소프트웨어공학

프로젝트 Proposal

셰어택시-택시 동승 어플리케이션

NULL 사로잡겠어 리턴즈

팀장 컴퓨터학부 이민석 2016117244

팀원 컴퓨터학부 곽준창 2012105001

컴퓨터학부 우정우 2016111829

컴퓨터학부 차수진 2016111813

일어일문학과 황원규 2011009034

|  |
| --- |
| **Introduction** |
| * Objective: 택시 동승자를 쉽게 찾기 위한 소프트웨어 * Constraints   + Schedule: 4/4 프로젝트 시작, 6/17 프로젝트 종료   + Budget: 0 (AWS-12 months free)   + Resources:     - HW: PC 및 Laptop 5대, Android 2~3대     - SW: AWS, Android Studio, Git, MS Office, Atom, MySQL     - Human Resource: Android app 개발 경력자 3명(서버 1명, 클라이언트 2명), 디자이너 최소 1명, 기획자 최소 1명 * Motivation: 버스를 타기 불편한 출퇴근이나 등, 하교 같은 시간대에 사람들은 택시를 많이 이용한다. 하지만 부족한 택시 수와 금전적, 시간적인 문제로 인해 불편함을 호소하는 사람들이 많다. 이를 해결하고자 이 프로젝트를 계획하게 되었다. |
| **Project Function** |
| * 탑승 정보 설정   + 지도 API를 이용해 출발지/도착지 설정.   + 탑승 시간을 필수 입력사항으로 함. * 동승자 검색   + 출발지 반경 300m 이내, 도착지 반경 300m 이내 검색   + 검색 완료 시 조건에 맞는 방 목록을 보여줌. * 방 만들기   + 설정한 조건에 기반해 방을 만듦. * 방 입장   + 입장 후 동승자 간 그룹 채팅 형식으로 시작   + 모든 사용자는 닉네임으로 보여짐   + 타 사용자 인기도를 볼 수 있음   + 방 나가기, 출발하기(준비) 선택 가능   + 추가사항: 모임 장소와 동승자 간 위치 표시 * 상호 평가 * 출발하기 후 상호평가 화면으로 이동함 * 동승자 간 평가(좋아요, 싫어요, 아무것도 아니함) * 평가 후 처음 화면으로 돌아감 * 회원 가입   + ID, PW를 입력하고, email 인증을 진행한다. * 로그인/로그아웃 |

|  |
| --- |
| **Project Organization** |
| * 기획   + 곽준창, 우정우, 이민석, 차수진, 황원규 * 서버&DB   + 주: 곽준창   + 보조: 우정우, 차수진 * 클라이언트 (Android)   + 주: 곽준창, 황원규   + 보조: 우정우, 이민석, 차수진 * Documentation   + 주: 이민석   + 보조: 곽준창, 우정우, 차수진, 황원규 |
| **Project Plan & Schedule** |
|  |
| **Risk analysis** |
| |  |  |  |  | | --- | --- | --- | --- | | **Risk** | **Probability** | **Effect** | **Management Strategy** | | 인력 추가 불가 | High | Tolerable |  | | 팀원의 drop | Moderate | Catastrophic | 역할 배분을 겹치게 함(위험 최소화) | | 프로젝트 경제적 문제 | Low | Insignificant |  | | 일부 기능이 구현이 불가능할 경우 | Moderate | Tolerable |  | | 소프트웨어 개발 시간 계획이 과소평가됨 | High | Serious | 반복적으로 비기능, 기능적인 요구사항을 자세하게 파악해 수행 시간을 정확하게 예측(위험 최소화) | | 요구사항 변경 | High | Tolerable |  | | 제품 경쟁 | Low | Tolerable |  | | 관리 도구 미숙 | High | Tolerable |  | | 주요 인력이 사고를 당함 | Low | Serious | 역할 배분을 겹치게 함(위험 최소화) | | 프로젝트 설계 변경 사항 발생 | High | Serious | 설계 변경을 대처할 수 있는 Project Process 방법론 적용(위험 최소화) | | 개발 H/W 분실 | Low | Serious | Cloud 및 Github 등을 사용한 주기적인 파일 백업 및 버전 관리(위험 최소화) | |

