텍스트, 폰트, 로고, 그래픽이(가) 표시된 사진

자동 생성된 설명

**주차타워 자동관리 프로그램**

**최종 보고서**

18010416 정 주 완

19011451 손 현 준

20003206 유 명 주

20003214 최 병 우

학과 : 컴퓨터공학과

과목 : C#프로그래밍(001)

조원 :

**1. 프로젝트 개요**

사용자가 주차장을 이용 할 시 차량 등록을 통하여 주차 위치를 안내해주고, 사용자의 주차 시간과 요금 정보를 간편하게 확인할 수 있도록 도와주는 프로그램입니다. 사용자 뿐만 아니라 관리자를 통하여 차량 관리를 고려한 인터페이스를 적용합니다

**2. 프로젝트 업무 분담**

|  |  |
| --- | --- |
| 손 현 준 | 데이터 베이스 설계 및 구현 |
| 정 주 완 | 차량 등록 화면 및 기능 구현 |
| 최 병 우 | 차량 찾기 화면 및 기능 구현 |
| 유 명 주 | UI디자인 및 사용자 인터페이스 구현 |

**3. 프로젝트 세부사항**

**3-1. 데이터베이스 구현**

주차 자동관리 프로그램은 NoSQL 유형의 MongoDB 데이터베이스를 사용하여 제작되었습니다. 프로그램에 데이터베이스를 연결하기 위해 .NET 드라이버인 MongoDB.Driver를 사용하였고, 데이터를 저장하고 있는 Collection은 vehicles(주차 차량 정보), parkingSpace(주차장 정보) 그리고 managerData(관리자) 총 3개가 있습니다.

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명※ mongoDB에 저장된 데이터 예시**

**※ Collection**

|  |  |
| --- | --- |
| 주차 차량 정보 Data | |
| VehicleNum | 차량의 번호를 저장. 차량 등록 페이지에서 받은 정보를 데이터베이스에 저장하고  차량 찾기 페이지에서 차량을 찾는데 사용. (관리 페이지에서 수정 가능) |
| ModelName | 차량의 모델명을 저장. 차량 등록 페이지에서 받은 정보를 데이터베이스에 저장하고 차량 찾기 페이지에서 차량을 찾는데 사용. (관리 페이지에서 수정 가능) |
| ParkingLocation | 주차된 위치를 저장. 차량 등록 페이지에서 차량이 등록될 때 1층부터 비어 있는 공간을 배정하고 자동으로 데이터베이스에 저장. |
| UserPhoneNumber | 차주의 전화번호를 저장. 차량 등록 페이지에서 받은 정보를 데이터베이스에 저장하고 차량 찾기 페이지에서 차량을 찾는데 사용. (관리 페이지에서 수정 가능) |
| ParkingTime | 이용 시작 시간을 저장. 차량 등록 페이지에서 차량이 등록될 때 DateTime.Now을 데이터베이스에 저장. |
| 주차장 정보 Data | |
| 1F / 2F / 3F | 각 층의 추가 최대 주차 공간을 저장. 차량을 등록하거나 찾을 경우 남은 주차공간을 계산하는 데 사용. |
| 관리자 Data | |
| ManagerNum | 관리자 번호를 저장. 관리자 페이지에서 받은 정보가 데이터베이스에 저장되고 로그인 시의 Id와 같은 역할로 관리자 구분에 사용. 관리자 페이지에 접근할 때 요구. |
| ManagerPassword | 관리자의 암호를 저장. 관리자 페이지에서 받은 정보를 단방향 해시 함수로 Salt를 사용하여 암호화한 후 데이터베이스에 저장. |

**3-2. 메인 화면 Form**

프로그램 실행 시에 메인으로 보여지는 화면입니다. Button 컨트롤에 이미지를 삽입하고, Button 클릭 시 사용자가 원하는 Form으로 이동시켜줍니다. 그리고 각각의 Form에 뒤로 가기 Button 컨트롤을 생성하여 다시 메인 화면으로 돌아올 수 있게 구현해주었습니다.

스크린샷, 휴대 전화, 텍스트, 정보기기이(가) 표시된 사진

자동 생성된 설명

• **해당하는 폼으로 이동**

**버튼 클릭**

**3-3. 차량 등록 폼 (RegisterForm)**

사용자로부터 차량의 정보를 입력 받아 주차타워DB에 등록하는 폼입니다.

사용자는 이 폼을 통해 주차 타워의 주차 여유공간을 확인 후 주차할 차량의 정보를 입력할 수 있으며, 차량 등록 폼은 입력받은 차량정보를 검사 후, 적절한 위치를 계산 하여 주차 정보를 DB에 저장합니다.

|  |  |
| --- | --- |
| 구 현 내 용 | |
| MongoDB 연결 | MongoDB에서 주차타워 층별 최대 주차 공간 불러오기 |
| MongoDB에서 지금까지 주차된 차량 정보를 모두 불러와, 각 층별 여유공간 계산 |
| 계산된 여유공간을 Window Form에 나타내기 |
| 유효성 검사 | 1. 등록하기 버튼 클릭 시간이 운영시간 내 인지 확인 |
| 2. 입력되지 않은 정보가 있는지 확인 |
| 3. 입력된 차량 번호가 한국 차량 번호판 형식과 일치하는지 검사  - 한국 차량 번호판 형식 : [2~3자리 숫자][1자리 한글][4자리 숫자]  - 예) 45바6251(O) 123가4567(O) 11112가나다(X) |
| 4. 입력된 전화 번호가 한국 전화번호 형식과 일치하는지 검사  - 한국 전화번호 형식 : [01][1자리 숫자]-[3~4자리 숫자]-[4자리 숫자]  - 예) 010-1234-5678(O) 011-123-4567(O) 75425-124(X) |
| 5. 여유 주차공간 여부 검사 |
| 검사 후  MongoDB에  정보 등록 | 1층부터 차례대로 주차할 수 있도록, 주차할 층수 계산 후 저장  (1층이 다 찼는지 확인후, 다 찼으면 2층 → 3층 순으로 저장) |
| 등록 시간을 자동으로 계산하여 저장  ( MongoDB 서버에 DateTime.Now를 이용하여 저장하면 UTC+0 시간으로 저장되므로, 한국시간인 UTC+9로 저장하도록 시간을 9시간 더한 후 저장 ) |

(유효성 검사- 1~5까지 하나라도 문제가 있는 경우 알림 메시지 띄우고 차량 등록 진행하지 않음)

**※ 실행화면**

(1) 화면 구성

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

(2) 차량 등록 과정

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

**3-4. 차량 찾기 폼 (FindForm)**

등록된 차량을 조회하여 이용 시간과 이용료를 확인할 수 있는 기능을 제공하는 폼입니다.

유효성 검사를 통해 올바른 형식의 차량 정보를 입력하고, MongoDB를 이용하여 데이터를 저장하고 조회합니다. 사용자는 검색된 차량의 이용 정보를 확인하고 결제를 진행할 수 있습니다. 이를 통해 편리한 차량 이용 서비스를 제공하고자 합니다.

|  |  |
| --- | --- |
| 기 능 | |
| 차량 정보 입력 | 사용자는 등록 시 입력한 차량 번호, 모델명, 휴대폰 번호 입력 |
| 입력된 정보는 유효성 검사를 거쳐 올바른 형식인지 확인 |
| 이용 정보 확인 | 간단한 이용 안내 문구(요금 및 주의사항)를 적어 쉽게 이용할 수 있도록 함 |
| 차량을 찾으면 등록 시간을 기준으로 총 이용 시간 계산 |
| 기본 요금과 추가 요금에 대한 정보 반환 |
| 결제 | 사용자는 이용한 시간과 결제 요금을 확인한 후 결제 진행 |
| 결제가 정상적으로 완료되면 이용한 차량의 정보 초기화 |

|  |  |
| --- | --- |
| 구 현 내 용 | |
| MongoDB 연결 | MongoDB 연결 정보를 설정하여 등록된 차량 정보를 저장하고 조회 |
| 데이터 베이스 연결을 위해 MongoDB.Driver 패키지 사용 |
| 유효성 검사 | 입력된 차량 번호와 휴대폰 번호에 대한 한국의 형식에 맞는지 유효성 검사 |
| 차량 번호는 정규식을 이용하여 유효성을 확인 |
| 이용료 결제 | 최종적으로 나오는 이용 시간, 이용료, 위치를 확인 후 결제 진행 |
| 결제 완료 후, 문구와 함께 데이터 베이스에 저장된 차량의 정보가 초기화 |

**※ 실행화면**

(1) 주차 조회

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

• **등록 시 저장된 정보와 유효성 검사**

• **입력된 정보가 정상일 시 결제창으로 이동**

• **입력된 정보가 비정상일 시 경고 알림**

**버튼 클릭**

**고객 정보 입력**

(2) 결제창

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

• **입력된 정보가 비정상일 시 결제 취소 알림**

- **이용 금액 30분 이후로 10분에 500원씩 추가 계산**

- **이용 금액 30분 미만일 시 0원 반환**

**버튼 클릭**

• **입력된 정보가 정상일 시 결제 완료 알림과 함께 고객 정보 초기화**

• **이용 금액에 맞추어 시간을 계산하여 금액 반환**

• **지정된 차량이 주차된 층**

• **등록 시간 부터 현재 시간 까지의 총 이용 시간**

**이용 정보 확인**

**3-5. 안내사항 폼 (GuidanceNote Form)**

안내사항 페이지에서는 주차장의 이용요금과 운영시간, 그 외의 주의 사항 등 주차장 이용 관련된 내용을 포함하고 있습니다. 기존에는 일일권과 정기권을 도입하여 프로그램을 구현하려 했으나 실제 주차 회원권 운영 방식의 이해도가 미흡하여 영수증 지참 시 1시간 무료 안내를 기입 하였습니다.

텍스트, 스크린샷, 디스플레이, 번호이(가) 표시된 사진

자동 생성된 설명

**3-6. 관리 폼 (Management Form)**

관리자 페이지에서는 Windows Forms의 dataGridView로, 저장된 차량의 정보를 한눈에 확인해 볼 수 있습니다. 여기에서는 원하는 차량 데이터를 선택하고 수정하거나 삭제/등록이 가능합니다. 또한, 영수증 번호를 입력한 후 차감을 누르면 요금 결제 시 사용시간에서 1시간을 차감해 주는 기능을 추가하였습니다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

관리자 등록은 로그인 시, ID역할을 하는 관리자번호와 암호를 데이터베이스에 등록하는 기능을 가지고 있습니다. 이 중에 암호는 단방향 해시 함수를 이용해 암호화되어 데이터베이스에 저장됩니다. 하지만 단방향 해시 함수를 이용한 암호화는 동일한 메시지에 동일한 결과값을 가지기 때문에 입력 값에 대한 결과값을 모아둔 빅데이터(Rainbow Table)를 이용하거나 무차별 대입 공격으로 해킹 될 확률이 높습니다. 때문에 이 프로그램에는 해시 함수를 돌리기 전에 원문에 Salt라고 불리는 임의의 문자열을 붙여서 설령 결과값을 알아낸다 하더라도 암호를 알아내기 더욱 어렵게 하였습니다. 또한 이 방법은 사용자마다 다른 Salt를 사용하기 때문에 한 명의 암호가 유출되더라도 같은 비밀번호를 사용하는 다른 사용자는 안전할 수가 있습니다. 예시로 같은 암호 ‘password’를 사용하는 두 사용자가 있을 때 암호화된 암호는 다음과 같습니다.

|  |  |
| --- | --- |
| 1번 사용자 | 2번 사용자 |
| EhPaYZv8HXYO867B6dds2riOEoIVWw9sSYmfwbmG5BwnI0Zq | PE731SapyyyelNoLfmZWDG4B2ZcuDZx4GM+OMg0Ewq1RBjD2 |

(위의 두 암호화된 문자열을 비교해보면 같은 암호를 사용하더라도 암호화된 문자열이 다름을 확인 해 볼 수 있습니다.)