



# 소프트웨어 요구사항 정의서 (SRS)

프로젝트명: 디지털 담벼락 (Digital Wall)

버전: v1.4 (Final)

작성일: 2025-11-22

상태: 승인됨 (Approved)

## 1. 프로젝트 개요 (Introduction)

### 1.1. 목적

로그인 과정 없이 누구나 즉시 참여할 수 있는 익명 기반의 실시간 협동 웹 플랫폼을 구축한다. 사용자는 무한히 확장되는 캔버스에 자신의 흔적(포스트잇)을 남기고, 타인의 게시물을 자유롭게 이동시키며 상호작용한다.

### 1.2. 핵심 가치

- Extreme Anonymity:** 회원가입 없음, 최소한의 정보 수집.
- Infinite Freedom:** 공간의 제약이 없는 캔버스, 타인의 글도 움직이는 자유도.
- Hidden Control:** 관리자는 보이지 않는 곳에서 악성 행위를 감시하고 제어한다.

## 2. 사용자 역할 (User Roles)

역할	접근 경로	권한 및 설명
일반 사용자 (User)	메인 도메인 접속 ( / )	<ul style="list-style-type: none"><li>포스트잇 작성, 조회, 이동(전체), 본인 글 삭제</li><li>관리자 문의 발송</li><li>별도의 로그인 절차 없음</li></ul>
관리자 (Admin)	시크릿 URL 접속 ( /admin-xyz... )	<ul style="list-style-type: none"><li>전체 게시물 조회 및 강제 삭제</li><li>작성자 IP/기기 정보 확인</li><li>IP 차단(Ban)</li><li>문의 내역 확인</li></ul>

## 3. 기능적 요구사항 (Functional Requirements)

### 3.1. 캔버스 및 UI (Canvas Interface)

- **FR-C-01 (무한 좌표계):** 캔버스는 상하좌우 경계가 없는 무한 좌표계(Int range)를 가진다.
- **FR-C-02 (패닝 & 줌):** \* **PC:** 마우스 드래그로 화면 이동, 휠로 확대/축소.
  - **Mobile:** 두 손가락 드래그로 화면 이동/줌 (Pinch-to-Zoom).
- **FR-C-03 (원점 복귀):** 길을 잃었을 때  $(0, 0)$  좌표로 즉시 이동하는 버튼을 제공한다.
- **FR-C-04 (반응형 UI):** 모바일 기기 접속 시, 글쓰기 버튼과 메뉴가 모바일 화면 비율에 맞춰 재배치된다.

## 3.2. 게시물 상호작용 (Interaction)

- **FR-I-01 (포스트잇 작성):** \* 내용, 배경색, 폰트 스타일을 선택하여 작성한다.
  - 작성된 포스트잇은 현재 사용자가 보고 있는 화면의 중앙에 생성된다.
- **FR-I-02 (글로벌 드래그 앤 드롭):** \* 사용자는 본인 및 타인의 게시물을 모두 이동시킬 수 있다.
  - 이동 결과는 WebSocket을 통해 모든 접속자에게 실시간 동기화된다.
- **FR-I-03 (시간 표시):** \* 일반 사용자 화면: 방금 전, 10분 전 등의 상대 시간으로 표시.
  - 관리자 화면: 2024-11-22 14:30:00 형식의 절대 시간으로 표시.
- **FR-I-04 (본인 글 삭제):** \* 작성 시 발급된 Secret Key가 로컬 스토리지에 존재하는 경우에만 삭제 버튼 [x]이 활성화된다.
  - 삭제 시 서버에서 키 검증 후 데이터를 영구 제거한다.

## 3.3. 관리자 감시 및 보안 (Admin & Security)

- **FR-A-01 (정보 수집):** \* 게시글 작성 시 User-Agent를 파싱하여 OS, 브라우저, 기기 종류 (Mobile/PC)를 저장한다.
  - 작성자의 IP 주소를 저장한다.
  - 이 정보들은 관리자 API 호출 시에만 응답 데이터에 포함된다.
- **FR-A-02 (차단 및 삭제):** \* 관리자는 특정 IP를 차단(Ban)할 수 있으며, 차단된 IP는 글 작성이 불가능하다.
  - 관리자는 모든 게시물을 강제로 삭제할 권한을 가진다.
- **FR-A-03 (1:1 문의함):** \* 사용자는 별도의 모달 창을 통해 관리자에게 비공개 메시지를 전송할 수 있다.
  - 관리자는 대시보드에서 문의 내역(내용 + 작성자 IP/기기 정보)을 확인한다.

# 4. 데이터베이스 스키마 설계 (Database Schema)

## 4.1. Posts Collection (게시글)

```
{  
  "_id": "UUID",  
  "content": "String (Max 200 chars)",  
  "style": {  
    "color": "String (Hex)",  
    "font": "String (Font-Family)",  
    "rotation": "Number (Float, -5 ~ +5)"  
}
```

```
},
"position": {
  "x": "Number (Integer)",
  "y": "Number (Integer)",
  "zIndex": "Number (Integer)"
},
"auth": {
  "deleteSecret": "String (Hashed Secret Key for User Deletion)"
},
"meta": {
  "createdAt": "Date (ISO 8601)",
  "ipAddress": "String (Hidden from Public)",
  "deviceInfo": {
    "os": "String",
    "browser": "String",
    "type": "String (mobile | desktop)"
  },
  "isBanned": "Boolean"
}
}
```

## 4.2. Inquiries Collection (문의)

```
{
  "_id": "UUID",
  "message": "String",
  "contactInfo": "String (Optional Email/Phone)",
  "isRead": "Boolean (Default: false)",
  "meta": {
    "createdAt": "Date",
    "ipAddress": "String",
    "deviceInfo": "Object"
  }
}
```

## 5. 비기능적 요구사항 (Non-Functional Requirements)

- 성능 (Performance):** 캔버스 내 포스트잇이 1,000개 이상일 때도 렌더링 렉이 없어야 한다. (Viewport Culling 기술 권장)
- 보안 (Security):** 관리자 페이지 URL은 추측 불가능해야 하며, 일반 사용자에게는 IP 정보가 절대 노출되어어서는 안 된다.
- 확장성 (Scalability):** 추후 트래픽 증가 시 Firebase 또는 클라우드 DB로 이관이 용이한 구조여야 한다.

## 6. 기술 스택 제안 (Tech Stack)

- Frontend:** React.js (Vite), Canvas API or DOM Rendering

- **Backend:** Node.js (Express)
- **Communication:** Socket.io (실시간 양방향 통신)
- **Database:** SQLite (파일 기반, 초기 개발용) -> 추후 MongoDB 등 교체 가능