

□ 기말 대치 프로젝트

(1) 프로젝트 주제는 본인이 자유롭게 선택합니다

(2) 필수 구현 내용

- I. 한 개 이상의 사용자 정의 클래스
- II. 한 개 이상의 컬렉션
- III. 한 개 이상의 파일 사용
- IV. 원 폼으로 구현

(3) 수업 시간에 다루었던 범위 내에서 구현해야 합니다. 단, 원 폼 컨트롤의 경우 프로퍼티와 메소드가 추가 되어도 상관없습니다

(4) "프로젝트 결과 보고서" <개발 내용> 항목에는 핵심 코드를 제시합니다

(5) "프로젝트 결과 보고서" <개발 결과> 항목에는 실행 과정을 제시합니다

(6) "프로젝트 결과 보고서" <참고 자료> 항목에는 참고 자료와 함께 생성형 AI를 활용한 내용을 캡처하여 제출합니다. 생성형 AI 활용이 반드시 있어야 합니다

(7) 핵심 코드 정리 시 코드에 대한 주석을 작성합니다

(8) pdf로 제출합니다

(9) 성적 산출

- I. 제시된 조건 구현 여부(10점)
- II. 아이디어의 독창성(5점)
- III. 구현 과정에서 코드 최적화(불필요하게 반복되는 코드 제거, 디버깅과 유지보수 효율성, 가독성)(7점)
- IV. 중간 발표(3점)
- V. 보고서 작성(5점)

(10) 중간 발표 : 프로젝트 개발 계획,

>> 프로젝트 개발 계획

>> 일시 : 6월 5일 수요일 5,6교시 2519

(11) 제출 기간

>> 6월 21일 금요일 23:00

## 프로젝트 결과 보고서

프로젝트 명	다목적 편의점 포스기		
기간	2024.05. ~ 2024.06.	학번	20195138
제출일	2024.06.19	이름	김준호

### 1. 개발 배경

상용되는 포스기 프로그램은 모두 편의점 기업 내부용으로 사용되고 일반인이 사용할 수 없을 뿐더러 QR코드가 아닌 EAN-13 바코드 구조를 사용하기 때문에 상대적으로 입력할 수 있는 데이터 양이 적고 수평 방향으로 스캔해야만 인식이 가능하다는 단점이 존재했다.

따라서 일반인도 데스크탑 PC 환경에서 조금의 수정을 통해 포스기 프로그램을 사용할 수 있도록 더 많은 기회를 제공하고 개개인의 사용 환경에 맞춰 사용할 수 있는 해당 프로그램을 제작하게 되었다.

### 2. 개발 내용

!! 나름 줄인다고 줄였는데 원본 코드가 1400줄이 넘어서 좀 깁니다. 죄송합니다.. 그래도 대부분 주석 작성하였으니 크게 어렵지 않을 것 같습니다...!! !!

[ 로그인 폼 로그인 함수 ]

```
/// <summary>
/// Firebase의 정보와 비교 후 로그인 허용/거부 함수
/// </summary>
async void GetMultipleDocumentsFromACollection()
{
    // ID와 PW를 이용한 Hash값 생성
    string hash = form3.SHA256Hash($"{textBox1.Text.Trim()} {textBox2.Text.Trim()}");

    DocumentReference docref = db.Collection("Add_Document_Width_AutoID").Document(hash);
    DocumentSnapshot snap = await docref.GetSnapshotAsync();

    if (snap.Exists)
    {
        Dictionary<string, object> identityDocument = snap.ToDictionary();
        object id;
        object pw;

        identityDocument.TryGetValue("ID", out id);
        identityDocument.TryGetValue("PW", out pw);

        if (id.Equals(textBox1.Text) && pw.Equals(textBox2.Text))
        {
            Close();
            DialogResult = DialogResult.OK;
            su = identityDocument["Su"].Equals("T") ? true : false;
            name = identityDocument["Name"].ToString();
        }
        else
```

```

        {
            MessageBox.Show("ID 또는 PW가 일치하지 않습니다.", "오류", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
}

```

#### [ 회원가입 폼 계정 생성 함수 ]

```

/// <summary>
/// 계정 생성 함수
/// </summary>
async void DataRegistration()
{
    string hash = SHA256Hash($"{textBox1.Text.Trim()}{textBox2.Text.Trim()}");

    DocumentReference coll = db.Collection("Add_Document_Width_AutoID").Document(hash);
    Dictionary<string, object> identityDocument = new Dictionary<string, object>()
    {
        { "ID", $"{textBox1.Text}" },
        { "PW", $"{textBox2.Text}" },
        { "Su", $"{domainUpDown1.Text}" },
        { "Name", $"{textBox4.Text}" },
    };

    coll.SetAsync(identityDocument);

    close();
}

```

#### [ 홈 페이지 이미지 관련 함수 ]

```

/// <summary>
/// 이미지 경로를 리스트에 추가하는 함수
/// </summary>
private void LoadImages()
{
    // 프로젝트 내부의 WbinWDebugWImages 폴더 경로 얻어오기
    // 제 컴퓨터 기준으로는 G:\4학년 1학기\원도우 프로그래밍\과제\기말
    // 프로젝트\Final_Project\WbinWDebugWImages 입니다.
    string imagesFolder = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Images");

    // 폴더 경로에서 이미지를 얻어온 뒤 images List에 추가
    images.Add(Path.Combine(imagesFolder, "image01.png"));
    images.Add(Path.Combine(imagesFolder, "image02.png"));
    images.Add(Path.Combine(imagesFolder, "image03.png"));
}

/// <summary>
/// 최초 로드 시 첫번째 이미지를 보여주는 함수
/// </summary>
private void ShowCurrentImage()
{
    // iamges List에 이미지가 등록되어 있으면
    if (images.Count > 0)
    {

```

```

        // 홈 페이지의 pictureBox에 이미지 등록
        pictureBox2.Image = Image.FromFile(images[0]);
    }
}

```

#### [ 계정 페이지 유저 등록 함수 ]

```

/// <summary>
/// 계정 페이지에 등록하기 위해 모든 계정 정보를 읽어오는 함수
/// </summary>
async void GetAllDocumentsFromACollection()
{
    // Firestore 데이터베이스의 "Add_Document_Width_AutoID" 컬렉션을 참조하는 쿼리 객체를 생성
    Query qref = db.Collection("Add_Document_Width_AutoID");
    QuerySnapshot snap = await qref.GetSnapshotAsync(); // 비동기적으로 쿼리 스냅샷을 가져옴

    // 가져온 스냅샷에서 각 문서를 반복하며
    foreach (DocumentSnapshot docsnap in snap)
    {
        // FirebaseProperty 객체로 변환
        FirebaseProperty fp = docsnap.ConvertTo<FirebaseProperty>();

        // 문서가 존재하고, 문서의 ID가 "Calendar", "Statistics"가 아닌 경우에만 리스트뷰에 추가
        if (docsnap.Exists && !docsnap.Id.Equals("Calendar") && !docsnap.Id.Equals("Statistics"))
        {
            listView2.Items.Add(new ListViewItem(new string[] { docsnap.Id, fp.ID, fp.PW, fp.Su ,
fp.Name }));
        }
    }
}

```

#### [ 홈 페이지 캘린더 함수 ]

```

/// <summary>
/// Firebase에서 Calendar를 읽어서 등록된 일정을 불러오거나 추가하는 함수
/// </summary>
async void GetMultipleDocumentsFromACollection()
{
    // "Add_Document_Width_AutoID" 컬렉션의 "Calendar" 문서를 참조하는 DocumentReference 객체를 생성
    DocumentReference docref = db.Collection("Add_Document_Width_AutoID").Document("Calendar");
    DocumentSnapshot snap = await docref.GetSnapshotAsync(); // 비동기적으로 "Calendar" 문서의
스냅샷을 가져옴

    // monthCalendar1에서 선택된 날짜를 문자열 형식으로 가져옴
    string dateString = monthCalendar1.SelectionStart.ToShortDateString();

    // 문서가 존재하면
    if (snap.Exists)
    {
        // 문서를 Dictionary<string, object> 형식으로 변환
        Dictionary<string, object> schedule = snap.ToDictionary();

        object memo = string.Empty;

        // 선택된 날짜(dateString)가 schedule에 존재하는지 확인
        if (schedule.TryGetValue(dateString, out memo))

```

```

    {
        // 존재하면 리스트박스에 날짜와 기존 메모에 새로운 메모를 추가하여 표시
        listBox1.Items.Add($"{dateString} : {memo + "\n" + textBox13.Text}");
        schedule[dateString] = memo + "\n" + textBox13.Text;
    }
    else
    {
        // 존재하지 않으면 리스트박스에 날짜와 새로운 메모를 추가하여 표시
        listBox1.Items.Add($"{dateString} : {textBox13.Text}");
        schedule.Add(dateString, textBox13.Text);
    }

    // 업데이트된 schedule을 비동기적으로 Firestore에 저장
    docref.SetAsync(schedule);
    Clear(textBox13);
}
}

```

#### [ 홈 페이지 검색, 유튜브 함수 ]

```

/// <summary>
/// 폼 로드 시 웹 브라우저에 유튜브를 띄우는 함수
/// </summary>
/// <param name="e"></param>
protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);

    // HTML 문자열을 생성, IE 브라우저 호환성을 위해 메타 태그와 유튜브 영상을 포함하는 iframe을
    포함
    var embed = "<html><head>" +
        "<meta http-equiv=W\"X-UA-CompatibleW\" content=W\"IE=Edge, chrome=1W\"/>" +
        "</head><body>" +
        "<iframe width=W\"440W\" height=W\"415W\" src=W\"{0}W\" " +
        "frameborder = W\"0W\" allow = W\"autoplay; encrypted-mediaW\" allowfullscreen></iframe>" +
        "</body></html>";
    var url = "https://www.youtube.com/embed/kh52htRZsi8"; // 임의의 유튜브 링크 설정
    this.webBrowser1.DocumentText = string.Format(embed, url); // 웹 브라우저 컨트롤에 HTML
    문자열을 설정, 유튜브 URL을 embed 문자열의 {0} 위치에 삽입
}

/// <summary>
/// 검색어를 크롬을 이용해 검색하는 함수
/// </summary>
/// <param name="url">검색할 쿼리가 추가된 최종 URL</param>
private void OpenUrlInChrome(string url)
{
    try
    {
        // 크롬을 이용해 전달받은 URL 사용
        Process.Start("chrome.exe", url);
    }
    catch (Exception ex)
    {
        // 크롬이 설치되지 않은 경우
        MessageBox.Show("크롬 브라우저를 열 수 없습니다. 크롬이 설치되어 있는지 확인해 주세요.\n" +

```

```

ex.Message, "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

#### [ 결제 페이지 QR코드 인식 버튼 함수 ]

```

/// <summary>
/// QR코드 인식 버튼 함수
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button7_Click(object sender, EventArgs e)
{
    // ZXing 라이브러리를 사용하여 바코드를 읽는 BarcodeReader 객체를 생성
    ZXing.BarcodeReader barcodeReader = new ZXing.BarcodeReader();

    // 파일 경로를 저장할 문자열 초기화
    string filePath = string.Empty;

    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.InitialDirectory = "c:\\www"; // 초기 디렉토리를 C 드라이브로 설정
    openFileDialog.Filter = "All Files | *.jpg"; // 파일 필터를 설정하여 .jpg 파일만 표시
    openFileDialog.CheckFileExists = true; // 파일이 실제로 존재하는지 확인
    openFileDialog.CheckPathExists = true; // 경로가 실제로 존재하는지 확인

    // 사용자가 파일을 선택하고 확인 버튼을 눌렀을 때
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        // 선택한 파일의 경로를 filePath에 저장
        filePath = openFileDialog.FileName;

        // 선택한 파일로부터 비트맵 이미지를 생성
        Bitmap barcodeBitmap = (Bitmap)Image.FromFile(filePath);

        // 바코드 이미지를 디코딩하여 제품 정보를 추출
        var productInfo = barcodeReader.Decode(barcodeBitmap);

        // ListView에 제품 정보를 추가
        listView1.Items.Add(ProductAdd(productInfo.Text));

        // 가격 정보를 업데이트
        UpdatePrice();
    }
}

```

#### [ 계정 페이지 프린터 함수 ]

```

/// <summary>
/// 계정 정보 출력 함수
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void printDocument1_PrintPage(object sender, PrintPageEventArgs e)
{
    Font printFont = new Font("Arial", 14, FontStyle.Bold);
    int index = 1;
}

```

```

        if (listView.Name.Equals("listView2"))
        {
            foreach (ListViewItem item in listView.Items)
            {
                ListViewItem.ListViewSubItemCollection subItem = item.SubItems;
                e.Graphics.DrawString("유저 코드 : " + subItem[0].Text + "
아이디 : " + subItem[1].Text
+ "
패스워드 : " + subItem[2].Text + "
권한 : " + subItem[3].Text + "
이름 : " + subItem[4].Text
+ "
" + pageSet, printFont, Brushes.Black, 10, index * 10);
                index += 20;
            }
        }
        else if (listView.Name.Equals("listView3"))
        {
            foreach (ListViewItem item in listView.Items)
            {
                ListViewItem.ListViewSubItemCollection subItem = item.SubItems;
                e.Graphics.DrawString("번호 : " + subItem[0].Text + "
상품 코드 : " + subItem[1].Text +
"
상품명 : " + subItem[2].Text + "
단가 : " + subItem[3].Text + "
수량 : " + subItem[4].Text +
"
금액 : " + subItem[5].Text + "
입고 : " + subItem[6].Text + "
" + pageSet, printFont,
Brushes.Black, 10, index * 10);
                index += 20;
            }
        }
    }
}

```

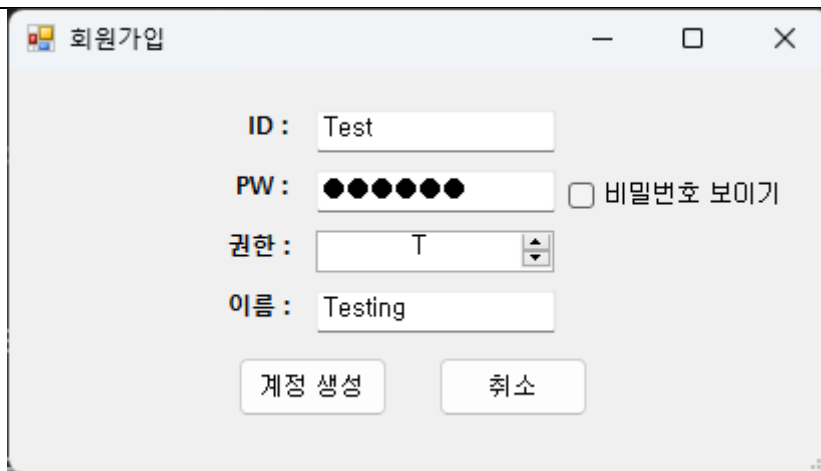
### 3. 개발 결과

최초 실행 화면 :



기능 : ID, PW를 입력해 로그인하거나 회원가입을 할 수 있습니다. ID, PW의 경우 Junho, 123456를 사용하시거나 회원가입을 통해 새로 계정을 만든 뒤 로그인하시면 됩니다.

회원 가입 화면 :



회원가입

ID : Test

PW : ●●●●●● ☐ 비밀번호 보이기

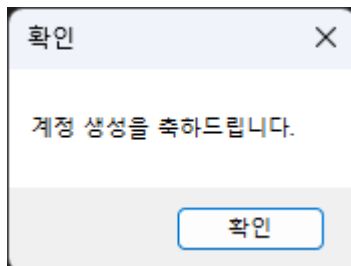
권한 : T

이름 : Testing

계정 생성 취소

기능 : ID와 PW, 권한, 이름을 입력해 계정을 만들 수 있습니다. 권한은 추후 다른 사용자를 추가, 수정, 제거 하는 등의 기능에 쓰입니다.

계정 생성 화면 :



확인

계정 생성을 축하드립니다.

확인



9a931c55ac02bf216550c464b1992a30c522dfabf6cb31deada5c716bc13a263

+ 컬렉션 시작

+ 필드 추가

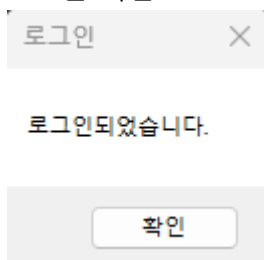
ID: "Test"

Name: "Testing"

PW: "123456"

Su: "T"

로그인 화면 :



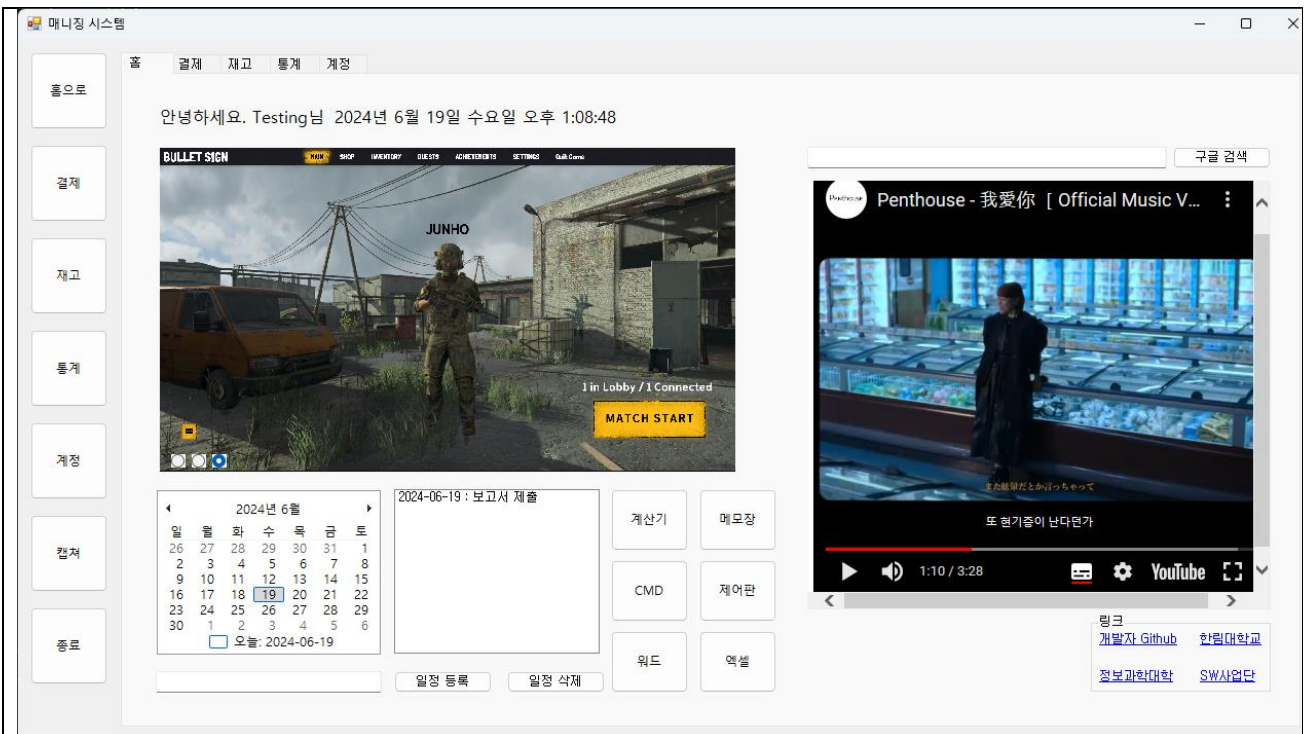
로그인

로그인되었습니다.

확인

홈페이지 :





기능 : 좌측 버튼을 통해 각 탭으로 이동할 수 있고 프로그램 캡처와 종료가 가능합니다. 탭 이동은 탭을 직접 클릭하여 이동할 수도 있습니다. 페이지 상단에는 먼저 로그인한 유저의 이름과 현재 시간이 나타나며 그 아래에는 radio 버튼을 통해 확인할 수 있는 여러 이미지가 있습니다. 캘린더의 경우 원하는 날짜를 클릭하여 일정 등록과 삭제가 가능하며 Firebase와 연동되어 프로그램을 종료하고 다시 키더라도 확인할 수 있습니다.

계산기, 메모장, CMD, 제어판, 워드, 엑셀은 컴퓨터에 기본적으로 설치된 기능들을 사용할 수 있도록 불러오는 버튼들이며 워드와 엑셀은 새 문서를 만들어 실행됩니다. 워드와 엑셀이 설치되지 않은 경우는 에러 메시지 박스를 띄워 사용자에게 알립니다.

우측 상단의 구글 검색은 검색할 내용을 입력하고 버튼을 누르면 크롬의 새로운 탭에 검색 결과를 보여줍니다. 이 경우에도 크롬이 설치되어 있지 않은 경우 에러 메시지 박스가 띄워집니다. 아래에는 webbrowser를 이용한 유튜브 화면입니다. 스크립트 상에서 원하는 유튜브로 링크를 변경할 수 있고 동영상 재생도 가능합니다.

마지막으로 우측 하단의 링크라벨입니다. 제 Github 링크와 한림대학교, 정보과학대학, SW사업단 링크가 배치되어 있으며 실제 이동이 가능합니다.

결제 페이지 :

[illegible]

기능 :

- QR코드 인식 버튼: QR코드 인식 버튼을 누르면 파일 탐색기가 열리며 파일 탐색기에서 QR코드 파일을 선택하면 해당 QR코드의 정보를 읽은 다음 ListView에 추가합니다. QR코드가 저장하는 정보는 번호, 상품 코드, 상품명, 단가, 수량, 금액, 입고입니다.
- 상품 등록 버튼 : 상품 등록 버튼을 누르면 위의 TextBox에 입력된 내용을 기반으로 QR코드를 만든 뒤 바탕화면에 저장합니다.
- 상품 수정 버튼 : 상품 수정 버튼을 누르면 선택한 ListView의 상품 정보를 TextBox에 쓰여진 내용으로 수정합니다.
- 상품 삭제 버튼 : 상품 삭제 버튼을 누르면 선택한 ListView의 상품을 삭제합니다.
- 상품 결제 버튼 : 상품 결제 버튼을 누르면 총 결제 금액이 적힌 우측 TextBox의 Text가 지워지고 결제 금액을 Firebase의 Statistics에 저장된 금일 매출액에 더합니다.

재고 페이지 :

매니징 시스템

홈으로 결제 재고 통계 계정

번호 상품 코드 상품명 단가 수량 금액 입고

번호 : 상품코드 : 상품명 :

단가 : 수량 : 금액 : 입고 :

☐ 초코송이
 ☐ 열라면
 ☐ 토레타
 ☐ 새우깡
 ☐ 신라면
 ☐ 파워에이드
 ☐ 흥분볼
 ☐ 튀김우동
 ☐ 게토레이
 ☐ 프링글스
 ☐ 불닭볶음면
 ☐ 코카콜라
 ☐ 오에스
 ☐ 짜파게티
 ☐ 캡시
 ☐ 초코파이
 ☐ 짜파구리
 ☐ 칠성사이다
 ☐ 초코하임
 ☐ 진라면
 ☐ 나랑드사이다
 ☐ 뽕빠로
 ☐ 진라면 순한맛
 ☐ 스프라이트
 ☐ 포스틱
 ☐ 삼양라면
 ☐ 환타
 ☐ 자갈치
 ☐ 육개장
 ☐ 데자와
 ☐ 고알콘
 ☐ 왕뚜껍
 ☐ 레쓰비
 ☐ 츠러스
 ☐ 너구리
 ☐ TOP
 ☐ 웨하스
 ☐ 통새라면
 ☐ 솔의눈
 ☐ 바나나킥
 ☐ 김치라면
 ☐ 바나나우유

>> <<

수량 : 입고 :

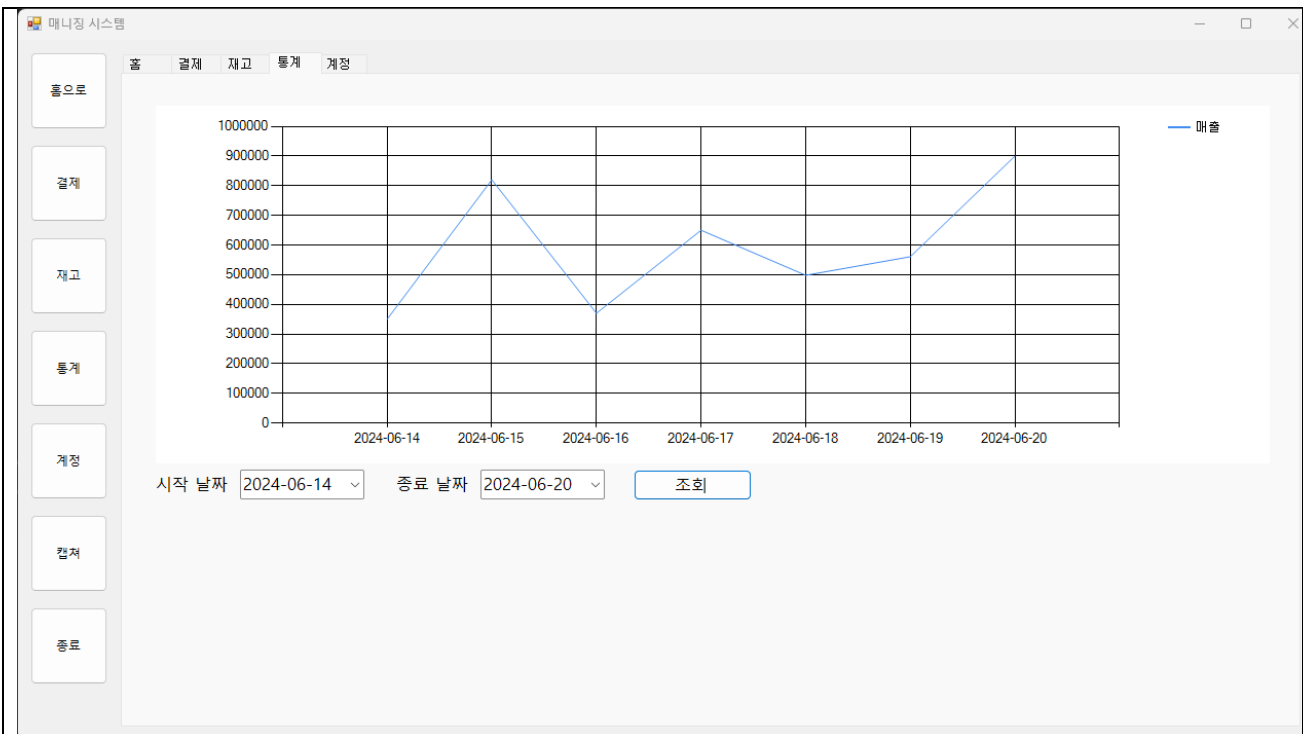
단가 : 금액 :

재고 등록 재고 수정 재고 삭제 엑셀 저장 인쇄

#### 기능 :

- >> 버튼 : TextBox에 추가하려는 상품에 대한 정보를 작성한 다음 재고 등록 하거나 이미 존재하는 상품 정보를 수정한 다음 ">>" 버튼을 클릭하면 해당 정보를 checkedListBox에 추가할 수 있습니다. 단, 상품명이 동일한 경우 추가할 수 없습니다.
- << 버튼 : 우측 checkedListBox에서 상품을 선택하고 하단의 콤보박스에서 수량, 입고, 단가, 금액을 선택한 다음 "<<" 버튼을 클릭하면 해당 정보를 ListView에 추가하여 재고 목록에 추가할 수 있습니다.
- 재고 등록 버튼 : TextBox에 상품 정보를 작성한 다음 재고 등록 버튼을 누르면 ListView에 작성한 상품 정보가 뜨게 됩니다.
- 재고 수정 버튼 : 이미 ListView에 존재하는 아이템을 클릭하여 TextBox에 띄운 다음 수정하고 재고 수정 버튼을 누르면 ListView의 아이템이 수정된 정보로 바뀌게 됩니다.
- 재고 삭제 버튼 : ListView에 선택한 아이템을 삭제합니다.
- 엑셀 저장 버튼 : 현재 ListView에 작성된 내용들을 엑셀로 옮겨 표시합니다.
- 인쇄 버튼 : 현재 ListView에 작성된 내용을 프린트합니다.

#### 통계 페이지 :



기능 : 시작 날짜와 종료 날짜를 선택한 뒤 조회 버튼을 누르면 Firebase의 Statistics에서 시작 날짜부터 종료날짜 까지의 매출 정보를 가져와 차트에 표시합니다. 매출은 결제 페이지에서 결제할 경우 결제한 날짜의 매출에 추가됩니다.

계정 페이지 :

매니징 시스템

홈 결제 재고 통계 계정

홈으로

결제

재고

통계

계정

캡처

종료

유저 코드	아이디	패스워드	권한	이름	
169e877703902735ff4...	Jiho	45678	T	김지호	유저 등록
9a931c55ac02bf21655...	Test	123456	T	Testing	
e28ddbf9b617960ed2...	Junho	123456	T	김준호	유저 수정
					유저 삭제

유저 코드 :  아이디 :  패스워드 :  권한 :  이름 :

초기화

엑셀 저장

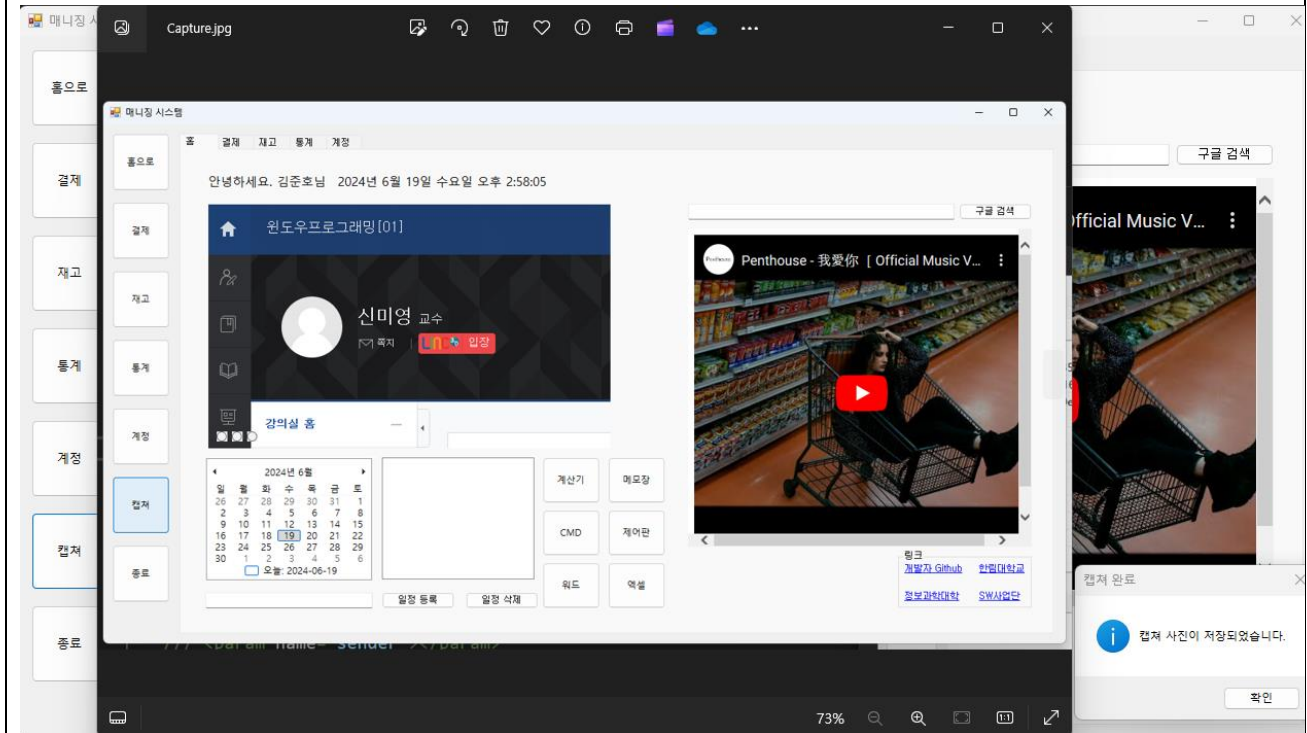
인쇄

기능 :

- 유저 등록 버튼 : TextBox에 작성한 정보를 이용해 새로운 유저를 추가합니다. 이때, 권한이 T 인 유저로 로그인 한 상태여야만 유저 등록이 가능합니다.

- 유저 수정 버튼 : ListView에서 선택하여 TextBox에서 유저 정보를 수정한 다음 유저 수정 버튼을 누르면 해당 정보로 수정됩니다. 이 역시 권한이 T인 유저로 로그인 해야 가능합니다.
- 유저 삭제 버튼 : ListView에서 선택한 유저 정보를 삭제합니다. 권한이 T인 유저로 로그인 해야 가능합니다.
- 엑셀 저장 버튼 : 현재 ListView에 띄워진 전체 계정 정보를 엑셀로 열어 표시합니다.
- 인쇄 버튼 : 현재 ListView에 띄워진 전체 계정 정보를 프린트합니다.
- 초기화 버튼 : 모든 TextBox를 지웁니다.

좌측 캡처 버튼 :



#### 4. 참고자료

ChatGPT :

원폼에서 오늘 날짜를 기준으로 앞 뒤 일주일을 DateTime을 이용해 출력하는 코드를 알려줘



원폼 (Windows Forms) 애플리케이션에서 오늘 날짜를 기준으로 앞뒤 일주일을 `DateTime`을 이용해 출력하는 방법을 알려드리겠습니다. 이 예제에서는 `Label` 컨트롤을 사용하여 날짜를 출력하도록 하겠습니다.

먼저, 새로운 원폼 프로젝트를 생성한 후 `Form1`에 아래 코드를 추가합니다.

1. `Label` 컨트롤을 폼에 추가합니다.
2. 폼의 로드 이벤트에서 날짜를 계산하고 출력하는 코드를 작성합니다.

### Form1.cs

csharp

코드 복사

```
using System;
using System.Windows.Forms;

namespace DateRangeApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // 오늘 날짜를 가져옴
            DateTime today = DateTime.Now;

            // 일주일 전 날짜 계산
            DateTime oneWeekAgo = today.AddDays(-7);

            // 일주일 후 날짜 계산
            DateTime oneWeekLater = today.AddDays(7);

            // Label에 출력
            label1.Text = $"일주일 전: {oneWeekAgo.ToShortDateString()}\n" +
                $"오늘: {today.ToShortDateString()}\n" +
                $"일주일 후: {oneWeekLater.ToShortDateString()}";
        }
    }
}
```



메시지 ChatGPT



뤼튼 :



Lithium

윈도우 폼에서 리스트뷰에 아이템이 추가된 순간에 다른 행동을 하려는데 어떤 함수를 써야하지?



뤼튼 AI 과제와 업무

Windows Forms에서 ListView에 아이템이 추가된 순간에 특정 행동을 수행하려면 ListView의 Items 컬렉션에 대한 변경을 감지해야 합니다. 기본적으로 ListView 자체에는 아이템이 추가될 때 발생하는 이벤트가 없지만, 이를 해결하기 위해 ListView를 상속받아 커스텀 클래스를 만들고 아이템이 추가될 때 이벤트를 발생시키는 방법을 사용할 수 있습니다.

예를 들어, ListView를 상속받아 새로운 아이템이 추가될 때 이벤트를 발생시키는 CustomListView 클래스를 만들 수 있습니다:

csharp



```
using System;
using System.Windows.Forms;

public class CustomListView : ListView
{
    // 아이템이 추가될 때 발생하는 이벤트를 정의
    public event EventHandler<ItemAddedEventArgs> ItemAdded;

    // 새로운 아이템이 추가될 때 호출되는 메소드
    protected virtual void OnItemAdded(ItemAddedEventArgs e)
    {
        ItemAdded?.Invoke(this, e);
    }

    // Items 컬렉션을 오버라이드하여 아이템 추가를 감지
    public new ListViewItemCollection Items
    {
        get { return base.Items; }
    }

    public new void Add(ListViewItem item)
    {
        base.Items.Add(item);
        OnItemAdded(new ItemAddedEventArgs(item));
    }
}

public class ItemAddedEventArgs : EventArgs
{
    public ListViewItem Item { get; }

    public ItemAddedEventArgs(ListViewItem item)
    {
        Item = item;
    }
}

// 사용 예제
public class MainForm : Form
{
    private CustomListView customListView;
```



뤼튼에게 무엇이든 물어보세요

이미지 경로를 상대경로를 사용하려면 어떻게 해야하지??

### 뤼튼 AI 과제와 업무

이미지 경로를 상대 경로로 사용하려면 프로젝트 디렉토리를 기준으로 이미지 파일을 배치하고, 해당 경로를 설정해주면 됩니다. 상대 경로를 사용하면 코드가 다른 환경에서도 쉽게 동작하도록 할 수 있습니다.

다음은 상대 경로를 사용하는 예제입니다.

#### 프로젝트 구조

먼저, 프로젝트 디렉토리 구조를 다음과 같이 구성한다고 가정합니다:

```
MyProject
|
|-- bin
|   |-- Debug
|   |-- Release
|
|-- Images
|   |-- image1.jpg
|   |-- image2.jpg
|   |-- image3.jpg
|
|-- MainForm.cs
```

#### 상대 경로를 이용한 코드 예제

이미지 파일이 Images 폴더에 있다고 가정하고, 상대 경로로 이미지를 불러오는 방법을 설명합니다.

csharp

```
public partial class MainForm : Form
{
    private List<string> images = new List<string>(); // 이미지 경로 목록
    private int currentIndex = 0; // 현재 표시 중인 이미지의 인덱스

    public MainForm()
    {
        InitializeComponent();
        LoadImages();
        ShowCurrentImage();
    }

    private void LoadImages()
    {
        // 이미지 폴더의 상대 경로
        string imagesFolder = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "Images");

        // 이미지 파일 경로 추가 (상대 경로 사용)
        images.Add(Path.Combine(imagesFolder, "image1.jpg"));
        images.Add(Path.Combine(imagesFolder, "image2.jpg"));
        images.Add(Path.Combine(imagesFolder, "image3.jpg"));
    }
}
```

뤼튼에게 무엇이든 물어보세요

기타 :

<https://gameprogramming.tistory.com/14>

<https://darusamu.tistory.com/34>

<https://hubbleconstant.tistory.com/93>

<https://zadd.tistory.com/46>