Date. 2025.09.25

현대오토에버 모빌리티 SW 스쿨_3조

Trever

끝나지 않는 거래 이야기

중고차 거래 플랫폼

팀장 : **오창은**

베스트 팀원 : **김태민**

팀원 : **이지수, 양지선, 채상윤, 최동진**

고용노동부 HYUNDAI AutoEver HYUNDAI NGV RAPA 한국전파진흥협회

목차

프로젝트 배경

1. 거래 절차 복잡

- 다수 앱이 여러 단계를 거쳐야 계약이 완료됨
- 계약서 작성이 별도의 프로세스로 진행되어 사용자 불편 발생

2. 계약절차의 불투명성

- 서류 작성 과정이 자동화 되지 않아 거래 조건이 명확하지 않은 경우
- 사용자 간 오해 또는 분쟁 가능성 존재

“간편한 거래 및 계약 절차를 통한

거래 신뢰성 확보”

프로젝트 배경

1. 딜러 중심 시세 선정

- 매입 딜러들이 과거 거래 경험과 시장 흐름으로 가격 책정

2. 온라인 시세 제공 서비스

- 빅데이터(최근 거래 이력, 차량 조건, 시장 수요) 기반 평균값 산출
- 비교적 투명해 보이지만, 실제 거래가와 차이가 있을 수 있음

3. 플랫폼 자체 산정 시세

- 특정 플랫폼이 보유한 거래 데이터, 조회수, 인기 등을 반영해 산출
- 플랫폼마다 기준이 다르므로 “표준 시세"라 하기 어려움

“기존 중고차 시세 선정 방식에 대한 소비자 불만 해소”

핵심 서비스

경매 시스템

- **입찰 방식을 통한 공정한 가격 형성과 경쟁 유도**
- 투명한 가격 결정으로 사용자 신뢰와 참여도 상승
- **자동 입찰 로직과 실시간 상태 표시 기능**에 중점 둠

디지털 계약서

- 거래 확정 후 법적 신뢰성 보장할 장치 필요함
- **계약서를 자동 생성해 사용자 편의성과 안전성 높임**
- 자동 PDF 발급과 안전한 저장 방식에 중점 둠

찜하기

- **관심 차량 저장으로 불필요한 반복 검색 감소**
- 구매 결정 전 차량 간 비교와 재확인 쉽게 가능
- 개인화된 저장 기능과 직관적 접근성에 중점

거래 진행

- 거래 과정을 단계별로 보여줘 소비자 불안감 감소
- **‘구매 요청 → 판매자 승인 → 거래 확정’**의 거래 과정으로 **거래 투명성 강화**
- 단계별 알림과 진행 상태 시각화에 중점 둠

핵심 서비스

차량 등록

- 매물을 쉽게 등록할 수 있어야 플랫폼 활성화 가능
- **간단히 입력할 수 있도록 직관적인 흐름 설계**
- 사용자 친화적인 등록 UI에 중점

검색 필터

- **조건별 탐색**으로 **사용자의 효율적 선택 지원**
- 원하는 차량을 빠르게 걸러 낼 수 있도록 다양한 필터 지원

검색

- 다양한 조건에서 차량을 빠르게 찾을 수 있어야 거래 효율성 증대
- 불필요한 탐색 시간을 줄이기 위해 **최근 검색어 지원**
- 직관적인 UI와 빠른 검색 성능에 중점

상세 조회

- 구매 전 의사 결정을 위해 충분한 정보 제공
- **차량 이미지와 세부 스펙을 시각적으로 보기 쉽게 구성**
- 정보 전달력과 가독성 높은 UI에 중점

요구사항 정의 - 필수 기능

01

회원가입 및 로그인

- 구글 Oauth 기반 계정 생성 및 로그인 지원
- 토큰 인증으로 안전한 로그인 환경 제공

02

차량 등록

- 상세 정보 입력과 사진 업로드로 차량 매물 등록
- 모델·연식·주행거리·가격·사진(최대 5장) 입력 지원

03

차량 목록 조회 & 키워드 검색

- 등록된 차량을 최신순으로 정렬하여 제공
- 키워드 검색으로 원하는 차량을 빠르게 탐색 가능

04

차량 상세 보기

- 차량 스펙, 등록된 사진, 판매자 정보까지 한눈에 확인
- 거래 의사결정을 위한 충분한 정보 제공

05

거래 진행

- 구매 요청 → 판매자 승인 → 거래 상태 변경(진행/완료)으로 거래 확정

요구사항 정의 - 추가 기능

01

검색 필터링

- 연식·가격·주행거리·브랜드 조건별 탐색 구현
- 구매자가 원하는 조건의 차량을 빠르게 좁혀나갈 수 있음

02

관심 차량 찜하기

- 관심 매물 저장 및 관리 기능 제공

03

마이페이지

- 사용자가 진행 중인 거래, 완료된 거래 내역 한눈에 확인
- 찜한 차량과 계약서를 관리할 수 있어 편의성 강화

04

경매 시스템★★★

- 자동 입찰 기능을 통해 공정한 가격 산출 지원
- 종료 시 최고 입찰자와 거래 자동 확정

05

디지털 계약서★★

- 거래 확정 시 계약서 자동 발급
- 종이 없이도 간편하게 계약 관리, 사용자 편의성 강화

팀 구성 및 역할 분배

김태민
안드로이드 개발

- 구글 로그인 구현
- 일반 거래 및 경매 구현
- 차량 검색 및 필터링 구현
- 계약서 구현
- 찜하기 구현

이지수
안드로이드 개발

- 내차 팔기 페이지 구현
- 차량 등록 플로우 구현
- 마이페이지 구현

채상윤
iOS 개발

- 구글 로그인 구현
- 일반 거래 구현
- 경매 구현
- 마이페이지 구현
- 찜하기 구현
- 계약서 구현

오창은
iOS 개발

- 차량 등록 플로우 구현
- 차량 검색 및 필터링 구현

최동진
백엔드 개발

- 경매 시작, 마감 구현
- 입찰 처리 구현
- 차량 등록, 검색, 상세 조회 구현

양지선
백엔드 개발

- 계약서 생성, 조회 구현
- 최근 검색 구현
- 거래 흐름 구현
- 차량 찜하기 구현

개발 프로세스

애자일 방법론 적용

반복적 개선과 빠른 대응을 위한 애자일 프로세스 도입

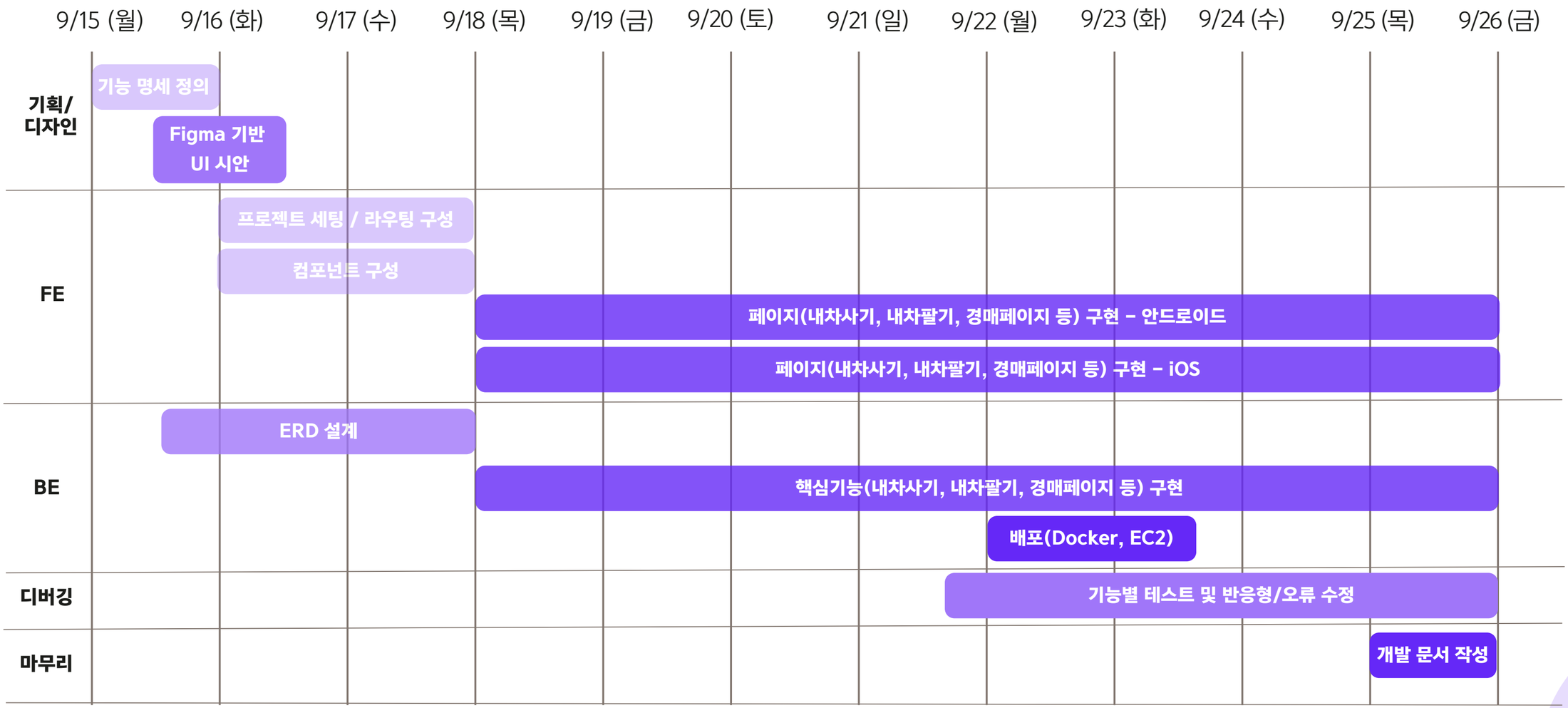
정기적인 스크럼 미팅

매일 정기적인 회의로 진행 상황 공유 및 요구사항 조율

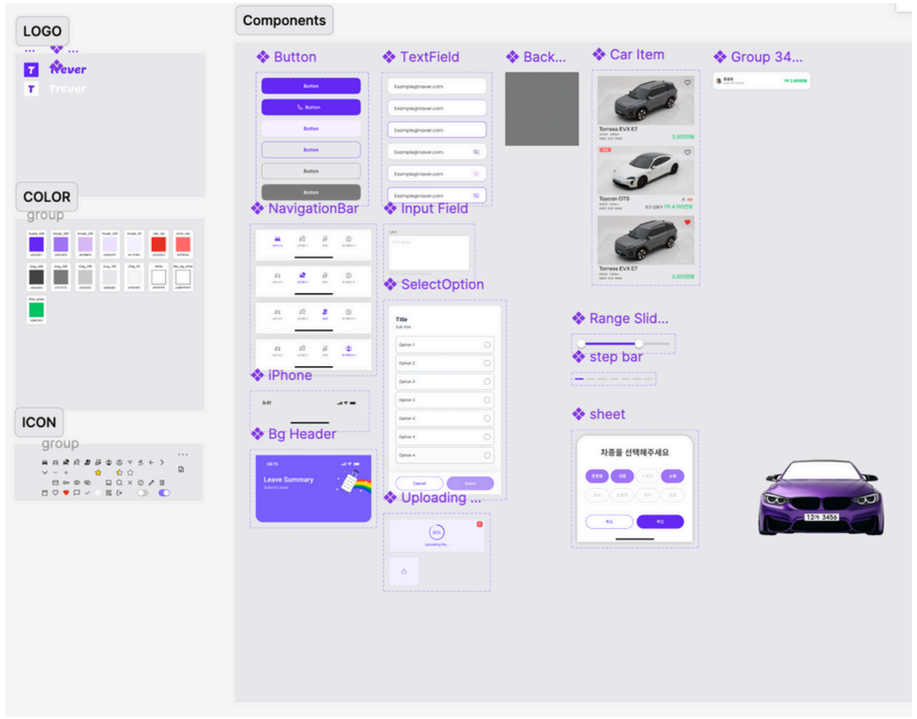
지속적인 피드백 반영

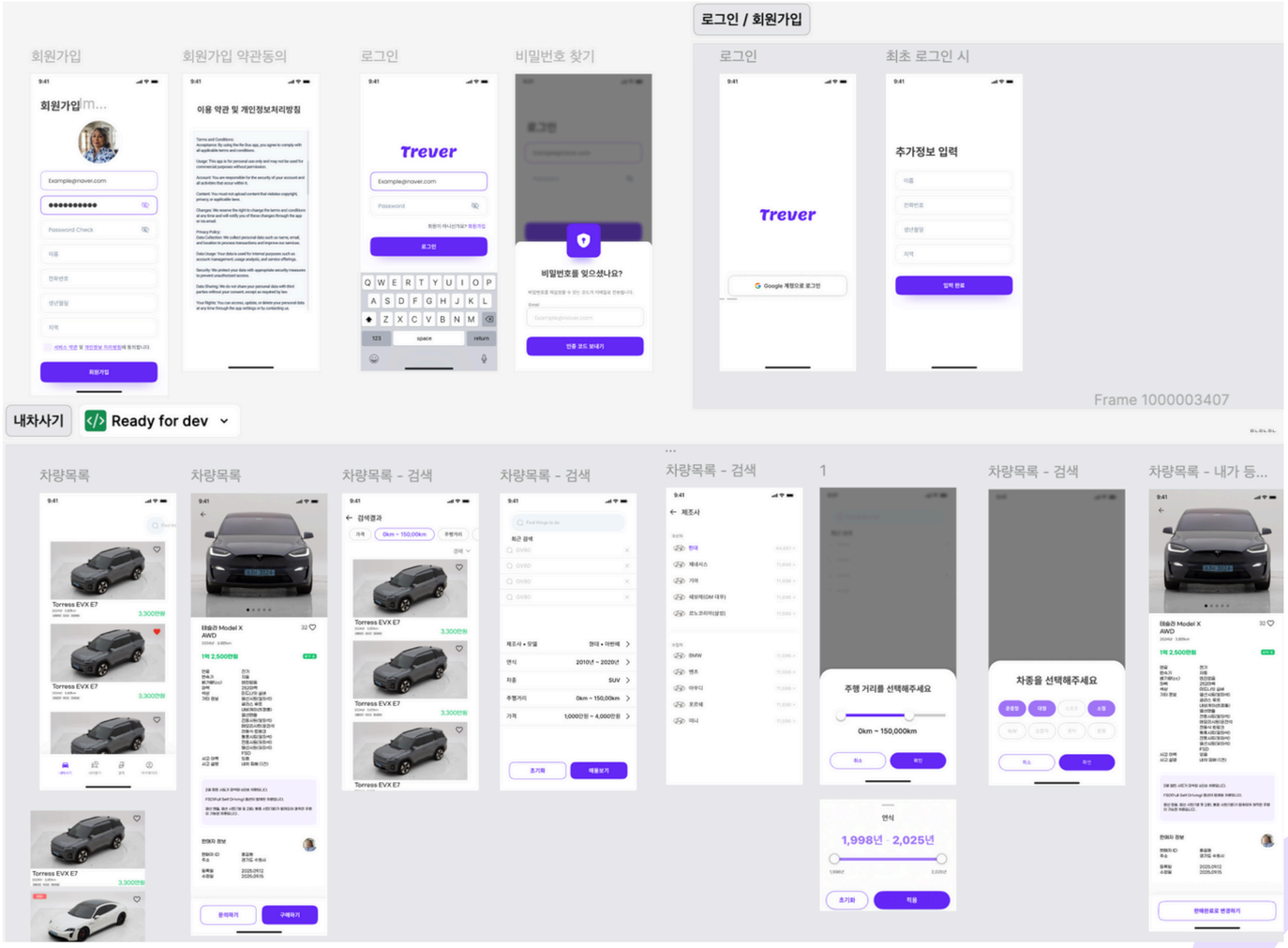
기능 테스트와 사용자 피드백을 즉시 반영하여 개선

개발 일정


9/15 (월)
9/16 (화)
9/17 (수)
9/18 (목)
9/19 (금)
9/20 (토)
9/21 (일)
9/22 (월)
9/23 (화)
9/24 (수)
9/25 (목)
9/26 (금)
기획/ 디자인
기능 명세 정의
Figma 기반 UI 시안
FE
프로젝트 세팅 / 라우팅 구성
컴포넌트 구성
페이지(내차사기, 내차팔기, 경매페이지 등) 구현 – 안드로이드
페이지(내차사기, 내차팔기, 경매페이지 등) 구현 – iOS
BE
ERD 설계
핵심기능(내차사기, 내차팔기, 경매페이지 등) 구현
배포(Docker, EC2)
디버깅
기능별 테스트 및 반응형/오류 수정
마무리
개발 문서 작성

UI/UX 디자인 시안


LOGO
Components
Button
TextField
Back...
Car Item
Group 34...
NavigationBar
Input Field
SelectOption
iPhone
Bg Header
Uploading ...
Range Slid...
step bar
sheet
COLOR
group
ICON
group


로그인 / 회원가입
회원가입
회원가입 약관동의
로그인
비밀번호 찾기
로그인
최초 로그인 시
Frame 1000003407
내차사기
Ready for dev
차량목록
차량목록
차량목록 - 검색
차량목록 - 검색
차량목록 - 검색
1
차량목록 - 검색
차량목록 - 내가 등...

기술 스택 및 아키텍처

Frontend (Android)
Kotlin, Android Studio를 사용하여 사용자 친화적인 Android 앱을 개발했습니다.

Frontend (iOS)
Swift, Xcode를 활용하여 직관적이고 반응성이 뛰어난 iOS 앱을 구현했습니다.

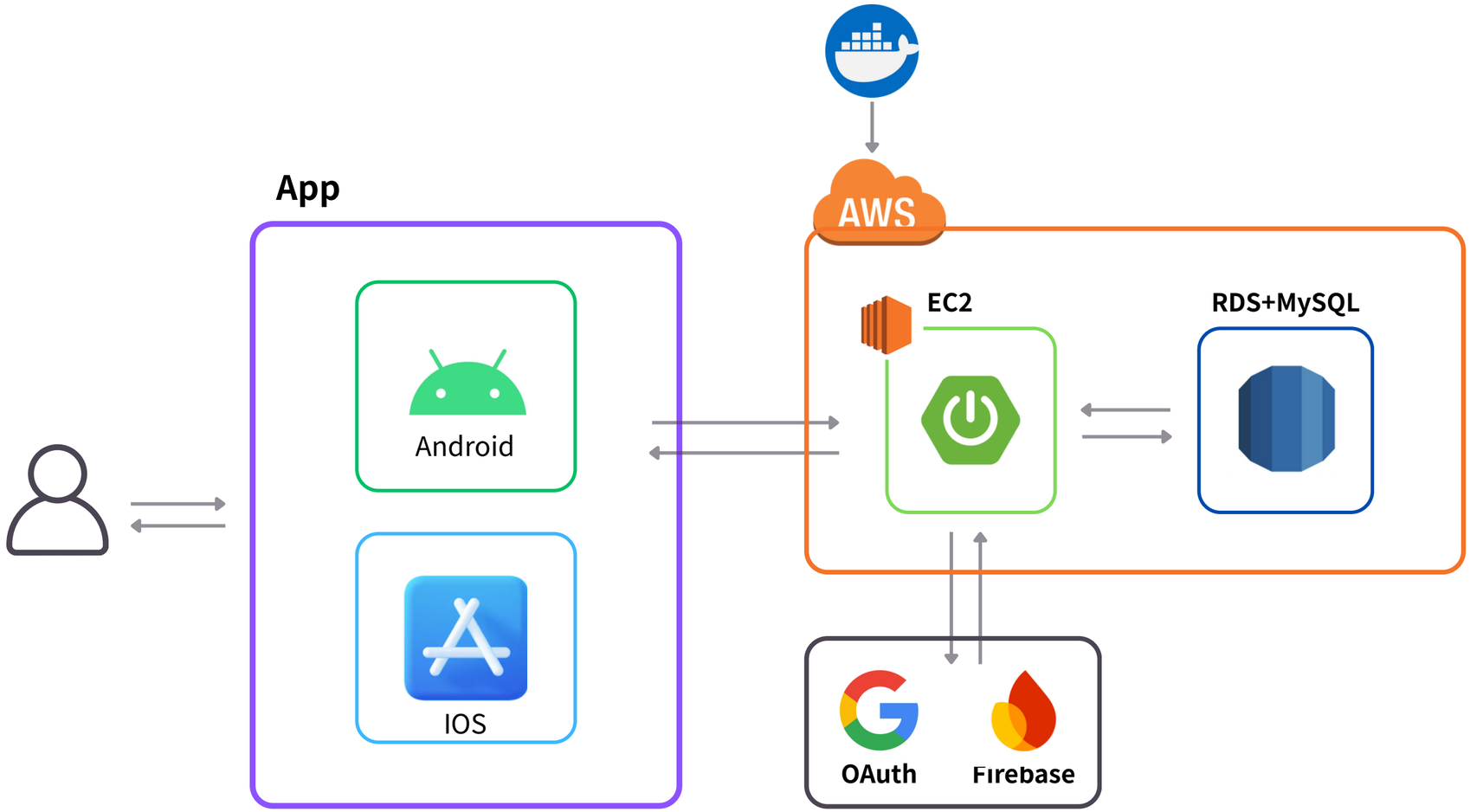
Backend (서버/API)
Spring Boot (Java) 기반으로 강력하고 확장 가능한 서버 및 API를 구축했습니다.

Database
MySQL 및 **Firebase Realtime DB**를 사용하여 경매 데이터의 실시간 업데이트를 지원합니다.

협업 도구
GitHub를 통해 효율적인 코드 관리와 팀원 간의 협업을 진행했습니다.

디자인 도구
Figma를 활용하여 사용자 인터페이스 및 경험을 설계했습니다.

기술 스택 및 아키텍처



기술 스택 및 아키텍처

Vehicle 차량 관리 API

GET	/api/vehicles	차량 목록
POST	/api/vehicles	차량 등록
POST	/api/vehicles/search	차량 검색
GET	/api/vehicles/{vehicleId}	차량 상세 조회
DELETE	/api/vehicles/{vehicleId}	차량 삭제
GET	/api/vehicles/my-vehicles	내가 등록한 차량 목록조회
GET	/api/vehicles/manufacturers	제조사별 차량 수 조회
GET	/api/vehicles/manufacturers/{manufacturer}/car-names	제조사별, 차명 별 차량 수 조회
GET	/api/vehicles/manufacturers/{manufacturer}/car-names/{carName}/car-models	제조사별, 차명,차 모델 별 차량 수 조회
GET	/api/vehicles/check-car-number	차량 번호 존재여부 조회

차량 관리 api 명세 문서

User User 관련 API 입니다.

POST	/api/v1/users/signup	회원가입 API
POST	/api/v1/users/reissue	토큰 재발급
POST	/api/v1/users/me/complete	회원 추가 정보 입력(완성) API
POST	/api/v1/users/logout	로그아웃 API
POST	/api/v1/users/login	로그인 API
POST	/api/v1/users/auth/google/login	
PATCH	/api/v1/users/profile	프로필 수정 API
GET	/api/v1/users/me	사용자 정보 조회 API

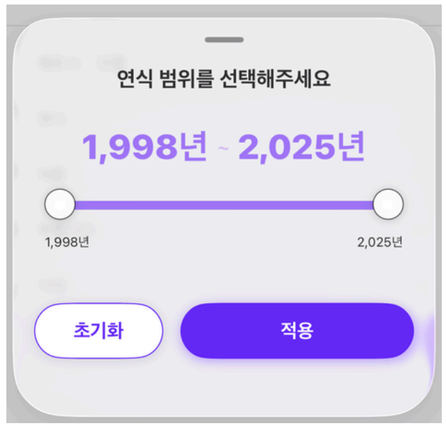
User api 명세 문서

시연

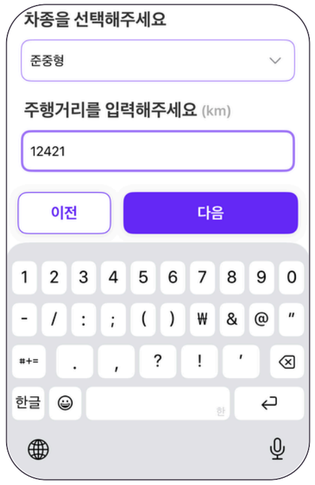
UI/UX 의도(프론트엔드 공통)

브랜드 차별성 확보

보라색을 메인 테마 컬러로 지정하여 **브랜드 아이덴티티를 강화**.
채도가 높은 컬러 특성상 과도한 시각적 피로를 줄이기 위해 **핵심 요소(버튼, 강조 포인트)에만 제한적으로 적용**.
나머지 UI 영역은 중립적인 색상 톤을 사용하여 시각적 안정성 유지.



브랜드 색을 적용한 바텀 시트



스텝 기반 입력

UI/UX 구성 전략

바텀 시트(Bottom Sheet) 활용
주요 입력 및 옵션 선택 영역을 바텀 시트 형태로 구현하여 화면 전환 최소화 및 사용성 향상.

숫자 입력 UX 개선
숫자 스텝 이동 시 원형 인디케이터가 함께 이동하는 형태(Progress Indicator with Stepper)를 적용하여 직관적인 입력 경험 제공.

스텝 기반 입력 흐름
입력 항목이 많다는 점을 고려하여 단계별 입력 프로세스(Step-by-Step Form)를 적용. 한 단계 입력 완료 후 다음 단계로 자동 이동 및 Focus 전환.
페이지 단위로 입력을 분리하여 사용자 피로도를 줄이고 진행 상황을 명확히 전달.

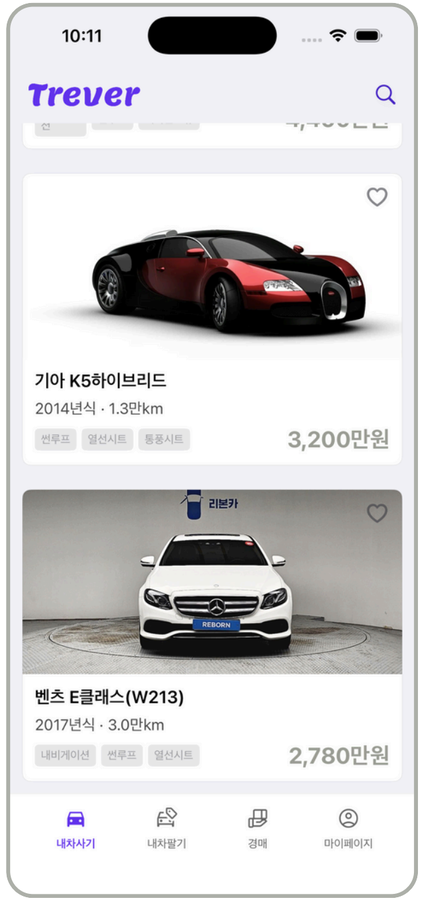
UI/UX 의도(프론트엔드 공통)

UI/UX 구성 전략

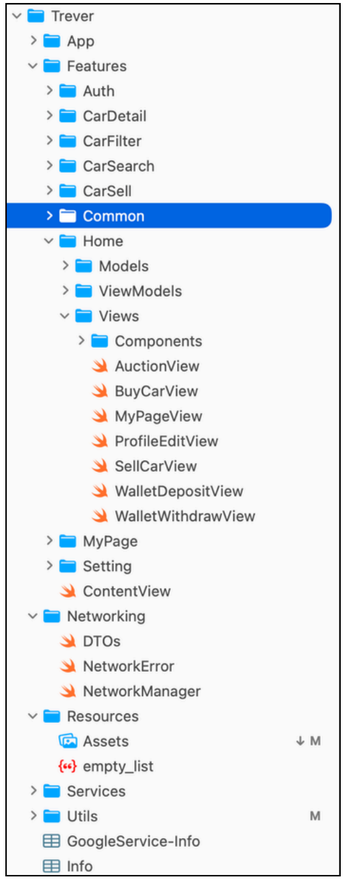
키보드 인터랙션 최적화
키보드 활성화 시 버튼 및 입력 위치가 가려지지 않도록 레이아웃 자동 조정.

차량 리스트 카드화(Card Layout)
차량 정보를 카드 단위로 표시하여 시각적 구분과 정보 전달력을 높임.

무한 스크롤(Infinite Scroll) 구현
대량의 데이터를 효율적으로 표시하기 위해 페이징 대신 무한 스크롤을 적용, 사용자 경험 개선.



구현(프론트엔드 공통)



Feature-Based MVVM 구조

기존 프로젝트 규모가 크지 않아
Layer-Based 구조를 통해 역할(Layer)별로 분리하였으나

협업 효율성과 모듈 재사용성을 위해 **책임 범위를 명확하게 구분 짓는** Feature-Based MVVM를 적용

구현(프론트엔드 공통)

```swift
struct ApiResponse<T: Decodable>: Decodable {
    let status: Int
    let success: Bool
    let message: String
    let data: T?
}
```

ApiResponse<T> (Response 관리)

- 모든 API 응답 구조(status, success, message, data)를 일관성 있게 관리
- data 부분만 제네릭으로 받아 다양한 모델(Car, User 등) 적용 가능
- 공통 처리(성공 여부, 에러 메시지 처리 등) 로직 단순화

APIEndpoint (Request 관리)

- 문자열 하드코딩 제거 → 오타 및 중복 방지
- Base URL 변경 시, 한 군데만 수정

```swift
enum APIEndpoint {
    static let baseURL = "https://www.trever.store/api"

    case vehicles
    case manufacturers
    case carNames
    case modelNames
    case years
    case recentSearch
    case deleteRecentSearch(keyword: String)
    case vehicleSearch
    case vehicleManufacturers
    case vehicleNames(manufacturer: String)
    case vehicleModels(manufacturer: String, carName: String)
    case vehicleCheckCarNumber(carNumber: String)
    case myVehicles(currentPage: Int, pageSize: Int)

    var url: String {
        switch self{
        case .vehicles:
            return "\(APIEndpoint.baseURL)/vehicles"
        case .manufacturers:
            return "\(APIEndpoint.baseURL)/cars/manufacturers" // 제조사
        case .carNames:
            return "\(APIEndpoint.baseURL)/cars/carnames" // 차명
```

구현(안드로이드)

```
composable(
    route = ROUTE_AUCTION_DETAIL,
    arguments = listOf(
        navArgument( name = "carId") { type = NavType.StringType },
        navArgument( name = "auctionId") { type = NavType.StringType }
    )
) { backStackEntry ->
    val carId = backStackEntry.arguments?.getString( key = "carId") ?: ""
    val auctionId = backStackEntry.arguments?.getString( key = "auctionId") ?: ""

    AuctionDetailScreen(
```

```
53      const val ROUTE_BID_HISTORY = "auction/bid-history/{auctionId}"
        6 Usages
54      const val ROUTE_LOGIN = "login"
        2 Usages
55      const val PROFILE_INPUT = "profile_input"
        3 Usages
56      const val ROUTE_SEARCH = "search"
        1 Usage
57      const val ROUTE_MAIN = "main"
        1 Usage
58      const val ROUTE_SEARCH_RESULTS = "search/results"
```

AppNavHost 기반 라우팅 및 화면 분리 구조

- AppNavHost에서 모든 주요 화면을 route별로 분리하여 관리
- 화면 이동 시 필요한 파라미터를 route에 포함시켜 동적 네비게이션 지원
- UI/UX 흐름이 명확해지고, 유지보수 및 확장성이 향상됨

구현(백엔드)

동시 입찰 처리 문제

- 여러 사용자가 동시에 입찰 시, DB의 같은 값이 동시에 수정 될 수 있음
- 순서 보장 불가 → 입찰 무효/덮어쓰기 위험 발생

해결 방법

초기 접근

- 단순 CRUD 업데이트

개선 방법

- 입력큐(Queue): 입찰 요청을 순서대로 정렬하여 처리
- 락(Lock): 같은 경매 아이템에 동시에 접근하지 못하도록 제어

→ 안정적인 입찰 순서와 데이터 정합성 보장

계약서 자동 생성

- 거래 확정 시 계약서 생성·관리 방식 고민
- 단순 저장만으로는 사용자 신뢰성과 편의성 부족

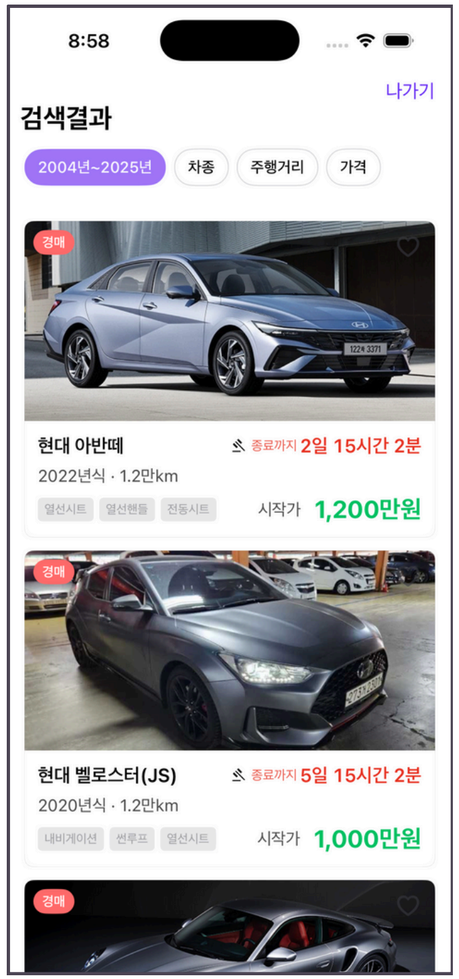
해결 방법

초기 접근

- 거래 완료 시 단순히 계약 정보만 DB에 저장

개선 방법

- PDF 자동 생성 라이브러리를 적용하여 계약서 생성
- 계약 정보(DB)와 연동해 PDF 파일 자동 발급
- URL 제공 방식으로 접근성을 높이고 사용자 편리성 강화



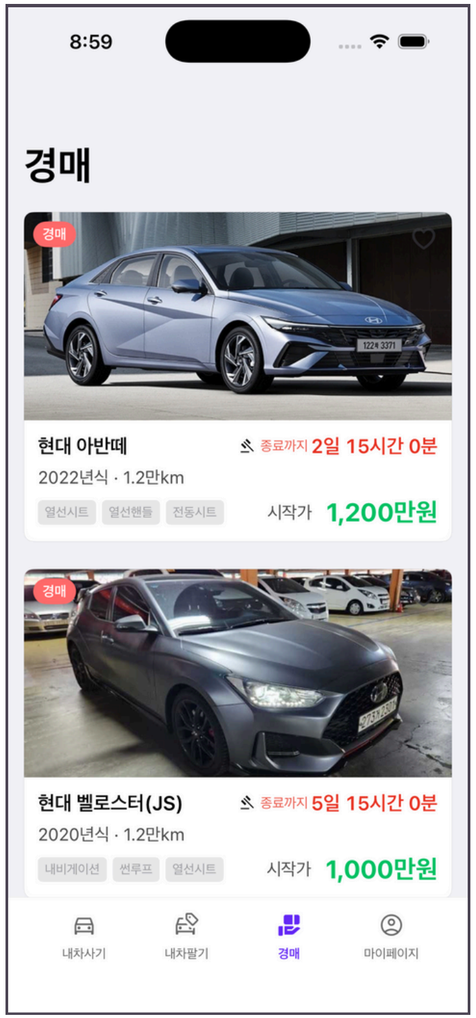
검색 필터링

차별화된 기능

사용자가 원하는 조건(차종, 연식, 가격 등)을 세밀하게 설정하여 차량을 빠르게 탐색할 수 있습니다.
키워드 검색과 함께 필터 기반 검색이 가능합니다.

사용자 효용 및 개선 포인트

- 원하는 차량을 빠르게 찾을 수 있어 탐색 시간을 단축합니다.
- 맞춤형 결과 제공으로 사용자 만족도가 향상됩니다.
- 향후 AI 추천 기능과 결합하여 더욱 정교한 검색 경험을 제공할 수 있습니다



실시간 거래 상태 표시

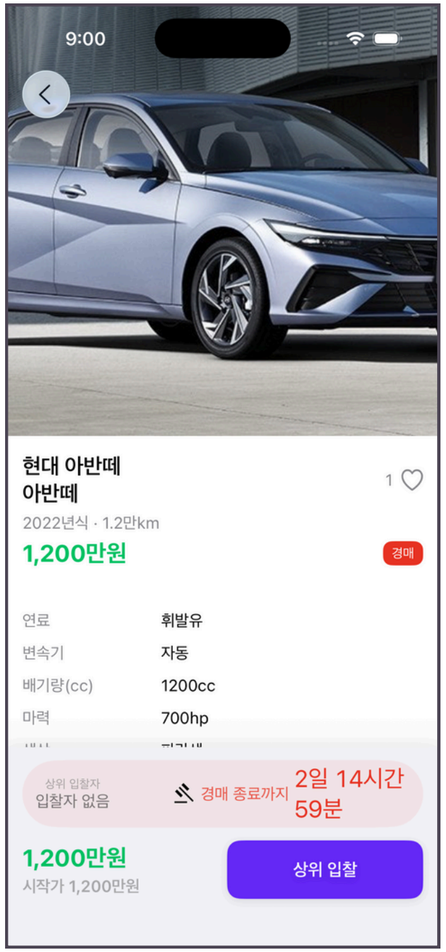
차별화된 기능

거래 진행 상황을 실시간으로 표시하여 사용자에게 현재 상태를 투명하게 제공합니다.
입찰 현황, 거래 진행 단계(검토, 계약, 결제 등)를 직관적인 UI로 표시합니다.

사용자 효용 및 개선 포인트

- 거래 진행 상황을 즉시 확인할 수 있어 불확실성이 줄고 신뢰도가 향상됩니다.
- 거래 과정이 가시화되어 적극적인 참여를 유도합니다.
- 향후 AI 기반 상태 예측 기능과 결합하면 보다 스마트한 거래 전략 제안이 가능합니다

경매 시스템



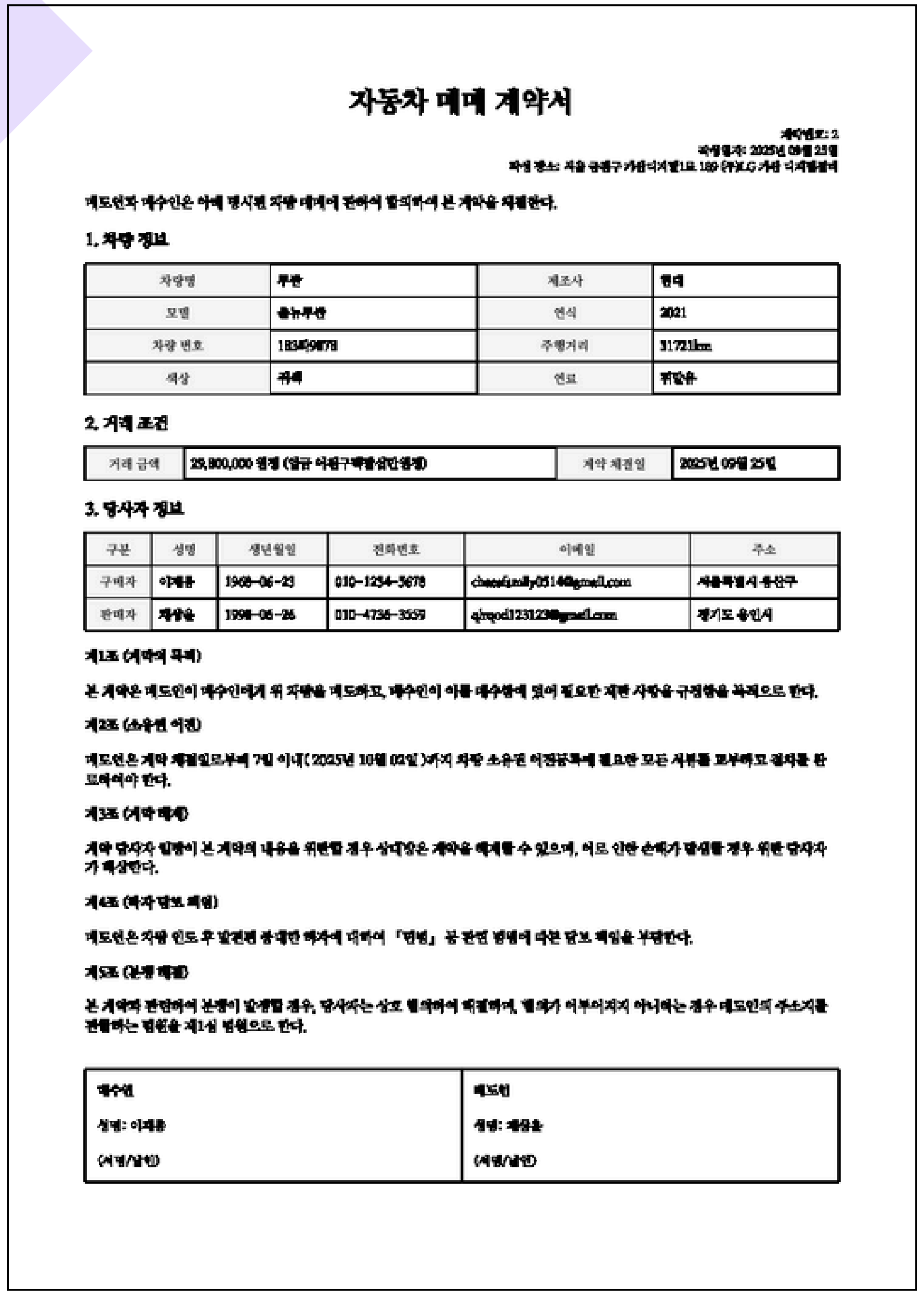
차별화된 기능

단순 매매를 넘어, 다수의 구매자가 입찰에 참여할 수 있는 경매 기능을 도입했습니다.
판매자는 더 높은 가격을, 구매자는 합리적인 가격에 접근할 수 있습니다.

사용자 효용 및 개선 포인트

- 시장 경쟁성을 도입하여 공정한 가격 형성을 유도합니다.
- 거래의 재미 요소를 더해 사용자 참여도를 높입니다.
- 추후 자동 입찰, 알림 기능 등을 추가해 편의성을 강화할 수 있습니다.

자동 계약서



자동차 매매 계약서

계약번호: 2
작성일자: 2025년 09월 25일
작성 장소: 서울 금천구 가산디지털1로 189 (주)LG 가산 디지털센터

매도인과 매수인은 아래 명시된 차량 매매에 관하여 합의하여 본 계약을 체결한다.

1. 차량 정보

차량명	투싼	제조사	현대
모델	올뉴투싼	연식	2021
차량 번호	183[illegible]9878	주행거리	31721km
색상	쥐색	연료	휘발유

2. 거래 조건

거래 금액	29,800,000 원정 (일금 이천구백팔십만원정)	계약 체결일	2025년 09월 25일

3. 당사자 정보

구분	성명	생년월일	전화번호	이메일	주소
구매자	[illegible]	1968-06-23	010-1234-5678	[illegible]@gmail.com	서울특별시 용산구
판매자	[illegible]	1998-06-26	010-4736-3559	[illegible]@gmail.com	경기도 용인시

제1조 (계약의 목적)

본 계약은 매도인이 매수인에게 위 차량을 매도하고, 매수인이 이를 매수함에 있어 필요한 제반 사항을 규정함을 목적으로 한다.

제2조 (소유권 이전)

매도인은 계약 체결일로부터 7일 이내(2025년 10월 02일)까지 차량 소유권 이전등록에 필요한 모든 서류를 교부하고 절차를 완료하여야 한다.

제3조 (계약 해제)

계약 당사자 일방이 본 계약의 내용을 위반할 경우 상대방은 계약을 해제할 수 있으며, 이로 인한 손해가 발생할 경우 위반 당사자가 배상한다.

제4조 (하자 담보 책임)

매도인은 차량 인도 후 발견된 중대한 하자에 대하여 「민법」 등 관련 법령에 따른 담보 책임을 부담한다.

제5조 (분쟁 해결)

본 계약과 관련하여 분쟁이 발생할 경우, 당사자는 상호 협의하여 해결하며, 협의가 이루어지지 아니하는 경우 매도인의 주소지를 관할하는 법원을 제1심 법원으로 한다.

매수인 성명: [illegible] (서명/날인)	매도인 성명: [illegible] (서명/날인)

차별화된 기능

거래 성사 시 계약서를 자동으로 PDF로 생성하고, 안전하게 저장할 수 있는 기능을 제공합니다.
법적 효력을 갖는 문서를 자동화하여 사용자 편의성과 신뢰성을 확보했습니다.

사용자 효용 및 개선 포인트

- 계약서 작성 과정을 단축함으로써 거래 효율성을 향상 시킬 수 있습니다.
- 저장 및 관리를 통해 사용자에게 편의성을 제공합니다.
- 향후 전자서명 기능을 연계해 완전한 디지털 계약 환경을 구축 할 수 있습니다.

잘한 점

- **실제 개발하고자 했던 MVP 기능 및 추가 기능 모두 구현하였습니다.**
- 단순히 각자 맡은 부분만 작업하는 것이 아니라, **피그마를 활용해 와이어프레임과 실제 디자인을 공유**하면서 각 **OS별 팀원들의 피드백을 빠르게 반영**했고, 덕분에 의도와 결과물이 일치하는 경험을 할 수 있었습니다.
- **매일 아침 Scrum 회의**를 통해 **팀원 간의 업무 진행 상황을 공유** 및 **발생한 이슈를 신속하게 해결**할 수 있었습니다.
- 개발 중간마다 UI에 대한 **팀 내·외부의 피드백**을 받아 적극 반영했고, 그 결과 프로젝트 후반으로 갈수록 화면의 완성도와 사용자 편의성이 향상되었습니다.

아쉬운 점

- 예상보다 구현 난이도가 높거나, 기능 세분화가 부족하여 **개발 일정 산정이 부정확**했고 **개발 마감 시점에 대한 부담**이 컸습니다.
- 프로젝트 초기 단계에서 **API 명세 등의 프로젝트 구조를 모호**하게 잡아 실질적인 개발이 지연되었습니다.
- **예상치 못한 응답값(예: null, 빈 배열 등)**에 대한 **방어 로직이 부족**해, 개발 중 일부 페이지에서 발생한 오류의 원인을 파악하는 데 시간이 소요되었습니다.

THANK YOU

감사합니다