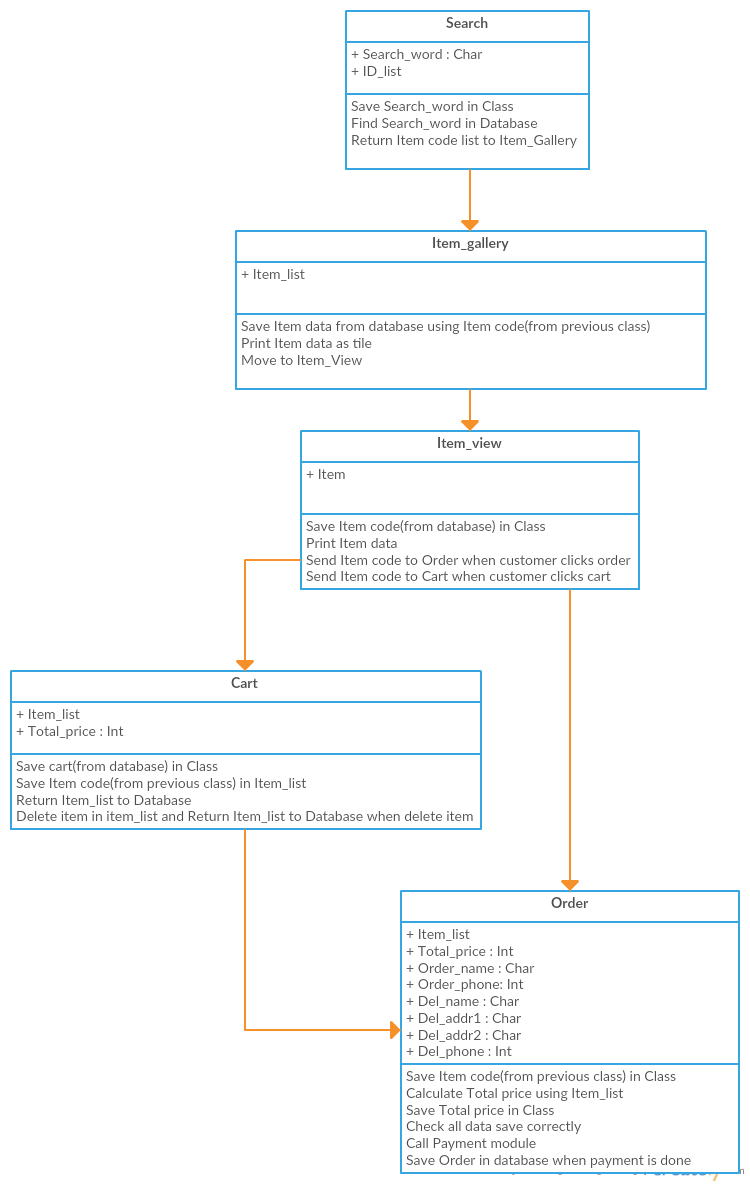
* + - 1. Description
         1. 입력받은 아이디, 비밀번호와 일치하는 계정이 있는지 데이터베이스에서 확인 후 일치 하는 것이 있으면 로그인 상태로 변경
         2. Find\_id, Find\_pw, Sign\_up 페이지로 이동 가능
    1. Sign\_up
       1. Attributes
          1. ID: 아이디
          2. PW: 비밀번호, Check PW: 비밀번호 확인
          3. Name: 이름, Mobile\_phone: 휴대폰 번호, Phone: 일반 전화
          4. Addr1: 주소, Addr2: 상세주소, Email: 이메일 주소
       2. Description
          1. 회원가입 데이터를 받아서 각 Attribute에 저장
          2. 중복되는 ID가 있는지 데이터베이스에서 확인
          3. 데이터에 하자가 없으면(누락된 데이터 없음.중복되는 정보 없음) 이를 데이터베이스에 입력
    2. Find\_id
       1. Attributes
          1. Name: 계정 주 이름, Email: 이메일 주소
       2. Description
          1. 계정 주의 이름과 이메일 주소를 받아서 일치하는 아이디를 반환
    3. Find\_pw
       1. Attributes
          1. Name: 계정 주 이름, ID: 아이디, Email: 이메일 주소
       2. Description
          1. 계정 주의 이름과 아이디,이메일 주소를 받아서 모두 일치하는 계정이 있으면 패스워드 변경 창을 띄워줌.
  1. Search Diagram

Diagram Search Class Diagram

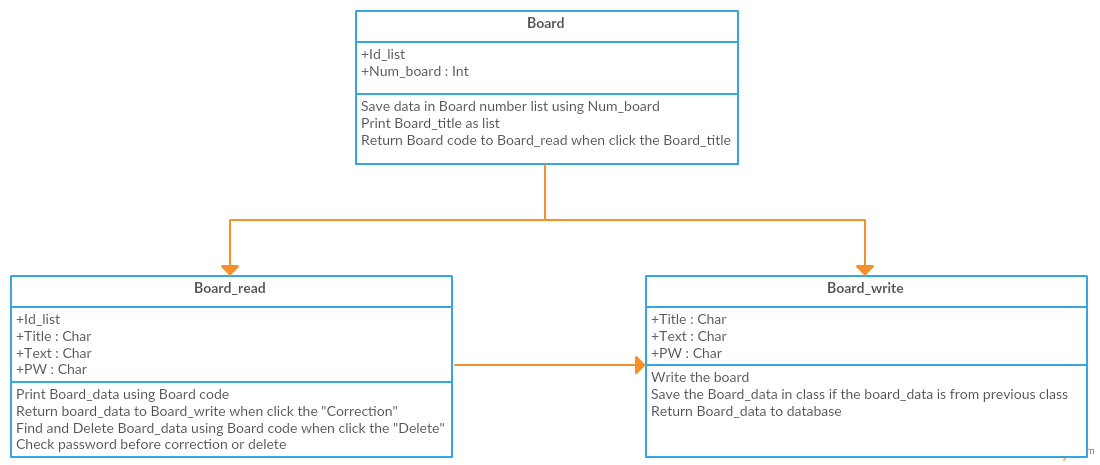
* + 1. Search
       1. Attributes
          1. Search\_word: 검색하고자 하는 단어
          2. ID\_list: 검색된 상품 코드 리스트
       2. Description
          1. 입력받은 검색문을 클래스에 저장
          2. 저장된 search\_word를 통해 데이터베이스 내에서 상품 검색 실행
          3. 검색된 상품 코드 리스트를 Item\_gallery에 넘겨줌
    2. Item\_Gallery
       1. Attributes
          1. Item\_list: 표시할 상품 코드 리스트
       2. Description
          1. 이전 클래스에서 넘겨 받은 상품 코드 리스트를 이용해 데이터베이스에서 상품 데이터를 가져와 클래스에 저장.
          2. 해당 상품 데이터를 바둑판식으로 출력
          3. Item\_view로 이동
    3. Item\_view
       1. Attributes
          1. Item:상품 관련 데이터
       2. Description
          1. 이전 페이지에서 관련 상품 코드를 가져와 데이터베이스에서 찾아 클래스에 저장
          2. 저장된 상품 데이터를 화면에 출력
          3. 상품 주문 시 해당 상품 코드를 order에 전달
          4. 장바구니에 넣기 시 해당 상품 코드를 cart에 전달
    4. Cart
       1. Attributes
          1. Item\_list: 장바구니 상품 코드 리스트
          2. Total\_price: 총 가격 합계
       2. Description
          1. 해당 계정의 기존 장바구니 리스트를 데이터베이스에서 가져와 저장
          2. 이전 클래스에서 가져온 상품 코드를 item\_list에 저장 후 데이터베이스에 반환
          3. 장바구니 목록에서 상품 삭제 시 item\_list에서 해당 데이터를 삭제하고 이를 데이터베이스에 반환
    5. Order
       1. Attributes
          1. Item\_list: 주문 상품 정보 리스트
          2. Total Price: 최종가격
          3. Order\_name: 주문자 성함, Order\_phone: 주문자 전화번호
          4. Del\_name: 수령자 이름
          5. Del\_addr1: 배송지 주소, Del\_addr2: 배송지 상세주소
          6. Del\_phone: 수령자 전화번호
       2. Description
          1. 주문 상품들 코드 리스트를 이전 클래스에서 받아서 해당 클래스에 저장
          2. Item\_list를 이용해 최종 가격을 산출해 Total\_price에 저장
          3. 주문자 데이터,수령자 데이터를 입력 받아 클래스에 저장
          4. 해당 데이터들이 전부 맞게 입력되었는지 확인
          5. 데이터가 확인되면 외부 주문 결제 모듈을 호출
          6. 결제가 완료되면 주문 정보를 데이터베이스에 저장
  1. Board Diagram

Diagram Board Class Diagram

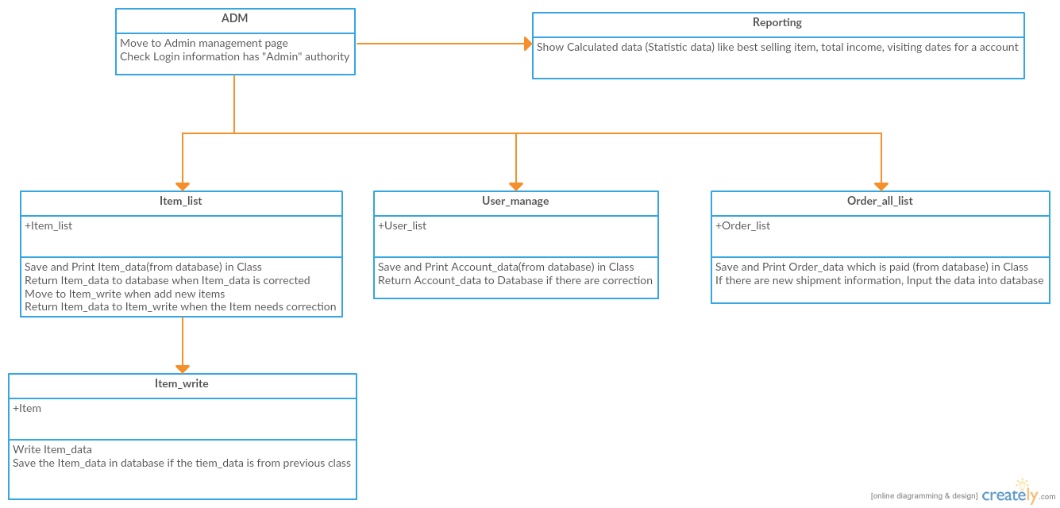
* + 1. Board
       1. Attributes
          1. ID\_list: 게시글 코드 번호 리스트
          2. Num\_board: 게시글 목차 번호
       2. Description
          1. 게시글 목차 번호를 이용하여 최신부터 일부 데이터를 가져와 게시글 번호 리스트에 저장
          2. 게시글 제목을 리스트 형태로 출력
    2. Board\_read
       1. Attributes
          1. ID: 게시글 코드
          2. Title: 게시글 제목, Text: 게시글 내용
          3. PW: 게시글 비밀번호
       2. Description
          1. 넘겨받은 게시글 코드번호를 이용해 해당 게시글 데이터를 데이터베이스로 가져와 이를 출력
          2. 수정 시 해당 게시글 데이터를 Board\_write에 넘겨줌
          3. 삭제 시 해당 게시글 코드를 이용해 데이터베이스에서 색인 후 게시글 데이터 삭제
          4. 게시글 수정 혹은 삭제 전,게시글 비밀번호와 입력 받은 비밀번호의 일치 여부 확인
    3. Board\_write
       1. Attributes
          1. Title: 게시글 제목, Text: 게시글 내용
          2. PW: 게시글 비밀번호
       2. Description
          1. 게시글 작성 클래스
          2. 이전 클래스에서 게시글 데이터를 받아 왔다면,해당 게시글 데이터를 클래스에 저장.
          3. 작성 완료 시 해당 게시글 데이터를 데이터베이스에 반환.
  1. ADM Diagram

Diagram ADM Class Diagram

* + 1. ADM: admin관리 페이지로 이동하는 클래스,해당 로그인 계정이 admin인지 아닌지 확인함.
    2. Item\_list
       1. Attributes
          1. Item\_list: 상품 데이터 리스트
       2. Description
          1. 상품 데이터들을 데이터베이스에서 가져와 클래스에 저장 및 출력
          2. 특정 상품 데이터의 간단한 수정 혹은 삭제 시,이를 데이터베이스에 반환
          3. 새 상품 등록 시, Item\_write 로 이동함
          4. 특성 상품 수정 시, Item\_write에 해당 상품 데이터를 넘겨줌
    3. User\_manage
       1. Attributes
          1. User\_list: 회원 계정 데이터 리스트
       2. Description
          1. 회원 계정 데이터를 데이터베이스에서 가져와 클래스에 저장 및 출력
          2. 계정 수정,삭제 시 이를 데이터베이스에 반환
    4. Order\_all\_list
       1. Attributes
          1. Order\_list: 주문 리스트
       2. Description
          1. 결제 완료되어 저장된 주문들을 데이터베이스에서 가져와 클래스에 저장 및 출력
          2. 배송정보를 주문에 입력 혹은 수정 시 이를 데이터베이스에 반영
    5. Reporting: 회원의 방문, item 판매,매출 등의 통계자료를 보여줌.
  1. Image Analysis Diagram

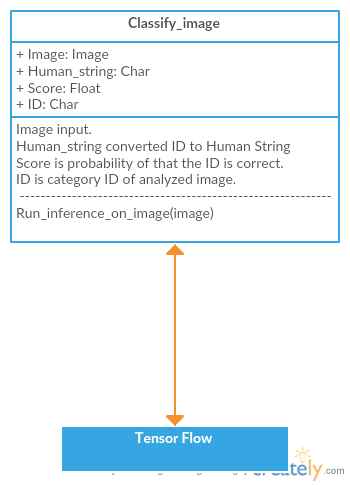


Diagram Image Analysis Class Diagram

* + 1. Classify\_image
       1. Attributes
          1. Image: 입력 받은 이미지 파일
          2. Human\_string: 해당 ID를 사람의 언어로 변환한 값
          3. Score: 해당 ID일 확률
          4. ID: 분석한 이미지의 카테고리 ID

1. **Sequential Diagram**
   1. Objectives

각 Actor들과 시스템,그리고 시스템 컴포넌트 사이의 관계와 활동(Interaction)에 대해 보여준다.

* 1. Sign\_up Diagram

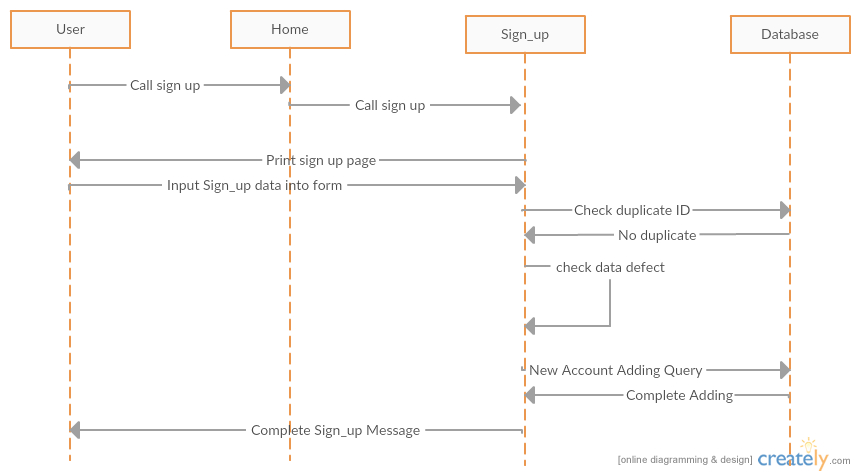


Diagram Sign\_up Sequence Diagram

* 1. Sign\_in Diagram

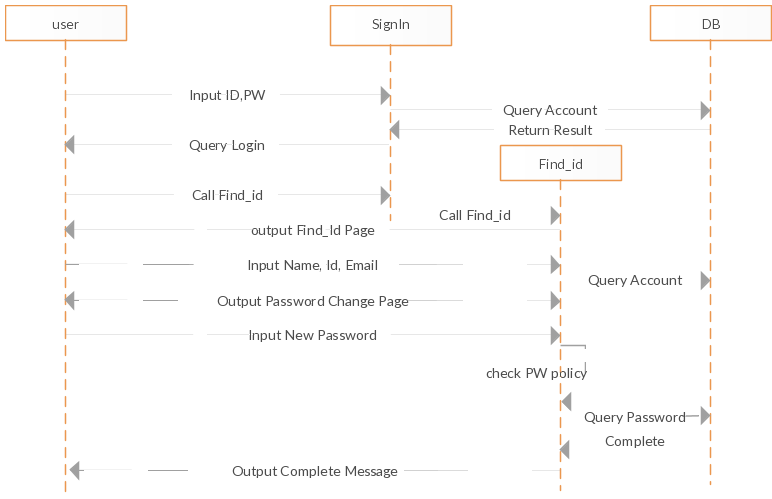


Diagram Sign\_in Sequence Diagram

* 1. Search Diagram

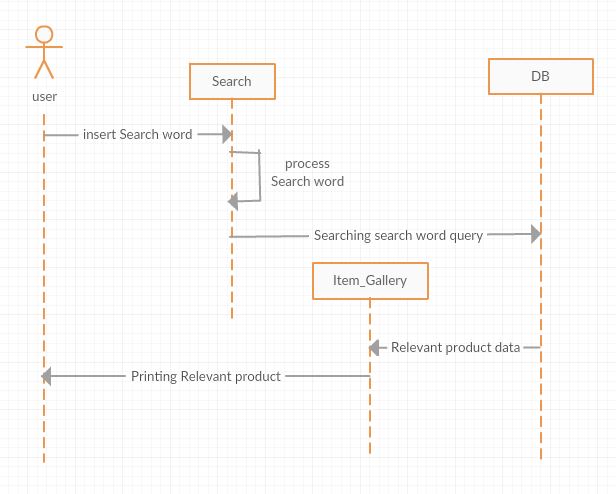


Diagram Search Sequence Diagram

* 1. Cart Diagram

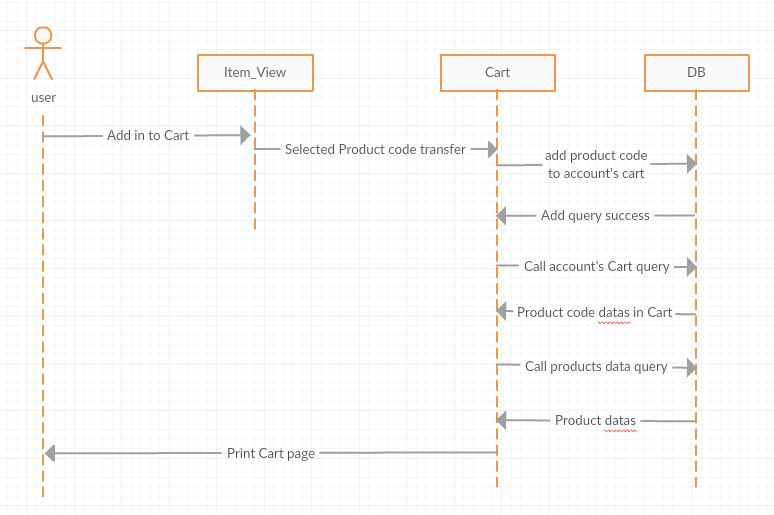


Diagram Cart Sequence Diagram

* 1. Purchasing Cart Diagram

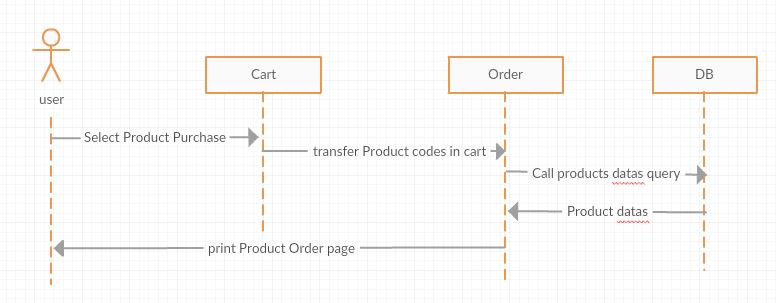


Diagram Purchasing Cart Sequence Diagram

* 1. Purchasing Item Diagram

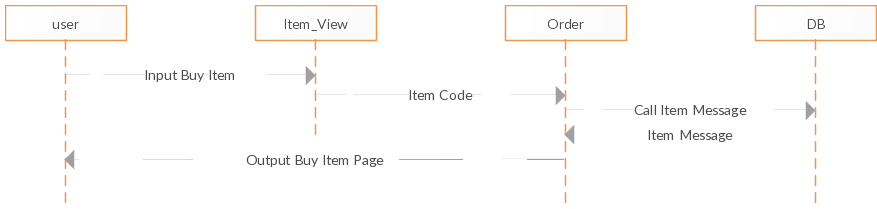


Diagram Purchasing Item Sequence Diagram

* 1. Purchasing and Payment Diagram

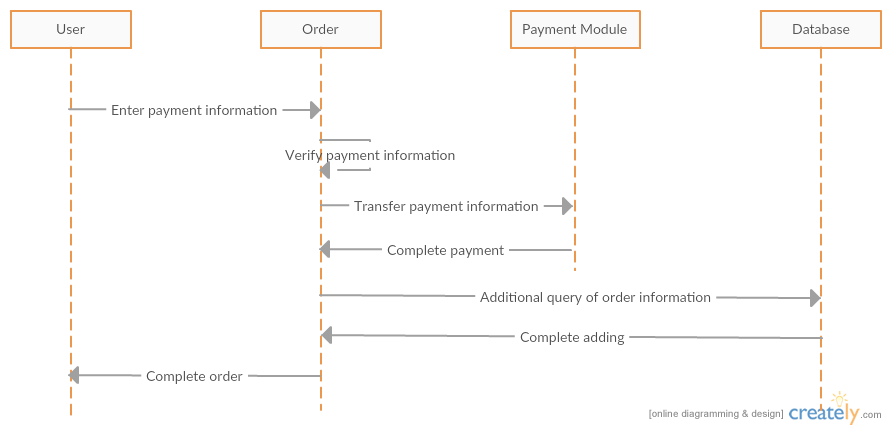


Diagram Purchasing and Payment Sequence Diagram

* 1. Board Diagram

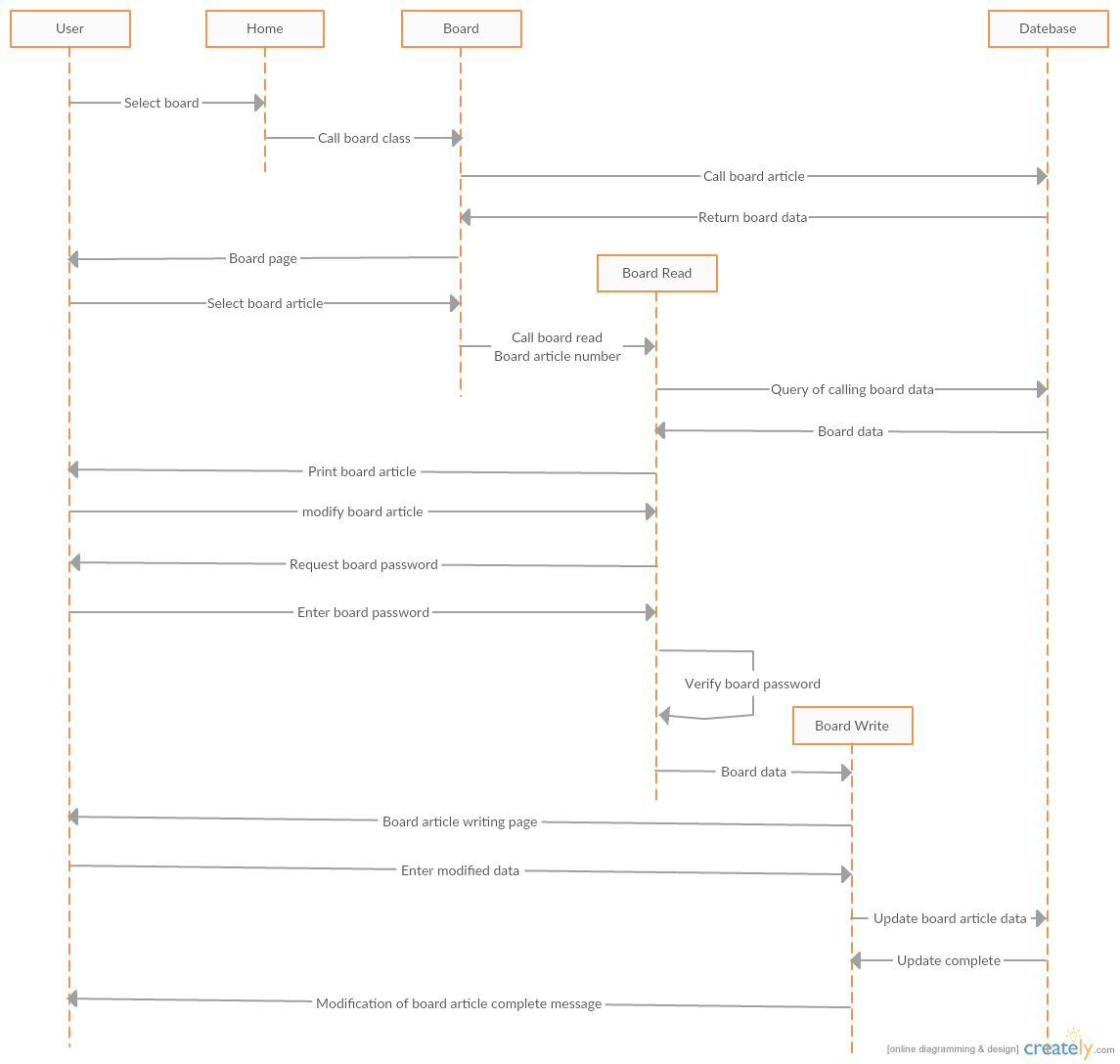


Diagram Board Sequence Diagram

* 1. Mypage Diagram

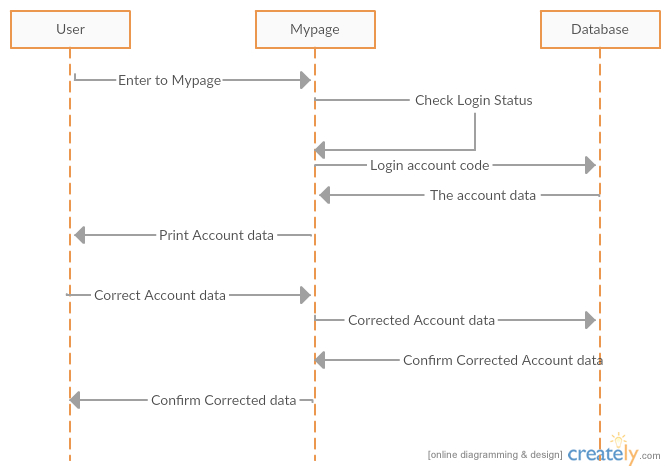


Diagram Mypage Sequence Diagram

* 1. ADM Diagram

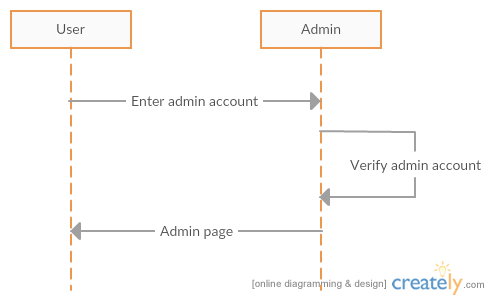


Diagram ADM Sequence Diagram

* 1. Item Add and Correction Diagram

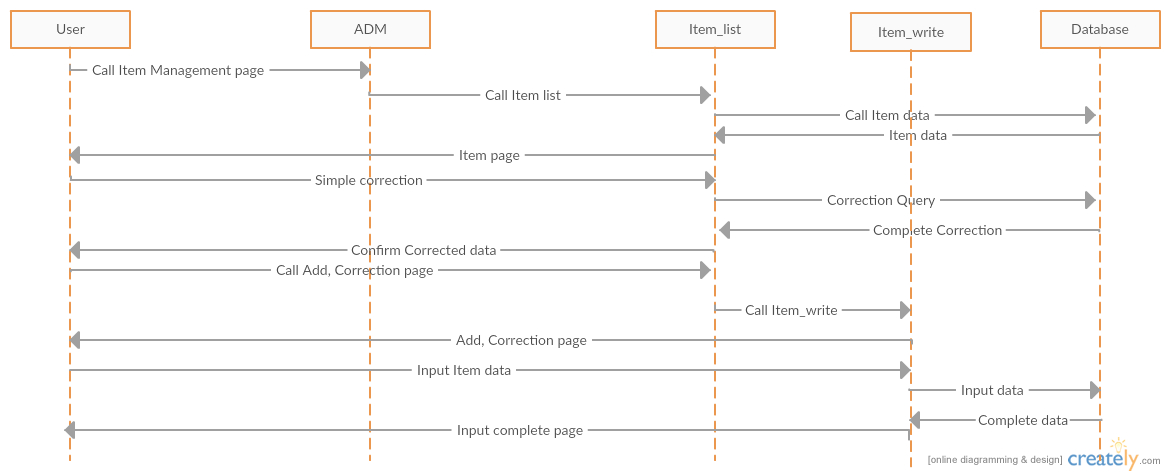


Diagram Item Add and Correction Sequence Diagram

* 1. User Management Diagram

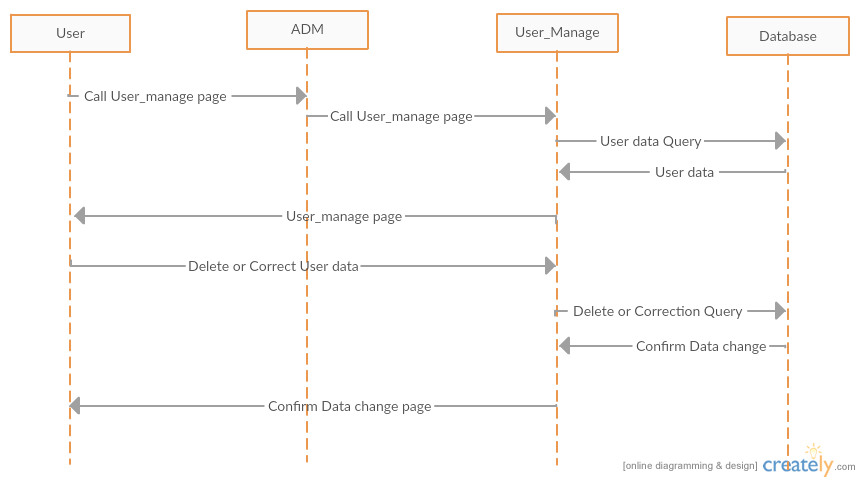


Diagram User Management Sequence Diagram

* 1. Order Management Diagram

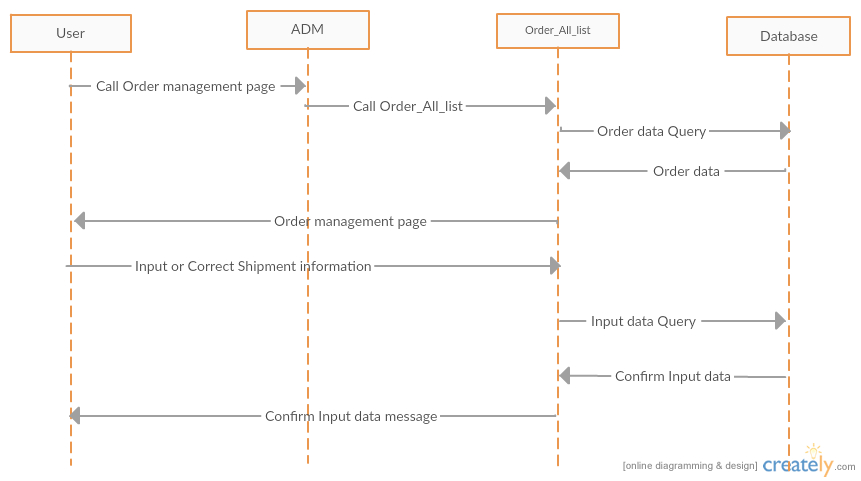


Diagram Order Management Sequence Diagram

1. **State Diagram**
   1. Objectives

외부,내부 상황(event)에 따른 시스템의 반응을 보여준다.

* 1. Overall Diagram

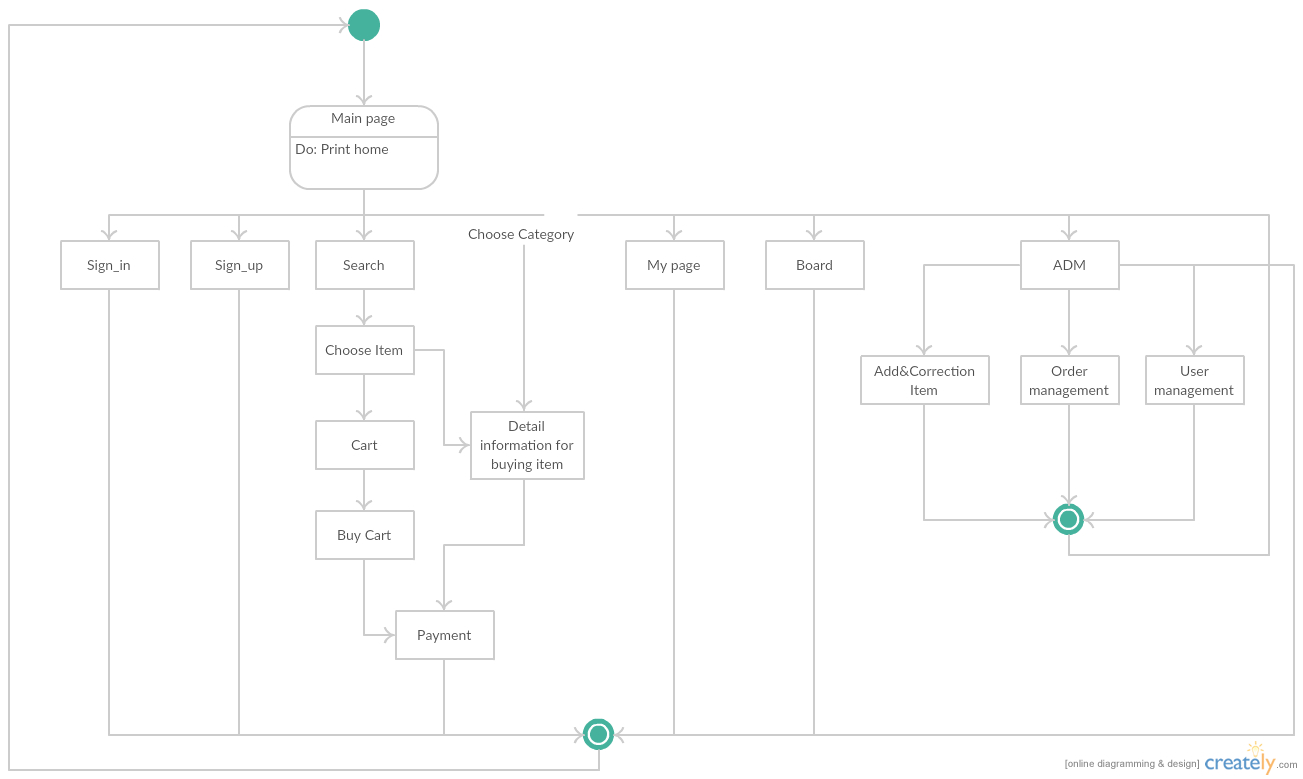


Diagram Overall State Diagram

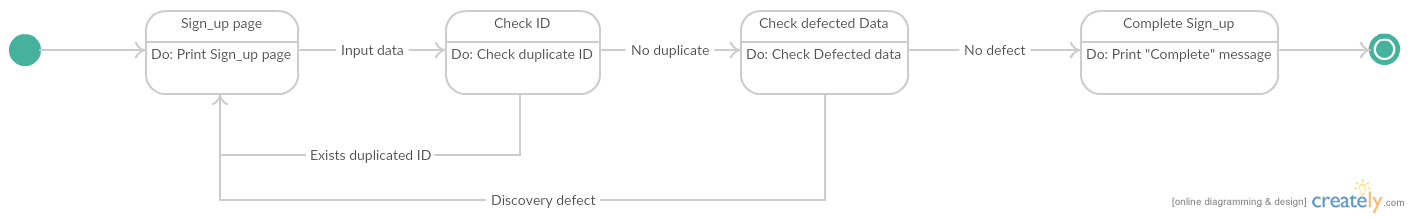
* 1. Sign\_up Diagram

Diagram Sign\_up State Diagram

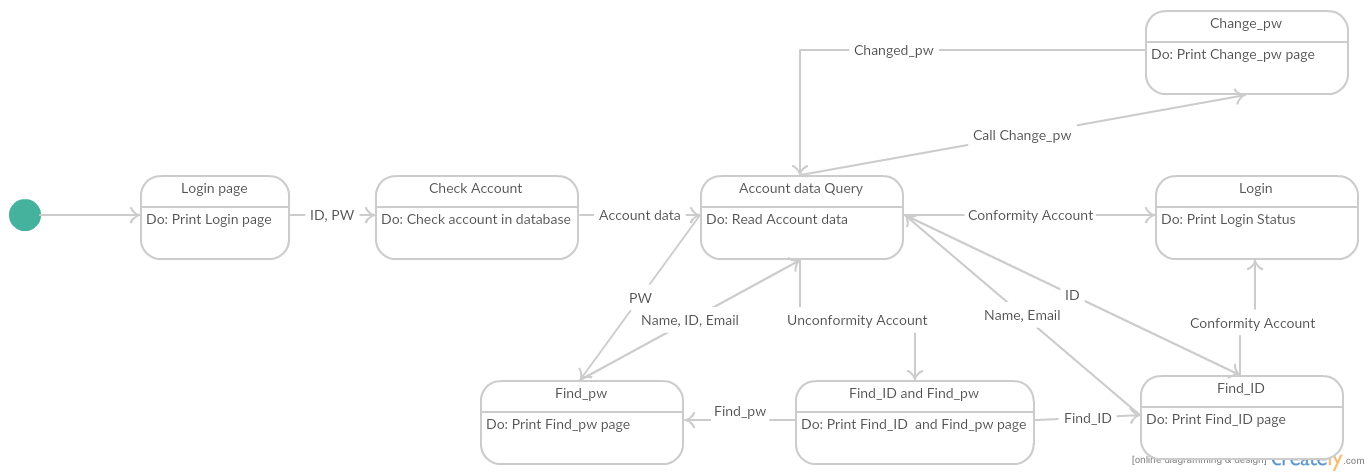
* 1. Sign\_in Diagram

Diagram Sign\_in State Diagram

* 1. Search Diagram

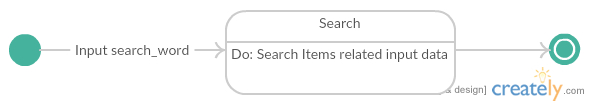


Diagram Search State Diagram

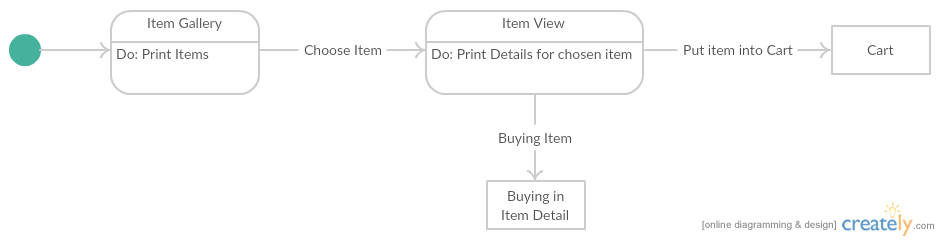
* 1. Choose\_Item Diagram

Diagram Choose\_item State Diagram

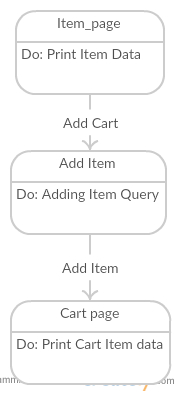
* 1. Cart Diagram

Diagram Cart State Diagram

* 1. Purchasing Diagram



Diagram Purchasing State Diagram

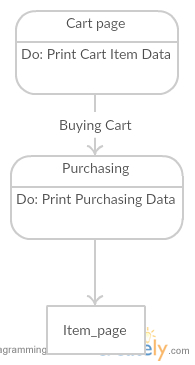
* 1. Purchasing Cart Diagram

Diagram Purchasing Cart State Diagram

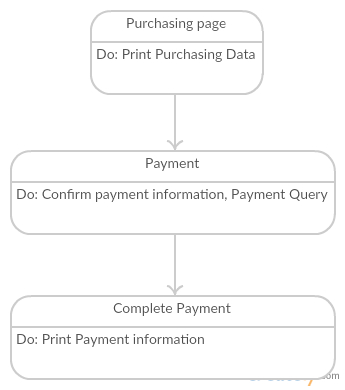
* 1. Payment Diagram

Diagram Payment State Diagram

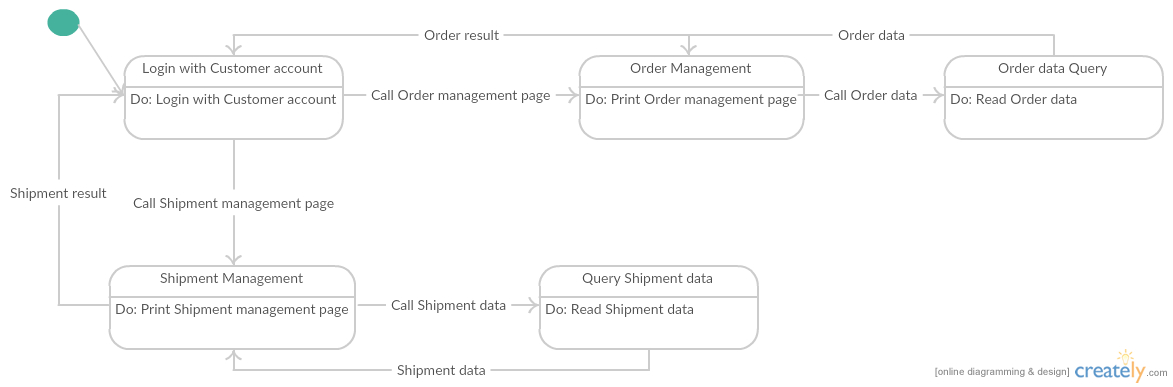
* 1. Shipment Diagram

Diagram Shipment State Diagram

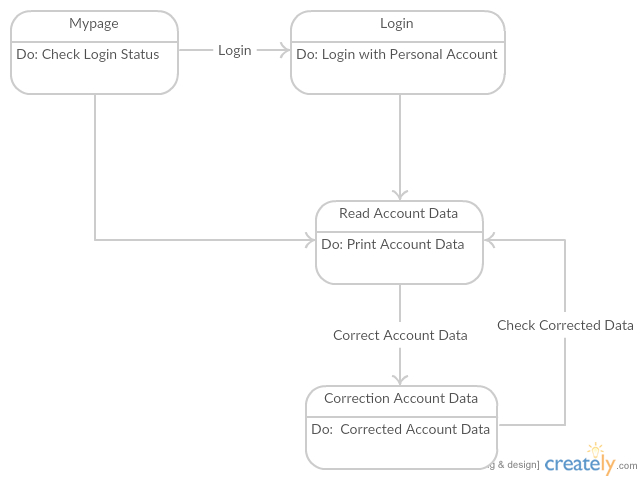
* 1. Mypage Diagram

Diagram Mypage State Diagram

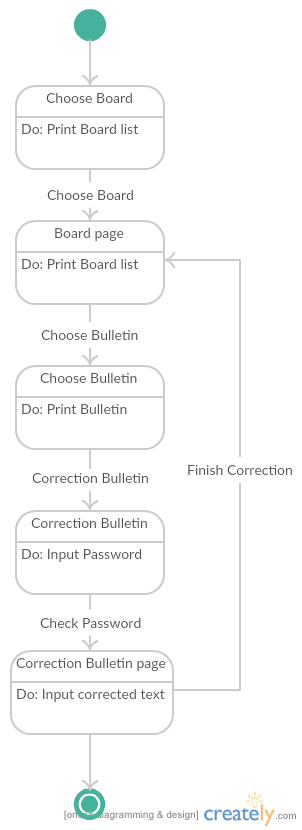
* 1. Board Diagram

Diagram Board State Diagram

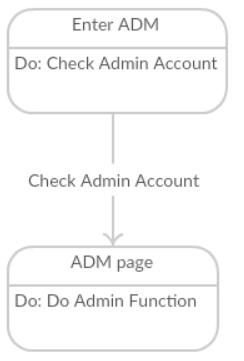
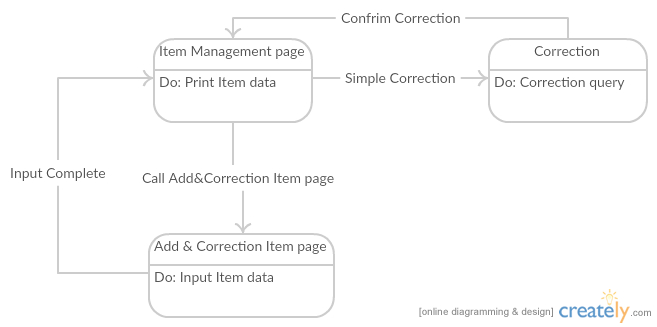
* 1. ADM Diagram
     1. Overall Diagram
     2. Item Management

Diagram Item Management State Diagram

Diagram ADM State Diagram

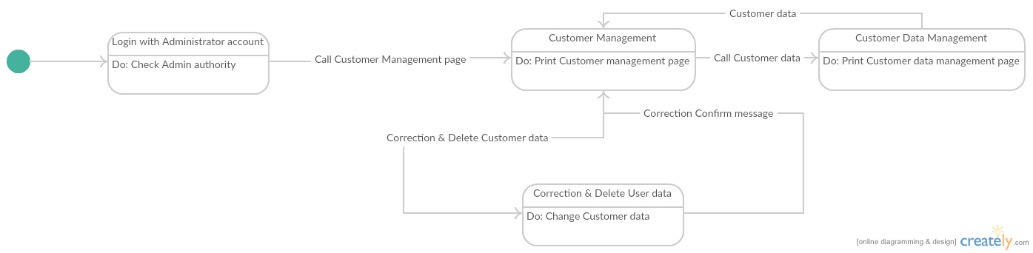
* + 1. User Management Diagram

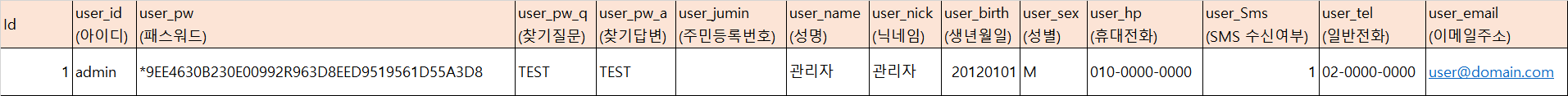
Diagram User Management State Diagram

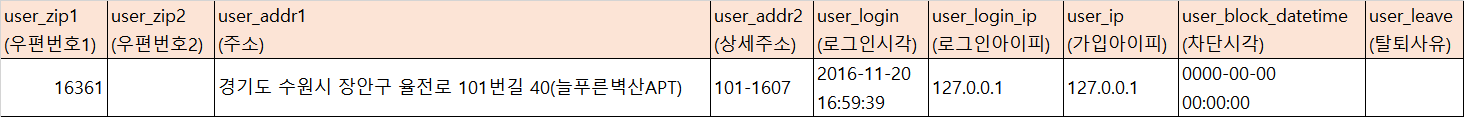
1. **Database Design**
   1. User Database

Table User Database



Table User Database2

Table 3 User Database3



* + 1. 필드:각 필드의 이름에 맞는 데이터를 입력한다.
    2. 종류:각 필드에 입력하는 Input Type을 의미한다.
    3. 예시: 위의 입력 예시(Table2-User Database2, Table3-User Database3)처럼 구성되며, 이때 각 필드가 의미하는 것은 예시의 괄호 안에 한글로 설명이 되어있다.
  1. Item Database

Table Item Database

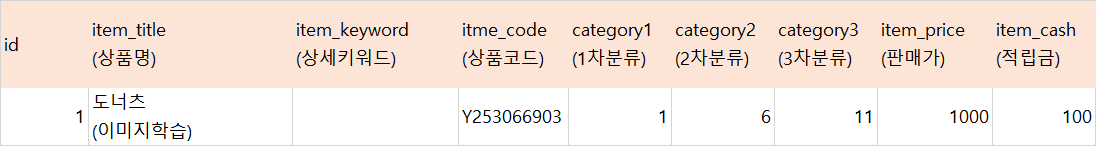
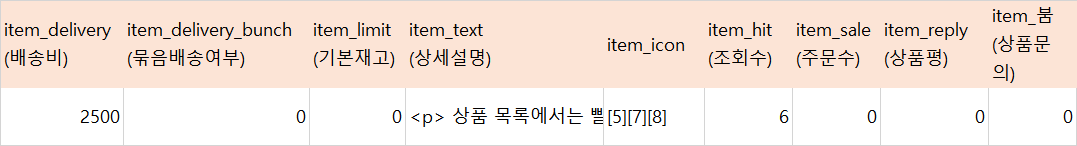
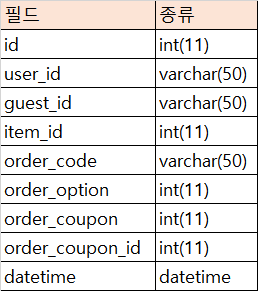


Table Item Database2



* + 1. 예시:상품을 등록할 때,위의 입력 예시(Table4-Item Database, Table5-Item Database2)처럼 구성되며,각 필드가 의미하는 내용은 괄호 안에 한글로 설명이 되어있다.
  1. Cart database

Table Cart Database

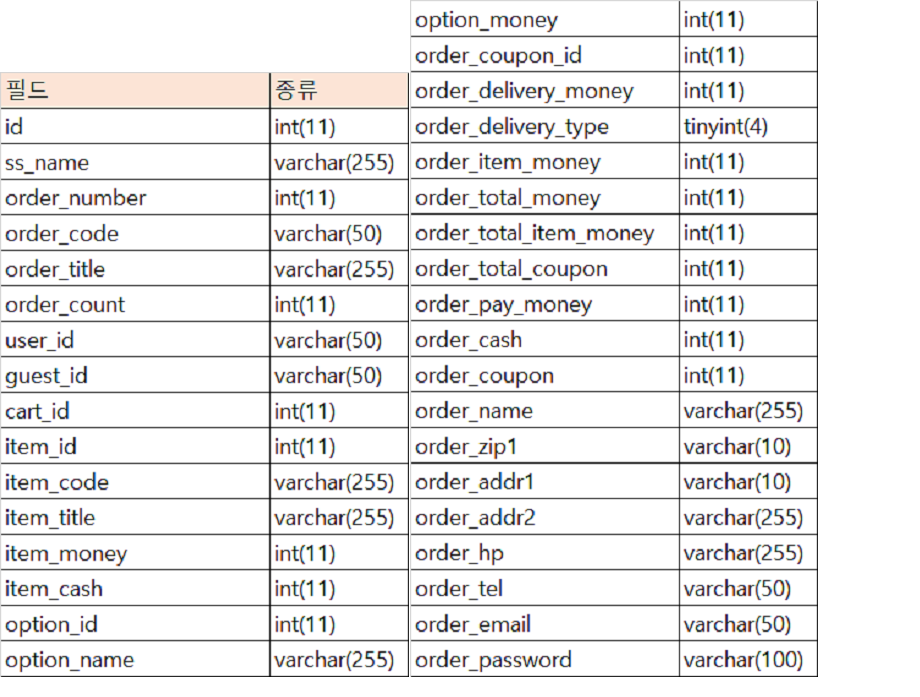


* + 1. 필드:각 필드의 이름에 맞는 데이터를 입력한다.
       1. ID: 장바구니 ID
       2. User\_Id: 로그인 된 구매자(회원)의 ID
       3. Guest\_Id: 로그인 하지 않은 구매자(방문자)의 ID
       4. Item\_Id: 장바구니에 담긴 상품정보 ID
       5. Order\_code: 주문코드
       6. Order\_option: 추가 주문 옵션
       7. Order\_coupon: 주문 시 적용 쿠폰 이름
       8. Order\_coupon\_id: 적용 쿠폰 ID
       9. Datetime: Cart마지막 수정일
    2. 종류:각 필드에 입력하는 Input Type을 의미한다.

D

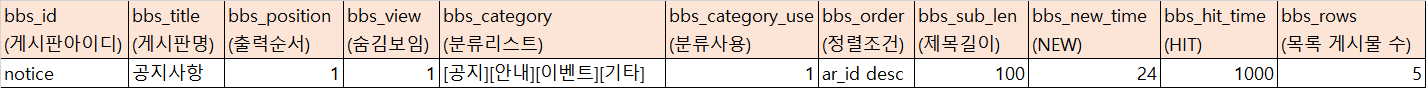
* 1. Order database

Table Order Database



* + 1. 필드:각 필드의 이름에 맞는 데이터를 입력한다.각 필드에 대한 설명은 아래의 리스트를 참고하도록 한다.단,위의 Cart, Item, User의 항목과 겹치는 부분은 따로 설명하지 않았다.
       1. ID: 주문 ID
       2. Ss\_name: 주문자 이름
       3. Order\_number: 주문번호
       4. Order\_code: 주문내역 코드
       5. Order\_title: 주문 이름
       6. Order\_count: 주문 개수
       7. Item\_money: 각 상품에서 사용한 적립금액
       8. Item\_cash: 각 상품 당 현금 혹은 카드 결제 금액
       9. Option:주문자가 회원이 아닐 경우를 의미함
       10. Order\_item\_money: 한 주문에서 사용한 적립금액
       11. Order\_total\_money: 지불해야할 총 금액
       12. Order\_total\_item\_money: 주문에서 사용한 총 적립금액
       13. Order\_total\_coupon: 쿠폰을 사용해 지불한 금액
       14. Order\_pay\_money: 최종 지불 금액
       15. Order\_cash: 한 주문에서 사용한 현금 혹은 카드 결제 금액
       16. Order\_coupon: 한 주문에서 사용한 쿠폰
    2. 종류:각 필드에 입력하는 Input Type을 의미한다.
  1. Board Database

Table Board Database



* + 1. 예시:상품을 등록할 때,위의 입력 예시(Table8-Board Database)처럼 구성되며,각 필드가 의미하는 내용은 괄호 안에 한글로 설명이 되어있다.

1. **Testing Plan**
   1. Objectives

시나리오를 바탕으로,목표 시스템이 어떤 방식으로 작동이 되는지 그 과정과 Test case, 그리고 Testing policy에 대해 설명한다.

* 1. Testing Policy

목표 시스템을 개발 도중에 중간중간 test를 진행한다.이를 Development testing이라 하며, 각 컴포넌트 별로 unit test를 진행하고,각 컴포넌트가 결합 되었을 때(Subsystem이 됨) Integration test를 진행한다.그리고, system이 완성되었을 때 System testing를 진행한다.Test를 하면서 기능적인 부분뿐만 아니라,비기능적인 부분(Emergent Properties)또한 테스트 해야만 한다.목표 시스템은 개인 정보를 다루고 있기 때문에 “보안”에 중점을 두고 테스트할 수 있도록 한다.

Development testing과 System testing이 완료되면,실 사용자를 대상으로 Acceptance testing을 진행한다.

* 1. Test Case
     1. 판매자Test Case
        1. 로그인
           1. 사용자: ID, PW로 로그인을 시도함.
           2. 시스템 동작:데이터베이스 내에서 일치하는 ID, PW가 있는지 확인함.
        2. ADM page 접근
           1. 사용자:고객 관리 서버에 접근한다.
           2. 시스템 동작:해당 ID가 접근 권한이 있는지 확인함.
        3. 고객 정보 열람
           1. 사용자:고객 ID를 통해 고객 정보를 열람한다.
           2. 시스템 동작:사용자가 클릭한 고객 ID와 일치하는 ID를 데이터베이스에서 가져와 출력한다.
        4. 고객 정보 수정
           1. 사용자:고객정보 열람페이지에서 수정을 누른다
           2. 시스템 동작:고객 정보 수정 페이지로 이동한다.
           3. 사용자:수정하고자 하는 정보를 수정하고 저장하기를 누른다.
           4. 시스템 동작:업데이트 된 정보를 데이터베이스에 반영한다.
        5. 고객 정보 삭제
           1. 사용자:고객 정보 열람페이지에서 삭제를 누른다.
           2. 시스템 동작:삭제된 정보를 데이터베이스에 반영한다.
        6. 상품 등록
           1. 사용자:상품등록 페이지에서 이미지 추가를 누른다.
           2. 시스템 동작:이미지 추가 팝업을 띄운다.
           3. 사용자:이미지를 추가한다.
           4. 시스템 동작:이미지를 분석해,이와 일치하는 카테고리와 태그를 찾고,이를 화면에 출력한다.
           5. 사용자:추가하고자 하는 정보를 입력하고 저장하기를 누른다.
           6. 시스템 동작:새로운 정보를 데이터베이스에 반영한다.
        7. 상품 수정
           1. 사용자:상품 정보에서 상품 수정을 누른다.
           2. 시스템 동작:상품 등록 페이지로 이동한다
           3. 사용자:바꾸고자 하는 정보를 바꾸고 저장하기를 누른다.
           4. 시스템 동작:업데이트 된 정보를 데이터베이스에 반영한다.
        8. 주문 관리
           1. 사용자:주문 관리 페이지로 이동한다.
           2. 시스템 동작:주문 관리 페이지로 이동한다.이번 달 가장 많이 팔린 상품,팔린 상품 목록을 출력한다.
           3. 사용자:고객 ID를 클릭해 한 고객 당 주문 내역을 확인한다.
           4. 시스템 동작:데이터베이스에서 입력된 고객 ID에 해당하는 주문내역을 가져온다.
     2. 구매자 Test Case
        1. 회원가입
           1. 사용자:회원가입 페이지에서 요구하는 모든 데이터(ID, PW, 이름,주소,핸드폰번호,전화번호,이메일)을 입력하고 저장하기를 누른다.
           2. 시스템 동작:중복된 ID가 있는지 확인 후,중복이 없으면 데이터베이스에 새로 저장하고,중복이 있으면,가입 실패 문구를 띄운다.
        2. 로그인
           1. 사용자: ID, PW로 로그인을 시도함.
           2. 시스템 동작:데이터베이스 내에서 일치하는 ID, PW가 있는지 확인함.
        3. 개인 정보 수정
           1. 사용자:Mypage에서 개인 정보 수정을 누른다
           2. 시스템 동작:고객 정보 수정 페이지로 이동한다.
           3. 사용자:수정하고자 하는 정보를 수정하고 저장하기를 누른다.
           4. 시스템 동작:업데이트 된 정보를 데이터베이스에 반영한다.
        4. 상품 주문
           1. 사용자:원하는 상품을 클릭해 장바구니에 담기를 누른다.
           2. 시스템 동작:사용자가 클릭한 상품의 ID와 데이터를 장바구니에 넘긴다.장바구니에 있는 모든 상품의 total\_price를 계산한다.
           3. 사용자:장바구니 페이지 에서 주문하기를 누른다.
           4. 시스템 동작:장바구니 내의 상품 데이터들과 total\_price가 주문하기 페이지로 넘어간다.
           5. 사용자:주문 정보를 확인하고 결제하기를 누른다.
           6. 시스템 동작: total price를 외부 결제 모듈에 넘긴다.

1. **Development Environment**
   1. Objectives

목표 시스템 개발을 위해 사용할 프로그램 언어를 밝히고,해당 언어에 대한 간략한 소개를 한다.

* 1. Programming Language
     1. PHP

PHP는 서버측 스크립트 언어로 이용되며,웹서버에서 실행된다.PHP 스크립트 문서가 요구될 때 마다,서버에서 PHP 스크립트가 실행하여 결과를 웹 브라우저로 보낸다. C언어와 자바 언어와 비슷한 문법을 따르며,클래스 기반의 객체지향프로그래밍을 지원하고있다.최근,오픈소스 웹 어플리케이션 개발에 많이 이용되고 있으나, 보안이 매우 취약한 약점이 있다.

Figure PHP

* + 1. Python



Figure Python

최근활용도가 높아지고 있는 언어로,머신러닝,그래픽,웹 개발 등 여러 분야에서 사용되고 있다.간결한 문법 구조와 풍부한 라이브러리로 생산성이 매우 높고,배우기 쉽다는 장점이 있다.속도가 느리고,모바일 환경에서는 사용이 어렵다는 단점이 있으나,본 시스템은 모바일을 지원하고 있지 않으므로,파이썬을 활용하였다.

* + 1. Javascript



Figure Javascript

크로스 플랫폼,객체지향 스크립트 언어로,웹페이지의 동작을 담당한다.컴파일 과정이 없기 때문에,빠른 시간 안에 스크립트 코드를 작성할 수 있도록 해주며,기존의 C언어, 자바언어와 다르게 단순한 구조를 가지고 있어 초보 개발자들이 배우기 쉽다.하지만,자바스크립트 역시, PHP와 마찬가지로 보안이 매우 취약하다는 단점이 있다. HTML 소스코드에 함께 작성되면서 소스코드가 외부로 공개되는데,이 때 보안이 매우 취약해진다.

1. **Index**
   1. Objectives

본 문서에 등장하는 인덱스(Inex)를 정리한다.

* 1. Alphabet Index

Ａ

Admin, 7

Ｃ

Creately, 5

Ｄ

Diagram, 3, 5, 6, 7, 8, 11, 12, 16, 17, 19, 20, 21, 22, 23, 24, 25, 38

Ｐ

PHP, 34, 35, 36

Ｕ

UML Diagram, 5

ㄱ

개인정보, 31, 33

개인화, 5

객체지향프로그래밍, 35

고객관리, 7, 32

관리자, 7

기능, 5, 31

ㄷ

데이터베이스, 7, 9, 10, 11, 12, 14, 15, 17, 18, 32, 33, 34

딥러닝, 5

ㄹ

라이브러리, 35

ㅁ

머신러닝, 35

ㅂ

보안, 31, 35, 36

ㅅ

서버, 32, 35

서브시스템, 3, 6

설계명세서, 1, 4, 5

소프트웨어엔지니어, 3

스크립트, 35, 36

시나리오, 4, 31

시스템, 3, 4, 5, 6, 7, 8, 19, 31, 32, 33, 34, 35

시스템아키텍처, 3

ㅇ

아이템관리, 7

아이템정보, 7

오픈소스, 35

요구사항명세서, 5

인터넷쇼핑몰, 5

ㅈ

주문내역관리, 7

주문정보, 7, 10, 15, 34

ㅋ

카테고리화, 5

컴포넌트, 3, 6, 19, 31

컴퓨터, 5

클라우드, 5

클래스, 3, 8, 9, 10, 14, 15, 17, 18, 35

ㅌ

태그, 5, 32

ㅍ

판매자, 7, 31

* 1. Figure Index

[Figure 1 Creately.com 9](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서.docx#_Toc467613965)

[Figure 2 Creatly objects 9](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서.docx#_Toc467613966)

[Figure 3 PHP 49](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서.docx#_Toc467613967)

[Figure 4 Python 50](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서.docx#_Toc467613968)

[Figure 5 Javascript 50](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서.docx#_Toc467613969)

* 1. Diagram Index

[Diagram 1 Overview System Architecture 10](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757340)

[Diagram 2 System Architecture 11](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757341)

[Diagram 3 Deployment Diagram 12](#_Toc468757342)

[Diagram 4 System Overview Class Diagram 13](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757343)

[Diagram 5 Sign\_in Class Diagram 15](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757344)

[Diagram 6 Search Class Diagram 17](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757345)

[Diagram 7 Board Class Diagram 20](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757346)

[Diagram 8 ADM Class Diagram 22](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757347)

[Diagram 9 Image Analysis Class Diagram 24](#_Toc468757348)

[Diagram 10 Sign\_up Sequence Diagram 25](#_Toc468757349)

[Diagram 11 Sign\_in Sequence Diagram 25](#_Toc468757350)

[Diagram 12 Search Sequence Diagram 26](#_Toc468757351)

[Diagram 13 Cart Sequence Diagram 27](#_Toc468757352)

[Diagram 14 Purchasing Cart Sequence Diagram 27](#_Toc468757353)

[Diagram 15 Purchasing Item Sequence Diagram 28](#_Toc468757354)

[Diagram 16 Purchasing and Payment Sequence Diagram 28](#_Toc468757355)

[Diagram 17 Board Sequence Diagram 29](#_Toc468757356)

[Diagram 18 Mypage Sequence Diagram 30](#_Toc468757357)

[Diagram 19 ADM Sequence Diagram 30](#_Toc468757358)

[Diagram 20 Item Add and Correction Sequence Diagram 31](#_Toc468757359)

[Diagram 21 User Management Sequence Diagram 31](#_Toc468757360)

[Diagram 22 Order Management Sequence Diagram 32](#_Toc468757361)

[Diagram 23 Overall State Diagram 33](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757362)

[Diagram 24 Sign\_up State Diagram 33](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757363)

[Diagram 25 Sign\_in State Diagram 34](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757364)

[Diagram 26 Search State Diagram 34](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757365)

[Diagram 27 Choose\_item State Diagram 34](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757366)

[Diagram 28 Cart State Diagram 35](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757367)

[Diagram 29 Purchasing State Diagram 35](#_Toc468757368)

[Diagram 30 Purchasing Cart State Diagram 36](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757369)

[Diagram 31 Payment State Diagram 36](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757370)

[Diagram 32 Shipment State Diagram 37](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757371)

[Diagram 33 Mypage State Diagram 37](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757372)

[Diagram 34 Board State Diagram 38](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757373)

[Diagram 35 Item Management State Diagram 39](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757374)

[Diagram 36 ADM State Diagram 39](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757375)

[Diagram 37 User Management State Diagram 40](file:///C:\Users\Jeongsu%20Jang\Desktop\과제\소프트웨어%20공학\final%20project\Software%20Engineering%20Final%20Project%20Team%20D-설계명세서(수정본).docx#_Toc468757376)

* 1. Table Index

[Table 1 User Database 41](#_Toc468788336)

[Table 2 User Database2 41](#_Toc468788337)

[Table 3 User Database3 41](#_Toc468788338)

[Table 4 Item Database 42](#_Toc468788339)

[Table 5 Item Database2 42](#_Toc468788340)

[Table 6 Cart Database 42](#_Toc468788341)

[Table 7 Order Database 44](#_Toc468788342)

[Table 8 Board Database 45](#_Toc468788343)