

개발완료 보고서

제출일 : 2024년 7월 27일

	참여인원													
참여자	유 시 온													
개발프로젝트 소개														
프로젝트 명	“HealthGuard : 건강 위험도 예측 시스템”													
활동일시	7/15 ~ 7/27				장소			광주인력개발원 공학1관 2층 드론융합실						
주요주제	빅데이터, 머신러닝을 활용한 개인 건강 위험도 예측 프로그램 구현													
개발목적	빅데이터를 활용한, 머신러닝 기법을 통해 검진 결과 입력 시 뇌·심혈관, 대사증후군, 고혈압, 당뇨 위험도를 예측하는 프로그램 개발													
개발환경	Windows 10 Home / Visual Studio 2022 / C# / WPF / Python Ubuntu 16.40 Linux OS / Visual Studio Code / Maria DB													
일 정 표	항 목		07/15	07/16	07/17	07/18	07/19	07/20	07/22	07/23	07/24	07/25	07/26	07/27
	계획	주제 선정 및 데이터 수집												
		머신러닝, 딥러닝 학습												
		개발계획서 및 일정표작성												
	구현	DB 생성 및 연동												
		C++ 서버 구현												
		회원가입, 로그인 기능												
		데이터 활용 AI 모델 학습												
		트레이닝, 테스트 데이터 활용 AI 모델학습												
		C#(WPF) 클라이언트 구현												
	제출	완료보고서, PPT 작성 및 제출												

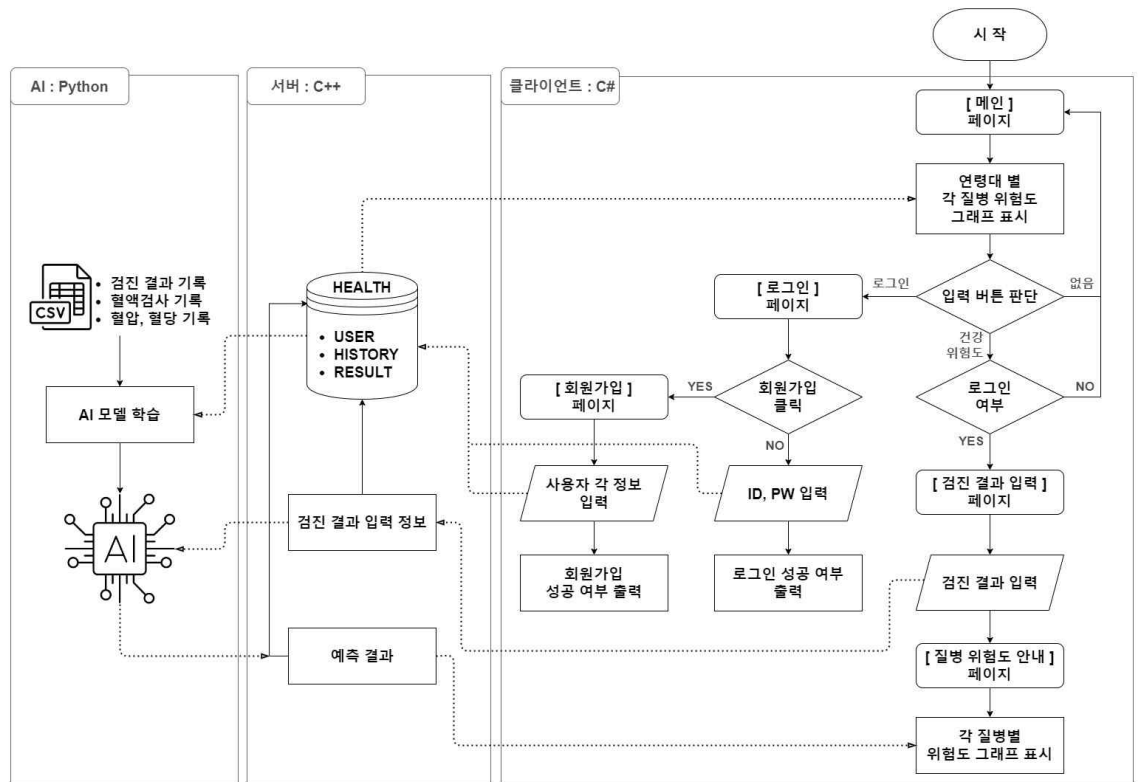
요구
분석서

분 류	요구 분석 내용		
	발생 조건 세부 분류	세 부 내 용	
AI 파트	AI모델 학습	검진결과 기록, 혈액검사, 대사증후군 CSV파일을 활용하여 AI모델 학습 진행	
	트레이닝 학습	C++ 서버에서 새로운 트레이닝 데이터를 받아와 AI 모델 학습 진행	
	AI모델 테스트	C++ 서버에서 새로운 테스트 데이터를 받아와 AI 모델 테스트를 진행	
서버 파트	회원가입	DB안에서 전송 받은 ID 중복확인 후 결과 클라이언트로 전송, 중복되지 않은 경우 회원정보 DB에 저장	
	로그인	DB안에서 전송 받은 ID, PW 일치 여부 확인 후 결과 클라이언트로 전송	
	건강 위험도 예측	AI파트 데이터 전송	전송받은 건강검진 결과를 AI파트로 전송한 후, 건강 위험도 예측 결과를 받음
		예측 결과 전송	건강 위험도 예측결과를 클라이언트로 전송
		DB 저장	전송받은 건강검진 결과, 건강 위험도 예측 결과를 DB에 저장
클라이언트 파트	메인	통계 그래프	메인 페이지에 연령대 별 각 질환 별 위험도 통계 그래프 표시
		건강 위험도	'건강 위험도' 버튼 클릭 시 로그인 여부 확인 후 검진 결과 기록 입력 페이지로 이동
		로그인	'로그인' 버튼 클릭 시 로그인 페이지로 이동
	로그인	ID, PW 입력	ID, PW 입력 후 '로그인' 버튼 클릭 시 서버로 해당 정보 전송 후 로그인 결과 받음
		회원가입	'회원가입 버튼 클릭 시 회원가입 페이지로 이동
	검진결과 입력		사용자의 검진 결과 입력 후 '확인' 버튼 클릭시 서버로 해당 정보 전송 후 건강 위험도 페이지로 이동
	건강 위험도 안내		서버에서 전송받은 데이터로 뇌심혈관, 고혈압, 당뇨, 대사증후군의 위험도를 수치화, 그래프로 표시

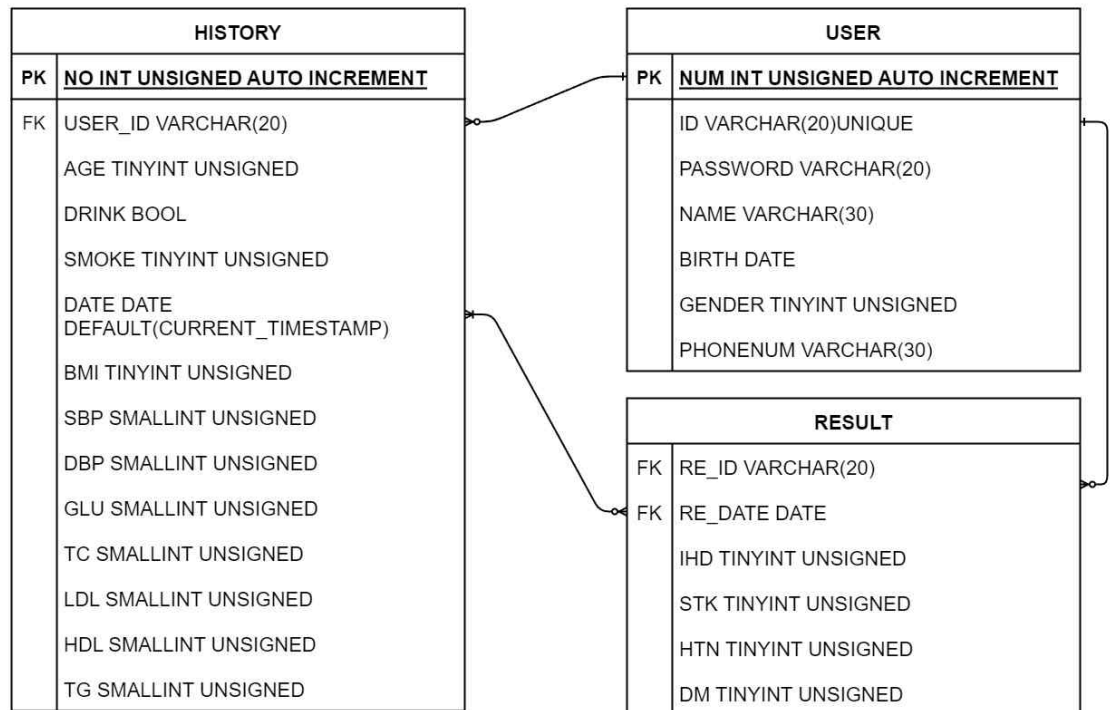
목업 이미지



순서도



ERD



테이블
명세서

데이터 베이스명	HEALTH		테이블명	USER	
컬럼ID	타입 및 길이	KEY	NULL?	DEFAULT	COMMENTS
NUM	INT UNSIGNED	PK	N	AUTO_INCREMENT	
ID	VARCHAR(20)	UK	N		아이디
PASSWORD	VARCHAR(20)		N		비밀번호
NAME	VARCHAR(30)		N		이름
BIRTH	DATE		N		생년월일
GENDER	TINYINT UNSIGNED		N		성별
PHONENUM	VARCHAR(30)		Y		전화번호

데이터 베이스명	HEALTH		테이블명	HISTORY	
컬럼ID	타입 및 길이	KEY	NULL?	DEFAULT	COMMENTS
NO	INT UNSIGNED	PK	N	AUTO_INCREMENT	
USER_ID	VARCHAR(20)	FK	N		아이디
AGE	TINYINT UNSIGNED		Y		나이
SMOKE	TINYINT UNSIGNED		N		담배
DRINK	BOOL		N		술
DATE	DATE		N	CURRENT_TIMESTAMP	입력 날짜
BMI	TINYINT UNSIGNED		N		비만도
SBP	SMALLINT UNSIGNED		N		수축기혈압
DBP	SMALLINT UNSIGNED		N		이완기혈압
GLU	SMALLINT UNSIGNED		N		공복혈당
TC	SMALLINT UNSIGNED		Y		총 콜레스테롤
LDL	SMALLINT UNSIGNED		Y		저밀도콜레스테롤
HDL	SMALLINT UNSIGNED		Y		고밀도콜레스테롤
TG	SMALLINT UNSIGNED		Y		중성지방

데이터 베이스명	HEALTH		테이블명	RESULT	
컬럼ID	타입 및 길이	KEY	NULL?	DEFAULT	COMMENTS
RE_ID	VARCHAR(20)	FK	N		아이디
RE_DATE	TIMESTAMP	FK	N		입력 날짜
STK	TINYINT UNSIGNED		N		뇌혈관 위험도
IHD	TINYINT UNSIGNED		N		심혈관 위험도
HTN	TINYINT UNSIGNED		N		고혈압
DM	TINYINT UNSIGNED		N		당뇨 위험도

주요 함수설명

서버 함수 (C++)	
class DB	
sql::Connection* Connect	mariaDB와 Visual Studio Code와 연결하는 함수
void Disconnect	mariaDB와 Visual Studio Code와 연결 끊는 함수
class Client	
void ID_Check	해당 ID DB 유저 테이블 안에 중복되어 있는지 확인 후, 결과를 클라이언트로 전송하는 함수
void Login	ID, PW가 DB 유저 테이블에 일치하는지 확인 후, 결과를 클라이언트로 전송하는 함수
void Join	사용자 정보를 DB 유저 테이블에 저장한 후, 클라이언트로 회원가입 완료 메시지 전송하는 함수
void CheckItem_Value	사용자 검사 데이터를 AI 파트로 전송한 후, 예측값을 받아 클라이언트로 예측값을 전송하는 함수
void InsertDB_CheckItem	사용자 검사 데이터를 DB 'HISTORY' 테이블에 저장하는 함수
void Load_History	DB 'HISTORY' 테이블에서 사용자 기록을 조회하여 클라이언트로 전송하는 함수
void Search_Result	DB 'RESULT' 테이블에서 사용자가 요청한 위험도 결과 기록을 클라이언트로 전송하는 함수
class AI	
int ConnectPython	Python으로 연결하여 소켓을 반환하는 함수
RESULT Predict_Value	Python에서 예측값을 전송받아 RESULT 구조체에 대입한 후, 그 구조체를 반환하는 함수
void InsertDB_PredictValue	예측값을 DB 'RESULT' 테이블에 저장하는 함수
AI파트 함수 (Python)	
handler	클라이언트와 메시지를 송 수신하는 함수
read_json	수신받은 json 메시지를 읽고 numpy 배열로 변환하여 model에 사용할 배열을 반환하는 함수
read_model	저장된 모델을 읽어들인 다음 그 해당 모델을 반환하는 함수
predict_value	로드된 모델을 사용하여 주어진 데이터로 예측을 수행한 다음 질환 위험도에 대한 결과를 반환하는 함수
reference_value	혈압 기준치를 참고하여 혈압에 대한 위험도를 측정하여 반환하는 함수

<div> <div>주요</div> <div>함수설명</div> </div>	클라이언트 함수 (C#, WPF)		
	MAIN	private void btnm_Login_Click	‘로그인&회원가입’ 버튼 클릭 시 로그인 페이지로 이동
		private void btnm_MyPage_Click	‘My Page’ 버튼 클릭 시 로그인 여부 확인 후 사용자 건강 데이터 기록 페이지로 이동
		private void btnm_Health_Click	‘나의 건강 예측하기’ 버튼 클릭 시 로그인 여부 확인 후 건강 데이터 입력 페이지로 이동
	LOGIN	private void btnl_Join_Click	‘회원가입’ 버튼 클릭 시 회원가입 페이지로 이동
		private void btnl_Login_Click	‘로그인’ 버튼 클릭 시 로그인 페이지로 이동
		private void btnl_back_Click	‘뒤로가기’ 버튼 클릭 시 메인 페이지로 이동
	JOIN	private void btnj_IDcheck_Click	‘중복 확인’ 버튼 클릭 시 서버로 ID를 전송하여 ID 중복 확인하는 함수
		private void btnj_Join_Click	‘회원가입’ 버튼 클릭 시 입력한 사용자 정보를 서버로 전송하여 회원가입하는 함수
		private void btnj_back_Click	‘뒤로가기’ 버튼 클릭 시 메인 페이지로 이동
		public void UserList_value	textbox에 입력된 정보를 USER 클래스 리스트에 대입하는 함수
	MY PAGE	private void btnm_Main_Click	‘메인으로’ 버튼 클릭 시 메인 페이지로 이동
		public void init_MyPage_UserValue	페이지 상단 Label에 사용자 이름, 연령, 성별을 바인딩하는 함수
		public void init_HistoryList	서버에서 받은 사용자 데이터 기록을 리스트뷰에 바인딩하는 함수
		public void Select_HistoryLog	리스트뷰에서 선택한 데이터에 대한 결과를 서버에 요청한 후 건강 위험도 페이지로 이동하는 함수
	HEALTH DATA	private void btnd_Predict_Click	‘위험도 확인’ 버튼 클릭 시 입력한 정보를 서버로 전송한 후, 결과 페이지로 이동하는 함수
		private void btnd_back_Click	‘뒤로가기’ 버튼 클릭 시 메인 페이지로 이동
		public void checkItem_value	textbox에 입력된 정보를 CHECK_ITEM 클래스 리스트에 대입하는 함수
	HEALTH PREDICT	private void btnp_Main_Click	‘메인으로’ 버튼 클릭 시 메인 페이지로 이동
		public void init_Predict_UserValue	페이지 상단 Label에 사용자 이름, 연령, 성별을 바인딩하는 함수
		public void LoadChart	심장질환, 뇌질환, 고혈압, 당뇨 위험도에 대한 예측값을 그래프에 표시하는 함수

<div>주요 함수설명</div>	클라이언트 함수 (C#, WPF)	
	PROCESS	private int Connect 서버와 연결하는 함수
		private void Disconnect 서버와 연결 끊는 함수
		public string ReadMsg 서버에서 전송한 메시지를 읽는 함수
		public void SendMsg 메시지를 서버에 보내는 함수
		public int Login ID, PW를 서버로 전송한 후 로그인 결과를 받는 함수
		public int ID_Matching ID 서버로 전송한 후 ID 중복 결과를 받는 함수
		public int Join 사용자 정보를 서버로 전송한 후 회원가입 결과를 받는 함수
		public int Check_Item 사용자가 입력한 검사 데이터를 서버로 전송한 후 질환 위험도에 대한 예측값을 받는 함수
		public int Load_History 사용자 기록을 서버에 요청한 후 데이터 기록을 받는 함수
		public string Today 오늘 날짜를 계산하는 함수
		public int BMI 사용자 키, 몸무게 입력 시 비만도 계산하는 함수
		public void SetAge 사용자 생년월일 입력 시 나이 계산하는 함수
<div>후 기</div>	<p>이번 프로젝트에서 텐서플로우(TensorFlow) 라이브러리를 사용하여 건강 검진 데이터 결과를 학습시키고, 당뇨 및 뇌·심혈관 질환에 대한 위험도를 예측하는 모델을 생성했습니다. 딥러닝을 학습하면서 주제에 따라 사용하는 모델, 활성화 함수 등이 각각 다르다는 것을 알게 되었습니다.</p> <p>특히, 과적합 문제에 대해 신경을 썼습니다.. 과적합을 방지하기 위해 여러 방법이 있었는데 이 방법 또한 너무 많이 사용할 시 오히려 과적합이 발생할 수 있어 지속적으로 변경하면서 확인이 필요했습니다. 또한, 적절한 데이터를 찾는 것도 매우 중요하다는 것을 느꼈습니다. 주제를 정하고 데이터를 사용하여 모델을 학습시킬 때 예측값이 잘 나오지 않아서 데이터를 다시 확인해 보니 사용하기에 적합하지 않은 데이터여서 중간에 데이터를 변경하기도 했습니다.</p> <p>비록 이번 처음 제가 계획한 대로 프로젝트를 완료했다고 할 수 없지만, 딥러닝에서 주의해야 할 부분들을 배우고 기본적인 개념, 코드적인 이해를 높이는 시간이 되었습니다. 딥러닝 AI 부분은 처음에는 어렵게 느껴졌지만, 이번 프로젝트를 통해 조금씩 다뤄볼 수 있었습니다.</p>	