# SW 설계서

## 1. 웹 UI 디자인 설계

### 1.1 개요

웹 UI 디자인 설계는 사용자 경험을 고려하여 직관적이고 사용하기 쉬운 인터페이스를 제공하는 것을 목표로 한다. 본 문서에서는 프로젝트의 웹 UI 구조와 주요 디자인 요소를 설명한다.

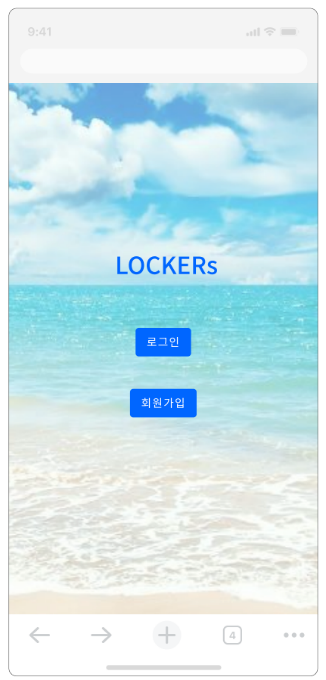
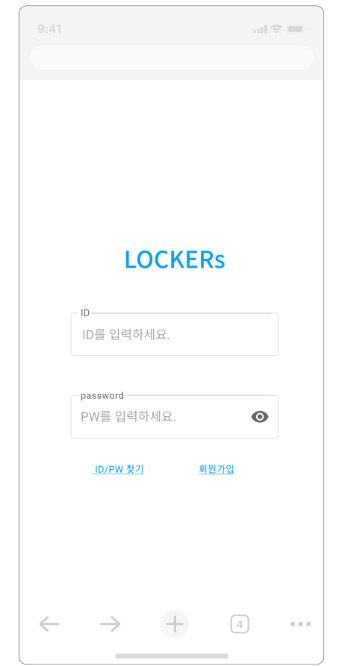
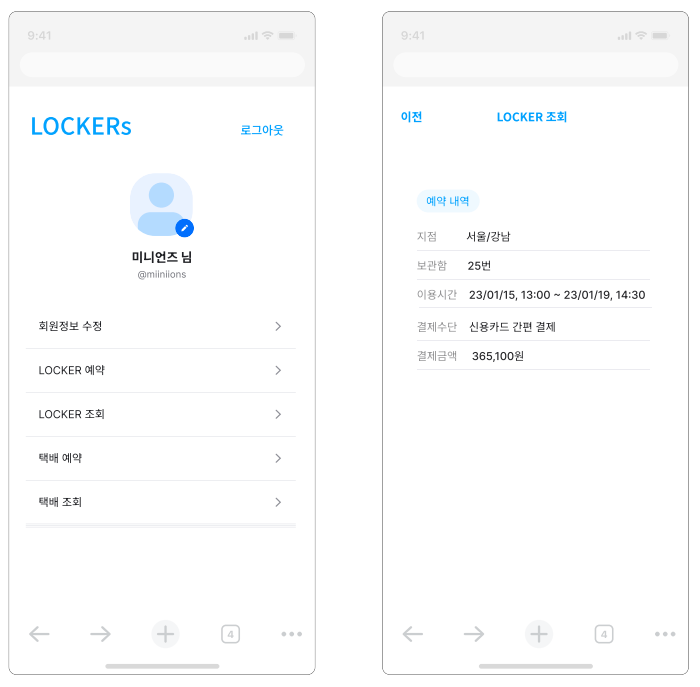
### 1.2 요구사항 분석

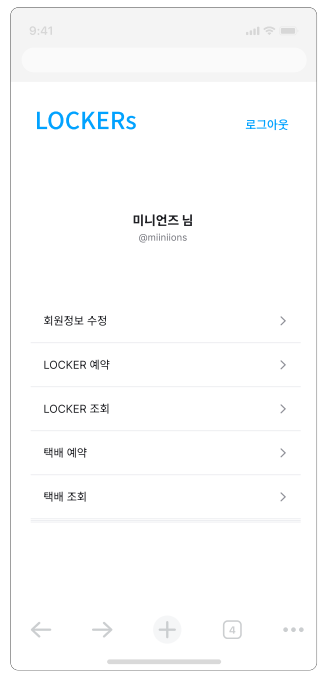
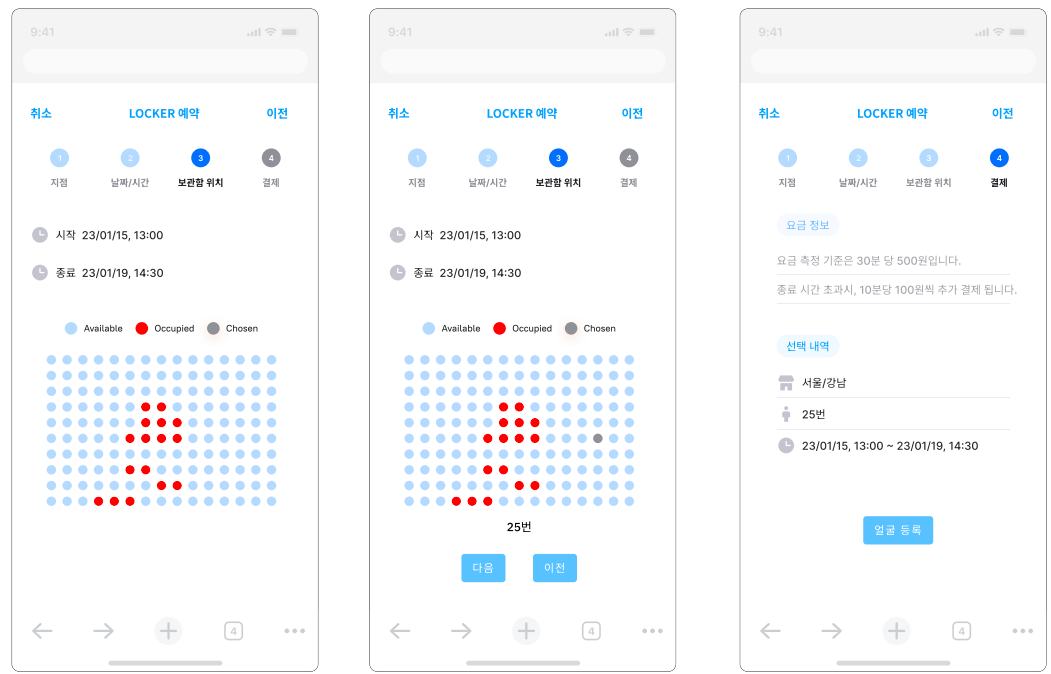
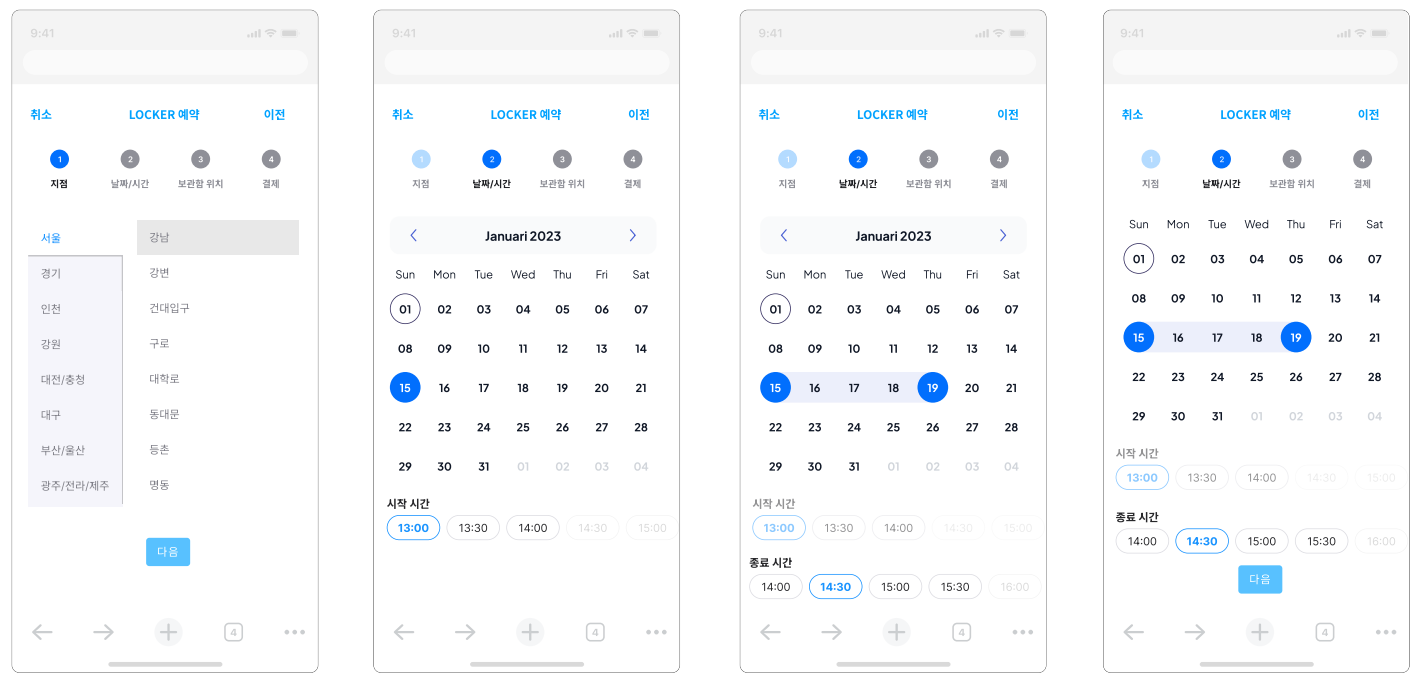
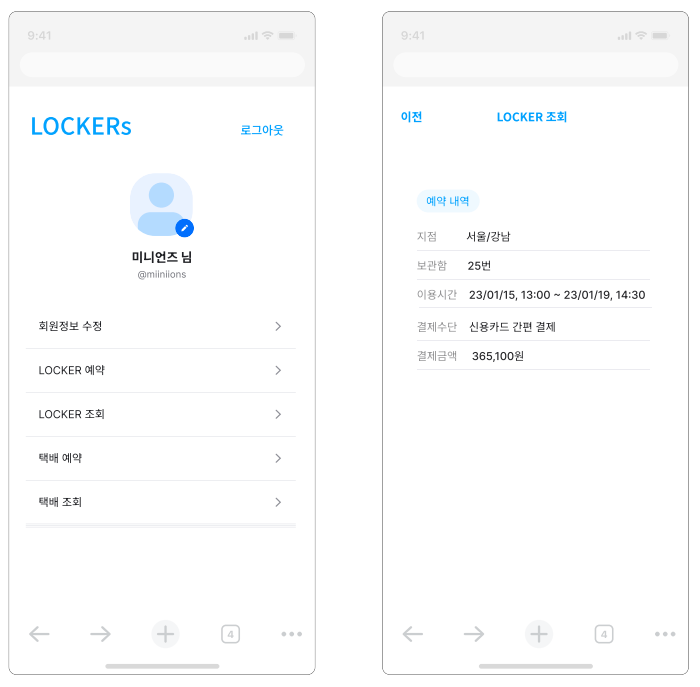
* **사용자 역할**: 시스템을 사용할 다양한 사용자 역할 정의
  + **일반 사용자**: 보관함 예약, 택배 예약, 예약 내역 조회, 결제, 회원 정보 수정 등
  + **관리자**: 사용자 관리, 예약 내역 관리, 결제 내역 관리 등 (추후 필요 시 추가)
* **주요 기능**: 각 사용자 역할에 따라 제공되어야 할 주요 기능 목록
  + **회원가입 및 로그인**: 사용자 계정 생성 및 로그인 기능
  + **회원정보 수정**: 사용자 정보 수정 및 관리
  + **LOCKER 예약**: 지점 및 구 선택, 날짜/시간 선택, 보관함 선택, 결제
  + **택배 예약**: 출발 지점 및 도착 지점 선택, 날짜/시간 선택, 보관함 선택, 결제
  + **예약 조회**: 사용자 예약 내역 및 결제 정보 조회

### 1.3 UI 흐름도

사용자의 전체적인 흐름을 나타내는 UI 흐름도

1. **시작화면**
   * 주요 링크: 로그인, 회원가입
2. **회원가입 페이지**
   * 사용자는 자신의 정보를 입력하고 회원가입을 완료
   * 가입 완료 후 시작화면으로 돌아옴
3. **로그인 페이지**
   * 사용자는 자신의 아이디와 비밀번호를 입력하여 로그인
4. **홈화면**
   * 사용자는 여러 가지 기능을 선택 가능: 회원정보수정, LOCKER 예약, LOCKER 조회, 택배 예약, 택배 조회, 로그아웃



1. **LOCKER 예약 흐름**
   * **지점 선택 페이지**: 사용자는 원하는 지점과 구를 선택
   * **날짜/시간 선택 페이지**: 사용자는 보관함 사용을 원하는 시작 및 종료 날짜/시간을 선택
   * **보관함 선택 페이지**: 사용자는 선택 가능한 보관함 중 하나를 선택
   * **결제 페이지**: 사용자는 결제수단을 선택하고 결제를 완료, 필요 시 얼굴 등록 및 카드 등록을 수행
2. **택배 예약 흐름**
   * **출발/도착 지점 선택 페이지**: 사용자는 출발 지점과 도착 지점을 선택
   * **날짜/시간 선택 페이지**: 출발 지점의 시작 날짜/시간과 도착 지점의 종료 날짜/시간을 선택
   * **보관함 선택 페이지**: 출발 및 도착 지점의 보관함을 선택
   * **결제 페이지**: 배송비를 추가하여 결제를 완료
3. **예약 조회 흐름**
   * 사용자는 LOCKER 조회 또는 택배 조회를 선택하여, 자신의 예약 내역과 결제 정보를 확인

## 2. DB: ERD

### 2.1 개요

본 섹션에서는 프로젝트의 데이터베이스 구조를 설계하고, 각 테이블 간의 관계를 정의한 ERD를 설명한다.

### 2.2 요구사항 분석

* **데이터 엔터티**:
  + 사용자(User): 시스템을 이용하는 모든 사용자 정보를 저장
  + 보관함 예약(ReservationLocker): 사용자가 예약한 보관함 내역을 저장
  + 택배 예약(ReservationDelivery): 사용자가 예약한 택배 내역을 저장
  + 결제(Payments): 사용자의 결제 정보를 저장
  + 위치(Locations): 보관함 및 택배의 위치 정보를 저장
  + 보관함(Lockers): 각 위치에 배치된 보관함 정보를 저장
  + 얼굴(Faces): 사용자 얼굴 데이터를 저장
  + 예약(Reservations): 사용자가 예약한 정보들을 저장
  + 카드(Card) : 사용자가 등록한 카드 정보를 저장

사용자 테이블 (users)

* **user\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **username**: VARCHAR(50) NOT NULL UNIQUE
* **name**: VARCHAR(50) NOT NULL
* **email**: VARCHAR(255) NOT NULL UNIQUE
* **password**: VARCHAR(255) NOT NULL
* **phone**: VARCHAR(20)
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP
* **is\_active**: BOOLEAN DEFAULT TRUE
* **is\_staff**: BOOLEAN DEFAULT FALSE
* **is\_superuser**: BOOLEAN DEFAULT FALSE
* **last\_login**: TIMESTAMP
* **date\_joined**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

위치 테이블 (locations)

* **location\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **city**: VARCHAR(100) NOT NULL
* **district**: VARCHAR(100) NOT NULL
* **address**: VARCHAR(255)
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

보관함 테이블 (lockers)

* **locker\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **locker\_number**: INT NOT NULL
* **status**: ENUM('available', 'occupied') DEFAULT 'available'
* **location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

예약 테이블 (reservations)

* **reservation\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)
* **start\_locker\_id**: INT FOREIGN KEY REFERENCES lockers(locker\_id)
* **end\_locker\_id**: INT FOREIGN KEY REFERENCES lockers(locker\_id)
* **start\_location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **end\_location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **start\_datetime**: DATETIME
* **end\_datetime**: DATETIME
* **status**: ENUM('reserved', 'completed') DEFAULT 'reserved'
* **reservation\_type**: ENUM('locker', 'delivery') DEFAULT 'locker'
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

보관함 예약 테이블 (reservation\_locker)

* **reservation\_id**: INT PRIMARY KEY FOREIGN KEY REFERENCES reservations(reservation\_id)
* **locker\_id**: INT FOREIGN KEY REFERENCES lockers(locker\_id)
* **location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)

택배 예약 테이블 (reservation\_delivery)

* **reservation\_id**: INT PRIMARY KEY FOREIGN KEY REFERENCES reservations(reservation\_id)
* **delivery\_fee**: DECIMAL(10, 2)
* **start\_location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **end\_location\_id**: INT FOREIGN KEY REFERENCES locations(location\_id)
* **start\_locker\_id**: INT FOREIGN KEY REFERENCES lockers(locker\_id)
* **end\_locker\_id**: INT FOREIGN KEY REFERENCES lockers(locker\_id)
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)

결제 테이블 (payments)

* **payment\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **amount**: DECIMAL(10, 2) NOT NULL
* **card**: VARCHAR(255) NULL
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)
* **reservation\_id**: INT FOREIGN KEY REFERENCES reservations(reservation\_id)

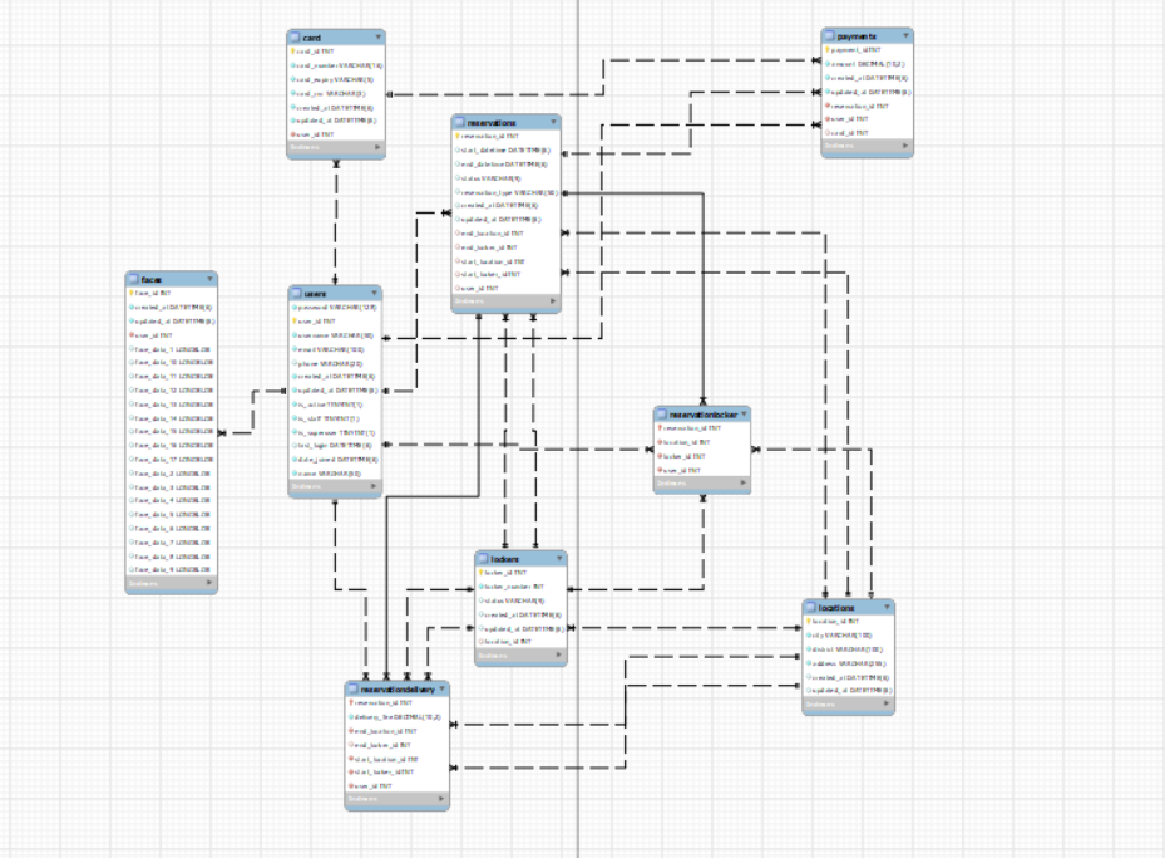
카드 테이블 (card)

* **card\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **card\_number**: VARCHAR(50) NOT NULL UNIQUE
* **card\_expiry**: VARCHAR(50) NOT NULL UNIQUE
* **card\_cvc**: VARCHAR(50) NOT NULL UNIQUE
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)

얼굴 테이블 (faces)

* **face\_id**: INT AUTO\_INCREMENT PRIMARY KEY
* **user\_id**: INT FOREIGN KEY REFERENCES users(user\_id)
* **face\_data[1:17]**: BLOB
* **created\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
* **updated\_at**: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

### 2.4 ERD (Entity Relationship Diagram)



프로젝트의 데이터베이스 구조를 시각적으로 나타내는 ERD이다. 테이블 간의 관계는 다음과 같다.

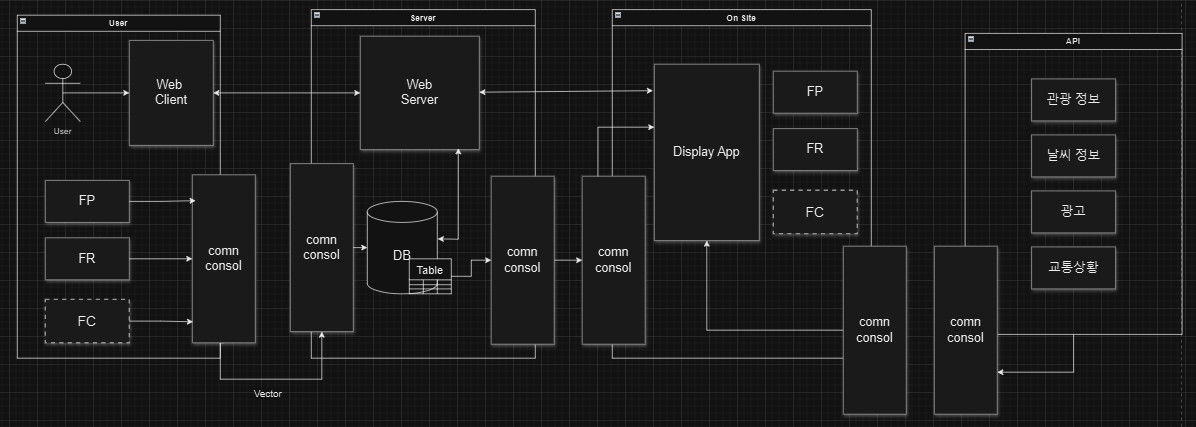
* **users** : **reservations** = 1 : N
* **locations** : **lockers** = 1 : N
* **reservations** : **reservation\_locker** = 1 : 1
* **reservations** : **reservation\_delivery** = 1 : 1
* **users** : **payments** = 1 : N
* **reservations** : **payments** = 1 : N
* **users** : **faces** = 1 : 1
* **payments : card =** 1 : 1

### 2.5 데이터베이스 인덱스 및 제약 조건

각 테이블에 필요한 인덱스 및 제약 조건을 정의

* **users 테이블**:
  + email 필드에 대한 유니크 인덱스
* **lockers 테이블**:
  + location\_id 필드에 대한 외래키 제약 조건
* **reservations 테이블**:s
  + user\_id 필드에 대한 외래키 제약 조건
* **reservation\_locker 테이블**:
  + reservation\_id, locker\_id, location\_id 필드에 대한 외래키 제약 조건
* **reservation\_delivery 테이블**:
  + reservation\_id, start\_location\_id, end\_location\_id, start\_locker\_id, end\_locker\_id 필드에 대한 외래키 제약 조건
* **payments 테이블**:
  + user\_id, reservation\_id 필드에 대한 외래키 제약 조건
* **card 테이블** :
  + user\_id, payment\_id 필드에 대한 외래키 제약 조건
* **faces 테이블**:
  + user\_id 필드에 대한 외래키 제약 조건

## System Diagram



## System Flow Chart

