
컴퓨터과학부 종합설계 4조 사용자가 참여하는 **키투드 주차존**

2015920021 박인규
2017920011 김은해
2017920036 양다은
2018920059 허정우

CONTENT

01 프로젝트 개요

02 프로젝트 설계

03 진행 상황

CONTENT

01

프로젝트 개요

02

프로젝트 설계

03

진행 상황

❖ 문제 상황

- 전동 킥보드 방치 문제
 - 교통의 불편함을 초래
- 관련 법안 개정 및 방치된 킥보드 단속
 - 여전히 문제 해결 X

❖ 원인

- 전동 킥보드를 주차할 장소를 찾기 어려움
- 단체에서 지정한 위치의 경우 사용자 편의를 고려하지 않음



❖ 과제 목표

- 주차장 정보 제공
 - 어디에 전동 킥보드를 주차해야 하는가?
 - 주차장 위치를 지도에 표시하여 제공
- 주차장 정보 수집
 - 어디를 주차장으로 사용할 수 있을까?
 - 사용자로부터 주차장으로 적합한 지역을 추천받음
- 주차장 정보 공유
 - 우리가 가진 정보를 효과적으로 사용할 수 있을까?
 - API 형태로 수집한 데이터 제공

❖ 기대 효과

- 전동 킥보드 사용자
 - 사용한 전동 킥보드를 어디에 주차해야 할지 쉽게 알 수 있음
 - 가까운 전동 킥보드 주차공간의 위치를 쉽게 파악할 수 있음
- 일반 시민
 - 방치된 전동 킥보드로 인해 발생하는 교통의 불편함을 줄일 수 있음
- 지자체
 - 전동 킥보드 관리를 위한 피로 완화
(지정된 영역에 전동 킥보드가 밀집해 있으므로)
- 전동 킥보드 업체
 - 자사 플랫폼에 활용 가능



❖ 프로젝트 평가

- 정확성 (모든 기능이 결함 없이 동작하는가)
 - 모든 기능을 반복 실행하여 오류 탐지
- 어플리케이션 로딩 속도 (대기 시간이 오래 걸리는가)
 - 어플리케이션 실행 시간을 10회 반복 측정하여 얻은 최댓값과 평균값
- UI 편의성 및 정확성 (사용하기 편리한가)
 - 서비스에 대해 무지한 사용자를 대상으로 테스트 실행

❖ 프로젝트 평가

- API 서버 응답 속도 (API를 통해 정보를 빠르게 받을 수 있는가)
 - 데이터 요청 후 응답받기까지의 시간 측정
- API 서버 동시성 (요청이 급격히 늘어났을 때 대응할 수 있는가)
 - 테스트 프로그램을 별도로 작성하여 1초 내 처리할 수 있는 최대 쿼리 수 측정
- API 문서 가독성 (제공되는 API를 사용하기 편리한가)
 - 관련 지식이 있는 몇몇 사용자를 대상으로 테스트 실행

CONTENT

01

프로젝트 개요

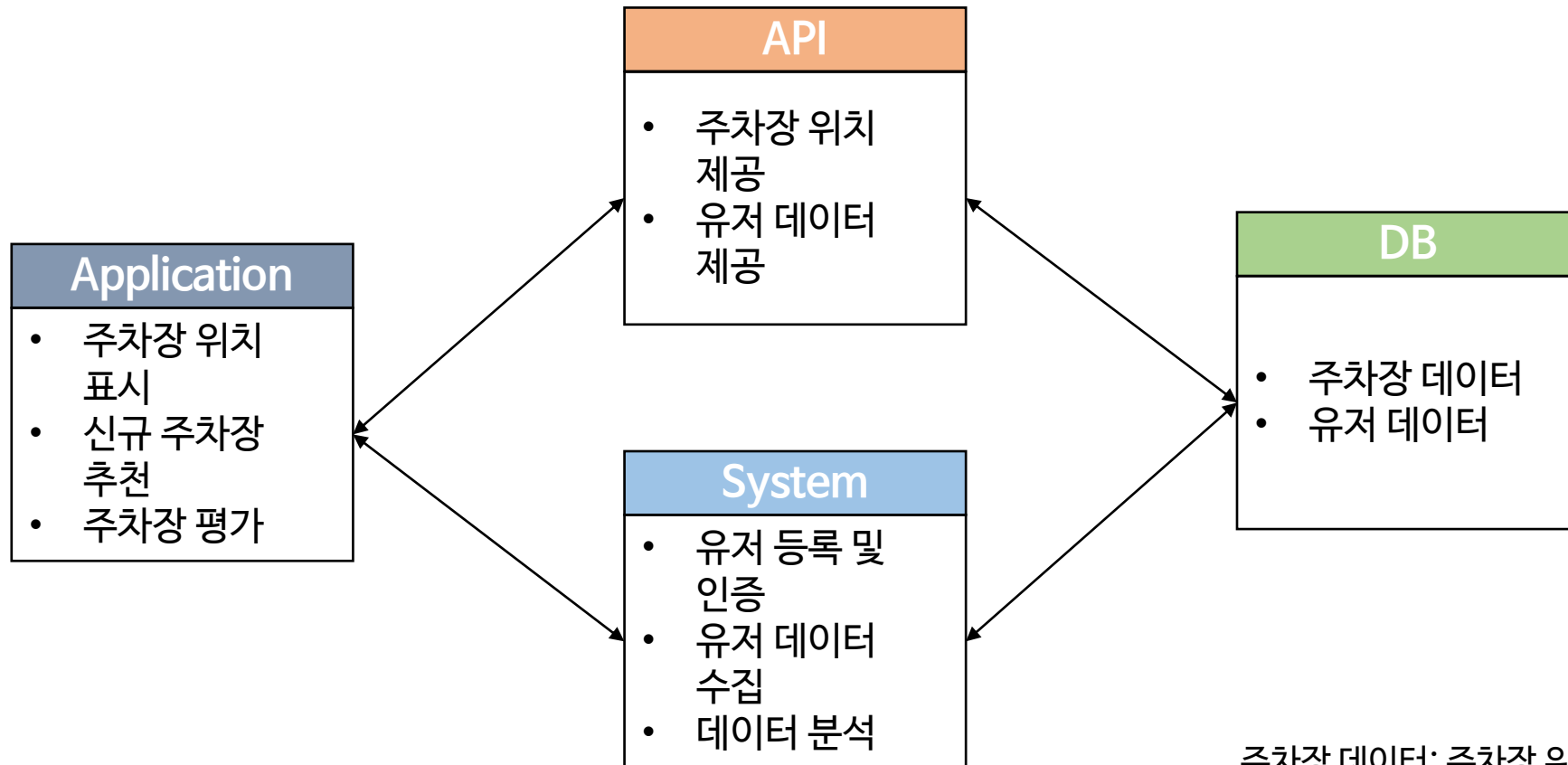
02

프로젝트 설계

03

진행 상황

❖ 프로젝트 구상도



주차장 데이터: 주차장 위치 정보
유저 데이터: 추천 장소, 별점 등 서비스 제공
과정에 유저로부터 수집한 정보

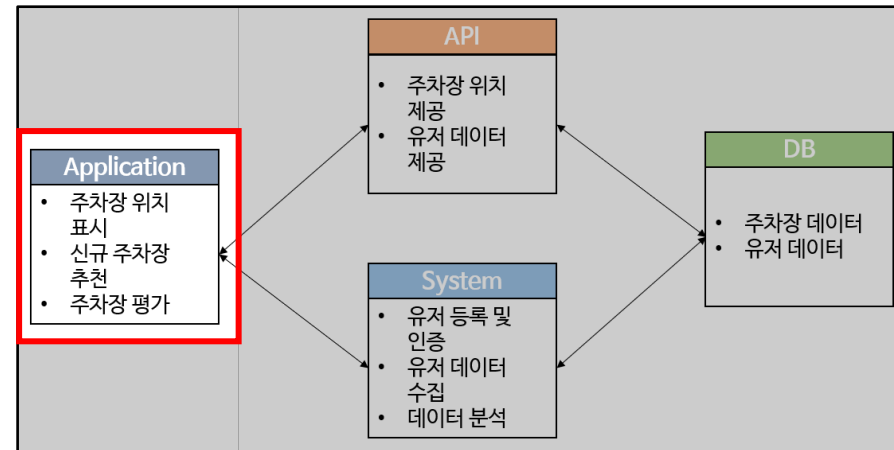
❖ 어플리케이션 (Application)

- 개요

- 스마트폰에 설치되어 서비스 제공
 - 주차장 위치 표시
 - 사용자 인증
 - 유저 데이터 수집

- 구현 포인트

- Graphic User Interface (GUI)
 - 타 시스템과 달리 GUI가 존재
 - 디자인, 사용성 모두를 고려한 GUI 설계가 핵심



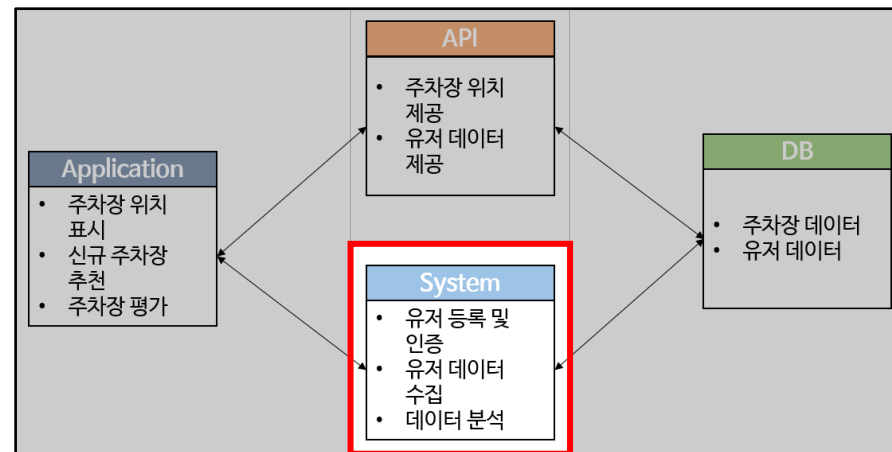
❖ 시스템 서버 (System)

• 개요

- 어플리케이션의 동작 지원
 - 사용자 인증
 - 유저 데이터 수집
 - 수집한 데이터 분석 (신규 주차장 선정 & 기존 주차장 폐쇄)

• 구현 포인트

- 동시성 & 정확성
 - 여러 클라이언트의 요청을 병렬적으로 처리
- 보안
 - 사용자의 개인 정보를 다루므로 높은 수준의 보안을 요구
- 정확한 분석 알고리즘
 - 올바른 주차장 선정을 위해 효과적인 분석 알고리즘 필요



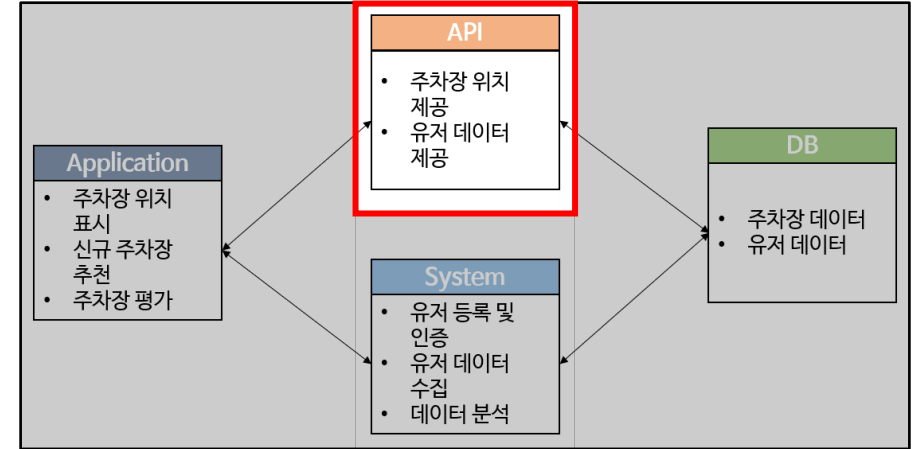
❖ API 서버 (API)

• 개요

- 서비스에 필요한 정보 제공
 - 주차장 데이터
 - 유저 데이터
- 공식 document

• 구현 포인트

- 동시성
 - 여러 클라이언트의 요청을 병렬적으로 처리
- 안정성 & 장애 대응
 - 장애 발생 시 자사 서비스 외 API를 활용하는 모든 서비스가 타격을 받으므로 높은 안정성 요구
- 사용 편의성
 - 개발자들이 쉽게 활용할 수 있도록 고려



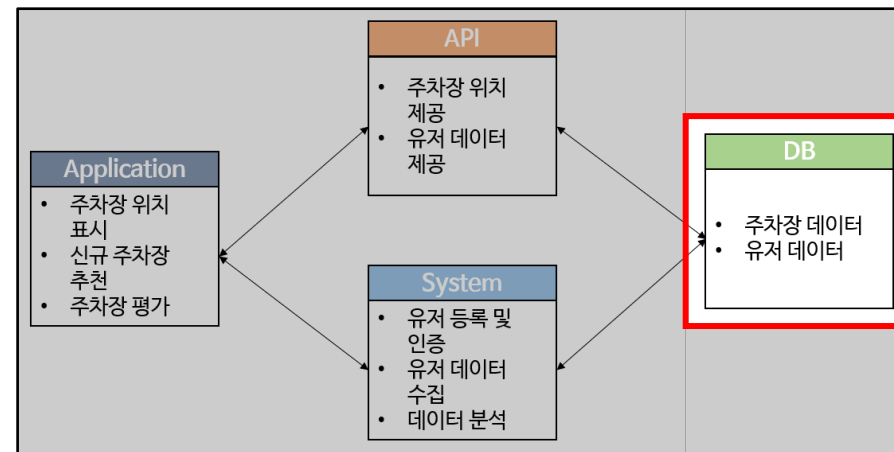
❖ 데이터베이스 서버 (DB)

• 개요

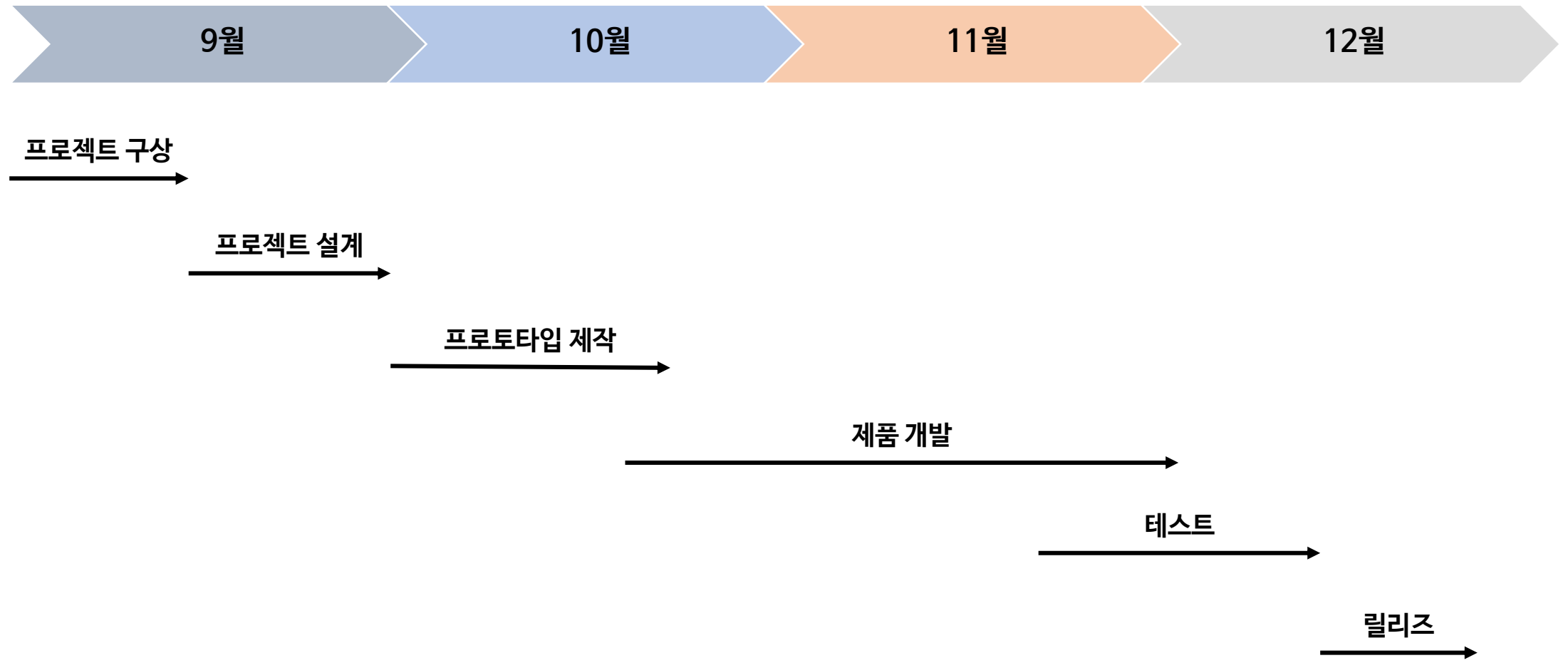
- 서비스 제공에 필요한 데이터 관리
 - 주차장 데이터
 - 유저 데이터
 - 사용자 개인 정보

• 구현 포인트

- 동시성
 - 여러 클라이언트의 요청을 병렬적으로 처리
- 안정성 & 장애 대응
 - 장애 발생 시 자사 서비스 외 API를 활용하는 모든 서비스가 타격을 받으므로 높은 안정성 요구
- 보안
 - 사용자의 개인 정보를 다루므로 높은 수준의 보안을 요구



❖ 개발 일정



❖ 업무 분담

- 인당 하나의 파트를 책임지고 구현
 - Application - 김은해
 - System - 허정우
 - API - 박인규
 - DB - 양다은
- 주간 보고서는 매주 돌아가며 작성
- 그 외 보고서는 회의를 통해 업무 분담

❖ 예산 계획

(단위: 천원)

	항목	수량	단가	계
직접개발비	클라우드 서버	3	78,000	234,000
	공인 IP	3	10,000	30,000
	스터디룸	-	-	56,000
	회의비	-	-	80,000
	합계			400,000

CONTENT

01

프로젝트 개요

02

프로젝트 설계

03

진행 상황

❖ 어플리케이션

- 프로토타입
 - Adobe XD를 이용해 UI 구상
 - node.js와 react native를 이용한 통합 개발
 - google map API 사용
 - react state 관리를 위한 hook 사용
- API 및 서버와의 통신을 위한 작업 진행중

❖ 시스템 서버

- 프로토타입
 - JAVA를 활용하여 제작
 - TCP 연결을 통해 사용자 인증을 수행하는 기능 테스트
 - 사용자 인증을 위한 토큰을 발급하고, 갱신하는 기능 테스트
 - 사용자로부터 정보를 수집하는 쿼리 테스트
- 본 서버로의 마이그레이션을 위한 작업 진행중

❖ API 서버

- 프로토타입
 - Node.js기반 express 모듈을 이용하여 REST API 구현
 - Mongoose 모듈을 이용하여 DB와 연결
 - API URI 작성 및 request, response document 작성
 - Request, response 기능 및 에러 테스트

node.js: JavaScript를 사용하는 네트워크 애플리케이션 개발 플랫폼
mongoose: mongo DB와 웹 애플리케이션 프레임 워크 간의 연결을 생성하는
JavaScript 객체 지향 프로그래밍 라이브러리

❖ 데이터베이스 서버

- 프로토타입
 - MongoDB 설치 및 사용법 익히기
 - JavaScript, npm 개발환경을 위한 Mongoose 설치
 - 데이터베이스와 서버연동 테스트
 - 데이터베이스 CRUD 설계
- 본 서버로의 마이그레이션을 위한 작업 진행중
 - 데이터베이스 구체화
 - 서버환경 구축

npm: JavaScript 프로그래밍 언어를 위한 패키지 관리자
CRUD: Create, Read, Update, Delete

❖ 장비 대여

- 시스템, API, DB 서버 제작을 위한 클라우드 컴퓨터 대여
 - 네이버 클라우드 플랫폼에서 대여
 - 총 3대의 서버를 대여, 시스템, API, DB 각 파트에서 하나씩 사용
 - 서버 접속용 개인 IP 배분 완료

❖ 클라이언트 UI 그래픽 리소스 제작

- 완성도 높은 UI/UX를 제공하기 위해 회의를 진행 중
 - 주차장을 어떻게 표시할 것인가
 - 사용자가 주차장을 추천할 때, 위치 선택을 수행하는 UI는 어떻게 만들어야 하는가
 - 로그인, 회원가입을 어떻게 하면 최대한 단순화 할 수 있을까

사용자 경험 (UX): 시스템, 제품, 서비스를 이용하면서 느끼고 생각하게 되는 총체적 경험