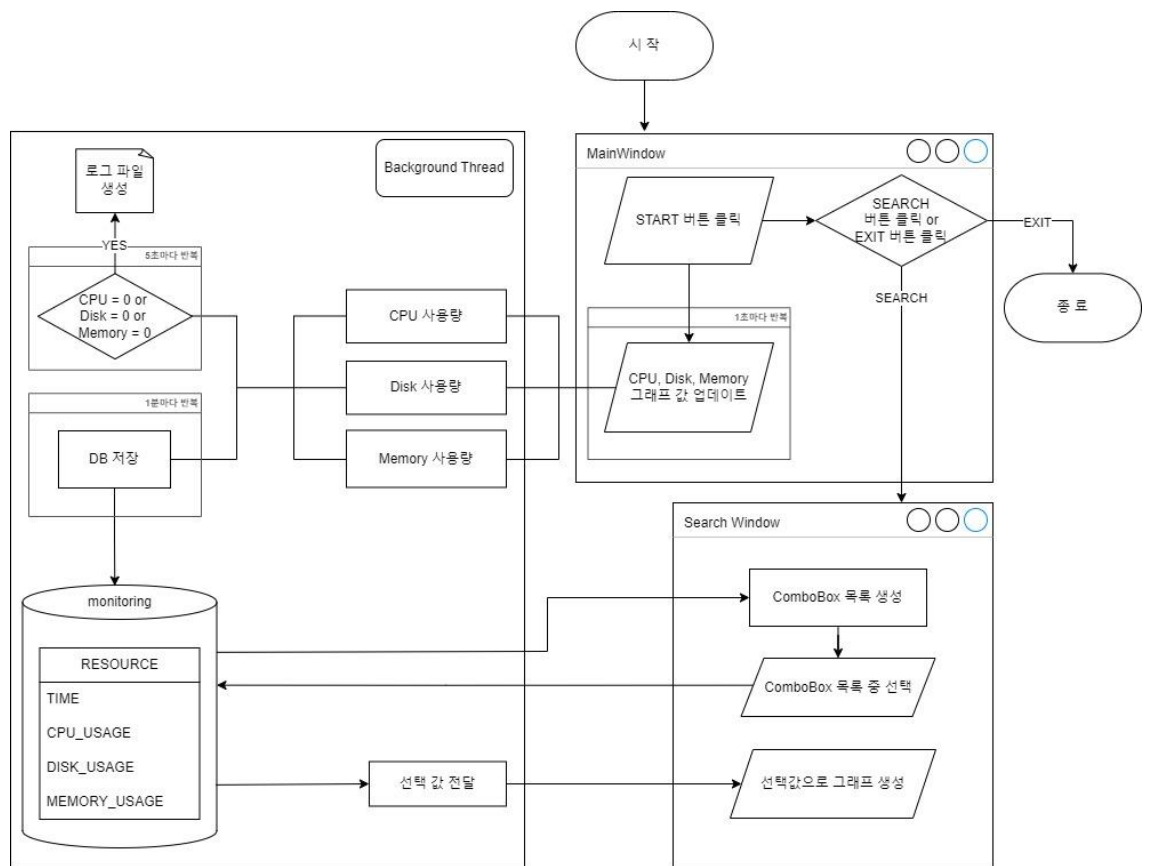


개발완료 보고서

	참여인원																																																																																																																																																						
참여자	유 시 온																																																																																																																																																						
개발프로젝트 소개																																																																																																																																																							
프로젝트 명	"Asynchronous_Resource Monitoring Tool"																																																																																																																																																						
활동일시	06 / 03 ~ 06 / 08				장소			광주인력개발원 공학1관 2층 드론융합실																																																																																																																																															
주요주제	C#, WPF, mariaDB를 활용한 “실시간(비동기) 컴퓨터 자원 모니터링 프로그램” 구현																																																																																																																																																						
개발목적	스레드, 람다식, DB연동에 대한 학습 후 C#, WPF, mariaDB를 활용하여 실시간 컴퓨터 자원 모니터링 프로그램 개발																																																																																																																																																						
개발환경	Windows 10Home / Visual Studio 2022 / C# / WPF / mariaDB																																																																																																																																																						
일 정 표	<table><tr><th colspan="2">항 목</th><th colspan="2">06/03</th><th colspan="2">06/04</th><th colspan="2">06/05</th><th colspan="2">06/06</th><th colspan="2">06/07</th><th colspan="2">06/08</th></tr><tr><td colspan="2">개발계획서 및 일정표 작성</td><td colspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td colspan="2">해당 정보 학습</td><td colspan="2"></td><td colspan="2"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td rowspan="4">필수 구현</td><td>리소스 모니터링 정보 수집</td><td></td><td></td><td colspan="4"></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Visual Studio, DB 연동</td><td></td><td></td><td colspan="4"></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>DB 생성 및 저장</td><td></td><td></td><td></td><td></td><td colspan="7"></td><td></td></tr><tr><td>GUI 시각화</td><td></td><td></td><td></td><td></td><td colspan="7"></td><td></td></tr><tr><td rowspan="2">추가 구현</td><td>정보 조회 기능</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5"></td><td></td></tr><tr><td>정보 로그 파일 생성</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5"></td><td></td></tr><tr><td>제출</td><td colspan="2">완료보고서 작성 및 제출</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2"></td></tr></table>												항 목		06/03		06/04		06/05		06/06		06/07		06/08		개발계획서 및 일정표 작성														해당 정보 학습														필수 구현	리소스 모니터링 정보 수집													Visual Studio, DB 연동													DB 생성 및 저장													GUI 시각화													추가 구현	정보 조회 기능														정보 로그 파일 생성														제출	완료보고서 작성 및 제출													
	항 목		06/03		06/04		06/05		06/06		06/07		06/08																																																																																																																																										
	개발계획서 및 일정표 작성																																																																																																																																																						
	해당 정보 학습																																																																																																																																																						
	필수 구현	리소스 모니터링 정보 수집																																																																																																																																																					
		Visual Studio, DB 연동																																																																																																																																																					
		DB 생성 및 저장																																																																																																																																																					
		GUI 시각화																																																																																																																																																					
	추가 구현	정보 조회 기능																																																																																																																																																					
정보 로그 파일 생성																																																																																																																																																							
제출	완료보고서 작성 및 제출																																																																																																																																																						

순서도



요 구 분 석 서

분 류	요 구 분 석 내 용	
	세 부 분 류	세 부 내 용
GUI	메 인 화 면	C# 라이브러리 'Livecharts'의 그래프 종류 중 'Gauge' 그래프를 사용하여 cpu, 메모리, 디스크 사용량을 실시간으로 표시 'CartesianChart' 그래프를 사용하여 cpu 사용량을 실시간으로 표시
		'Button'을 사용하여 [START] : 실시간 모니터링 시작 [SEARCH] : 과거 데이터 조회 창 띄우기 , [EXIT] : 종료 버튼 생성
	조 회 화 면	'ComboBox'를 사용하여 조회 가능한 날짜 리스트 출력
		'Button'을 사용하여 [SEARCH] 버튼 생성.
		'Livecharts' 그래프 중 'CartesianChart'를 사용하여 선택한 날짜에 조회된 결과를 그래프로 생성
	기능	
	리소스 모니터링	'PerformanceCounter', 'DriveInfo'를 사용하여 CPU, 메모리, 디스크 사용량 정보를 가져온다.
	DB저장	C# 라이브러리 'mysql'를 사용하여 DB와 연결을 통해 리소스 모니터링 정보를 쿼리문을 사용하여 DB에 저장
	정보 조회	ComboBox 목록에서 선택한 값을 가져와 DB 쿼리문 조건문으로 사용하여 해당 시간대의 리소스 모니터링 정보를 'CartesianChart' 그래프로 생성
	로그파일 생성	CPU, 메모리, 디스크에서 '0' 값이 나온 경우 .txt 파일 생성 후 날짜, 시간, 리소스 모니터링 정보를 작성하여 저장

코드 설명

```
private System.Timers.Timer DB_timer
// DB정보 업데이트 적용 타이머

private System.Timers.Timer Error_timer
// 오류파일 생성시 적용 타이머

private CPULineChart cpuchartUpdate
// CPU 'CartesianChart'값 실시간 업데이트 위해 클래스 인스턴스

private static double cpuValue // CPU 사용량 변수

private static double memValue // Memory 사용량 변수

private static double diskValue // Disk 사용량 변수

public MainWindow()
// MainWindow 클래스 생성자, DBthread 생성한 후 백그라운드 스레드로 생성

public void UI_Start
// CPU그래프, 메모리 그래프, 디스크 그래프값을 업데이트 하는 함수를 모은 함수

public void CpuChartValue
// Task를 사용하여 비동기로 작동하도록 하며, Dispatcher.BeginInvoke 안에서
    CPU그래프 값을 1초 마다업데이트하는 함수

public void MemoryChartValue
// Task를 사용하여 비동기로 작동하도록 하며, Dispatcher.BeginInvoke 안에서
    Memory 그래프 값을 1초 마다 업데이트하는 함수

public void DiskChartValue
// Task를 사용하여 비동기로 작동하도록 하며, Dispatcher.BeginInvoke 안에서
    Disk 그래프 값을 1초 마다 업데이트하는 함수

public static double GetCpuValue
// PerformanceCounter을 사용하여 CPU 사용량을 리턴받는 함수

public static double GetMemValue
// PerformanceCounter을 사용하여 Memory 사용량을 리턴받는 함수

public static double GetDiskValue
// DriveInfo를 사용하여 Disk 사용량을 리턴받는 함수

public void BackThread
// DB에 정보 입력하는 함수, 오류파일 생성하는 함수에 System.Timers를 적용하여
    각 함수에 실행 타임을 적용한 함수

public void InsertDB
// CPU 사용량, 메모리 사용량, 디스크 사용량을 DB에 연결하여 INSERT문으로
    RESOURCE 테이블에 시간, 각 값들을 입력하는 함수

public void ErrorCheck
// 현재 CPU사용량, 메모리사용량, 디스크사용량 중 0값이 있는 지 판단 후 있다면
    오류파일 생성하는 함수 호출하는 함수

public void ErrorFileCreate
// .txt 파일 생성 후 현재시간, 현재 CPU사용량, 메모리사용량, 디스크사용량을
    작성하는 함수
```

코드 설명	private void btn_start_Click // START 버튼을 누르면 리소스 모니터링을 시작하는 함수
	private void btn_search_Click // SEARCH 버튼을 누르면 과거 결과 조회 창을 띄우는 함수
	private void btn_exit_Click // EXIT 버튼을 누르면 모든 창을 닫는 함수
	public static List<string> DateTimeList = new List<string> // ComboBox 목록을 대입하는 리스트
	public SeriesCollection SeriesData { get; private set; } // CartesianChart 차트 데이터
	public List<string> XLabel { get; set; } // CartesianChart X축 출력 값 변수
	public Func<int, string> Values { get; set; } // CartesianChart CPU, 메모리, 디스크 데이터 대입 변수
	public search // search 클래스 생성자, ComboBox 목록 생성하는 함수
	public async Task ComboListInit // Task로 비동기 방식으로 진행, DB연결하여 쿼리문을 이용하여 시간 목록에서 날짜와 '시'만 값으로 가져와서 ComboBox목록을 생성하는 함수 Dispatcher.BeginInvoke를 사용하여 ComboBox목록에 값 입력.
	public async Task SearchData // Task로 비동기 방식으로 진행, 콤보박스에서 선택한 값을 매개변수로 받아 조건문에 사용하여 값으로 가져와 그래프에 값을 업데이트 하는 함수 Dispatcher.BeginInvoke를 사용하여 그래프 값에 입력.
후 기	public async void btn_result_Click // SEARCH 버튼 누를 시 ComboBox에 선택된 값이 null 값이 아니라면 선택한 값에 해당하는 값으로 그래프를 생성하는 함수
	이번 개인 프로젝트는 람다식, 스레드와 Task에 대해 먼저 학습해야 진행할 수 있었던 프로젝트였던 것 같습니다. 개인적으로 공부하면서 헛갈리기도 하고 비동기로 코드를 구성하는 부분에 대해서 어떤 식으로 해야하는 지 감이 잡히지 않아 초반에는 난감하기도 했습니다. 그래도 계속 찾아보고 제가 이해한 방식으로 비동기 방식으로 코드를 구성하였습니다. 그리고 C# 라이브러리 'livechart'를 이용하여 그래프를 만들어 리소스 모니터링을 표현하는 부분에서도 어려움을 겪었습니다. 이 프로젝트 이후에도 UI 부분에서 그래프를 자주 사용할 것 같아 이번에 이 부분도 잘 익혀두자는 마음으로 진행했습니다. 이번 프로젝트를 진행하면서 C#의 비동기 방식(Task)에 대한 개념을 이해하는 데 도움이 많이 되었으며, WPF를 검색하면 자주 보이는 MVVM, MVC 모델에 대한 개념의 이해가 필요하다고 느꼈습니다.