

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100

version
v1.0

【P-실무프로젝트】

챗봇을 이용한 중고거래 서비스

설 계 서

2019.12.4

가천대학교 컴퓨터공학과

- 팀장 ■ 한혜경(201736058)
- 팀원 ■ 이우석(201433860)
- 팀원 ■ 이다은(201632230)
- 팀원 ■ 서지연(201637615)

문서 정보

본 문서는 챗봇을 이용한 중고거래 서비스 개발을 위한 요구규격 및 설계서입니다.

버전	1.0
작성일	2019-12-04
상태	<input type="checkbox"/> 완료 <input checked="" type="checkbox"/> 진행 중 <input type="checkbox"/> 초안

버전	변경한 사람	변경한 날짜	버전업 변경(또는 추가)내용
0.1	이병문 교수님	11/27	설계서 기본 템플릿 제공
0.2	이우석	11/28	프로젝트 개요 및 환경구성 작성
0.3	이우석	11/28	서비스기능 정의 작성
0.4	한혜경	11/29	프로세스(기능) 설계 작성
0.4	이다은	11/30	데이터베이스 설계 작성
0.5	한혜경	12/1	완료된 부분 버전 통합
0.6	이다은	12/3	데이터베이스 설계 수정
0.7	한혜경	12/3	서비스기능 정의 수정
0.8	이다은	12/4	REST API 인터페이스 설계 수정
0.9	서지연	12/4	UI 디자인 수정
1.0	한혜경	12/4	설계서 작성 완료

목 차

1. 프로젝트 개요	
1.1 개발목표와 범위	4
1.2 개발일정/산출물	6
1.3 개발조직/역할	6
2. 서비스기능 정의	
2.1 서비스 개요	7
2.2 유스케이스 모델	9
2.3 요구기능 정의(사용자)	12
2.4 요구기능 정의(관리자)	15
3. 프로세스(기능) 설계	
3.1 시스템 아키텍쳐	18
3.2 사용자기능 설계(Sequence Diagram)	19
3.3 관리자기능 설계(Sequence Diagram)	42
4. 화면(UI)설계	
4.1 스토리보드 설계	64
4.2 초기화면(로그인 전화면)설계	65
4.3 사용자UI 설계	66
4.4 관리자UI 설계	105
5. REST API 인터페이스 설계	
5.1 서버/클라이언트 구조	128
5.2 인터페이스 정의	129
5.3 상세 REST API 설계	132
6. 데이터베이스 설계	
6.1 ERD	174
6.2 논리적설계(테이블명세서)	175
6.3 물리적설계(SQL스크립트)	179
7. 환경구성	
7.1 개발환경 및 운영환경	185
7.2 소스디렉터리 구조	186

1. 프로젝트 개요

1.1. 개발목표와 범위

■ 개발목표

본 설계서는 PC환경에서 중고거래를 위한 쇼핑몰 웹 서비스를 개발하기 위한 설계서이다. 주요 사용자는 중고물을 거래하고 싶어하는 개인들이며, 챗봇(카카오톡 서비스)을 통해서 접속하고 사용하기 편리하도록 고안되는 것을 전제로 한다. 쇼핑몰 웹 서비스를 효과적으로 관리하기 위해 관리기능도 구성하였으며 최대한 편리성과 수익성을 추구할 수 있도록 개발한다. 관리자는 웹기반 시각화된 UI를 통해 쇼핑몰을 관리할 수 있도록 한다.

아래 그림1-1은 본 서비스의 전체적인 구성을 나타내며, 중고 물품의 구매와 판매를 희망하는 사용자가 본 서비스를 사용한다.

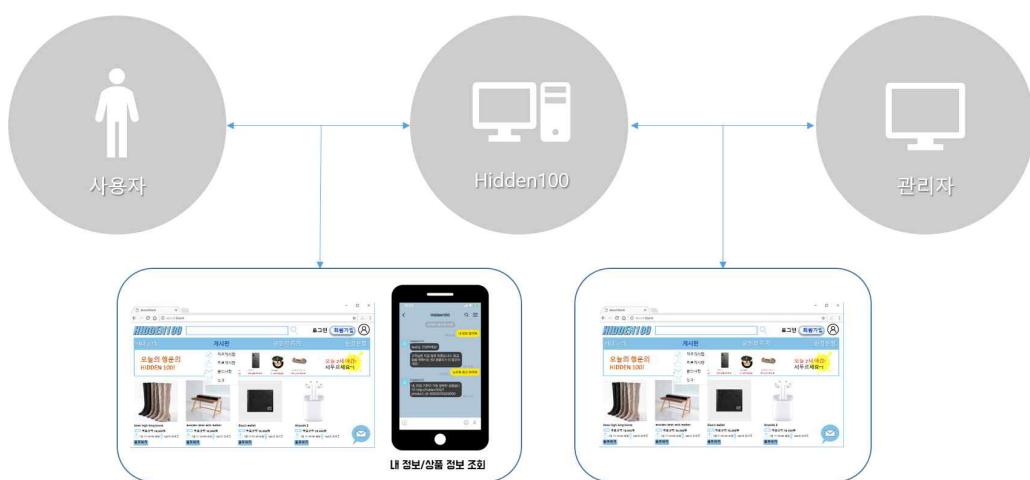


그림 1.1 Hidden100 웹서비스 설계범위

■ 개발필요성

현재 가장 보편적인 중고 거래 사이트들은 회원 관리와 거래 방식의 구조적 문제로 인하여 사기 행위를 하여도 별다른 제재를 가할 수단이 없다. 오픈마켓과 비슷한 형식으로 쇼핑몰이 제재하고 관리하는 방식도 중고거래의 소량상품, 다수의 판매자 특성 때문에 비효율적이다. 다수의 판매자와 구매자를 유도해야 하는 중고거래에서는 오픈마켓처럼 판매자 등록 절차가 있는 경우 불리하다. 그리고 기존 방식은 구매자와 판매자 간에 거래 과정에서 판매자가 제시한 금액과 구매자가 원하는 가격에 차이가 있어 이를 조정하는데 시간, 감정 등이 소요 되는 경우가 많다. 따라서 이러한 기존 방식을 개선할 필요가 있다.

대부분의 중고거래 쇼핑몰은 이 해결책으로 한 종류의 상품에 집중하여 판매와 구매를 모두 대행하는 경우가 많으나 수동으로 해결하기에는 한계가 있다. 또 앞서 말한 방법으로는 구매가와 판매가에서 차익을 낸다 하더라도 구매 심리를 이용하기 위해서는 한계가 있고, 인력이 소모되기 때문에 쇼핑몰에서는 인건비를 감수하고 큰 이득을 보기 어렵다. 따라서 관리자가 하나하나 확인하지 않아도 중고거래가 가능하며 거래사기를 방지하기 위해 자체 재화를 사용하지만, 자체 재화의 필요성을 사용자가 납득 가능한 서비스가 필요하다.

본 서비스는 그러한 과정을 경매, 투자, 추첨 방식을 혼합하여 거래 방식 단순화, 구매자의 흥미 유발, 판매자의 수익 최대화를 통해 판매자와 구매자 모두 만족하는 결과를 만들어낸다. 판매자는 예상 가격보다 높은 값을 받을 수 있으며, 구매자는 경품추첨과 비슷한 방식으로 운이 좋다면 아주 적은 금액에 좋은 중고 물건을 구매할 수 있다. 현재 중고 거래 방식에서는 거래가 성사되기까지 판매자와 구매자 모두 중고 거래 사이트에 오래 머무를 수밖에 없다. 본 서비스만의 독특한 거래 방식과 챗봇 서비스는 구매자와 판매자 모두를 모니터 앞에서 해방시키며 언제 어디에서나 거래 과정에 참여 할 수 있는 동적인 서비스를 제공한다.

1.2. 개발일정/산출물

본 설계서는 아래 그림1.3과 같이 개발일정에 따라 진행하며, 각 단계별로 산출물을 개발한다.

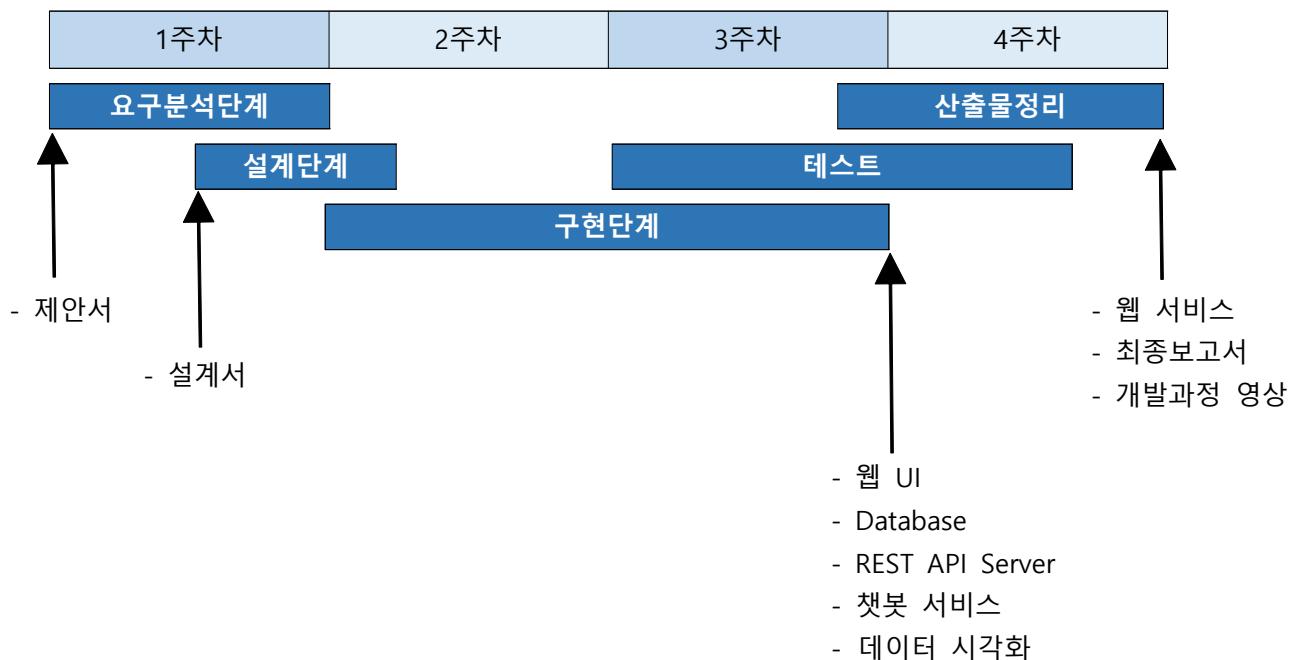


그림 1.2 개발일정 및 산출물

1.3. 개발조직/역할

역할	이름	역할(개발기능)	비고
팀장	한혜경	사용자 인증, 관리자 인증, 회원 관리, 수익 관리	
팀원	이우석	사용자 게시판, 사용자 설정, 관리자 설정	
팀원	이다은	챗봇 사용자 기능, 챗봇 관리자 기능	
팀원	서지연	중고거래(사용자), 중고거래(관리자)	

2. 서비스기능 정의

2.1. 서비스 개요

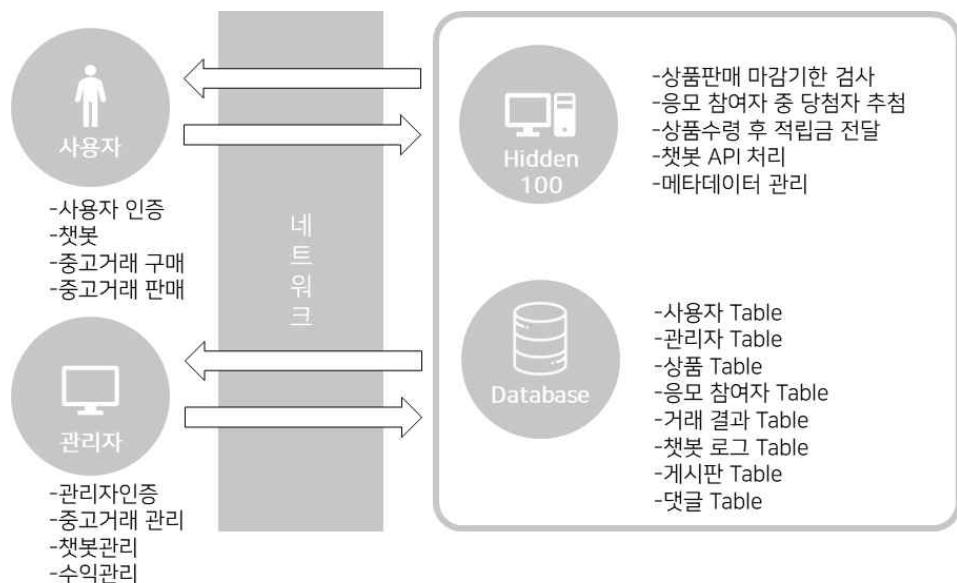


그림 2.1 Hidden100 서비스 모델 구성도

Hidden 100은 중고거래를 지원하는 웹쇼핑몰이다. 사용자는 중고거래 판매자와 구매자가 될 수 있고 관리자는 거래 수수료로 이득을 얻는다. 판매자가 선정한 가격만큼 구매자들이 소액 적립을 통해 도달하면 추첨하여 당첨자에게 중고 물품을 전달하는 방식이다. 중고거래에 자체 재화를 사용하여 사기를 방지할 수 있고 사용자에게 코인을 충전할 수 있는 기능을 지원한다. 관리자는 수익을 조회하고 회원을 관리할 수 있다. 사용자와 관리자는 RESTful API를 이용해서 서버에 저장된 데이터를 확인하고 거래에 참여하거나 정보를 수정 가입하는 등의 처리를 Web UI를 통해 수행할 수 있다.

사용자는 판매물을 조회할 수 있다. Hidden100의 비밀 소액적립형 추첨경매 서비스로 현재 적립금을 확인할 수 없고 참여자만 확인할 수 있다. 그러나 자신이 등록한 상품의 경우 적립금과 참여자 수를 모두 확인할 수 있다. 또 사용자가 상품을 조회할 때 챗봇을 통해 편하게 조회할 수 있으며, 상품 검색이 가능하다. 일부 조회를 제외한 대부분의 기능은 회원가입과 로그인으로 사용자 인증을 마친 다음에 이용 가능하다.

관리자가 상품을 조회하는 경우 관리자는 상품 등록에 대한 모든 정보를 확인할 수 있다. 현재 적립금과 가격 등록한 사용자까지 정보를 파악할 수 있으며 쇼핑몰 안에서 진행중인 전체 중고거래에 대한 확인이 가능하다. 관리자는 누적 매출과 수수료를 포함한 거래 정보에 대해 확인 가능하다. 또 챗봇 로그에서 사용자가 검색한 결과에 대한 합산

을 조회하여 쇼핑몰의 거래 흐름과 사용자의 관심에 대해 확인할 수 있다.

사용자는 코인을 충전하여 상품에 적립하는 방식으로 거래에 참여한다. 따라서 코인의 현재 적립액은 관리자와 판매자만 확인할 수 있다. 마감시간까지 적립금이 판매자가 설정한 목표금액에 도달하면 적립한 사람 중에 당첨자를 뽑아 중고 물건을 보낸다. 적립금이 높을수록 당첨 확률이 높아지며 배송은 판매자가 하기 때문에 배송정보를 입력하는 기능을 지원한다. 구매자가 배송수령을 확인하면 거래 대금이 판매자에게 전달되며 배송으로 중고거래 사기를 할 경우 구매자는 거래신고를 할 수 있다.

로그인한 사용자는 게시판의 게시글을 조회할 수 있다. 게시글을 통해 사용자들은 커뮤니티와 후기를 공유할 수 있고 신고와 문의도 게시글을 통해 처리한다. 컴퓨터로 완전히 처리할 수 없는 거래사기 같은 경우 구매자가 신고를 하면 관리자가 확인하여 구매자에게 제재를 가하거나 판매 적립금을 보내지 않고 환불하는 방식으로 사기 피해를 최소화할 수 있다.

관리자는 회원의 정보를 조회하고 회원 정보를 수정할 수 있다. 회원 계정을 정지상태로 만들면 사용자는 로그인이 불가능하며, 회원을 탈퇴시키거나 회원 정보를 수정하는 기능을 사용할 수 있다. 관리자도 일반 회원과 마찬가지로 회원가입과 로그인으로 관리자인증을 마친 다음에 관리자 기능을 이용할 수 있다.

2.2. 유스케이스 모델

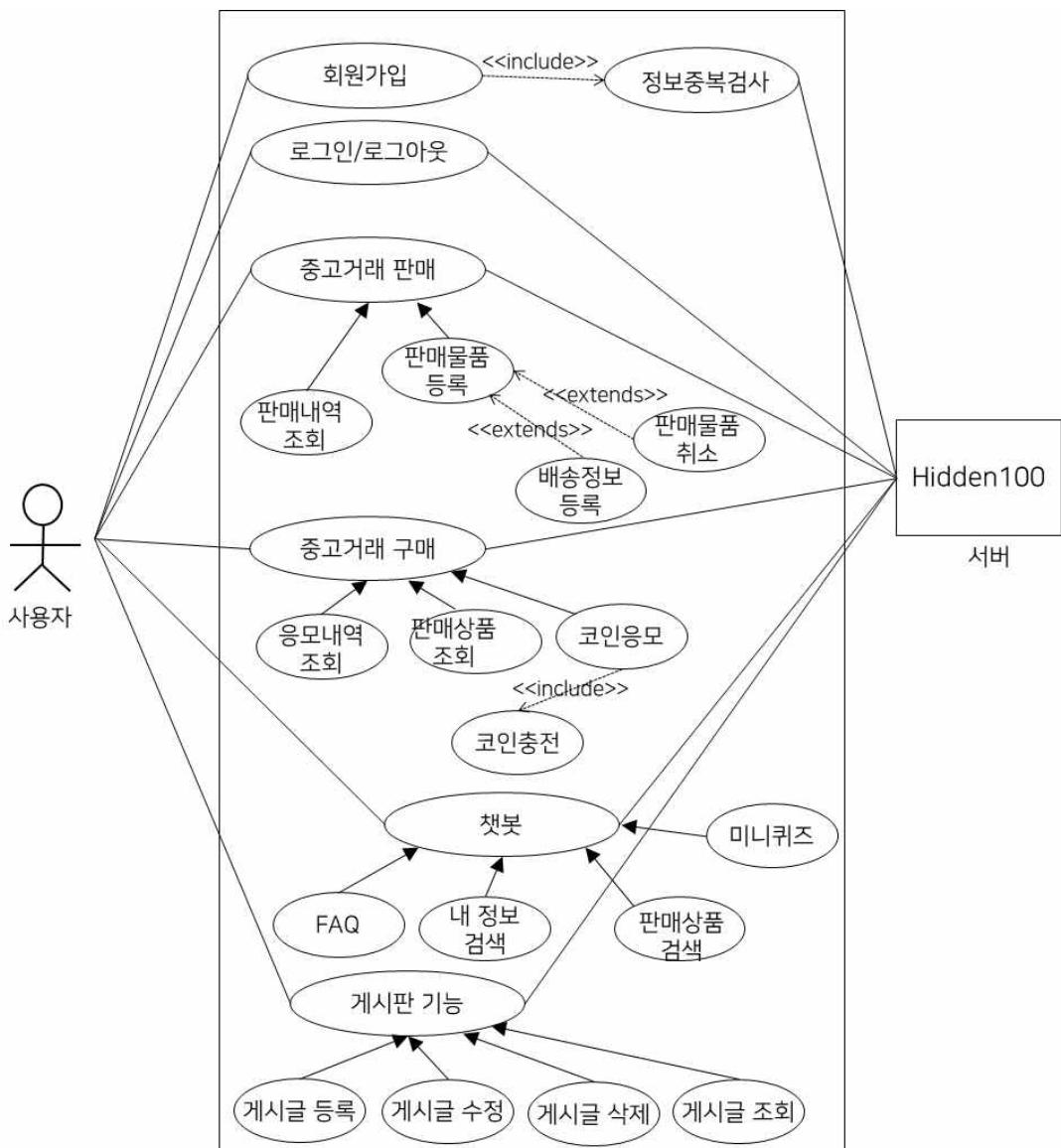


그림 2.2 사용자용 유스케이스 다이어그램

사용자는 공통적으로 게시판, 로그인, 로그아웃, 회원 탈퇴, 챗봇, 회원 가입, 정보 수정 페이지에 접근할 수 있다. 사용자는 구매자와 판매자로 나뉘게 된다. 하지만 구매자는 동시에 판매자가 될 수 있다. 판매자의 경우에도 마찬가지이다. 구매자는 챗봇을 통해 FAQ, 판매 상품 조회, 내 정보 조회, 미니 퀴즈를 할 수 있다. Hidden100의 모든 기능은 사용자 인증이 필요하다.

사용자는 구매하고자 하는 상품에 코인을 적립하고 나면 그 거래에 참여하고 있는

참여자 수 등 거래에 관련 된 정보 조회 페이지에 접근할 수 있다. 구매자는 자신이 코인을 얼마나 사용했는지 등을 조회 할 수 있는 거래 내역 조회 페이지에 접근 할 수 있다. 거래 내역 조회 페이지를 통해 거래 신고 페이지에 접근 할 수 있다. 사용자는 물품 등록 페이지에 접근 할 수 있다. 물품을 등록하고 나면 자신이 등록한 물품의 목록을 조회할 수 있는 페이지에 접근 할 수 있다. 등록한 물품을 판매 중지 할 수 있는 페이지로 접근이 가능 하다.

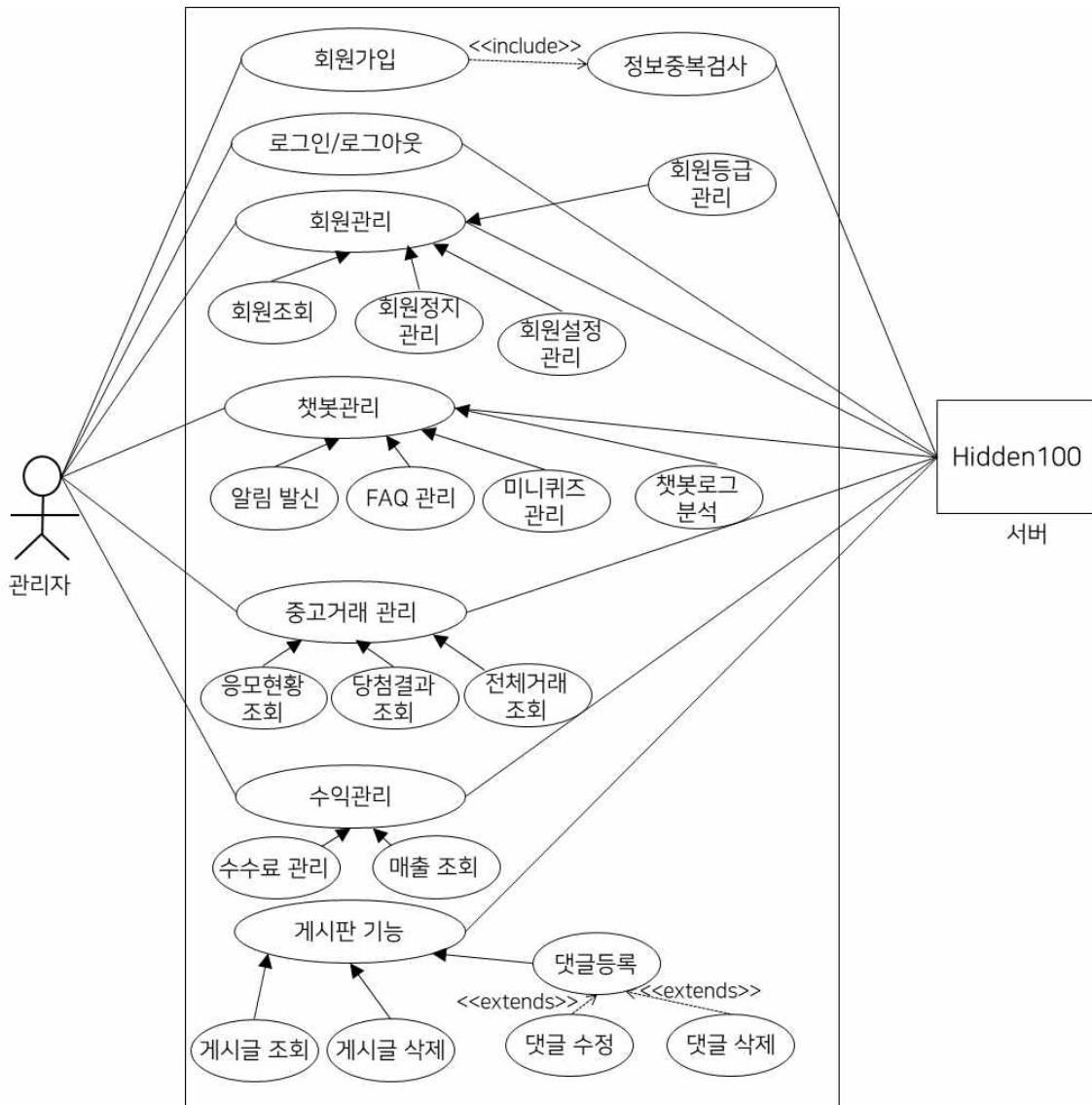


그림 2.3 관리자용 유스케이스 다이어그램

관리자는 로그인, 로그아웃, 회원가입 페이지에 접근 할 수 있다. 회원 관리 페이지를 통해 회원조회, 회원 정지 관리, 회원 탈퇴 관리, 회원 설정 관리 페이지로 이동 할 수 있다. 챗봇의 로그를 조회, 조회된 데이터를 시각화 해주는 페이지에 접근할 수 있다. 챗봇을 통해 사용자에게 알람을 발신하거나 챗봇에 새로운 정보를 등록할 수 있는 페이지에 접근할 수 있다. 관리자는 수익관리 페이지에 접근할 수 있으며, 수수료 수익 조회, 매출 총액 조회 페이지에 접근 할 수 있다. 관리자도 로그인 이후 관리자 기능을 사용할 수 있다.

2.3. 요구기능 정의 (사용자)

번호	주요기능명	상세기능명	개발담당
1)	사용자 인증	회원가입	한혜경
2)		로그인	한혜경
3)		로그아웃	한혜경
4)		정보수정	이우석
5)		회원탈퇴	이우석
6)	챗봇 기능	내 정보 조회	이다은
7)		미니퀴즈	이다은
8)		판매상품 검색	이다은
9)		FAQ	이다은
10)	중고거래(구매자)	코인충전	이우석
11)		코인응모	이우석
12)		판매상품 조회	이우석
13)		응모내역 조회	이우석
14)		당첨내역 조회	이우석
15)	중고거래(판매자)	판매물품 등록	한혜경
16)		판매물품 조회	서지연
17)		판매물품 취소	한혜경
18)		배송정보 등록	이다은
19)	게시판 기능	게시글 등록	이우석
20)		게시글 조회	이우석
21)		게시글 수정	이우석
22)		게시글 삭제	이우석

1) 회원가입

- 본 서비스를 이용하기 위해 사용자 정보를 등록한다.
- 웹페이지에서 아이디, 비밀번호, 이름, 주소 등의 개인정보를 받는다.
- 받은 정보는 사용자 데이터베이스에 저장한다.

2) 로그인

- 사용자로부터 아이디와 비밀번호를 품을 통해 받아서 데이터베이스에 저장된 값과 비교한다.
- 비교 결과에 따라 로그인 성공화면과 실패화면을 보여준다.
- 사용자는 로그인으로 사용자 인증을 마친 다음에 서비스 기능을 사용할 수 있다.
- 계정 정지 상태의 사용자는 로그인할 수 없다.

3) 로그아웃

- 사용자의 로그인 정보를 세션에서 종료시킨다.
- 로그아웃한 사용자는 다시 로그인해야 기능을 사용할 수 있다.

4) 정보수정

- 사용자의 정보를 품을 통해 수정한다.
- 수정된 값을 데이터베이스에 저장한다.
- 사용자는 아이디를 제외한 정보를 수정할 수 있다.

5) 회원탈퇴

- 회원탈퇴 기능을 통해 사용자는 자신의 계정을 삭제할 수 있다.
- 탈퇴한 사용자의 데이터는 데이터베이스에서 삭제된다.

6) 내 정보 조회

- 사용자의 이름, 등급, 코인 등의 정보를 챗봇에서 안내한다.
- 사용자가 챗봇 버튼을 누르면 간편하게 조회할 수 있다.

7) 미니퀴즈

- 사용자는 Hidden100에 관련된 간단한 퀴즈를 풀어서 보상을 받을 수 있다.
- 버튼으로 퀴즈의 답을 보낼 수 있고 정답에 따라 사용자는 소정의 보상을 받는다.

8) 판매상품 검색

- 사용자는 챗봇을 통해 상품을 검색할 수 있다.
- 사용자가 챗봇 화면에서 버튼으로 상품의 종류를 선택한다. 챗봇에서는 해당 상품 중에 마감시간이 가장 짧은 것을 골라 사용자에게 URL형태로 안내한다.

9) FAQ

- 사용자가 자주 하는 질문을 모아 챗봇으로 확인할 수 있다.
- 사용자는 Hidden100 웹쇼핑몰에 대한 기능을 안내받는다.

10) 코인충전

- Hidden100에서 응모에 사용하는 자체 화폐를 충전한다.
- 충전한 전자화폐 값을 데이터베이스에 저장한다.
- 사용자는 충전한 코인을 이용하여 응모에 참여할 수 있다.

11) 코인응모

- 사용자는 구매하고자 하는 상품에 코인을 응모할 수 있다.
- 응모한 사용자는 거래 참여자 응모금액과 함께 데이터베이스에 저장되고 해당 상품의 응모금액이 증가한다.
- 응모금액이 목표금액을 넘으면 거래가 확정되며, 사용자는 자신이 등록한 물품을 제외하고 현재 적립금액을 확인할 수 없다.

12) 판매상품 조회

- 사용자는 판매하는 상품을 조회할 수 있다.
- 특정 단어에 대한 검색 형식의 판매상품 조회도 가능하다.

13) 응모내역 조회

- 사용자는 자신이 코인으로 응모한 물품을 테이블로 모아서 확인할 수 있다.

14) 당첨내역 조회

- 사용자는 자신이 응모한 상품의 당첨 결과를 테이블로 확인할 수 있다.
- 당첨 중고 상품에 대한 배송 정보도 확인 가능하다.

15) 판매물품 등록

- 중고거래 쇼핑몰인 Hidden100에서는 사용자가 별도의 인증 과정 없이 중고물품을 등록할 수 있다.
- 물건의 종류, 상품 명, 목표 금액, 마감기간, 이미지 등을 폼을 통해 입력한다.
- 입력된 내용은 데이터베이스에 저장되며 입력과 동시에 마감기간까지 판매가 시작된다.

16) 판매물품 조회

- 자신이 등록한 물품을 데이터베이스에서 조회할 수 있다.
- 자신이 등록한 물품에 제한해서 현재 적립금액을 확인할 수 있다.

17) 판매물품 취소

- 자신이 등록한 물품의 판매를 취소한다.
- 판매한 상품은 데이터베이스에서 삭제되며 쌓인 적립금은 환불된다.

18) 배송정보 등록

- 구매자에게 물건을 배송한 후 이를 증명하기 위해 폼에 배송장 정보를 입력한다.
- 이 정보를 당첨자가 확인할 수 있다.

19) 게시글 등록

- 사용자가 게시글을 작성하면 데이터베이스에 등록한다.
- 게시글 제목, 내용, 이미지 등의 정보를 작성할 수 있다.

20) 게시글 조회

- 사용자가 작성한 게시글을 확인할 수 있으며, 문의게시판과 신고게시판은 작성한 사용자만 자신의 게시글을 확인할 수 있다.

21) 게시글 수정

- 작성한 사용자가 자신의 게시글을 수정할 수 있다.

22) 게시글 삭제

- 작성한 사용자가 자신의 게시글을 삭제할 수 있다.

2.4. 요구기능 정의 (관리자)

번호	주요기능명	상세기능명	개발담당
1)	관리자 인증	관리자가입	한혜경
2)		로그인	한혜경
3)		로그아웃	한혜경
4)		관리자탈퇴	한혜경
5)	회원관리	회원조회	서지연
6)		회원등급 관리	서지연
7)		회원설정 관리	서지연
8)		회원정지 관리	서지연
9)	챗봇관리	알림 발신	이다은
10)		FAQ 관리	이다은
11)		챗봇로그 분석	이다은
12)		미니퀴즈 관리	이다은
13)	중고거래 기능	응모현황 조회	이우석
14)		당첨결과 조회	이우석
15)		전체거래 조회	이우석
16)	수익관리	수수료 관리	한혜경
17)		매출 조회	한혜경
18)	게시판 기능	게시글 조회	한혜경
19)		게시글 삭제	한혜경
20)		댓글 등록	이우석
21)		댓글 수정	이우석
22)		댓글 삭제	이우석

1) 관리자가입

- 폼에 정보를 입력하여 관리자를 등록한다.
- 입력된 관리자 정보는 데이터베이스에 저장된다.

2) 로그인

- 관리자 아이디와 비밀번호를 폼에 입력하여 로그인을 진행할 수 있다.
- 입력받은 아이디와 비밀번호가 데이터베이스에 있는 정보와 비교하여 로그인을 진행 한다. 데이터가 일치하는 경우 로그인 된 상태의 메인페이지를 안내하고, 불일치하는 경우 알림 메시지를 전송한다.
- 관리자 기능은 회원가입과 로그인으로 관리자 인증에 성공한 다음 사용할 수 있다.

3) 로그아웃

- 관리자는 로그아웃으로 세션에 저장된 로그인 데이터를 삭제할 수 있다.
- 로그인 해있는 동안 변경된 정보들을 데이터베이스에 저장하고 로그아웃한다.

- 다시 관리자 기능을 이용하기 위해서는 로그인으로 관리자 인증이 필요하다.

4) 관리자탈퇴

- 해당 관리자 계정을 사용하지 않는 경우나 유출이 되어 삭제할 때 사용한다.
- 관리자 계정을 데이터베이스에서 삭제하여 해당 계정으로 로그인할 수 없도록 만든다.

5) 회원조회

- 관리자는 데이터베이스에 저장된 사용자의 모든 정보를 조회할 수 있다.
- 사용자의 정보는 테이블 형태로 관리자 페이지에 출력된다.

6) 회원등급 관리

- 관리자는 데이터베이스에 있는 사용자의 등급을 조회하고 수정하여 데이터베이스에 저장할 수 있다.

7) 회원설정 관리

- 관리자는 데이터베이스에 저장되어 있는 사용자의 설정을 조회하고 수정할 수 있다.

8) 회원정지 관리

- 관리자는 사용자의 계정을 정지상태로 만들 수 있다. 정지상태가 된 사용자는 로그인으로 사용자 인증을 할 수 없다.

9) 알림 발신

- 관리자는 챗봇을 통해 사용자에게 수동 메시지를 전달한다.

10) FAQ 관리

- 관리자는 사용자가 챗봇을 통해 확인하는 FAQ를 편집할 수 있다.

11) 챗봇로그 분석

- 관리자는 챗봇에서 가장 많이 사용하는 기능을 챗봇로그 분석을 통해 확인할 수 있다.
- 챗봇 사용은 데이터베이스에 저장되어 해당 내용을 분석하여 관리자에게 안내한다.

12) 미니퀴즈 관리

- 사용자가 챗봇을 통해 진행하는 미니퀴즈의 질문, 정답을 관리자가 수정할 수 있다.

13) 응모현황 조회

- 관리자는 현재 응모 중인 중고 상품의 정보를 모두 조회한다.
- 관리자의 상품검색 및 조회 화면은 사용자와 거의 동일한 UI를 사용하나 응모하기 버튼 대신 현재 적립금을 확인할 수 있다.

14) 당첨결과 조회

- 관리자는 목표금액까지 적립금을 모아 성사된 중고 거래에 대한 정보를 확인한다.

15) 전체거래 조회

- 관리자는 중고 거래의 성사 결과에 상관 없이 모든 정보를 열람할 수 있다.

16) 수수료 관리

- 거래가 성사되는 경우 받는 수수료의 비율을 관리자가 수정할 수 있다.

17) 매출 조회

- 관리자는 수수료로 생긴 이익을 확인할 수 있다.

18) 게시글 조회

- 관리자는 문의게시판과 신고게시판을 포함하여 모든 게시판을 조회할 수 있다.
- 관리자는 게시글을 조회하여 관리자의 판단 하에 게시글 삭제 혹은 댓글 등의 기능을 사용한다.

19) 게시글 삭제

- 논란의 여지가 있거나 주제를 벗어난 게시물을 삭제하여 관리한다.

20) 댓글 등록

- 문의게시판과 신고게시판에서 관리자는 사용자의 문의사항이나 신고 결과에 대해 댓글을 통해 안내할 수 있다.

21) 댓글 수정

- 관리자는 작성한 댓글을 수정할 수 있다.

22) 댓글 삭제

- 관리자는 작성한 댓글을 삭제할 수 있다.

3. 프로세스(기능) 설계

3.1. 시스템 아키텍쳐

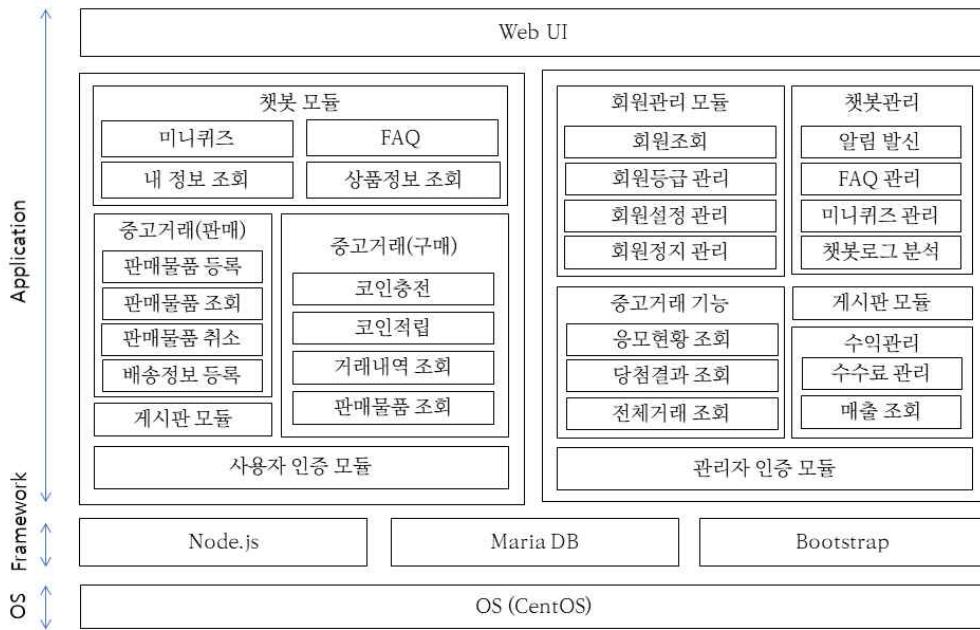


그림 3.1 시스템 아키텍쳐

소액적립형 경매 서비스 Hidden100은 그림 3.1과 같이 시스템을 사용자 기능과 관리자 기능으로 구분할 수 있다. 웹서버는 CentOS 기반으로 NPM(Node Package Manager), Node.js, MariaDB, Bootstrap 그리고 외부모듈을 이용하여 구현한다. NPM 을 이용하여 외부모듈을 관리한다. 사용자와 관리자는 Web UI를 이용하여 서비스를 사용할 수 있다.

사용자 기능과 관리자 기능은 각각 인증을 마친 후에 사용할 수 있지만, 물품 조회 기능과 회원가입은 사용자 인증이 없어도 이용 가능하다. 사용자 기능으로 사용자 인증, 중고거래 구매와 판매, 사용자 설정, 게시판, 챗봇 기능을 지원한다. 관리자 기능으로 관리자 인증, 중고거래 관리, 회원관리, 수익관리, 게시판, 챗봇 관리 기능을 제공한다. Hidden100의 서비스 특성으로 사용자는 응모 가격 등을 확인할 수 없으나 관리자는 모두 확인할 수 있기 때문에 별도의 기능으로 모듈화한다.

3.2. 사용자기능 설계 (Sequence Diagram)

1) 사용자 인증 - 회원가입

사용자는 사용자 인증 후 Hidden100 서비스를 이용할 수 있기 때문에 회원가입이 필요하다. 회원가입 기능을 구현하기 위해 아래와 같이 설계한다.

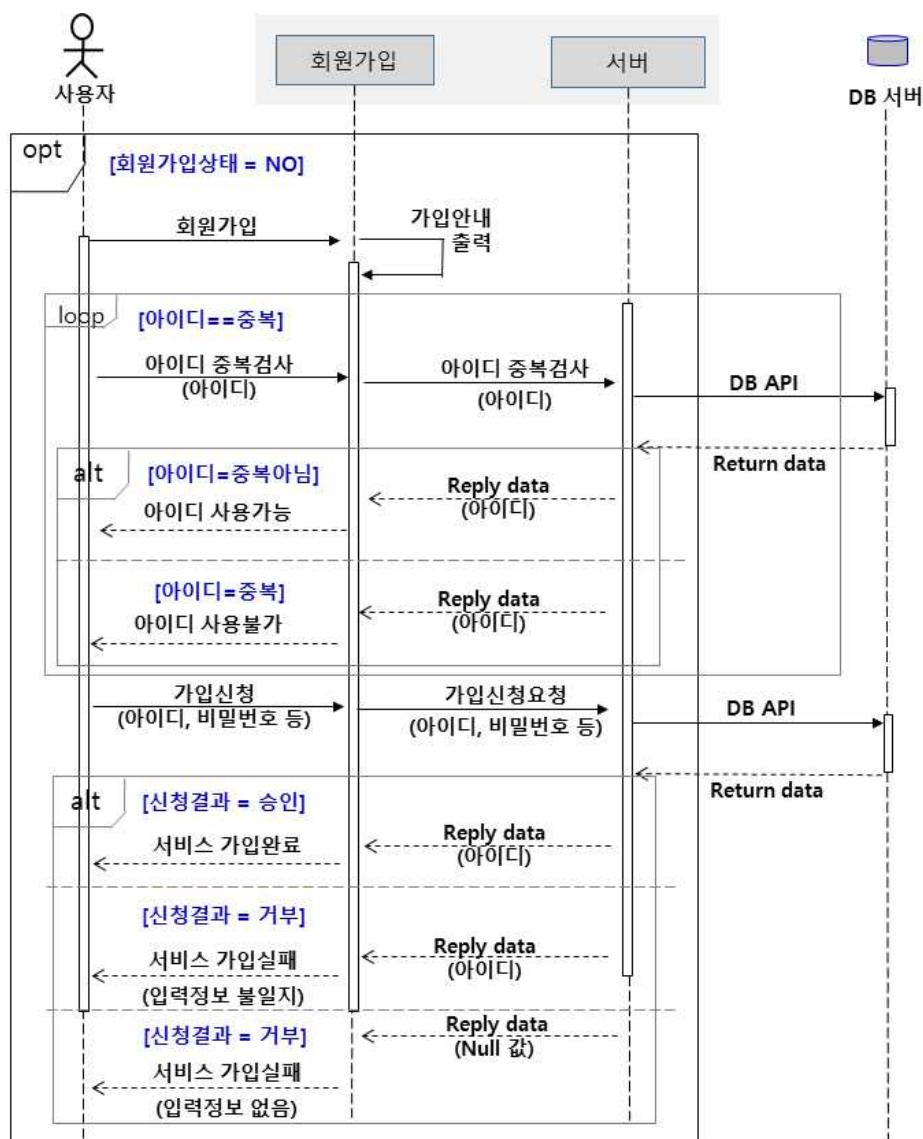


그림 3.2 회원가입의 Sequence Diagram

그림 3.2를 보면 사용자가 회원가입 하지 않은 경우에 회원가입을 진행한다. 사용자는 아이디를 중복검사한다. 아이디의 경우 DB에서 키로 사용되기 때문에 중복을 허용하지 않으며 중복이 아닌 경우에만 다음 단계로 넘어갈 수 있다. 아이디가 중복

이 아닐 때까지 검사를 반복한다. 사용자는 중복검사를 마친 후에 회원가입 정보를 입력하여 회원가입을 진행할 수 있다. 만약 입력한 값에 오류가 있으면 회원가입신청이 거부된다. 입력 정보가 없는 경우에도 회원가입 신청이 거부된다.

2) 사용자 인증 - 로그인기능

Hidden100의 사용자 기능을 이용하기 위해서는 아이디와 패스워드를 이용하여 로그인 기능이 필요하다. 로그인을 구현하기 위해 아래와 같이 설계한다.

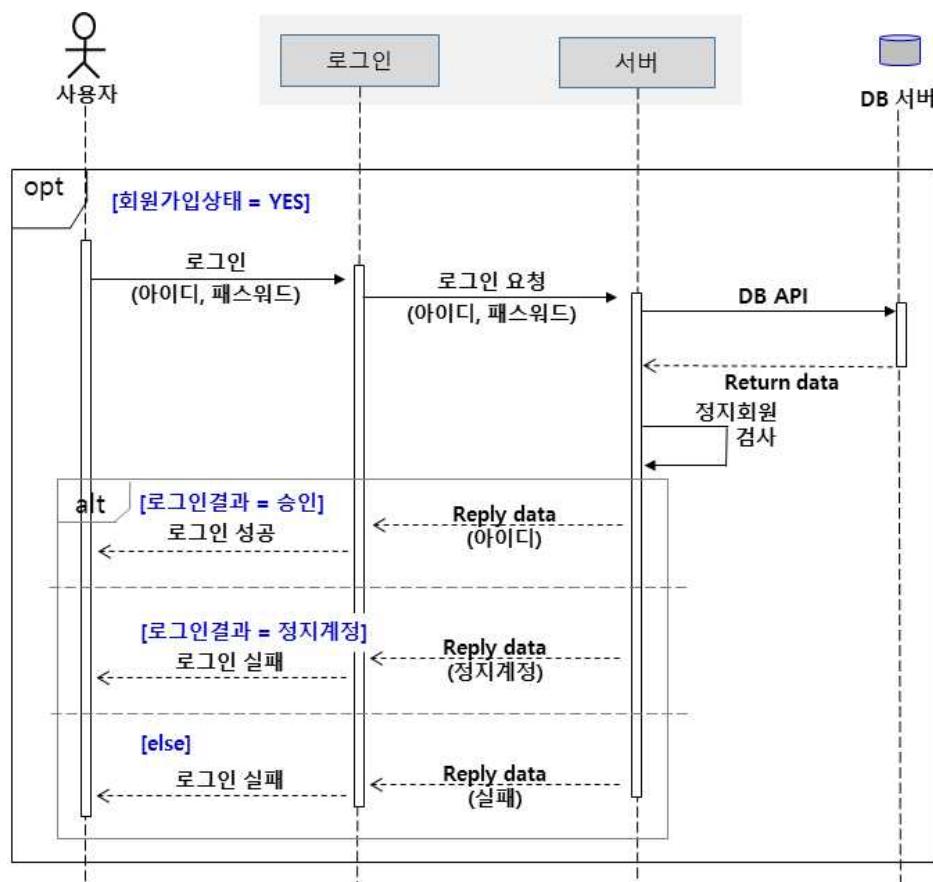


그림 3.3 로그인의 Sequence Diagram

그림 3.3를 보면 회원가입한 경우에 로그인으로 사용자 인증이 가능하다. 사용자는 아이디와 패스워드를 입력하고 로그인 버튼을 눌러 로그인을 요청한다. 서버에서는 로그인 요청을 받아 DB에서 아이디를 키로 회원정보를 받아와 입력받은 정보와 비교한다. 아이디와 패스워드가 DB에 저장된 회원정보와 일치하면 로그인에 성공한다. 로그인하고 나서 다른 Hidden100 사용자 서비스를 이용할 수 있다. 패스워드가 다르거나 정지계정인 경우 로그인에 실패한다. 로그인에 실패한 경우 실패 메시지와 실패 사유를 안내한다.

3) 사용자 인증 - 로그아웃 기능

사용자가 서비스 이용을 종료하거나 다른 계정으로 전환이 필요한 순간에 로그아웃 기능을 수행한다. 로그아웃으로 사용자는 세션에 저장된 로그인 상태를 종료할 수 있다.

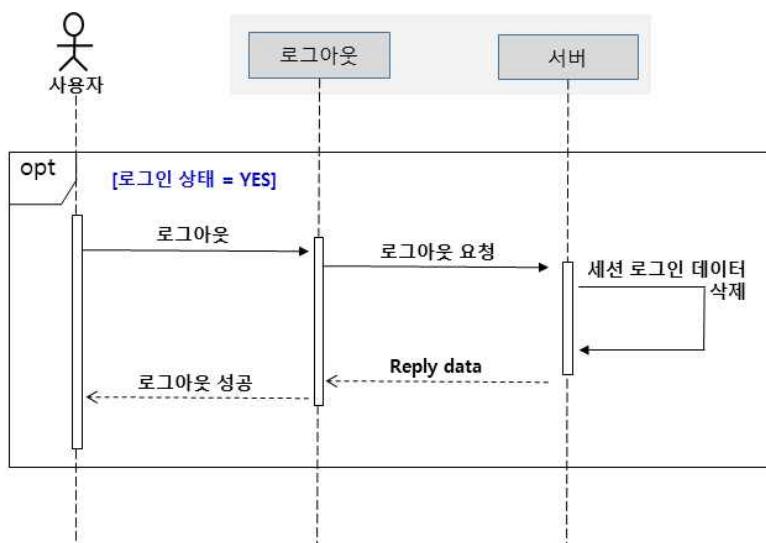


그림 3.4 로그아웃의 Sequence Diagram

그림 3.4를 보면 사용자는 로그아웃을 수행하기 위해 로그아웃 버튼을 누른다. 로그아웃 기능은 서버로 가서 현재 세션에 저장된 사용자의 로그인 데이터를 삭제한다. 서버에서 로그아웃을 처리한 후에 사용자에게 로그아웃된 화면으로 전송하여 보여준다.

4) 사용자 인증 - 정보수정 기능

사용자가 본인의 정보를 변경하기 위해 정보수정 기능을 사용한다. 사용자는 로그인한 다음 회원 본인의 데이터를 수정할 수 있으며 아래와 같이 설계한다.

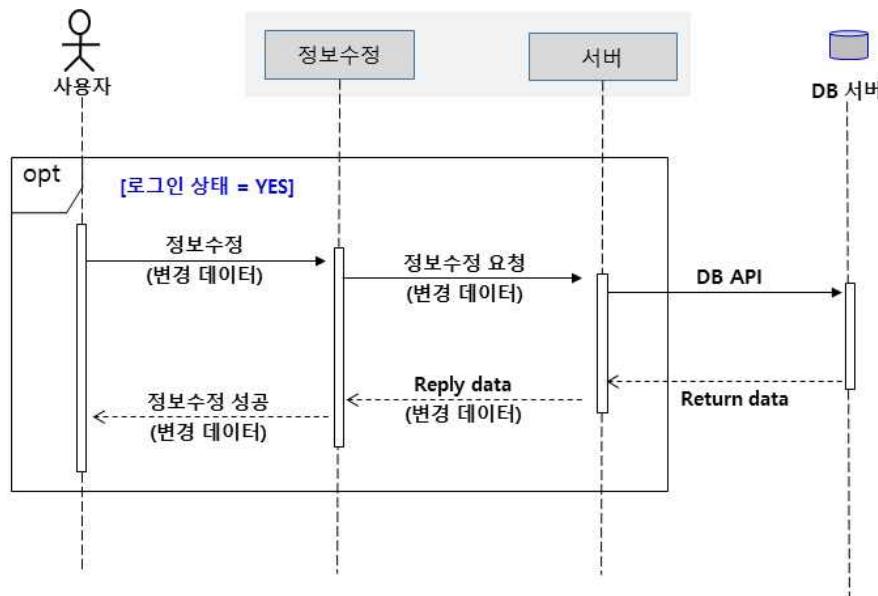


그림 3.5 정보수정의 Sequence Diagram

그림 3.5를 보면 사용자는 웹화면을 통해 자신의 정보를 수정할 수 있다. 변경하고 싶은 정보를 수정하고 정보수정 버튼을 누르면 입력한 데이터가 서버로 넘어가 DB에 저장된다. 변경하면 사용자의 화면에도 변경 데이터로 출력한다.

5) 사용자 인증 - 회원탈퇴 기능

사용자가 자신의 계정을 삭제하기 위해 회원탈퇴 기능을 사용한다. 사용자는 로그인 한 다음 회원탈퇴를 진행할 수 있으며 다음과 같이 설계한다.

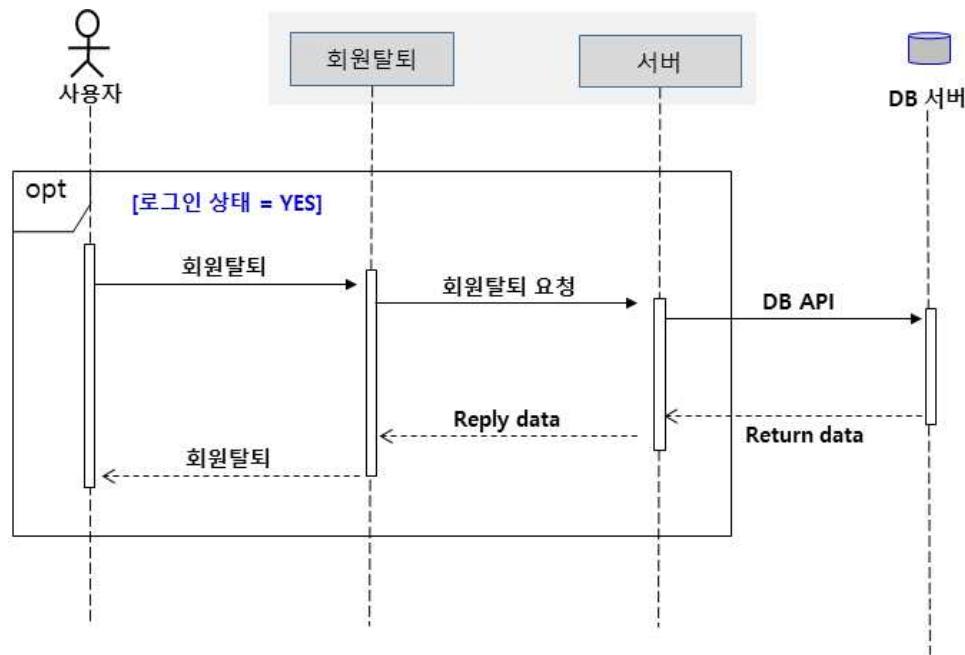


그림 3.6 회원탈퇴의 Sequence Diagram

그림 3.6을 보면 사용자는 로그인한 상태에서 회원탈퇴를 진행한다. 회원탈퇴 버튼을 누르면 회원탈퇴 요청이 서버로 올라가고 DB 서버에서 사용자를 삭제한다. 탈퇴된 회원은 서비스를 사용할 수 없고 탈퇴된 결과를 사용자에게 보낸다.

6) 챗봇기능 - 내 정보 조회

사용자는 챗봇에서 자신의 정보를 조회할 수 있다. 내 정보 조회 기능을 구현하기 위해 아래와 같이 설계한다.

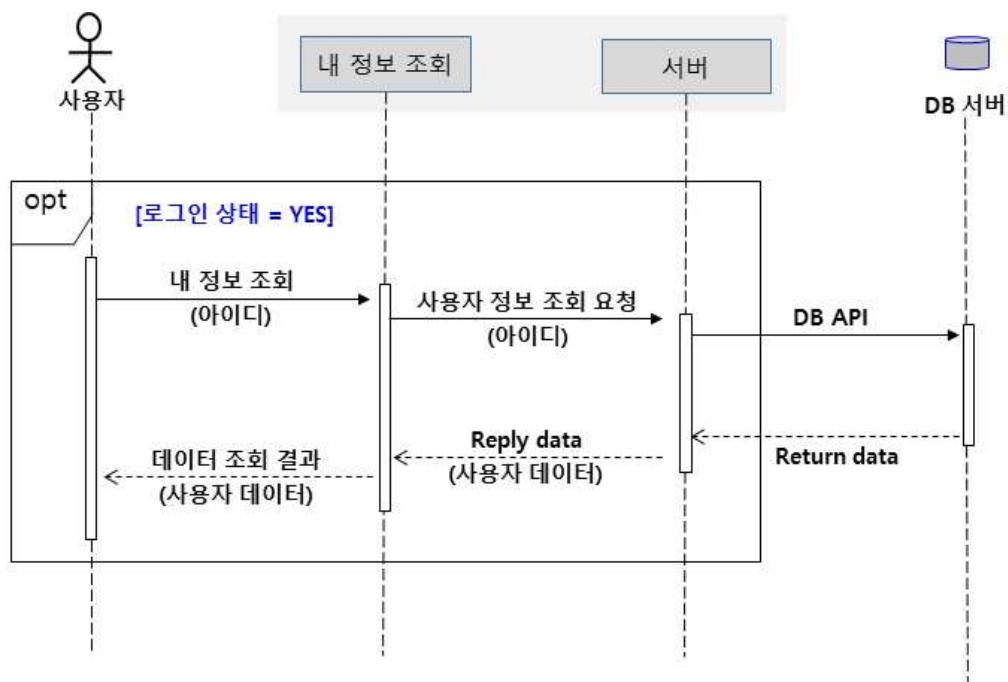


그림 3.7 내 정보 조회의 Sequence Diagram

그림 3.7과 같이 사용자는 로그인 상태에서 자신의 정보를 조회할 수 있다. 챗봇 UI로 내 정보 조회를 요구하면 서버로 사용자 정보 조회 요청을 보낸다. 서버에서 해당 사용자의 정보를 DB에서 조회해 돌려주면 사용자는 챗봇 UI를 통해 자신의 정보를 조회한다.

7) 챗봇기능 - 판매상품 조회

사용자는 챗봇에서 간편하게 대화하는 방식으로 판매상품을 조회할 수 있다. 판매 상품 조회 기능을 구현하기 위해 아래와 같이 설계한다.

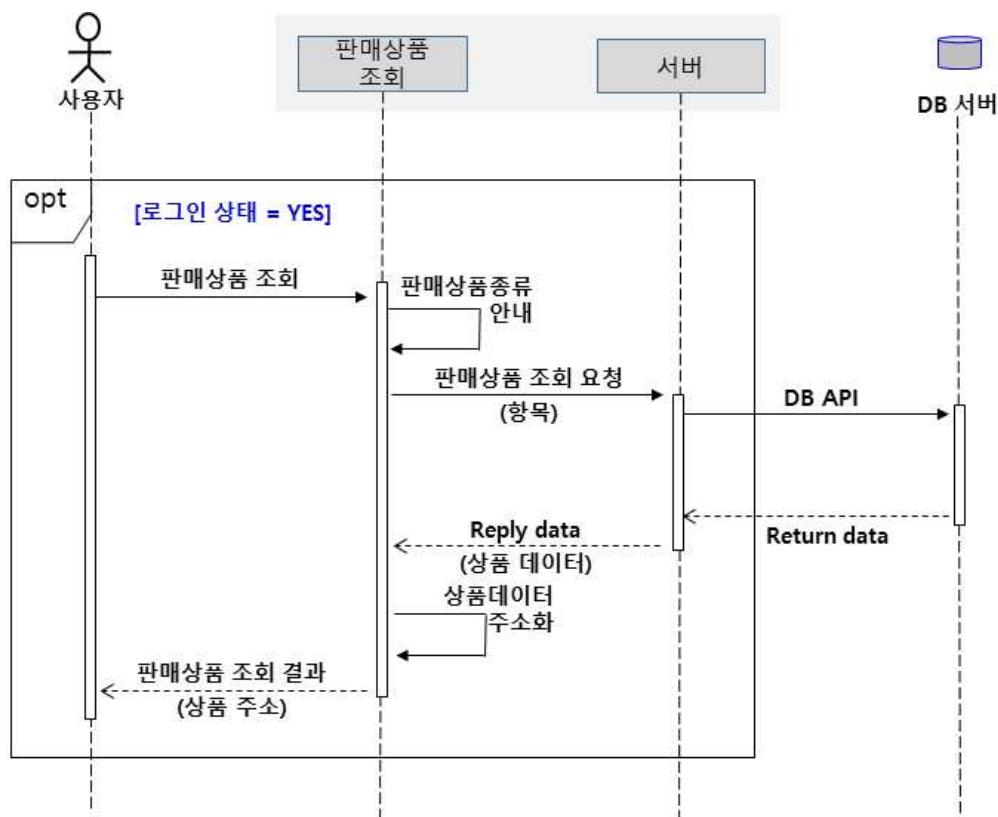


그림 3.8 내 정보 조회의 Sequence Diagram

그림 3.8과 같이 사용자는 로그인한 다음 챗봇으로 판매상품을 조회할 수 있다. 사용자는 상품 종목을 선택하여 물품을 조회할 수 있다. 사용자가 선택한 종류의 상품을 DB에서 검색하여 URL형식으로 사용자에게 전달한다.

8) 챗봇기능 - 미니퀴즈

사용자는 챗봇에서 미니퀴즈 이벤트에 참여할 수 있다. 미니퀴즈 기능을 구현하기 위해 아래와 같이 설계한다.

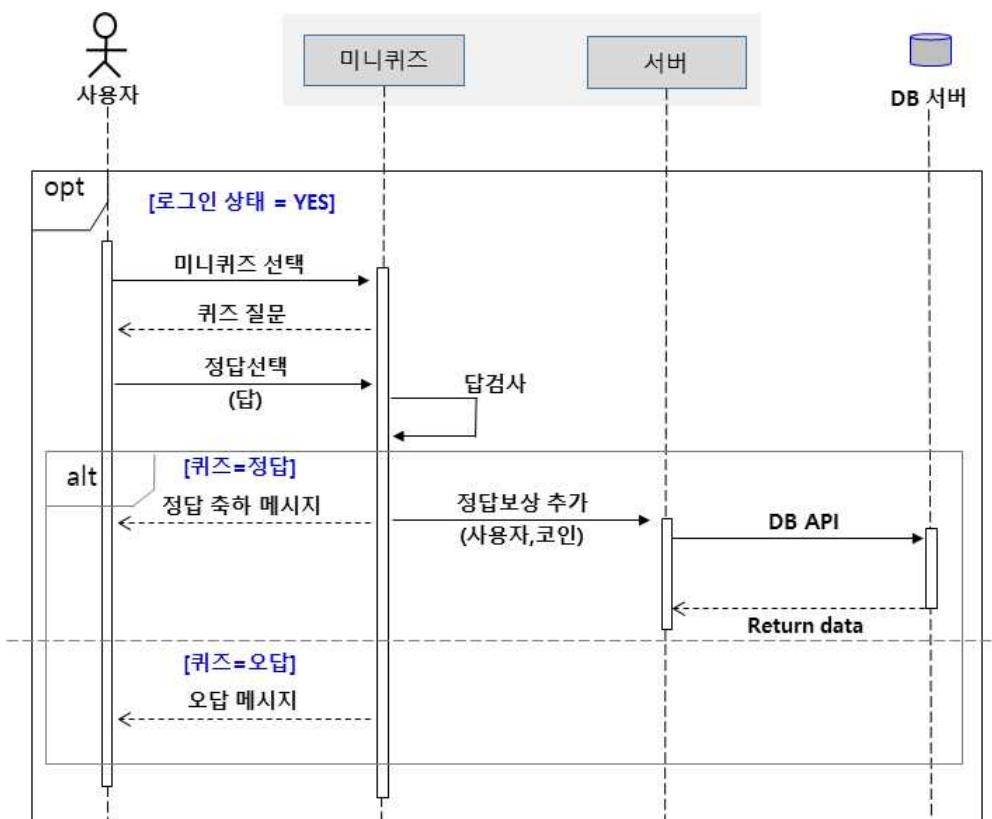


그림 3.9 미니퀴즈의 Sequence Diagram

그림 3.9과 같이 사용자는 로그인한 다음 챗봇에서 미니퀴즈 이벤트를 실행할 수 있다. 미니퀴즈 질문을 받으면 사용자는 답을 버튼으로 고를 수 있고 고른 답의 결과에 따라 보상 코인을 추가하거나 다른 메시지를 받는다. 퀴즈는 관리자가 수정할 수 있다.

9) 챗봇기능 - FAQ

사용자는 챗봇에서 간편하게 대화하는 방식으로 FAQ를 이용할 수 있다. FAQ 기능을 구현하기 위해 아래와 같이 설계한다.

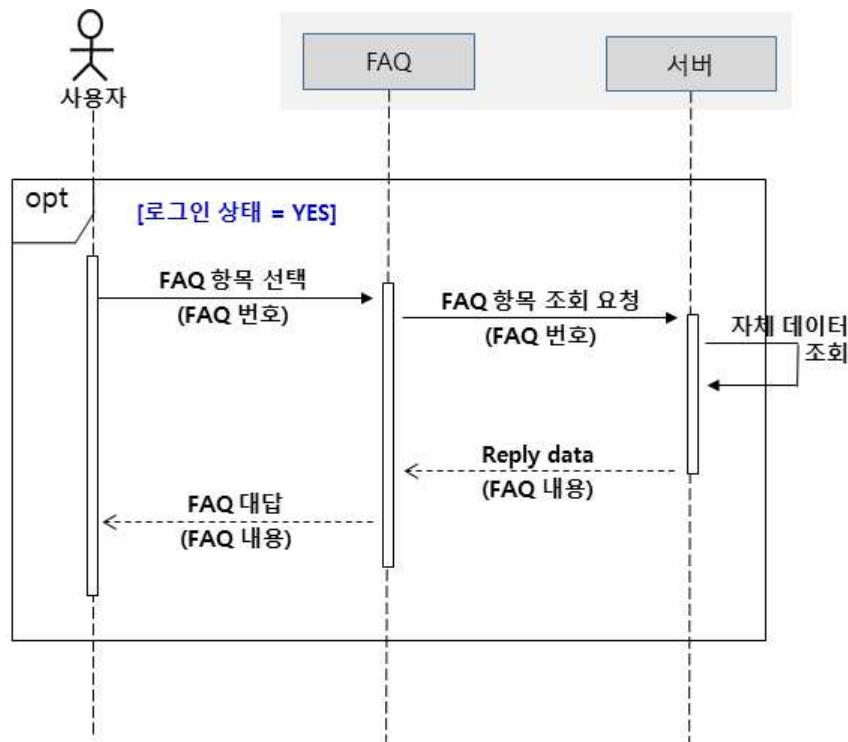


그림 3.10 FAQ 기능의 Sequence Diagram

그림 3.10과 같이 사용자가 로그인한 다음 챗봇 UI에서 FAQ 항목을 선택한다. 선택된 FAQ 번호는 서버로 넘어가서 데이터를 조회하고 해당하는 결과를 다시 반환한다. 데이터를 챗봇 UI로 처리하여 사용자는 간편하게 FAQ 서비스를 이용한다.

10) 중고거래(구매) - 코인충전

사용자는 서비스를 이용하기 위해 쇼핑몰 재화인 코인(coin)을 충전해야 한다. 코인 충전을 위해 아래와 같이 기능을 설계한다.

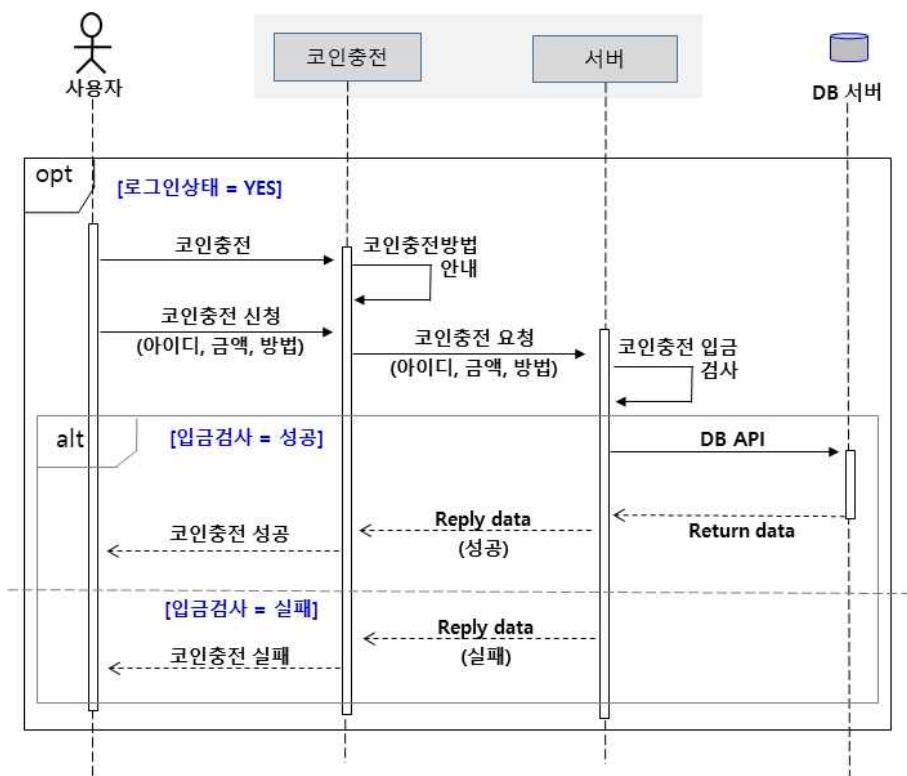


그림 3.11 코인충전의 Sequence Diagram

그림 3.11에서 사용자는 웹화면으로 코인 충전에 대한 안내를 받는다. 사용자가 코인충전을 선택하여 아이디, 금액이 서버로 넘어간다. 서버에서는 코인충전 방식에 맞게 입금을 검사하고 결과에 따라 사용자가 코인충전에 성공한다. 입금되지 않았을 경우에는 코인이 충전되지 않고 결과를 사용자에게 안내한다. 사용자는 코인을 이용해 Hidden100의 서비스를 이용한다.

11) 중고거래(구매) - 코인응모

사용자는 Hidden100 서비스는 상품에 코인을 응모하여 추첨하는 랜덤경매의 형태이다. 코인적립을 위해 아래와 같이 기능을 설계한다.

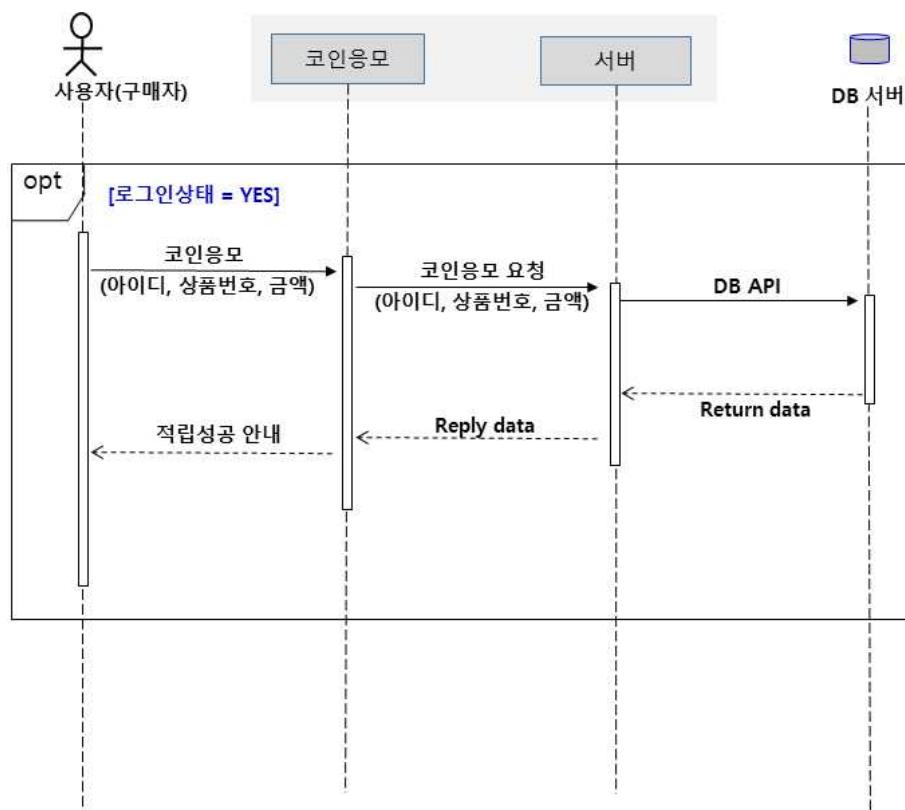


그림 3.12 코인응모의 Sequence Diagram

그림 3.12를 보면 사용자가 코인을 적립하면 적립내역이 DB에 저장된다. 사용자는 Web UI를 통해 코인을 적립한다. 상품에 응모버튼을 눌러 금액을 선택하고 아이디, 상품번호 정보를 서버에 같이 전달하여 코인을 적립한다. 적립한 참여자는 DB에 저장되고 이후 추첨을 통해 중고 상품을 받을 수 있다.

12) 중고거래(구매) - 판매상품 조회

판매상품 조회는 로그인 없이도 가능한 기능으로 사용자는 상품을 검색하거나 조회할 수 있다. 판매상품 조회 기능은 아래와 같이 설계한다.

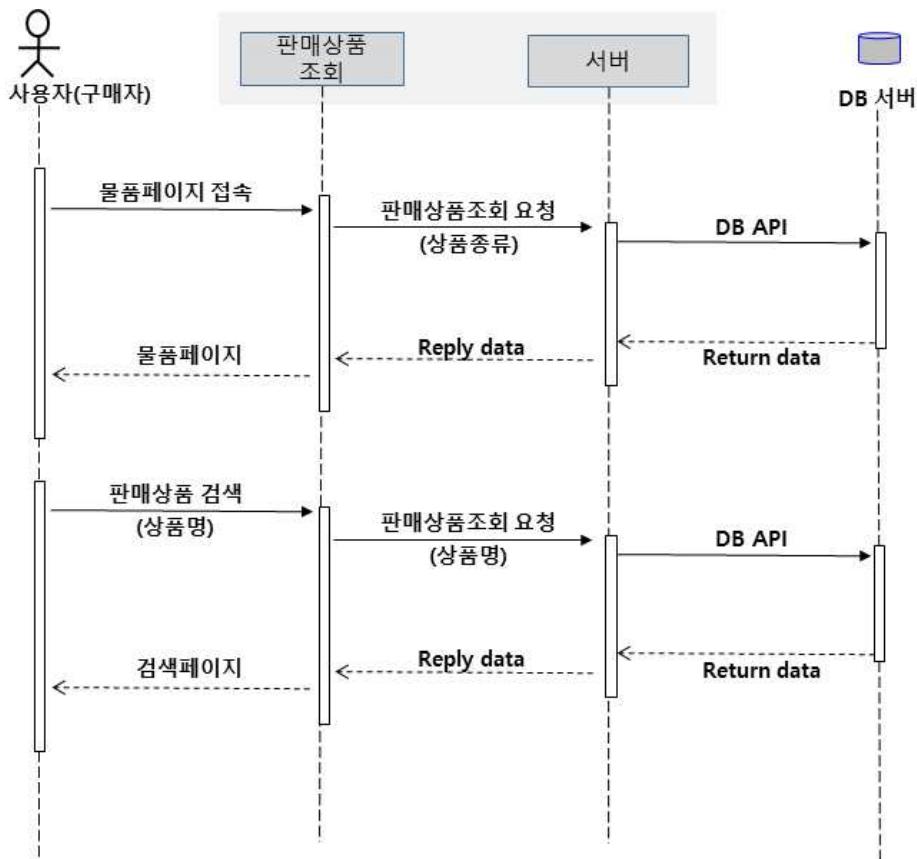


그림 3.13 판매상품 조회의 Sequence Diagram

그림 3.13과 같이 판매상품 조회는 웹페이지에 접속하거나 특정 버튼을 눌렀을 경우 상품에 대한 조회 기능을 수행한다. 상품 종류에 대해 요청하면 해당하는 상품의 정보를 가져와 웹페이지에 출력한다. 혹은 상품 이름, 키워드로 검색하면 해당하는 상품을 DB에 검색하여 결과를 웹페이지에 출력하여 사용자가 상품을 조회한다. 검색과 조회는 사용자 인증이 없어도 사용 가능하다.

13) 중고거래(구매) - 응모내역 조회

사용자는 자신이 참여한 응모내역을 조회한다. 응모내역 조회 기능은 아래와 같이 설계한다.

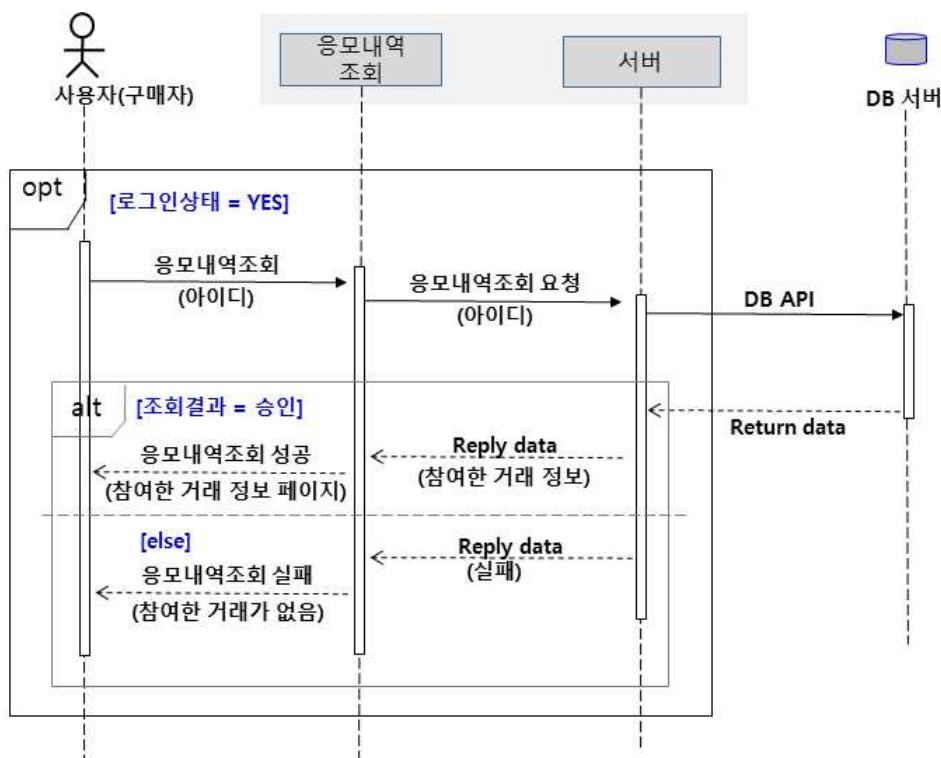


그림 3.14 응모내역 조회의 Sequence Diagram

그림 3.14과 같이 사용자가 로그인한 상태에서 응모내역 조회를 요청하면 서버로 사용자의 아이디를 전달하고, DB에서 해당 아이디가 참여한 응모내역을 가져온다. 받은 정보는 시각화 기능을 이용하여 웹페이지 형태로 사용자에게 전달되고 참여한 거래 정보가 없으면 실패로 처리하여 조회할 정보가 없다고 안내한다.

14) 중고거래(구매) - 당첨내역 조회

사용자는 자신이 참여한 당첨내역을 조회한다. 당첨내역 조회 기능은 아래와 같이 설계한다.

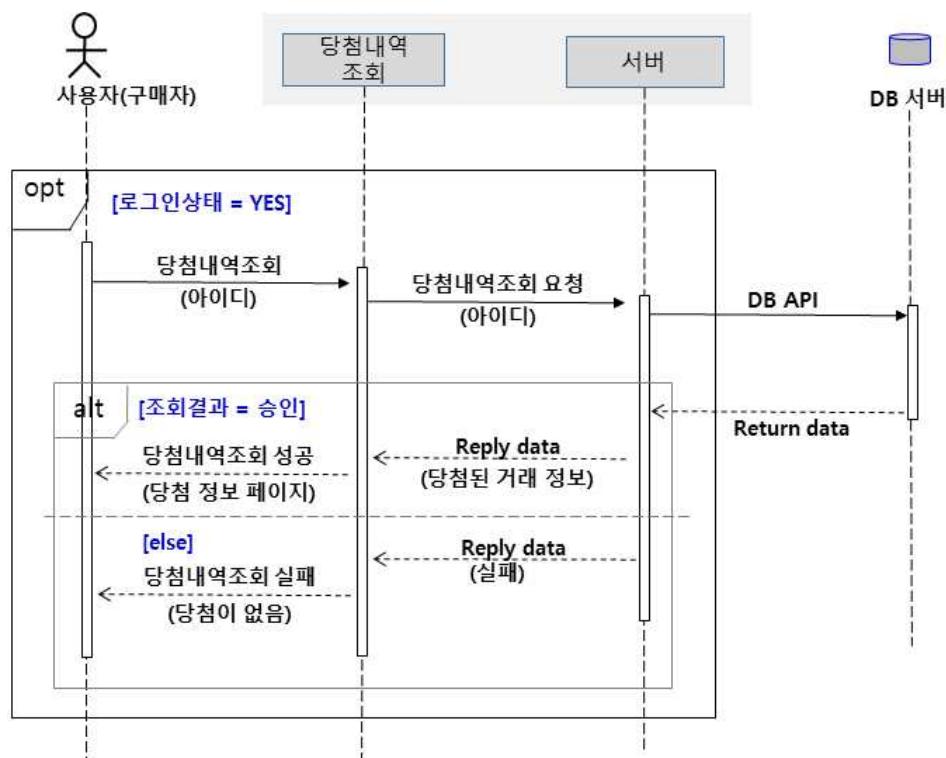


그림 3.15 당첨내역 조회의 Sequence Diagram

그림 3.15와 같이 사용자가 로그인한 상태에서 응모내역 조회를 요청하면 서버로 사용자의 아이디를 전달하고, DB에서 해당 아이디가 참여한 응모내역을 가져온다. 받은 정보는 시각화 기능을 이용하여 웹페이지 형태로 사용자에게 전달되고 참여한 거래 정보가 없으면 실패로 처리하여 조회할 정보가 없다고 안내한다.

15) 중고거래(판매) - 판매물품 등록

중고 물건을 판매하고 싶은 사용자는 판매물품 등록을 이용해 판매할 수 있다. Hidden100에서는 중고거래 판매물품 등록을 위해 다음과 같이 설계한다.

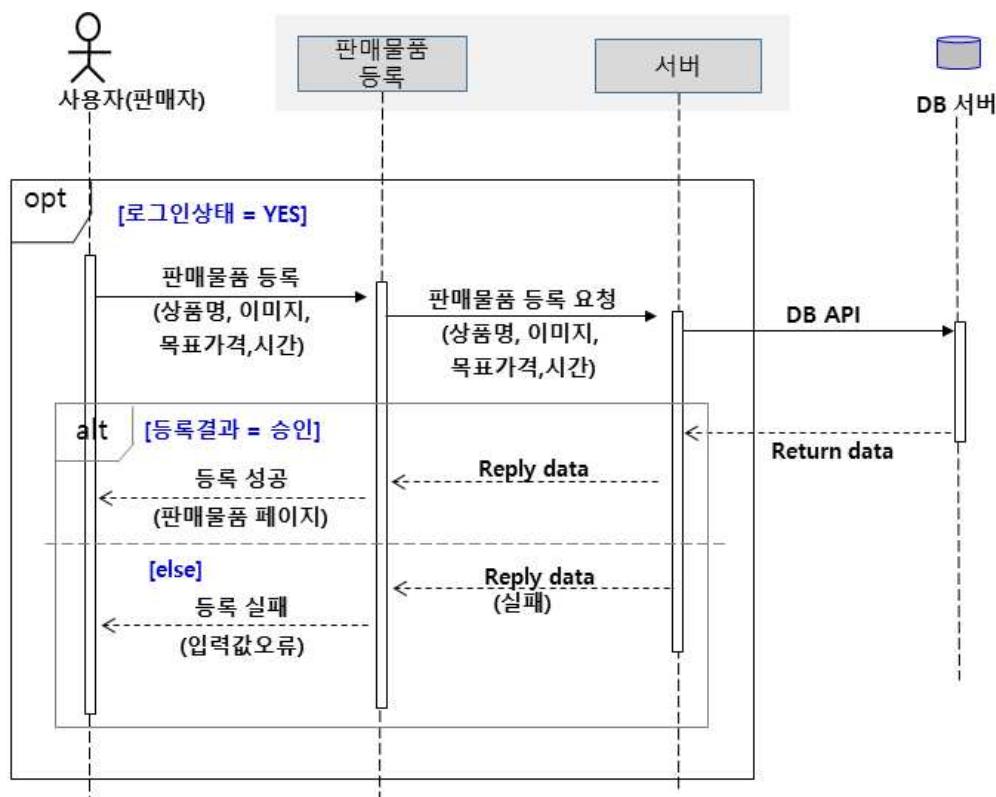


그림 3.16 판매물품 등록의 Sequence Diagram

Hidden100은 사용자가 판매자와 구매자의 역할을 수행하는 중고거래 사이트이며 사용자가 로그인만 하면 상품을 판매할 수 있다. 그림 3.16을 보면 사용자는 로그인한 상태에서 물품등록을 할 수 있다. 물품등록을 위해 판매할 물품의 상품명, 이미지, 목표가격, 시간 등을 설정한다. 사용자가 Web UI를 통해 입력한 정보는 서버로 전달하고 DB에 저장된다. 입력 값이 비어있거나 오류가 있는 경우에는 상품등록이 거절된다. 물품 등록에 성공하면 중고 물품은 판매자가 설정한 시간 동안 적립을 시작한다. 저장 결과에 따라 사용자에게 등록결과를 안내한다.

16) 중고거래(판매) - 판매물품 조회

Hidden100은 적립금을 비밀로 진행하는 소액적립형 경매 서비스이기 때문에 일반적으로는 상품의 적립금을 확인할 수 없으나 자신이 등록한 상품은 적립금을 확인할 수 있다. 따라서 판매자의 판매물품 조회를 다음과 같이 설계한다.

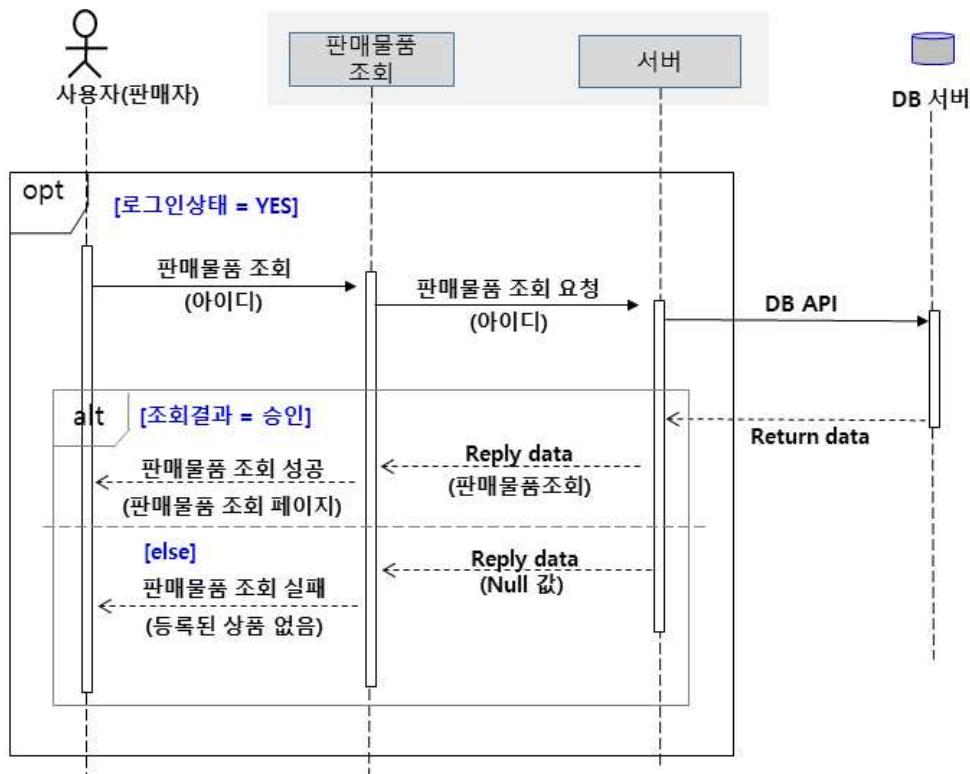


그림 3.17 판매물품 조회의 Sequence Diagram

판매자의 판매물품 조회는 로그인 상태에서 진행된다. 그림 3.17을 보면 사용자가 페이지에 들어가 물품조회가 발생하면 서버로 판매물품 조회 요청이 전송된다. DB에서 사용자 아이디를 가지고 검색해 상품이 있으면 해당하는 상품에 대한 정보를 사용자에게 전달한다. DB에 저장된 정보는 웹 화면에서 사용자가 열람하기 편하게 시각화되어 보여준다. 만약 사용자가 등록한 상품이 없어 상품 조회에 실패한 경우 등록된 상품이 없다는 안내를 사용자에게 보낸다.

17) 중고거래(판매) - 판매물품 취소

판매자가 부득이한 사정으로 중고 물품의 판매를 취소할 수 있다. 판매물품 취소를 구현하기 위해 다음과 같이 설계한다.

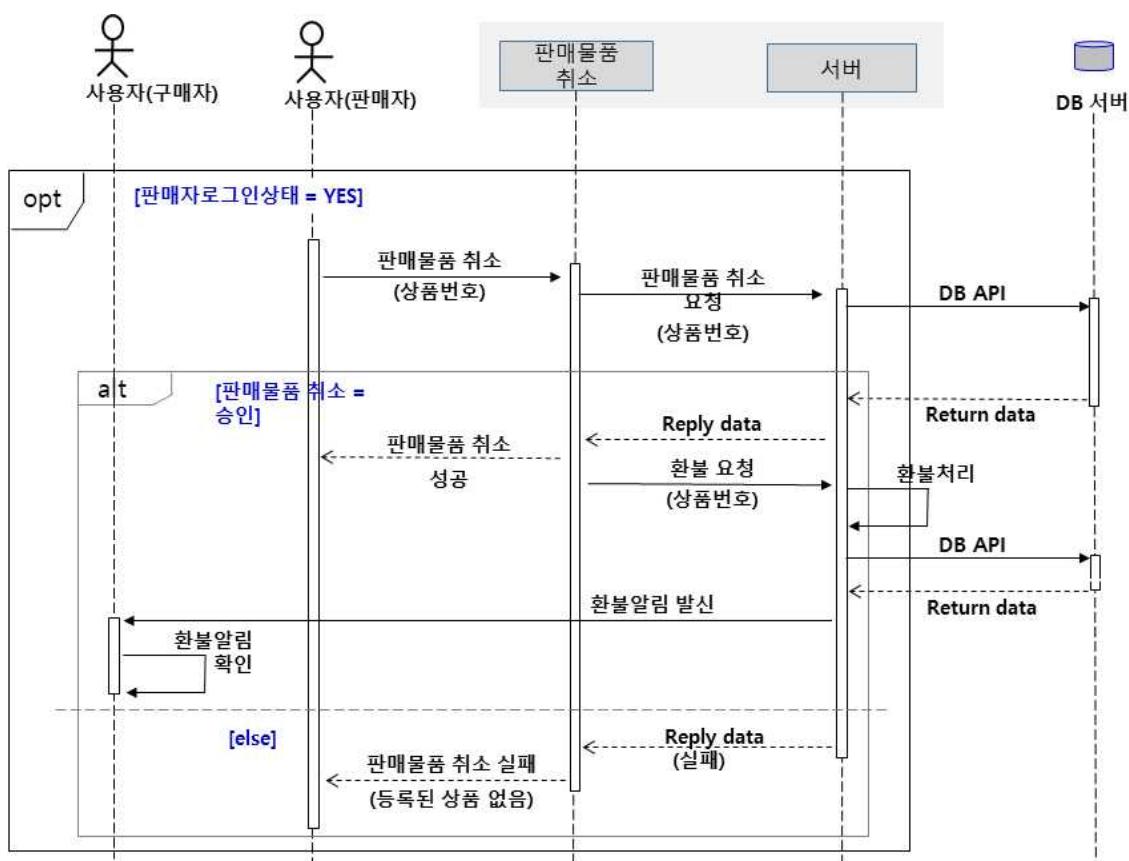


그림 3.18 판매물품 취소의 Sequence Diagram

그림 3.18와 같이 사용자(판매자)는 로그인한 상태에서 자신의 상품을 판매물품 취소 요청할 수 있다. 판매자가 판매물품 취소 요청을 보낼 때 상품 번호를 포함하여 서버에 전달한다. 서버는 DB에서 상품을 조회하고 삭제한다. DB에서 상품을 찾지 못 한 경우에는 판매물품 취소 실패를 판매자에게 보낸다. 이미 취소한 상품에 적립한 사람이 있는 경우에 환불처리를 수행하고 환불에 대해 사용자에게 푸시알림을 보내 안내한다.

18) 중고거래(판매) - 배송정보 등록

중고거래 적립이 끝난 후에 당첨자에게 물건을 보내고 사용자는 배송정보에 대해 입력한다. 배송정보 등록을 구현하기 위해 다음과 같이 설계한다.

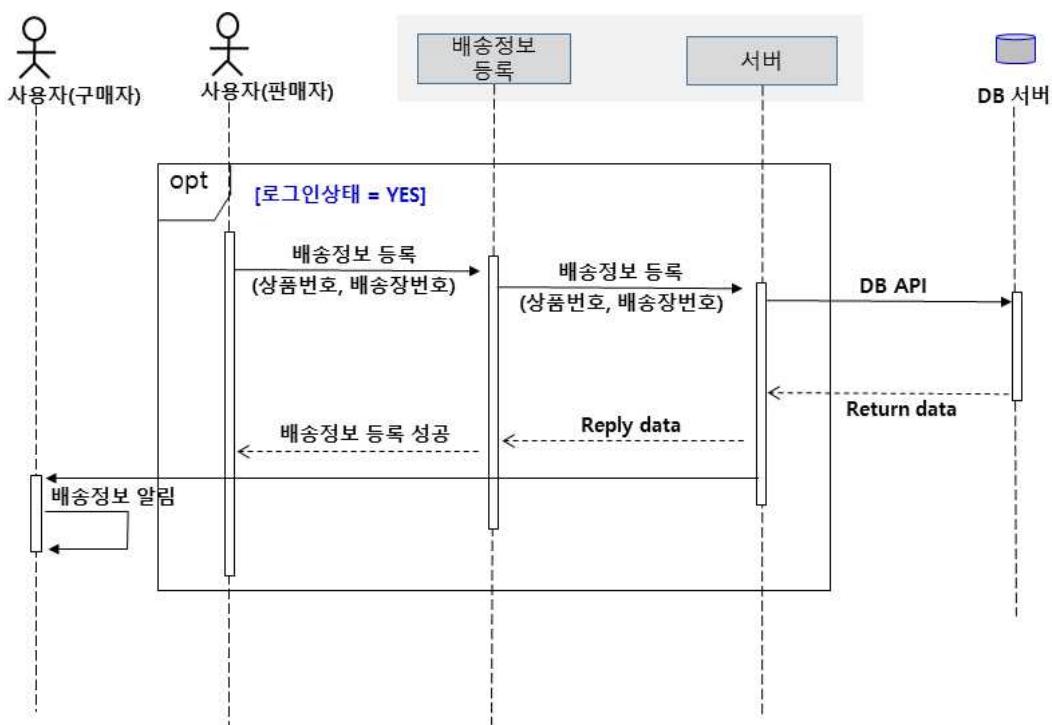


그림 3.19 배송정보 입력의 Sequence Diagram

그림 3.19를 보면 사용자(판매자)는 로그인한 상태에서 자신의 상품에 배송정보를 입력하여 사용자(구매자)에게 전달한다. 판매자가 입력한 정보는 서버로 넘어가고 서버에서는 DB를 조회해서 당첨자를 찾아 배송 정보를 알림으로 전달한다.

19) 게시판 모듈 - 게시글 등록

사용자는 게시글을 작성할 수 있어야 한다. 게시글 등록 기능을 구현하기 위해 아래와 같이 설계한다.

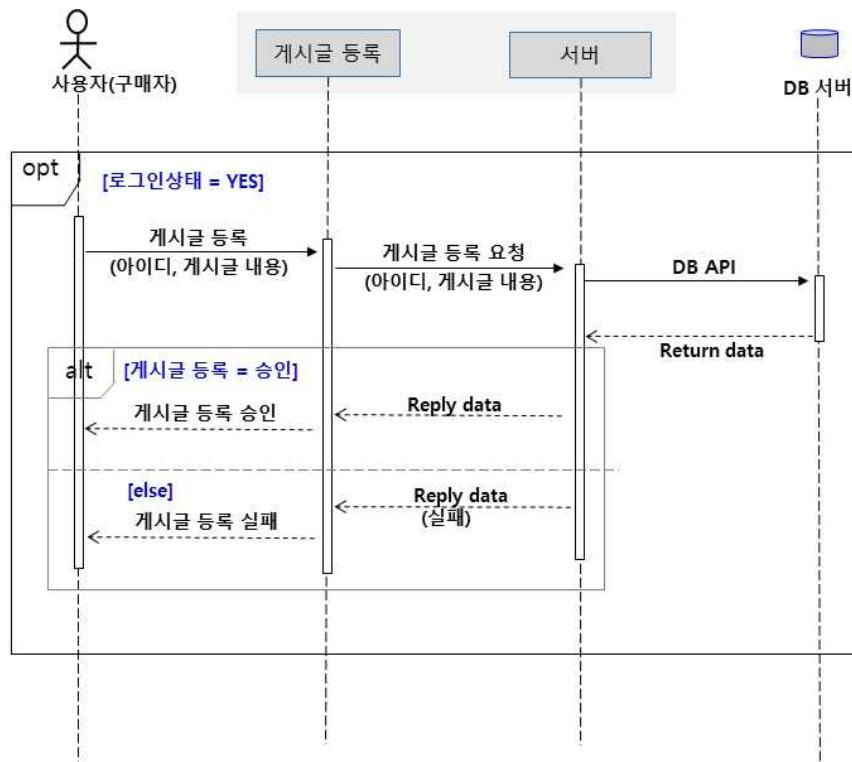


그림 3.20 게시글 등록의 Sequence Diagram

그림 3.20과 같이 사용자는 게시글을 등록하기 위해서 사용자는 로그인해야 한다. 로그인한 사용자가 게시글 내용을 작성하여 등록버튼을 누르면 서버로 정보를 전송하여 DB에 저장한다. 내용이 비어있거나 오류가 있는 경우에는 게시글 등록 실패로 연결하며 게시글 등록에 성공한 경우 성공 메시지를 전달한다.

20) 게시판 모듈 - 게시글 수정

사용자는 작성한 게시글의 내용을 수정할 수 있다. 게시글 수정 기능을 구현하기 위해 아래와 같이 설계한다.

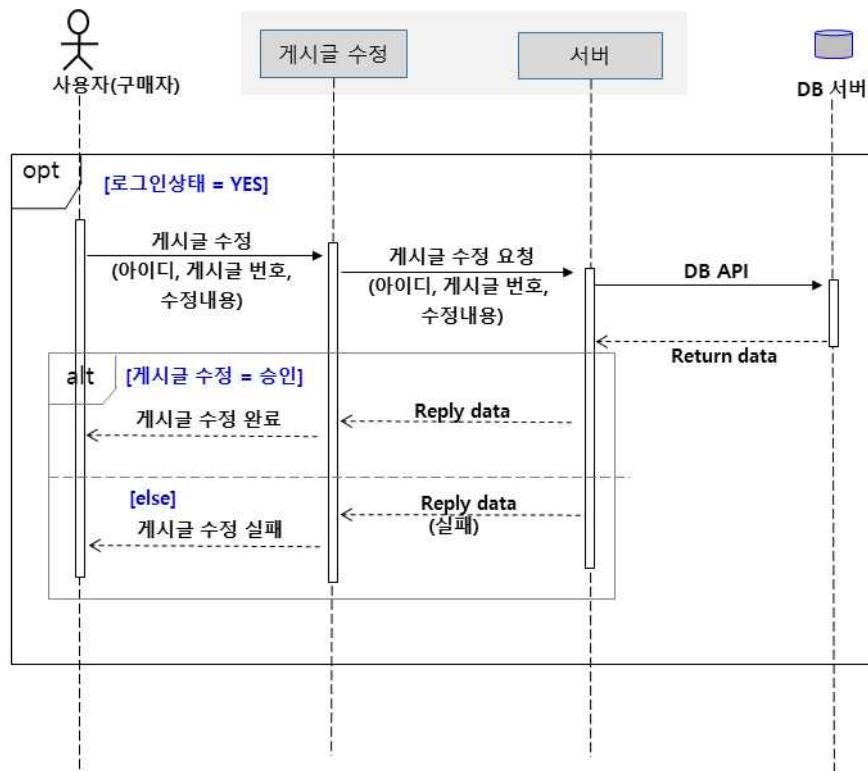


그림 3.21 게시글 수정의 Sequence Diagram

그림 3.21과 같이 사용자는 게시글을 수정하기 위해서 사용자는 로그인해야 한다. 로그인한 사용자가 자신이 작성한 게시글의 수정 버튼을 누르면 게시글을 수정할 수 있다. 새로운 게시글 내용을 작성하여 게시글을 저장하면 변경된 게시글 내용과 게시글 번호, 아이디가 서버로 전달된다. 서버는 해당하는 게시글의 내용을 DB에서 변경하고 사용자에게 수정 결과를 안내한다.

21) 게시판 모듈 - 게시글 삭제

사용자는 작성한 게시글의 내용을 삭제할 수 있다. 게시글 삭제 기능을 구현하기 위해 아래와 같이 설계한다.

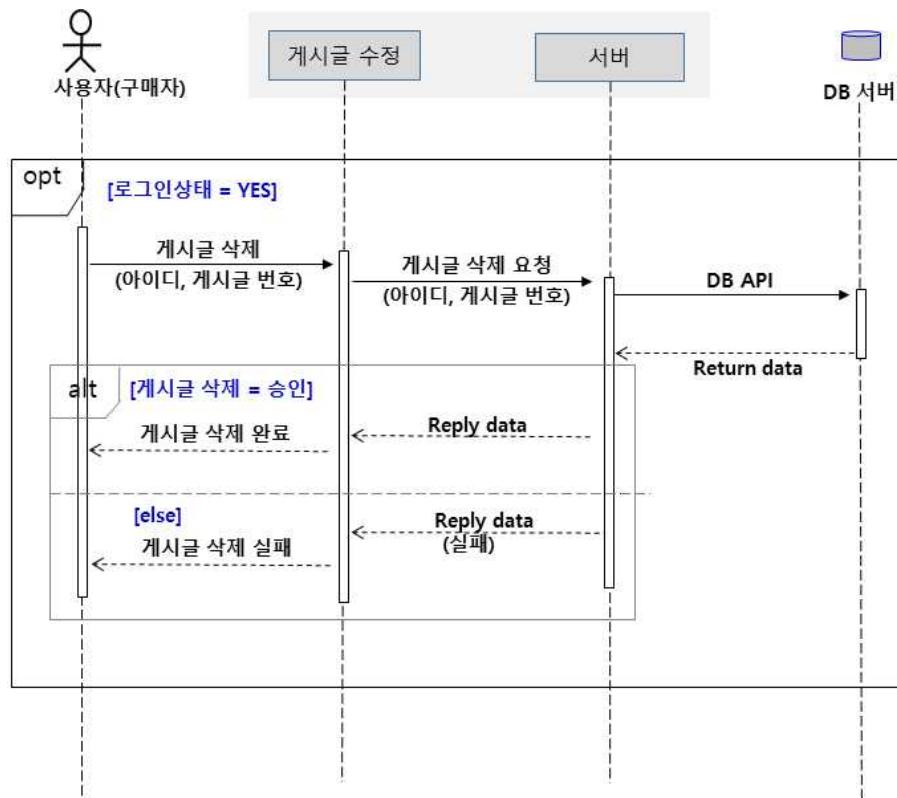


그림 3.22 게시글 삭제의 Sequence Diagram

그림 3.22과 같이 사용자는 자신의 게시글을 삭제하기 위해서 로그인해야 한다. 사용자가 자신의 게시물에 삭제 버튼을 누르면 아이디와 게시글 번호를 전달하여 삭제를 요청한다. DB에서 해당 게시글 데이터를 삭제하고 서버로 결과를 반환하며, 서버는 사용자에게 반환 결과를 안내한다.

22) 게시판 모듈 - 게시글 조회

사용자가 작성한 게시글을 조회할 수 있다. 문의게시판과 신고게시판은 댓글 기능을 포함하고 있어 댓글 정보도 같이 조회한다. 게시글 조회 기능을 구현하기 위해 아래와 같이 설계한다.

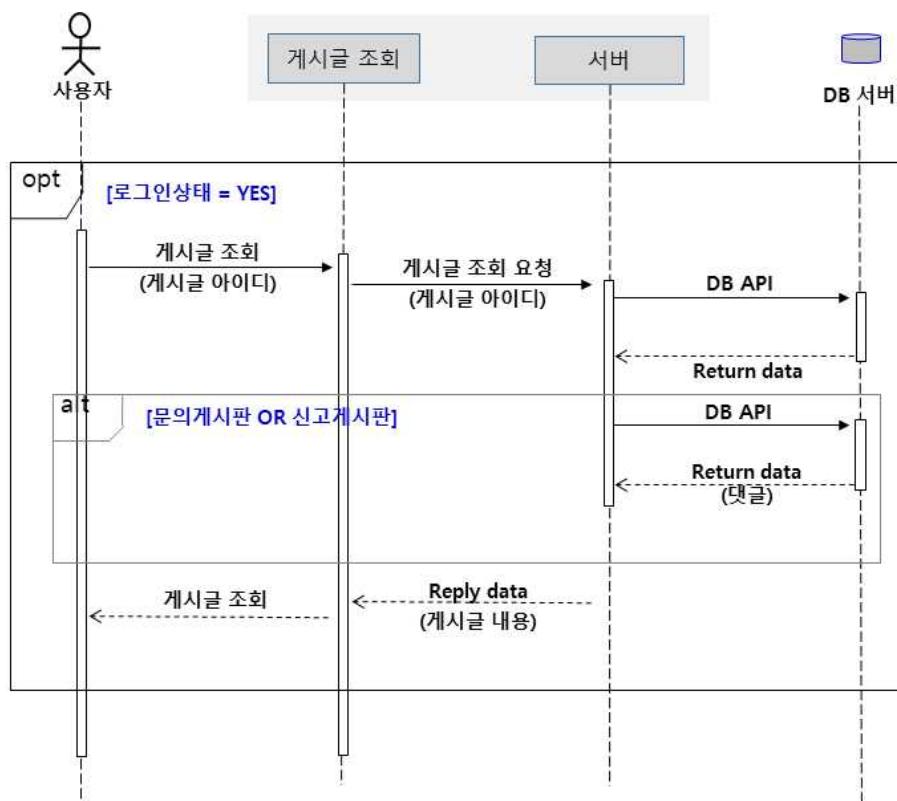


그림 3.23 게시글 조회의 Sequence Diagram

그림 3.23과 같이 사용자는 로그인 상태에서 다른 사용자의 게시글을 조회할 수 있다. 웹페이지에서 게시글을 조회하면 해당 게시글에 대한 아이디로 게시글 정보를 서버에 요청하고 서버는 DB에서 해당 게시글의 내용을 받아 전달한다. 문의게시판과 신고게시판의 경우 댓글 정보도 같이 조회한다. 사용자는 전달된 데이터를 웹페이지 형태로 조회한다.

3.2. 관리자기능 설계 (Sequence Diagram)

1) 관리자 인증 - 관리자가입

관리자는 인증키를 이용하여 회원가입을 진행할 수 있다. 관리자의 회원가입 기능을 구현하기 위해 아래와 같이 설계한다.

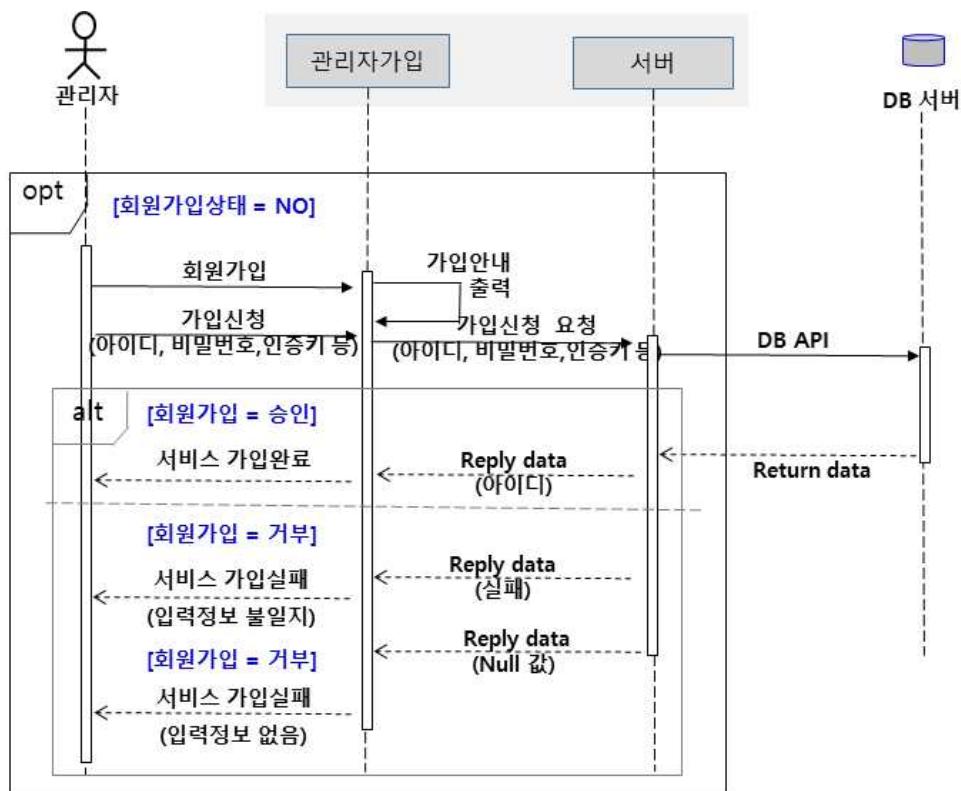


그림 3.24 관리자가입의 Sequence Diagram

그림 3.24를 보면 관리자가 회원가입을 신청하면 가입정보를 입력할 수 있는 페이지로 연결한다. 기존에 등록된 인증키와 관리자가 입력한 인증키가 일치하는 경우에 회원가입에 성공하며 정보가 일치하지 않거나 비어있을 경우 회원가입에 거부된다. 회원가입에 성공한 관리자는 관리자 인증을 통해 관리자 기능을 수행할 수 있다. 가입 결과에 대한 메시지를 관리자에게 전달한다.

2) 관리자 인증 - 관리자 로그인

관리자는 회원가입 이후 로그인을 통해 관리자 인증을 하고 관리자 기능을 사용할 수 있다. 관리자의 로그인 기능을 구현하기 위해 아래와 같이 설계한다.

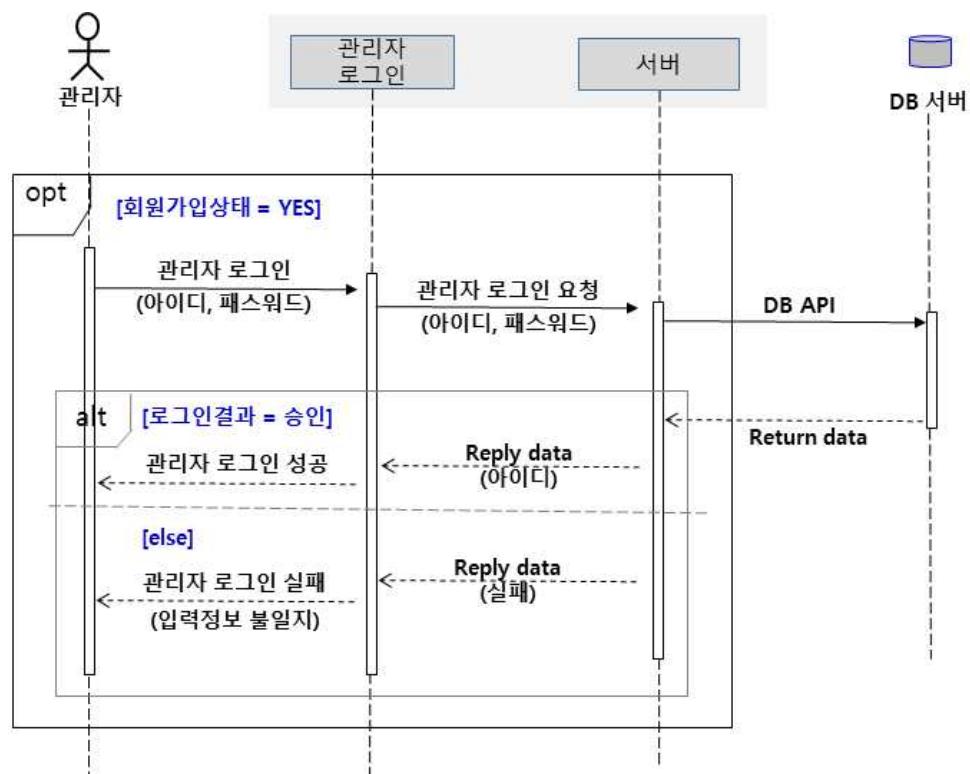


그림 3.25 관리자 로그인의 Sequence Diagram

그림 3.25를 보면 관리자가 아이디와 패스워드를 입력하여 로그인할 수 있다. 서버에서는 로그인 요청을 받아 DB에서 아이디를 키로 관리자 정보를 받아와 입력받은 정보와 비교한다. 아이디와 패스워드가 DB에 저장된 관리자 정보와 일치하면 로그인에 성공한다. 로그인하고 나서 다른 Hidden100 사용자 서비스를 이용할 수 있다. 패스워드가 다르거나 정지계정인 경우 로그인에 실패한다. 로그인에 실패한 경우 실패 메시지와 실패 사유를 안내한다.

3) 관리자 인증 - 관리자 로그아웃

관리자는 관리자 기능의 사용을 종료할 때 로그아웃으로 관리자 로그인 데이터를 삭제한다. 관리자의 로그아웃 기능을 구현하기 위해 아래와 같이 설계한다.

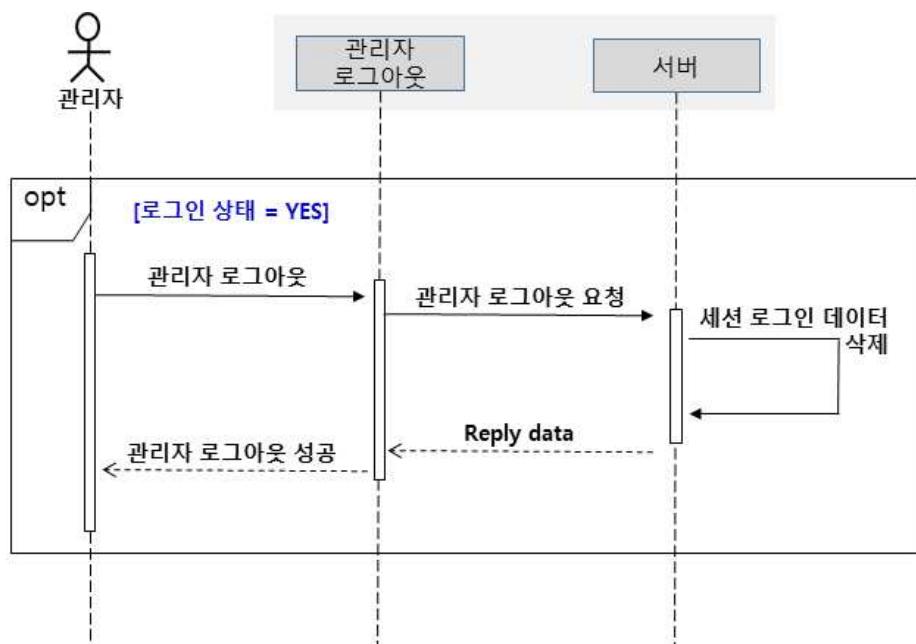


그림 3.26 관리자 로그아웃의 Sequence Diagram

그림 3.26를 보면 관리자는 로그아웃을 수행하기 위해 로그아웃 버튼을 누른다. 로그아웃 기능은 서버로 가서 현재 세션에 저장된 관리자의 로그인 데이터를 삭제한다. 서버에서 로그아웃을 처리한 후에 사용자에게 로그아웃된 화면으로 전송하여 보여준다.

4) 관리자 인증 - 관리자탈퇴

관리자도 자신의 관리자 계정 정보를 삭제할 수 있다. 관리자탈퇴 기능을 구현하기 위해 아래와 같이 설계한다.

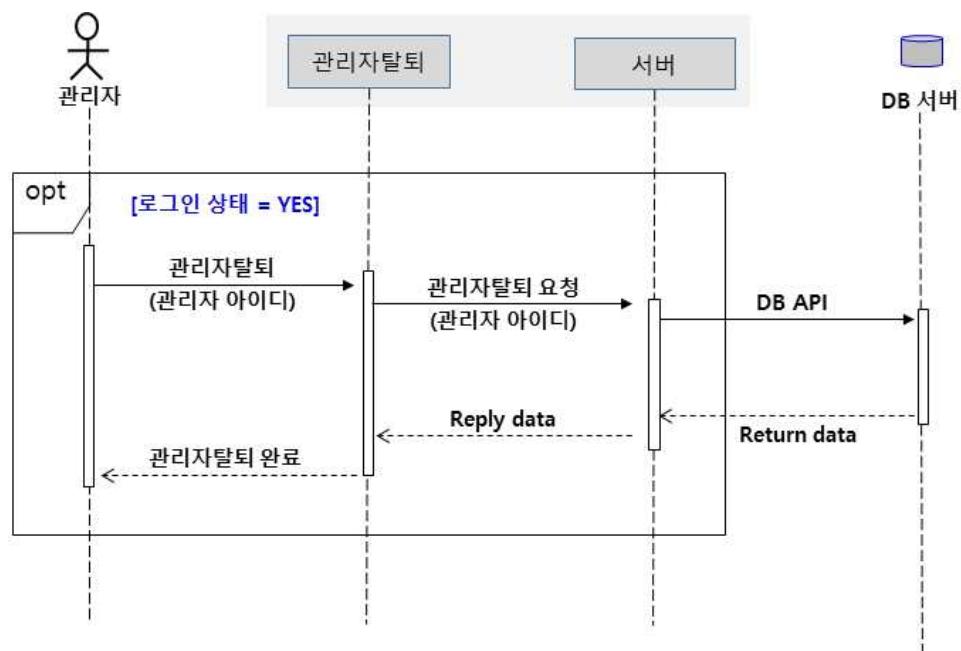


그림 3.27 관리자탈퇴의 Sequence Diagram

그림 3.27를 보면 관리자는 탈퇴를 수행하기 위해 탈퇴 버튼을 누른다. 관리자탈퇴 기능은 관리자의 아이디를 서버로 보내고 서버에서는 DB에 저장된 관리자 계정을 삭제한다. DB에서 데이터를 삭제하면 결과를 서버로 반환하고 서버에서는 관리자에게 탈퇴 안내를 한다.

5) 회원관리 - 회원조회

관리자는 관리자 인증을 마친 후에 회원관리 기능을 사용할 수 있다. 관리자의 회원조회 기능을 구현하기 위해 아래와 같이 설계한다.

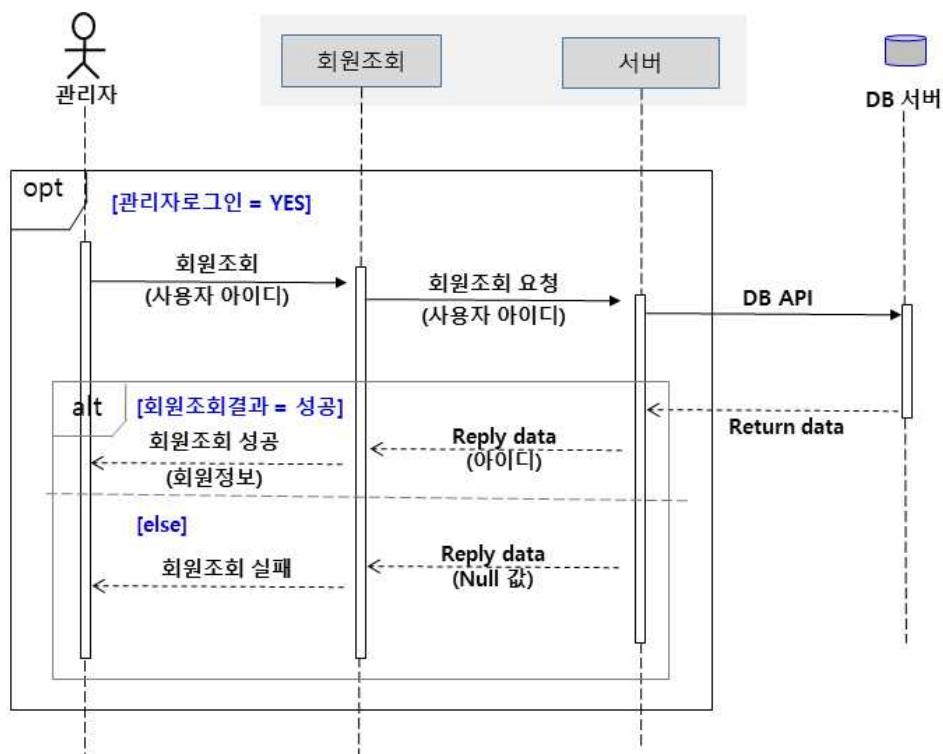


그림 3.28 회원조회의 Sequence Diagram

그림 3.28를 보면 관리자는 관리자 인증을 수행한 후에 사용자의 아이디로 회원 정보를 조회할 수 있다. 관리자가 사용자 아이디를 입력하면 사용자 아이디를 서버로 전달해 DB를 조회할 수 있도록 한다. DB에 조회에 성공하면 관리자에게 해당 사용자의 정보를 제공하고 DB에서 조회에 실패했을 경우 실패 메시지로 안내한다.

6) 회원관리 - 회원등급 관리

관리자는 회원의 정보를 관리할 수 있으며, 회원 등급을 변경할 수 있다. 관리자의 회원등급 관리 기능을 구현하기 위해 아래와 같이 설계한다.

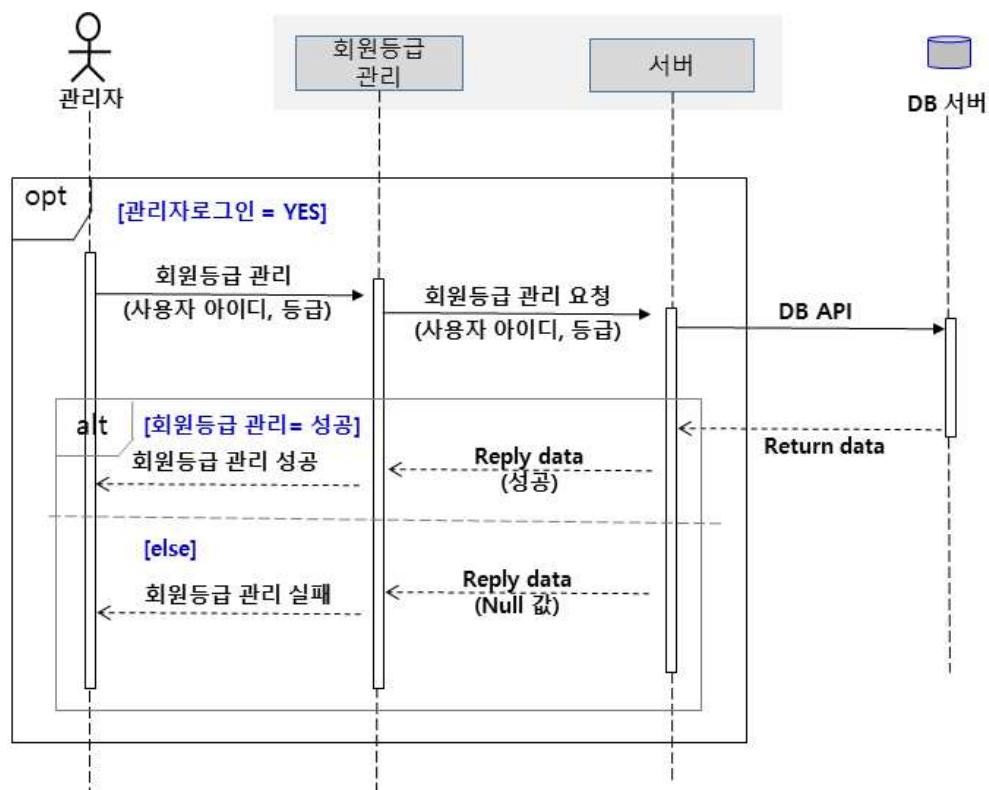


그림 3.29 회원등급 관리의 Sequence Diagram

그림 3.29를 보면 관리자는 관리자 인증을 성공한 후에 사용자의 등급을 변경할 수 있다. 변경하고자 하는 사용자의 아이디와 변경할 등급을 입력하면 해당 내용을 서버에 전달하고 서버에서 DB에 접근해 해당 사용자의 정보를 변경한다. 변경에 성공한 경우 관리자에게 성공 안내 메시지를 전달하고 실패한 경우 실패 안내 메시지를 전달한다.

7) 회원관리 - 회원설정 관리

관리자는 회원의 정보를 관리할 수 있으며, 회원설정을 변경할 수 있다. 관리자의 회원설정 관리 기능을 구현하기 위해 아래와 같이 설계한다.

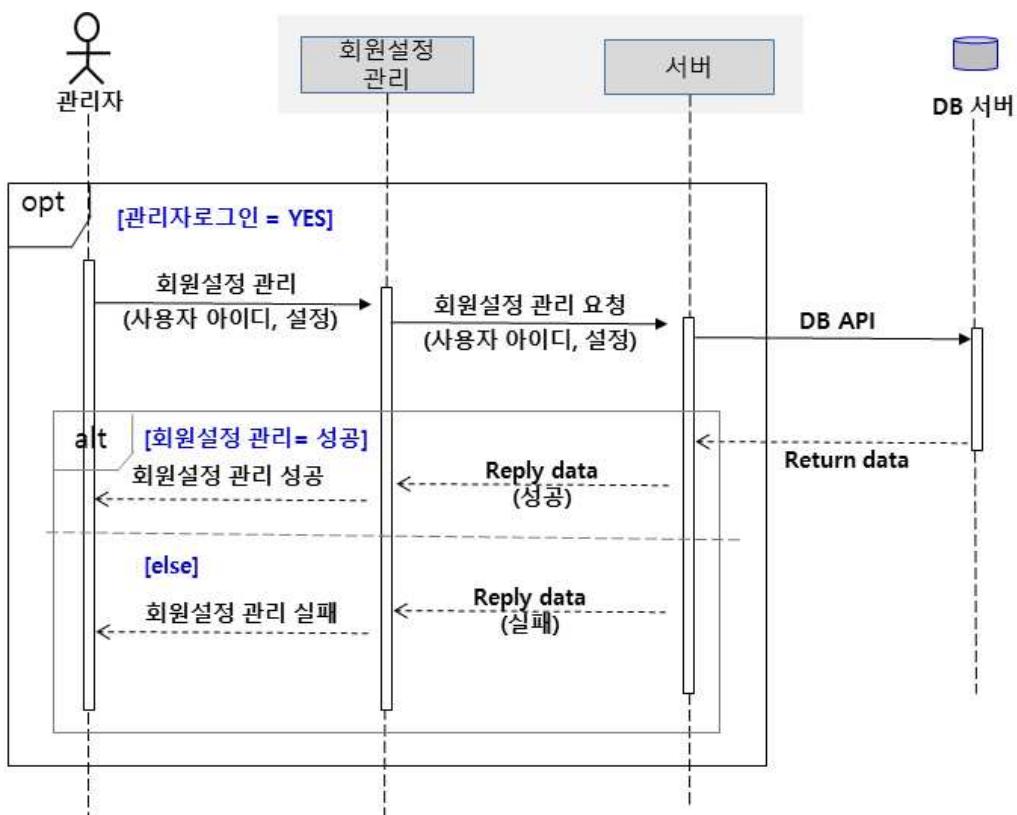


그림 3.30 회원설정 관리의 Sequence Diagram

그림 3.30를 보면 관리자는 관리자 인증을 성공한 후에 사용자의 설정을 변경할 수 있다. 변경하고자 하는 사용자의 아이디와 변경할 설정을 입력하면 해당 내용을 서버에 전달하고 서버에서 DB에 접근해 해당 사용자의 정보를 변경한다. 웹으로 사용자의 정보 데이터를 변경하며, 변경에 성공한 경우 관리자에게 성공 안내 메시지를 전달하고 실패한 경우 실패 안내 메시지를 전달한다.

8) 회원관리 - 회원정지 관리

관리자는 회원을 정지할 수 있으며 정지한 사용자는 로그인할 수 없다. 관리자의 회원정지 관리 기능을 구현하기 위해 아래와 같이 설계한다.

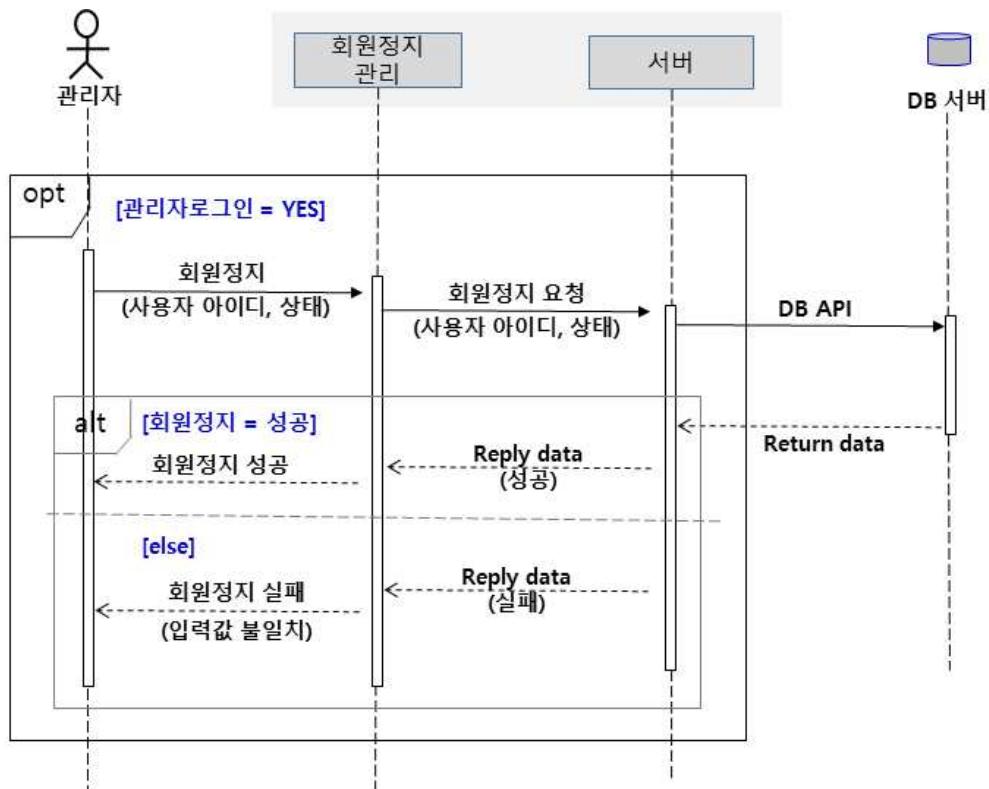


그림 3.31 회원정지 관리의 Sequence Diagram

그림 3.31를 보면 관리자는 관리자 인증을 성공한 후에 사용자를 정지시킬 수 있다. 정지할 사용자의 아이디를 입력하면 해당 내용을 서버에 전달하고 서버에서 DB에 접근해 해당 사용자를 정지회원으로 변경한다. 변경에 성공한 경우 관리자에게 성공 안내 메시지를 전달하고 실패한 경우 해당하는 실패 안내 메시지를 전달한다.

9) 챗봇기능 - 알림 발신

관리자는 사용자에게 알림을 보낼 수 있으며 알림 발신 기능을 구현하기 위해 아래와 같이 설계한다.

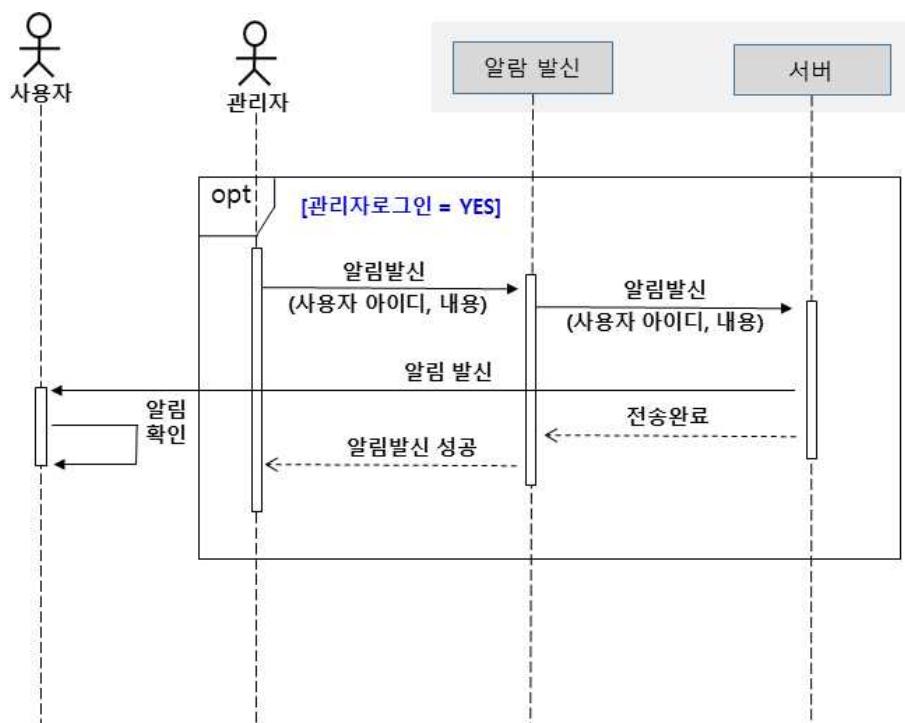


그림 3.32 알림 발신의 Sequence Diagram

그림 3.32를 보면 관리자는 관리자 인증을 성공한 후에 사용자에게 알림을 보낼 수 있다. 알림을 받을 사용자의 아이디와 보낼 내용을 작성하여 입력하면 서버로 해당 내용이 전송된다. 서버에서는 해당 사용자에게 내용을 알림으로 보내고 전송 완료 메시지를 다시 관리자에게 보낸다. 관리자는 사용자 아이디를 이용하여 사용자에게 알림을 발신할 수 있다.

10) 챗봇기능 - FAQ 관리

관리자는 사용자가 챗봇으로 확인하는 FAQ 내용을 관리할 수 있다. FAQ 관리 기능을 구현하기 위해 아래와 같이 설계한다.

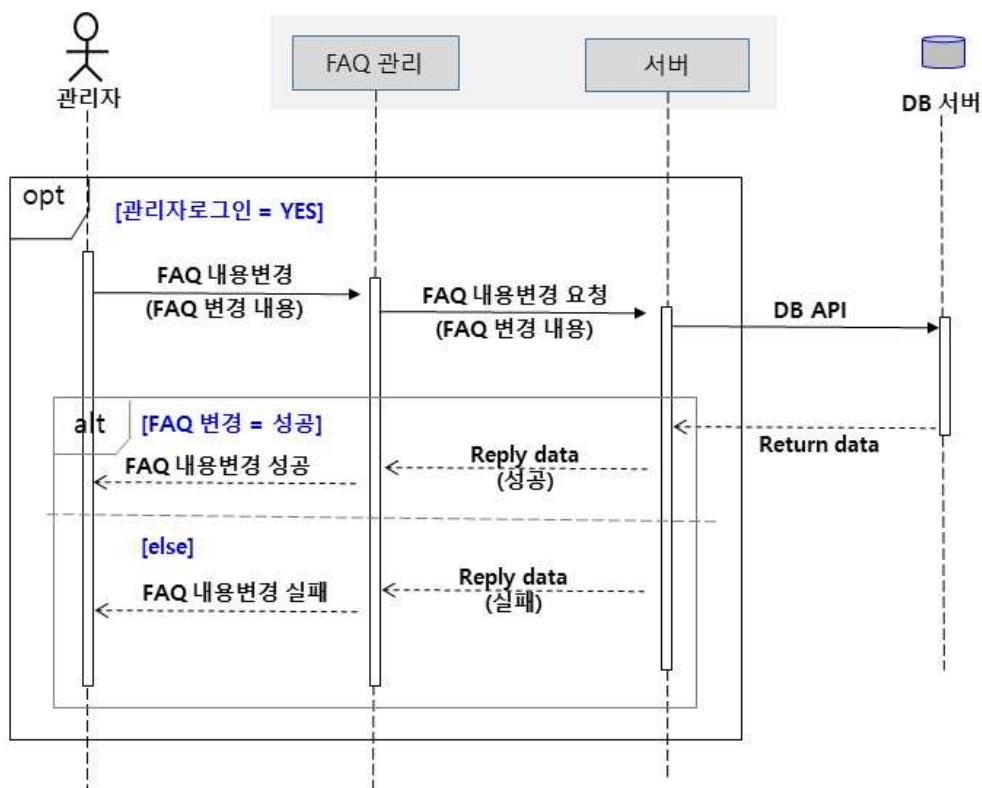


그림 3.33 FAQ 관리의 Sequence Diagram

그림 3.33을 보면 관리자는 관리자 인증을 성공한 후에 FAQ를 편집할 수 있다. 관리자가 변경 내용에 대해 서버로 보내면 서버에서는 DB에 해당 내용을 저장한다. 저 장에 성공한 경우 관리자에게 내용 변경 성공 메시지를 전달하고 실패한 경우 실패 메시지를 전달한다. 관리자는 웹페이지를 통해 챗봇에 대한 지식이 없어도 간편하게 FAQ를 관리할 수 있다.

11) 챗봇기능 - 챗봇로그

관리자는 사용자가 챗봇으로 입력한 내용에 대한 분석자료를 확인할 수 있다. 챗봇로그 기능을 구현하기 위해 아래와 같이 설계한다.

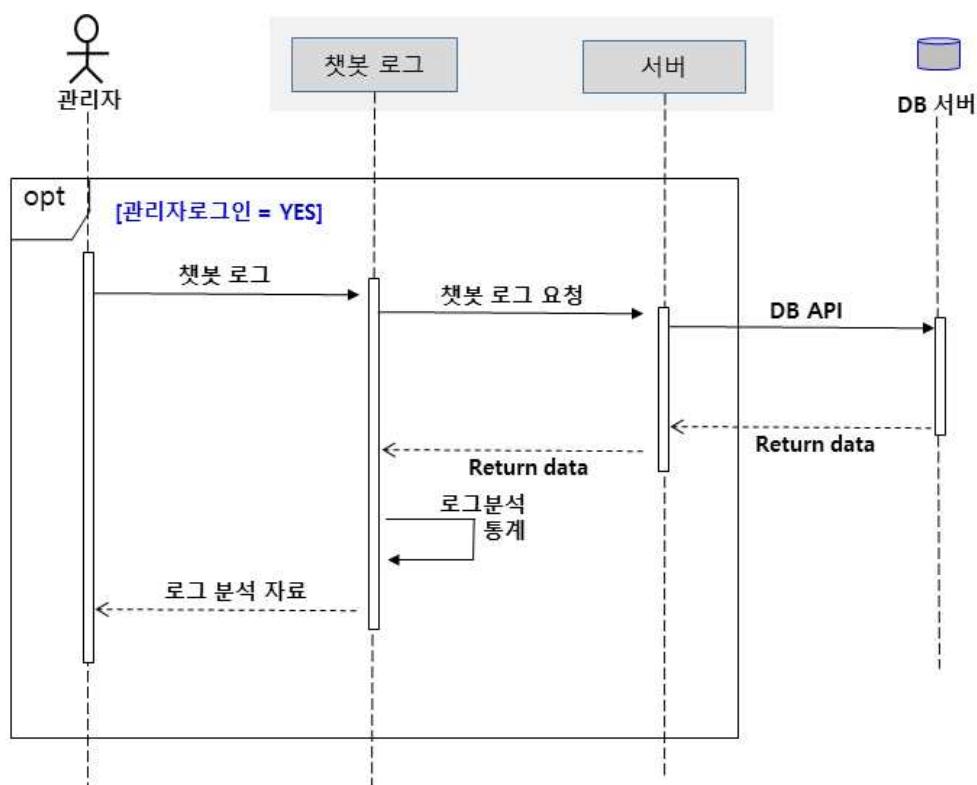


그림 3.34 챗봇로그의 Sequence Diagram

그림 3.34를 보면 관리자는 관리자 인증을 성공한 후에 챗봇 로그를 확인할 수 있다. 관리자가 챗봇로그 조회를 요청하면 서버에서는 챗봇을 이용한 사용자의 로그 데이터를 분석한다. 검색한 내용에 대해 통계를 내서 사용자가 주로 검색하고 사용하는 기능에 대해 웹페이지로 제공한다. 관리자는 시각화된 자료로 쉽게 사용자가 챗봇의 어떤 기능을 주로 이용하고 어떤 내용을 검색하는지 등을 확인할 수 있다.

12) 챗봇기능 - 미니퀴즈 관리

관리자는 사용자가 챗봇으로 하는 미니퀴즈의 내용을 변경할 수 있다. 미니퀴즈 관리 기능을 구현하기 위해 아래와 같이 설계한다.

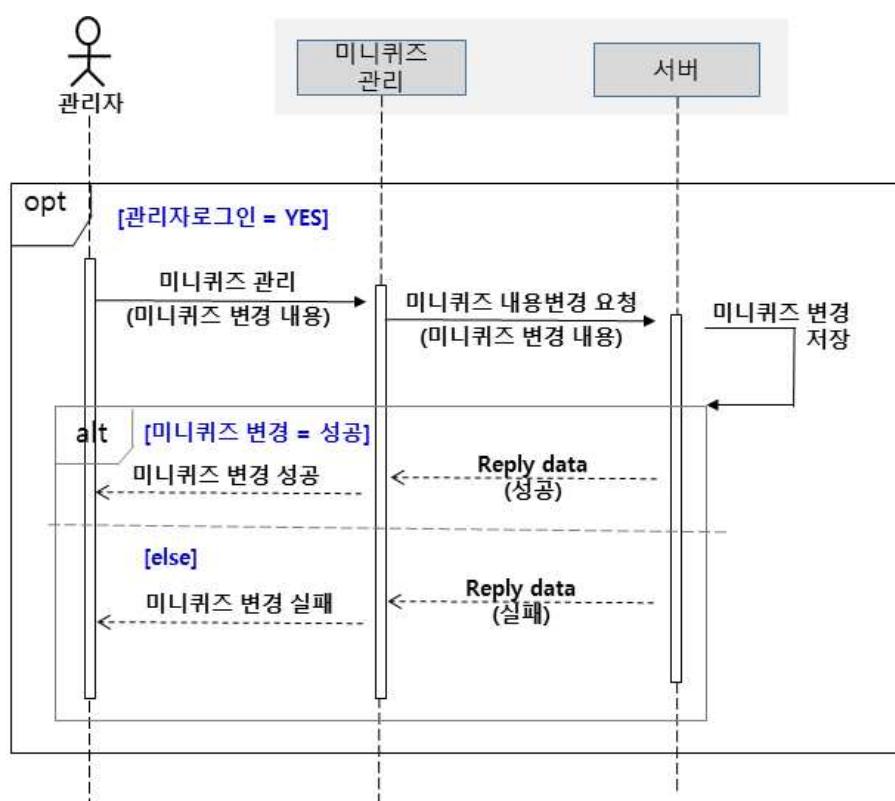


그림 3.35 미니퀴즈의 Sequence Diagram

그림 3.35를 보면 관리자는 관리자 인증을 성공한 후에 미니퀴즈의 내용을 변경할 수 있다. 미니퀴즈는 사용자가 챗봇에서 즐길 수 있는 작은 이벤트로 관리자는 미니퀴즈의 질문과 답을 설정할 수 있다. 관리자가 퀴즈를 변경하면 해당 내용은 서버에 저장된다.

13) 중고거래 기능 - 응모현황 조회

관리자는 사용자와 다르게 본인이 등록한 상품이 아니어도 모든 상품에 대해 모든 상품의 응모 정보를 확인할 수 있다. 응모현황 기능을 구현하기 위해 아래와 같이 설계한다.

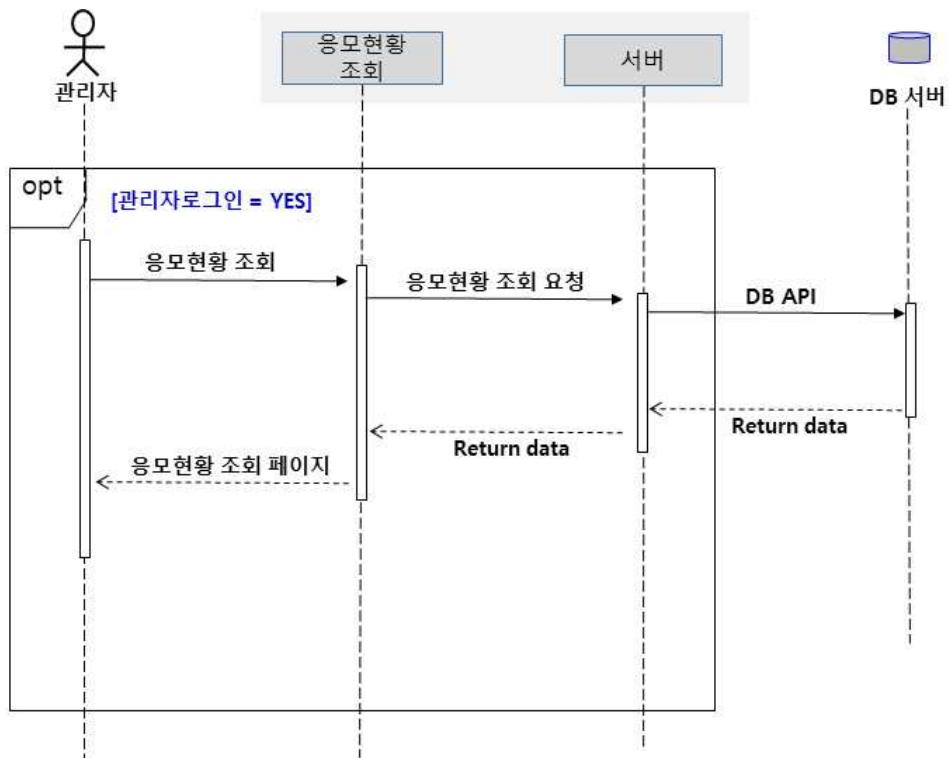


그림 3.36 응모현황 조회의 Sequence Diagram

그림 3.36을 보면 관리자는 관리자 인증을 성공한 후에 응모현황 조회 기능을 사용할 수 있다. 사용자가 등록한 중고 상품과 상품의 응모내역을 확인할 수 있으며, 서버로 응모현황 조회를 요청하면 응모현황 조회 페이지로 응모현황을 확인할 수 있다.

14) 중고거래 기능 - 당첨결과 조회

관리자는 중고거래의 당첨자 결과만 확인할 수 있다. 당첨결과 조회 기능을 구현하기 위해 아래와 같이 설계한다.

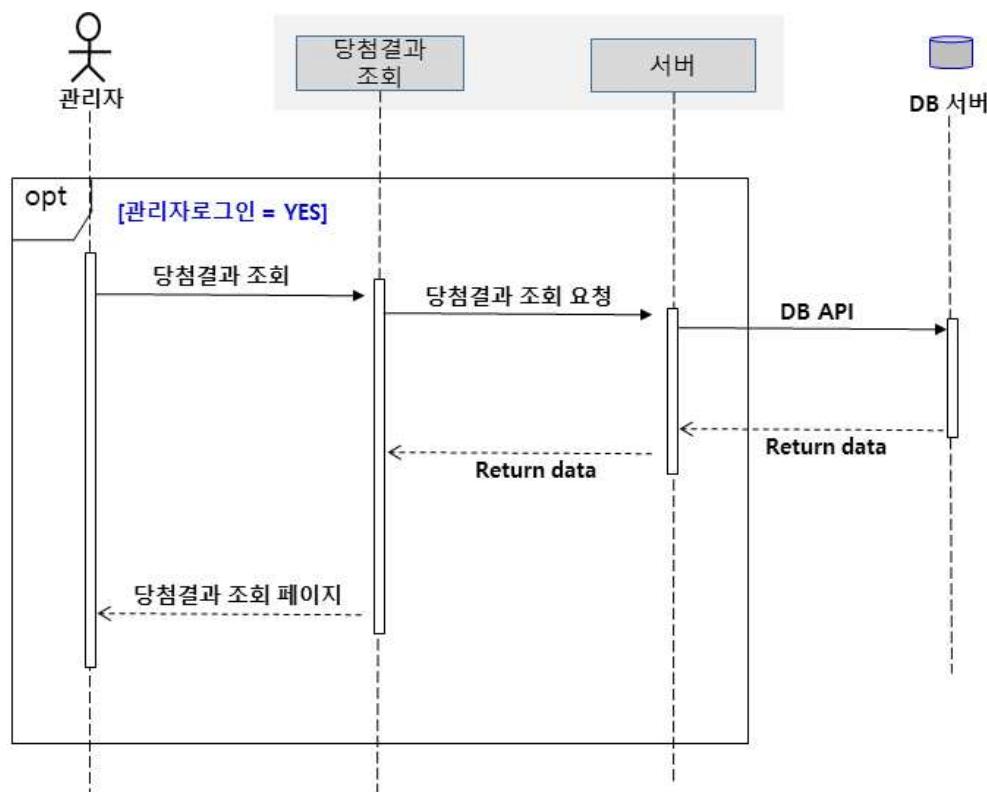


그림 3.37 당첨결과 조회의 Sequence Diagram

그림 3.37을 보면 관리자는 관리자 인증을 성공한 후에 당첨결과 조회를 사용할 수 있다. 관리자가 당첨결과를 조회 요청하면 서버로 전달되어 DB에서 성사된 중고거래 정보를 가져온다. 해당 정보는 웹페이지 형태로 관리자에게 제공된다.

15) 중고거래 기능 - 전체거래 조회

관리자는 거래 성사 여부와 상관없이 전체거래에 대한 정보를 조회할 수 있다. 전체거래 조회 기능을 구현하기 위해 아래와 같이 설계한다.

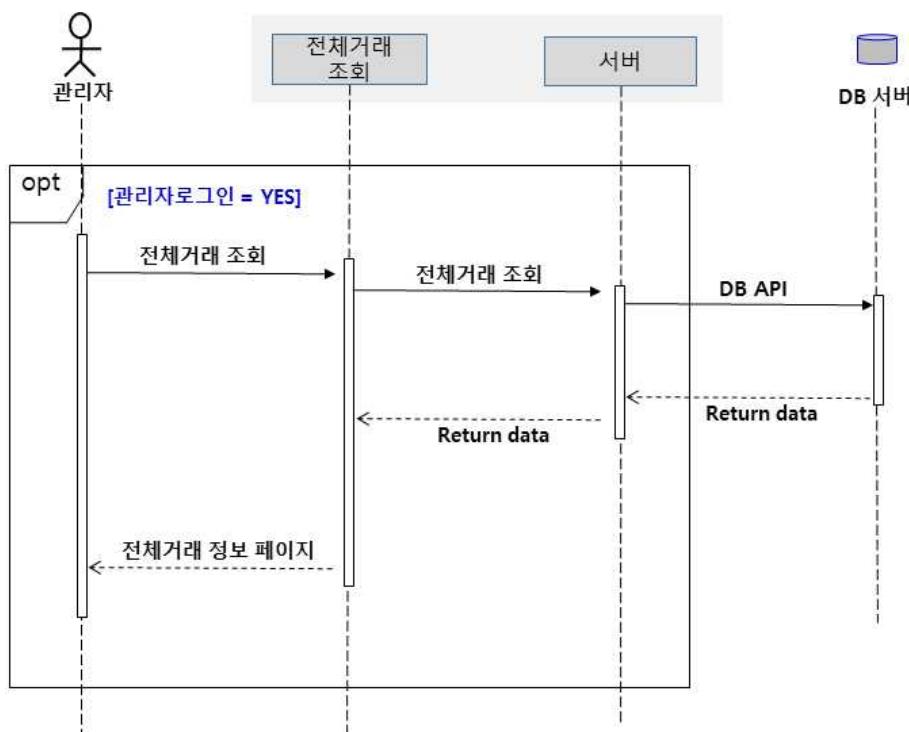


그림 3.38 전체거래 조회의 Sequence Diagram

그림 3.38을 보면 관리자는 관리자 인증을 성공한 후에 전체거래를 조회한다. 조회 요청이 서버로 들어가면 DB에서 해당하는 데이터를 받아오고 해당 데이터는 관리자가 파악하기 편한 형태로 가공하여 웹페이지를 통해 전달한다.

16) 수익관리 - 수수료 관리

관리자는 중고거래로 발생한 수수료 비율을 다시 설정할 수 있다. 수수료 관리 기능을 구현하기 위해 아래와 같이 설계한다.

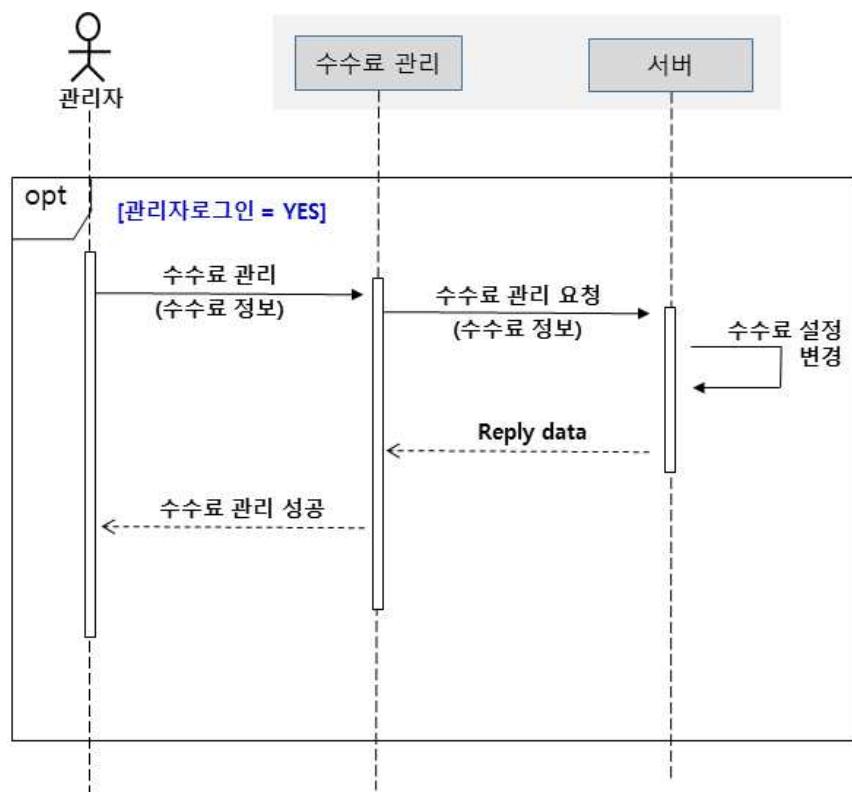


그림 3.39 수수료 관리의 Sequence Diagram

그림 3.39를 보면 관리자는 관리자 인증을 성공한 후에 수수료 관리 기능을 사용한다. 수수료 정보를 변경하여 서버로 보내면 서버 설정에서 수수료 비율을 변경한다. 수수료 비율의 경우 메타데이터로 DB에 저장되지 않기 때문에 서버에서 설정 파일을 변경한다.

17) 수익관리 - 매출 조회

관리자는 중고거래로 발생한 매출액 대해 파악할 수 있다. 매출 조회 기능을 구현하기 위해 아래와 같이 설계한다.

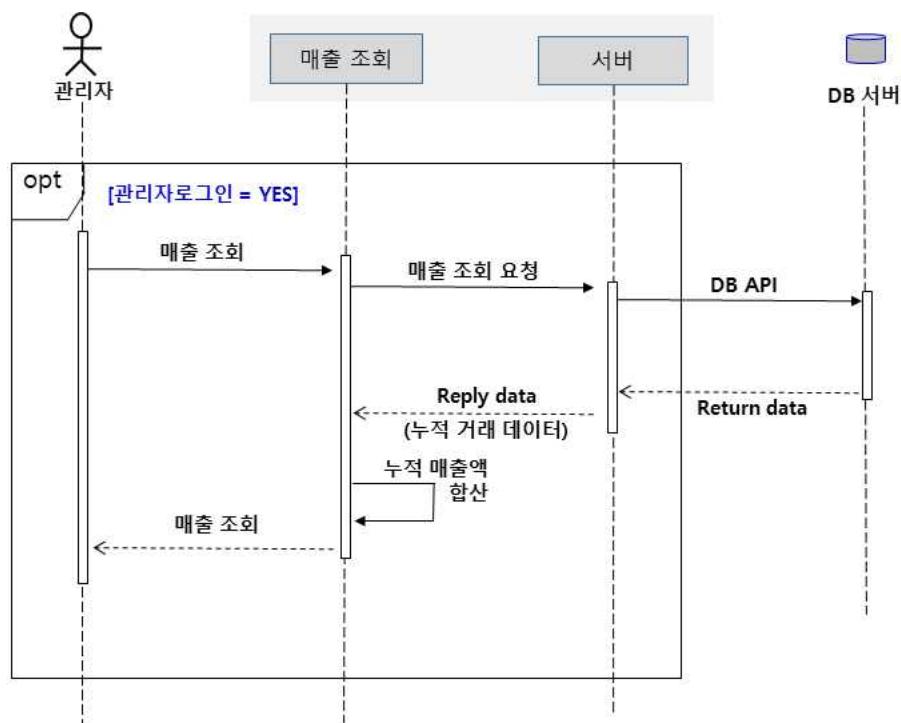


그림 3.40 매출 조회의 Sequence Diagram

그림 3.40을 보면 관리자는 관리자 인증을 성공한 후에 매출총액 조회 기능을 사용한다. 관리자가 매출총액 조회를 요청하면 서버에서 해당하는 정보를 DB에서 받아오고 합산하여 누적 매출액을 관리자에게 전달한다. 관리자는 웹페이지를 통해 매출 정보를 확인할 수 있다.

18) 게시판 기능 - 게시글 조회

관리자는 사용자가 작성한 게시글을 조회할 수 있다. 문의게시판과 신고게시판은 댓글 기능을 포함하고 있어 댓글 정보도 같이 조회한다. 게시글 조회 기능을 구현하기 위해 아래와 같이 설계한다.

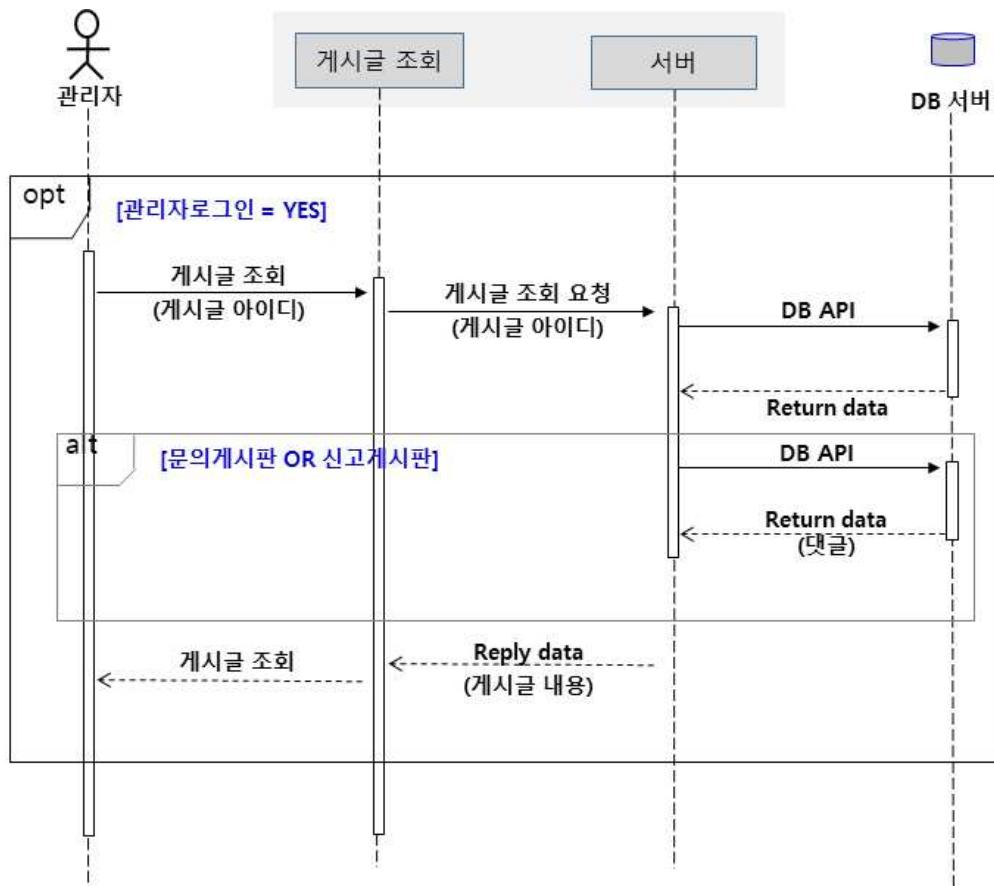


그림 3.41 매출총액 조회의 Sequence Diagram

그림 3.41을 보면 관리자는 관리자 인증을 성공한 후에 게시글을 자유롭게 조회할 수 있다. 관리자가 게시글 조회를 요청하면 게시글 아이디가 서버로 전달된다. 서버에서는 게시글 아이디로 DB에서 해당 내용을 검색하고 결과를 반환한다. 문의게시판과 신고게시판의 경우 댓글이 있을 수 있기 때문에 댓글 내용도 조회해서 관리자에게 안내한다.

19) 게시판 기능 - 게시글 삭제

관리자는 사용자가 작성한 게시글이 문제가 있다고 판단될 경우 게시글을 삭제할 수 있어야 한다. 관리자의 게시글 삭제 기능을 구현하기 위해 아래와 같이 설계한다.

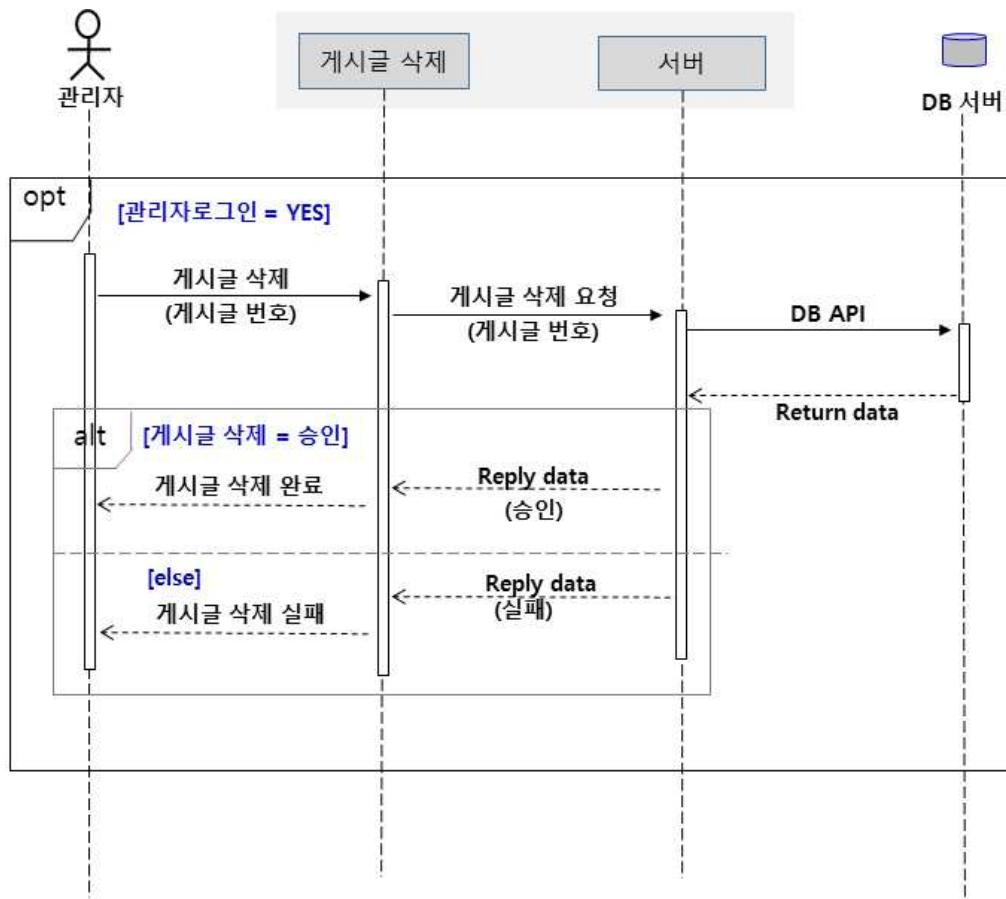


그림 3.42 게시글 삭제의 Sequence Diagram

그림 3.42를 보면 관리자는 관리자 인증을 성공한 후에 게시글 삭제 기능을 사용한다. 관리자가 게시글 삭제 버튼을 누르면 서버로 게시글 삭제 요청과 게시글 번호를 보낸다. 서버에서는 DB에 저장된 게시글 정보를 삭제하고 게시글 삭제 결과를 반환한다. 반환된 결과를 가지고 관리자에게 안내한다.

20) 게시판 기능 - 댓글 등록

관리자는 사용자가 작성한 문의, 신고 게시글에 대하여 댓글을 작성할 수 있다. 댓글 등록 기능을 구현하기 위하여 아래와 같이 설계한다.

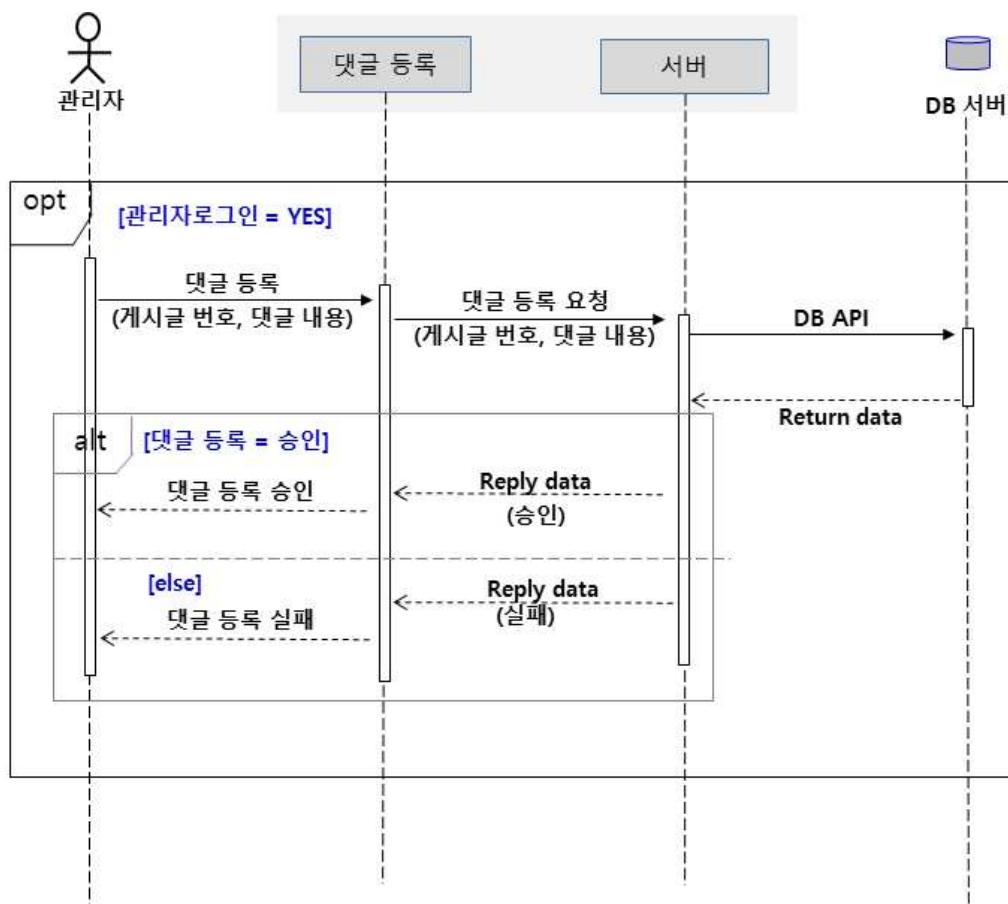


그림 3.43 댓글 등록의 Sequence Diagram

그림 3.43을 보면 관리자는 관리자 인증을 성공한 후에 댓글 등록 기능을 사용할 수 있다. 관리자가 댓글을 등록하면 게시글 번호와 댓글 내용을 받아 서버로 전송한다. DB에서 해당 내용을 저장하고 결과를 서버에 반환한다. 서버에서는 결과에 따라 관리자에게 다른 안내 메시지를 전달한다.

21) 게시판 기능 - 댓글 수정

관리자는 작성한 댓글을 수정할 수 있다. 댓글 수정 기능을 구현하기 위해 아래와 같이 설계한다.

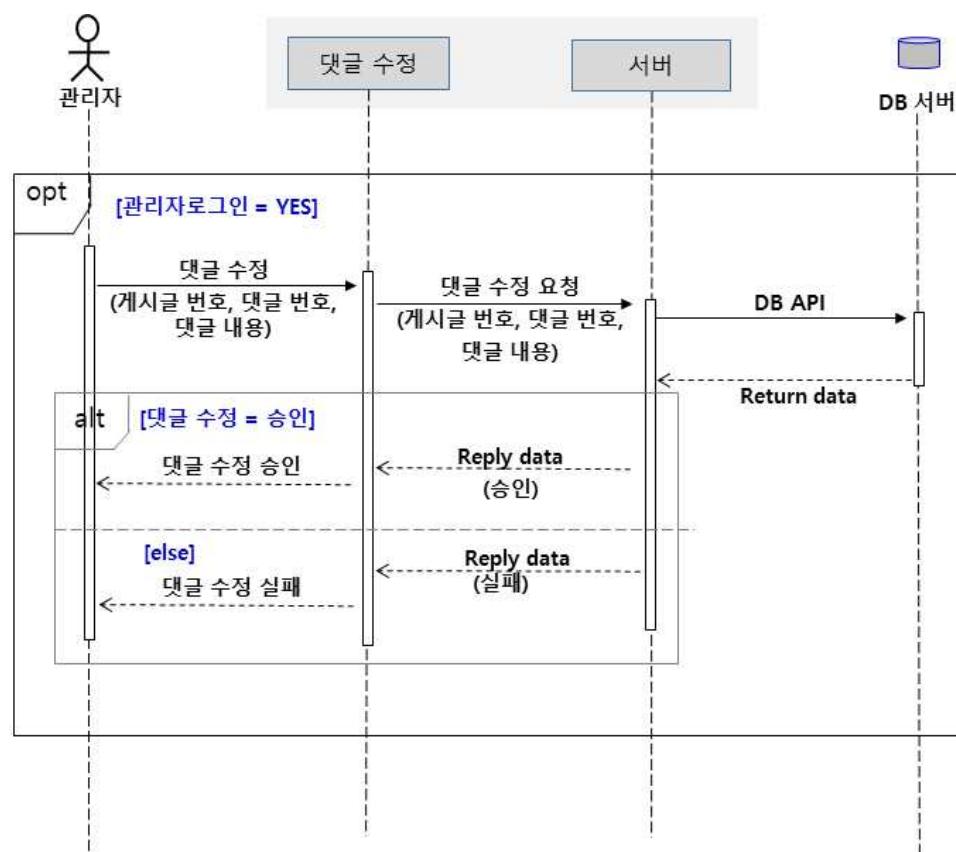


그림 3.44 댓글 수정의 Sequence Diagram

그림 3.44를 보면 관리자는 관리자 인증을 성공한 후에 댓글수정 기능을 사용한다. 해당 게시글의 댓글 번호와 변경할 댓글 내용을 전달하면 웹서버에서 해당 내용을 받아서 DB에 변경한다. 서버에서 DB를 변경한 결과에 따라 댓글 수정 결과를 관리자에게 안내한다.

22) 게시판 기능 - 댓글 삭제

관리자는 작성한 댓글에 대해 삭제할 수 있다. 댓글삭제 기능을 구현하기 위해 아래와 같이 설계한다.

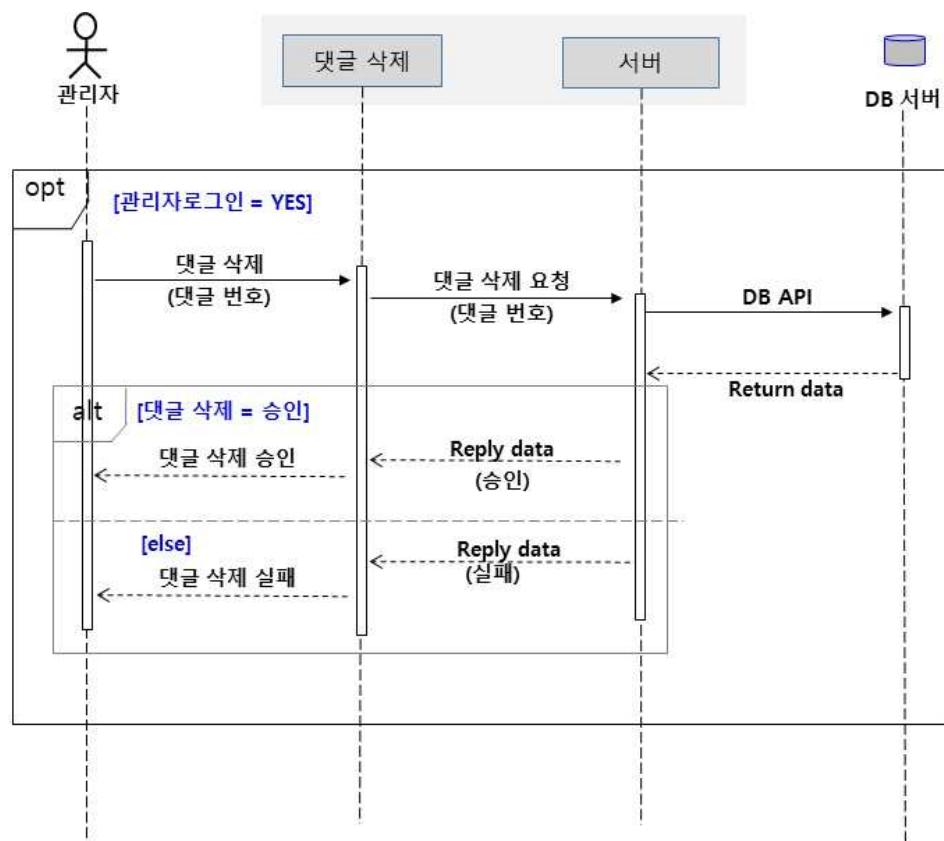


그림 3.45 댓글 삭제의 Sequence Diagram

그림 3.45를 보면 관리자는 관리자 인증을 성공한 후에 댓글삭제 기능을 사용한다. 댓글 삭제를 누르면 댓글 번호를 가지고 서버에 댓글 삭제를 요청한다. 서버에서는 댓글 번호를 아이디로 사용하여 DB에서 댓글을 삭제할 수 있다.

4. 화면(UI) 설계

4.1. 스토리보드(메뉴) 구성

■ 사용자

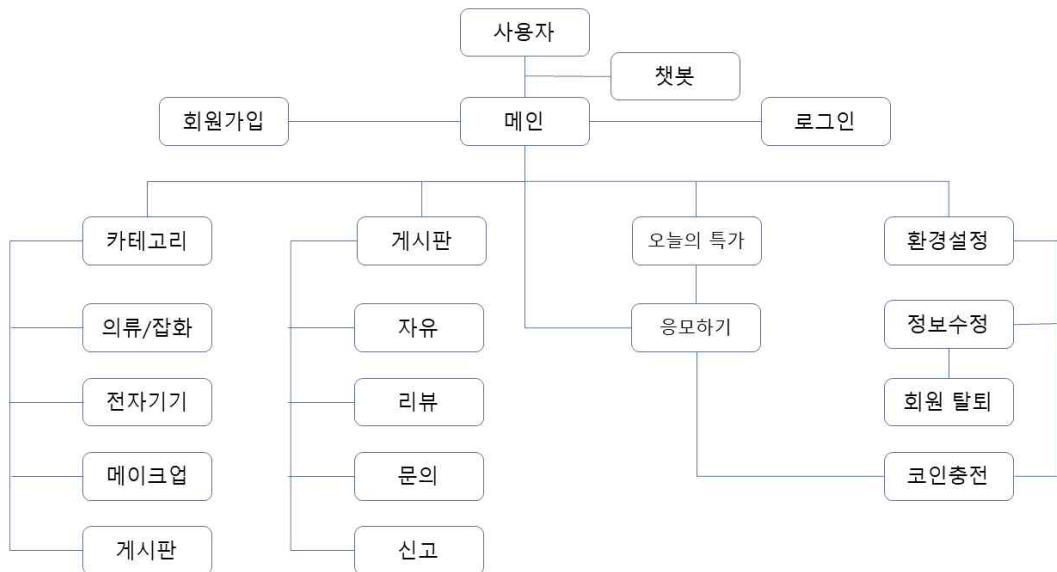


그림 4.1 사용자용 메뉴구조도

■ 관리자

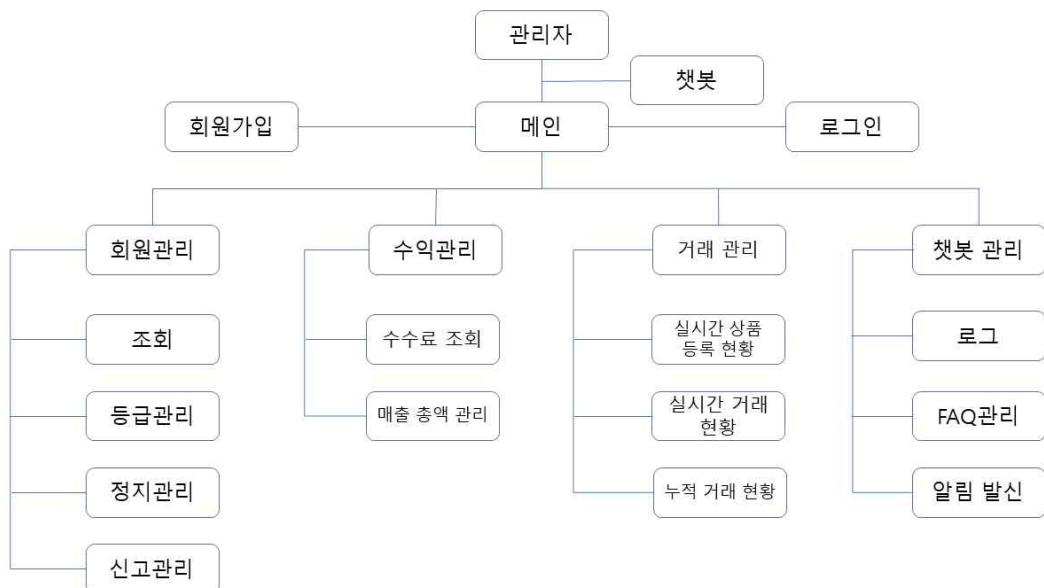
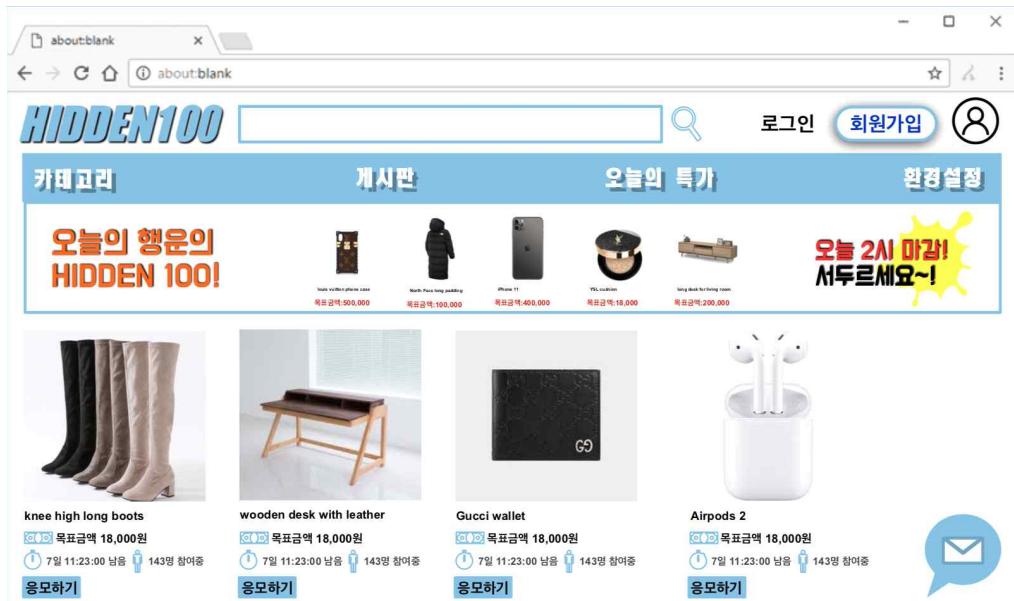


그림 4.2 관리자용 메뉴 구조도

4.2. 초기화면(로그인 전화면) 설계

■ 초기 화면(로그인 전)



- 웹에 접속하면 가장 먼저 보이는 메인 페이지이다.
- 네비게이션 바에서 카테고리, 게시판, 오늘의 특가, 환경설정 기능을 선택할 수 있다.
- 오른쪽 상단의 로그인, 회원가입을 통해 사용자 인증이 가능하다.
- 인기응모제품들이 메인 페이지 상단에 보여진다.
- 오른쪽 하단의 메시지 버튼을 클릭하여 챗봇 기능을 사용할 수 있다.
- 최상단의 네모박스에서 찾고자 하는 제품을 검색할 수 있다.
- 제품 하단의 돈 아이콘 옆의 목표금액은 응모 성사를 위한 목표금액이다. 만약 총 응모 금액이 목표금액을 넘지 못할 시에는 응모 성사가 이루어지지 않게 된다. 이는 실패한 응모로 간주되고 총 응모금액은 관리자가 소유한다.
- 돈 아이콘 하단의 시계 아이콘은 응모 가능한 남은 시간이다. 시간이 0이 되면 응모는 더 이상 불가능하다.
- 시계 아이콘 오른편의 사람 아이콘은 해당 상품에 응모한 총 참여자수이다.
- 사용자는 상품 최하단의 파란색 응모하기 버튼을 클릭하여 응모에 참여할 수 있다. ┼

4.3. 사용자UI 설계

■ 회원가입 초기 화면

The screenshot shows a web browser window for 'about:blank'. The main content area displays the 'HIDDEN100' logo and a search bar. On the right side of the header, there are '로그인' (Login) and '회원가입' (Member Registration) buttons, with '회원가입' being highlighted. Below the header, there are navigation links: '카테고리' (Category), '의류/잡화' (Clothing/General Goods), '게시판' (Board), '오늘의 특가' (Today's Special Offer), and '환경설정' (Environment Settings). The main content area has a title '3초만에 회원가입하기' (Join in 3 seconds) and a link '로그인 하러가기' (Go to login). It features three input fields: 'ID' (아이디를 입력해주세요), 'PW' (비밀번호를 입력해주세요), and 'PW확인' (비밀번호를 다시 입력해주세요). A blue button labeled '완료' (Complete) is at the bottom left, and a blue speech bubble icon with an envelope symbol is at the bottom right.

- 회원가입을 누르면 나타나는 화면이다.
- 사용자는 이 화면에서 회원가입을 할 수 있다.

■ 회원 가입 화면(텍스트 창이 공란일 때)

The screenshot shows a web browser window for 'about:blank'. The main content is a registration form titled '3초만에 회원가입하기' (Join in 3 seconds). The form has three text input fields: 'ID' (containing '아이디를 입력해주세요'), 'PW' (containing '*****'), and 'PW확인' (containing '*****'). To the right of the 'ID' field is a button labeled '아이디 중복 체크' which has a red error message '아이디가 입력되지 않았습니다'. Below the form is a blue button labeled '완료' and a blue speech bubble icon.

- 텍스트 창이 공란이면 경고 문구가 나타난다.

■ 회원 가입 화면(아이디 중복 확인)

The screenshot shows a web browser window for 'about:blank'. The main content is a registration form titled '3초만에 회원가입하기' (Join in 3 seconds). The form has three text input fields: 'ID' (containing 'sonsin'), 'PW' (containing '*****'), and 'PW확인' (containing '*****'). To the right of the 'ID' field is a button labeled '아이디 중복 체크' which has a red error message '사용 불가능한 아이디입니다.' Below the form is a blue button labeled '완료' and a blue speech bubble icon.

- 아이디를 입력하고 아이디 중복 체크를 한다.
- 이미 사용 중인 아이디라면 ‘사용 불가능한 아이디입니다’ 경고 문구가 나타난다.

■ 회원 가입 화면(아이디 중복 확인)

3초만에 회원가입하기

ID: soonsin (아이디 중복 체크: 사용 가능한 아이디입니다.)

PW: 비밀번호를 입력해주세요

PW확인: 비밀번호를 다시 입력해주세요

완료

- 아이디를 입력하고 아이디 중복 체크를 한다.
- 미사용 중인 아이디라면 ‘사용 가능한 아이디입니다’ 문구가 나타난다.

■ 회원 가입 화면(입력한 비밀번호와 비밀번호 확인이 불일치할 때)

3초만에 회원가입하기

ID: soonsin (아이디 중복 체크: 사용 가능한 아이디입니다.)

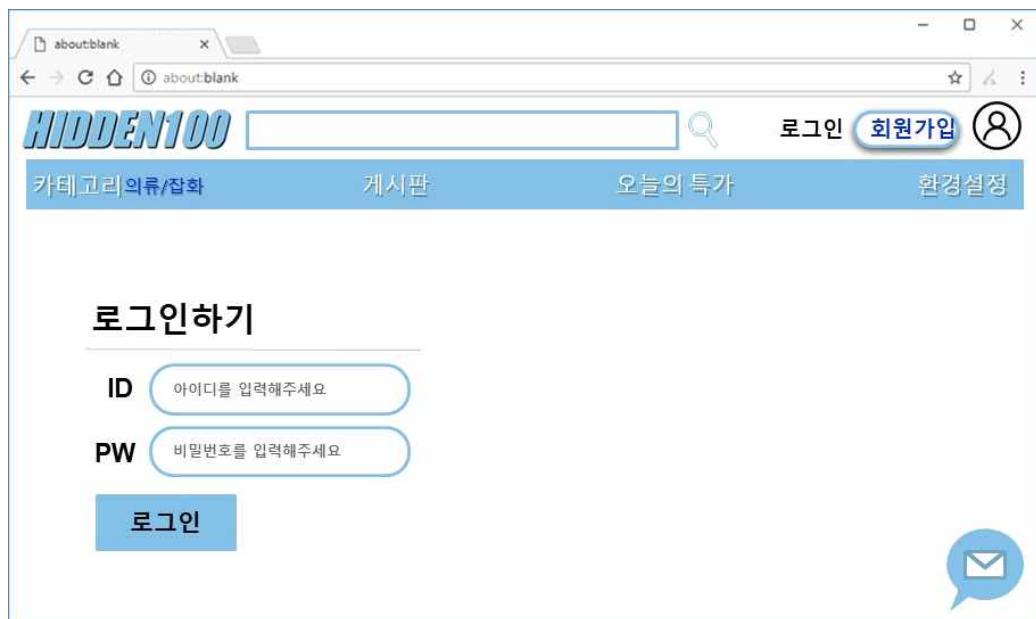
PW: *****

PW확인: ***** (비밀번호가 일치하지 않습니다)

완료

- 비밀번호와 비밀번호 확인을 입력하고 완료 버튼을 클릭한다.
- 두 데이터가 불일치하면 ‘비밀번호가 일치하지 않습니다’ 경고 문구가 나타난다.

■ 로그인 초기 화면



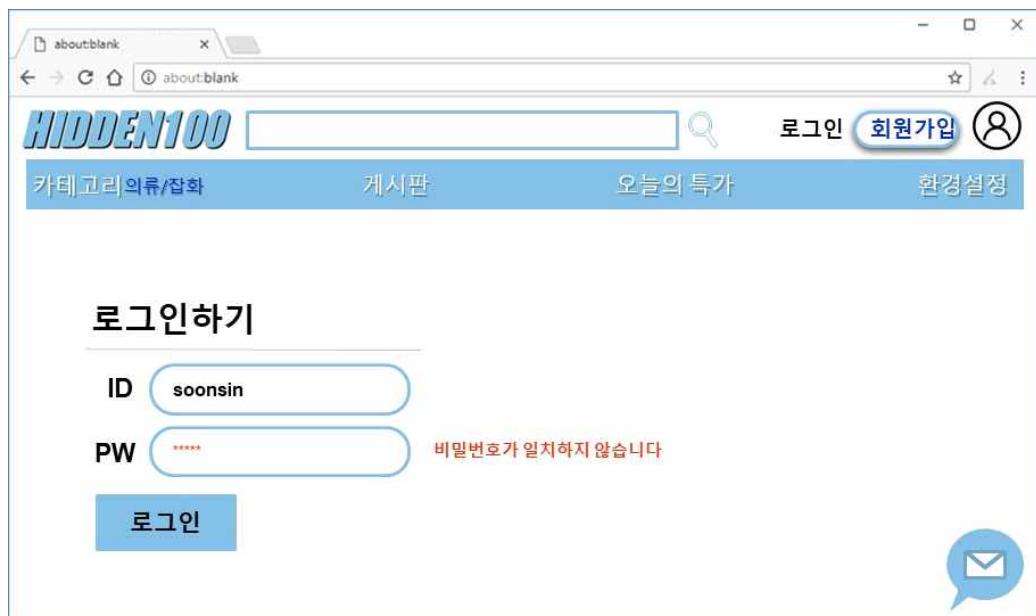
- 사용자가 로그인을 하는 화면이다.
- 아이디와 비밀번호를 입력해서 로그인한다.

■ 로그인 화면(텍스트 창이 공란일 때)



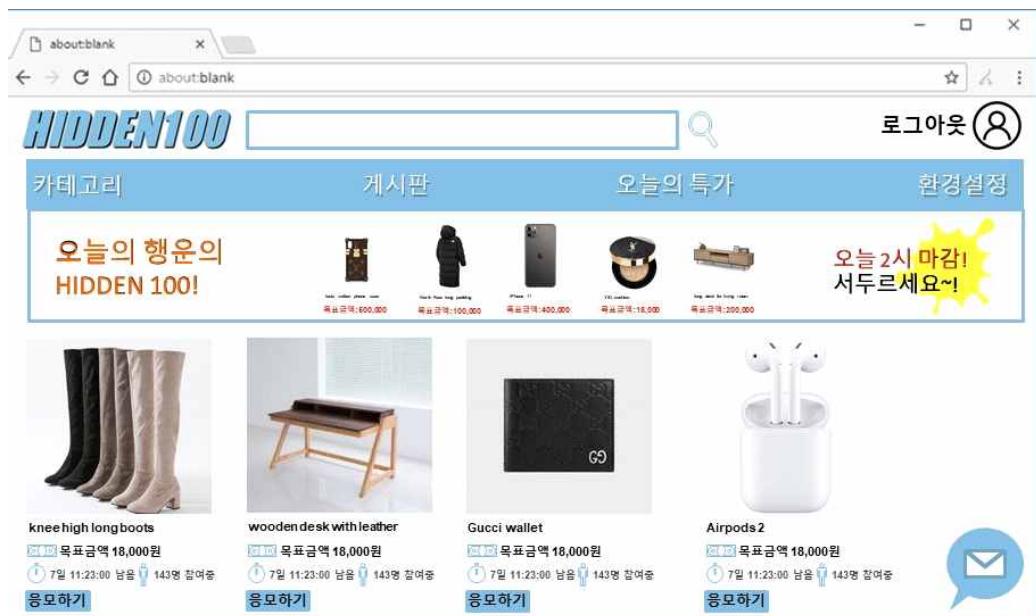
- 아이디 텍스트 창이 공란이면 ‘아이디가 입력되지 않았습니다’ 경고 문구가 나타난다.
- 비밀번호 텍스트 창이 공란이면 ‘비밀번호가 입력되지 않았습니다’ 경고 문구가 나타난다.

- 로그인 화면(텍스트 창의 데이터와 데이터베이스의 데이터가 불일치할 때)



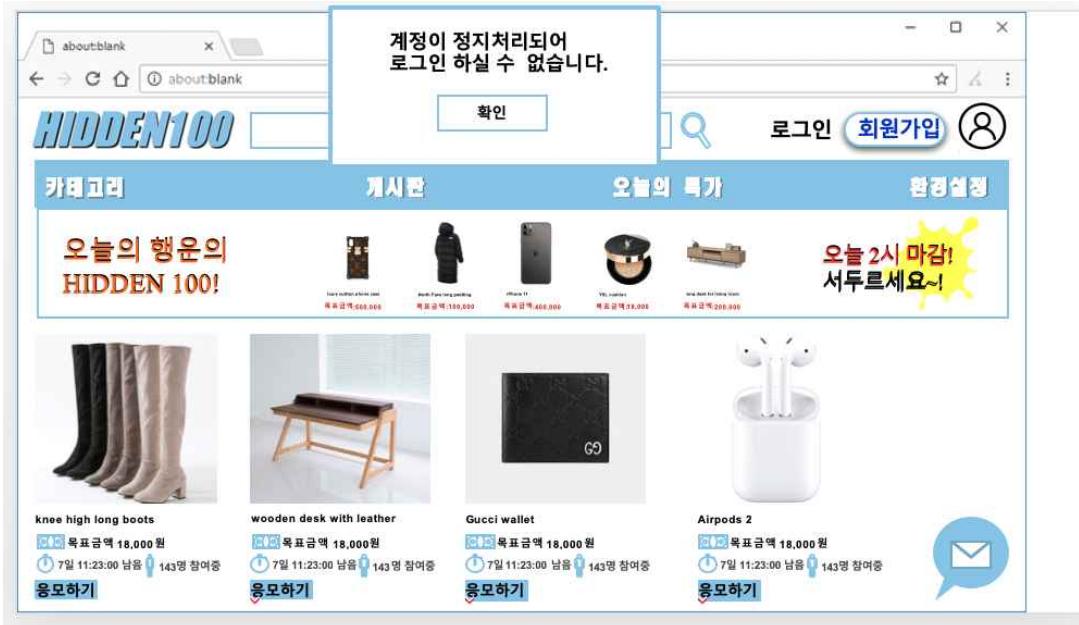
- 아이디 텍스트 창의 데이터와 데이터베이스의 데이터가 불일치하면 ‘아이디가 일치하지 않습니다’ 경고 문구가 나타난다.
- 비밀번호 텍스트 창의 데이터와 데이터베이스의 데이터가 불일치하면 ‘비밀번호가 일치하지 않습니다’ 경고 문구가 나타난다.

- 로그인 완료화면



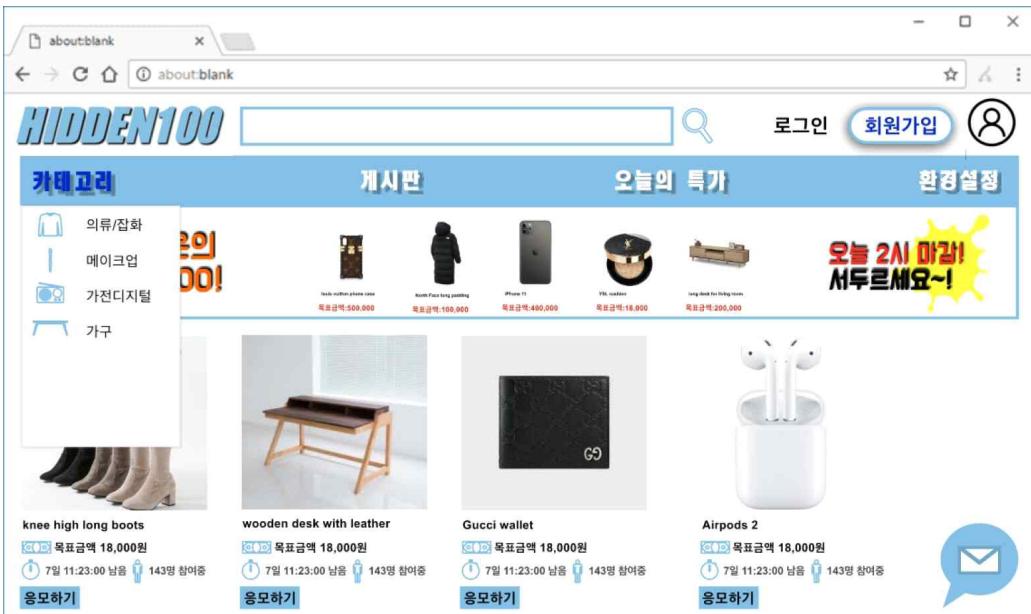
- ‘로그인’이 ‘로그아웃’으로 바뀐다.
- ‘회원가입’ 버튼이 사라진다.

■ 로그인 완료화면(계정이 정지 되었을 때)



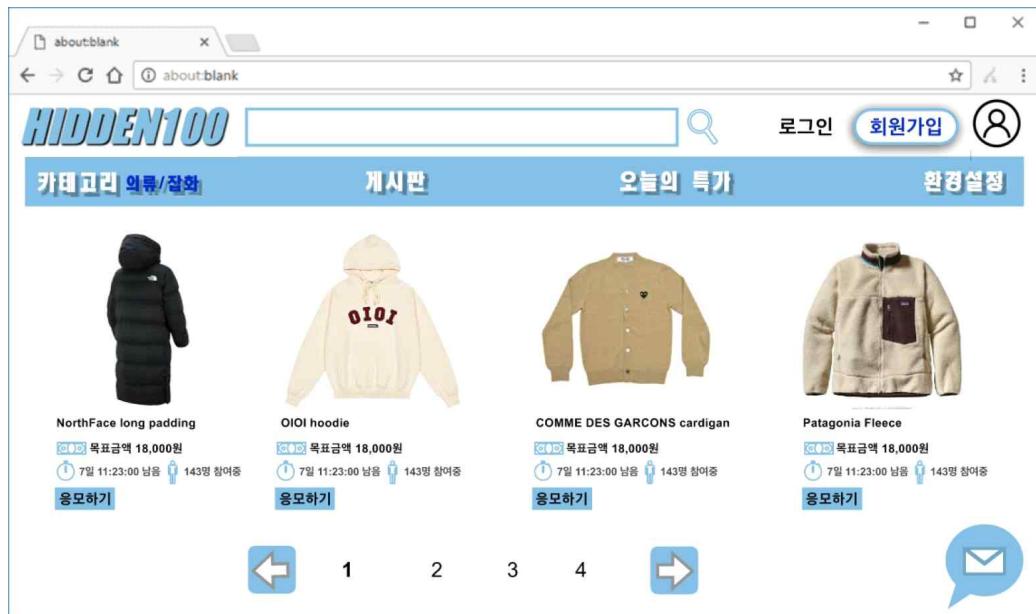
- 계정이 정지 된 것을 알려주는 알람창이 뜬다.

■ 초기화면(네비게이션 바 마우스 호버링)



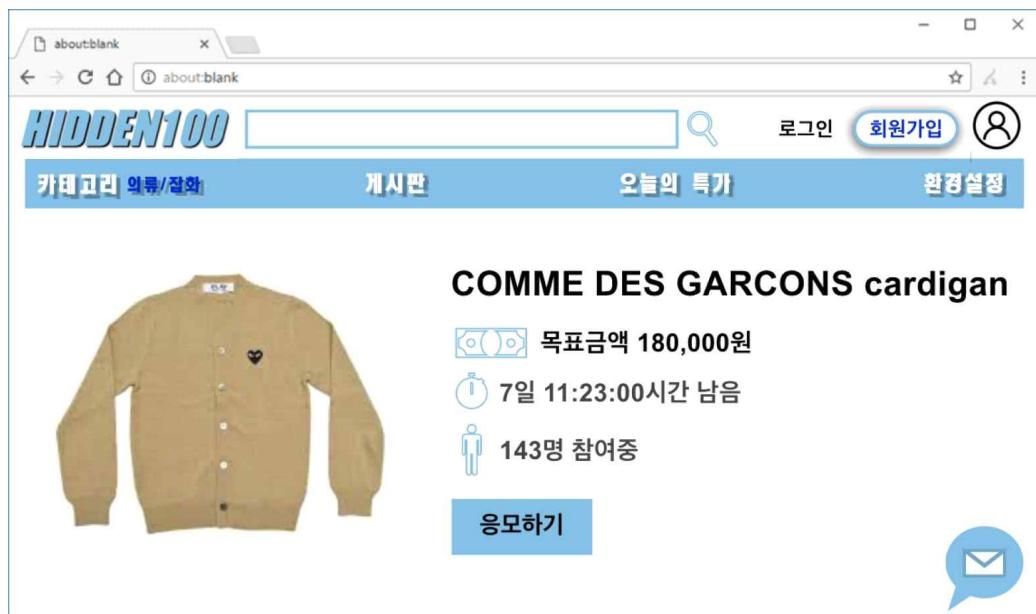
- 메인 페이지에서 네비게이션 바의 '카테고리' 분류에 마우스를 호버링한 상태이다.
- 각 분류에 마우스를 호버링하면 폰트색이 변경되면서 드롭다운 메뉴가 나타난다.

■ 의류/잡화 판매 페이지



- ‘카테고리’에서 ‘의류/잡화’를 선택한 페이지이다.
- ‘카테고리’ 옆에 ‘의류/잡화’ 텍스트가 나타난다.
- 사용자는 해당 페이지에서 물건판매를 할 수 있다.
- 사용자는 해당 페이지에서 물건등록을 할 수 있다.
- 페이지네이션 기능을 이용하여 페이지 이동이 가능하다.

■ 상품 페이지



- 판매자가 등록한 상품의 이미지, 이름, 목표금액, 시간이 나타난다.
- 구매자가 응모할수록 응모 참여자수가 증가한다.
- 구매자가 응모하기 버튼을 클릭하면 상품에 응모할 수 있다.

■ 상품 페이지(로그인 상태가 아닐 때)



- 로그인 상태가 아닌 경우 응모하기 버튼을 클릭하면 '로그인이 필요한 서비스입니다' 경고 메시지가 나타난다.
- 사용자는 로그인 상태인 경우에만 상품 응모가 가능하다.

■ 응모 페이지



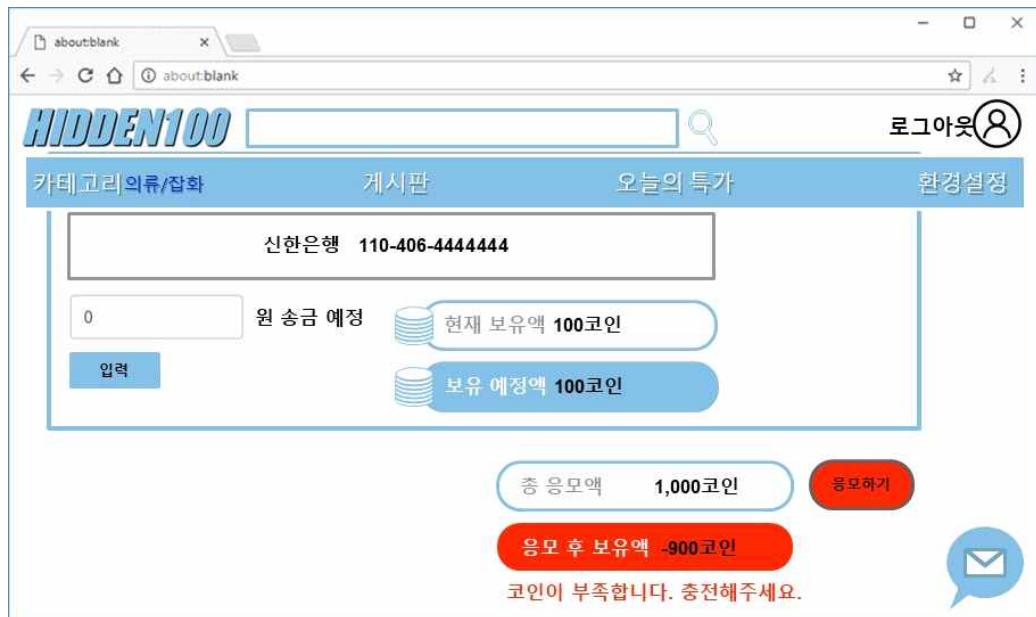
- 기본 상품응모횟수는 1번이다.
- 상품응모횟수를 변경하면 응모금액이 자동으로 업데이트되고 응모금액 부분에 변경
- 된 응모금액이 나타난다.

■ 응모 페이지(응모금액 변경)



- 상품응모금액을 변경하면 총 응모액과 응모 후 보유액이 업데이트된다.
- 구매자는 노란색의 응모하기 버튼을 클릭하여 상품응모가 가능하다.

■ 응모 페이지(현재 보유액이 부족한 경우)



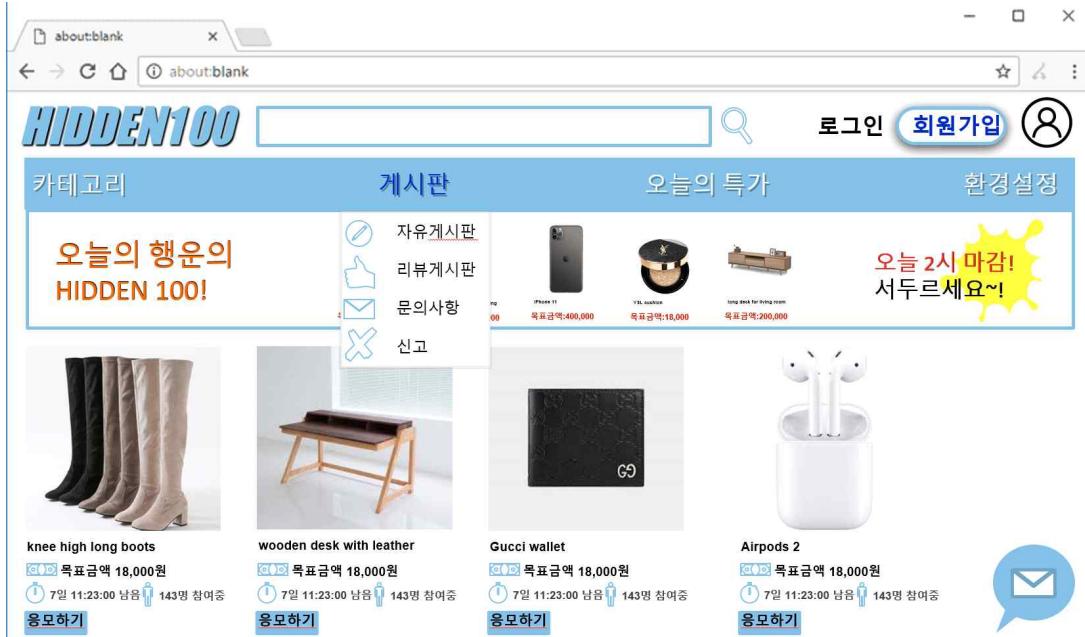
- 현재 보유액이 총 응모액보다 부족하면 응모 후 보유액이 음수로 변한다.
- 현재 보유액이 총 응모액보다 부족하면 ‘코인이 부족합니다. 충전해주세요’ 경고 문구
- 가 나타난다.
- 사용자는 코인을 충전해서 현재 보유액을 증가시킬 수 있다.

■ 응모 페이지(응모에 성공한 경우)



- 현재 보유액이 총 응모액보다 높아 응모에 성공한 경우 ‘축하합니다. 응모에 성공하셨습니다!’ 문구가 화면 상단에 나타난다.
- 사용자는 화면 하단의 다른 응모 보러 가기 버튼을 클릭하여 등록되어 있는 다른 상품들을 볼 수 있다.

■ 초기화면(네비게이션 바 마우스 호버링)



- 메인 페이지에서 네비게이션 바의 ‘게시판’ 분류에 마우스를 호버링한 상태이다.
- 각 분류에 마우스를 호버링하면 폰트색이 변경되면서 드롭다운 메뉴가 나타난다.

■ 자유게시판 화면

The screenshot shows the 'HIDDEN100' website interface. At the top, there is a navigation bar with links for '로그인' (Login) and '회원가입' (Sign Up). Below the navigation, there are tabs for '카테고리' (Category), '게시판' (Board), '자유게시판' (Free Classifieds), '오늘의 특가' (Today's Specials), and '환경설정' (Environment Settings). A banner on the left says '오늘의 행운의 HIDDEN 100!' and features five items with their names and prices: 'Black leather phone case' (500,000), 'North Face long padding' (100,000), 'iPhone 11' (400,000), 'VLC cushion' (15,000), and 'Long desk for living room' (200,000). On the right, a yellow speech bubble says '오늘 2시 마감! 서두르세요~!' (Today's deadline at 2 PM! Hurry up!). Below the banner is a list of posts in the '자유게시판' section:

번호	제목	작성자	작성일	조회수
1	나 당첨 좀 시켜주라	이우석	2019-11-30	0
2	노트북 급처합니다!!!!	한혜경	2019-10-20	40
3	후드티 팔아요	이다은	2019-10-30	129
4	아이폰 6s 삽니다~~~~.	서지연	2019-09-20	342

At the bottom of the list, there are navigation arrows for page 1, 2, 3, 4, and 5, along with buttons for '글쓰기' (Write a post) and '검색' (Search).

- ‘게시판’에서 ‘자유게시판’을 선택한 페이지이다.
- ‘게시판’ 옆에 ‘자유게시판’ 텍스트가 나타난다.
- 챗봇 링크의 위치가 화면 하단에서 중간으로 이동한다.
- 사용자는 글쓰기 버튼을 클릭해서 게시판에 글작성이 가능하다.
- 사용자는 텍스트 창에 단어를 입력해서 원하는 글을 검색할 수 있다.
- 페이지네이션 기능을 이용하여 페이지 이동이 가능하다.

■ 자유게시판 화면(로그인 상태가 아닌 경우 글쓰기 버튼을 눌렀을 때)

The screenshot shows the 'HIDDEN 100' website's free classifieds section. At the top right, there are 'Login' and 'Member Join' buttons. Below the header, there are sections for 'Today's Specials' and 'Today's Trending'. The main area displays a list of posts with columns for '번호' (Number), '제목' (Title), '작성자' (Author), '작성일' (Date), and '조회수' (Views). The 'Write' button is located at the bottom right of the post list.

번호	제목	작성자	작성일	조회수
1	나 당첨 좀 시켜주라	이우석	2019-11-30	0
2	노트북 급처합니다!!!!	한혜경	2019-10-20	40
3	후드티 팔아요	이다은	2019-10-30	129
4	아이폰 6s 삽니다~~~.	서지연	2019-09-20	342

- 사용자가 로그인 상태가 아닌 경우 글쓰기 버튼을 클릭하면 ‘로그인이 필요한 서비스입니다’ 경고 메시지가 나타난다.
- 사용자는 로그인을 한 상태에서만 글쓰기가 가능하다.

■ 자유게시판 화면(로그인 상태인 경우)

The screenshot shows the same website structure as above, but for a logged-in user. The 'Login' button has been replaced by a 'Logout' button. The rest of the interface, including the post list and navigation buttons, remains identical.

번호	제목	작성자	작성일	조회수
1	나 당첨 좀 시켜주라	이우석	2019-11-30	0
2	노트북 급처합니다!!!!	한혜경	2019-10-20	40
3	후드티 팔아요	이다은	2019-10-30	129
4	아이폰 6s 삽니다~~~.	서지연	2019-09-20	342

- 사용자가 로그인 상태인 경우 글쓰기 버튼을 클릭하면 게시판 글작성이 가능하다.
- 화면 오른쪽 상단의 ‘로그인/회원가입’이 ‘로그아웃’으로 변경된다.

■ 글쓰기 화면



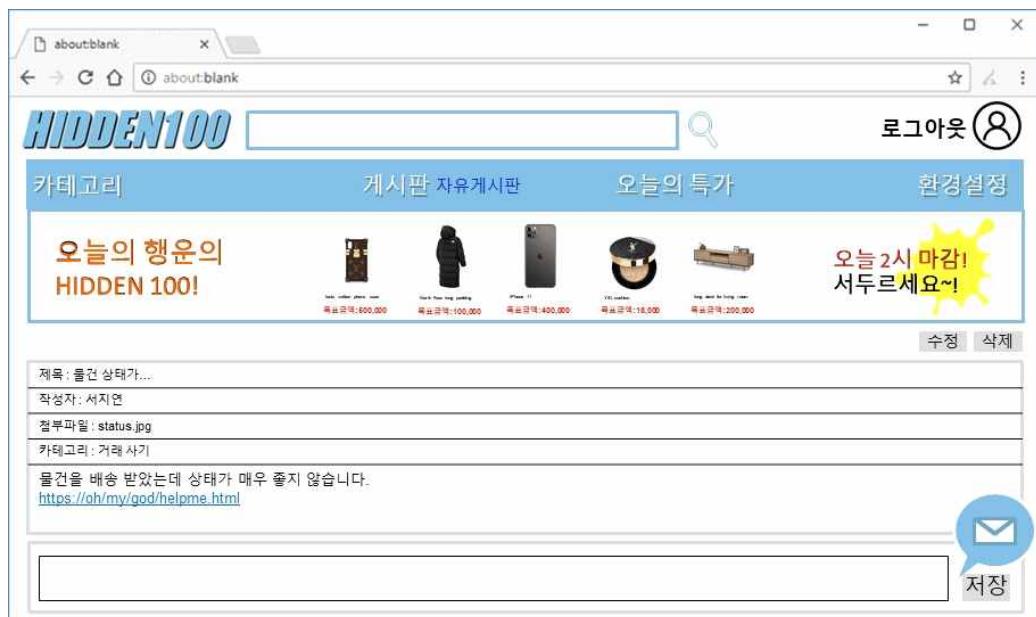
- ‘카테고리’에서 게시판 종류를 선택할 수 있다.
- 게시판은 자유게시판, 문의게시판, 신고게시판이 있다.
- 첨부파일은 ‘찾기’ 버튼을 클릭하여 추가할 수 있다.

■ 글쓰기 화면(내용이 공란인 상태에서 글작성을 시도했을 때)



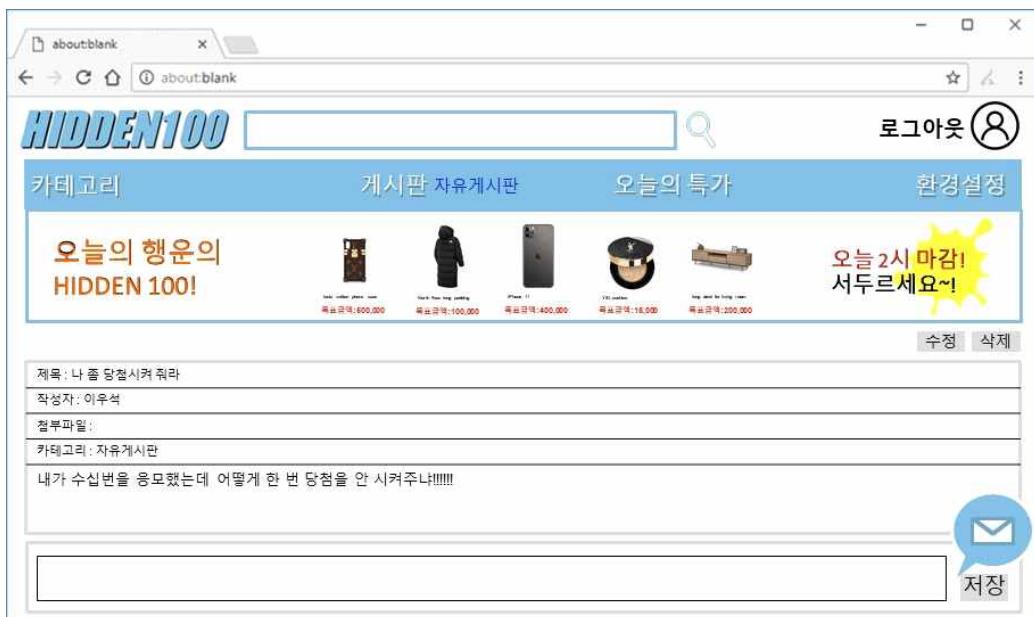
- 내용이 빈 상태에서 ‘저장’ 버튼을 클릭하면 글작성이 거부되고 ‘내용을 입력해 주세요’ 경고 메시지가 나타난다.
- 경고 메시지의 ‘확인’ 버튼을 클릭하면 경고 메시지가 사라진다.
- 화면 하단의 ‘취소’ 버튼을 클릭하면 글작성이 취소되고 게시판 페이지로 돌아간다.

■ 글쓰기 화면(내용이 입력되고 글작성을 시도했을 때)



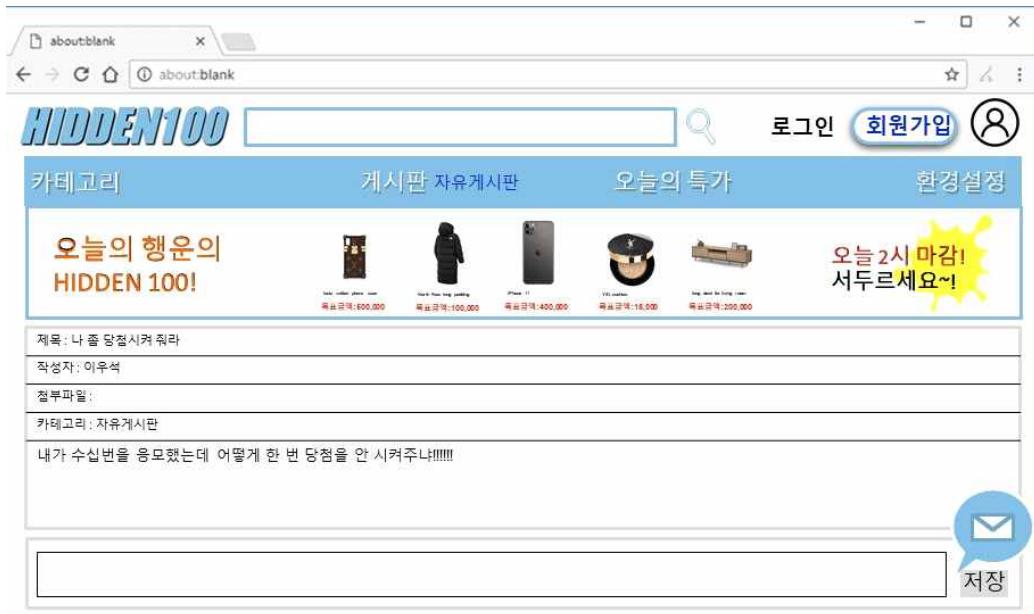
- 내용이 입력된 상태에서 ‘저장’ 버튼을 클릭하면 글작성이 완료된다.
- 제목, 작성자, 카테고리, 내용에 데이터가 채워진다.
- 첨부파일을 추가하고 글작성을 하면 첨부파일이 나타나고 다운로드 가능하다.

■ 게시글 조회 화면(글 작성자인 경우)



- 게시글 위편에 ‘수정’과 ‘삭제’ 버튼이 나타난다.
- 사용자는 댓글창에서 댓글을 입력할 수 있다.
- 챗봇 링크는 화면 하단에 위치한다.

■ 게시글 조회 화면(글 작성자가 아닌 경우)



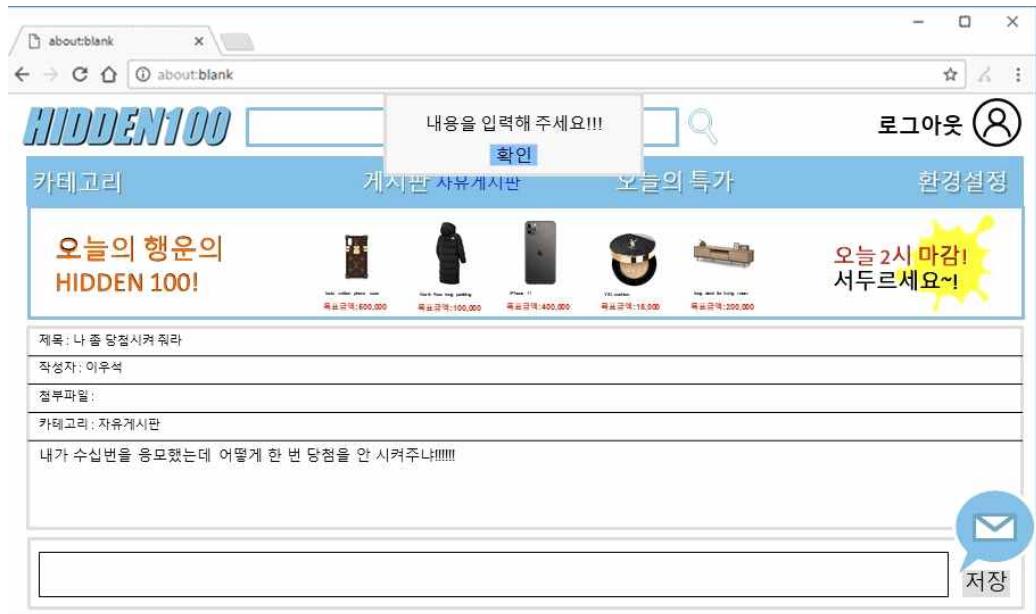
- 게시글 위편에 ‘수정’과 ‘삭제’ 버튼이 나타나지 않는다.
- 사용자는 댓글창에서 댓글을 입력할 수 있다.
- 챗봇 링크는 화면 하단에 위치한다.

■ 댓글 입력 화면(로그인 상태가 아닌 경우)



- 댓글입력을 시도하면 ‘로그인이 필요한 서비스입니다’ 경고 메시지가 나타난다.
- 경고 메시지의 ‘확인’ 버튼을 클릭하면 로그인 화면으로 이동한다.
- 사용자는 로그인 상태인 경우에만 댓글작성이 가능하다.

■ 댓글 입력 화면(댓글 작성칸이 공란인 경우)



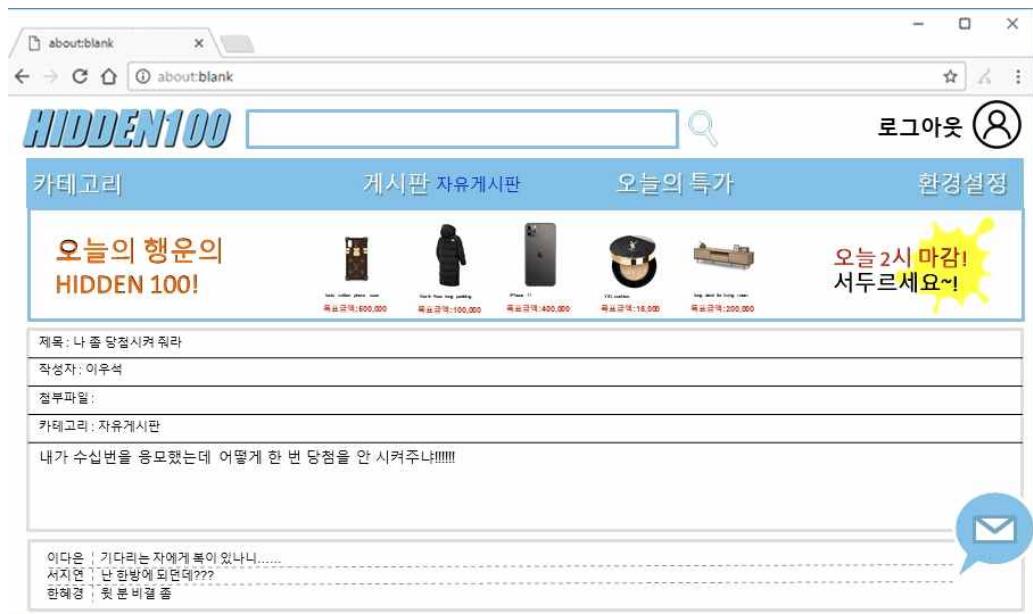
- 댓글 작성칸이 비어있는 경우 ‘저장’ 버튼을 클릭하면 ‘내용을 입력해 주세요’ 경고 메시지가 나타난다.
- 경고 메시지의 ‘확인’ 버튼을 클릭하면 현재 페이지에 머무른다.

■ 댓글 조회 화면(댓글 작성자인 경우)



- 사용자는 자신이 작성한 댓글에 대해서 수정 혹은 삭제가 가능하다.
- 사용자는 자신이 작성한 댓글 옆의 ‘수정’과 ‘삭제’ 버튼을 클릭하여 원하는 기능을 이용할 수 있다.

■ 댓글 조회 화면(댓글 작성자가 아닌 경우)



- 사용자는 자신이 작성한 댓글이 아니면 수정 혹은 삭제가 불가능하다.
- 사용자는 자신이 작성한 댓글이 아니면 ‘수정’과 ‘삭제’ 버튼을 볼 수 없다.

■ 게시물 검색 결과 화면(검색창이 공란인 경우)

번호	제목	작성자	작성일	조회수
1	나 당첨 좀 시켜주세요	이우석	2019-11-30	0
2	노트북 급처합니다!!!	한혜경	2019-10-20	40
3	후드티 팔아요	이다은	2019-10-30	129
4	아이폰 6s 삽니다~~~~~	서지연	2019-09-20	234

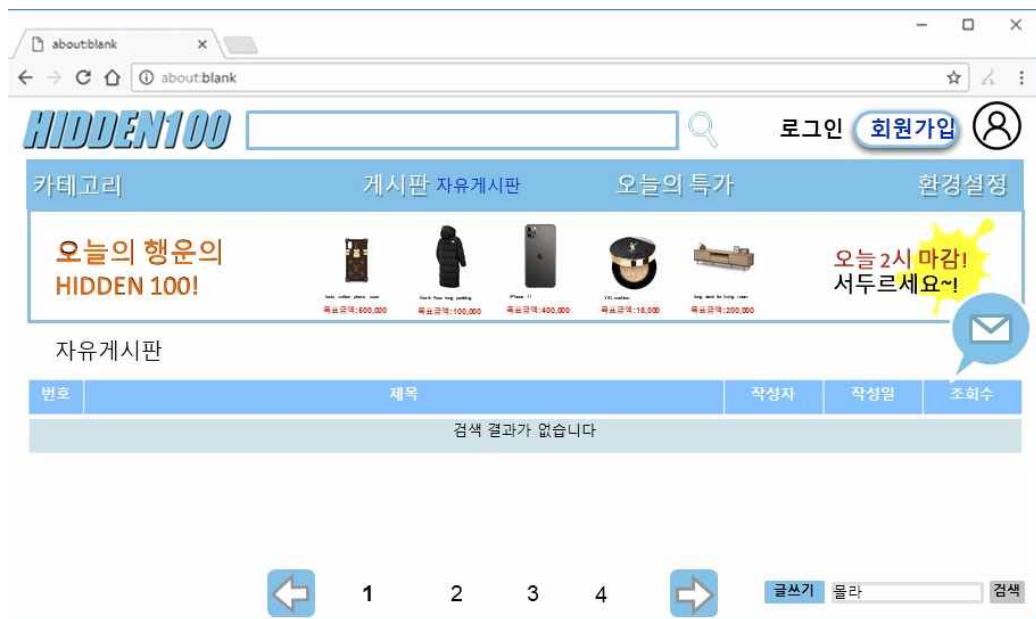
- 검색창이 비어있는 상태에서 ‘검색’ 버튼을 클릭하면 ‘한 글자 이상 입력해주세요’ 경고 메시지가 나타난다.
- 경고 메시지의 ‘확인’ 버튼을 클릭하면 현재 페이지에 머무른다.

■ 게시물 검색 결과 화면(일치하는 게시글이 존재하는 경우)

번호	제목	작성자	작성일	조회수
1	아이폰 6s 삽니다~~~~~.	이우석	2019-11-30	0
2	아이폰 내놓습니다.	한혜경	2019-10-20	40
3	아이폰 중고로 살까 신상 살까	이다은	2019-10-30	129
4	요즘 아이폰 별로인 듯	서지연	2019-09-20	234

- 검색창에 입력한 단어와 일치하는 게시글이 존재하면 일치하는 게시글만 보여준다.
- 검색한 단어와 일치하는 단어에는 강조 처리해준다.

■ 게시물 검색 결과 화면(일치하는 게시글이 존재하지 않는 경우)



- 검색창에 입력한 단어와 일치하는 게시글이 존재하지 않으면 ‘검색 결과가 없습니다’가 나타나고 아무 게시글도 보여지지 않는다.

■ 리뷰게시판

번호	제목	작성자	작성일	조회수
1	단돈 5000원에 아이폰11pro 득템ㅎㅎ	이우석	2019-11-30	0
2	홀길동 판매자님 감사합니다😊	한혜경	2019-10-20	40
3	10000원에 룽패딩 샀다:)	이다은	2019-10-30	129
4	드디어 첫 당첨!!!!!!신 이시어!!!!!!	서지연	2019-09-20	324

- ‘게시판’에서 ‘리뷰게시판’을 선택한 페이지이다.
- ‘게시판’ 옆에 ‘리뷰게시판’ 텍스트가 나타난다.
- 화면 중앙의 리뷰게시판 옆에 ‘리뷰 남기면 포인트 지급!!!’ 문장이 보여진다.
- 챗봇 링크의 위치가 화면 하단에서 중간으로 이동한다.
- 사용자는 글쓰기 버튼을 클릭해서 게시판에 글작성이 가능하다.
- 사용자는 텍스트 창에 단어를 입력해서 원하는 글을 검색할 수 있다.
- 페이지네이션 기능을 이용하여 페이지 이동이 가능하다.

■ 문의게시판

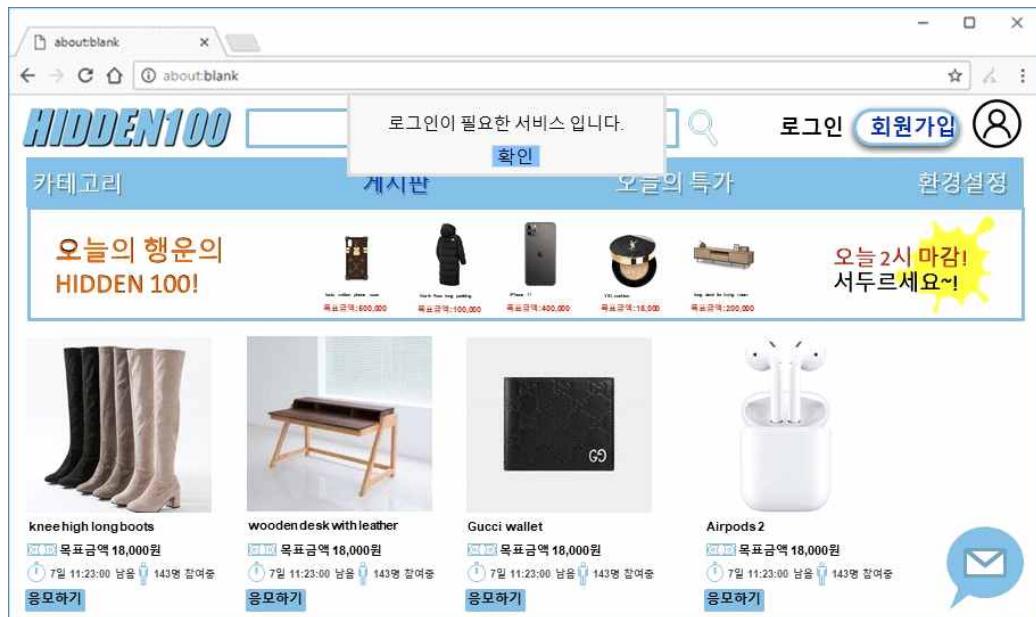
번호	제목	작성자	작성일	조회수
1	배송 일자 문의	이우석	2019-11-30	0
2	환불 되나요???	한혜경	2019-10-20	40
3	배송 조회 좀.....	이다은	2019-10-30	129
4	회원 탈퇴 방법??	서지연	2019-09-20	234

- ‘게시판’에서 ‘문의게시판’을 선택한 페이지이다.
- ‘게시판’ 옆에 ‘문의게시판’ 텍스트가 나타난다.
- 챗봇 링크의 위치가 화면 하단에서 중간으로 이동한다.
- 사용자는 글쓰기 버튼을 클릭해서 게시판에 글작성이 가능하다.
- 사용자는 텍스트 창에 단어를 입력해서 원하는 글을 검색할 수 있다.
- 페이지네이션 기능을 이용하여 페이지 이동이 가능하다.

■ 신고게시판

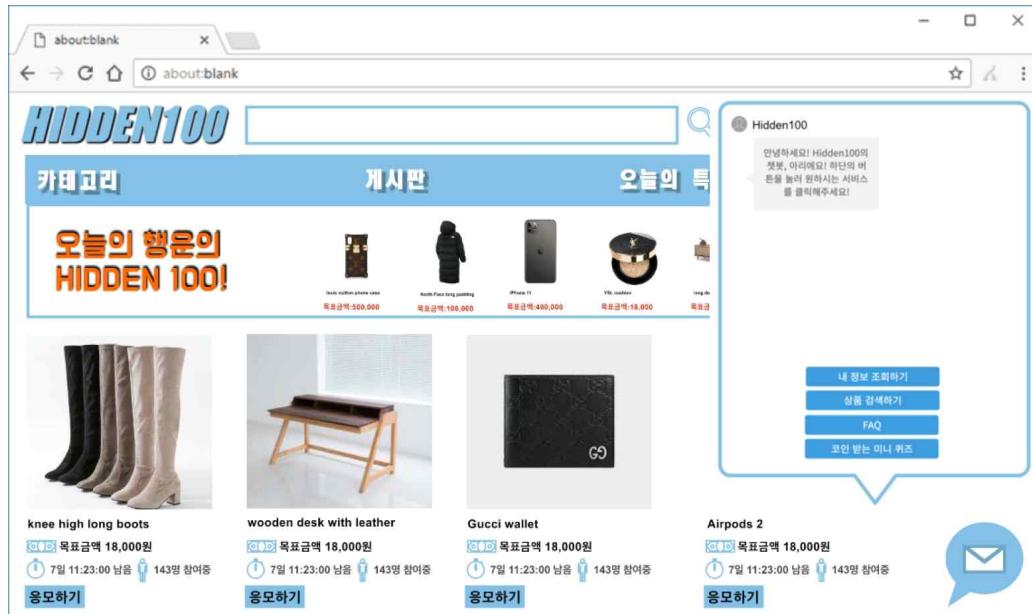
번호	제목	작성자	작성일	조회수
1	돈 떼임ㄷㄷ	이우석	2019-11-30	0
2	신고합니다.	한혜경	2019-10-20	40
3	물건이 안 와요	이다은	2019-10-30	129
4	물건 상태가.....	서지연	2019-09-20	234

- ‘게시판’에서 ‘신고게시판’을 선택한 페이지이다.
- ‘게시판’ 옆에 ‘거래 사기 신고’ 텍스트가 나타난다.
- 챗봇 링크의 위치가 화면 하단에서 중간으로 이동한다.
- 사용자는 글쓰기 버튼을 클릭해서 게시판에 글작성이 가능하다.
- 사용자는 텍스트 창에 단어를 입력해서 원하는 글을 검색할 수 있다.
- 페이지네이션 기능을 이용하여 페이지 이동이 가능하다.

■ 챗봇 사용 화면(로그인 상태가 아닌 경우)

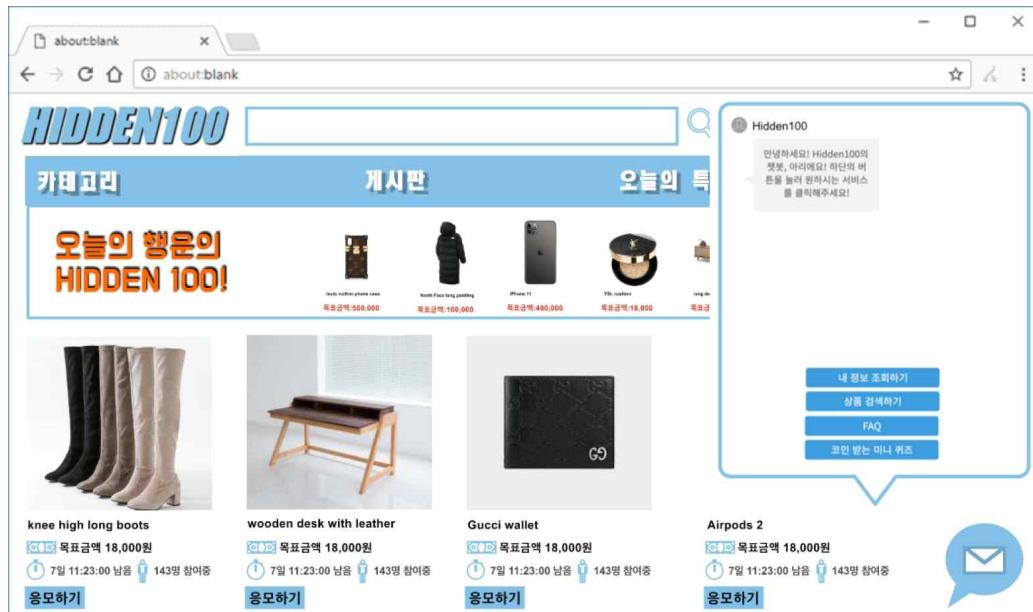
- 사용자가 로그인 상태가 아닌 경우 챗봇 서비스 이용을 시도하면 ‘로그인이 필요한 서비스입니다’ 경고 메시지가 나타난다.
- 사용자는 로그인 상태에서만 챗봇 서비스를 이용할 수 있다.

■ 챗봇 사용 화면(로그인 상태인 경우)



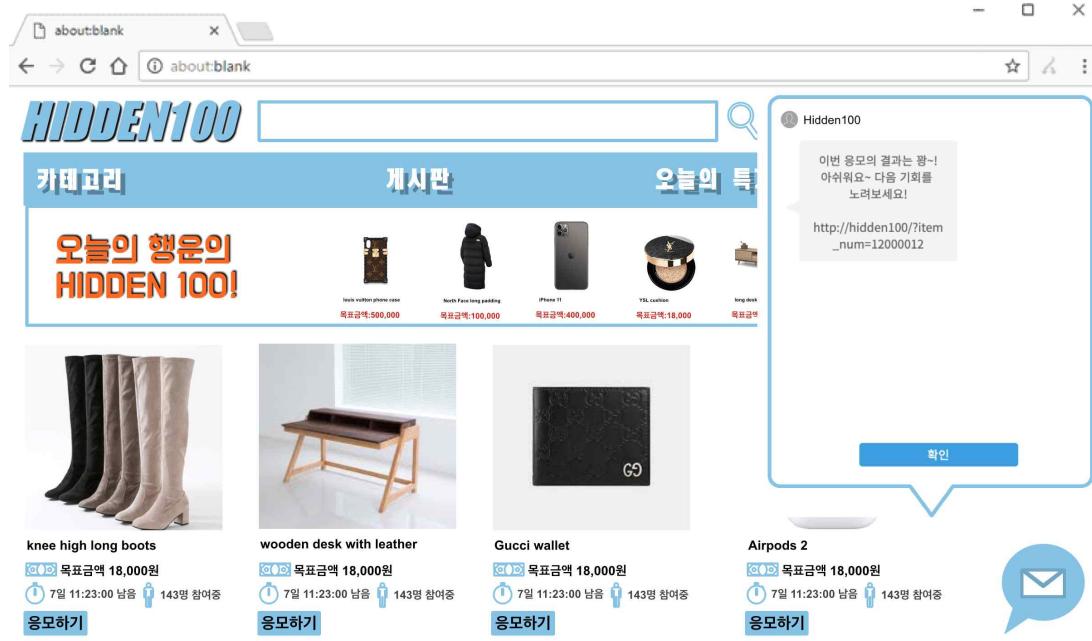
- 챗봇 서비스를 클릭하면 화면과 같은 UI가 나타난다.
- 원하는 내용의 선택지를 선택하면 메시지 피드가 업데이트 된다.
- 챗봇은 사용자의 요청에 따른 적절한 응답을 메시지 피드에 보여준다.

■ 챗봇 사용 화면(내 정보 조회)



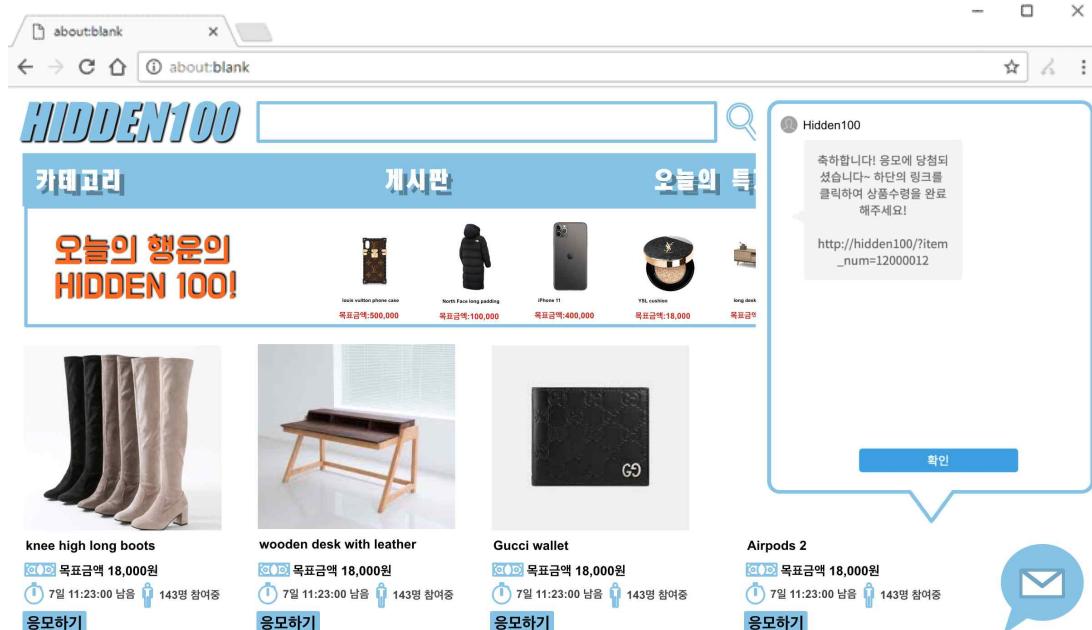
- 챗봇 서비스를 통해 내 정보를 조회 할 수 있다.

■ 챗봇 사용 화면(미당첨인 경우)



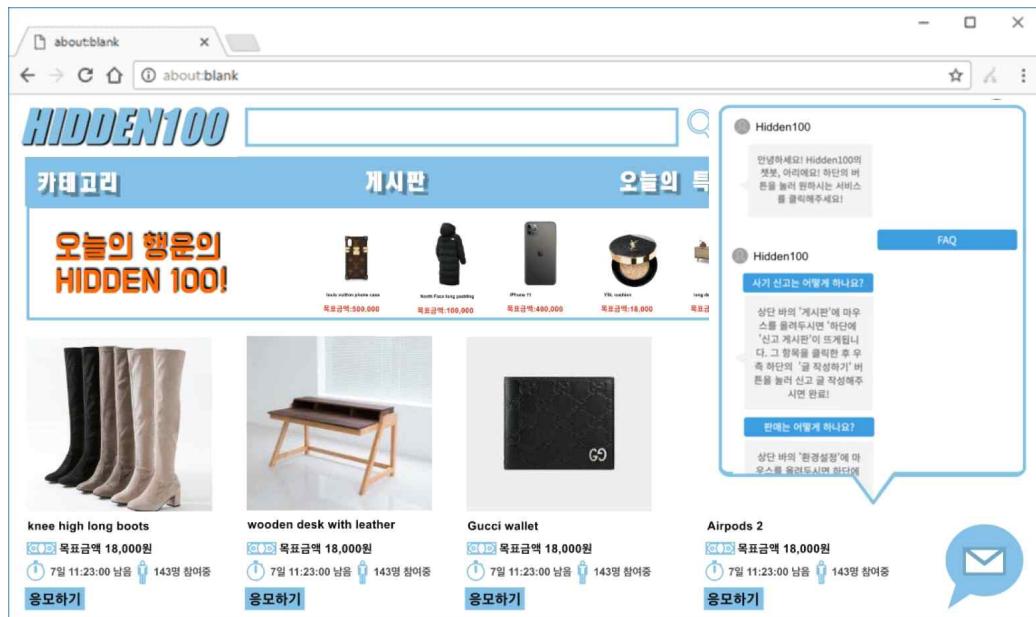
- 미당첨 응모결과가 챗봇 메시지 피드에 나타난다.
- 챗봇이 전송한 메시지를 통해 사용자는 어떤 상품에 응모한 결과인지 알 수 있다.

■ 챗봇 사용 화면(당첨인 경우)



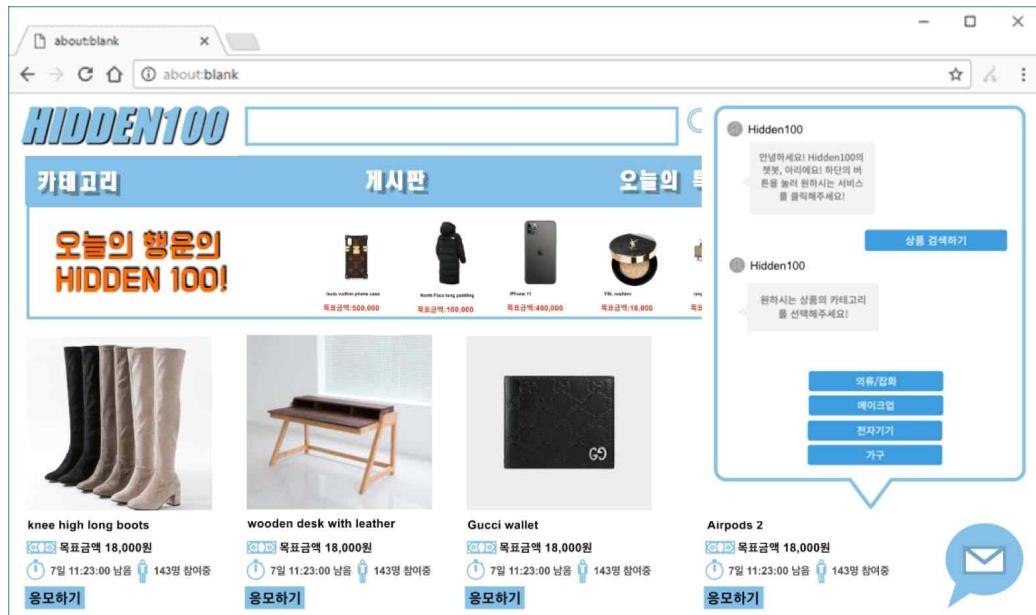
- 당첨 응모결과가 챗봇 메시지 피드에 나타난다.

- 챗봇이 전송한 메시지를 통해 사용자는 어떤 상품에 응모한 결과인지 알 수 있다.
- 챗봇 사용 화면(FAQ 조회 화면)



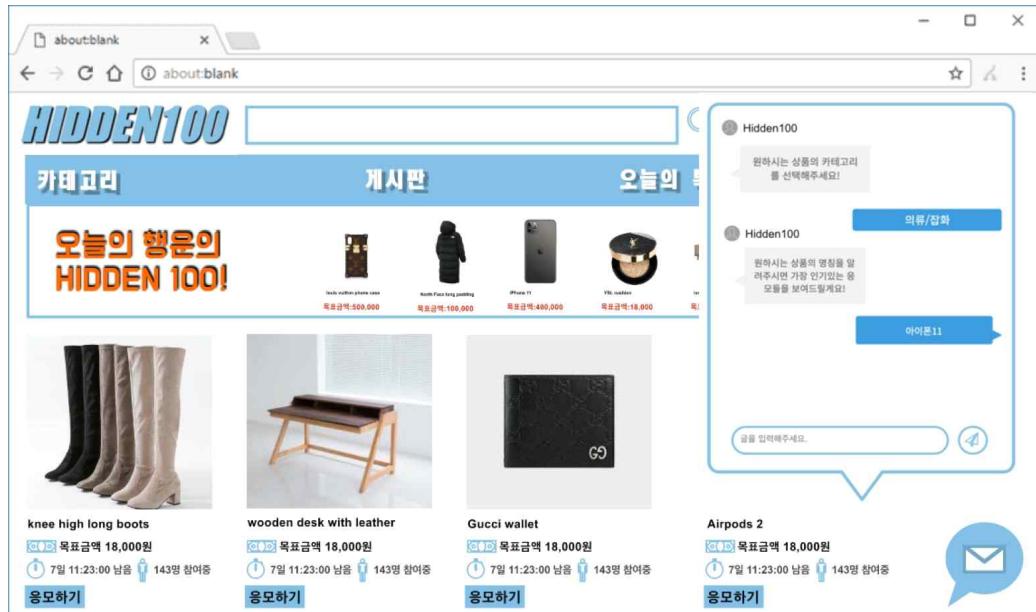
- 챗봇의 FAQ 기능을 통해 사용자는 질의응답을 할 수 있다.

■ 챗봇 사용 화면(상품 검색)



- 챗봇 서비스를 클릭하면 화면과 같은 UI가 나타난다.
- 판매 중인 상품을 검색할 수 있다.
- 상품의 카테고리 버튼을 클릭하면 화면과 같은 UI가 나타난다.

■ 챗봇 사용 화면(상품 검색)



- 카테고리를 선택하여 상품 검색을 할 수 있다.

■ 사용자 환경 설정 초기 화면(사용자 정보수정)



- ‘환경설정’을 선택한 페이지이다.
- ‘환경설정’ 텍스트의 색이 변경된다.
- 사용자는 해당 페이지에서 자신의 정보를 수정할 수 있다.
- 사용자는 내 정보, 코인, 거래내역, 배송 조회 등을 할 수 있다.

■ 사용자 환경 설정 화면(정보수정을 했을 때)



- 사용자가 개인정보를 수정하고 ‘수정 완료’ 버튼을 클릭하면 ‘수정 완료되었습니다’ 메시지가 화면 상단에 나타난다.
- 사용자가 입력한 텍스트대로 데이터베이스의 회원 테이블도 업데이트 된다.

■ 사용자 환경 설정 화면(회원 탈퇴)



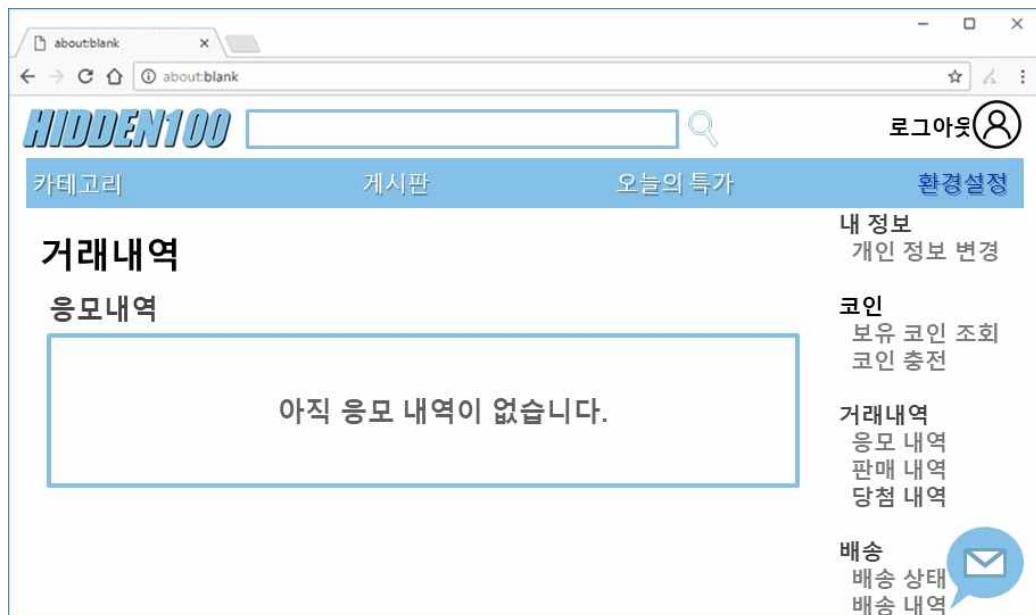
- 사용자가 화면 중앙하단의 ‘탈퇴하기’ 버튼을 누르면 ‘정말 탈퇴하시겠습니까?’ 알림 메시지가 나타난다.
- 사용자가 ‘예’를 누르면 ‘탈퇴 완료되었습니다’ 알림 메시지가 나타난다.
- 사용자가 ‘아니요’를 누르면 현재 페이지에 머무른다.

■ 사용자 환경 설정 화면(회원 탈퇴 되었을 때)



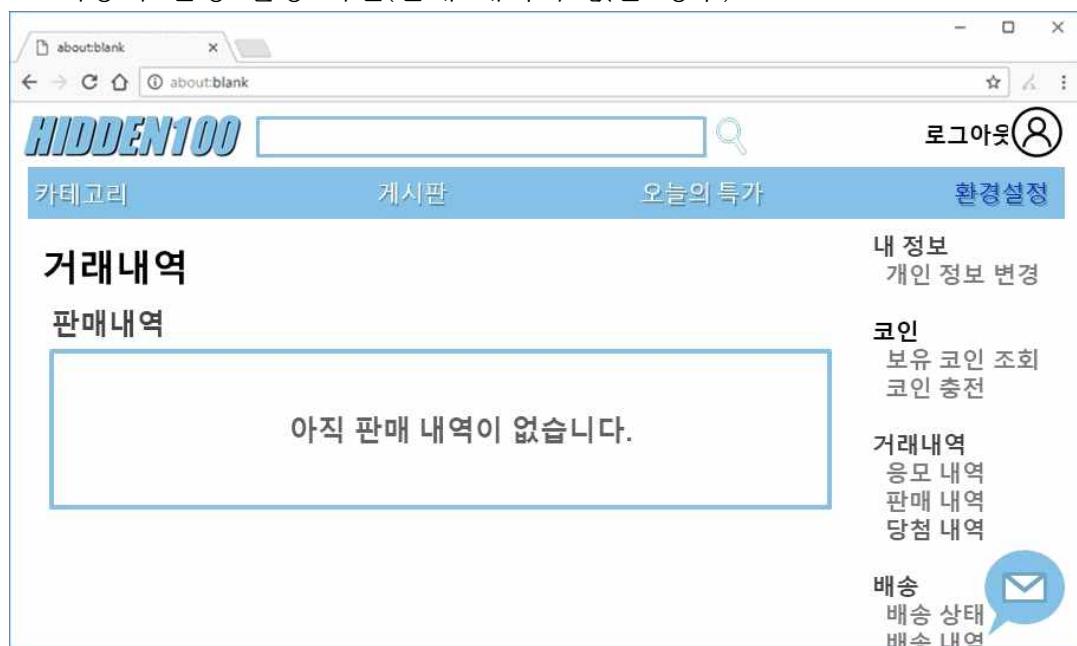
- 사용자가 회원 탈퇴를 하면 ‘탈퇴 완료되었습니다’ 알림 메시지가 나타난다.
- 사용자가 ‘예’ 버튼을 클릭하면 웹 사이트의 메인 페이지로 돌아간다.

■ 사용자 환경 설정 화면(응모 내역이 없는 경우)



- 사용자가 어떤 거래도 하지 않은 경우 응모내역과 판매내역에 각각 ‘아직 응모 내역이 없습니다’와 ‘아직 판매 내역이 없습니다’ 텍스트가 나타난다.

■ 사용자 환경 설정 화면(판매 내역이 없는 경우)



■ 사용자 환경 설정 화면(응모내역이 있는 경우)



- 사용자가 최소 하나의 응모라도 한 경우 응모내역에 상품응모리스트가 나타난다.



- 사용자가 최소 하나의 판매라도 한 경우 판매내역에 판매상품리스트가 나타난다.

- 사용자 환경 설정 화면(판매내역이 있는 경우)
- 사용자 환경 설정 화면(당첨 내역 조회)



- 사용자는 자신이 당첨 되었던 내역을 확인 할 수 있다.
- 물건을 수령하고, 이상이 없다면 수령확인 버튼을 눌러야 한다.
- 배송지를 수정할 수 있다.

■ 배송정보 조회 화면(배송 상태)



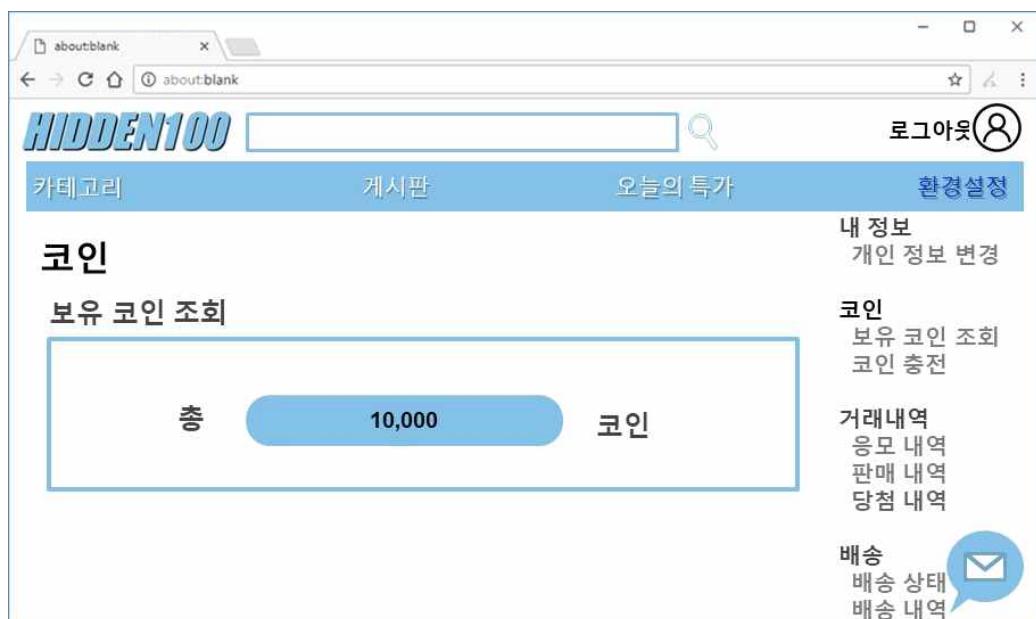
- 사용자는 거래 성사된 상품에 한하여 배송 상태를 조회할 수 있다.
- 판매자는 물건을 배송 붙이고 난 후 배송 정보(송장 번호 등)을 입력해야 한다.

■ 배송정보 조회 화면(배송 내역)



- 판매자는 자신의 상품이 거래 완료되고 배송한 내역을 확인 할 수 있다.

■ 코인 조회 화면



- 사용자는 자신이 보유하고 있는 코인을 조회할 수 있다.

■ 코인 충전 화면



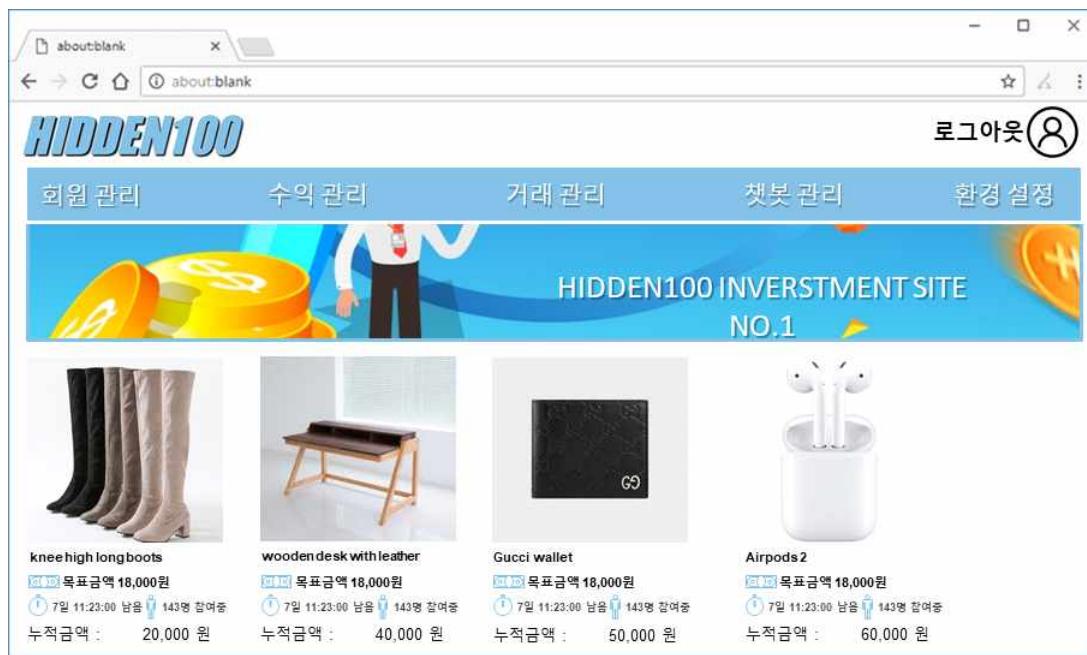
- 사용자는 자신이 보유하고 있는 코인을 증가시키기 위해 코인을 충전할 수 있다.
- 사용자는 충전할 만큼의 금액을 입력해서 코인을 충전할 수 있다.

■ 코인 충전 화면(입력 텍스트 창에 금액을 입력했을 때)

- 사용자는 입력 텍스트 창에 충전할 만큼의 금액을 입력하여 코인을 충전할 수 있다.
- 사용자가 입력한 금액에 따라 보유 예정액의 숫자가 자동으로 업데이트 된다.

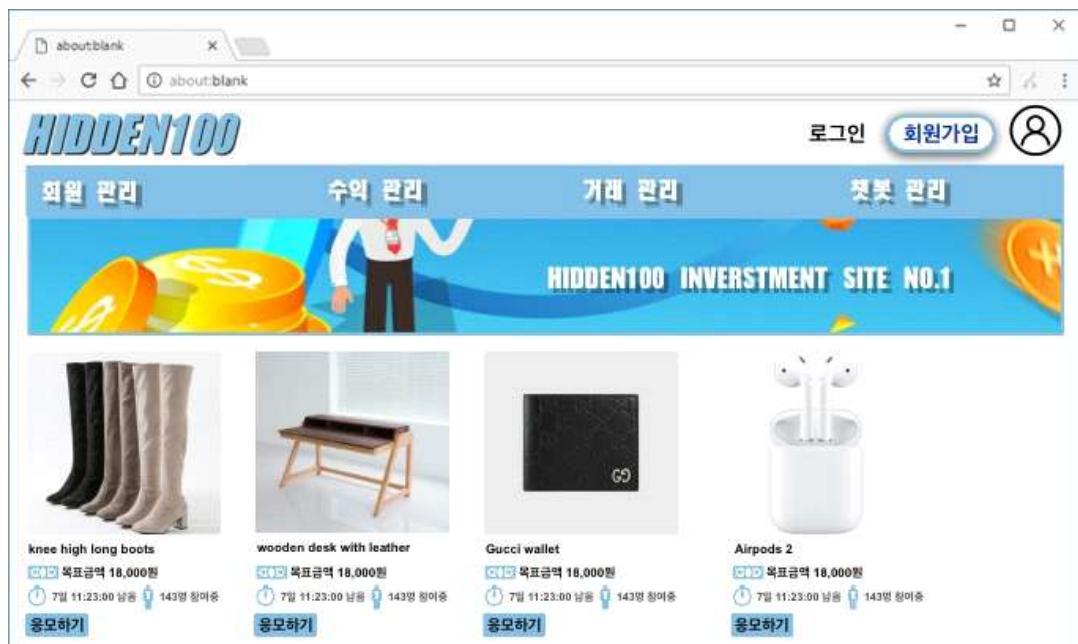
4.4. 관리자UI 설계

■ 관리자 기본 페이지



- 관리자가 관리자 인증을 마치면 확인하는 페이지이다.
- 사용자와 달리 ‘응모하기’ 대신 누적 금액을 확인할 수 있다.
- 상단바 메뉴를 통해 회원관리, 수익관리, 거래관리, 챗봇관리, 환경설정을 수행한다.
- 관리자도 로그아웃으로 관리자 기능 사용을 종료할 수 있다.

■ 관리자 페이지



- 관리자도 사용자와 같이 웹페이지에 접속해서 관리자 인증 기능을 수행한다.
- 회원가입과 로그인으로 관리자 인증한다.
- 기본 페이지는 일반 사용자와 관리자가 동일하다.
- 관리자 회원가입은 다른 데이터베이스에 저장된다.

■ 관리자 로그인



- 관리자 기능을 이용하기 위해서는 관리자 로그인이 필요하다.
- 로그인 페이지에서 관리자 아이디와 비밀번호를 입력하면 관리자 로그인이 가능하다.
- '로그인'버튼을 선택하여 데이터를 전달한다.

■ 관리자 로그인(아이디 비밀번호 입력)



- 관리자 아이디와 비밀번호를 입력하고 로그인 버튼을 통해 관리자 인증할 수 있다.
- 관리자 인증에 성공하면 관리자 기능을 이용할 수 있다.
- 관리자 인증을 포함한 관리자 기능은 Web UI로 진행된다.

■ 관리자 회원가입

The screenshot shows a web browser window for the 'about:blank' page. The main content area displays the 'Hidden 100' logo at the top left. To its right is a search bar with a magnifying glass icon. On the far right of the header are '로그인' (Login) and '회원가입' (Sign Up) buttons, with the '회원가입' button being highlighted with a blue oval. Below the header, a blue navigation bar contains five items: '회원 관리', '수익 관리', '거래 관리', '챗봇 관리', and '환경 설정'. The main content area has a light gray background and features a large title '회원가입하기' (Sign Up) in bold black font. Below it are three input fields: 'ID' (아이디를 입력해주세요), 'PW' (비밀번호를 입력해주세요), and 'PW확인' (비밀번호를 다시 입력해주세요). To the right of the 'ID' field is a blue rectangular button labeled '아이디 중복 체크' (Check for Duplicate ID). At the bottom of the form are two buttons: a blue rectangular button labeled '완료' (Complete) and a smaller blue rectangular button labeled '로그인 하러가기' (Go to Login).

- 관리자도 사용자와 같이 웹페이지에서 회원가입을 진행할 수 있다.
- 관리자가 아이디와 패스워드 정보를 입력하여 '완료'버튼을 누르면 회원가입을 신청한다.
- 입력 정보에 오류가 없으면 회원가입 성공으로 넘어간다.

■ 관리자 회원가입(데이터 입력 완료)

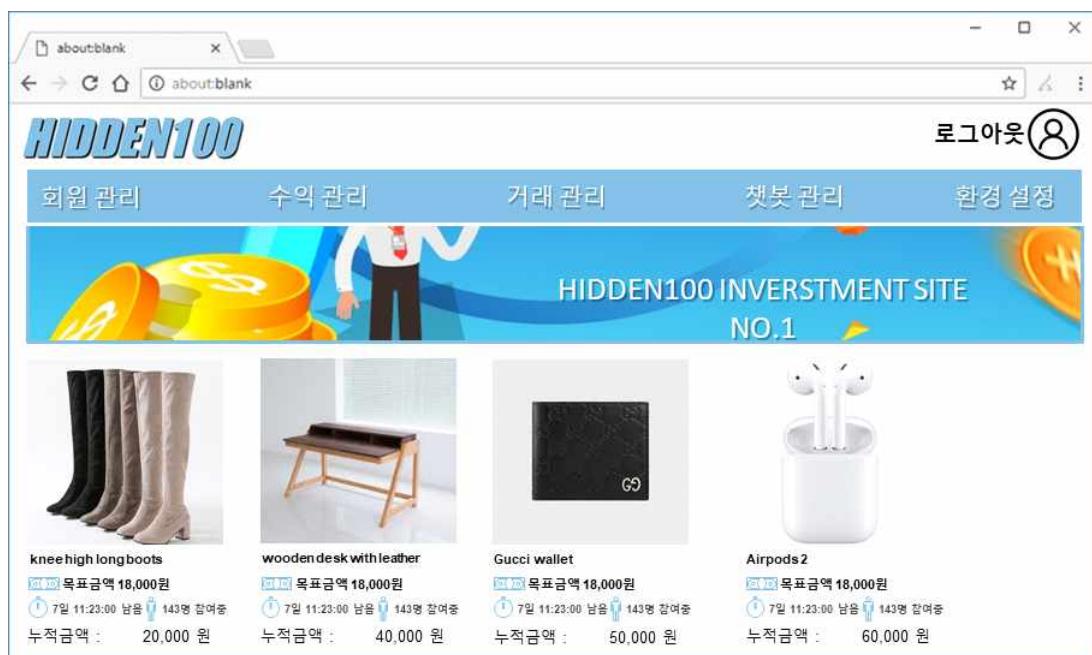
The screenshot shows a web browser window for 'about:blank' with the title 'HIDDEN100'. The main menu bar includes '회원 관리', '수익 관리', '거래 관리', '챗봇 관리', and '환경 설정'. A central modal window displays the message '회원가입이 완료되었습니다.' (Registration completed successfully) with a '로그인하기' (Log in) button. Below this, the '회원가입' (Registration) button in the top right corner of the main menu is highlighted with a blue oval. The main content area is titled '회원가입하기' (Register). It contains three input fields: 'ID' (soonsin), 'PW' (*****), and 'PW확인' (*****). To the right of the ID field is a status message: '아이디 중복 체크' (Id check) and '사용 가능한 아이디입니다.' (Available id). At the bottom are two buttons: a blue '완료' (Complete) button and a link '로그인 하러가기' (Go to log in).

- 관리자도 사용자와 같이 웹페이지에 접속해서 관리자 인증 기능을 수행한다.
- 회원가입과 로그인으로 관리자 인증한다.

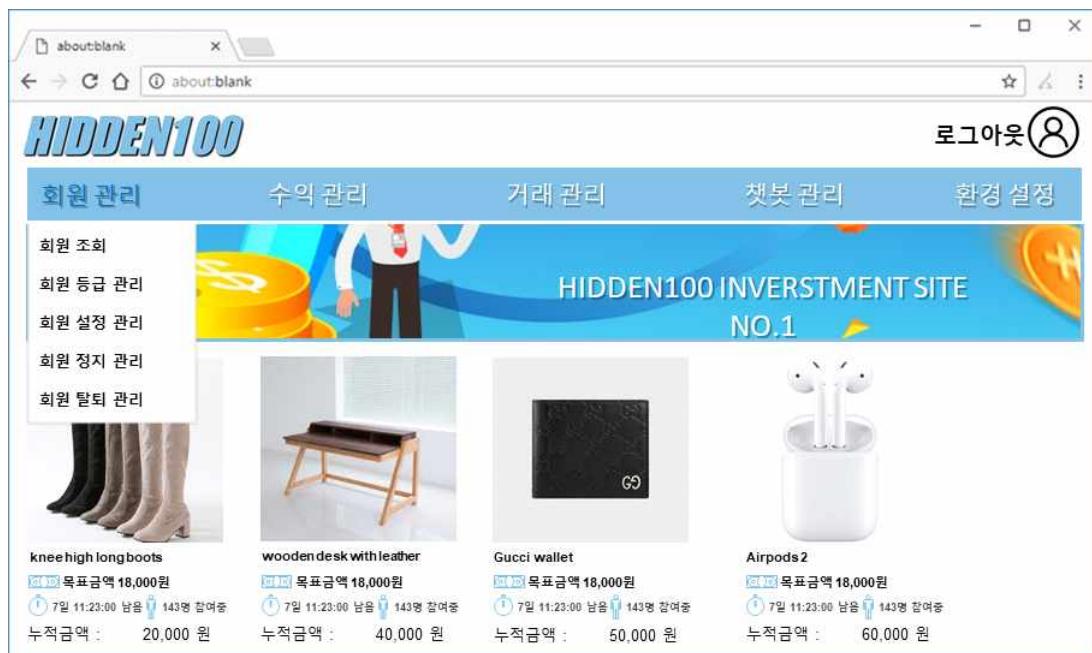
■ 관리자 로그아웃



- 관리자도 사용자와 같이 로그아웃으로 세션에 저장된 관리자 인증 정보를 삭제하고 로그아웃할 수 있다.
- 로그아웃하면 기존의 관리자 기능은 다시 인증해야 이용 가능하다.

■ 관리자 로그인 성공 화면

- 관리자가 페이지에 로그인하면 상단바에서 관리기능을 확인할 수 있다.
- 관리자는 상품에 대해 ‘응모하기’ 대신 현재 응모된 누적 금액을 확인할 수 있다.
- 관리자는 상단바 메뉴를 통해 ‘회원관리, 수익관리, 거래관리, 챗봇관리, 환경설정’ 기능을 수행한다.

■ 관리자 페이지 회원 관리 드롭다운

- 관리자 로그인에 성공하면 상단바에서 관리자 기능을 선택할 수 있다.
- 드롭다운으로 등장한 버튼을 눌러 관리자는 회원관리 기능을 사용한다.
- 관리자는 모든 일반 회원에 대해 ‘회원조회’, ‘회원등급관리’, ‘회원설정 관리’, ‘회원정지 관리’, ‘회원탈퇴 관리’가 가능하다.

■ 회원등급 관리

The screenshot shows a web browser window titled 'about:blank' with the URL 'about:blank'. The page itself is titled 'HIDDEN100' and features a navigation menu with tabs: '회원 관리', '수익 관리', '거래 관리', '챗봇 관리', and '환경 설정'. On the right side of the menu, there is a user profile icon labeled '로그아웃'. Below the menu, the title '회원관리' is displayed, followed by '회원 조회 / 등급 관리'. To the right of this title, it says '총 19,623 명'. A sidebar on the right lists several administrative links: '회원 관리', '회원 조회', '회원 등급 관리', '회원 설정 관리', and '회원 정지 관리'. The main content area displays a table with four rows of member data:

번호	아이디	비밀번호	회원명	주소	전화번호	등급
1	sister01	1321	이다은	경기도 성남시 구미동 영진로 12 112-302	010-2333-4322	5
2	wowow123	wow12345	이우석	경기도 성남시 서현동 연구로 15 123-301	010-3433-6564	3
3	haehae435	hael98766	한혜경	서울 특별시 마포동 연남로 153 503-201	010-2323-6554	4
4	yo1yo1	YOoyo040321	서지연	서울 특별시 강남구 강남로 143-203	010-4343-7654	3

- 관리자는 회원의 등급을 웹페이지에서 변경할 수 있다.
- 회원 정보를 조회하고 등급을 변경한다.
- 등급은 경험치에 따라 자동으로 변경되나 신고가 들어오는 경우 등급을 하락시킬 수 있다.

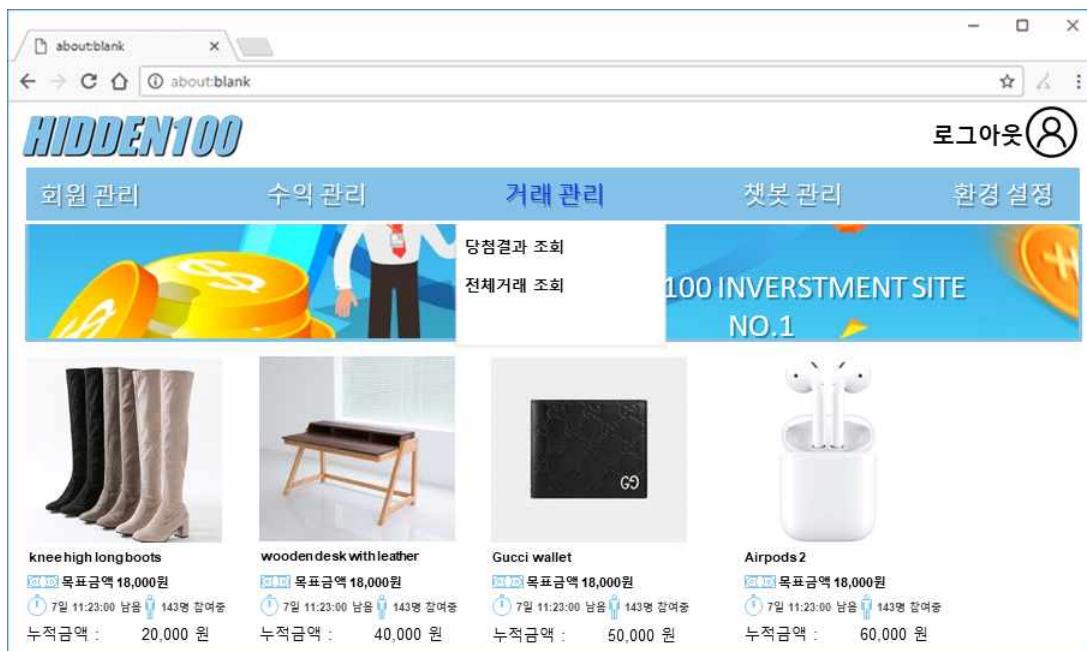
■ 회원정지 관리

The screenshot shows a web browser window titled 'about:blank' with the URL 'about:blank'. The main content area is titled 'HIDDEN100' and features a navigation bar with tabs: '회원 관리' (selected), '수익 관리', '거래 관리', '챗봇 관리', and '환경 설정'. On the right side of the header, there is a '로그아웃' (Logout) button with a user icon. Below the header, the page title is '회원관리' (Member Management). A sub-section title '회원 정지 관리' (Member Suspension Management) is displayed. To the right of this title, there is a summary box showing '총 19,623 명' (Total 19,623 members). Further to the right, a vertical menu lists: '회원 관리', '회원 조회', '회원 등급 관리', '회원 설정 관리', and '회원 정지 관리'. The main content area contains a table with four rows of member data. The columns are: 번호 (Number), 아이디 (ID), 비밀번호 (Password), 회원명 (Member Name), 주소 (Address), 전화번호 (Phone Number), and 등급 (Grade). Each row includes a red '정지하기' (Suspend) button in the last column.

번호	아이디	비밀번호	회원명	주소	전화번호	등급
1	sister01	1321	이다은	경기도 성남시 구미동 영진로 12 112-302	010-2333-4322	5 정지하기
2	wowow123	wow12345	이우석	경기도 성남시 서현동 연구로 15 123-301	010-3433-6564	3 정지하기
3	haehae435	hael98766	한혜경	서울 특별시 마포동 연남로 153 503-201	010-2323-6554	4 정지하기
4	yo1yo1	YOoyo040321	서지연	서울 특별시 강남구 강남로 143-203	010-4343-7654	3 정지하기

- 관리자는 회원의 정지상태를 웹페이지에서 변경할 수 있다.
- 관리자가 정지한 회원은 로그인할 수 없다.
- 등급은 경험치에 따라 자동으로 변경되나 신고가 들어오는 경우 등급을 하락시킬 수 있다.

■ 거래관리화면(호버링)



- ‘거래 관리’부분이 강조 되고 드롭 다운 메뉴가 나타난다.
- 관리자는 마감된 거래의 ‘당첨 결과’와 ‘전체 거래’를 테이블로 확인할 수 있다.

■ 거래관리 화면(전체거래 조회)

The screenshot shows a browser window with the title "about:blank". The main content area has a header "HIDDEN100" and a navigation bar with tabs: 회원 관리, 수익 관리, 거래 관리, 챗봇 관리, and 환경 설정. On the right, there's a sidebar titled "거래 관리" with links: 당첨현황 조회 and 전체거래 조회. The main content area is titled "거래 관리" and "전체거래 조회". It shows a table with the following data:

번호	물품 번호	물품명	응모 가격(원)	판매자	총 응모 인원(명)	총 응모 금액	수수료
1	M120002	Bose Speaker	100,000	이다은	103	103,000	1.5
2	M120003	Gucci Wallet	180,000	이우석	230	230,000	1.5
3	M120004	iPhone	320,000	한혜경	521	210,000	2
4	M120005	Wooden Desk	60,000	서지연	61	61,000	1.5

- 응모 가격에 미달한 상품도 확인할 수 있다.

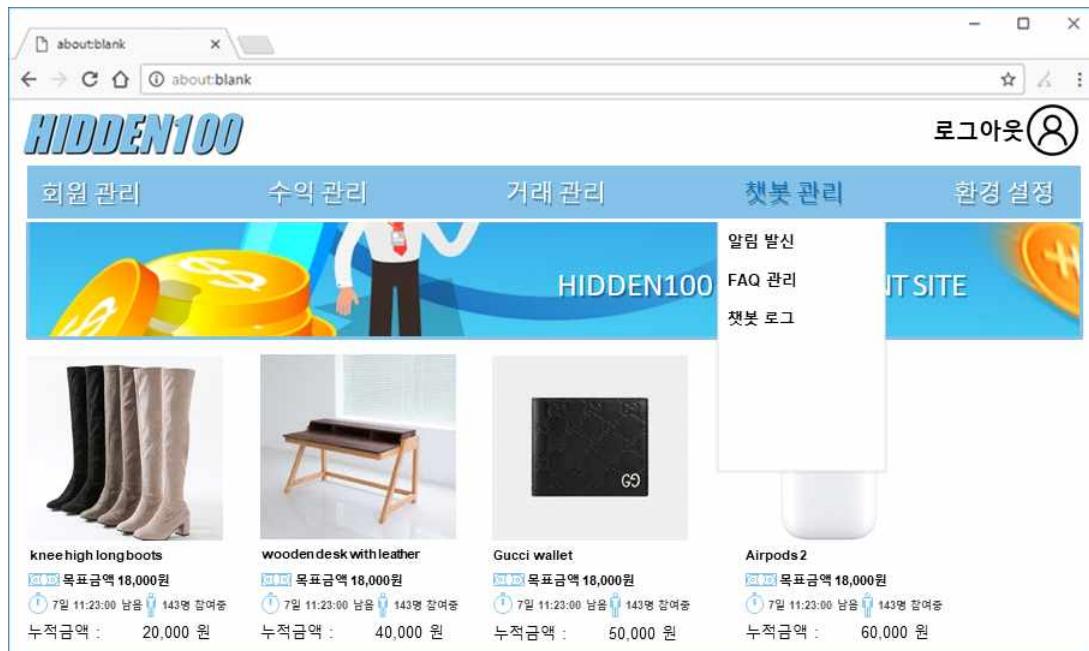
■ 거래관리 화면(당첨현황 조회)

The screenshot shows a web browser window with the title 'about:blank'. The main content area is titled 'HIDDEN100' and features a navigation bar with tabs: 회원 관리, 수익 관리, 거래 관리, 챗봇 관리, and 환경 설정. The '거래 관리' tab is active. On the right side, there is a sidebar with the heading '거래 관리' and two sub-options: '당첨현황 조회' and '전체거래 조회'. The main content area is titled '당첨현황 조회' and shows a table with the following data:

번호	물품 번호	물품명	등록 가격(원)	판매자	총 등록 인원(명)	총 등록 금액	수수료
1	M120002	Bose Speaker	100,000	이다은	103	103,000	1.5
2	M120003	Gucci Wallet	180,000	이우석	230	230,000	1.5
3	M120004	iPhone	320,000	한혜경	521	521,000	2
4	M120005	Wooden Desk	60,000	서지연	61	61,000	1.5

- 성사된 거래를 확인할 수 있다.

■ 챗봇 관리 화면



- 관리자 페이지에서 챗봇관리를 할 수 있다.
- 챗봇으로 알림을 보낼 수 있으며 FAQ 관리, 챗봇 로그 분석이 가능하다.

■ 챗봇 관리 화면(FAQ관리)

The screenshot shows a web browser window with the title 'about:blank'. The main content area displays the 'HIDDEN100' logo and a navigation bar with tabs: 회원 관리, 수익 관리, 거래 관리, 챗봇 관리, and 환경 설정. The '챗봇 관리' tab is active. On the right side of the screen, there is a sidebar with links: 챗봇 관리, 알림 발신, FAQ 관리, and 챗봇 로그. Below these links are two buttons: '추가하기' and '삭제하기'. The central part of the screen contains a table titled 'FAQ 관리' with five rows of data:

Q.	A.
판매 방법?	http://how/to/sell/product 3분 안에 물건 판매 등록 하기!!!
구매 방법?	http://how/to/buy/product 3초만에 응모하기!!!
회원 탈퇴 방법?	http://how/to/quit 진짜 탈퇴하기를 원하십니까?? 아직 기회가 있습니다.
신고 방법?	http://how/to/report 신속히 처리해드립니다.

- 관리자는 FAQ 내용을 수정할 수 있다.
- 질문을 추가하거나 삭제할 수 있다.

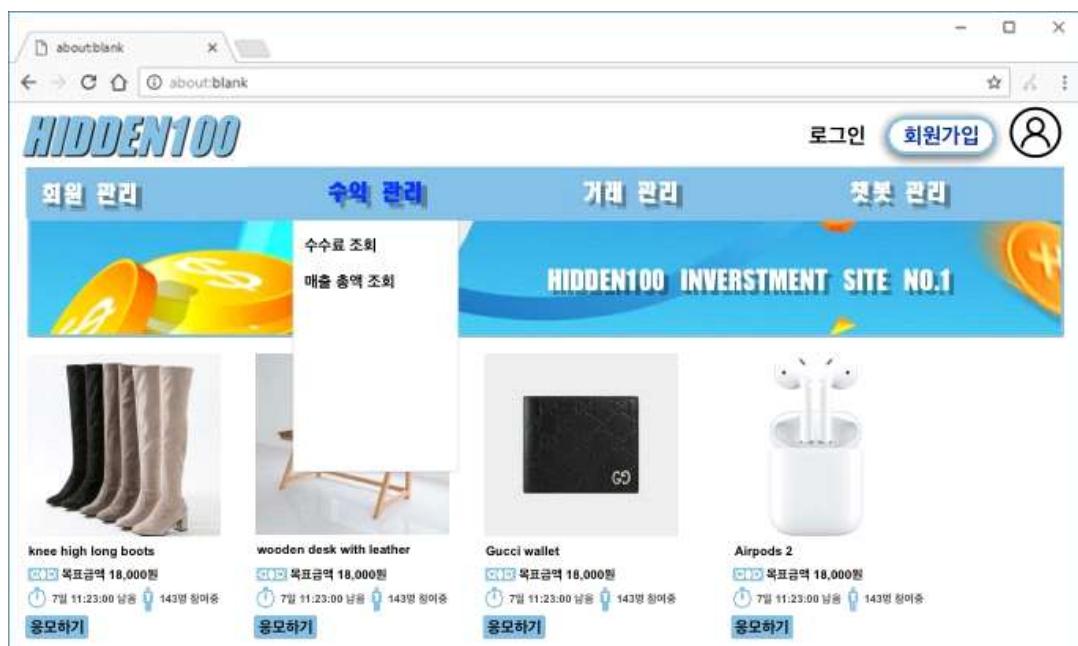
■ 챗봇 관리 화면(알림발신)

- 사용자의 아이디와 메시지 내용을 입력하면 관리자는 챗봇으로 알림을 보낼 수 있다.

■ 챗봇 관리 화면(로그)



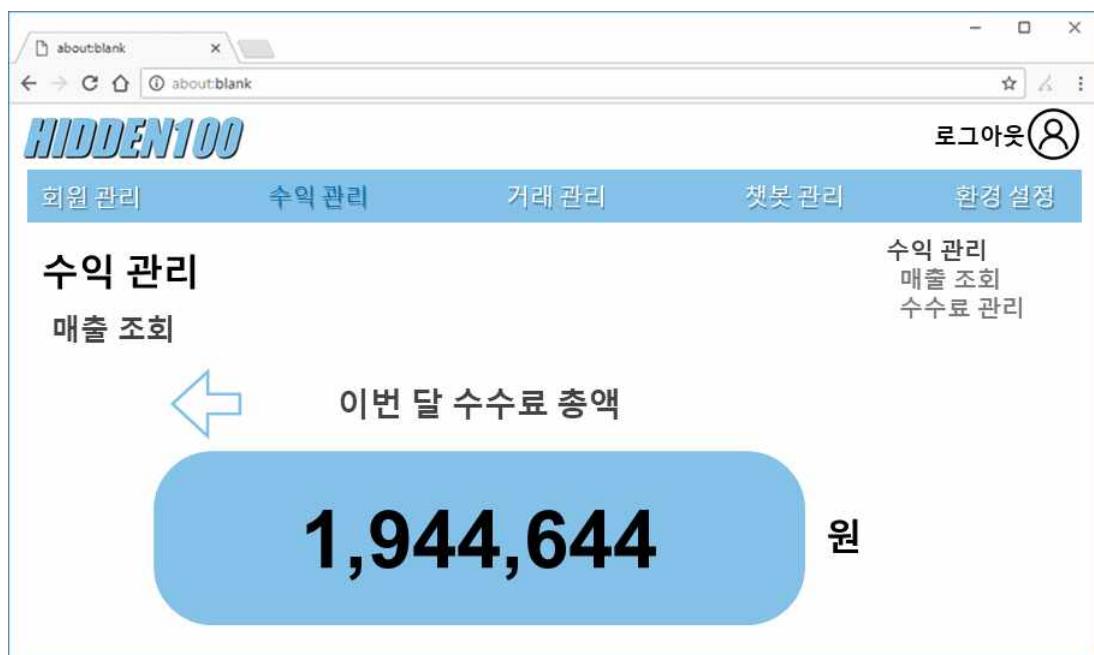
- 사용자가 챗봇으로 가장 많이 검색한 내용을 확인할 수 있다.
- 관리자는 사용자의 관심 추이를 파악할 수 있다.

■ 관리자 메뉴 수익관리 드롭다운

- 수익관리에 마우스를 올리면 드롭다운으로 메뉴가 표시된다.
- 관리자는 ‘수수료 조회’와 ‘매출총액 조회’가 가능하다.

■ 수수료 조회

- 쇼핑몰은 중고거래가 성사된 경우와 코인을 적립할 때 수수료를 받는다.
- 관리자는 수수료를 쉽게 변경할 수 있다.

■ 매출총액 조회

- 거래가 성사된 경우 총 적립액을 기록하고 수수료를 계산한다.
- 관리자는 쉽게 매출액을 확인할 수 있다.

■ 관리자 개인정보 변경



- 관리자 개인정보 변경한다.

■ 관리자 개인정보 변경(확인)

- 주소 정보를 변경하고 수정완료 버튼을 누르면 성공 안내가 뜬다.

5. REST API 인터페이스 설계

5.1. 서버/클라이언트 구조

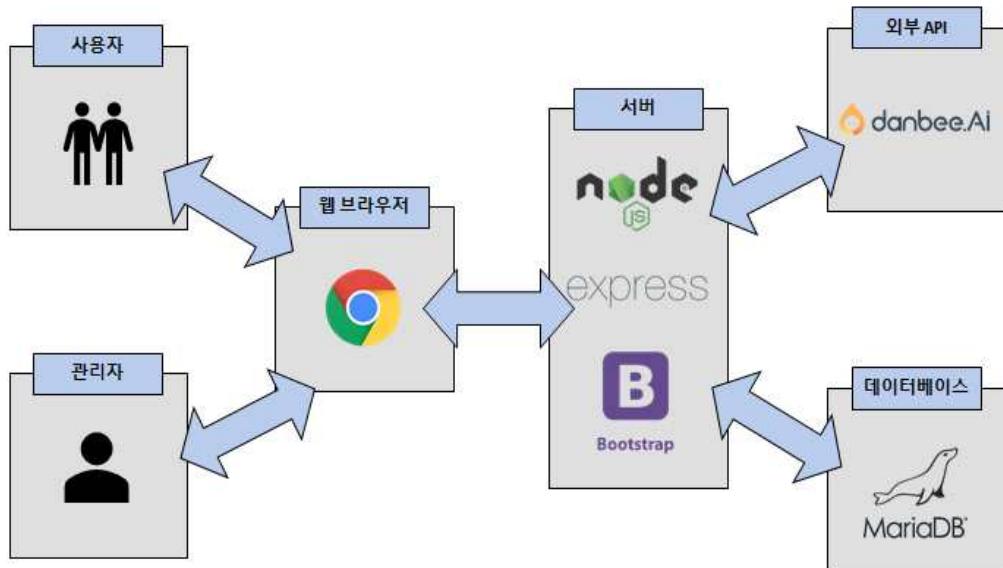


그림 5.1 서비스 인터페이스 구조

본 설계는 그림 5.1과 같이 클라이언트(사용자(판매자, 구매자), 관리자)와 서버(웹 서버)가 웹 브라우저인 구글 크롬을 통해 필요한 정보를 송수신하는 인터페이스 전송 구조이다. 즉, 클라이언트가 서버에게 정보를 요청하거나 응답하는 방식으로 REST API를 이용하여 설계한다.

Hidden 100의 메인 서버는 Node.js, Express, Bootstrap 등 내부에 여러 모듈로 구성되어 있으며, 데이터베이스는 MariaDB를 사용한다. 그리고 외부 API인 danbee.AI를 통해 사용자에게 챗봇 서비스를 제공한다.

클라이언트가 서버에게 요청하는 메시지 타입은 크게 4가지인 조회(GET), 생성(POST), 수정(PUT), 삭제(DELETE)로 구분하고, 각 요청별로 알맞은 REST API 방식으로 메시지를 설계한다. 클라이언트와 서버의 통신을 원활히 설계하기 위해 다음과 같이 각 연동별로 구분하여 REST API를 정의한다.

- 웹 서비스 - 서버와의 연동
- 데이터베이스 - 서버와의 연동

이어서 각 연동별로 REST API를 정의한 후 상세설계를 한다.

5.2. REST API 정의

Method	URI	Description
GET	/user/chatbot/myinfo	내 정보 조회
	/user/chatbot/mini_quiz	미니퀴즈 수신
	/user/chatbot/goods/<goo_name>	판매상품 조회(챗봇)
	/user/chatbot/faq/<option>	FAQ 조회
	/user/goods/buy/all	판매상품 전체 검색(구매자)
	/user/goods/buy/<goo_name>	판매상품 부분 검색(구매자)
	/user/goods/draw	응모내역 조회
	/user/goods/winning	당첨내역 조회
	/user/goods/sell	판매물품 조회(판매자)
	/user/board/<type>/<num>	게시글 조회
	/admin/user/all	회원 전체 조회
	/admin/user/<mem_id>	회원 부분 조회
	/admin/chatbot/log/analysis	챗봇로그 분석
	/admin/goods/draw	응모현황 조회
	/admin/goods/winning	당첨결과 조회
	/admin/goods/deal/all	전체거래 조회(당첨, 미당첨 등 포함)
	/admin/profit	매출 조회
	/admin/board/<type>/<num>	게시글 조회

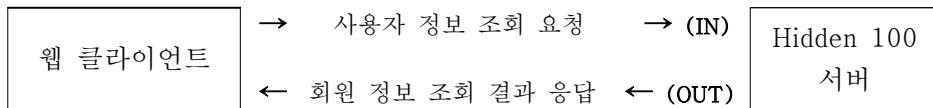
Method	URI	Description
POST	/user/singin	회원가입(사용자)
	/user/goods/register	판매물품 등록
	/user/goods/register/shipment	배송정보 등록
	/user/board/<type>/post	게시글 등록
	/admin/signin	회원가입(관리자)
	/admin/chatbot/message	챗봇알림 발신
	/admin/board/<type>/<num>/post/comment	게시글 댓글 등록

Method	URI	Description
PUT	/user/<userid>/myinfo	본인 정보수정
	/user/coin/charge/<option>	본인 코인충전
	/user/coin/draw/<option>	본인 코인응모
	/user/board/<type>/<num>/put	본인 게시글 수정
	/admin/<userid>/level	회원등급 관리
	/admin/<userid>/userinfo	회원설정 관리
	/admin/<userid>/status	회원정지 관리
	/admin/chatbot/faq	챗봇 FAQ 관리
	/admin/chatbot/mini_quiz	챗봇 미니퀴즈 관리
	/admin/profit/commission	수수료 관리
	/admin/board/<type>/<num>/put/comment	게시글 댓글 수정

Method	URI	Description
DELETE	/user/<userid>/delete	회원탈퇴
	/user/goods/<goo_id>/delete	판매물품 취소
	/user/board/<type>/<num>/delete	본인 게시글 삭제
	/admin/<adminid>/delete	관리자탈퇴
	/admin/board/<type>/<num>/delete	게시글 삭제
	/admin/board/<type>/<num>/comment/delete	게시글 댓글 삭제

5.3. 상세 REST API 설계

GET /user/chatbot/myinfo



Parameter

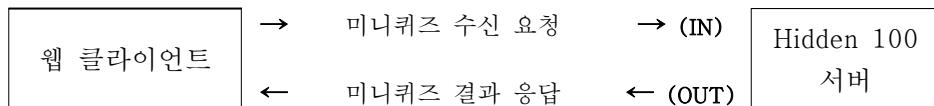
속성	전송방향		Type	Description
	IN	OUT		
userid	O		String	요청한 사용자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	회원 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
  "userid" : "user1234"
}
전송방향 : OUT
{
  "content_type" : "json" ,
  "result_code" : 200 ,
  "result_req" : "" ,
  "user_info" :
  [
    "mem_id" : "user1234" ,
    "mem_pass" : "123456" ,
    "mem_name" : "김하든" ,
    "mem_addr" : "서울특별시 강남구 강남대로 78" ,
    "regist_day" : "2019-12-01" ,
    "coin" : 8000 ,
    "level" : 2 ,
    "point" : 38 ,
    "status" : "활성"
  ]
}
  
```

GET /user/chatbot/mini_quiz



Parameter

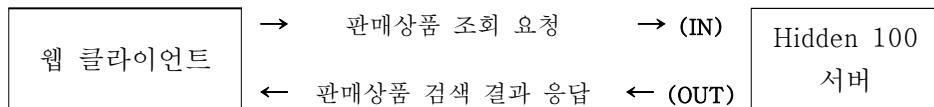
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
mini_quiz_info		O	Array	미니퀴즈 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "mini_quiz_info" :
    [
        "question" : "Hidden 100을 개발한 사람들의 출신 학과는?" ,
        "answer1" : "컴퓨터공학과" ,
        "answer2" : "전자공학과" ,
        "answer_real" : "컴퓨터공학과"
    ]
}
  
```

GET /user/chatbot/goods/<goo_name>



Parameter

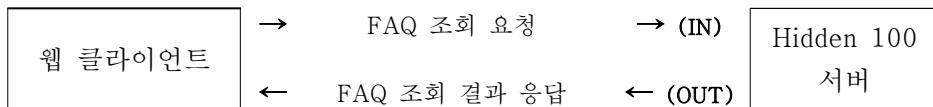
속성	전송방향		Type	Description
	IN	OUT		
goo_name	O		String	요청한 판매상품명
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
goods_info		O	Array	판매상품 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "goods" : "아이폰"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "goods_info" :
    [
        [
            {
                "name" : "아이폰8 64GB 스페이스 그레이" ,
                "type" : "휴대폰",
                "price" : 2000,
                "time_year" : 2019 ,
                "time_month" : 12 ,
                "time_day" : 25 ,
                "time_hour" : 0 ,
                "time_minute" : 0 ,
                "coin" : 500,
                "participation" : 4
            },
            ...
        ]
    }
}
  
```

GET /user/chatbot/faq/<option>



Parameter

속성	전송방향		Type	Description
	IN	OUT		
option	O		String	요청한 FAQ 질문
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
faq_info		O	Array	FAQ 답변 정보

실제 전송내용 (JSON 의 경우라면)

```

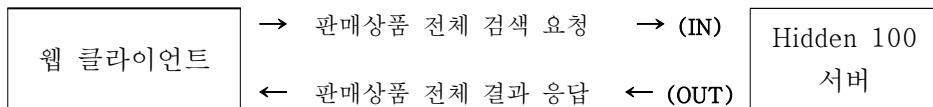
전송방향 : IN
{
    "option" : "판매상품 아이폰 조회해줘"
}
전송방향 : OUT
{
    [
        [
            {
                "name" : "아이폰8 64GB 스페이스 그레이" ,
                "type" : "휴대폰",
                "price" : 2000,
                "time_year" : 2019 ,
                "time_month" : 12 ,
                "time_day" : 25 ,
                "time_hour" : 0 ,
                "time_minute" : 0 ,
                "coin" : 500,
                "participation" : 4
            },
        ]
    }
}
  
```

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100version
v1.0

【P-실무프로젝트】

GET /user/goods/buy/all



Parameter

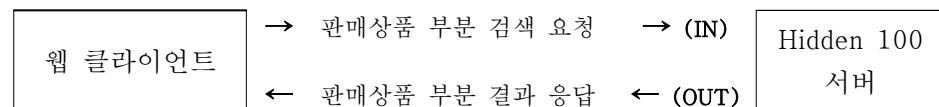
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
goods_info		O	Array	판매상품 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
  [
    [
      {
        "name" : "아이폰8 64GB 스페이스 그레이" ,
        "type" : "휴대폰",
        "price" : 2000,
        "time_year" : 2019 ,
        "time_month" : 12 ,
        "time_day" : 25 ,
        "time_hour" : 0 ,
        "time_minute" : 0 ,
        "coin" : 500,
        "participation" : 4
      },
      ...
    ]
}
  
```

GET /user/goods/buy/<goo_name>



Parameter

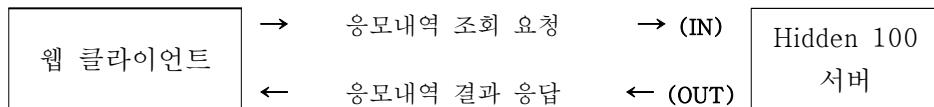
속성	전송방향		Type	Description
	IN	OUT		
goo_name	O		String	요청한 판매상품명
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
goods_info		O	Array	판매상품 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "goo_name" : "아이폰"
}
전송방향 : OUT
{
    [
        [
            {
                "name" : "아이폰8 64GB 스페이스 그레이" ,
                "type" : "휴대폰",
                "price" : 2000,
                "time_year" : 2019 ,
                "time_month" : 12 ,
                "time_day" : 25 ,
                "time_hour" : 0 ,
                "time_minute" : 0 ,
                "coin" : 500,
                "participation" : 4
            },
        ]
    }
  
```

GET /user/goods/draw



Parameter

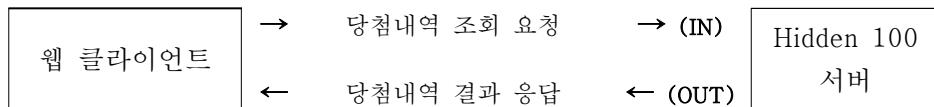
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
draw_info		O	Array	응모내역 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "draw_info" :
    [
        [
            "goo_id" : 39684329 ,
            "goo_name" : "에어팟 프로" ,
            "goo_type" : "전자기기" ,
            "goal_price" : 89000 ,
            "seller_id" : "user6789" ,
            "invest_coin" : 30000 ,
            "invest_day" : "2019-12-02" ,
            "status" : "진행중"
        ],
        ]
}
  
```

GET /user/goods/winning



Parameter

속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
winning_info		O	Array	당첨내역 정보

실제 전송내용 (JSON 의 경우라면)

```

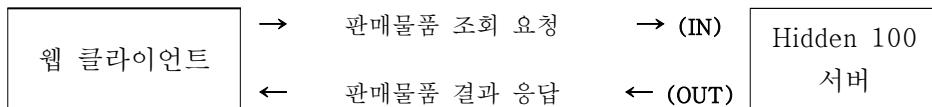
전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "winning_info" :
    [
        [
            "goo_id" : 39684329 ,
            "goo_name" : "에어팟 프로" ,
            "goo_type" : "전자기기" ,
            "goal_price" : 89000 ,
            "seller_id" : "user6789" ,
            "invest_coin" : 30000 ,
            "invest_day" : "2019-12-02" ,
            "status" : "당첨"
        ],
        ]
}
  
```

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100version
v1.0

【P-실무프로젝트】

GET /user/goods/sell



Parameter

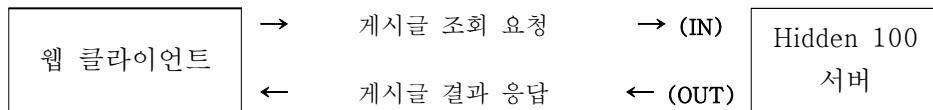
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
goods_sell_info		O	Array	판매물품 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "goods_sell_info" :
    [
        [
            "name" : "아이폰8 64GB 스페이스 그레이" ,
            "type" : "휴대폰",
            "price" : 2000,
            "time_year" : 2019 ,
            "time_month" : 12 ,
            "time_day" : 25 ,
            "time_hour" : 0 ,
            "time_minute" : 0 ,
            "coin" : 500,
            "participation" : 4
        ],
        ]
    }
  
```

GET /user/board/<type>/<num>



Parameter

속성	전송방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
board_info		O	Array	게시글 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "bulletin" ,
    "num" : 5
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "board_info" :
    [
        {
            "mem_id" : "user5678" ,
            "subject" : "나도 당첨돼 보고싶다" ,
            "content" : "몇 번을 시도해도 항상 미당첨이야 슬프다" ,
            "regist_day" : "2019-11-30" ,
            "hit" : 128
        }
    ]
}

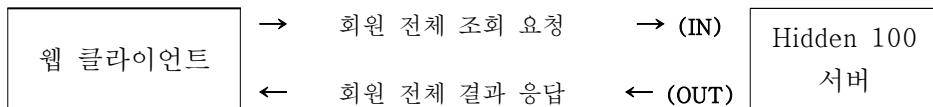
```

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100version
v1.0

【P-실무프로젝트】

GET /admin/user/all



Parameter

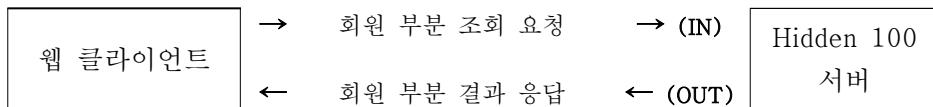
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	회원 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "goods_info" :
    [
        [
            "mem_id" : "user1234" ,
            "mem_pass" : "123456" ,
            "mem_name" : "김하든" ,
            "mem_addr" : "서울특별시 강남구 강남대로 78" ,
            "regist_day" : "2019-12-01" ,
            "coin" : 8000 ,
            "level" : 2 ,
            "point" : 38 ,
            "status" : "활성"
        ] ,
    ]
}
  
```

GET /admin/user/<mem_id>



Parameter

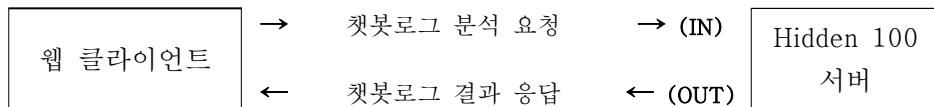
속성	전송방향		Type	Description
	IN	OUT		
mem_id	O		String	요청한 회원 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
user_info		O	Array	회원 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "mem_id" : "user1234"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "deal_active_info" :
    [
        [
            {
                "mem_id" : "user1234" ,
                "mem_pass" : "123456" ,
                "mem_name" : "김하든" ,
                "mem_addr" : "서울특별시 강남구 강남대로 78" ,
                "regist_day" : "2019-12-01" ,
                "coin" : 8000 ,
                "level" : 2 ,
                "point" : 38 ,
                "status" : "활성"
            }
        ]
    }
}
  
```

GET /admin/chatbot/log/analysis



Parameter

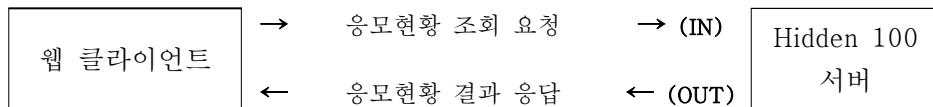
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
chat_log_info		O	Array	챗봇로그 분석 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "chat_log_info" :
        [
            [
                {
                    "num" : 1 ,
                    "mem_id" : "user1234" ,
                    "mem_name" : "김하든" ,
                    "question" : "내 포인트 조회해줘" ,
                    "answer" : 38 ,
                    "use_day" : "2019-12-03"
                },
                ...
            ]
        ]
}
  
```

GET /admin/goods/draw



Parameter

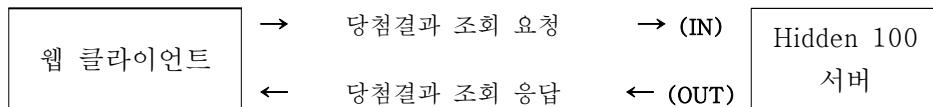
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
draw_info		O	Array	응모현황 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "draw_info" :
    [
        [
            [
                {
                    "goo_id" : 39684329 ,
                    "goo_name" : "에어팟 프로" ,
                    "goo_type" : "전자기기" ,
                    "goal_price" : 89000 ,
                    "seller_id" : "user6789" ,
                    "invest_coin" : 30000 ,
                    "invest_day" : "2019-12-02" ,
                    "status" : "진행중"
                },
                ...
            ]
        ]
    }
}
  
```

GET /admin/goods/winning



Parameter

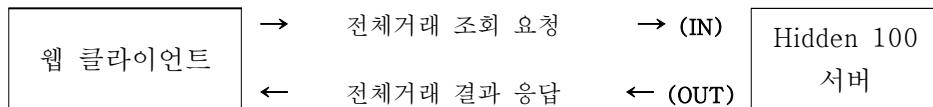
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
winning_info		O	Array	당첨결과 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "winning_info" :
    [
        [
            "mem_id" : "user1234" ,
            "goo_id" : 39684329 ,
            "goo_name" : "에어팟 프로" ,
            "goo_type" : "전자기기" ,
            "goal_price" : 89000 ,
            "seller_id" : "user6789" ,
            "invest_coin" : 30000 ,
            "invest_day" : "2019-12-02" ,
            "status" : "당첨"
        ],
        ]
    }
  
```

GET /admin/goods/deal/all



Parameter

속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
deal_info		O	Array	전체거래 정보

실제 전송내용 (JSON 의 경우라면)

```

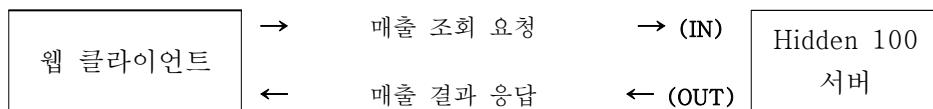
전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "deal_info" :
    [
        [
            "mem_id" : "user2345" ,
            "goo_id" : 39684329 ,
            "goo_name" : "에어팟 프로" ,
            "goo_type" : "전자기기" ,
            "goal_price" : 89000 ,
            "seller_id" : "user6789" ,
            "invest_coin" : 25000 ,
            "invest_day" : "2019-12-03" ,
            "status" : "미당첨"
        ],
        ]
}
  
```

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100version
v1.0

【P-실무프로젝트】

GET /admin/profit



Parameter

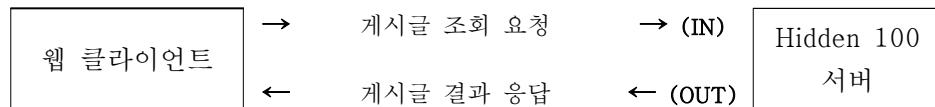
속성	전송방향		Type	Description
	IN	OUT		
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
profit_info		O	Array	매출 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "profit_info" :
    [
        "coin_commission" : 78000 ,
        "goods_commission" : 90000
    ]
}
  
```

GET /admin/board/<type>/<num>



Parameter

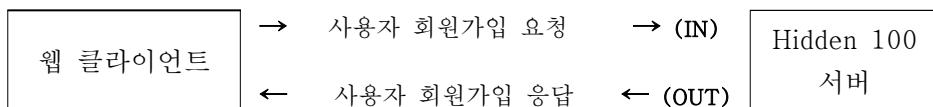
속성	전송방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지
board_info		O	Array	게시글 정보

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "bulletin" ,
    "num" : 5
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
    "board_info" :
    [
        {
            "mem_id" : "user5678" ,
            "subject" : "나도 당첨돼 보고싶다" ,
            "content" : "몇 번을 시도해도 항상 미당첨이야 슬프다" ,
            "regist_day" : "2019-11-30" ,
            "hit" : 128
        }
    ]
}
  
```

POST /user/signin



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
user_info	O		Array	사용자 회원가입 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

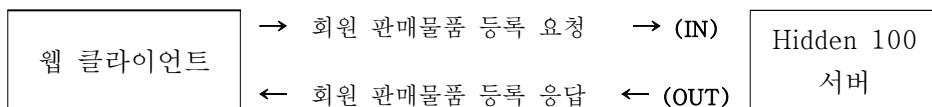
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "user_info" :
    [
        "mem_id" : "user1234" ,
        "mem_pass" : "123456",
        "mem_name" : "김하든" ,
        "mem_addr" : "서울특별시 강남구 강남대로 78"
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
}

```

POST /user/goods/register



Parameter

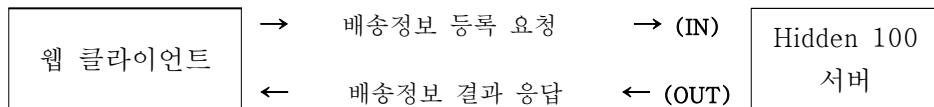
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	회원 아이디
goods_info	O		Array	판매물품 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user5678",
    "goods_info" :
    [
        "name" : "아이폰8 64GB 스페이스 그레이" ,
        "type" : "휴대폰",
        "price" : 2000,
        "time_year" : 2019 ,
        "time_month" : 12 ,
        "time_day" : 25 ,
        "time_hour" : 0 ,
        "time_minute" : 0 ,
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
}
  
```

POST /user/goods/register/shipment



Parameter

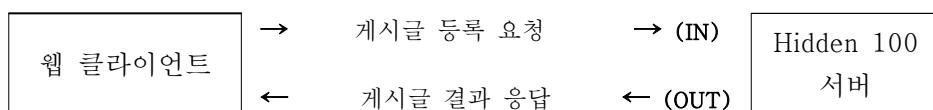
속성	전송 방향		Type	Description
	IN	OUT		
goo_id	O		int	판매물품 고유번호
num_shipment	O		String	운송장번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "goo_id" : 83927839,
    "num_shipment" : "KR-8901"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
}
  
```

POST /user/board/<type>/post

**Parameter**

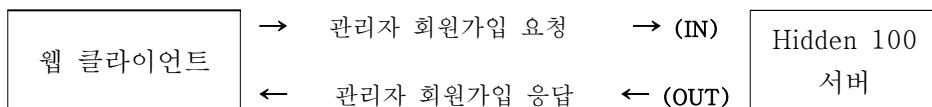
속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	게시글 종류
board_info	O		Array	게시글 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "bulletin",
    "board_info" :
    [
        "subject" : "챗봇기능 좋아요",
        "content" : "당첨결과 바로 알려주는 거 편리하네요"
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

POST /admin/signin



Parameter

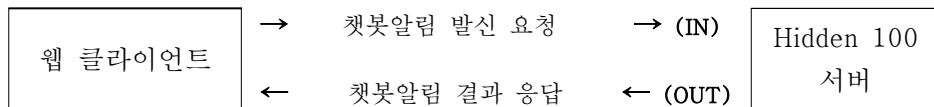
속성	전송방향		Type	Description
	IN	OUT		
admin_info	O		Array	관리자 회원가입 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "admin_info" :
    [
        "admin_id" : "user1234" ,
        "admin_pass" : "123456",
        "admin_name" : "김 히든"
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
}
  
```

POST /admin/chatbot/message



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
chatbot_msg	O		Array	챗봇 메시지
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

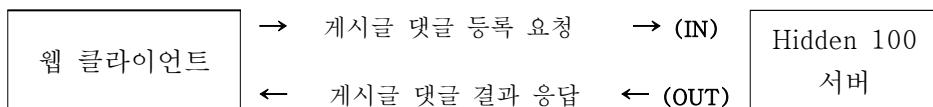
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "chatbot_msg" :
    [
        "mem_id" : "user1234" ,
        "content" : "축하합니다! 응모한 에어팟 프로에 당첨되었습니다!"
    ]
}

전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

POST /admin/board/<type>/<num>/post/comment



Parameter

속성	전송방향		Type	Description
	IN	OUT		
type	O		String	게시판 종류
num	O		int	게시글 번호
comment	O		String	게시글 댓글 내용
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

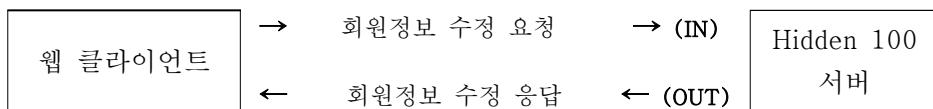
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "question" ,
    "num" : 8 ,
    "comment" : "고객님 발송은 3-7일 이내에 이루어집니다"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}

```

PUT /user/<userid>/myinfo



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
user_info	O		Array	요청한 회원 수정사항 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

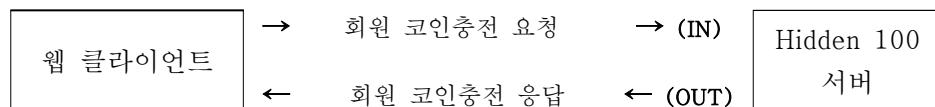
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234"
    "user_info" :
        [
            "mem_pass" : "hidden5678",
            "mem_addr" : "서울특별시 종로구 사직대로 56"
        ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : "" ,
}

```

PUT /user/coin/charge/<option>



Parameter

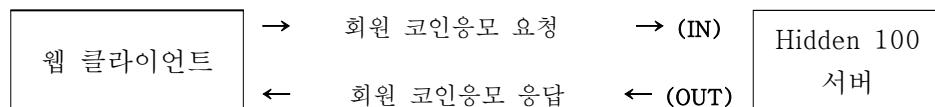
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
option	O		int	요청한 코인충전금액
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234",
    "option" : 15000
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /user/coin/draw/<option>



Parameter

속성	전송방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
goo_id	O		int	요청한 판매상품 고유번호
option	O		int	요청한 코인응모금액
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

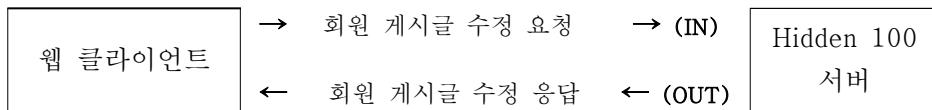
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234",
    "goo_id" : 28396029
    "option" : 3000
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}

```

PUT /user/board/<type>/<num>/put



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
board_info	O		Array	요청한 게시글 수정사항 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

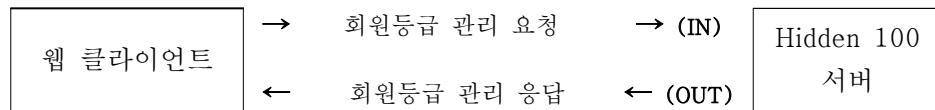
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "review" ,
    "num" : 16 ,
    "board_info" :
    [
        "subject" : "캡 후드티 정말 마음에 들어요" ,
        "content" : "지난주에 응모 당첨돼서 어제 받았어요. 입어보니 더 예쁘네요"
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""

}
  
```

PUT /admin/<userid>/level



Parameter

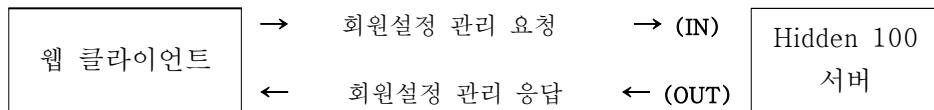
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
level	O		int	요청한 회원 변경등급
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234",
    "level" : 2
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /admin/<userid>/userinfo



Parameter

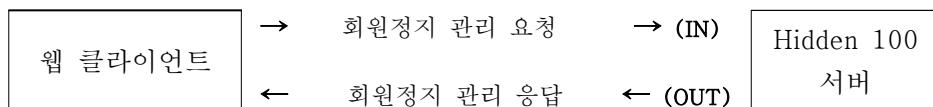
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
user_info	O		Array	요청한 회원 변경 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234",
    "user_info" :
    [
        {
            "level" : 2 ,
            "point" : 560
        }
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /admin/<userid>/status



Parameter

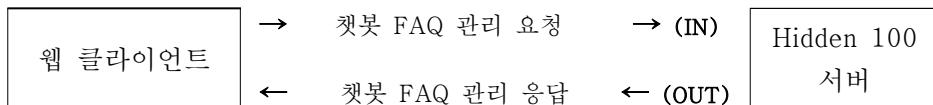
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
status	O		String	요청한 회원 변경상태
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user5678",
    "status" : "정지"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /admin/chatbot/faq



Parameter

속성	전송방향		Type	Description
	IN	OUT		
faq_info	O		Array	요청한 FAQ 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

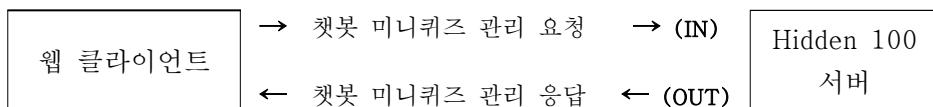
```
전송방향 : IN
{
    "faq_info" :
    [
        [
            "question1" : "내 정보 보여줘" ,
            "answer1" : "고객님의 소중한 정보입니다 ..." ,
        ] ,
        [
            "question2" : "아이폰 조회해줘" ,
            "answer2" : "현재 등록돼 있는 상품은 총 89개입니다 ..." ,
        ] ,
    ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
```

팀번호: 01

챗봇을 이용한 중고거래 서비스
Hidden 100version
v1.0

【P-실무프로젝트】

PUT /admin/chatbot/mini_quiz



Parameter

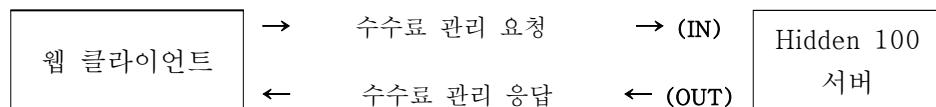
속성	전송 방향		Type	Description
	IN	OUT		
mini_quiz_info	O		Array	요청한 미니퀴즈 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "mini_quiz_info" :
        [
            [
                "question1" : "사용자들과 자유롭게 소통 가능한 게시판은?" ,
                "answer1_1" : "문의게시판" ,
                "answer1_2" : "자유게시판"
            ] ,
            ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /admin/profit/commission



Parameter

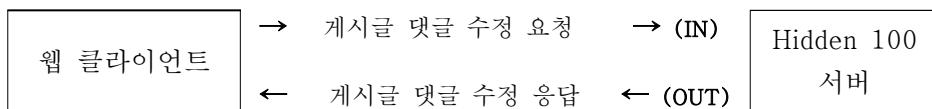
속성	전송 방향		Type	Description
	IN	OUT		
commission_info	O		Array	요청한 수수료 정보
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "comission_info" :
        [
            "charge_coin_1" : "10%" ,
            "charge_coin_2" : "5%" ,
            "register_goods" : "8%"
        ]
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

PUT /admin/board/<type>/<num>/put/comment



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
comment	O		String	요청한 게시글 댓글 수정 내용
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

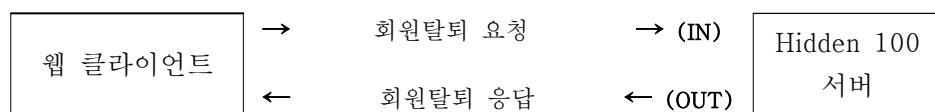
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "report" ,
    "num" : 22 ,
    "comment" : "고객님, 신고접수가 완료되었습니다 ..."
}

전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

DELETE /user/<userid>/delete



Parameter

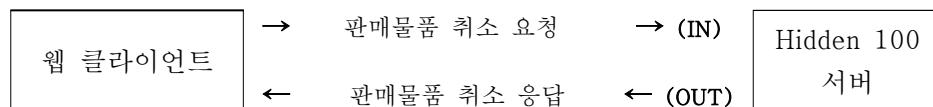
속성	전송 방향		Type	Description
	IN	OUT		
userid	O		String	요청한 회원 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "userid" : "user1234"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

DELETE /user/goods/<goo_id>/delete



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
goo_id	O		int	요청한 판매물품 고유번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

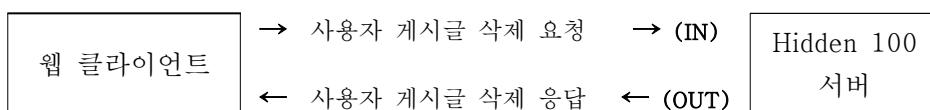
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "goo_id" : 82754801
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}

```

DELETE /user/board/<type>/<num>/delete



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

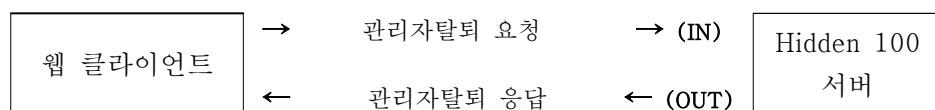
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "bulletin" ,
    "num" : 58
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}

```

DELETE /admin/<adminid>/delete



Parameter

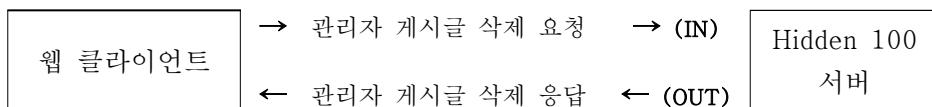
속성	전송 방향		Type	Description
	IN	OUT		
adminid	O		String	요청한 관리자 아이디
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "adminid" : "admin2345"
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

DELETE /admin/board/<type>/<num>/delete



Parameter

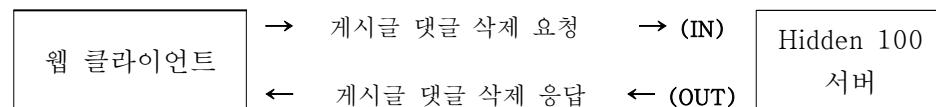
속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "question" ,
    "num" : 78
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

DELETE /admin/board/<type>/<num>/<com_num>/delete



Parameter

속성	전송 방향		Type	Description
	IN	OUT		
type	O		String	요청한 게시판 종류
num	O		int	요청한 게시글 번호
com_num	O		int	요청한 게시글 댓글 번호
content_type		O	String	application/json
result_code		O	int	처리결과 코드(200: 성공, 600: 오류코드)
result_req		O	String	처리결과 오류메세지

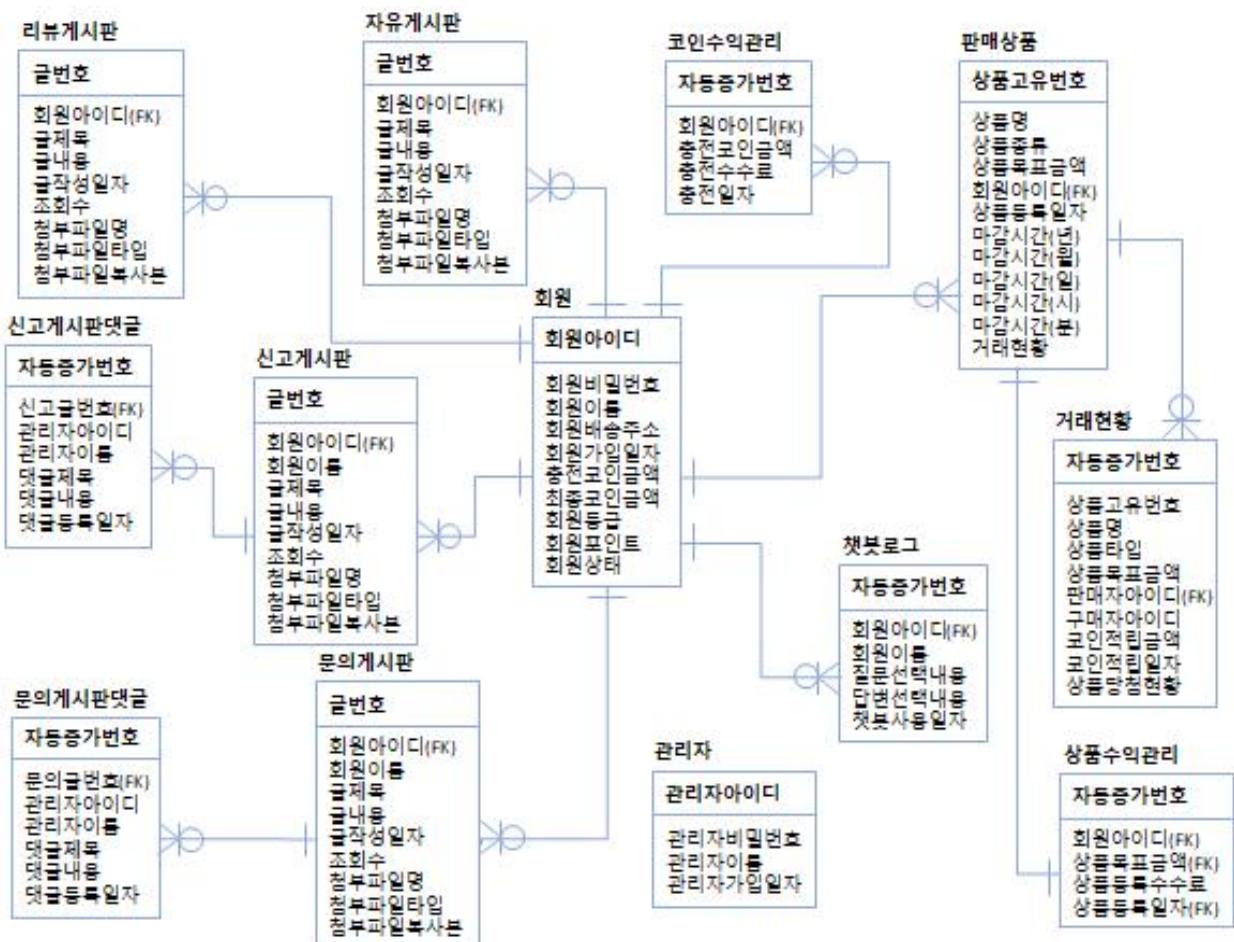
실제 전송내용 (JSON 의 경우라면)

```

전송방향 : IN
{
    "type" : "report" ,
    "num" : 67 ,
    "com_num" : 3
}
전송방향 : OUT
{
    "content_type" : "json" ,
    "result_code" : 200 ,
    "result_req" : ""
}
  
```

6. 데이터베이스 설계

6.1. ERD



6.2. 논리적 DB설계 (테이블명세서)

테이블명(회원) :			T1_MEMBER			
NO	Attribute	Data Type	NN	PK	FK	Description
1	MEM_ID	VARCHAR(20)	O	O		회원아이디
2	MEM_PASS	VARCHAR(30)	O			회원비밀번호
3	MEM_NAME	VARCHAR(20)	O			회원이름
4	MEM_ADDR	VARCHAR(100)	O			회원배송주소
5	REGIST_DAY	TIMESTAMP	O			회원가입일자
6	CHARG_COIN	INTEGER	O		O	충전코인금액
7	COIN	INTEGER	O			최종코인금액
8	LEVEL	VARCHAR(10)	O			회원등급
9	POINT	INTEGER	O			회원포인트
10	STATUS	VARCHAR(10)	O			회원상태

테이블명(관리자) :			T1_ADMIN			
NO	Attribute	Data Type	NN	PK	FK	Description
1	ADMIN_ID	VARCHAR(15)	O	O		관리자아이디
2	ADMIN_PASS	VARCHAR(20)	O			관리자비밀번호
3	ADMIN_NAME	VARCHAR(15)	O			관리자이름
4	REGIST_DAY	TIMESTAMP	O			관리자가입일자

테이블명(판매상품) :			T1_GOODS			
NO	Attribute	Data Type	NN	PK	FK	Description
1	GOO_ID	INTEGER	O	O		상품고유번호
2	GOO_NAME	VARCHAR(15)	O			상품명
3	GOO_TYPE	VARCHAR(50)	O			상품타입
4	GOAL_PRICE	VARCHAR(50)	O			상품목표금액
5	MEM_ID	VARCHAR(20)	O		O	회원아이디
6	REGIST_DAY	TIMESTAMP	O			상품등록일자
7	TIME_YEAR	INTEGER	O			마감시간(년)
8	TIME_MONTH	INTEGER	O			마감시간(월)
9	TIME_DAY	INTEGER	O			마감시간(일)
10	TIME_HOUR	INTEGER	O			마감시간(시)
11	TIME_MINUTE	INTEGER	O			마감시간(분)
12	STATUS	VARCHAR(20)	O			거래현황

테이블명(거래현황) :			T1 DEAL			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	GOO_ID	INTEGER	O		O	상품고유번호
3	GOO_NAME	VARCHAR(50)	O		O	상품명
4	GOO_TYPE	VARCHAR(50)	O		O	상품타입
5	GOAL_PRICE	INTEGER	O		O	상품목표금액
6	SELLER_ID	VARCHAR(20)	O		O	판매자아이디
7	BUYER_ID	VARCHAR(20)	O			구매자아이디
8	INVEST_COIN	INTEGER	O			코인적립금액
9	INVEST_DAY	TIMESTAMP	O			코인적립일자
10	STATUS	VARCHAR(20)				상품당첨현황

테이블명(챗봇로그) :			T1_CHATBOT_LOG			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	MEM_ID	VARCHAR(20)	O		O	회원아이디
2	MEM_NAME	VARCHAR(20)	O			회원이름
3	QUESTION	TEXT	O			질문선택내용
4	ANSWER	TEXT	O			답변선택내용
5	USE_DAY	TIMESTAMP	O			챗봇사용일자

테이블명(코인수익관리) :			T1 PROFIT_COIN			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	MEM_ID	VARCHAR(20)	O		O	회원아이디
3	CHARG_COIN	INTEGER	O			충전코인금액
4	COMMISSION	INTEGER	O			충전수수료
5	CHARGE_DAY	TIMESTAMP	O			충전일자

테이블명(상품수익관리) :			T1 PROFIT_GOODS			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	MEM_ID	VARCHAR(20)	O		O	회원아이디
3	GOAL_PRICE	INTEGER	O		O	상품목표금액
4	COMMISSION	INTEGER	O			상품등록수수료
5	REGIST_DAY	TIMESTAMP	O		O	상품등록일자

테이블명(자유게시판) :			T1_BULLETIN			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		글번호
2	MEM_ID	VARCHAR(15)	O		O	회원아이디
3	SUBJECT	VARCHAR(200)	O			글제목
4	CONTENT	TEXT	O			글내용
5	REGIST_DAY	TIMESTAMP	O			글작성일자
6	HIT	INTEGER	O			조회수
7	FILE_NAME	VARCHAR(40)				첨부파일명
8	FILE_TYPE	VARCHAR(40)				첨부파일타입
9	FILE_COPIED	VARCHAR(40)				첨부파일복사본

테이블명(문의게시판) :			T1_QUESTION			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		글번호
2	MEM_ID	VARCHAR(15)	O		O	회원아이디
3	MEM_NAME	VARCHAR(20)	O			회원이름
4	SUBJECT	VARCHAR(200)	O			글제목
5	CONTENT	TEXT	O			글내용
6	REGIST_DAY	TIMESTAMP	O			글작성일자
7	HIT	INTEGER	O			조회수
8	FILE_NAME	VARCHAR(40)				첨부파일명
9	FILE_TYPE	VARCHAR(40)				첨부파일타입
10	FILE_COPIED	VARCHAR(40)				첨부파일복사본

테이블명(문의게시판댓글) :			T1_Q_COMMENT			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	QUEST_NUM	INTEGER	O		O	문의글번호
3	ADMIN_ID	VARCHAR(20)	O			관리자아이디
4	ADMIN_NAME	VARCHAR(20)	O			관리자이름
5	SUBJECT	VARCHAR(200)	O			댓글제목
6	CONTENT	TEXT	O			댓글내용
7	REGIST_DAY	TIMESTAMP	O			댓글등록일자

테이블명(리뷰게시판) :			T1 REVIEW			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		글번호
2	MEM_ID	VARCHAR(15)	O		O	회원아이디
3	SUBJECT	VARCHAR(200)	O			글제목
4	CONTENT	TEXT	O			글내용
5	REGIST_DAY	TIMESTAMP	O			글작성일자
6	HIT	INTEGER	O			조회수
7	FILE_NAME	VARCHAR(40)				첨부파일명
8	FILE_TYPE	VARCHAR(40)				첨부파일타입
9	FILE_COPIED	VARCHAR(40)				첨부파일복사본

테이블명(신고게시판) :			T1 REPORT			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		글번호
2	MEM_ID	VARCHAR(15)	O		O	회원아이디
3	MEM_NAME	VARCHAR(20)	O			회원이름
4	SUBJECT	VARCHAR(200)	O			글제목
5	CONTENT	TEXT	O			글내용
6	REGIST_DAY	TIMESTAMP	O			글작성일자
7	HIT	INTEGER	O			조회수
8	FILE_NAME	VARCHAR(40)				첨부파일명
9	FILE_TYPE	VARCHAR(40)				첨부파일타입
10	FILE_COPIED	VARCHAR(40)				첨부파일복사본

테이블명(신고게시판댓글) :			T1 RP COMMENT			
NO	Attribute	Data Type	NN	PK	FK	Description
1	NUM	INTEGER	O	O		자동증가번호
2	REPORT_NUM	INTEGER	O		O	신고글번호
3	ADMIN_ID	VARCHAR(20)	O			관리자아이디
4	ADMIN_NAME	VARCHAR(20)	O			관리자이름
5	SUBJECT	VARCHAR(200)	O			댓글제목
6	CONTENT	TEXT	O			댓글내용
7	REGIST_DAY	TIMESTAMP	O			댓글등록일자

6.3. 물리적 DB설계 (SQL스크립트)

테이블명	T1_MEMBER
<pre>CREATE TABLE t1_member (mem_id varchar(20) NOT NULL UNIQUE, mem_pass varchar(30) NOT NULL, mem_name varchar(20) NOT NULL, mem_addr varchar(100) NOT NULL, regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, charg_coin integer NOT NULL, coin integer NOT NULL, level varchar(10) NOT NULL, point integer NOT NULL, status varchar(20) NOT NULL, PRIMARY KEY (mem_id), FOREIGN KEY (charge_coin) REFERENCES t1_profit_coin (charge_coin));</pre>	

테이블명	T1_ADMIN
<pre>CREATE TABLE t1_admin (admin_id varchar(15) NOT NULL UNIQUE, admin_pass varchar(20) NOT NULL, admin_name varchar(15) NOT NULL, regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (admin_id));</pre>	

테이블명	T1_GOODS
<pre>CREATE TABLE t1_goods (goo_id integer NOT NULL UNIQUE, goo_name varchar(50) NOT NULL, goo_type varchar(50) NOT NULL, goal_price integer NOT NULL, mem_id varchar(20) NOT NULL, regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, time_year integer NOT NULL, time_month integer NOT NULL, time_day integer NOT NULL, time_hour integer NOT NULL, time_minute integer NOT NULL, status varchar(20) NOT NULL, PRIMARY KEY (goo_id), FOREIGN KEY (mem_id) REFERENCES t1_member (mem_id));</pre>	

테이블명	T1 DEAL
<pre>CREATE TABLE t1_deal (num integer NOT NULL AUTO_INCREMENT, goo_id integer NOT NULL, goo_name varchar(50) NOT NULL, goo_type varchar(50) NOT NULL, goal_price integer NOT NULL, seller_id varchar(20) NOT NULL, buyer_id varchar(20) NOT NULL, invest_coin integer NOT NULL, invest_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, status varchar(20), PRIMARY KEY (num), FOREIGN KEY (goo_id, goo_name, goo_type, goal_price, seller_id) REFERENCES t1_goods (goo_id, goo_name, goo_type, goal_price, mem_id));</pre>	

테이블명 | T1_CHATBOT_LOG

```
CREATE TABLE t1_chatbot_log
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(20) NOT NULL,
    mem_name varchar(20) NOT NULL,
    question text NOT NULL,
    answer text NOT NULL,
    use_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id)
    REFERENCES t1_member (mem_id)
);
```

테이블명 | T1_PROFIT_COIN

```
CREATE TABLE t1_profit_coin
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(20) NOT NULL,
    charg_coin integer NOT NULL,
    commission integer NOT NULL,
    charge_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id)
    REFERENCES t1_member (mem_id)
);
```

테이블명 | T1_PROFIT_GOODS

```
CREATE TABLE t1_profit_goods
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(20) NOT NULL,
    goal_price integer NOT NULL,
    commission integer NOT NULL,
    regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id, goal_price, regist_day)
    REFERENCES t1_goods (mem_id, goal_price, regist_day)
);
```

테이블명 | T1_BULLETIN

```
CREATE TABLE t1_bulletin
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(15) NOT NULL,
    subject varchar(200) NOT NULL,
    content text NOT NULL,
    regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    hit integer NOT NULL,
    file_name varchar(40),
    file_type varchar(40),
    file_copied varchar(40),
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id)
    REFERENCES t1_member (mem_id)
);
```

테이블명 | T1_QUESTION

```
CREATE TABLE t1_question
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(15) NOT NULL,
    mem_name varchar(20) NOT NULL,
    subject varchar(200) NOT NULL,
    content text NOT NULL,
    regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    hit integer NOT NULL,
    file_name varchar(40),
    file_type varchar(40),
    file_copied varchar(40),
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id)
    REFERENCES t1_member (mem_id)
);
```

테이블명 | T1_Q_COMMENT

```
CREATE TABLE t1_q_comment
(
    num integer NOT NULL AUTO_INCREMENT,
    quest_num integer NOT NULL,
    admin_id varchar(20) NOT NULL,
    admin_name varchar(20) NOT NULL,
    subject varchar(200) NOT NULL,
    content text NOT NULL,
    regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (num),
    FOREIGN KEY (quest_num)
    REFERENCES t1_question (num)
);
```

테이블명 | T1 REVIEW

```
CREATE TABLE t1_review
(
    num integer NOT NULL AUTO_INCREMENT,
    mem_id varchar(15) NOT NULL,
    subject varchar(200) NOT NULL,
    content text NOT NULL,
    regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    hit integer NOT NULL,
    file_name varchar(40),
    file_type varchar(40),
    file_copied varchar(40),
    PRIMARY KEY (num),
    FOREIGN KEY (mem_id)
    REFERENCES t1_member (mem_id)
);
```

테이블명	T1_REPORT
<pre>CREATE TABLE t1_report (num integer NOT NULL AUTO_INCREMENT, mem_id varchar(15) NOT NULL, mem_name varchar(20) NOT NULL, subject varchar(200) NOT NULL, content text NOT NULL, regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, hit integer NOT NULL, file_name varchar(40), file_type varchar(40), file_copied varchar(40), PRIMARY KEY (num), FOREIGN KEY (mem_id) REFERENCES t1_member (mem_id));</pre>	

테이블명	T1_RP_COMMENT
<pre>CREATE TABLE t1_rp_comment (num integer NOT NULL AUTO_INCREMENT, report_num integer NOT NULL, admin_id varchar(20) NOT NULL, admin_name varchar(20) NOT NULL, subject varchar(200) NOT NULL, content text NOT NULL, regist_day timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (num), FOREIGN KEY (report_num) REFERENCES t1_report (num));</pre>	

7. 환경구성

7.1. 개발환경 및 운영환경

■ 서버

구분	개발환경	규격	비고
개발툴	편집툴	ATOM	
	개발IDE	VSCODE, BRAKET	
	프레임워크	Node.js	
	사용할 미들웨어(모듈)	Express Router Jade Mysql Body-parser Http .	
서버	TCP Port주소		
	IP주소		
	OS	CentosOS	
	DB	MariaDB	
서버 하드웨어	CPU	Intel Core2Duo	
	RAM	4G RAM	
	DISK	500G DISK (SATAII)	
	Network	100Mbps	

■ 클라이언트

구분	개발환경	규격	비고
클라이언트	OS	Window/Linux/macOs	

7.2. 소스디렉터리 구조

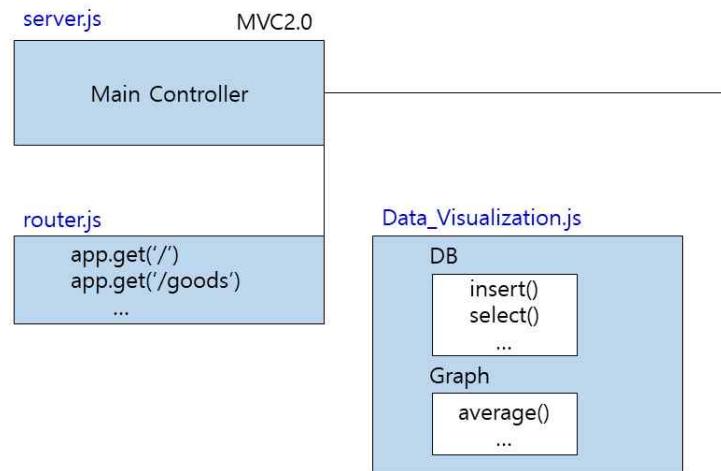


그림 7.1 디렉토리 구조

그림 7.1과 같이 MVC 구조로 개발 디렉토리 구조를 정의하며, 그림 7.2와 같이 세부 디렉터리와 파일명으로 모듈화를 정의한다.

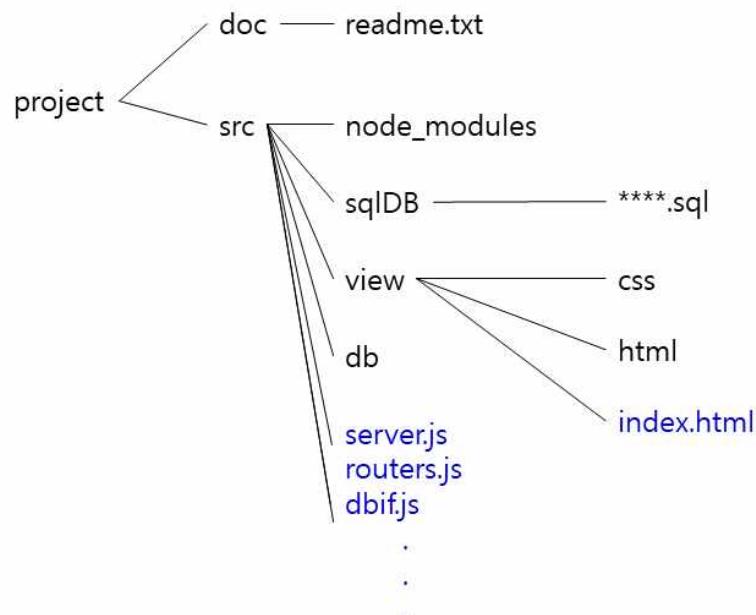


그림 7.2 디렉터리 및 파일환경