

# Local Trucking Platform 개발 중간 보고서

정의왕 김수현  
경희대학교 컴퓨터공학과  
[euiwang0323@gmail.com](mailto:euiwang0323@gmail.com) [suhyunk@khu.ac.kr](mailto:suhyunk@khu.ac.kr)

## Research for Local Trucking Platform

### 요 약

본 보고서는 기초 조사서 에서 제시된 로컬 트럭킹 플랫폼 개발 프로젝트의 중간 진행 상황을 상세히 기술한다. 현재 시스템의 아키텍처 수립, 핵심 데이터베이스 모델 구축, 그리고 초기 API 구조 개발에 중점을 두어 진행했다.

### 1. 서 론

#### 1.1 프로젝트 배경 및 필요성

로컬 트럭킹 플랫폼은 미국 내 지역단위 화물 운송 시장이 직면한 고질적인 문제점을 해결하고자 기획되었다. 기초 조사서에서 언급한 대로, 미국의 화물 운송 사업은 주(state)간 횡단 서비스 부문에서 상당한 발전을 이루었으나, 도시나 군 단위를 범위로 하는 소규모 지역 운송 서비스는 상대적으로 활성화되지 못한 상태이다. 이러한 시장의 구조적 특성으로 인해 사용자들이 겪는 불편함을 해소하기 위해 프로젝트가 기획되었다.

(본 프로젝트는 Woojin Interlogis와 협업으로 진행된다.)

#### 1.2 프로젝트 목표

기초 조사서에 제시된 초기 목표는 다음과 같다.

1. 플랫폼 핵심 기능 정의 및 요구사항 도출
2. 데이터베이스 설계
3. 시스템 아키텍처 설계
4. 시나리오 설계

본 보고서에서는 현재까지 진행된 연구(개발) 및 시스템 설계 방향성을 서술한다.

### 2. 시스템 분석 및 설계

#### 2-1. 핵심 요구사항 분석 및 구현 현황

기초 조사서에서 정의된 핵심 요구사항들은 플랫폼 개발의 주요 지침이 된다. 이 요구사항들은 사용자 관리, 견적 관리 및 생성, 지역 및 위치 관리로 분류된다.

**사용자 관리**는 일반 사용자와 관리자로 구분되며, 누적 결제 금액에 따라 등급별 할인율이 차등 적용되는 기능이 포함된다. 이 요구사항을 구현하기 위해 User 및 Client 모델의 설계 및 API 개발이 완료되었다.

**견적 관리 및 생성**은 플랫폼의 핵심 기능 중 하나로 화물 정보, 픽업/배송 위치 및 날짜, 부가 서비스를 기반으로 견적이 책정되고, 최종 견적서가 이메일로 발송되는 과정을 포함한다. 또한, 배송 당일 관리자 또는 운송 담당자의 최종 승인 절차가 요구된다. 이러한 복잡한 견적 생성 로직을 지원하기 위해 Quote 및 QuoteCargo, QuoteLocation 모델이 설계되었으며, 최종 견적 제출 및 이메일 전송 기능을 포함한 API 개발이 완료되었다.

**지역 및 위치 관리**는 플랫폼의 "지역별 요금 체계"를 적용하기 위한 필수 요소이다. 군(혹은 도시)을 기준으로 하는 Area 설정과 각 Area 별 최소 주문 금액, 최대 적재량 등의 데이터 관리를 위한 기능이다.

이 부분은 RateRegion, RateArea, RateLocation 모델을 활용하며, 운영사의 원활한 마이그레이션을 위한 작업 및 API 개발이 완료되었다.

#### 2-2. 데이터베이스 설계 및 구현

주요 데이터베이스 설계는 다음과 같다.

- User와 Client는 1:1 고유 user\_id로 연결된다.
- Quote는 Client와 N:1 관계를 가진다.
- Quote는 QuoteLocation 및 QuoteCargo와 각각 1:N 관계를 가진다.
- RateRegion은 RateArea와 1:N 관계를 가지며, RateArea는 RateLocation과 1:N 관계를 가진다.

현재 개발 단계에서는 MySQL 데이터베이스를 사용하고 있다. Alembic을 활용한 DB 마이그레이션을 통해 데이터베이스 스키마의 버전 관리 및 변경 사항을 적용하고 있으며, ORM을 통해 파이썬 모델과 데이터베이스 테이블이 매핑된다.

### 2-3. 시스템 아키텍처

백엔드 아키텍처는 Python 3.12와 FastAPI 프레임워크를 기반으로 구축되었다. FastAPI는 높은 성능과 비동기 처리에 강점을 가지며, API 개발을 신속하게 진행할 수 있도록 지원한다. 데이터베이스로는 MySQL이 사용되며, DB 마이그레이션 도구로 Alembic이, 프로젝트 의존성 관리에는 Poetry를 활용했다.

확장성, 유지보수성 및 유연성을 고려하여 계층형 아키텍처(Layered Architecture)와 의존성 주입(Dependency Injection, DI) 패턴을 기반으로 설계되었다. 이러한 아키텍처 선택은 각 계층 간의 관심사를 분리하고, 객체 간의 결합도를 낮춰 시스템의 전반적인 안정성과 개발 효율성을 높이기 위해 선택되었다.

계층 구조는 API, Service, Repository, Model 계층으로 구성되어 각 계층이 명확한 역할을 수행하도록 설계했다. 예를 들어, Repository 계층은 데이터베이스와의 상호작용을 담당하고, Service 계층은 비즈니스 로직을 처리하며, API 계층은 클라이언트 요청을 받아 Service 계층으로 전달하고 응답을 반환하는 역할을 수행한다.

## 3. 개발 현황 및 구현 내용

### 3-1. 주요 기능 구현

주요 시나리오 별 개발 현황은 아래와 같다.

- 유저, 인증 시나리오
  - 회원가입 및 로그인: API 개발 및 연동 완료
  - 사용자 정보 조회: API 개발 완료
  - 사용자 주소록(생성 및 조회): API 개발 완료
- 위치 시나리오
  - 사용 가능 지역 목록 조회: API 개발 완료

- 운송 시나리오
  - 운송 형태, 화물 종류 조회: API 개발 완료
- 견적 시나리오
  - 견적 생성: API 개발 완료
  - 견적 제출 알림(이메일): API 개발 완료

## 4. 향후 계획

전체 백엔드 개발은 완료되었으며, 현재 프론트엔드 개발을 진행 중이다. 6월 8일까지 1차 개발을 완료할 예정이고, 이후 백엔드, 프론트엔드 모두 Docker를 활용한 배포 환경 구성 자동화 설정을 완료할 예정이다.

데모 운영기간에는 AWS 클라우드 서비스(EC2 등)를 이용하여 서버 및 웹 어플리케이션을 배포할 예정이다.