**Software Requirements Specification**

**Version 2.0**

|  |  |
| --- | --- |
| **May 11, 2024** | **V 1.0** |
| **May 18, 2024** | **V 1.1** |
| **October 6, 2024** | **V 2.0** |

**October 6, 2024**

**Omnibus;**

**Voice-Based Bus Route Guidance System**

**강은빈, 남현서, 양수연**

**임수빈, 임하나**

**Submitted in partial fulfillment**

**Of the requirements of**

**Capstone Design 1**

### **Preface**

이 문서는 2024년 1학기 SW보안융합학과 캡스톤디자인(1) 수업의 산출물 중 하나입니다.

프로젝트에 참여한 학생은 배화여자대학교 SW보안융합학과 202254013 강은빈, 202154026 남현서, 202154027 양수연, 202254031 임수빈, 202254012 임하나이며, 김병국 교수님의 지도를 받아 진행되었습니다.

# **Table of Contents**

[**Preface** i](#_Toc179124696)

[**Table of Contents** ii](#_Toc179124697)

[**List Of Figures** iv](#_Toc179124698)

[**1.**  **Introduction** １](#_Toc179124699)

[**1.1** **Purpose** １](#_Toc179124700)

[**1.2**  **Scope** １](#_Toc179124701)

[**1.3**  **Terms and Abbreviations** ２](#_Toc179124702)

[**2. Overall Description** ３](#_Toc179124703)

[**2.1** **Context Diagram** ３](#_Toc179124704)

[**2.2** **System Environment** ３](#_Toc179124705)

[**2.3**  **Product Functions** ５](#_Toc179124706)

[**3.**  **Specific Requirements** ６](#_Toc179124707)

[**3.1** **User Requirements** ６](#_Toc179124708)

[***3.1.1*** ***Functional Requirements*** ６](#_Toc179124709)

[***3.1.2*** ***Non-functional Requirements*** ７](#_Toc179124710)

[**3.2** **System Requirements** ８](#_Toc179124711)

[***3.2.1*** ***User Interfaces*** ８](#_Toc179124712)

[***3.2.2*** ***User Interfaces Example*** ９](#_Toc179124713)

[***3.2.3*** ***Hardware Design and External Status*** ９](#_Toc179124714)

[**3.2.3.1** **User Input Method** １０](#_Toc179124715)

[**3.2.3.2** **LCD display board** １０](#_Toc179124716)

[**3.2.3.4** **Responsiveness and errors** １０](#_Toc179124717)

[**3.2.3.4** **Hardware Interface** １１](#_Toc179124718)

[***3.2.4*** ***Software Interface*** １１](#_Toc179124719)

[**3.2.4.1** **STT** １１](#_Toc179124720)

[**3.2.4.2**  **NLP** １２](#_Toc179124721)

[**3.2.4.3** **Bus Route Searching** １２](#_Toc179124722)

[**3.2.4.4** **TTS** １２](#_Toc179124723)

[***3.2.5*** ***Cost Requirements*** １２](#_Toc179124724)

[**3.3** **Other Requirements** １３](#_Toc179124725)

[***3.3.1*** ***Organizational Requirements*** １３](#_Toc179124726)

[***3.3.2*** ***External Requirements*** １４](#_Toc179124727)

[**3.4**  **Restrictions** １４](#_Toc179124728)

[**3.5**  **Data Flowchart** １５](#_Toc179124729)

[***3.5.1*** ***음성 입력 모듈*** １５](#_Toc179124730)

[***3.5.2*** ***자연어 처리 모듈*** １７](#_Toc179124731)

[***3.5.3*** ***버스 노선 검색 모듈*** １８](#_Toc179124732)

[***3.5.4*** ***음성 출력 모듈 (스피커)*** ２０](#_Toc179124733)

[***3.5.5*** ***표시 출력 모듈 (LCD 모니터)*** ２１](#_Toc179124734)

[**4.**  **Appendices** ２２](#_Toc179124735)

[**A.** **Dictionary** ２２](#_Toc179124736)

[**B.**  **References** ２３](#_Toc179124737)

# **List Of Figures**

[Figure 1 – Context Diagram ３](#_Toc179124738)

[Figure 2 - System Environment ４](#_Toc179124739)

[Figure 3 - User Interfaces Example ９](#_Toc179124740)

[Figure 4 - 음성 입력 모듈 １５](#_Toc179124741)

[Figure 5 - 자연어 처리 모듈 １７](#_Toc179124742)

[Figure 6 - 버스 노선 검색 모듈 １８](#_Toc179124743)

[Figure 7 - 음성 출력 모듈 (스피커) ２０](#_Toc179124744)

[Figure 8 - 표시 출력 모듈 (LCD 모니터) ２１](#_Toc179124745)

# **1. Introduction**

이 프로젝트는 대중교통 정보 접근에 불편함을 겪는 정보 취약 계층을 대상으로 정보 접근성을 향상하기 위한 음성 기반의 버스 노선 안내 시스템을 개발하는 것이다. 대중교통 정보를 얻을 수 있는 지도 애플리케이션이나 스마트폰 장치 자체를 능숙하게 사용할 수 없는 다양한 사회 계층의 정보 약자가 존재하고, 이로 인해 정보 격차가 발생한다. 이러한 문제점을 분석하여 음성 인식 기술을 기반으로 한 대중교통 정보를 쉽게 얻을 수 있는 시스템을 구축하였다. 사용자가 원하는 목적지를 자연어 음성으로 입력 받아 목적지 정보를 파악하고 해당 장소로 향하는 버스 노선을 탐색해 다시 음성으로 안내하므로 정보 접근성을 높일 수 있고 공공 정보를 효율적으로 제공할 수 있을 것으로 기대된다.

## **1.1 Purpose**

본 요구사항 명세서는 ‘정보취약계층의 대중교통정보 접근성 향상을 위한 음성기반 버스노선 안내 시스템’ 개발 프로젝트에 대한 요구사항을 정의하는데 목적이 있다. 이 문서의 작성을 통해 시스템의 목표, 대상 사용자, 주요 기능, 비기능 요구사항, 개발 환경 등을 명시함으로써 개발 범위와 방향성을 결정하고 체계적이고 일관된 시스템 개발을 가능하게 한다. 아울러 향후 설계, 구현, 테스트 등 전반적인 개발 프로세스에서 지침서 역할을 할 수 있을 것이다.

## **1.2 Scope**

본 요구사항 명세서는 프로젝트의 배경과 필요성, 대상 사용자 및 사용 환경, 시스템의 주요 기능적 요구사항과 비기능적 요구사항, 개발 환경 등을 포함한다. 아울러 시스템의 전체적인 프로세스 이해를 돕기 위해 자료 흐름도와 사용 시나리오도 포함하고 있다.

한편 이 문서는 상세한 시스템 기능 규격이나 설계 구조와 같은 사항은 다루지 않는다. 시스템 아키텍처, 데이터 모델, 설계도, 알고리즘 등 구체적인 설계 · 구현 내용은 별도의 문서로 분리하여 작성할 예정이다. 본 명세서의 목적은 개발 범위와 방향성을 명확하게 하고, 시스템 전반에 걸친 요구사항을 정의하는 데 있다. 또, 소프트웨어 위주의 요구사항 명세에 집중한다. 하드웨어 부품의 규격이나 설계도 등의 내용은 다루고 있지 않다.

## **1.3 Terms and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Omnibus | 프로젝트의 결과물로 음성 기반 버스 노선 안내 시스템을 총칭 |
| STT (Speech To Text) | 음성 신호(Speech)를 해석하여 그 내용을 문자 데이터(Text)로 전환하는 처리 |
| TTS (Text To Speech) | 문자 데이터(Text)를 음성 신호(Speech)로 전환하여 목소리를 합성하는 처리 |

# **2. Overall Description**

## **2.1 Context Diagram**

시스템 외부의 사용자가 목적지로 향하는 버스 노선 정보를 알고자 하면 목적지 정보를 자연어 음성을 통해 요청한다. 내부 버스 노선 검색 시스템은 작업 수행 후 검색 결과를 다시 음성으로 출력하여 사용자는 필요한 정보를 얻는다.

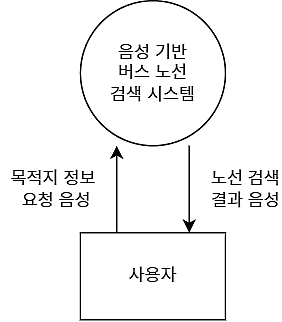


Figure 1 – Context Diagram

## **2.2 System Environment**

Omnibus의 시스템 환경은 Figure 2와 같다. 시스템은 직접적으로 사용자와 상호작용하는 하드웨어 컴포넌트와 내부 작업이 수행되는 애플리케이션으로 구성한다. 애플리케이션은 입출력 컴포넌트, 자연어 처리 프로세스, 버스 노선 탐색 프로세스를 포함하고, 세부적으로 입출력 컴포넌트는 STT, TTS, 결과 표현을 포함한다. 시스템 환경에서는 하드웨어의 자세한 묘사를 생략하고 소프트웨어 시스템을 중심적으로 표현하였다.

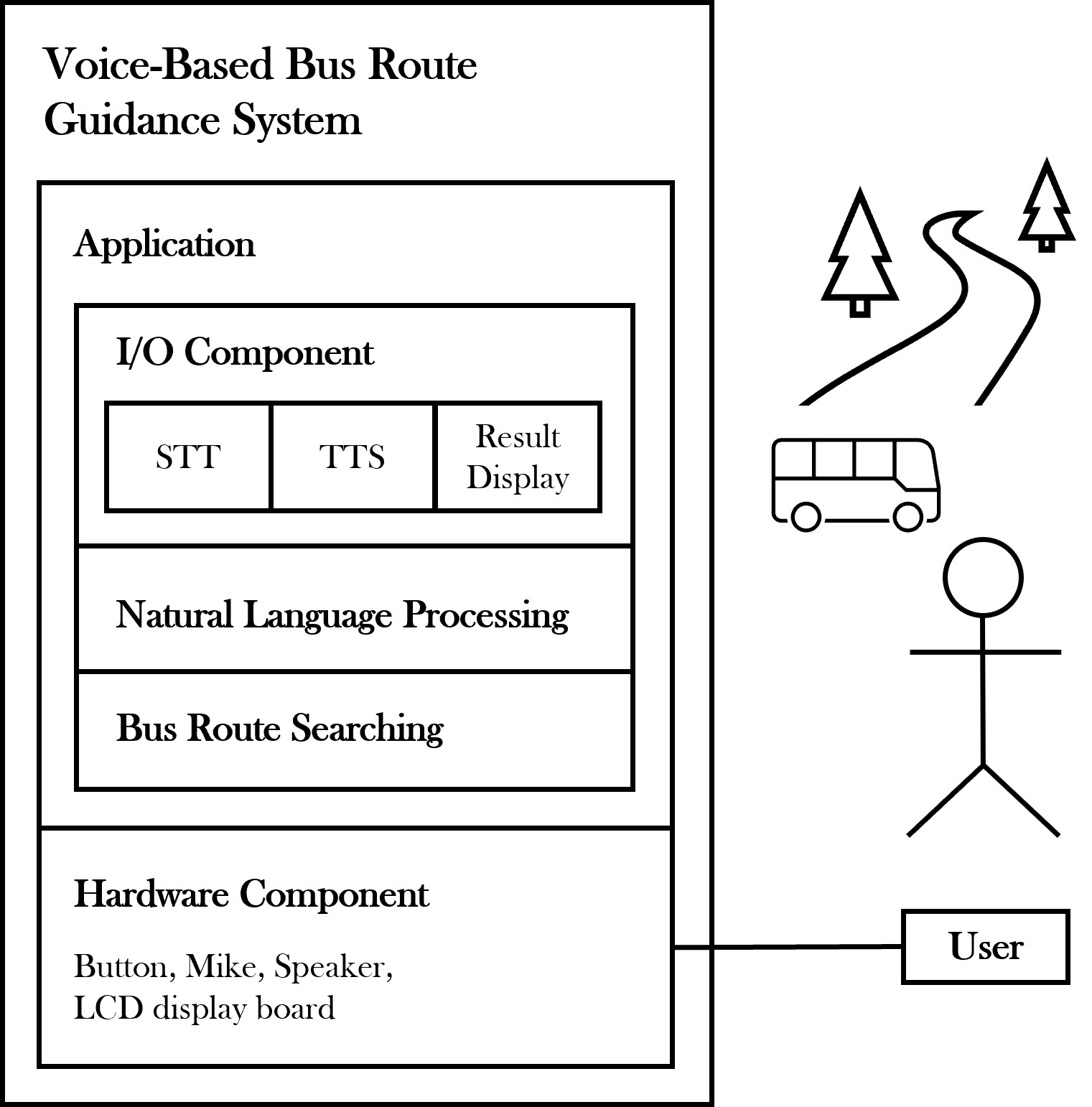


Figure 2 - System Environment

## **2.3 Product Functions**

|  |  |  |
| --- | --- | --- |
| **ID** | **Function** | **Description** |
| **1** | **음성 입력**  (STT Execution) | 마이크를 통해 입력 받은 음성에 대하여 STT(Speech to Text)를 수행함. |
| **2** | **자연어처리**  (Natural Language Processing) | 텍스트로 변환된 데이터를 이용하여 자연어 처리와 문장 분석을 수행함. |
| **3** | **버스노선 검색**  (Bus Route Searching) | 목적지 데이터를 기반으로 버스 노선을 탐색함. |
| **4** | **음성 출력**  (TTS Execution) | 출력 문장을 TTS(Text to Speech)를 통하여 음성으로 변환함. |
| **5** | **화면 표시**  (Display) | 현재 작업의 진행도 및 작업 결과 등을 LCD 모니터에 표현함. |

# **3. Specific Requirements**

## **3.1 User Requirements**

### ***3.1.1 Functional Requirements***

**음성 인식 기능**

사용자는 시스템에게 ‘경복궁 역’과 같은 특정 음성 명령어를 사용해 본인이 가고자 하는 목적지의 경로 안내를 요청할 수 있다. ‘목적지를 말씀해주세요.’ 같은 안내 음성을 통해 본인이 해야 할 행위를 파악할 수 있다. 사용자는 기기 시작 버튼을 이용해 작업을 시작하고 기기를 활성화할 수 있다.

**자연어 처리 기능**

원할한 TTS 기능이 선행되어야 한다. 자연어 음성 만으로도 사용자의 요청을 이해하여 사용자가 원하는 목적지에 가장 근접한 정보를 받아갈 수 있도록 해야 한다.

**데이터 처리 및 검색 기능**

공공 API, 혹은 데이터 베이스에서 사용자가 정보를 제공받았을 시 가장 효율적인 노선 정보를 불러와야 한다. 공공데이터가 추후 업데이트 되어도 길찾기 프로그램은 이상 없이 수행되어야 하며, 새롭게 업로드된 공공데이터 정보를 원활하게 활용해야 한다.

**음성 합성 및 출력 기능**

탑재된 스피커에서 출력 음성이 나오면 사용자는 오류 없이 필요한 정보를 받을 수 있도록 한다. LCD 모니터를 사용하여 음성으로 출력되는 사운드적 부분뿐만 아니라 시각적으로도 요청한 명령 정보 등을 확인할 수 있다. 스피커의 음량이 최소 65dB을 이하가 되지 않게 주의한다. 사용자가 음성을 들었을 때 명확히 이해할 수 있어야 하며, 정보 전달에 문제가 없어야한다.

**오류 처리 및 예외 상황 대응 기능**

부정확한 정보나 시스템이 사용자의 명령을 인식하지 못했을 경우 사용자는 경로 검색을 다시 요청할 수 있다. 사용자 명령을 입력하지 못했을 경우, ‘다시 말씀해주세요’ 혹은 ‘목적지가 검색되지 않았습니다’ 같은 안내 문구를 사용자가 들을 수 있어야 한다. 사용자의 음성 명령 대기 시간이 길어지는 경우, 사용자의 원활한 사용성을 위해 리마인드 안내 문구나 처음으로 되돌아가는 리셋 작업이 필요할 수 있다. 사용자가 재검색을 요청했을 시에도 똑같은 오류가 반복된다면 사용자는 고객 센터에 연락을 취해 오류 처리를 요청할 수 있다.

### ***3.1.2 Non-functional Requirements***

**사용성**

안내 음성의 멘트를 최소화 하여 사용자가 음성을 이해하기에 도움을 주어 사용성을 높여야 한다. 비교적 사용이 쉬운 버튼형 기기를 이용해서 재사용성을 유도한다. 스피커, 마이크, LCD모니터 같은 입출력 장치와 연동해 사용자에게 결과 전달을 확실하게 해주어야 한다. 사용자가 음성을 듣는 것에 있어서 불편함을 최소화 하기 위해 음질이 좋은 스피커와 증폭기를 사용한다.

**효율성**

**성능:** 시스템 처리 시간(10초)를 따져보고 음성 처리가 너무 늦어지지 않게 효율성을 따져보아야 한다. 버튼이 눌린 후 안내 음성이 나오기까지의 시간은 2초 이내여야 한다. 목적지 입력 후 결과 출력까지의 시간은 15초 이내여야 한다.

**공간:** 모든 프로세스는 라즈베리 파이의 메모리 사용량을 초과하지 않아야 한다.

**신뢰성**

사용자에게 정확한 정보를 주고 사용자의 피드백이나 후기 등을 수집하여서 신뢰성을 높인다. 사용자가 잘못된 장소를 말하거나, 발음이 불명확해 입력 받은 음성을 사용할 수 없을 경우 재입력을 요구하는 안내 멘트로 사용자가 다시 말할 수 있게 해주어야 한다. 사용자가 말한 목적지 경로에 해당하는 버스 번호가 출력되어야 한다. 우천시에 외부로 노출된 입출력 장치를 보호하고, 사용자의 시야 확보를 하여 이용에 어려움이 없도록 하기 위해 차양막을 설치한다.

**이식성**

사용하는 운영체제나 프레임워크, DB 등의 버전이 업데이트 되어도 정상적으로 동작해야 한다. 동일한 프로그램을 상위 버전의 라즈베리 파이에 똑같이 적용해도 정상적으로 수행되어야 한다.

## **3.2 System Requirements**

### ***3.2.1 User Interfaces***

제품은 사용자가 원하는 목적지의 버스 노선 정보를 제공하는 것을 주된 기능으로 한다. 기본적으로 마이크, 전광판, 버튼, 스피커를 이용한 인터페이스가 제공되며 전체 시스템 정보와 유지 및 관리는 시스템 관리자에 의해 처리 된다.

### ***3.2.2 User Interfaces Example***

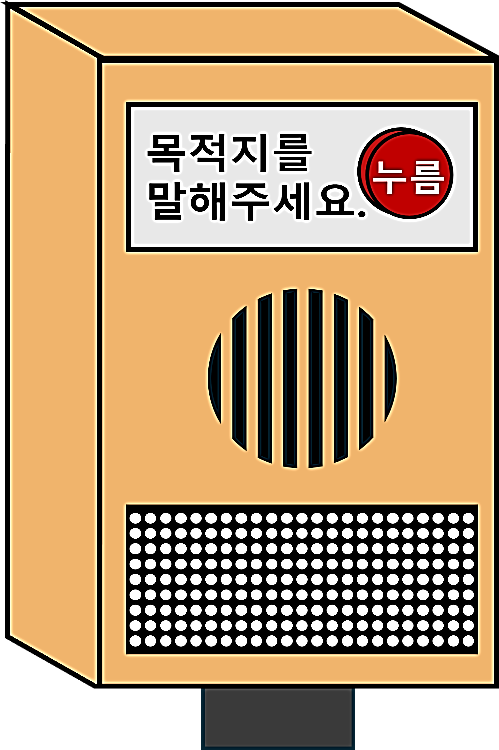


Figure 3 - User Interfaces Example

### ***3.2.3 Hardware Design and External Status***

버스 노선 표지판 또는 별개의 장치로 설치되어 사용자가 쉽게 기기를 알아볼 수 있도록 해야 한다. 기기는 정보 약자의 큰 비중을 차지하는 여성 노인들의 평균 키와 눈높이에 맞춰140~150cm로 제작되어야 한다. 시작 버튼은 가장 먼저 작동되는 장치로 사용자가 쉽게 인식할 수 있도록 빨간색과 같은 확장색을 사용해야 한다. 그러나 심미적 이유로 다른 색상으로 교체할 수 있다. 설치 장소가 기기의 훼손 가능성이 높은 야외라는 점을 고려해 응융 아연 도금 강판을 사용해야 한다. 그러나 금속의 가공이 어렵다는 점을 인지하여 가제작시 폼보드 혹은 크라프트 박스 등을 사용할 수 있다. 안내문은 문구를 쉽게 알 수 있도록 LCD 모니터에 띄워야 한다. 설치 장소가 야외이기 때문에 우천 시 고장을 방지하기 위해 작은 차양막을 설치해야 한다. 이 외에 제작하는 과정 또는 제작 후 필요성을 느끼지 못하는 디자인 및 기기의 일부는 수정이 가능해야 한다.

### **3.2.3.1 User Input Method**

사용자는 ‘누름’ 버튼을 눌러 시스템을 가동시킬 수 있어야 한다. 사용자가 원하는 목적지를 말하기 위해 목소리를 입력 받을 수 있는 마이크가 내장되어야 한다. 사용자에게 간단한 안내와 원하는 결과를 음성으로 반환할 수 있는 스피커가 내장되어야 한다. 사용자가 음성 외에 시각적으로도 정확히 확인 할 수 있는 전광판이 설치되어야 한다. 이 외에 구현 과정에서 필요 유무에 따라 기능 수정이 가능해야 한다.

### **3.2.3.2 LCD display board**

사용자가 쉽게 인식할 수 있도록 제품 크기에 맞춘 사이즈로 제작되어야 한다. 정보 약자, 즉 노인 계층을 겨냥하여 만든 제품이기 때문에 시력이 약한 점을 고려하여 검은색 바탕에 알아보기 쉬운 형광색 글자색으로 설정해야 한다. 경로 안내 시 버스 번호를 표기하여 사용자가 시각적으로도 인지할 수 있도록 해야 한다. 개발자의 판단 하에 따라 LCD모니터는 포터블 모니터와 같은 일반 디스플레이로 교체될 수 있다. 이 외에 구현 과정에서 필요 유무에 따라 기능과 하드웨어 수정이 가능해야 한다.

### **3.2.3.4 Responsiveness and errors**

사용자가 ‘누름’ 버튼을 누르면 간단한 안내 메시지가 음성으로 출력 되어야 한다. 시스템이 작동하는 동안 사용자가 어떤 상태인지 알 수 있도록 중간 안내 메세지를 음성으로 출력해야 한다. 오류가 발생한 경우 사용자에게 명확한 오류 메시지가 표시 또는 음성으로 출력 되어야 한다. 시스템이 사용자의 의도를 정확히 인지하지 못했을 때 다시 입력해야 함을 안내해야 한다. 사용자가 정확한 목적지를 말했음에도 시스템이 잘못된 정보를 반환한 경우, 사용자가 ‘누름’ 버튼을 누르면 시스템이 다시 처음 상태로 돌아가 음성 입력을 받을 수 있어야 한다. 이 모든 오류가 사용자의 선에서 해결되지 않을 때는 관리자 번호로 문의할 수 있도록 안내 문구를 표시해야 한다.

### **3.2.3.4 Hardware Interface**

라즈베리 파이 기반의 SBC 및 사용자와 상호작용을 하기 위한 푸시 버튼, 마이크, 스피커 등의 부품이 필요하다. 이 과정에서 라즈베리 파이와 기타 장치를 연결하기 위한 전자공학 부품이 요구된다. 효율적인 작업과 안전한 작업을 위해 무납땜 프로토타입 제작 후 본품을 제작해야 한다. 라즈베리 파이는 안정적인 기능 구현을 위해 최소 4세대 이후의 모델이 필요하다.

### ***3.2.4 Software Interface***

### **3.2.4.1 STT**

Google Cloud Platform에서 Speech-to-Text API를 사용하기 위한 API 키를 생성하고 관리해야 한다. 하드웨어 요구사항과 Google Speech-to-Text API의 호환성을 확인해야 한다. API 버전 관리 및 업데이트에 대한 정책을 고려해야 한다. 실시간 음성 인식 기능으로 사용자가 음성으로 버스 노선 정보를 요청할 수 있어야 한다. 음성 인식 결과를 텍스트로 변환하는 기능이 요구된다. 음성 입력에 대한 빠른 응답 시간이 필요하며, 정확한 음성 음식 기능이 보장되어야 한다. API 키의 안전한 관리 및 사용이 보장되어야 한다. API 호출 중 발생 가능한 에러 처리 방법과 예외 상황에 대한 대응 방안을 명시해야 한다. 작업 중 사용된 오디오 데이터는 SpeechRecognition 라이브러리를 사용해 형식과 크기에 상관없이 처리 및 관리되어야 한다.

### **3.2.4.2 NLP**

Okt(Open Korean Text) 형태소 분석기와 TF-IDF, SVM(Support Vector Machine)을 이용하여 사용자의 요청 문장에서 목적지 정보를 파악하기 위한 NLP 작업이 수행되어야 한다. 사용되는 형태소 분석기와 NLP 모델은 다양한 언어의 텍스트 데이터를 처리할 수 있어야 하며, 그 중에서도 한국어 처리 능력이 필수로 요구된다. 자연어 처리 모델은 사용된 Raspberry Pi의 성능과 메모리 용량에 적합하도록 설계되어야 한다.

### **3.2.4.3 Bus Route Searching**

버스 노선을 검색하는 과정에서 실제 운행 정보와의 격차를 줄이기 위해 실시간 데이터를 API를 통해 제공받아야 한다. Naver Maps API를 사용해 목적지의 위도와 경도 정보를 조회해야 한다. 검색 알고리즘은 사용된 Raspberry Pi의 성능과 메모리 용량에 적합하도록 설계되어야 한다.

### **3.2.4.4 TTS**

Google Cloud Platform에서 Text-to-Speech를 사용하기 위한 API 키를 생성하고 관리해야 한다. 하드웨어의 요구사항과 TTS API의 호환성을 확인해야 한다. 음성의 톤, 속도, 음높이 등을 조절할 수 있어야 한다. 음성 출력에 대한 빠른 응답과 자연스러운 합성이 보장되어야 한다. 경우에 따라 eSpeak 라이브러리를 사용하여 시스템을 구현하여도 문제없이 작동해야 한다.

### ***3.2.5 Cost Requirements***

최초 제작 비용은 30만원 이내로 제작이 가능해야 한다. 차양막 설치에 추가 비용은 최초 제작 비용을 넘지 말아야 한다. 수집된 데이터를 유지하고 관리하기 위한 비용이 요구되어야 한다.시스템의 안정성과 성능을 유지하기 위한 업데이트 및 버그 수정 비용이 요구된다. 버스 위치 및 노선 정보를 저장할 데이터 저장소를 구축하고 운영하는 비용을 고려해야 한다. 실시간 버스 위치 및 노선 정보를 저장하고 관리하기 위한 데이터 베이스 구축 비용이 요구된다. 시스템의 운영과 성능 모니터링을 위한 비용이 요구된다. 이외의 부가적인 부품 또는 소프트웨어에 의한 비용은 최초 제작 비용을 넘지 말아야 한다.

## **3.3 Other Requirements**

### ***3.3.1 Organizational Requirements***

**배포**

미리 특정한 기능을 수행하는 소프트웨어를 내장한 임베디드 시스템이 적용된 기기를 필요한 장소마다 배치한다.

공공기관 배포 혹은 상업적 판매시 인터넷으로 직접 전송해 온라인으로 다운로드 가능하게 한다.

**구현**

애자일 방법론을 따라 고객의 피드백을 수집, 반영한다.

프로그래밍 언어로는 Python을 사용한다.

**표준**

소프트웨어 개발 및 유지 보수 과정은 ISO/IEC 12207 소프트웨어 생명 주기 프로세스 표준을 준수한다. 완성된 소프트웨어는 ISO/IEC 9126 품질 표준을 준수한다.

### ***3.3.2 External Requirements***

**상호 운용성**

공공데이터 API를 적극적으로 활용해 목적지까지의 최적 경로와 버스 노선을 정확하게 제공할 수 있어야 한다.

**윤리적**

본 프로젝트의 결과물 사용에는 연령에 제한을 두지 않는다. 서비스 제공 시 원래 목적이었던 정보 약자 외에도 장애인, 노약자, 어린이를 포함한 모든 사용자들이 사용하는데 차별이나 특정 대상에게만 해당되는 어려움이 없도록 한다.

**법적**

**사생활:** 본 프로젝트에서 저장된 정보나 기록은 오직 관리자 권한을 가진 자들만 확인할 수 있다.

**안전성:** 본 프로젝트에서 수집된 사용자의 음성은 악의적 사용을 방지하기 위해 문제 해결 후 저장하지 않고 삭제된다. 본 프로젝트에서는 사용자의 현 위치와 목적지를 제외한 개인정보를 알 수 있는 그 어떤 정보도 수집하지 않는다.

## **3.4 Restrictions**

시스템은 라즈베리 파이 4 이상 버전에서 정상으로 프로그램을 실행할 수 있어야 한다.

|  |  |
| --- | --- |
| 크기 | 구조물의 크기는 가로 60cm, 세로 160cm를 넘지 않는다. |
| 자원 | 제작 경비 30만원 이하로 결과물을 제작해야 한다. |
| 인력 | 프로젝트에 참여하는 최대 인원은 5명 이내로 해결해야 한다. |
| 기간 | 2024년 10월 31일까지 모든 개발을 완료해야 한다. |

## **3.5 Data Flowchart**

### ***3.5.1 음성 입력 모듈***

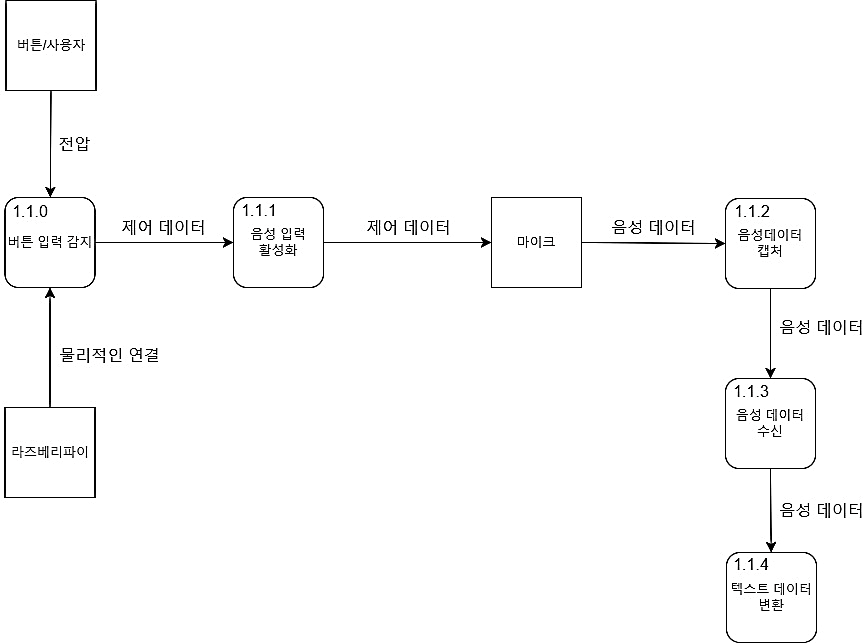
****

Figure 4 - 음성 입력 모듈

|  |  |
| --- | --- |
| **프로세스 번호** | 1.1.0 |
| **프로세스 이름** | 버튼 입력 감지 |
| **설명** | 버튼 입력 감지를 통해 이벤트 발생 |
| button\_pressed = True  if button\_pressed:  이벤트 실행  else:  실행하지 않음 |

|  |  |
| --- | --- |
| **프로세스 번호** | 1.1.1 |
| **프로세스 이름** | 음성 입력 활성화 |
| **설명** | 음성 입력을 시작하고 마이크를 활성화 |
| activate\_voice\_input()  *-- 함수 내용--* |

|  |  |
| --- | --- |
| **프로세스 번호** | 1.1.2 |
| **프로세스 이름** | 음성 데이터 캡처 |
| **설명** | 마이크를 사용하여 음성 캡처 후 처리 |
| def audio():  r = sr.Recognizer()  with sr.Microphone() as source:  print("원하시는 목적지를 말해주세요“)  audio = r.listen(source) |

|  |  |
| --- | --- |
| **프로세스 번호** | 1.1.3 |
| **프로세스 이름** | 음성 데이터 수신 |
| **설명** | 마이크를 통해 입력된 음성을 라즈베리 파이로 수신 |
| 2진 바이트 스트림 형태 |

|  |  |
| --- | --- |
| **프로세스 번호** | 1.1.4 |
| **프로세스 이름** | 텍스트 데이터 변환 |
| **설명** | 음성 데이터를 STT를 이용하여 텍스트로 변환 |
| import speech\_recognition as sr  ·  def audio()  ·  text = audio() |

### ***3.5.2 자연어 처리 모듈***

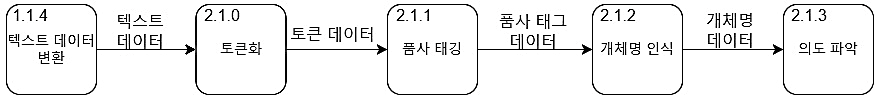
****

Figure 5 - 자연어 처리 모듈

|  |  |
| --- | --- |
| **프로세스 번호** | 2.1.0 |
| **프로세스 이름** | 토큰화 |
| **설명** | 문장을 단어 또는 구절로 분리 |
| import spacy  nlp = spacy.load('ko\_core\_news\_sm')  doc = nlp(text)  tokens = [token.text for token in doc] |

|  |  |
| --- | --- |
| **프로세스 번호** | 2.1.1 |
| **프로세스 이름** | 품사 태깅 |
| **설명** | 각 토큰이 문장에서 갖는 역할을 태깅 |
| 각 토큰에 대해 개별적으로 품사 태깅 수행  pos\_tags = [(nlp(token)[0].text, nlp(token)[0].pos\_) for token in tokens] |

|  |  |
| --- | --- |
| **프로세스 번호** | 2.1.2 |
| **프로세스 이름** | 개체명 인식 |
| **설명** | 문장에서 특정 유형의 개체 인식 |
| 주어진 텍스트에서 개체명 인식  entities = [(ent.text, ent.label\_) for ent in doc.ents] |

|  |  |
| --- | --- |
| **프로세스 번호** | 2.1.3 |
| **프로세스 이름** | 의도 파악 |
| **설명** | 사용자의 요청이나 의도 파악 |
| 사용자가 버스 번호를 요청하고 있는 것을 인식 |

### ***3.5.3 버스 노선 검색 모듈***

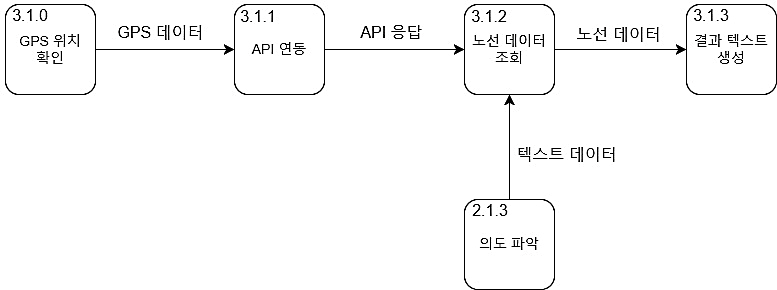
****

Figure 6 - 버스 노선 검색 모듈

|  |  |
| --- | --- |
| **프로세스 번호** | 3.1.0 |
| **프로세스 이름** | GPS 위치 확인 |
| **설명** | 사용자의 위치 확인 |
| 사용자의 현재 위치 확인  import googlemaps  gmaps = googlemaps.Client(API 키 발급)  geocode\_result = gmaps.geolocate() |

|  |  |
| --- | --- |
| **프로세스 번호** | 3.1.1 |
| **프로세스 이름** | API 연동 |
| **설명** | 버스 노선 정보 검색을 위해 API를 연동 |
| 버스 번호, 버스 노선, 정류장 이름 등이 포함된다.  def get\_businfo(가져올 정보):  개인 고유 발급 키  url = “API 주소” |

|  |  |
| --- | --- |
| **프로세스 번호** | 3.1.2 |
| **프로세스 이름** | 노선 데이터 조회 |
| **설명** | 사용자가 원하는 목적지를 통과하는 버스 노선 검색 |
| 사용자가 원하는 목적지로 가는 버스인지 확인  if destination in getbstopnm:  bus\_routes.append(routeno) |

|  |  |
| --- | --- |
| **프로세스 번호** | 3.1.3 |
| **프로세스 이름** | 결과 텍스트 생성 |
| **설명** | 검색된 결과를 텍스트로 생성 |
| 검색된 버스 번호를 사용하여 사용자에게 안내하는 텍스트 생성  버스 번호를 찾지 못한 경우와 버스 번호를 찾은 경우를 포함 |

### ***3.5.4 음성 출력 모듈 (스피커)***

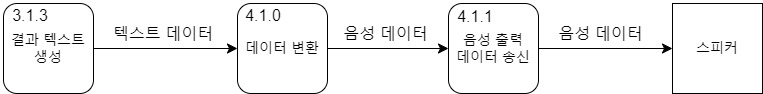
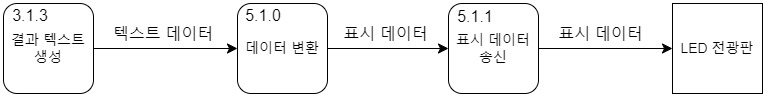
****

Figure 7 - 음성 출력 모듈 (스피커)

|  |  |
| --- | --- |
| **프로세스 번호** | 4.1.0 |
| **프로세스 이름** | 데이터 변환 |
| **설명** | 결과 텍스트를 음성 데이터로 변환 |
| from gtts import gTTS  text = “해당 목적지로 가는 버스 번호는 \*\*\*입니다.”  tts = gTTS(text = text, lang=’ko’)  tts.save(“output,mp3”) |

|  |  |
| --- | --- |
| **프로세스 번호** | 4.1.1 |
| **프로세스 이름** | 음성 출력 데이터 송신 |
| **설명** | 음성 데이터를 스피커로 송신 |
| import pygame  pygame.mixer.init()  pygame.mixer.music.load("output.mp3")  pygame.mixer.music.play() |

### ***3.5.5 표시 출력 모듈 (LCD 모니터)***

****

LCD 모니터

Figure 8 - 표시 출력 모듈 (LCD 모니터)

|  |  |
| --- | --- |
| **프로세스 번호** | 5.1.0 |
| **프로세스 이름** | 데이터 변환 |
| **설명** | 결과 텍스트를 표시 데이터로 변환 |
| 결과 텍스트에서 숫자만 추출  import re  text = “해당 목적지로 가는 버스 번호는 \*\*\*입니다.”  numbers = re.findall(r'\d+', text)  다시 문자열로 변환  str\_num = str(numbers) |

|  |  |
| --- | --- |
| **프로세스 번호** | 5.1.1 |
| **프로세스 이름** | 표시 데이터 송신 |
| **설명** | 표시 데이터를 LCD모니터로 송신 |
| LCD 모니터에 표시 데이터를 보낸다.  버스 번호만 포함된다. |

# **4. Appendices**

## **A. Dictionary**

**A.1 음성 입력 모듈**

1. 전압 = [0~5 사이의 실수]

2. 제어 데이터 = {true | false}

3. 음성 데이터 = [2진 바이트 스트림]

**A.2 자연어 처리 모듈**

1. 텍스트 데이터= [문자열]

2. 토큰 데이터= [문자열]

3. 품사 태그 데이터 = [문자열]

4. 개체명 데이터 = [문자열]

**A.3 버스 노선 검색 모듈**

1. GPS 데이터= [위도와 경도를 포함한 실수]

2. API 응답= [버스 번호, 버스 노선, 정류장 이름 등]

3. 노선 데이터 = [문자열]

4. 텍스트 데이터 = [문자열]

**A.4 음성 출력 모듈 (스피커)**

1. 텍스트 데이터 = [문자열]

2. 음성 데이터 = [2진 바이트 스트림]

**A.5 표시 출력 모듈 (LCD 모니터)**

1. 텍스트 데이터 = [문자열]

2. 표시 데이터 = [문자열]

## **B. References**

- MonkSimon. (2017). “Electronics Cookbook.” O'Reilly Media.

- 김성우. (2020). “사물인터넷을 품은 라즈베리 파이.” 제이펍.