

프로젝트명	택시 쉐어링 어플(Plan'T)				
개발 인원	3명	본인 역할	팀장	개발 기간	2016.2 ~ 8
개발 언어	Java, PHP, SQL				
개발툴	Android Studio, MySQL				
작품 소개(요약)	<p>출신 학교인 인하대학교에서 가장 가까운 지하철역은 주안역이다. 그러나 학교에서 주안역까지의 거리가 4km정도이다. 집에서 통학을 하는 학생들은 대부분 학교와 주안역 사이를 왕복하는 511번 버스를 이용한다. 그러나 집에 급한 볼일이 있거나 수업시간에 늦었을 때와 같이 급히 학교나 지하철역까지 가야하는 경우에는 택시를 이용한다. 또한, 많은 학생들이 비슷한 시간대에 등교하고 하교하기 때문에 버스줄도 상당히 길다. 그래서 이럴 때에 택시와 버스의 도착시간 차이는 많이 날 수 밖에 없다. 이러한 상황에서 동일한 목적지로 향하는 학생들을 묶어주는 실시간 매칭을 통해서 택시 쉐어링을 보다 간편하고 신속하게 해주는 것이 이 어플의 첫 번째 기능이다.</p> <p>두 번째 기능은 예약기능이다. 대학생들이 자주 보는 토익이나 토플, 각종 자격증 시험들은 대학교 근처 낯선 고등학교, 중학교에서 이루어진다. 학생들은 주로 길을 잘 모르거나 아침 일찍 시험이 치뤄져서 버스를 이용하기 보다는 대부분 택시를 이용하여 시험장으로 간다. 이러한 학생들을 묶어주어서 택시 쉐어링을 보다 효율적이고 편리하게 할 수 있게 해주는 것이 두 번째 기능이다.</p>				
설계 과정	<p>처음으로 진행 했던 것은 사전 조사였다. 작은 아이디어에서 출발하였기 때문에 이미 개발된 어플리케이션이 있을 거라고 판단하였다. 조사 결과 카이스트 학생들이 만든 “택시투게더”라는 어플이 있었다. 활성화 되어있지는 않아서 실패 원인을 분석한 결과, 카이스트 메일을 통한 인증으로 인한 확장성 제한, 실시간 매칭의 부재, 인하대와 같은 공통된 목적지의 부재, 사용자의 평가 부재가 있다고 판단하였다. 이를 해결하기 위해서 카카오톡 등의 각종 SNS 인증방식을 이용하기로 하였다. 또한, 실시간 매칭의 중요성에 대해서 판단 할 수 있었다. 그리고 사용자에 대한 평가를 도입해서 많은 부적절한 행동들의 규제를 도입하기로 하였다.</p> <p>이후에는 개념설계를 진행하였다. 이 과정에서 Use-Case-Diagram과 시스템 흐름도를 작성하여 도표화 하였다. (참고 자료)</p> <p>그리고 클래스 다이어그램을 작성하였다. 클래스 다이어그램은 각종 Activity와 그에 상응하는 Layout을 기준으로 하였으며, 이에 따른 필수적이고 기능적인 객체들을 작성하였고 이에 대한 네이밍 규칙도 선정하였다. 또한 객체들의 흐름을 중점적으로 작성하고자 하였다. (참고 자료)</p>				
개발 내용	<p>로그인 과정은 우선, 사용자가 이전에 로그인 했던 기록이 존재한다면 Shared Preferences를 통해서 저장해 두었다가 즉시 로그인을 가능하게 하였고, 그렇지 않다면 로그인 방식을 카카오톡이나 페이스북 중 하나를 선택해서 진행할 수 있게 하였다. 그 방식은 각 SNS의 API를 통해서 구현하였다. 또한 회원 정보는 DB에 저장해두고 MySQL을 통해서 관리하였다.</p> <p>로그인 이후에는 사용자가 선택 가능한 메뉴는 세 가지가 존재한다. 그 메뉴들은 실시간 매칭, 예약하기, 방 목록이 있다. 순서대로 실시간 매칭의 경우에는 위에서 언급한대로 인하대 후문이나 주안역을 고정된 출발지로 하여 각각 주안역과 인하대를 도착지로 하는 인원들에 대한 실시간 매칭을 해준다. 실시간 매칭에 필요한 옵션들을 선택하고 나면 사용자의 입장에서는 대기상태가 되고 이후 서버로 이 옵션들에 대해서 전송해준다. 서버에서는 이러한 입력들을 받아서 동일한 옵션을 가지고 있는 인원들과 묶어주고 완료를 하고나면 사용자는 AJAX를 통해서 계속 갱신을</p>				

	<p>하고 있다가 완료 상태를 확인한다.</p> <p>그리고 나서 다시 사용자에게 방에 대한 정보를 주게 되고 이를 받아서 사용자는 방 정보 인스턴스를 생성하여 관리하게 된다. 두 번째 메뉴인 예약하기의 경우는 예약하는 방을 생성하거나 직접 참여 할 수 있다. 두 가지 선택 모두 PHP를 통해서 데이터베이스에 있는 정보를 가지고 와서 직접 수정하는 형태로 이루어진다.</p> <p>마지막으로 방 목록 메뉴의 경우에는 해당 메뉴를 클릭하게 되면 사용자가 참여중인 방 목록을 받아오고 이 목록을 저장해둔다. 그리고서 방을 참여하거나 방을 생성하는 경우에는 저장해두었던 방 정보에 추가해준다. 그러나 방 정보에는 참여중인 인원이 포함되어 있어서 이를 갱신해주기 위해서 다시 방 목록 메뉴를 선택하게 되면 저장해두었던 정보에 대해서 비교하여 갱신해준다.</p> <p>방 목록 메뉴는 ListView 형태로 이루어져 있는데 해당 아이템을 선택하게 되면 방에 대한 구체적인 정보가 나오고 방에 할당된 채팅방으로 들어갈 수 있다. 채팅방은 서버의 문제로 연결지향적으로 구현되지 못했다. 그 이유는 서버를 구하는데 있어서 비용의 문제로 “가비아”라는 사이트에서 웹호스팅을 통해서 구축을 하였다. 웹호스팅을 통하다보니 포트를 할당받지 못하고 이 때문에 기존에 계획했던 소켓을 이용한 연결기반 방식이 불가능했다. 어쩔 수 없이 HttpRequest 객체를 이용하여 연결하였다. 또한 채팅을 입력하게 되면 이를 서버로 보내면 이를 데이터베이스에 보내고 안드로이드 GMS(Google Mobile Services)를 통해서 푸시알람을 보내도록 구현하였다. 여기서 사용자는 여러 개의 안드로이드 기기를 소유할 수 있기 때문에 데이터베이스에도 따로 테이블을 만들어 관리하였다. 채팅을 보는 입장에서는 연결기반 방식이 불가능하므로 계속 데이터 AJAX를 통해서 갱신함으로써 채팅방을 구현하였다.</p>
<p><b>프로젝트 문제점/해결방안</b></p>	<p>가장 큰 문제는 설계단계에서의 부족이었다. 우선, 클래스 설계가 빈약하였다. 프로젝트를 시작할 때에 개발 인원 모두가 설계에 대한 지식이 부족했다. 이 때문에 클래스 설계가 제대로 이루어지지 않았다. 구체적으로는 클래스의 추상화가 전혀 이루어지지 않아서 클래스간의 통일성이 부족했고, 여러 번 반복되는 인터넷 연결 과정을 하나의 클래스로 구현하지 못해서 코드가 난잡해졌다. 또한 다이얼로그와 같이 하나의 인스턴스만 사용하는 클래스에 대해서 이를 보장하지 못했다.</p> <p>프로젝트의 규모가 조금만 늘어나도 겪게 되는 실수들을 이번 기회에 경험하게 되었다. 그리고 이러한 설계의 중요성을 느끼고 이에 대해서 관심을 가지게 되었다. 그 결과 이러한 시행착오에서 비롯된 패턴들을 정리한 디자인 패턴에 관한 책을 공부하게 되었다.</p> <p>다른 문제점은 단체 프로젝트에서의 개발 영역에 관한 문제이다. 3명에서 프로젝트를 진행하는데 있어서 정확한 개발 영역을 정해놓지 않고 구현을 시작하다 보니 Activity간의 커뮤니케이션에서부터 사용자 데이터의 형태까지 통일되지 않아서 이 때문에 많은 시간과 노력이 들었다. 이 문제는 설계 문제의 연장선이기도 했다. 구체적으로 정해진 사항들이 얼마 없다보니 이에 대해서 개발 영역을 나누기도 힘들었고 또한 자료형태의 통일도 되지 않았다.</p> <p>마지막 문제점은 서버개발에 대한 무지였다. 프로젝트를 진행하기 앞서 준비기간을 두었는데 그 동안에 안드로이드에만 초점을 두고 공부하였다. 이 때문에 서버개발 시에 웹호스팅을 이용하였기 때문에 연결기반 방식의 불가능부터 여러 가지 문제에 당면하게 되었다. 애초에 서버개발을 염두해두지 않고 설계를 진행하였고 서버개발이라는 영역을 처음 발견하게 되었다.</p>
<p><b>기타</b></p>	<p>인하종합설계 경진대회 은상, 종합설계경진대회 동상</p>