**<PROJECT 스펙보고서>**

**피곤하쥬?**

**2016100930 고은성**

**2016104113 김예미**

**2014104121 유지훈**

**2014104141 장연우**

**김재홍 교수님**

**Table of Contents**

**1. Abstract**

**2. Introduction**

**3. Background Study**

**A. 관련 접근방법/기술 장단점 분석**

**B. 프로젝트 개발환경**

**4. Goal/Problem & Requirements**

**5. Approach**

**6. Project Architecture**

**A. Architecture Diagram**

**B. Architecture Description**

**7. Implementation Spec**

**A. Input/Output Interface**

**B. Inter Module Communication Interface**

**C. modules**

**8. Current Status**

**9. Future Work**

**10. Division & Assignment of Work**

**11. Schedule**

**Abstract**

‘피곤하쥬?’ 시스템은 강의실 내 강의자의 수업 내용을 녹음 및 분석하고, 이를 텍스트화하여 서버에 저장한다. 동시에 수업을 듣는 학생들의 심장박동을 센서로 측정하여 일정 심박이하로 뛸 경우 졸고 있다고 판단, 수면 시간 정보를 저장한다. 녹음된 음성은 특정 키워드, 혹은 일정 데시벨 이상인 부분은 하이라이팅하여 텍스트화 한 뒤 저장한다.

**Introduction**

최근 IT분야 기술발달로 5G와 IoT의 결합이 많은 주목을 받고 있다. 이러한 흐름에 발맞추어본 프로젝트 구성원의 본업인 학부생의 입장에서 기존 강의 시스템을 개선하고, 보다 공평한 교육의 기회를 제공할 방법을 구하고자 하였다. 아르바이트와 스터디, 프로젝트 등을 병행하느라 피로가 누적된 학생들을 위해, 졸음으로 수업을 듣지 못하는 동안 강의자가 전달하는 내용을 텍스트화하여 제공해주는 시스템을 구현해보고자 하였다.

**Background Study**

**관련 접근방법/기술 장단점 분석**

본 프로젝트의 구현은 음성 인식 및 처리, 심장박동 센서, 디바이스 간 통신의 방식으로

접근한다. aws transcribe로 음성을 인식하여 텍스트화하고 awsIoT 플랫폼을 이용하여

디바이스 간 데이터를 주고받는다. 사용할 Cloud 플랫폼은 AWS로 선정되어 있으며 서버로 구동 할 장치인 arduino와 raspberry pi의 성능의 장단점을 분석하여 선정한다.

**arduino vs raspberry pi**

**arduino** : 마이크로컨트롤러 마더보드. 마이크로컨트롤러는 한 번에 하나의 프로그램을 여러 번 반복해서 실행시킬 수 있는 단순한 컴퓨터. 단순한 반복 작업에 사용하기 좋음

**raspberry pi** : 범용 컴퓨터로 보통 운영체제로 리눅스를 사용하며 한 번에 여러 개의 프로그램을 실행 가능. 좀 더 복잡한 로봇을 구동하거나, 여러 작업을 실행시키거나, 비트코인이나 암호화처럼 계산이 많이 필요한 경우에 사용 본 프로젝트는 분석 과정과 멀티 IoT기기를 연동하기 때문에 raspberry pi로 채택한다.

**프로젝트 개발환경**

OS : Raspbian4.19

IDE : vim, VScode

버전관리 : git

Storage : AWS s3

DB : postgreSQL

Language : Python

그 외 AWS : transcribe, SES

**Goal/Problem & Requirements**

프로젝트 목표 : 사용자가 심장박동센서를 착용하여 수면 예측 시 수면 시간 동안의 강의

파일(텍스트)을 이메일로 전송한다. 강의 내용은 음성 인식 및 번역을 통해 텍스트화 된다.

요구사항 : 강의자의 동의, 고성능의 녹음기, AWS 비용, 정밀한 심장박동 센서 등이 요구된다.

**Approach**

실제 서비스와 같이 유연하게 작동하는 서비스를 제작하는 것이 프로젝트의 궁극적인

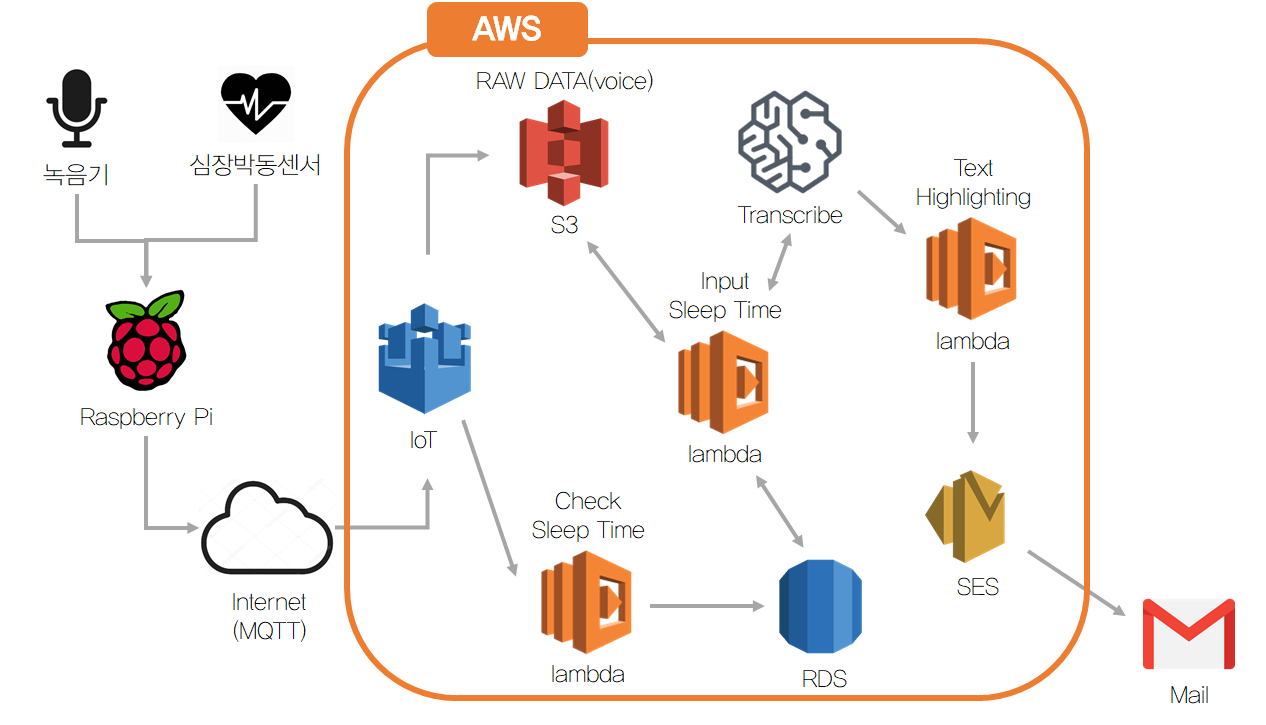
목표이다. 실제 제작 과정에서 기존 서비스와 비교해보며 필요하다고 생각되는 기능이

있다면 필요에 따라 추가하도록 한다. 또한, 철저히 사용자 입장에서 보안성, 비용

최적화를 우선적으로 고려하며 프로젝트를 제작한다.

**Project Architecture**

**Architecture Diagram**



**Architecture Description**

1. 수업이 종료 되면, 녹음기와 심장박동센서로 수집한 데이터는 Raspberry Pi로 수집된다.

2. 수집 된 데이터는 MQTT 프로토콜을 이용하여 AWS IoT와 연결되어 Raw data(음성)은 S3에 저장한다. 또한 동시에 수면시간을 측정한 결과 값을 사용자 key 값과 함께 DB에 저장한다.

3. Raw data에서 수면시간을 tokenizing 하여 음성인식 시스템인 Transcribe에 input 데이터로 사용한다.

4. 3.의 output값인 Text 데이터를 정렬, 문단 나누기, 중요부분 강조를 통해 사용자 입장에서 보기 편하게 정리한다.

5. AWS SES서비스를 이용하여 사용자에게 4.의 정리된 Text를 email로 전송한다.

MQTT protocol : IoT기기들에 최적화된 가벼운 메세징 프로토콜. 기존에 웹에서 통신하던 HTTP등의 프로토콜보다 제한적이고 특수한 상황에서 사용할 수 있는 모바일 특화 프로토콜

AWS IoT : IoT 제품과 AWS 클라우드 간에 안전한 양방향 통신을 제공

AWS S3 : 클라우드용 데이터 스토리지

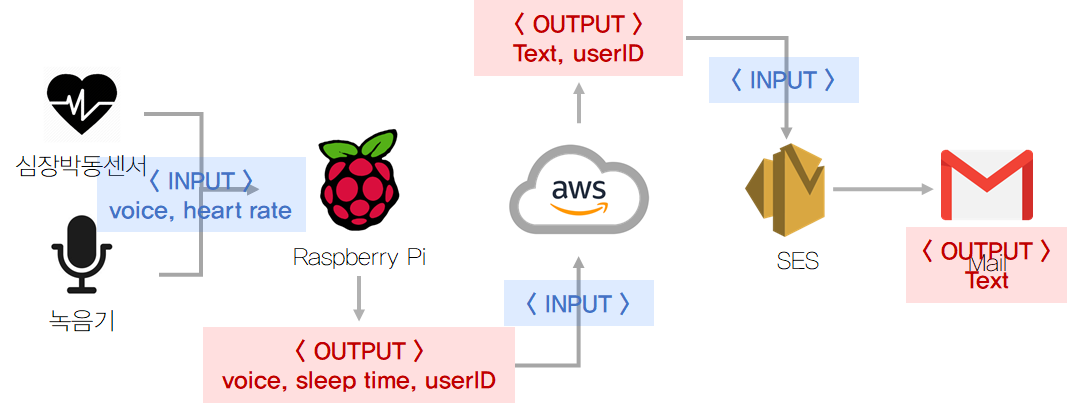
AWS lambda : 이벤트 발생 시 코드를 실행하고 해당 코드에 대한 컴퓨팅 리소스 자동 관리

AWS Transcribe : 음성 데이터를 텍스트 데이터로 변환

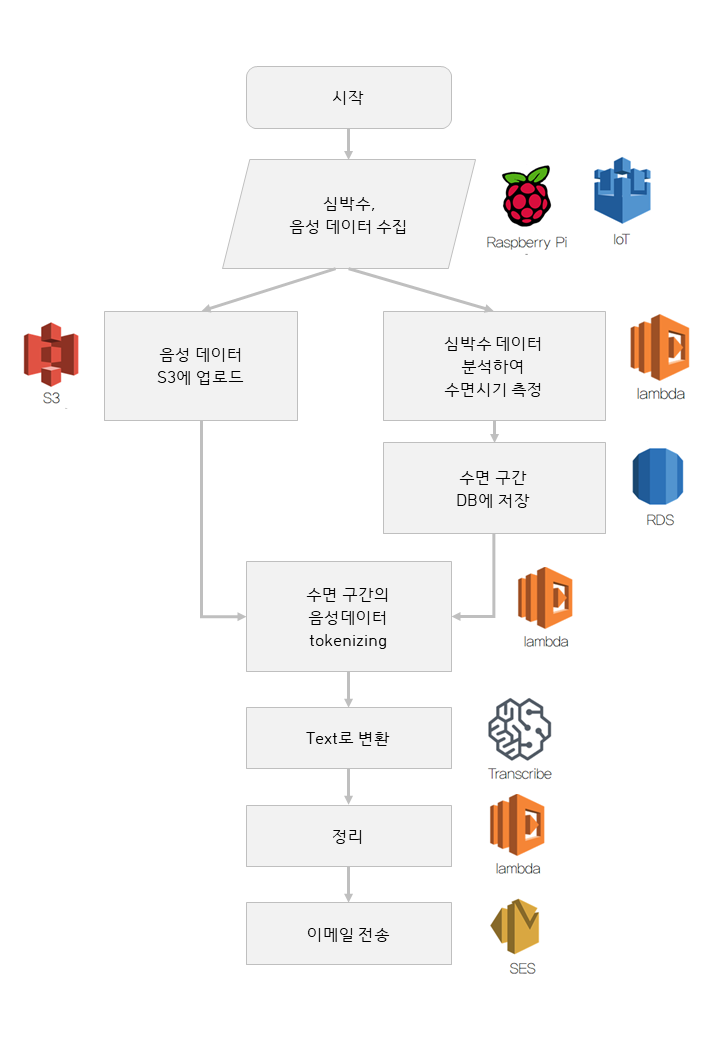
AWS SES : 사용자의 이메일 주소와 도메인을 사용하여 이메일을 보내고 받기 위한 플랫폼

**Basic Spec**

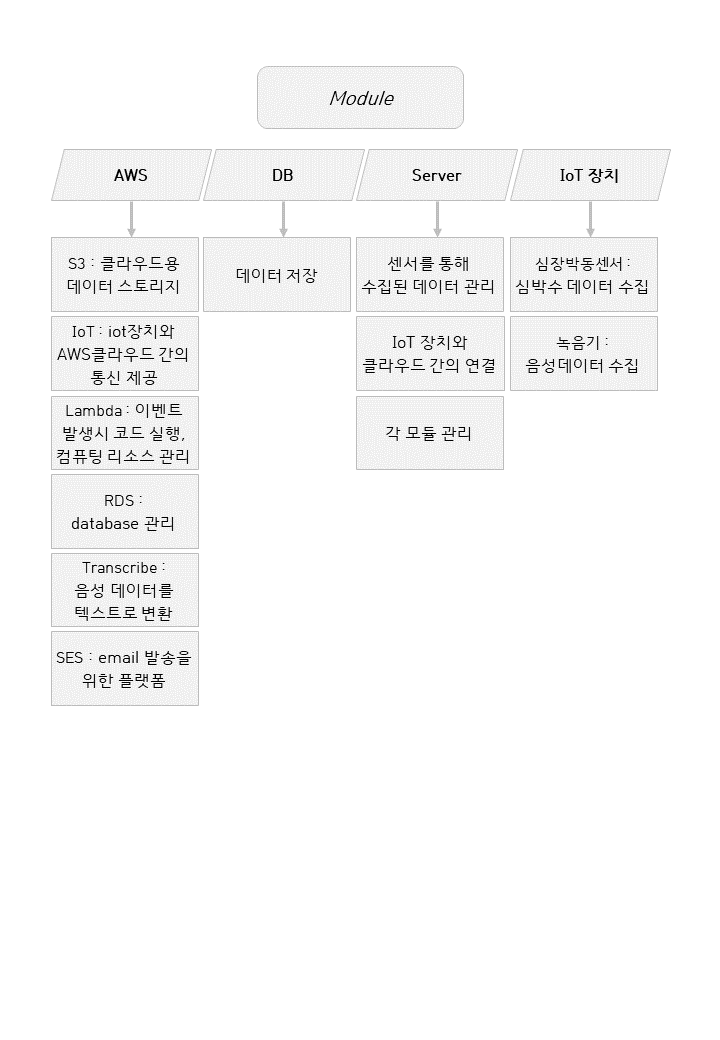
**A. Input/Output Interface**



**B. Inter Module Communication Interface**

****

**C. Module**



**Current Status**

프로젝트 아키텍처를 구상 후 각 단계에 맞는 요구사항을 분석해서 음성, AWS, 인터넷 통신, 서버개발로 task를 나누어 background 지식에 대해 습득 중에 있다. H/W는 심장박동 센서로 DM447(5,060원)을 선택 하였고 녹음기로는 KVR-21(39,800원), 무선랜카드로는 NEXT-202N(4,800원)으로 총 비용은 택배비 별도 49,660원이다.

**Future Work**

1. 음성처리 전처리를 통해 잡음을 최소화 시킨 후, 억양이나 강세를 파악해 텍스트의 output값 정확도 향상 및 텍스트 하이라이팅
2. DB 선택 및 스키마 설정 후 테이블 정규화
3. Cache data를 생성하여 cloud cost 최소화 방안 모색

++

1. 사용자에 대한 보안 향상(cognito)
2. 데이터 분석을 통해 교수에게 수업 흥미도 유발 부분, 흥미도 저하부분 insight 제공

**Division & Assignment of Work**

|  |  |
| --- | --- |
| 항목 | 담당자 |
| 음성처리 | 고은성, 유지훈 |
| 서버개발 | 김예미, 장연우 |
| DB, AWS관리 | 장연우 |
| 보고서, PPT | 고은성, 김예미 |
| 발표 | 유지훈 |

**Schedule**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 내용 | 9월 | | | 10월 | | | | 11월 | | | | 12월 | |
| 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 요구사항 분석 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 백그라운드 습득 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 음성 인식 모듈 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 서버 모듈 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 프로세스 병합 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 테스트 및 릴리즈 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종 보고서 |  |  |  |  |  |  |  |  |  |  |  |  |  |