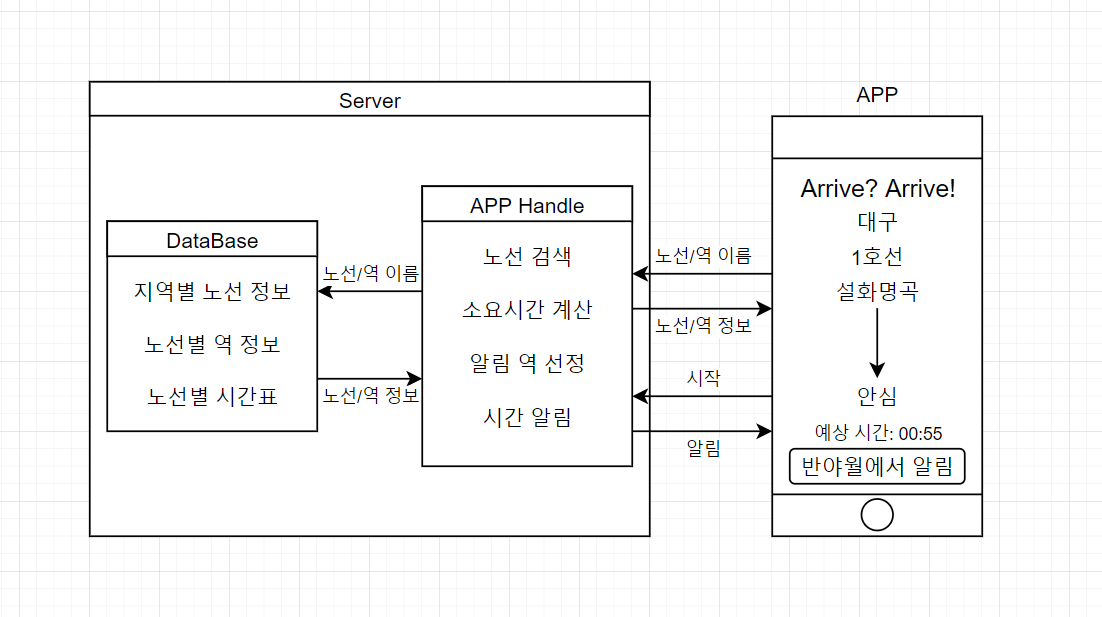
|  |
| --- |
| **모바일 앱 프로그래밍2 프로젝트 제안서** |

|  |  |  |  |
| --- | --- | --- | --- |
| **과 제 명** | arrive? - alive! | | |
| **참여학과** | IT대학 컴퓨터학부 | | |
| **팀원** | 김형진 박주홍 박효상 송영욱 | | |
| **책임교수** | 정기숙 | **과제대표** | 박효상 |
| **추진배경** |  | | |
| ◦ 대중 교통의 이용이 권장되고, 대중 교통의 사용량이 증가함에 따라 이용자들의 편의를 개선할 방법에 대해서 생각해보게 되었다. 출퇴근 시간과 다른 목적의 이동 시간을 활용하여, 대중 교통을 사용하는 중에 다른 일에 집중할 수 있도록 목적지 도착 전에 알림을 주는 어플리케이션을 개발하고자 한다. | | | |
| **개발 기능 및 내용** |  | | |
| ◦ 출발 지점과 도착 지점을 설정하여, 도착 전에 알림을 주는 기능 개발  - 출발 지점과 도착 지점은 지하철 역 이름을 기반으로 한다.  - 도착 전 알림을 주는 기준은 도착역으로부터 2정거장 전역에 도착했을 때로 한다.  ◦ 프론트엔드  - 초기화면 구성  - 운행 중 화면 구성 및 표현  - 도착역으로부터 2정거장 전역에 도착하였을 때의 화면 구성 및 알람 컨트롤  ◦ 백엔드  - 데이터베이스 구성  - 서버 구성  - 프론트엔드와 앱핸들러, 데이터베이스 간 통신 | | | |
| **기대효과** |  | | |
| ◦ 출퇴근 시간 및 기타 이동 시간을 효율적으로 활용할 수 있다.  ◦ 도착지에서 못 내리는 등의 개인 실수로 인한 불필요한 시간 소모를 방지할 수 있다.  ◦ 기존 지도 어플리케이션에 활용 가능한 기능이다. ex) Google map, Naver map | | | |
|  | | | |

**# 프로그램 개발 방법 및 프로토타입**

**1) 백엔드 및 서버, 데이터베이스**



* 초기상황 : 어플리케이션 내부에는 호선에 대한 정보들이 저장되어 있다.
* 어플리케이션은 노선에 해당되는 역들에 관한 request 를 서버로 보낸다.
* 서버는 데이터베이스에서 해당 노선에 속하는 역들에 대한 정보를 탐색하여 어플리케이션에게 response 한다.
* 어플리케이션이 시작 request 를 서버로 보낸다.
* APP Handle 은 데이터베이스에게 노선 번호, 출발역과 현재시간, 도착역에 대한 정보를 전송함과 동시에 출발시간과 알림역에 대한 정보를 request 한다.
* 데이터베이스는 노선별 역 정보에서 도착역으로부터 2정거장 전에 있는 알림역을 선정하고, 해당 노선별 시간표에서 출발역에서의 예상 출발시간, 알림역 도착시간에 관한 정보를 APP Handle 로 response 한다. 이 때 출발시간은 현재 시간으로부터 제일 가까운 시간을 기준으로 설정한다.
* APP Handle 은 해당 정보로 소요시간을 계산한 뒤에 어플리케이션으로 response 한다.

**2) 프론트엔드**

|  |  |  |
| --- | --- | --- |
| **Arrive? Alive!**  in Daegu  지하철 노선  [ “선택하세요” (목록) ]  출발지  [ “선택하세요” (목록) ]  도착지  [ “선택하세요” (목록) ]  [ “맡겨주세요!” (버튼) ] | **Arrive? Alive!**  in Daegu  [가는중 (라벨)]  [XX분 XX초]  도착하기 전  2정거장에  알려드릴게요 | **Arrive? Alive!**  in Daegu  [ 일어나세요! (라벨)]  목적지에  도착하기까지  2정거장 남았어요  [ 알림 끄기 (버튼) ] |
| 초기화면 | 가는 중 | 도착 2정거장 전부터 |

* 제목은 라벨로 구성되며, 크기와 폰트, 내용은 변경될 수 있다.
* 초기화면은 지하철 노선, 출발지, 도착지를 선택할 수 있는 목록으로 구성되어 있다. 각 목록은 “선택하세요” 로 초기화 되어 있고 지하철 노선을 선택하고 나서야 출발지, 도착지 목록을 출력할 수 있게 한다.

[예외1] 노선을 선택하지 않고 출발지나 도착지를 설정 시도하거나, 검색하기 버튼을 누르게 되면 경고 메시지 ( “노선을 먼저 선택해주세요!” ) 를 띄운다.

[예외2] 출발지와 도착지 간의 간격이 2개 정거장 이하일 경우 불가 메시지 ( “주무시면 안되는 거리입니다!” ) 를 띄운다.

[정상] 정상적으로 노선과 출발지, 도착지를 설정하고 나면 “맡겨주세요!“ 버튼을 눌러서 동작을 실시한다.

* 가는 중 화면은 라벨들로만 이루어져 있으며 출력 외의 다른 기능은 없다. 단, 서버로부터 받은 예상 소요 시간을 초단위로 줄어들게 해서 이를 어플리케이션 내에 표시한다.
* 도착 2정거장 전부터 화면은 라벨과 버튼으로 이루어져 있으며, 이때 알람을 울린다. 즉 예상 소요 시간이 0초가 되었을 때 알림이 울린다. 알람 벨소리 및 진동 설정은 추후에 설정 가능하도록 하고 현재는 고정 알람을 사용한다. 확인 버튼을 누르게 되면 알람이 꺼지고, 초기 화면으로 돌아간다.

**# 과제 추진 일정**

|  |  |
| --- | --- |
| 1주차 | 프로젝트 주제 브레인 스토밍, 제안서 작성 |
| 2주차 | 프로그램 개발 |
| 3주차 | 중간 발표 |
| 4주차 | 프로그램 개발 |
| 5주차 | 최종 발표 |

**깃허브 주소 : https://github.com/Arrive-Alive/Arrive-Alive-SWIFT**

**# 추가 활용 방안**

해당 어플리케이션이 개발이 된다면, 지하철을 타는 도중에 도착지에 제대로 하차하지 못하는 경우를 줄일 수 있다.

추가적으로, 향후 해당 어플리케이션이 업데이트 과정을 거쳐 지하철 뿐만 아니라 시내버스, 고속버스, 시외버스와 KTX, SRT, 무궁화호, 새마을호 같은 기차에 대한 정보를 활용한다면 사용 범위가 더욱 확장될 것이다. 현재는 대구에 한정되어 있으나 대구 외의 다른 지역들에 대한 정보를 다루게 된다면 더욱 더 활용성이 높아질 것이 자명하다.

기존에 자주 사용되는 지도 어플리케이션에 우리의 기술을 추가하게 된다면 사용자들의 만족도 역시 더 높아질 것이다.

서울의 경우 실시간 지하철 도착 정보가 공공데이터로 제공되어 있어서 API 를 활용하기 좋지만 그 외의 지역은 어렵기 때문에 시간 계산을 통해 도착을 예상하였다. 지하철의 경우는 이와 같이 계산이 가능하며, 시내 버스의 경우는 전국 모든 시내버스의 위치를 제공하는 API 를 확인했기에 이를 활용하면 추가적으로 기능을 확장시킬 수 있다.