

# 무인주차관제 시스템 및 시뮬레이션 개발

팀명 : 혼코

학과 : 컴퓨터공학부

학번 : 201711026

이름 : 강성원

역할 : 기획/설계/개발

# 목 차

## I. 서론

## II. 제안하는 무인주차관제 시스템 및 시뮬레이션

### 3.1 요구사항

- (1) 기능적 요구사항
- (2) 비기능적 요구사항

### 3.2 설계

- (1) 그래픽 사용자 인터페이스(GUI)
- (2) 소프트웨어(SW) 구조

### 3.3 구현

- (1) WPF
- (2) Thread
- (3) Singleton Pattern
- (4) 로그 저장
- (5) 가상 차량 데이터

# I. 서론

높은 밀집도를 지닌 도시는 차량의 증가에 따라서 불법주차 및 주차난이 발생하게 된다. 이러한 부분들을 개선하고자 지자체 또는 개인사업자가 차량들을 수용하기 위한 무/유료 주차장 시설들을 신설하였다. 과거 주차장 시설은 차량 입차 시 유인 요금소에서 주차증을 받고 신설된 주차 구역에서 주차하고, 출차 시 주차요금을 산출하기 위해서 주차증에 기재된 입차 시간과 출차 시간으로 주차 요금을 계산하였다.

하지만 유인 주차장 시설의 근무시간은 주간에만 근무하여 야간과 공휴일은 운영을 하지 않는 시설도 있어, 불법주차 및 주차난에 대한 문제점이 다시 초래하게 되었다. 과거에는 유인으로 운영하는 반면 최근에는 무인으로 주차장을 운영하는 무인주차관제 시스템이 개발되었다. 무인주차관제 시스템은 관리자 없이 주차장에 입차/출차 한 차량에 대한 요금과 주차시설관리를 할 수 있다. 이 시스템은 차량 입차 시 차 번호를 인식하여 차단기가 자동으로 열리고, 출차 시 차 번호를 인식하여 차량 조회한 다음 주차 요금 청구 및 결제 기능과 실시간 주차구역 표시, 차량 요금 조회, 이력, 시스템 정산 기능 등으로 갖추어져 있다. 그래서 시스템을 이용함으로써 근무자 없이 주차장 시설을 24 시간 운영을 할 수 있어, 근무자에 대한 경제적 및 시간적 측면에서 감소하였다.

오늘날 무인주차관제 시스템은 관리자가 요구한 기능을 분석 후 추가 구현하여 시스템이 점진적으로 발전이 되고 있다. 하지만 시스템 개발이 활성화됨에 따라 소프트웨어 테스트 과정 없이 적용 시 예상하지 못한 시스템 및 데이터 오류가 발생하게 되면서 해당 주차장의 운영이 중단되어 경제적, 시간적 손해와 주변 구역에 불법주차 및 주차난을 초래하게 된다. 경제적, 시간적 소모와 문제점을 초래될 수 있는 요인들을 해결하기 위해서는 개발된 무인주차관제 시스템에 대한 신뢰성과 성능을 검증할 수 있는 시뮬레이션이 필요하다.

본 프로젝트는 무인 주차관제 시스템 및 시뮬레이터를 제안한다. 무인 주차관제 시스템의 기본적인 기능은 주차구역 층마다 화면전환기능과 주차구역 표시, 입차/출차한 차량에 대한 데이터 저장, 주차요금 정산, 주차이력조회, 상태 메시지를 제공한다. 개발된 무인 주차관제 시스템에 대한 무결성과 성능, 신뢰성을 검증하기 위해 시뮬레이터를 구현한다. 시뮬레이터는  $n$  개의 차량 데이터를 생성할 수 있고, 입차/출차 이벤트 발생시간 설정 및 출력 기능을 제공한다. 무인 주차관제 시스템은 시뮬레이터에서 전송된 입차/출차 이벤트 발생 및 입차 차량 데이터를 수신하여 이벤트 발생에 따라서 입차/출차한 차량에 대한 데이터를 처리한다. 이로 통해서 무인 주차관제 시스템은 구현된 시뮬레이터로 통해 생성된 차량 데이터량만큼 이벤트 처리하면서 데이터에 대한 무결성과 시스템 성능, 신뢰성을 검증할 수 있다.

## II. 제안하는 무인 주차 관제 시스템 및 시뮬레이션

### 3.1 요구사항

#### (1) 기능적 요구사항

- 사용자가 입력한 차량의 대수만큼 시뮬레이터에서 차량 데이터를 생성한다.
- 사용자가 입력한 이벤트 최소/최대 발생시간 범위에서 이벤트를 발생한다.
- 이벤트로 발생한 입차/출차한 차량 및 시스템 데이터를 실시간으로 저장한다.
- 이벤트로 발생한 데이터를 실시간으로 처리할 수 있도록 자동화한다.
- 시스템 오류 또는 결함이 발생 시 동작 중인 시스템과 시뮬레이션을 중지시키고, 오류 메시지를 출력한 후 재동작할 수 있도록 전체적으로 초기화한다.

#### (2) 비기능적 요구사항

- 시뮬레이션 이벤트 최소 발생시간은 100ms부터 동작한다.
- 시뮬레이션에 생성할 차량의 대수는 사용자의 입력에 따라서 차량 데이터들을 생성한다.
- 로그 데이터 파일은 60,000 라인까지 데이터들을 저장한다.
- 무인 주차관제 시스템은 최소 100분 이상 시스템 오류 및 결함없이 실행한다.

## 3.2 설계

### (1) 그래픽 사용자 인터페이스(GUI)

사용자가 무인 주차관제 시스템의 프로그램을 사용하기 위해서 각층 주차공간화면, 상태 메시지 출력 및 삭제, 주차 현황, 주차 차량 이력, 로그 저장, 시뮬레이션, 설정, 날짜/시간의 UI 요소들을 배치하여 그림 1, 2, 3과 같이 레이아웃을 구성하였다.



Figure 1. 무인 주차관제 시스템 레이아웃

- 주차공간화면 : 차량이 주차된 구역은 빨간불을 표시하고, 비어있는 주차구역은 초록불로 표시
- 상태 메시지 : 주차 현황 기능과 시스템 메시지에 대한 시간, 입차/출차하는 차량 번호, 요금 출력
- 주차 현황 : 현재 주차된 모든 차량에 대한 차량 번호와 입차 시간을 상태 메시지UI에 출력
- 상태 메시지 삭제 : 상태 메시지UI에 출력된 데이터 삭제
- 1층/2층 : 각 층마다 주차 공간 화면UI 제공
- 주차 이력 : 사용자가 입력한 차량 번호로 통해 주차 이력 조회 제공
- ✓ 차량 검색 : 차량 리스트 검색 기능을 제공

- ✓ 차량 리스트 : 시뮬레이션으로 생성된 차량번호를 리스트로 제공
- ✓ 차량 이력 : 차량 리스트에서 선택된 차량에 대한 정보를 제공
- ✓ 새로 고침 : 차량 리스트 최신화



Figure 2. 무인 주차관제 시스템 주차 이력 레이아웃

- 로그 저장 : 입차/출차한 차량 및 시스템 데이터에 대한 로그 저장
- 시뮬레이션 : 무인 주차관제 시스템에 대한 시뮬레이션 기능 On/Off
- 설정 : 주차 요금과 시뮬레이션의 생성할 차량 대수, 이벤트 발생시간 최소/최대 설정
  - ✓ 생성할 차량 대수 : 시뮬레이션에서 생성할 가상 차량 대수 입력
  - ✓ 이벤트 발생 시간 : 시뮬레이션의 이벤트 최소/최대 발생 시간 입력
  - ✓ 30분 당 기본요금 : 무인 주차관제 시스템의 30분 당 기본요금 입력
  - ✓ 1시간 당 추가되는 기본요금 : 1시간 당 추가되는 기본요금 입력

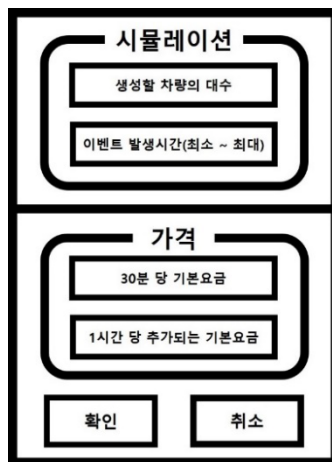


Figure 3. 무인 주차관제 시스템 설정 레이아웃

- 날짜/시간 : 현재 날짜와 시간을 출력

## (2) 소프트웨어(SW) 구조

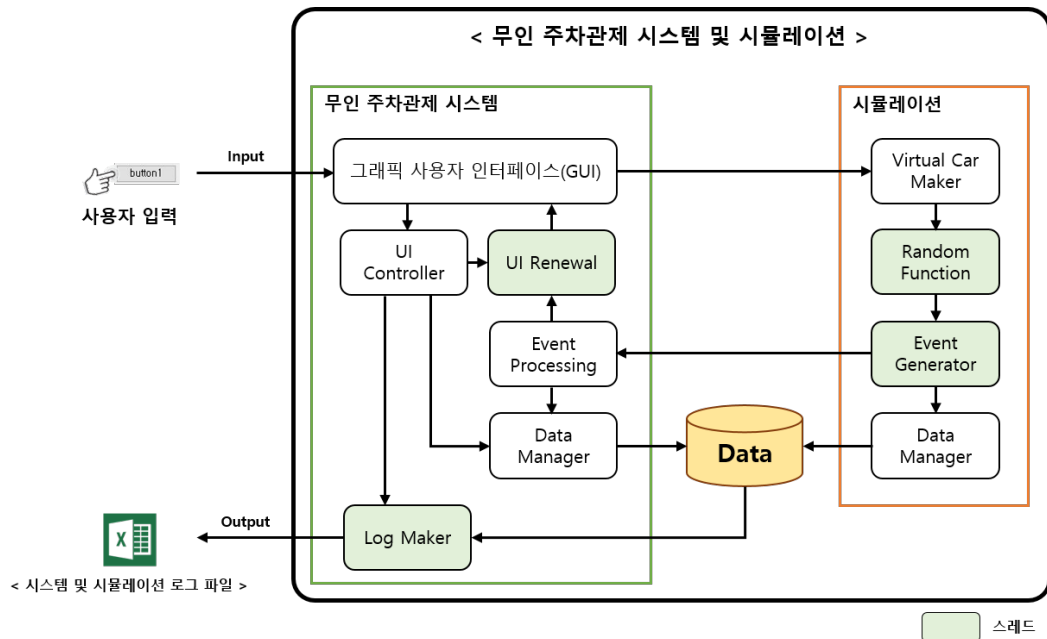


Figure 4. 무인 주차관제 시스템 및 시뮬레이션 구조

그림 4는 (1) 그래픽 사용자 인터페이스(GUI)를 백엔드 기능을 구성하여 무인 주차관제 시스템과 시뮬레이션 기능을 나누어 구조를 나타내었다. 무인 주차관제 시스템의 내부 구조는 사용자의 GUI 입력에 따라서 UI 기능을 실행하여 UI 갱신 및 출력과 로그 생성, 데이터 저장으로 구성된다. 그리고 GUI에서 시뮬레이션의 기능을 활성화 시 시작 또는 중지를 수행하여 가상 차량 데이터를 생성하고, 랜덤 값을 출력하여 값에 따라서 이벤트 발생 시간 출력 및 가상 차량 데이터를 전송한다. 무인 주차관제 시스템과 시뮬레이션에서 발생한 데이터는 데이터 관리 클래스로 통해서 저장하고, 로그 생성 GUI 활성화 시 저장된 데이터들을 문서 파일로 생성하여 저장한다.

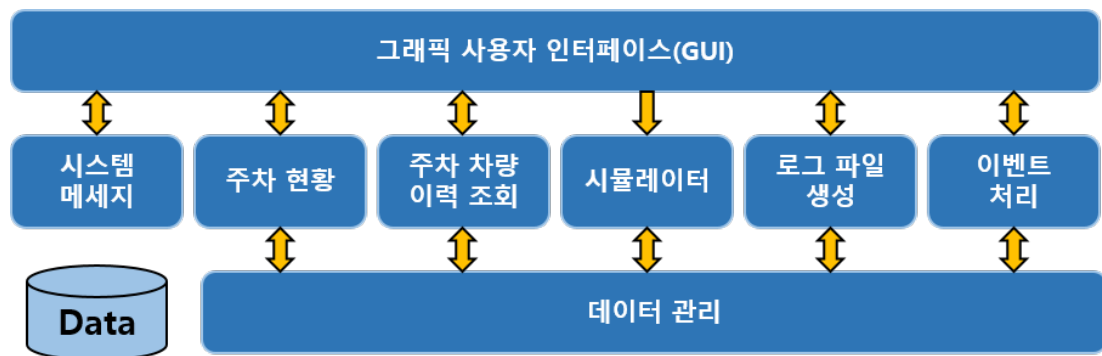


Figure 5. 소프트웨어 상세 구성도

그림 5은 (1) 그래픽 사용자 인터페이스(GUI)와 (2) 소프트웨어(SW) 구조기반으로 설계된 무인 주차관제 시스템 및 시뮬레이션 상세 구성도이다. 시스템 구성은 계층적으로 구성되어 기능 변경 및 추가가 용이하도록 설계되었다. 최상층은 사용자의 입력을 받는 C# 기반의 WPF UI로 구성된다. 중간층은 입력을 받은 시스템이 실제 수행되어야 할 기능으로 시스템 메세지, 주차 현황, 주차 차량 이력 조회, 시뮬레이터, 로그 파일 생성, 이벤트 처리로 구성된다. 최하층은 시스템 동작 시 수집되는 데이터들을 저장하는 데이터 저장과 시스템 또는 시뮬레이션에서 발생된 데이터들을 관리할 수 있게 구성되어 있다.



### 3.3 구현

제안한 프로젝트인 무인 주차관제 시스템 및 시뮬레이션을 개발하기 위해서 Table 1과 같이 개발 환경을 나타낸다.

Table 1. 개발 환경

시스템 모델	Microsoft Surface Pro 6
운영체제	Windows 10 64bit
CPU	i5-8250U
RAM	8GB
개발언어	C#
개발도구	Visual Studio 2017

제안한 프로젝트에 대해서 구현 가능성과 타당성을 검토 및 분석하여 아래와 같이 작성하였다. 사용자에게 제공할 UI 프레임워크와 데이터 및 UI 처리 등 동시 작업을 수행할 수 있는 병행 수행기법, 데이터 Save 또는 Load 할 수 있는 기법, 로그 데이터 파일 저장, 가상 데이터 생성에 대해서 제시한다.

#### (1) WPF

- GUI 프로그램 개발은 윈도우에서 작동하는 GUI 프로그램을 개발하기 위한 프로그래밍 모델인 WPF UI 프레임워크를 사용한다.
- 각 UI 요소에 대한 컨트롤 구성과 디자인 삽입, 데이터 바인딩을 사용하여 무인 주차관제 시스템의 레이아웃(프론트엔드)을 배치한다.
- 배치된 UI 요소마다 기능(백엔드)에 맞게 구현하여 사용자가 입력된 조작에 따라서 수행한다.

#### (2) Thread

- 무인 주차관제 시스템에서 시뮬레이션을 동작하면 프로그램의 성능이 전체적으로 저하되므로 동일 시간에 많은 작업을 수행하기 위해 작업에 따라서 각각 쓰레드를 사용한다.
- 각 쓰레드의 수행은 UI 갱신 및 출력, 시뮬레이션 동작, 데이터 저장, 예외처리를 병행한다.

#### (3) Singleton Pattern

- 무인 주차관제 시스템에서 발생된 데이터를 저장하고, 다른 클래스 및 폼에서 저장된 데이터를 사용하기 위해 싱글톤 패턴(Singleton Pattern) 기법을 사용한다.
- 싱글톤 패턴이란 프로그램이 시작될 때 클래스가 단 한번만 메모리를 할당하고, 전역 인스턴스를 만들어 사용하는 패턴을 말한다. 싱글톤 패턴은 메모리 낭비를 방지하고, 다른 클래스와 데이터를 공유하기가 쉽다.

#### **(4) 로그 저장**

- 무인 주차관제 시스템 및 시뮬레이션에서 발생 또는 생성된 데이터를 마이크로소프트사의 마이크로오피스 엑셀 문서 파일을 생성하여 모두 저장한다.
- 데이터는 시뮬레이션에서 생성된 차량 정보와 이벤트 발생 로그를 저장하고, 무인 주차관제 시스템에서 발생된 입차/출차에 대한 차량 정보와 주차요금, 상태메세지, 주차이력들을 저장한다.

#### **(5) 가상 차량 데이터**

- 무인 주차관제 시스템을 시뮬레이션하기 위해 입차/출차에 대한 차량 데이터로 시스템을 동작시켜야하므로 가상 차량 데이터를 생성한다.
- 사용자가 입력한 차량 대수만큼 가상 차량 데이터를 생성하며, 차량 데이터는 차량 번호, 입차/출차 날짜시간, 지불한 총 주차요금의 정보로 구조체형태로 지닌다.
- 차량 번호는 현 자동차관리법에 맞게 차량의 차종, 용도, 차량등록번호의 기준으로 구성하여 데이터를 난수로 생성한다.