***1.1 프로젝트 개요***

전국적으로 10여개의 전동킥보드 공유서비스가 존재한다.

사용자들이 이 서비스들을 이용할때 각각의 앱을 실행해야하는 불편한 점이 있다.

따라서 이들의 문제를 해결하고자, 우리는 이 공유서비스들의 정보를 한 곳에 모아 보여주는 서비스를 제안하고자 한다.

각 회사에서 운영되는 킥보드의 위치정보, 상태 및 가격과 함께 예약서비스를 사용자에게 하나로 제공하며,

더 경제적인 서비스를 이용할 수 있게 추천해준다.

이로부터 사용자에게 편리성을 제공하고, 기업들에겐 타사의 고객 유치 및 데이터 확보를 통해 서비스를 개선할 수 있을 것으로 기대된다.

***1.2 프로젝트의 산출물***

|  |  |  |
| --- | --- | --- |
| 단계 | 작업 | 산출물 |
| 프로젝트 계획 | 프로젝트 범위 확정  프로젝트 일정 확정  프로젝트 진행방향 확정 | 프로젝트 계획서  WBS |
| 분석/설계 | 백로그 분석 / 정의  구성 레이아웃 설계 | SRS(요구사항 정의서)  와이어프레임 |
|
| 개발/테스트 (스프린트 5회) | 클라이언트 개발  백엔드 개발  DB 모델링  유닛테스트 | 단계별 스프린트 백로그 (제품 프로토타입) |
| 전체테스트 및 릴리즈 준비 | 통합테스트  추가 개발시 가이드 마련  운영 메뉴얼 작성 | 추가/유지 개발 가이드  운영 메뉴얼 작성 |
| 릴리즈 | 배포 준비 | 제품 백로그 |

***1.3 정의, 약어***

-참고문헌 (10)-

**백로그(Backlog)**

요구사항 리스트, 제품의 개발 대상 목록

**스프린트(Sprint)**

작업이 반복되는 단위의 기간, 본 프로젝트에서는 4주로 둔다.

**제품 백로그(Product Backlog)**

전체 기간 동안 개발할 백로그. 최종 개발 결과물이 된다.

**스프린트 백로그(Sprint Backlog)**

1개의 스프린트에서 개발할 백로그를 말한다

*2. 자원 및 일정 예측*

***2.1 자원***

가. 인력 자원

|  |  |  |
| --- | --- | --- |
| 개발 | 프로덕트 오너 / 스크럼 마스터 | 리더/개발자 |
| 팀 멤버 1 | 개발자 |
| 팀 멤버 2 | 개발자 |
| 팀 멤버 3 | 개발자 |

나. 개발 자원

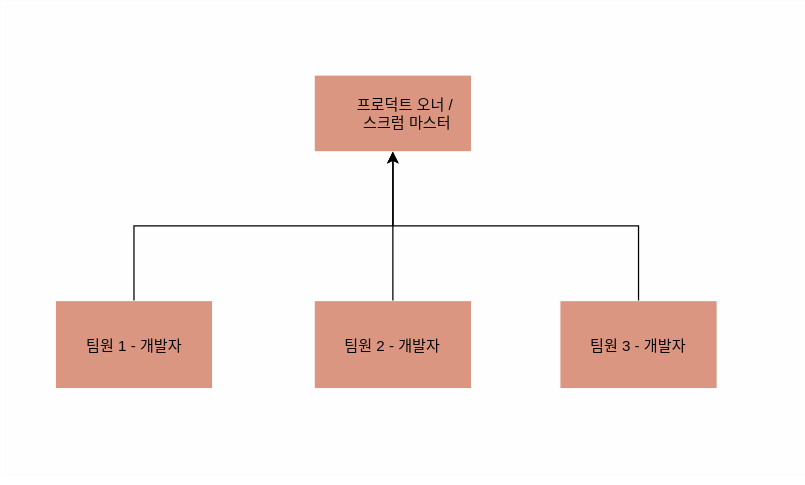
|  |  |  |
| --- | --- | --- |
| 구분 | TYPE | Spec |
| 서버 | OS | Ubuntu 18.04 |
| 클라이언트 | Flutter |
| back-end | Django 2.1  Apache2  Jenkins 2 |
| 데이터베이스 | Amazon RDS |
|  | Character Set | UTF-8 |

***2.2 일정***

|  |  |  |
| --- | --- | --- |
| 일정 | 단계 | 작업 |
| 2019/10/14 ~ 2019/10/31 | 프로젝트 계획 | 프로젝트 범위 확정  프로젝트 일정 확정  프로젝트 진행방향 확정 |
| 2019/10/26 ~  2019/11/20 | 백로그 분석/설계 | 백로그 분석 / 정의  세부 스프린트 작업 설정  구성 레이아웃 설계 |
| 2019/11/20 ~  2019/12/20 | 1차 스프린트 백로그 개발 | 클라이언트 1차 개발  백엔드 1차 개발  DB 모델링  유닛 테스트 |
| 2019/12/21 ~  2020/01/19 | 2차 스프린트 백로그 개발 | 1차 스프린트 피드백 반영  클라이언트 2차 개발  백엔드 2차 개발  추가 DB 모델링  유닛 테스트 |
| 2020/01/20 ~  2020/02/18 | 3차 스프린트 백로그 개발 | 2차 스프린트 피드백 반영  클라이언트 3차 개발  백엔드 3차 개발  추가 DB 모델링 및 최적화  유닛 테스트 |
| 2020/02/19 ~  2020/03/17 | 4차 스프린트 백로그 개발 | 3차 스프린트 피드백 반영  클라이언트 4차 개발  백엔드 4차 개발  추가 DB 모델링 및 최적화  유닛 테스트 |
| 2020/03/18 ~  2020/04/16 | 5차 스프린트 백로그 개발 | 4차 스프린트 피드백 반영  클라이언트 5차 개발  백엔드 5차 개발  추가 DB 모델링 및 최적화  유닛 테스트 |
| 2020/04/16 ~  2020/05/15 | 전체 테스트 및 릴리즈 준비 | 유닛테스트  통합테스트  추가 개발 가이드 제작  유지 가이드 제작 |
| 2020/05/16 | 배포 | 제품 백로그 배포 |

*3. 조직 구성 및 인력 배치*

***3.1 조직 구성***



***3.2 직무 기술***

|  |  |
| --- | --- |
| 프로덕트 오너, 스크럼 마스터 | 프로젝트 관리, 서버 개발 |
| 팀원1 | 서버 개발 |
| 팀원2 | 클라이언트 개발 |
| 팀원3 | 클라이언트 개발 |

*4. WBS*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 프로젝트 일정 | | 프로젝트 수행 기간 2019/10/14 ~ 2020/05/11 | | | | | | | | | | | | | | | |
| 단계 | | 10월 | | 11월 |  | 12월 |  | 1월 |  | 2월 |  | 3월 |  | 4월 |  | 5월 |  |
| 프로젝트 계획 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 백로그 분석/설계 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 개발/테스트 | 1차 스프린트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2차 스프린트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3차 스프린트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4차 스프린트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5차 스프린트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 전체 테스트 및 릴리즈 준비 | 통합테스트 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 추가 개발 가이드, 운영 메뉴얼 작성 |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 배포 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*5. 기술관리 방법*

***5.1 형상관리***

**형상관리 개요**

본 계획서에 명시 된 작업 범위의 개발에 있어서 작성되는 모든 공식 문서 및 산출물과 프로그램코드에 대한 형상관리 활동 및 절차를 포함한다.

**형상관리 절차**

* 변경사항 발생 (개발단계)
* 형상변경 요청 (Slack, 카카오톡 등 프로젝트를 위한 메신저 이용, Github내의 issue사항 요청 및 Pull)
* 회의를 통한 변경 여부 결정 (반려시 절차 종료)
* 변경
* 검토 및 테스트 (개발 부분에서 진행)
* 버전 관리

**버전 관리 표준**

검수 이전의 버전은 0.1 부터 시작, 변경시 0.1 증가

Base-line 등록시 1.0버전을 부여하며, 변경 시 0.1 증가

대규모 업데이트 진행 시 버전 1.0 증가

**형상 라이브러리**

문서 산출물 : 프로젝트를 위한 Google 공유 문서에 작성

프로그램 소스코드 : 프로젝트 버전 및 소스코드 관리의 용이성을 위해서 Github 사용

**5.2 위험관리**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 단계 | 위험 | 발생가능성 | 영향도 | 대응 방안 |
| 프로젝트  계획 | 일정 계획 오류 | 중 | 중 | WBS관리를 통하여 일정에 대한 꾸  준한 모니터링 실시 |
| 분석/설계 | 요구사항 및  분석 오류 | 중 | 중 | 현업과의 꾸준한 미팅으로 요구사항  변경관리 |
| 구현 | 인력 관리 및 일정 지연 | 중 | 하 | 팀원들과의 정보공유 및 진척사항 관리, 업무 효율성 및 부하 관리  각 개인의 피로도 파악 대처 |
| 통합테스트  /이행 | 테스트 시나리오 오류 | 하 | 중 | 테스트 시나리오 검증 및 오류 발견  시 시나리오 재작성 |
| 링크실패/성능 저하 | 중 | 중 | 현업과 단위 및 통합테스트 강화 |
| 기타 위험 요소 | 산출물 오류 | 중 | 중 | 각 단계별 산출물 검토 및 오류 발  견시 즉시 정정 |
| 비즈니스 지식의 부족 | 하 | 중 | 현업과의 상호작용을 증가시키고, 적  절한 지식전수 |
| 개발자 업무 부하로 인한 지원인력의 지원 미비 | 중 | 하 | 업무의 담당자 재조정 및 모빌리티 회사와의  협조요청을 통하여 지원 인력 확보 |

[표5-1] 위험관리 구분

**5.3 비용 및 진도 관리**

**비용관리**

프로젝트 수행을 위한 교통비 왕복비용(편도 3,000원기준)

식비 주 4일, 한달 점심가격 (6,000원)

교통비(왕복) + 식비 \* 주 4일 \* 4주 = 인당 월 192,000원 정도의 비용 발생

**진도관리**

매일 이뤄지는 데일리스크럼을 통해 확인하고, 지연이 되는 것을 확인하여 팀원들과의 일감 배분이 잘 되었는지 확인함과 매주 스크럼 매니저가 Burndown차트를 통해 매 스프린트의 진행상황을 파악한다.

*6. 표준 및 개발 절차*

**6.1 개발 방법론**

Scrum(스크럼)

**6.2 요구 목록 (Backlog)**

* 모빌리티 기기의 위치
* 기기의 현재 상태
* 사용 업체별 구역 표시
* 사용 업체별 가격 표시
* 회원 포인트 시스템
* 앱 전환 없이 바로 예약 서비스

**6.3 Sprint Backlog**

작업자 : 노아론

스프린트 기간 : 4주

**6.4 일일 스크럼 회의 (Daily Scrum Meeting)**

* 매일 10~15분 정도 짧은 회의를 진행
* 개발 진행 상황 및 변경 된 고객 요구사항 확인
* 모든 팀원 참석
* 각자 어제 한 일들과 오늘 할 일에 대하여 말하기
* 현재 나타난 버그 및 문제점들에 대한 토의
* 스프린트 회고 작성

**6.5 보고 및 검토 계획**

|  |  |  |  |
| --- | --- | --- | --- |
| 보고사항 | 주기/형식 | 주요 보고 및 검토사항 | 보고자 |
| 착수보고 | 2019.10.01 (화)  /프로젝트계획서 | * 사업의 목적, 목표, 수행 방안 및 개발계획 * 사업수행조직 및 인력투입계획 | SM |
| 주간보고 | 매주 목요일  /주간보고서 | * 주간 프로젝트 진척 현황 * 문제해결 처리 기록 * 변경 요청 처리 기록 | SM, |
| 이슈보고 | 문제,장애발생 시  /보고서 | * 장애 발생 및 기타 문제 사항 발생 보고 * 조치계획 및 조치 결과 보고 | SM |
| 중간보고 | 2019.10.17 (목)예정  /중간보고서 | * 분석/설계 진척 현황 중간 점검 * 본 사업 진행 이슈 파악 및 처리 보고 * 향후 사업 진행 내용 및 일정 보고 | SM |
| 완료보고 | 2019.12.9(월)예정  /완료보고서 | * 본 사업 요구사항 요약 * 본 사업 진행경과 / 실행 사항 * 단계별 계획 대비 실적 분석 | SM |
| 회의록 | 회의록 | - 회의 중 협의된 사항에 대한 확인 문서 | 회의 진행자 |

[표6-1] 보고 방안

*7. 개발환경*

**개발 OS**

Ubuntu 18.04

**클라이언트**

Flutter

**Back-end**

Django 2.1, Apache2, Jenkins2

**데이터베이스**

Amazon RDS

**Character Set**

UTF-8

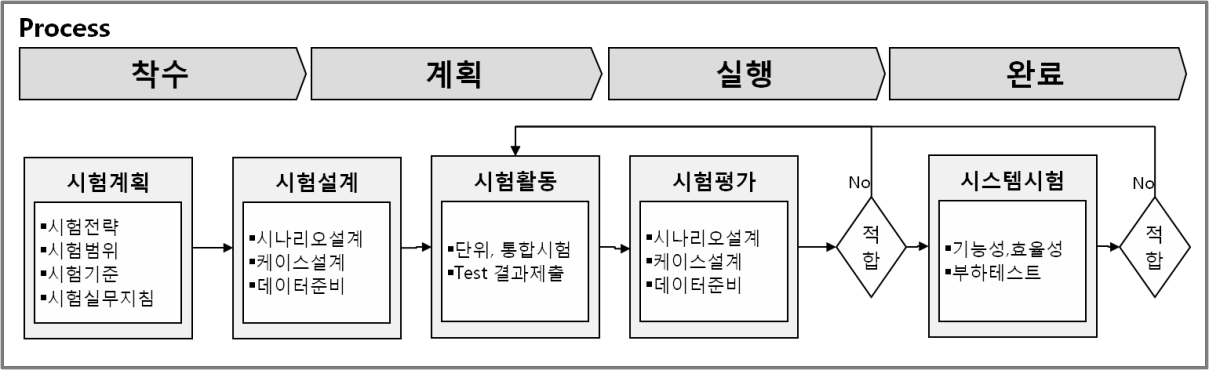
*8. 성능 시험 방법*

**8.1 테스트 계획**

방법 가. 시험항목

1. 단위테스트 : 단위프로그램의 기능 작동 시험
2. 통합테스트 : 시스템간 연동 및 전체 기능 시험
3. 성능테스트 : 사용자 폭주에 따른 과부하 시험

방법 나. 시험전략

1. 사용자의 요구사항을 효율적으로 검증할 수 있는 공동 시범운영조직 구성
2. 실 운영 환경에서 시스템 성능 검증
3. 실제와 동일한 조건의 운영환경 구축 및 시범운영 시나리오 적용
4. 오류 검증에 의한 품질 개선

[그림 8-1] 테스트 절차

*9. 문서화*

**문서화**

회의록 및 프로젝트 진행 상황, 메뉴얼 등 Google의 공유문서 기능을 이용하여 작성한다.

*10. 참고문헌 및 부록*

(10) - 애자일 스크럼 프로세스에 대해, 용어

<http://www.agileforall.com/intro-to-agile/>

스크럼 출처 : <https://zeddios.tistory.com/24>

AXA 다이렉트 프로젝트 계획서

****