

기능 요구사항 명세서 (FRD)

ForwardMax B2B 중고차 수출 플랫폼

문서 버전: 2.0

작성일: 2025-12-29

프로젝트: ForwardMax (carivdealer)

작성자: 기획/개발팀 류지환 작성

대상 독자: 내부 이해관계자

목차

1. 개요
 2. 기술 스택
 3. 시스템 아키텍처
 4. 화면 명세
 5. 기능 명세
 6. API 설계 개요
 7. 데이터 모델 설계
 8. 환경 설정
 9. 배포 전략
 10. 부록
 - 10.1 용어집
 - 10.2 엔터티 스키마 상세
 - 10.3 화면 전환 매트릭스
 - 10.4 구현 코드 참조
-

1. 개요

1.1 프로젝트 정보

- **프로젝트명:** ForwardMax
- **프로젝트 식별자:** carivdealer
- **플랫폼 유형:** B2B 중고차 수출 플랫폼
- **주요 사용자:** 딜러(판매자), 바이어(구매자), 평가사, 운영자

1.2 문서 목적 및 범위

본 문서는 ForwardMax 플랫폼의 기능 요구사항을 명세한 문서입니다. 특히 백엔드 개발자를 대상으로 API 설계, 데이터 모델, 비즈니스 로직을 중심으로 작성되었습니다.

문서 범위:

- 프론트엔드 화면 구조 및 화면 전환 흐름
- 백엔드 API 엔드포인트 설계
- 데이터 모델 및 엔터티 설계
- 주요 기능의 비즈니스 로직 명세
- 기술 스택 및 아키텍처 설계

제외 사항: - 바이어(구매자) 화면 (Phase 1 범위 외) - 운영자 대시보드 (Phase 1 범위 외) - 상세 구현 코드 (부록 참조)

2. 기술 스택

2.1 프론트엔드 기술 스택

| 카테고리 | 기술/라이브러리 | 버전 | 용도 |
|-------|----------|--------|-----------------|
| 프레임워크 | React | 19.2.3 | 사용자 인터페이스 프레임워크 |

| 카테고리 | 기술/라이브러리 | 버전 | 용도 |
|--------|-------------------|---------|---------------|
| 언어 | TypeScript | 5.8.2 | 타입 안정성 보장 |
| 빌드 도구 | Vite | 6.2.0 | 번들러 및 개발 서버 |
| 스타일링 | Tailwind CSS | CDN | 유ти리티 기반 스타일링 |
| 아이콘 | lucide-react | 0.562.0 | 아이콘 라이브러리 |
| AI 서비스 | Google Gemini API | 1.34.0 | OCR 및 시세 추정 |

2.2 백엔드 기술 스택

| 카테고리 | 기술/라이브러리 | 버전 | 용도 |
|------------|--------------------|------------|-----------------|
| 런타임 | Node.js | 20 | 서버 런타임 환경 |
| 프레임워크 | Firebase Functions | v2 (5.0.0) | 서버리스 함수 실행 환경 |
| 데이터베이스 | Firebase Firestore | - | NoSQL 문서 데이터베이스 |
| 스토리지 | Firebase Storage | - | 파일 스토리지 서비스 |
| 인증 | Firebase Auth | - | 사용자 인증 서비스 |
| HTTP 프레임워크 | Express | 4.18.2 | HTTP 요청 처리 미들웨어 |

2.3 인프라 및 배포 환경

| 서비스 | 용도 | 리전 |
|-----|----|----|
|-----|----|----|

| 서비스 | 용도 | 리전 |
|---------------------------|------------|-----------------|
| Firebase Hosting | 프론트엔드 호스팅 | - |
| Firebase Functions | 백엔드 API 서버 | asia-northeast3 |
| Firebase Firestore | 데이터베이스 | asia-northeast3 |
| Firebase Storage | 파일 스토리지 | asia-northeast3 |
| GCP Secret Manager | API 키 관리 | asia-northeast3 |

2.4 개발 도구

| 도구 | 용도 |
|---------------------|-------------|
| TypeScript | 타입 체크 및 컴파일 |
| Vite | 개발 서버 및 빌드 |
| Firebase CLI | 배포 및 관리 |
| Git | 버전 관리 |

3. 시스템 아키텍처

3.1 아키텍처 개요

시스템은 프론트엔드, 백엔드, 데이터베이스로 구성된 3 계층 아키텍처입니다.

프론트엔드 계층: - React 기반 단일 페이지 애플리케이션(SPA) - 컴포넌트 기반 UI 구조 - API 클라이언트를 통한 백엔드 통신

백엔드 계층: - Firebase Functions v2 기반 서버리스 아키텍처 - RESTful API 엔드포인트 제공 - Express 미들웨어를 통한 HTTP 요청 처리

데이터 계층: - Firestore: 구조화된 문서 데이터 저장 - Storage: 이미지 및 파일 저장 - Secret Manager: 민감 정보 관리

3.2 디렉토리 구조

프로젝트는 모노레포 구조로 구성되어 있으며, 프론트엔드와 백엔드가 분리되어 있습니다.

프론트엔드 구조:

- `src/components/`: 페이지 컴포넌트
- `src/services/`: API 클라이언트 및 외부 서비스 연동
- `src/config/`: 설정 파일 - `src/utils/`: 유틸리티 함수

백엔드 구조:

- `functions/src/index.ts`: 엔드포인트 등록
- `functions/src/member/`: 회원 관련 API
- `functions/src/vehicle/`: 차량 관련 API
- `functions/src/auction/`: 경매 관련 API
- `functions/src/trade/`: 거래 관련 API
- `functions/src/config/`: 설정 (Secret Manager)

상세 디렉토리 구조는 부록 10.4 를 참조하시기 바랍니다¹.

3.3 컴포넌트 구조

메인 애플리케이션:

- 단일 진입점에서 화면 라우팅 관리
- 화면 전환 함수를 통한 상태 관리
- 차량 식별자 등 파라미터 전달

¹ 부록 10.4: 구현 코드 참조

페이지 컴포넌트:

- 각 화면별 독립적인 컴포넌트 파일
- 화면 전환 콜백 함수를 통한 네비게이션
- 필요한 파라미터(차량 식별자 등) 전달

3.4 상태 관리 방식

- **로컬 상태:** React Hook 을 통한 컴포넌트 내부 상태 관리
 - **전역 상태:** 애플리케이션 레벨의 공유 상태(현재 차량 식별자, 편집 중인 차량 식별자)
 - **데이터 페칭:** 각 컴포넌트에서 비동기 데이터 로드
 - **Mock 데이터:** 프로토타입 단계에서 Mock 데이터 서비스 활용
-

4. 화면 명세

4.1 화면 목록

총 27 개의 화면이 구현되어 있습니다.

| 화면 식별자 | 화면명 | 컴포넌트 위치 | 상태 |
|------------|---------------|-------------------------------|---|
| SCR-0000 | 랜딩 페이지 | 메인 파일 (LandingPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0001 | 로그인 | 메인 파일 (LoginPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0002 | 회원가입 (약관) | 메인 파일 (SignupTermsPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0002-1 | 회원가입 약관 | 메인 파일 (SignupTermsPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0002-2 | 회원가입 정보 입력 | 메인 파일 (SignupInfoPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0003-1 | 승인 대기 | 메인 파일 (ApprovalStatusPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0003-2 | 승인 완료 | 메인 파일 | <input checked="" type="checkbox"/> 구현됨 |

| 화면 식별자 | 화면명 | 컴포넌트 위치 | 상태 |
|-------------------|---------------|--|---|
| | | (ApprovalStatusPage) | |
| SCR-0100 | 딜러 대시보드 | 메인 파일 (DashboardPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0101 | 차량 목록 | src/components/VehicleListPage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0102 | 일반 판매 제안 목록 | src/components/GeneralSaleOffersPage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0103 | 판매 내역 | src/components/SalesHistoryPage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0104 | 정산 내역 | src/components/SettlementListPage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0105 | 정산 상세 | src/components/SettlementDetailPage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0200 | 차량 등록 | 메인 파일 (RegisterVehiclePage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0200-Draft | 임시 저장 목록 | 메인 파일 (VehicleDraftListPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0201 | 검차 신청 | 메인 파일 (InspectionRequestPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0201-Progress | 검차 진행 상태 | 메인 파일 (InspectionStatusPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0202 | 검차 결과 조회 | 메인 파일 (InspectionReportPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0300 | 판매 방식 선택 | 메인 파일 (SalesMethodPage) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0301-N | 일반 판매 - 분석 중 | 메인 파일 (GeneralSalePageLoading) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0302-N | 일반 판매 - 가격 설정 | 메인 파일 (GeneralSalePagePrice) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0303-N | 일반 판매 - 완료 | 메인 파일 (GeneralSalePageComplete) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0400 | 경매 상세 | 메인 파일 (AuctionDetailPage) | <input checked="" type="checkbox"/> 구현됨 |

| 화면 식별자 | 화면명 | 컴포넌트 위치 | 상태 |
|------------|-------------|--|---|
| SCR-0401-A | 경매 - 시작가 설정 | 메인 파일 (AuctionSalePageStartPrice) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0402-A | 경매 - 기간 설정 | 메인 파일 (AuctionSalePageDuration) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0403-A | 경매 - 완료 | 메인 파일 (AuctionSalePageComplete) | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0600 | 택송 예약/배차 | src/components/LogisticsSchedulePage.tsx | <input checked="" type="checkbox"/> 구현됨 |
| SCR-0601 | 택송 내역 | src/components/LogisticsHistoryPage.tsx | <input checked="" type="checkbox"/> 구현됨 |

4.2 주요 화면 전환 플로우

플로우-01: 회원가입 플로우

랜딩 페이지 (SCR-0000)

- 로그인 (SCR-0001)
- 회원가입 약관 (SCR-0002)
- 회원가입 정보 입력 (SCR-0002-2)
- 승인 대기 (SCR-0003-1)
- 승인 완료 (SCR-0003-2)
- 딜러 대시보드 (SCR-0100)

플로우-02: 검차 플로우

차량 등록 (SCR-0200)

- 검차 신청 (SCR-0201)
- 검차 진행 상태 (SCR-0201-Progress)
- 검차 결과 조회 (SCR-0202)

플로우-03: 판매 플로우

검차 결과 조회 (SCR-0202)

- 판매 방식 선택 (SCR-0300)
 - |─ 일반 판매: 분석 중 (SCR-0301-N) → 가격 설정 (SCR-0302-N) → 완료 (SCR-0303-N) → 일반 판매 제안 목록 (SCR-0102)

└ 경매: 시작가 설정 (SCR-0401-A) → 기간 설정 (SCR-0402-A) → 완료 (SCR-0403-A) → 경매 상세 (SCR-0400)

플로우-04: 정산/탁송 플로우

정산 내역 (SCR-0104)

- 정산 상세 (SCR-0105)
- 탁송 예약 (SCR-0600)
- 탁송 내역 (SCR-0601)

전체 화면 전환 매트릭스는 부록 10.3 을 참조하시기 바랍니다².

4.3 주요 화면 상세 명세

SCR-0100: 딜러 대시보드

주요 기능:

- 통계 카드 표시 (차량 수, 판매 중, 경매 중 등)
- 빠른 액션 버튼 (매물 등록, 차량 목록 등)
- 최근 활동 목록

화면 전환:

- 매물 등록 → SCR-0200
- 차량 목록 → SCR-0101
- 일반 판매 제안 → SCR-0102
- 판매 내역 → SCR-0103
- 정산 내역 → SCR-0104
- 탁송 내역 → SCR-0601

² 부록 10.3: 화면 전환 매트릭스

SCR-0200: 차량 등록

주요 기능:

- 등록원부 OCR (Gemini API 연동)
- 차량 기본정보 입력
- 시세 추정 (Gemini API 연동)
- 검차 요청 버튼

API 호출:

- 등록원부 OCR API 호출
- 시세 추정 API 호출

화면 전환:

- 검차 요청 → SCR-0201 (차량 식별자 전달)

SCR-0201: 검차 신청

주요 기능: - 검차 장소 입력 - 방문 희망 일시 선택 - 예약 확정

API 호출: - 검차 신청 API 호출

화면 전환: - 예약 확정 → SCR-0201-Progress (차량 식별자 전달)

SCR-0202: 검차 결과 조회

주요 기능: - 검차 리포트 상세 조회 - AI 종합 진단 표시 - 평가사 정보 표시 - 미디어 갤러리 (사진/영상)

화면 전환: - 판매 등록 시작 → SCR-0300 (차량 식별자 전달)

SCR-0300: 판매 방식 선택

주요 기능:

- 일반 판매 / 경매 선택
- 각 방식별 설명 및 특징 표시

화면 전환: - 일반 판매 선택 → SCR-0301-N (차량 식별자 전달) - 경매 선택 → SCR-0401-A (차량 식별자 전달)

SCR-0600: 택송 예약/배차

주요 기능:

- 출발지 자동 채움 (차량 등록 주소)
- 도착지 자동 지정 (인천항 물류센터)
- 희망 날짜/시간 선택 - 특이사항 입력
- SKIP 버튼 (테스트용)

API 호출: - 택송 일정 조율 API 호출

화면 전환: - 예약 완료 → SCR-0601

SCR-0601: 택송 내역

주요 기능:

- 택송 내역 목록 조회
- 인계 승인 모달 (PIN 입력)
- 기사 정보 표시
- 차량 정보 표시

API 호출: - 인계 승인 API 호출

화면 전환: - 인계 승인 완료 → SCR-0105 (정산 상세, 차량 식별자 전달)

5. 기능 명세

5.1 회원 관련 기능

기능-01: 딜러 회원가입

화면: SCR-0002-2

기능 설명: 딜러가 회원가입을 수행합니다.

입력 데이터: - 이메일 주소 - 비밀번호 - 딜러명 - 전화번호 - 약관 동의 여부

출력 데이터: - 성공 여부 - 회원 식별자 - 메시지

API: 회원가입 API 호출

기능-02: 사업자 인증

화면: SCR-0003-1

기능 설명: 사업자등록증 이미지를 업로드하여 OCR 처리 및 진위 확인을 수행합니다.

입력 데이터: 사업자등록증 이미지 파일

출력 데이터: - 성공 여부 - 인증 완료 여부 - 사업자 정보 (회사명, 사업자등록번호, 대표자명) - 메시지

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 사업자 인증 API 호출

5.2 차량 관련 기능

기능-05: 차량 등록(등록원부 OCR)

화면: SCR-0200

기능 설명: 차량번호를 입력받아 등록원부에서 차량 기본정보를 OCR로 추출합니다.

입력 데이터: 차량번호

출력 데이터: - 차대번호(VIN) - 제조사 - 모델명 - 연식 - 주행거리

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 등록원부 OCR API 호출

기능-06: 검차 신청

화면: SCR-0201

기능 설명: 차량에 대한 검차 신청을 처리합니다.

입력 데이터: - 차량 식별자 - 희망 날짜 - 희망 시간

출력 데이터: - 성공 여부 - 검차 식별자 - 메시지

구현 상태: Firebase Functions 구현됨

API: 검차 신청 API 호출

데이터 저장: Firestore 의 검차 컬렉션에 저장

5.3 판매 관련 기능

기능-15: 판매 방식 변경

화면: SCR-0300, SCR-0400

기능 설명: 일반 판매에서 경매로 판매 방식을 변경하고 경매를 생성합니다.

입력 데이터: - 차량 식별자 - 경매 설정 (시작가, 즉시구매가)

출력 데이터: - 성공 여부 - 경매 식별자

구현 상태: Firebase Functions 구현됨

API: 판매 방식 변경 API 호출

데이터 저장: Firestore 의 경매 컬렉션에 새 문서 생성, 차량 컬렉션 업데이트

기능-16: 일반 판매 제안 수락/거절

화면: SCR-0102

기능 설명: 일반 판매 제안을 수락하거나 거절합니다.

입력 데이터: - 제안 식별자 - 동작 (수락/거절)

출력 데이터: - 성공 여부 - 메시지

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 제안 수락/거절 API 호출

기능-17: 바이어 최종 구매 의사 재확인

화면: SCR-0102

기능 설명: 바이어의 최종 구매 의사를 재확인합니다.

입력 데이터: - 제안 식별자 - 확인 여부

출력 데이터: - 성공 여부 - 메시지

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 구매 의사 확인 API 호출

5.4 경매 관련 기능

기능-18: 경매 입찰

화면: SCR-0400 (바이어 외부 시스템)

기능 설명: 경매에 입찰을 처리합니다. 동시성 제어를 위해 Firestore 트랜잭션을 사용합니다.

입력 데이터: - 경매 식별자 - 입찰 금액

출력 데이터: - 성공 여부 - 메시지

구현 상태: Firebase Functions 구현됨

API: 경매 입찰 API 호출

동시성 제어: Firestore 트랜잭션 사용

데이터 업데이트: Firestore 의 경매 컬렉션 업데이트 (최고가 비노출)

기능-19: 즉시구매

화면: SCR-0400 (바이어 외부 시스템)

기능 설명: 경매에서 즉시구매를 처리합니다. 원자성을 보장하기 위해 Firestore 트랜잭션을 사용합니다.

입력 데이터: 경매 식별자

출력 데이터: - 성공 여부 - 계약 식별자 - 메시지

구현 상태: Firebase Functions 구현됨

API: 즉시구매 API 호출

원자성 보장: Firestore 트랜잭션 사용

데이터 업데이트: Firestore 의 경매 컬렉션 및 차량 컬렉션 업데이트

5.5 탁송 관련 기능

기능-20: 탁송 일정 조율

화면: SCR-0600

기능 설명: 탁송 일정을 조율합니다.

입력 데이터: - 일정 날짜 - 일정 시간 - 주소

출력 데이터: - 성공 여부 - 일정 식별자

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 탁송 일정 조율 API 호출

기능-21: 배차 조율 및 확정

화면: SCR-0600, SCR-0601

기능 설명: 배차를 요청하고 확정합니다.

입력 데이터: 일정 식별자 또는 배차 식별자

출력 데이터: - 성공 여부 - 배차 식별자 (요청 시) - 기사 정보 (확정 시)

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 배차 요청 API 호출, 배차 확정 API 호출

기능-22: 인계 승인

화면: SCR-0601

기능 설명: 탁송 기사로부터 차량 인계를 승인합니다.

입력 데이터: - 탁송 식별자 - PIN (6 자리)

출력 데이터: - 성공 여부 - 인계 시각

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 인계 승인 API 호출

보안: PIN 번호는 로그에서 마스킹 처리

5.6 정산 관련 기능

기능-23: 정산 완료 알림

화면: SCR-0105

기능 설명: 정산 완료 알림을 전송합니다.

입력 데이터: 정산 식별자

출력 데이터: - 성공 여부 - 알림 식별자

구현 상태: Mock 응답 반환 (프로토타입 단계)

API: 정산 완료 알림 API 호출

5.7 AI 서비스 기능

AI 서비스: 사업자등록증 OCR

기능 설명: 사업자등록증 이미지에서 정보를 추출합니다.

입력 데이터: 이미지 파일

출력 데이터: - 회사명 - 사업자등록번호 - 대표자명

모델: gemini-3-pro-preview

구현 상태: 구현됨

AI 서비스: 등록원부 OCR

기능 설명: 자동차 등록원부 이미지에서 차량 정보를 추출합니다.

입력 데이터: 이미지 파일

출력 데이터: - 차량번호 - 차대번호(VIN) - 제조사 - 모델명 - 연식 - 연료 종류 - 등록일자 - 주행거리

모델: gemini-3-pro-preview

구현 상태: 구현됨

AI 서비스: 시세 추정

기능 설명: 차량 모델 및 연식을 기반으로 시세를 추정합니다.

입력 데이터: - 모델명 - 연식

출력 데이터: - 시세 정보 텍스트 - 참고 자료 목록

모델: gemini-3-flash-preview

도구: Google Search 연동

구현 상태: 구현됨

6. API 설계 개요

6.1 API 기본 URL

프로덕션: <https://asia-northeast3-carivdealer.cloudfunctions.net>

개발: (환경 변수로 설정 가능)

6.2 인증 방식

현재 상태: 인증 미구현 (프로토타입 단계)

계획: Firebase Auth 토큰 기반 인증 (향후 구현)

6.3 공통 헤더

Content-Type: application/json

6.4 에러 처리

에러 응답 형식:

```
{  
  "error": "에러 메시지"  
}
```

HTTP 상태 코드: - 200: 성공 - 400: 잘못된 요청 - 404: 리소스 없음 - 405: 허용되지 않은 메서드 - 500: 서버 오류

6.5 CORS 설정

- Firebase Functions v2에서 CORS 자동 처리 옵션 사용
- 모든 엔드포인트에서 CORS 허용

상세 API 명세는 별도 API 명세서 문서를 참조하시기 바랍니다.

7. 데이터 모델 설계

7.1 주요 엔터티

엔터티-01: 차량(Vehicle)

설명: 차량 정보를 관리하는 엔터티입니다.

주요 속성:

- 식별자 - 상태 (초안/검차 중/입찰 중/판매 완료/정산 대기/활성 판매) - 차량번호 - 모델명
- 제조사 - 연식 - 주행거리 - 가격 - 최고 입찰가 (선택) - 썸네일 URL (선택) - 업데이트 일시

상태 값:

- 초안 (draft) - 검차 중 (inspection) - 입찰 중 (bidding) - 판매 완료 (sold) - 정산 대기 (pending_settlement) - 활성 판매 (active_sale)

상세 스키마는 부록 10.2 를 참조하시기 바랍니다.³.

엔터티-02: 제안(Offer)

설명: 일반 판매 제안 정보를 관리하는 엔터티입니다.

주요 속성: - 식별자 - 입찰자명 - 금액 - 일자 - 최고가 여부 (선택) - 차량 식별자 - 만료 일시 (선택)

상세 스키마는 부록 10.2 를 참조하시기 바랍니다.⁴.

³ 부록 10.2: 엔터티 스키마 상세 - 차량

⁴ 부록 10.2: 엔터티 스키마 상세 - 제안

엔터티-03: 검차 리포트 (InspectionReport)

설명: 검차 결과 정보를 관리하는 엔터티입니다.

주요 속성: - 식별자 - 차량 식별자 - 평가사 정보 (이름, 식별자, 평점, 전화번호) - 요약 - 점수 - 상태 (외관, 내관, 기계, 차체) - AI 분석 (장점 목록, 단점 목록, 시장 판단) - 미디어 목록 (카테고리, 개수, 항목 목록)

상세 스키마는 부록 10.2 를 참조하시기 바랍니다 ⁵.

7.2 Firestore 컬렉션 구조

차량 컬렉션 (vehicles/): - 문서 식별자: 차량 식별자 - 필드: 식별자, 상태, 차량번호, 모델명 등

검차 컬렉션 (inspections/): - 문서 식별자: 검차 식별자 - 필드: 차량 식별자, 희망 날짜, 희망 시간, 상태, 생성 일시

경매 컬렉션 (auctions/): - 문서 식별자: 경매 식별자 - 필드: 차량 식별자, 시작가, 즉시구매가, 현재 최고 입찰가, 상태, 생성 일시

상세 컬렉션 구조는 부록 10.2 를 참조하시기 바랍니다 ⁶.

8. 환경 설정

8.1 환경 변수 목록

프론트엔드 (.env.local):

⁵ 부록 10.2: 엔터티 스키마 상세 - 검차 리포트

⁶ 부록 10.2: 엔터티 스키마 상세 - Firestore 컬렉션 구조

- Gemini API 키 - Firebase API 키 - Firebase 인증 도메인 - Firebase 프로젝트 식별자 -
Firebase 스토리지 버킷 - Firebase 메시징 발신자 식별자 - Firebase 앱 식별자 - Firebase
측정 식별자 - API 기본 URL

백엔드 (GCP Secret Manager): - Gemini API 키

8.2 설정 파일

빌드 설정 파일: - 환경 변수 로드 및 정의 - 경로 별칭 설정 - React 플러그인 설정

TypeScript 설정 파일: - TypeScript 컴파일 옵션 - 경로 매펑 설정

Firebase 설정 파일: - Firebase 프로젝트 설정 - Hosting, Functions, Firestore 설정

9. 배포 전략

9.1 빌드 프로세스

프론트엔드 빌드: 1. 의존성 설치 2. 빌드 실행 3. 빌드 결과물 생성 (dist/ 폴더)

백엔드 빌드: 1. Functions 디렉토리로 이동 2. 의존성 설치 3. 빌드 실행 4. 빌드 결과물 생성 (functions/lib/ 폴더)

9.2 배포 프로세스

전체 배포: - Firebase CLI 를 통한 전체 배포

선택적 배포: - 프론트엔드만 배포 - 백엔드만 배포 - Firestore 규칙만 배포

9.3 배포 환경

- **프로덕션 URL**: <https://carivdealer.web.app>
 - **Functions 리전**: asia-northeast3
 - **Firestore 리전**: asia-northeast3
-

10. 부록

10.1 용어집

| 용어 | 설명 |
|----------|---|
| 검차 | 차량 상태를 '증거'로 만들기 위한 평가사 수행 점검 |
| 등록원부 OCR | 자동차 등록원부 이미지를 OCR로 처리하여 차량 기본정보를 자동 추출하는 기능 |
| 낙찰가 | 경매 종료 후 최종 낙찰가. 딜러와 낙찰 바이어(기업)에게만 공개 |
| 즉시구매 | 경매 내에서 즉시구매가로 바로 구매하는 기능 |
| 판매 방식 변경 | 일반 판매로 등록된 차량을 경매 방식으로 변경하는 기능 |
| PIN | 인계 승인 시 사용하는 6 자리 보안 번호 |
| 택송 | 차량을 물류센터로 운송하는 서비스 |
| 배차 | 택송 기사를 배정하는 과정 |
| 인계 승인 | 택송 기사로부터 차량 인계를 승인하는 과정 |
| 정산 | 판매 대금을 정리하여 딜러에게 지급하는 과정 |

10.2 엔터티 스키마 상세

차량(Vehicle) 스키마

```
interface Vehicle {  
    id: string;  
    status: 'draft' | 'inspection' | 'bidding' | 'sold' | 'pending_settlement'  
    | 'active_sale';  
    plateNumber: string;  
    modelName: string;
```

```
    manufacturer: string;
    modelYear: string;
    mileage: string;
    price: string;
    highestBid?: string;
    thumbnailUrl?: string;
    updatedAt: string;
    fuelType?: string;
    registrationDate?: string;
    color?: string;
    vin?: string;
    inspectionId?: string;
    location?: string;
    endTime?: string;
    offers?: Offer[];
}
}
```

제안(Offer) 스키마

```
interface Offer {
    id: string;
    bidderName: string;
    amount: string;
    date: string;
    isHighest?: boolean;
    vehicleId: string;
    expiresAt?: string;
}
}
```

검사 리포트(InspectionReport) 스키마

```
interface InspectionReport {
    id: string;
    vehicleId: string;
    evaluator: {
        name: string;
        id: string;
        rating: number;
        phone: string;
    };
    summary: string;
    score: string;
    condition: {
        exterior: string;
        interior: string;
        mechanic: string;
        frame: string;
    };
    aiAnalysis: {

```

```

    pros: string[];
    cons: string[];
    marketVerdict: string;
};

media: {
    category: string;
    count: number;
    items: { type: 'image' | 'video', url: string, label: string }[];
}[];
}

```

Firebase 컬렉션 구조

```

vehicles/
{vehicleId}/
- id, status, plateNumber, modelName, ...

inspections/
{inspectionId}/
- vehicleId, preferredDate, preferredTime, status, createdAt

auctions/
{auctionId}/
- vehicleId, startPrice, buyNowPrice, currentHighestBid, status, createdAt

```

10.3 화면 전환 매트릭스

| 출발 화면 | 도착 화면 | 트리거 | 조건 |
|-------------------|-------------------|-------------|-------------|
| SCR-0000 | SCR-0001 | 로그인 버튼 | - |
| SCR-0001 | SCR-0002 | 회원가입 버튼 | - |
| SCR-0002-2 | SCR-0003-1 | 신청하기 버튼 | 정보 입력 완료 |
| SCR-0003-2 | SCR-0100 | 시작하기 버튼 | 승인 완료 |
| SCR-0100 | SCR-0200 | 매물 등록 버튼 | - |
| SCR-0200 | SCR-0201 | 검차 요청 버튼 | 차량 정보 저장 완료 |
| SCR-0201 | SCR-0201-Progress | 예약 확정 버튼 | 검차 신청 완료 |
| SCR-0201-Progress | SCR-0202 | 결과 확인 버튼 | 진행률 100% |
| SCR-0202 | SCR-0300 | 판매 등록 시작 버튼 | 검차 완료 |

| 출발 화면 | 도착 화면 | 트리거 | 조건 |
|------------|------------|-------------|------------|
| SCR-0300 | SCR-0301-N | 일반 판매 선택 | 판매 방식 = 일반 |
| SCR-0300 | SCR-0401-A | 경매 선택 | 판매 방식 = 경매 |
| SCR-0303-N | SCR-0102 | 제안 목록 보기 버튼 | 일반 판매 완료 |
| SCR-0403-A | SCR-0400 | 경매 생성 완료 | 경매 설정 완료 |
| SCR-0105 | SCR-0600 | 탁송 신청 버튼 | 정산 완료 |
| SCR-0600 | SCR-0601 | 예약 완료 | 탁송 일정 확정 |
| SCR-0601 | SCR-0105 | 인계 승인 완료 | PIN 입력 완료 |

10.4 구현 코드 참조

프론트엔드 메인 파일: index.tsx

프론트엔드 컴포넌트: - src/components/VehicleListPage.tsx -
src/components/GeneralSaleOffersPage.tsx - src/components/SalesHistoryPage.tsx
- src/components/SettlementListPage.tsx -
src/components/SettlementDetailPage.tsx -
src/components/LogisticsSchedulePage.tsx -
src/components/LogisticsHistoryPage.tsx

API 클라이언트: src/services/api.ts

AI 서비스: src/services/gemini.ts

백엔드 Functions: - functions/src/index.ts (엔드포인트 등록) -
functions/src/member/verifyBusiness.ts - functions/src/vehicle/inspection.ts -
functions/src/vehicle/ocrRegistration.ts - functions/src/auction/bid.ts -
functions/src/auction/buyNow.ts - functions/src/trade/changeSaleMethod.ts

설정 파일: - vite.config.ts - tsconfig.json - firebase.json

25.12.29 류지환 작성

문서 끝.