

# 요구사항 분석 및 설계서

TOOKTAK  
202011289 문찬규  
202011290 민지홍  
202011356 임제형  
202211328 윤찬규  
**202211404 황서진**

## 목 차

<b>1. 프로젝트 소개 . . . . .</b>	<b>2</b>
<b>2. 기본 사항 . . . . .</b>	<b>2</b>
2.1. 작동 환경 . . . . .	2
2.2. 프로그램 구성 . . . . .	2
2.3. 프로그램 설치 및 실행 . . . . .	2
<b>3. 프로그램 사용 흐름도 . . . . .</b>	<b>2</b>
<b>4. 용어 . . . . .</b>	<b>2</b>
<b>5. 데이터 요소 . . . . .</b>	<b>3</b>
5.1. 계정 . . . . .	3
5.1.1. 아이디 . . . . .	3
5.1.2. 비밀번호 . . . . .	3
5.1.3. 권한 . . . . .	3
5.2. 파일 . . . . .	3
5.2.1. 이름 . . . . .	3
5.2.2. 형식 . . . . .	3
5.2.3. 버전 . . . . .	3
5.2.4. 챕터 . . . . .	3
5.2.5. 길이(크기) . . . . .	4
5.2.6. 생성 일시 . . . . .	4
5.2.7. 유저 . . . . .	4
5.2.7.1. 이름 . . . . .	4
5.2.7.2. 나이 . . . . .	4
5.2.7.3. 성별 . . . . .	4
5.2.7.4. AI 분석 결과 . . . . .	4
5.2.7.5. 장소 . . . . .	4
5.2.8. 저장 경로 . . . . .	4
5.2.9. 이외의 데이터 요소 . . . . .	4
<b>6. 데이터 파일 . . . . .</b>	<b>4</b>
6.1. 병원 데이터 파일 . . . . .	5
6.2. 보이노시스 서비스 데이터 파일 . . . . .	5
6.3. 로그 파일 . . . . .	5
<b>7. 데이터베이스 구조 . . . . .</b>	<b>5</b>

<b>8. API 목록</b>	<b>5</b>
8.1. 계정	5
8.1.1. 로그인	5
8.1.2. 로그아웃	5
8.1.3. 서브 관리자 계정 생성	5
8.1.4. 서브 관리자 계정 제거	6
8.1.5. 서브 관리자 계정 목록	6
8.1.6. 관리자 계정 아이디 및 비밀번호 재설정	6
8.1.7. 로그 확인	6
8.2. 파일	6
8.2.1. 파일 업로드	7
8.2.2. 파일 다운로드	7
8.2.3. 파일 검색	7
<b>9. 프로그램 실행 화면</b>	<b>7</b>
9.1. 로그인 화면	7
9.2. 메인 화면	7
9.3. 계정 관리 화면	7
9.3.1. 생성	8
9.3.2. 삭제	8
9.4. 데이터 관리 화면	8
9.4.1. 삽입	8
9.4.2. 삭제	8
9.5. 데이터 분석 화면	8
9.5.1. 필터링	8
9.5.2. 로그	8

## 1. 프로젝트 소개

보이노시스 서비스는 음성 / 영상 데이터를 토대로 난청, 인지장애, 치매 질병 예방 및 관리를 목적으로 개발된 의료 AI 활용 디지털 헬스케어 서비스입니다. 기존 보이노시스 서비스에서는 사용자의 데이터를 이용해 AI 분석 결과를 얻는 과정에서 해당 데이터는 버려지고 있었습니다. 이번 프로젝트는 이 데이터를 효율적으로 저장 및 관리하는 도구를 제작하여 차후 AI 모델 성능 향상을 목적으로 합니다. 즉, 사내 관리자들이 데이터를 저장 및 분석할 수 있는 웹 서비스를 제공하는 것을 목표로 합니다.

## 2. 기본 사항

### 2.1. 작동 환경

Docker Container를 이미지로 만들어 제공하기에 Docker Container를 실행할 수 있는 환경이 필요합니다.

### 2.2. 프로그램 구성

프로그램은 다음과 같이 구성됩니다.

- Frontend: React.js
- Backend: Node.js
- Database: MongoDB(NoSQL)
- Container: Docker

### 2.3. 프로그램 설치 및 실행

## 3. 프로그램 사용 흐름도

## 4. 용어

**숫자:** (서)아라비아 숫자 중 표준 키보드로 직접 입력할 수 있는 10개(U+0030 '0' ~ U+0039 '9')만을 뜻합니다. 즉, 이 문서에서 말하는 "숫자"에는 아라비아 숫자가 아닌 로마 숫자나 각 언어별 고유 숫자 기호들은 포함되지 않으며, 아라비아 숫자 중에서도 전각 숫자, 원·괄호로 둘러싸인 숫자, 위·아래 첨자용 숫자, 수식용 글꼴별 숫자 등은 모두 제외됩니다.

**개행:** 텍스트 형식의 파일을 편집할 때 표준 키보드의 **Enter** 키로 입력할 수 있는 두 문자들(U+000A Line Feed, U+000D Carriage Return) 중 하나 혹은 이들의 조합입니다. 텍스트 편집기에서 **Enter** 키를 누를 때, 둘 중 어떤 문자나 조합이 입력 될 지는 운영체제 / 편집기마다 다르지만, 무조건 사용자가 사용하는 운영체제 / 편집기에서 입력되는 문자(조합)를 개행이라고 부를 것입니다. 사실, "넓은 의미의 개행"에는 이들 둘(의 조합) 외에도 세로 탭(U+000B Vertical Tab)이나 용지먹임(U+000C Form Feed) 등 몇 가지 문자들이 더 포함되지만, 이 기획서에서는 이런 문자들을 혹시 언급할 일이 있으면 반드시 ("개행"이 아니라) 각각의 구체적인 이름으로 부르겠습니다.

## 5. 데이터 요소

### 5.1. 계정

계정과 관련된 데이터 요소 목록입니다.

#### 5.1.1. 아이디

계정의 아이디를 나타내는 문자열입니다.

**문법 형식:** 정규식  $^[a-zA-Z]+[a-zA-Z0-9]*$$

**의미 규칙:** 계정의 아이디는 고유해야 합니다. 즉, 중복되는 아이디가 존재해서는 안됩니다.

#### 5.1.2. 비밀번호

계정의 비밀번호를 나타내는 문자열입니다.

**문법 형식:** 정규식  $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,20}$$

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

#### 5.1.3. 권한

계정의 권한을 나타내는 문자열입니다.

**문법 형식:** <권한><개행>

**의미 규칙:** <권한>은 admin, user1, user2 중 하나의 문자열이어야 합니다.

### 5.2. 파일

AI 분석에 사용되는 파일 데이터 요소 목록입니다.

#### 5.2.1. 이름

파일의 이름을 나타내는 문자열입니다.

**문법 형식:** <파일 이름><개행>

**의미 규칙:** <파일 이름>은 여러 운영 체제 (Windows, Linux 및 MacOS)에서 유효해야 합니다.

#### 5.2.2. 형식

파일의 확장자를 나타내는 문자열입니다.

**문법 형식:** <확장자><개행>

**의미 규칙:** <확장자>는 FTP 혹은 HTTP 방식으로 전송 가능한 파일 형식이어야 합니다.

#### 5.2.3. 버전

병원에서 수집하는 파일 구조의 버전을 나타내는 양의 정수입니다.

**문법 형식:** 정규식  $^[1-9]+[0-9]*$$

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

#### 5.2.4. 챕터

파일이 어떠한 정보를 수집했는지를 나타내는 양의 정수입니다.

**문법 형식:** 정규식  $^[1-9]+[0-9]*$$

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

### 5.2.5. 길이(크기)

파일의 길이를 초 단위로 나타내는 양의 정수입니다.

**문법 형식:** 정규식  $^{[1-9]+[0-9]*\$}$

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

### 5.2.6. 생성 일시

파일이 생성된 일시를 나타내는 문자열입니다.

**문법 형식:** 정규식  $[0-9]\{4\}-[0-9]\{2\}-[0-9]\{2\}$

• YYYY-MM-DD 문자열 형식입니다.

**의미 규칙:**

- MM을 올바르게 변환했을 때, 1 이상 12 이하의 정수입니다.
- DD를 올바르게 변환했을 때, 1 이상 31 이하의 정수이며, YYYY년 DD월에 존재하는 날짜를 나타내야 합니다. 예를 들어, 윤년 2월의 경우 DD가 올바르게 변환되었을 때 1 이상 29 이하의 값이어야 합니다.

### 5.2.7. 유저

파일을 수집한 유저의 정보를 나타내는 데이터 요소입니다.

#### 5.2.7.1. 이름

파일을 수집한 유저의 이름을 나타내는 문자열입니다.

**문법 형식:** <이름><개행>

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

#### 5.2.7.2. 나이

파일을 수집한 유저의 나이를 나타내는 양의 정수입니다.

**문법 형식:** 정규식  $^{[1-9]+[0-9]*\$}$

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

#### 5.2.7.3. 성별

파일을 수집한 유저의 성별을 나타내는 문자열입니다.

**문법 형식:** 정규식  $^{(M|F)\$}$

#### 5.2.7.4. AI 분석 결과

파일을 기반으로 생성된 AI 분석 결과인 MMSE 점수를 나타내는 정수입니다.

**문법 형식:** 정규식  $^{[1-9]+[0-9]*\$}$

**의미 규칙:** MMSE 점수는 0 이상 30 이하의 값이어야 합니다.

#### 5.2.7.5. 장소

파일을 수집된 장소의 이름을 나타내는 문자열입니다.

**문법 형식:** <장소><개행>

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

### 5.2.8. 저장 경로

파일이 저장된 경로를 나타내는 문자열입니다.

**문법 형식:** <경로><개행>

**의미 규칙:** 추가적인 의미 규칙이 존재하지 않습니다.

### 5.2.9. 이외의 데이터 요소

이외의 데이터 요소들이 파일에 추가될 수 있습니다.

## 6. 데이터 파일

## 6.1. 병원 데이터 파일

## 6.2. 보이노시스 서비스 데이터 파일

## 6.3. 로그 파일

# 7. 데이터베이스 구조

## 8. API 목록

### 8.1. 계정

계정과 관련된 API 목록입니다.

#### 8.1.1. 로그인

```
import axios from 'axios';

// 로그인 정보
var account = {
  id: 'id',
  password: 'password'
};

try {
  var response = await axios.post('http://localhost:3001/account/login', account);
} catch (error) {
}

}
```

#### 8.1.2. 로그아웃

```
import axios from 'axios';

try {
  var response = await axios.get('http://localhost:3001/account/logout');
} catch (error) {
}

}
```

#### 8.1.3. 서버 관리자 계정 생성

```
import axios from 'axios';

// 추가하려는 계정 정보
var account = {
  id: 'id',
  password: 'password',
  permission: 'permission'
};

try {
  var response = await axios.post('http://localhost:3001/account/create', account);
} catch (error) {
}

}
```

#### 8.1.4. 서버 관리자 계정 제거

```
import axios from 'axios';

// 삭제하려는 계정 정보
var account = {
  id: 'id'
};

try {
  var response = await axios.post('http://localhost:3001/account/delete', account);
} catch (error) {
}

}
```

#### 8.1.5. 서버 관리자 계정 목록

```
import axios from 'axios';

try {
  var response = await axios.get('http://localhost:3001/account/list');
} catch (error) {
}

}
```

#### 8.1.6. 관리자 계정 아이디 및 비밀번호 재설정

```
import axios from 'axios';

// 변경하려는 계정 정보
var account = {
  id: 'id',
  password: 'password'
};

try {
  var response = await axios.post('http://localhost:3001/account/reset', account);
} catch (error) {
}

}
```

#### 8.1.7. 로그 확인

```
import axios from 'axios';

try {
  var response = await axios.get('http://localhost:3001/account/log');
} catch (error) {
}

}
```

### 8.2. 파일

파일과 관련된 API 목록입니다.

### 8.2.1. 파일 업로드

```
import axios from 'axios';

// 업로드할 파일
var onSubmit = async (e) => {
  e.preventDefault();
  e.persist();
  var file = new FormData();
  for (var i = 0; i < e.target.files.length; i++) {
    file.append('file', e.target.files[i]);
    var metaData = {
      name: 'name',
      format: 'format',
      version: 'version',
      chapter: 'chapter',
      length: 'length',
      created: 'created',
      user: {
        name: 'name',
        age: 'age',
        gender: 'gender',
        mmse: 'mmse',
        place: 'place'
      },
      path: 'path'
    };
    file.append('metaData', JSON.stringify(metaData));
  }

  try {
    var response = await axios.post('http://localhost:3001/file/upload', file, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    });
  } catch (error) {
  }
}
```

### 8.2.2. 파일 다운로드

```
import axios from 'axios';
```

### 8.2.3. 파일 검색

```
import axios from 'axios';

// 검색할 파일 정보
var filter = "";

try {
  var response = await axios.get('http://localhost:3001/file/search', {
    params: {
      filter: filter
    }
  });
} catch (error) {
}
```

## 9. 프로그램 실행 화면

### 9.1. 로그인 화면

### 9.2. 메인 화면

### 9.3. 계정 관리 화면

**9.3.1. 생성**

**9.3.2. 삭제**

**9.4. 데이터 관리 화면**

**9.4.1. 삽입**

**9.4.2. 삭제**

**9.5. 데이터 분석 화면**

**9.5.1. 필터링**

**9.5.2. 로그**