작업자 안전 위험 빅데이터 /AI 분석 웹 서비스 : 스마트 밴드, 헬멧, 벨트 센서 기반의 위험 행동 인식

1. 프로젝트 개요

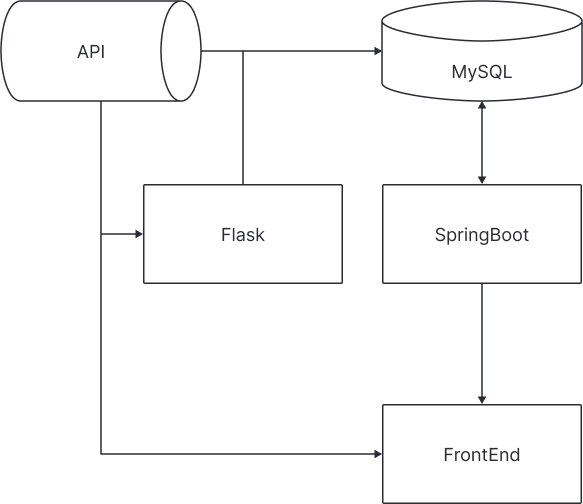
* 배경

건설 현장에서의 낙상 등 안전 사고를 미연에 방지하기 위해 이미 일어난 사고 데이터를 수집하여 사고가 발생할 가능성이 높은 상황을 예측하여 조치를 취할 수 있도록 AI 프로그램을 개발하고자 합니다.

* 목표
  + 스마트 밴드 또는 스마트 헬멧으로 데이터를 수집.
  + 수집된 생체 데이터를 통한 작업자 상태 관리.
  + 근로자의 Gyro sensor 데이터, 맥박, 체온을 통한 사고 특성 분석 및 예측.
  + 위험 예측을 활용해서 안전관리자에게 위험관리자 위치 및 상태 알림.
* 팀 : WorkerSafety

|  |  |
| --- | --- |
| 구민지 (팀장) | Front-end(React) |
| 김단우 | Data Analysis(+Flask) |
| 양철민 | Back-end(SpringBoot) |

* Project Structure



* 프로젝트 구현 환경

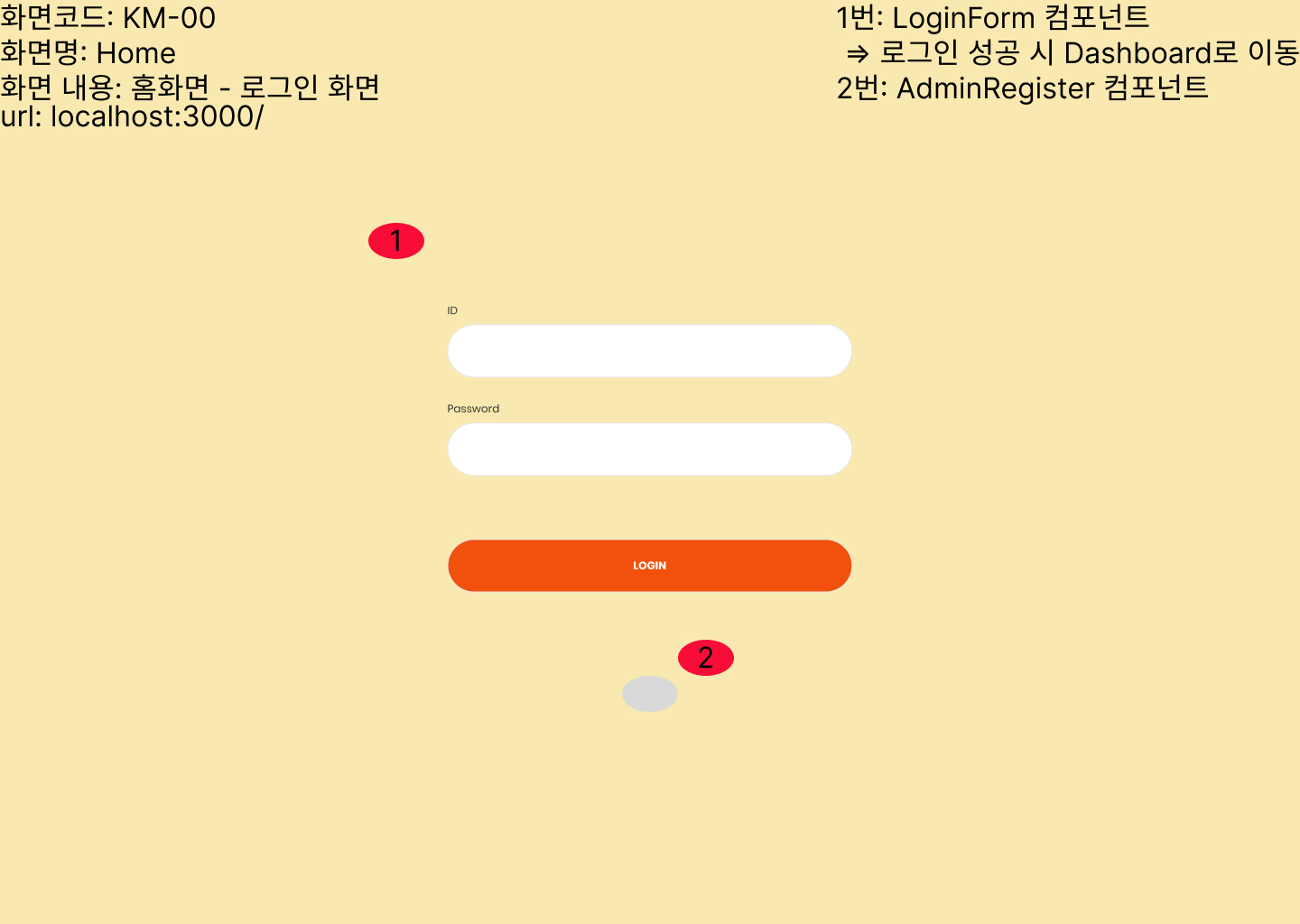
|  |  |  |  |
| --- | --- | --- | --- |
|  | 언어 및 프레임워크 | 개발 환경 | 라이브러리, API |
| DA | Python : 3.10.10  Flask : 2.3.2 | PyCharm | scikit-learn  Matplotlib  Pandas  NumPy |
| BE | Springboot : 3.0.6  Java : 17 | Eclipse | jpa |
| FE | Node.js : 18.13.0  JavaScript | VSCode | React : 18.2.0  react-router-dom : 6.11.0  axios : 1.4.0 |

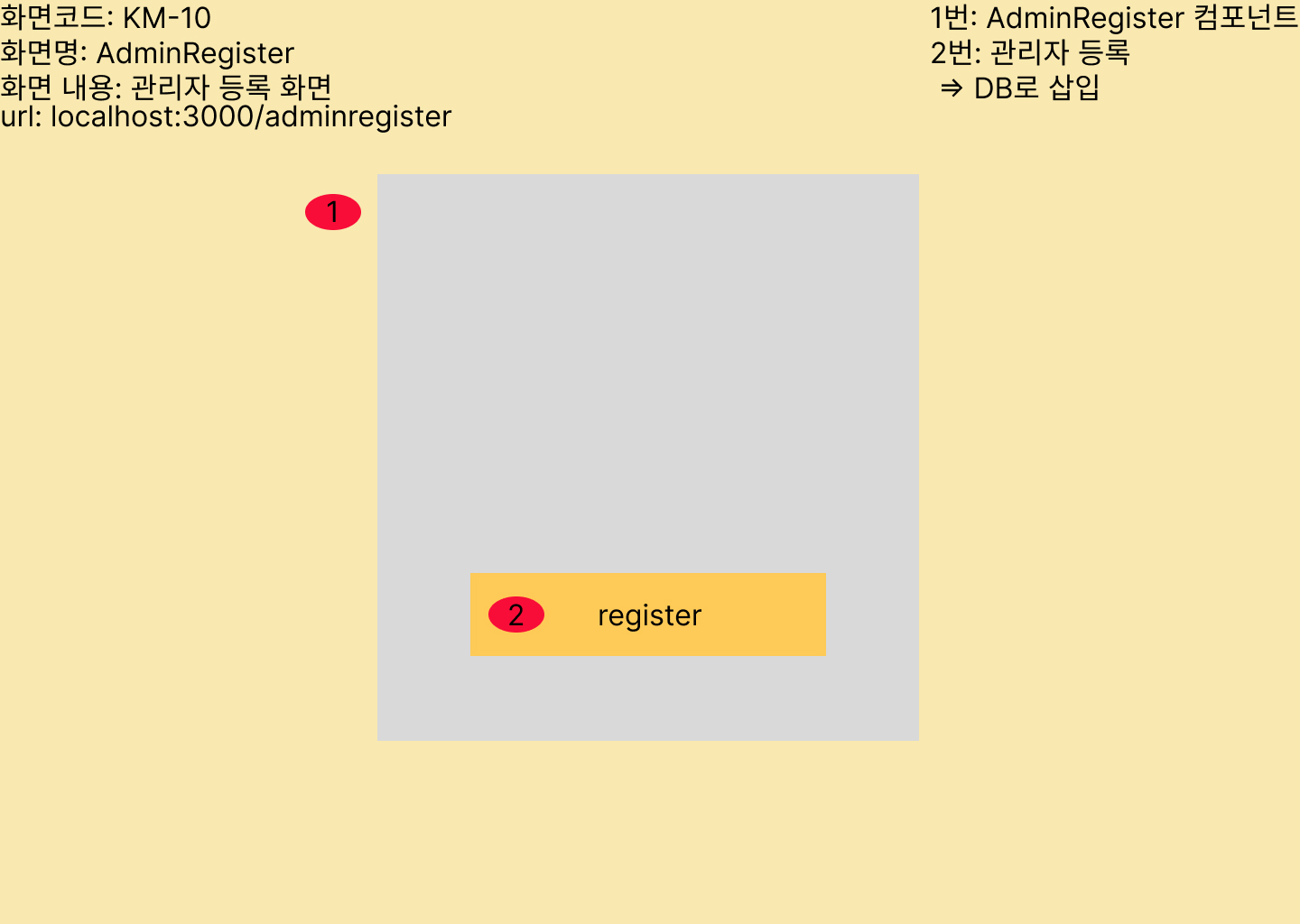
1. 요구 사항 명세

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 서비스 | ID | 요구사항명 | 요구사항 내용 | 날짜 | 버전 |
| Data Analysis | ws-001 | 작업자 상태 분석 및 예측 | 작업자의 이상 맥박 감지 | 05.18 | v0.5 |
| ws-002 | 작업자의 이상 체온 분석 | 05.18 | v0.5 |
| ws-003 | 작업자의 이상 Gyro sensor 분석 | 05.18 | v0.5 |
| ws-004 | 작업자 종합 데이터를 통한 사고 예측 | 05.18 | v0.5 |
| 로그인 | ws-005 | 로그인 | ID, Password Form 제공 | 05.18 | v0.5 |
| 지도창 | ws-006 | 작업자 위치 | GPS 데이터로 실시간 작업자 위치 표출 | 05.18 | v0.5 |
| ws-007 | 작업자 상태 | 아이콘을 통해 위험상태 표출 | 05.18 | v0.5 |
| ws-008 | 작업자 데이터 | 아이콘 선택 시 지도 아래에 상세 정보 표출 | 05.18 | v0.5 |
| 상세창 | ws-009 | 작업자 데이터 | 아이콘 선택 후 상세 보기 클릭 | 05.18 | v0.5 |
| 목록창 | ws-010 | 작업자 관리 | 작업자 목록 표출 | 05.18 | v0.5 |
| ws-011 | 작업자 추가 | 05.18 | v0.5 |
| ws-012 | 작업자 삭제 | 05.18 | v0.5 |
| ws-013 | 작업자 검색 | 05.18 | v0.5 |
| ws-014 | 작업자 정렬 | 05.18 | v0.5 |
| 생체 데이터 | ws-015 | 생체 데이터 표출 | 작업자 맥박 그래프 표출 | 05.18 | v0.5 |
| ws-016 | 맥박 데이터 위험 범위 시 알림 | 05.18 | v0.5 |
| ws-017 | 작업자 체온 그래프 표출 | 05.18 | v0.5 |
| ws-018 | 체온 데이터 위험 범위 시 알림 | 05.18 | v0.5 |
| 활동 데이터 | ws-019 | 활동 데이터 표출 | Gyro sensor 데이터 그래프로 표출 | 05.18 | v0.5 |
| ws-020 | Gyro sensor 데이터 위험 범위 시 알림 | 05.18 | v0.5 |

1. 화면설계 및 기능명세서

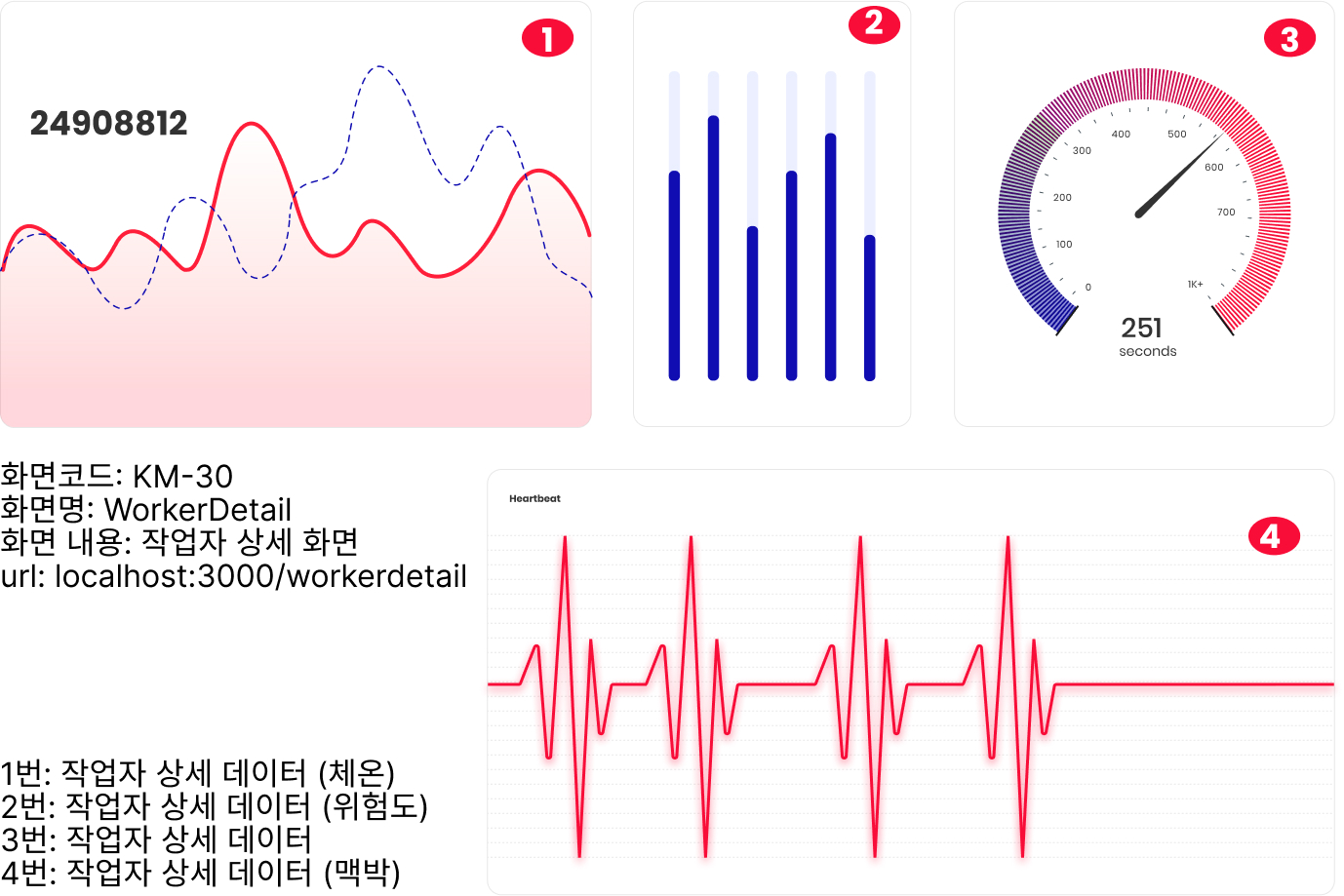
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 화면코드 | 화면명 | 화면 내용 | url | 버전 |
| KM-00 | Home | 홈화면: 로그인 화면 | localhost:3000/ | v0.5 |
| KM-10 | AdminRegister | 관리자 등록 화면 | localhost:3000/adminregister | v0.5 |
| KM-20 | Dashboard | 관리자 화면 | localhost:3000/dashboard | v0.5 |
| KM-21 | Dashboard | 관리자 화면  - 아이콘 클릭 시 | localhost:3000/dashboard | v0.5 |
| KM-30 | WorkerDetail | 작업자 상세 화면 | localhost:3000/workerdetail | v0.5 |









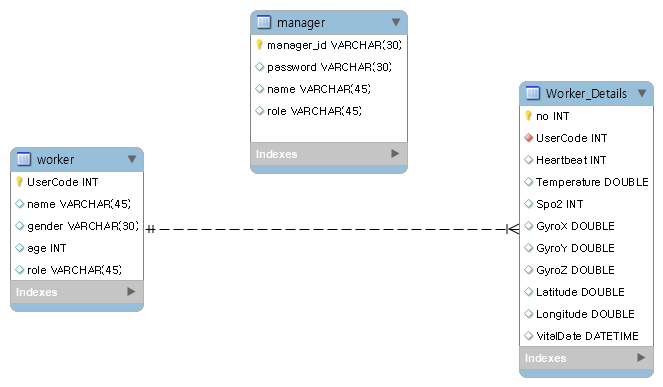


1. RestAPI 설계

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| /login |  |  |  |  |
| Index | Method | URI | Description | Response(JSON) |
| 1 | POST | /login | 로그인 | 로그인결과 |
|  |  |  |  |  |
|  |  |  |  |  |
| /login |  |  |  |  |
| Index | Method | URI | Description | Response(JSON) |
| 1 | PUT | /logout | 로그아웃 | 로그아웃 결과 |
|  |  |  |  |  |
|  |  |  |  |  |
| /user |  |  |  |  |
| Index | Method | URI | Description | Response(JSON) |
| 1 | POST | /join | 회원가입 | 회원가입 결과 |
| 2 | DELETE | /delete | 회원탈퇴 | 회원탈퇴 결과 |
| 3 | PUT | /update | 회원정보수정 | 회원정보수정 결과 |
|  |  |  |  |  |
|  |  |  |  |  |
| /worker |  |  |  |  |
| Index | Method | URI | Description | Response(JSON) |
| 1 | GET | /List | 작업자 조회 | 작업자 세부정보 결과 |
| 2 | POST | /add | 작업자 추가 | 작업자 추가결과 결과 |
| 3 | DELETE | /delete | 작업자 삭제 | 작업자 삭제결과 결과 |

1. DB 설계

- ERD



- DB 상세

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Type | Field | Remarks |
| manager | varchar | manager\_id | Primary Key |
|  | varchar | password |  |
|  | varchar | name |  |
|  | varchar | role |  |
| worker | integer | UserCode | Primary Key |
|  | varchar | name |  |
|  | varchar | gender |  |
|  | integer | age |  |
|  | varchar | role |  |
| worker\_Details | integer | no | Primary Key |
|  | integer | UserCode | Foreign Key(worker) |
|  | integer | Heartbeat | 심장박도 |
|  | double | Temperature | 체온 |
|  | integer | Spo2 | 산포도 |
|  | double | GyroX | 자이로센서x축데이터 |
|  | double | GyroY | 자이로센서x축데이터 |
|  | double | GyroZ | 자이로센서z축데이터 |
|  | double | Latitude | 위도 |
|  | double | Longitude | 경도 |
|  | datatime | VitalDate | 시간 |

1. 데이터 분석

* 데이터 특징 : 시계열 데이터
* 데이터 출처
  1. 강남앤인코누스 현장 수집 데이터
  2. Github data
     1. <https://github.com/laxmimerit/Human-Activity-Recognition-Using-Accelerometer-Data-and-CNN>
* 데이터 형태 : csv
* 데이터 내용 : 작업자별 Gyro sensor 데이터, 맥박, 체온, 위치, 시간
* 데이터 수집 및 전처리
  1. 건설 현장에서 근로자의 Gyro sensor 데이터, 맥박 등의 데이터를 수집합니다.
  2. 수집된 데이터를 전처리하여 이상치나 결측치 등을 제거하고, 필요한 경우 데이터를 정규화하는 등의 과정을 수행하여 데이터를 가공합니다.
* 가설설정
  1. 건설 현장에서는 근로자가 반복되는 작업을 하게 되는데, 이 과정에서 근로자의 자세나 움직임이 불규칙해지면 사고 발생 가능성이 높아질 것으로 가설을 설정합니다.
  2. 따라서, 근로자의 움직임 데이터와 맥박 변화를 사용하여 이상 감지 알고리즘을 구현하고, 일정 이상의 불규칙한 패턴이 발견될 경우 사고가 발생할 가능성이 높다는 가설을 설정합니다.
* 데이터 분석 및 모델링

본 프로젝트에서는 작업자 사고 예측을 위해 CNN(Convolutional Neural Network)과 LSTM(Long Short-Term Memory)을 활용한 분류모델을 구축합니다. 데이터 분석 과정은 다음과 같이 진행됩니다:

* + - 1. 데이터 수집: 작업자 활동, 사고와 관련된 데이터를 수집합니다. 이 데이터에는 작업자의 맥박과 체온 등 생체신호 데이터와 작업자의 Gyro sensor 데이터가 포함됩니다.
      2. CNN(Conv1D) 모델 생성: 맥박과 체온 데이터를 입력으로 사용하여 CNN(Conv1D) 모델을 생성합니다. 이 모델을 사용하여 예측 결과를 확인합니다. 그러나 이 모델은 학습 편향과 입력 데이터에 대한 일반화가 심하므로, 편향이 심한 후행 지표를 예측하는 결과를 얻게 될 것입니다.
      3. LSTM 조합 모델 생성: 성능 향상을 위해 LSTM 조합 모델을 생성합니다. CNN(Conv1D) 모델과 LSTM을 조합하여 더 정확한 예측 결과를 얻을 수 있습니다.

위의 분석 내용을 바탕으로 작업자 안전 예측 프로그램의 데이터 분석 결과를 작성할 수 있습니다. 이 결과에는 사용된 모델과 알고리즘, 예측 결과의 성능에 대한 정보가 포함됩니다.

* 모델 평가 및 보완

모델의 성능을 평가하기 위해 근로자의 실제 사고 데이터, 활동정보와 모델이 예측한 사고 데이터, 활동 정보를 비교하여 모델의 정확도를 평가하고, 모델의 성능을 향상시키기 위해 추가적인 데이터나 특성을 수집하여 모델을 보완합니다.

1. 일정 계획

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1주 | 2주 | 3주 | 4주 | 5주 | 6주 | 7주 | 8주 | 9주 | 10주 |
| DA | D1 | D2 | | D3 | | D4 | D5 | D6 | D7 | |
| BE | B1 | B2 | | B3 | | B4 | B5 | B6 | B7 | |
| FE | F1 | F2 | | F3 | | F4 | F5 | F6 | F7 | |

* DA
* D1 : 계획
* D2 : 데이터 전처리 & 머신러닝 기반 이상 감지 알고리즘 구현
* D3 : 모델 개선
* D4 : 플라스크를 활용한 API 작성
* D5 : 신경망 기반 이상 감지 알고리즘 구현 (딥러닝, 텐서플로)
* D6 : 모델 및 주요 서비스 개선
* D7 : 테스트, 보완 및 발표
* BE
* B1 : 계획
* B2 : 모델 호출해서 REST API 생성
* B3 : 스키마 기타 API 확정
* B4 : 1차 구현, 개발 환경 구축, 로그인 시큐리티, 관리자 테이블, 각 서버 데이터 연결
* B5 : 테스트 및 피드백
* B6 : 2차 구현- 작업자 테이블, 각 서버의 요구·요청에 따른 API 작성, 데이터 저장
* B7 : 테스트, 보완 및 발표
* FE
* F1 : 계획
* F2 : UI/UX 설계 및 시각화
* F3 : UI/UX 확정
* F4 : 1차 구현 - 컴포넌트별 구현, 디자인 레이아웃 구현
* F5 : 테스트 및 피드백
* F6 : 2차 구현 - BE와 연결, 페이지 구현 및 완성
* F7 : 테스트, 보완 및 발표