

개발완료 보고서

제출일 : 2024년 08월 03일

	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	<table><tr><th colspan="2">항 목</th><th>07/15</th><th>07/16</th><th>07/17</th><th>07/18</th><th>07/19</th><th>07/20</th><th>07/22</th><th>07/23</th><th>07/24</th><th>07/25</th><th>07/26</th><th>07/27</th></tr><tr><td rowspan="3">계획</td><td>주제 선정 및 데이터 수집</td><td colspan="4"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>머신러닝, 딥러닝 학습</td><td colspan="11"></td><td></td></tr><tr><td>개발계획서 및 일정표작성</td><td colspan="3"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td rowspan="7">구현</td><td>DB 생성 및 연동</td><td></td><td colspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>C++ 서버 구현</td><td></td><td></td><td colspan="3"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>회원가입, 로그인 기능</td><td></td><td></td><td colspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>데이터 활용 AI 모델 학습</td><td></td><td></td><td colspan="9"></td><td></td></tr><tr><td>트레이닝, 테스트 데이터 활용 AI 모델학습</td><td></td><td></td><td colspan="9"></td><td></td></tr><tr><td>C#(WPF) 클라이언트 구현</td><td></td><td></td><td></td><td></td><td colspan="6"></td><td></td></tr><tr><td>제출</td><td>완료보고서, PPT 작성 및 제출</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3"></td></tr></table>													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												
	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																																																																																																																																																													

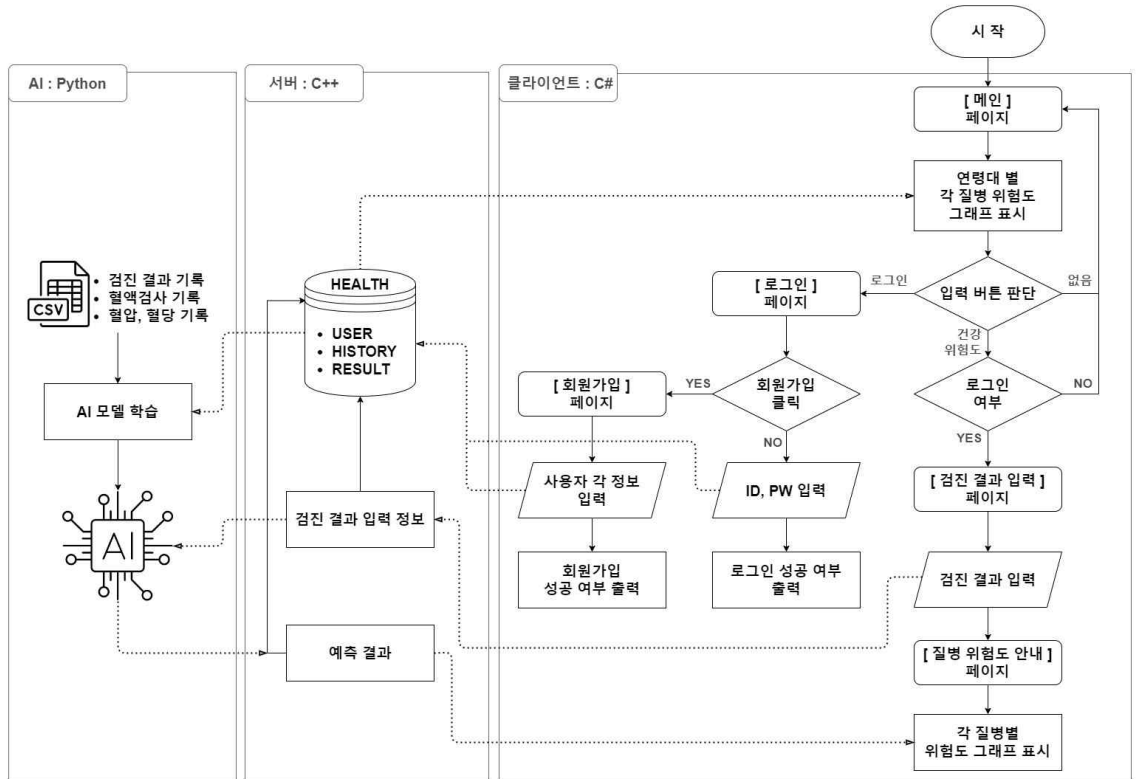
요구
분석서

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

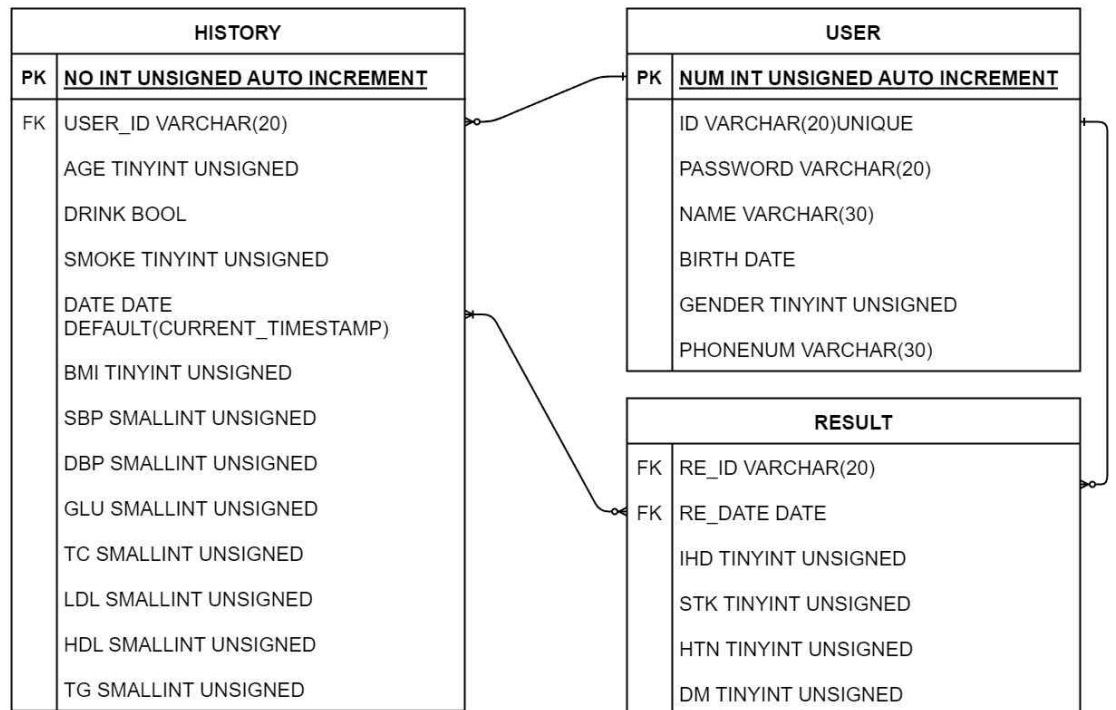
목업 이미지



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 ID 중복검사 함수

void Login 로그인 시 DB 조회하여 로그인 결과 판단 함수

void Join 회원가입 정보 DB 저장 함수

void InsertDB_Hitsory 미술품 작가 조회 기록 DB 저장 함수

void Artist_Info 화가 목록의 각 화가 정보 출력 함수

void Search_Artist 미술품 작가 정보 출력 함수

void InsertDB_PredictValue 예측값을 DB 'RESULT' 테이블에 저장하는 함수

AI파트 함수 (Python)

read_label CSV 파일 읽어오는 함수

read_images 이미지 데이터 읽어오는 함수

predictPainter C#에서 받아온 미술품 이미지로 화가를 찾는 함수

recvall 데이터를 완전히 다 읽어오는 함수

handler 클라이언트와 네트워크 통신하는 함수

<div> <div>주요</div> <div>함수설명</div> </div>	클라이언트 함수 (C#, WPF)	
	MAIN	private void btn_painter_Click 화가 목록 페이지 이동 함수
		private void btn_login_Click 로그인 & 회원가입 페이지 이동 함수
		private void btn_search_Click 미술품 화가 정보 조회 페이지 이동 함수
		private void btn_history_Click 조회 기록 페이지 이동 함수
	LOGIN	private void Init_LoinInfo() 로그인 시 ID, PW 정보 DB 저장 함수
		private void btn_loginCheck_Click ID, PW 일치 여부 확인하여 로그인 결과 출력 함수
		private void btn_join_Click 회원가입 진행 함수
	JOIN	private void btn_idcheck_Click ID 중복검사 함수
		private void Init_JoinInfo() 회원가입 정보 DB 저장 함수
		private void btn_joinCheck_Click 회원가입 완료 결과 출력 함수
	Paint	private void Init_ArtistList 화가 목록 리스트뷰 출력 함수
	Search	public void StartCamera 카메라 화면 표시 함수
		private void btn_result_Click 촬영 이미지의 화가 및 정보 조회 함수
		private void Init_ArtInfo 화가 및 정보 출력 함수
	Ask	private void btn_send_Click 메일 전송 후 메시지창 출력 함수
		private void Init_EmailInfo 유저 입력 정보 저장 함수

주요 함수설명	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 User_Info 로그인, 회원가입시 서버로 유저 정보 보내는 함수
		public int ID_Check ID 서버로 전송한 후 ID 중복 결과를 받는 함수
		public int ArtInfo 서버에서 화가 정보를 받는 함수
		public int ReadArtistInfo 프로그램에서 제공하는 화가 정보를 받아오는 함수
		public int Send_Image 파이썬으로 미술품 이미지 전송하여 화가이름 받는 함수
		public BitmapImage ConvertBitmapSourceToBitmapImage 카메라에 인식되는 프레임을 비트맵이미지로 변환하는 함수
후 기	김해빈	<p>이번 프로젝트는 이미지를 전송하여 분류 및 예측하는 CNN 활용 AI 솔루션 프로젝트였습니다. 지난번 프로젝트에서 큰 도움을 주었던 시온이와 같은 팀이 되어 매우 든든했습니다. 여러 주제를 찾다가 이미지를 판별하기에 좋을 것 같은 미술작품으로 주제를 선정했고, 작가마다 화풍이 다르기에 이에 따라 이미지를 판별하는 방식으로 프로젝트를 진행했습니다. 미술작품 데이터가 매우 많아 이를 골라내는 작업이 우선이었습니다. 분류한 데이터로 모델을 생성하고 학습시킬 때 정확도가 높지 않아 고전했으나, 이미지 증강 및 다양한 방법을 시도하여 정확도가 증가하였습니다. TCP/IP 통신 방식에서 JSON을 이용해 데이터를 주고받는 법을 배울 수 있었습니다. 팀장인 시온이가 여러 중요한 부분을 맡아 훌륭히 수행해준 덕에 기한 내에 결과물을 제출할 수 있었습니다. 또한 코드 작성 중에 헛갈리는 부분이 있으면 친절하고 자세하게 설명해준 덕에 명확히 이해할 수 있었고 어떤 구조로 로직이 진행되는지 파악할 수 있었습니다. 좋은 팀장을 만나 무사히 프로젝트를 마무리할 수 있음에 고마울 따름입니다. 다만 이 프로젝트에서 이해한 부분들을 앞으로 잘 활용하기 위해서는 코드 전문을 살펴보며 복습하는 시간이 필요하겠습니다.</p>

후 기

유시온

이번 프로젝트에서는 각 화가들의 미술품을 학습시켜 사용자가 미술품을 보여주면 해당 화가의 정보를 알려주는 프로그램을 구현했습니다. 이전 주와는 달리 이미지 기반으로 진행하다 보니 더 쉽게 진행될 줄 알았으나, 실제로는 학습 과정이 예상보다 어려웠습니다.

이미지 데이터를 학습시키다 보니 메모리 사용량이 많아지는 제약이 있었고, 서버와 클라이언트 구현은 비교적 수월했지만, AI 모델을 학습시키는 데 많은 시간이 소모되었습니다. 또한, 정확도가 기대만큼 오르지 않아 미술품을 보여주고 예측할 때 제대로 작동하지 않는 경우도 있었습니다. 저희 팀의 주제는 깊이 있는 학습이 필요했기 때문에 층을 많이 구성해야 했고, 미술품 데이터가 중복되지 않다 보니 데이터 증강이 필요했습니다.

이러한 문제를 수정하고 학습을 진행하면서 정확도가 향상되었고, 저희가 원하는 결과를 얻을 수 있었습니다. 비록 미흡한 부분과 힘든 과정이 있었지만, 팀원 혜빈 언니와 함께 코드에 대해 서로 이야기하고 수정하면서 재미있게 진행할 수 있었습니다.

다음에는 저희 주제에 맞는 모델과 구성 방법에 대해 더 깊이 연구하고 계획한 후에 프로젝트를 진행해야 할 것 같습니다.