

집밥 헛선생

- 건강식단과 레시피 자동구성

알파코 RPA 5기 3조

김주성 남진식 이혁재 차아란



목차



01

프로젝트 개요

02

프로젝트 구성 및 역할

03

프로젝트 수행 절차 및 방법

04

프로젝트 수행 결과

05

자체 평가 의견

01. 프로젝트 개요

배달비를 아끼고 건강을 챙기고 싶은 사람에게
집에서 효과적으로 요리를 할 수 있도록
도움을 주는 요리 맞춤 서비스

- 영양 밸런스를 고려한 건강한 일주일치 식단을 자동으로 구성
- 원하는 일자의 일일 식단 불러오기
- 원하는 음식의 레시피를 챗 GPT에 답변받고 WORD 파일로 저장
- 일주일치 식단의 칼로리 비교해서 시각화

활용 프로그램 : UIPATH, COLAB, EXCEL, WORD, 챗 GPT

니즈에 맞는 데이터 수집과 레시피 제작 및 시각화를
효율적으로 자동화 처리



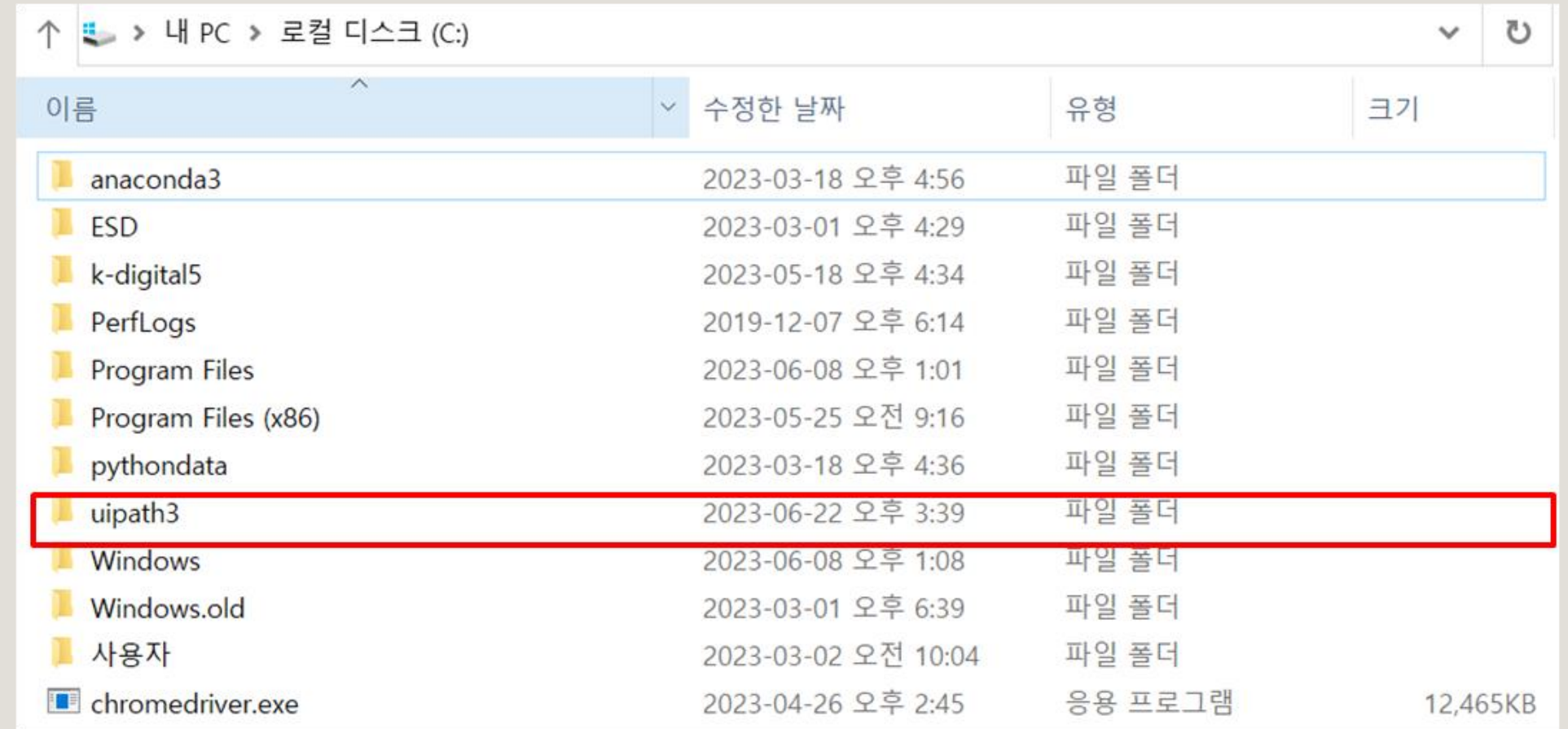
02. 프로젝트 구성 및 역할

이름	역할	담당 업무
김 주 성	팀 장	Flow Chart 작성, 식단 작성 부분 Part, 영상제작, 발표
이 혁 재	팀 원	식단 칼로리 비교 Part
차 아 란	팀 원	WorkFlow통합, PPT작성
남 진 식	팀 원	Recipe 검색 Part, PPT작성

03. 수행 절차 및 방법

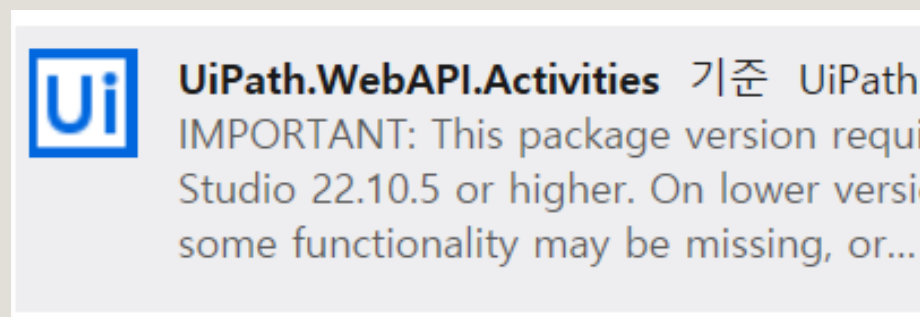
1. 실행하기 위해서는 UiPath와 Excel, Word가 필요합니다.
2. 동봉된 압축파일 (uipath3.zip)파일은 C드라이브에 바로 압축을 풀어주세요
3. Config파일의 APIkey 설정을 제외하고, Config 파일은 건드리지 말아주세요.
4. UiPath 패키지 2가지 설치가 필요합니다.
(UiPath.WebAPI.Activities v1.16.2 /
UiPath.Word.Activities v1.17.0)
5. 값들은 uipath3 폴더 내 input 엑셀에서 설정해 주시고, 시트명은 변경하지 말아주세요
6. 레시피 및 식단변경 시트를 제외하고는 시트당 B2 값을 설정하여 사용해야 합니다.
7. 매주 자동 실행되는 시간을 Input 파일에 기입하시면, 해당 시간에 다음 주 식단과 식단 관련 차트가 자동으로 생성됩니다.

- 기본 세팅

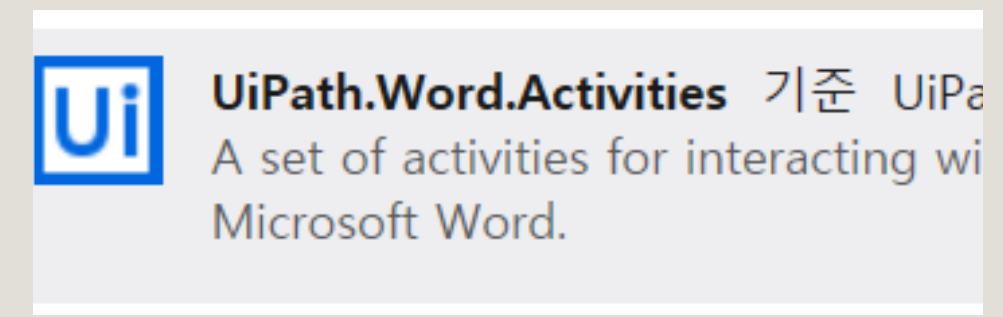


이름	수정한 날짜	유형	크기
anaconda3	2023-03-18 오후 4:56	파일 폴더	
ESD	2023-03-01 오후 4:29	파일 폴더	
k-digital5	2023-05-18 오후 4:34	파일 폴더	
PerfLogs	2019-12-07 오후 6:14	파일 폴더	
Program Files	2023-06-08 오후 1:01	파일 폴더	
Program Files (x86)	2023-05-25 오전 9:16	파일 폴더	
pythondata	2023-03-18 오후 4:36	파일 폴더	
uipath3	2023-06-22 오후 3:39	파일 폴더	
Windows	2023-06-08 오후 1:08	파일 폴더	
Windows.old	2023-03-01 오후 6:39	파일 폴더	
사용자	2023-03-02 오전 10:04	파일 폴더	
chromedriver.exe	2023-04-26 오후 2:45	응용 프로그램	12,465KB

<uipath3 폴더를 해당 위치에 만들어주세요>



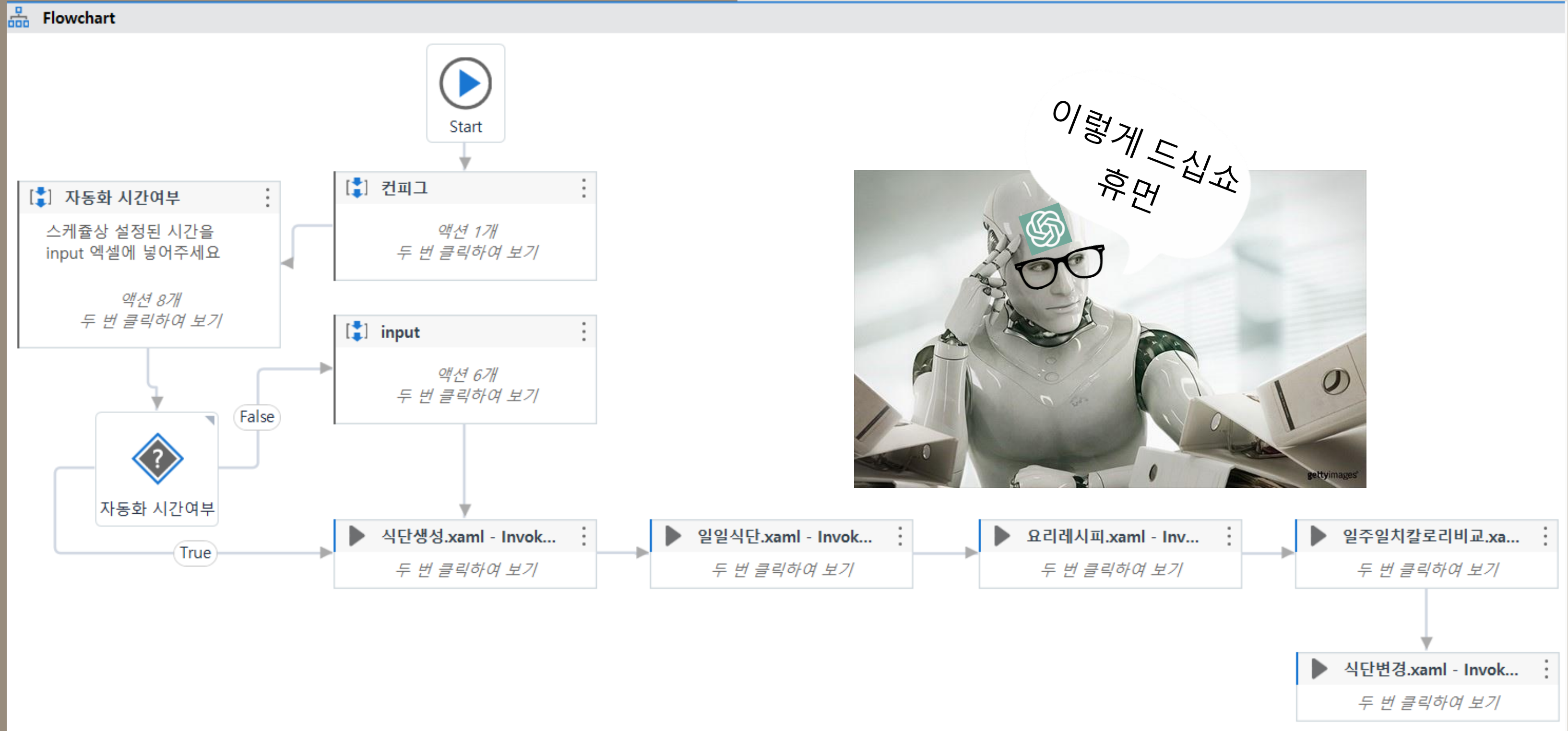
<UiPath.WebAPI.Activities>



<UiPath.Word.Activities>

03. 수행 절차 및 방법

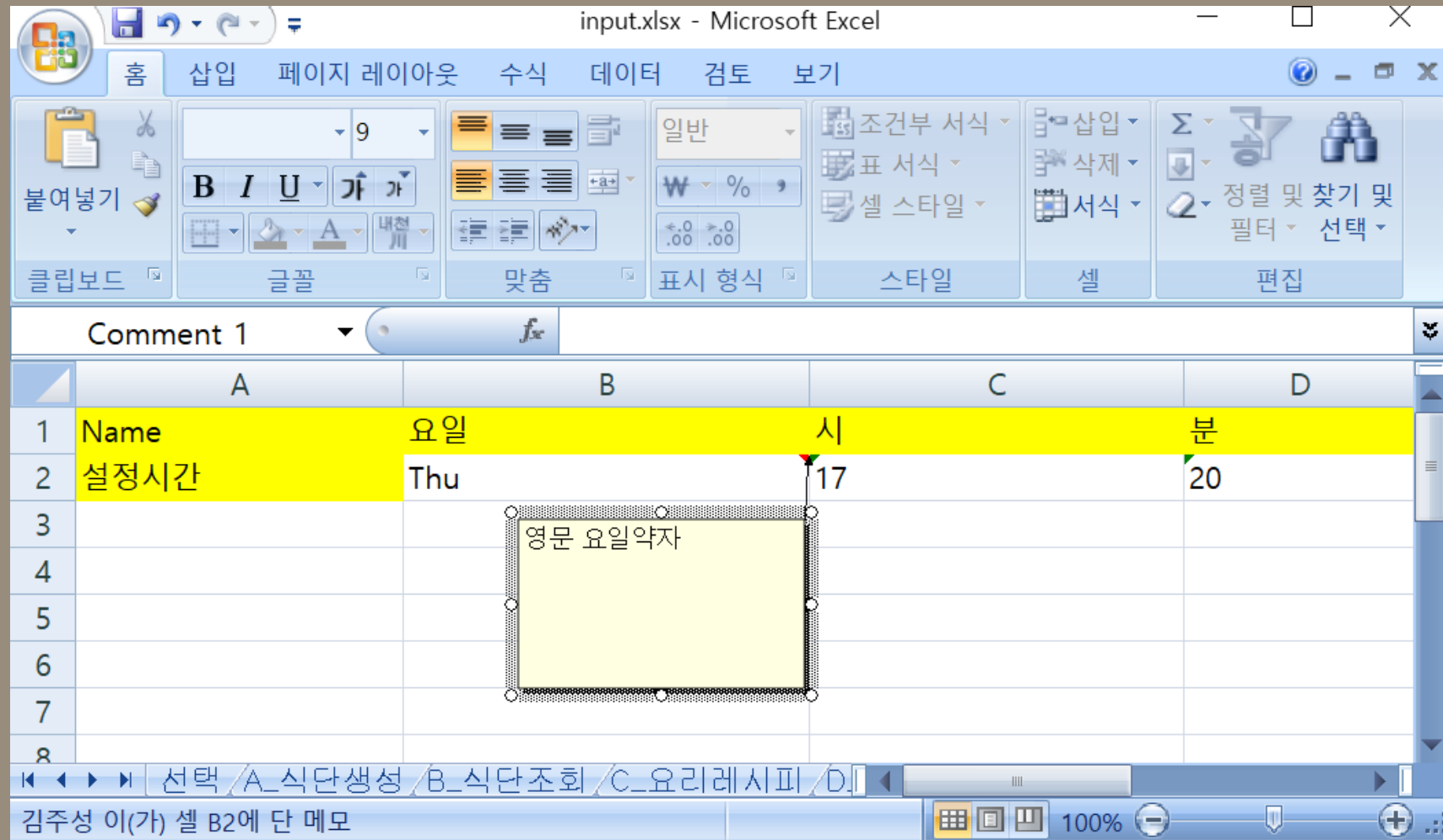
- 전체 로직



총 5개의 주요 로직으로 구성된 집밥헛선생

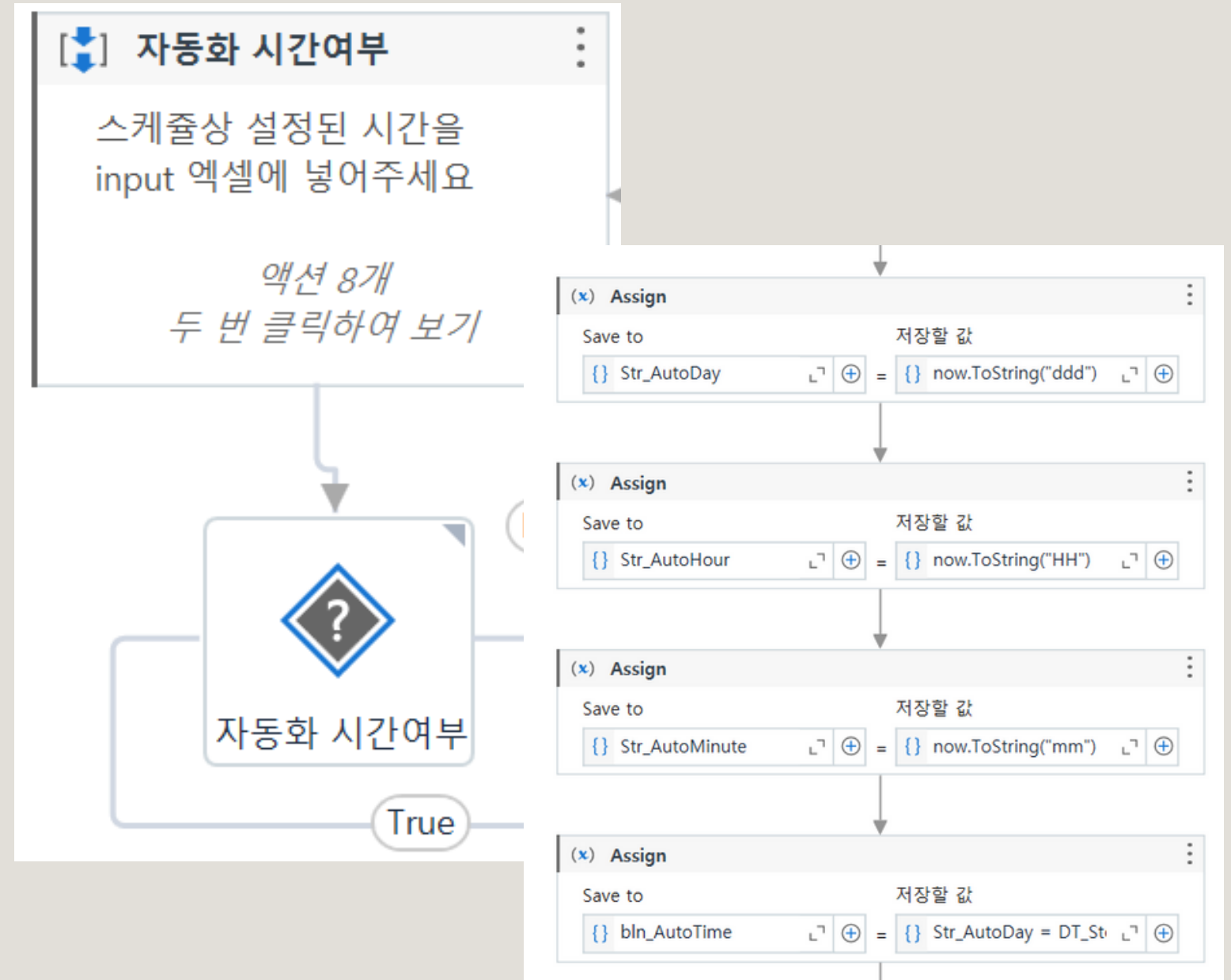
03. 수행 절차 및 방법

- 스케줄 자동화



<스케줄에 등록한 시간을 입력>

- Uipath 스케줄러에 등록되어 있는 요일, 시, 분을 Input시트의 "시간설정"시트에 입력해주어서 자동으로 프로그램이 돌아갈 수 있도록 함.



<실행 자동화 로직>

- 현재시간을 받아서 스케줄에 등록된 시간이면 고정값으로 실행

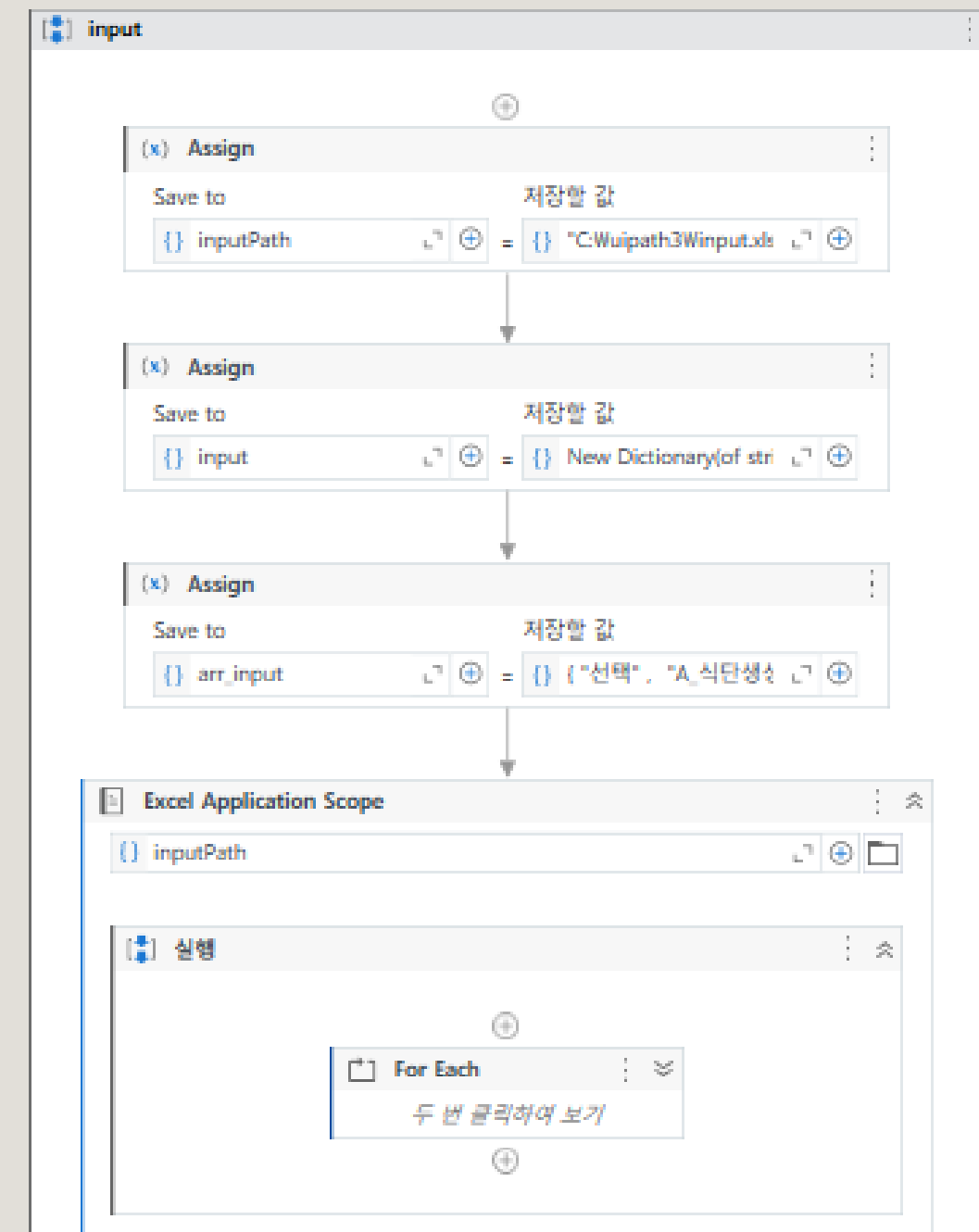
03. 수행 절차 및 방법

순번	요리명	해당 시트는 DT로 받기 때문에 두개의 항목만 작성 부탁드립니다.			
1	갈비찜				
2	계란찜				
날짜	언제	해당 시트는 DT로 받기 때문에 두개의 항목만 작성 부탁드립니다. 날짜 옆에는 yyyy-MM-dd 형식으로 언제 옆에는 아침, 점심, 저녁만 값으로 입력 부탁드립니다.			
2023-06-27	아침				
2023-06-27	점심				
2023-06-27	저녁				
Name	Value	설명			
choice	abcd	A = 1주일 식당생성			
		B = 식단 조회			
		C = 요리 레시피			
		D = 칼로리 차트 생성			
		E = 특정일 식단 변경			
▶	선택	A_식단생성	B_식단조회	C_요리레시피	D_칼로리차트

<Input 엑셀 파일 내부 변경 요청 사항 예시>

- 원하는 요청 내용을 Input 엑셀의 Sheet 별로 입력해서 수정

- Input

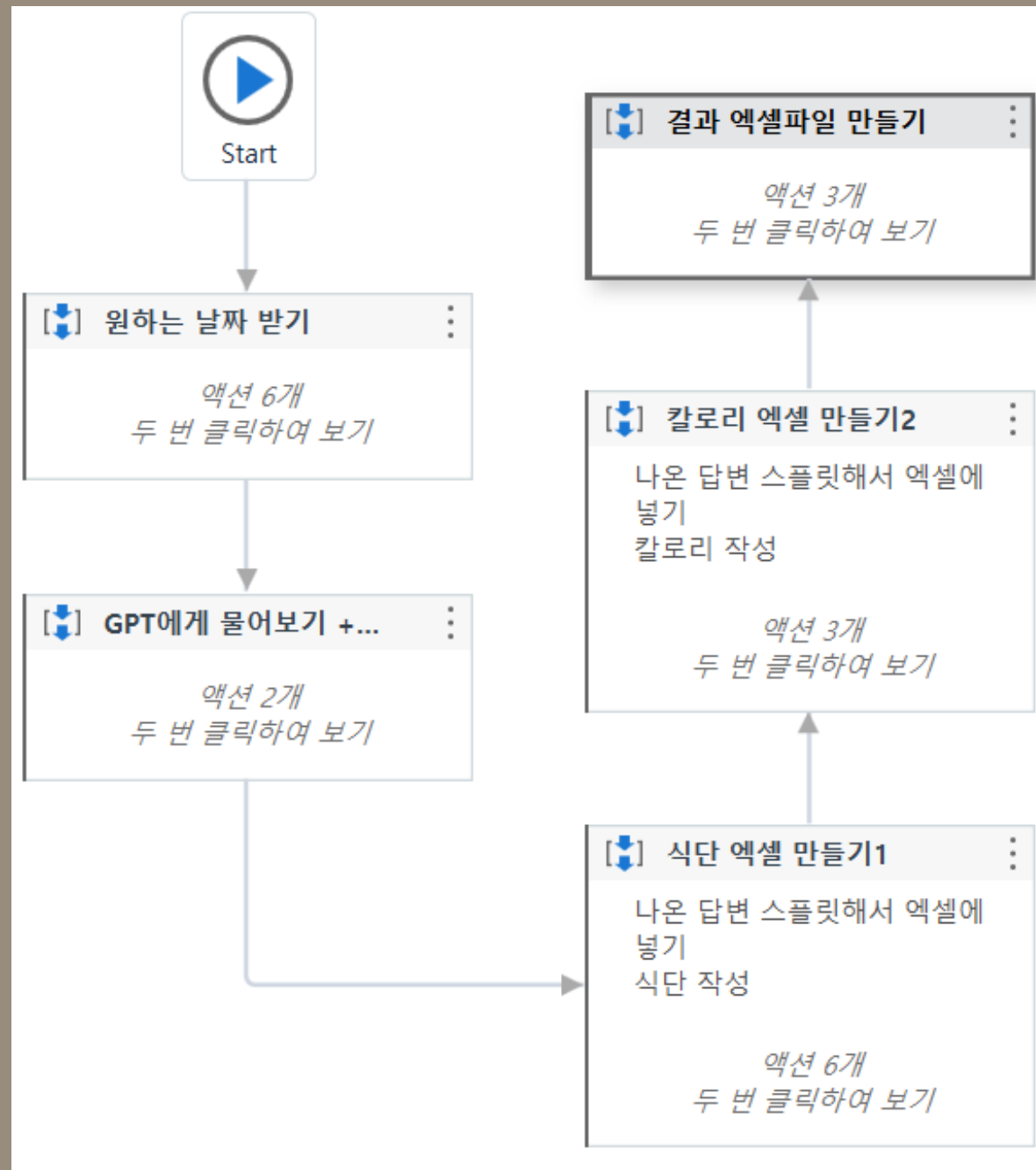


<Input파일 변수 지정>

- Input 엑셀로 요청 받은 값을 변경 사항 변수로 설정

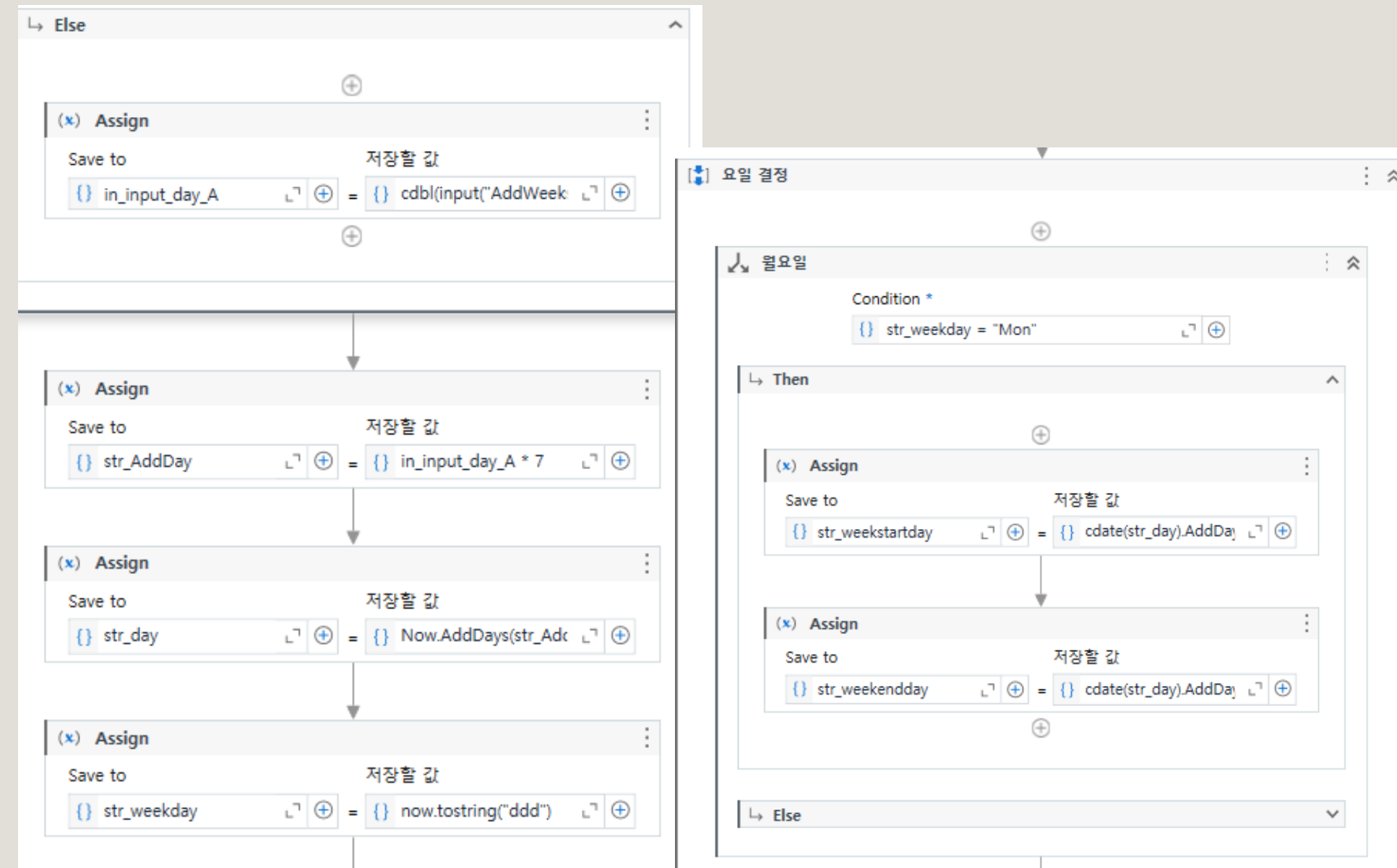
03. 수행 절차 및 방법

- 일주일 식단 구성 로직



<날짜에 맞는 일주일치 식단 구성 로직>

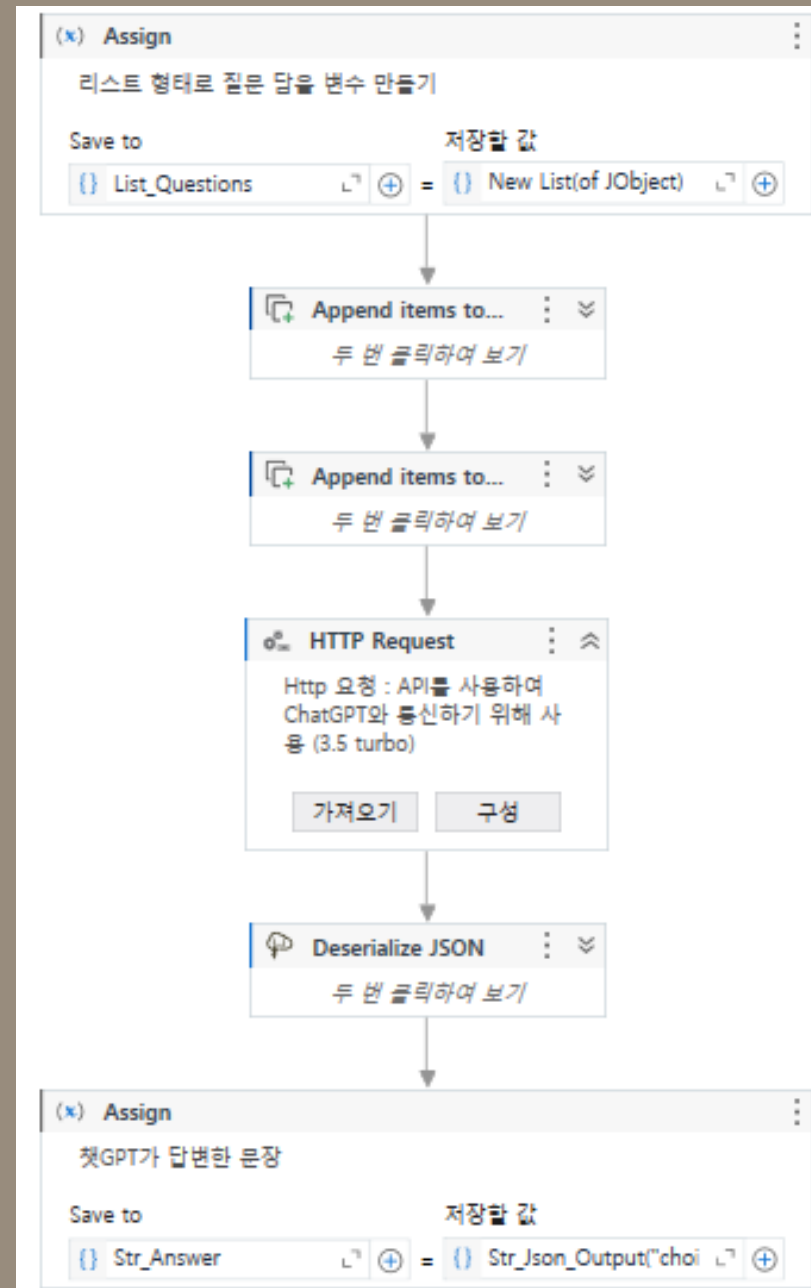
- Input에서 받아온 날짜를 자동으로 GPT에게 묻고 답을 엑셀로 만들어 저장



<시작일과 마지막일 계산 로직>

- AddDays메서드와 ToString을 활용해 요일을 확인한 후 오늘의 요일에 따라서 그 주의 시작일과 마지막일을 계산

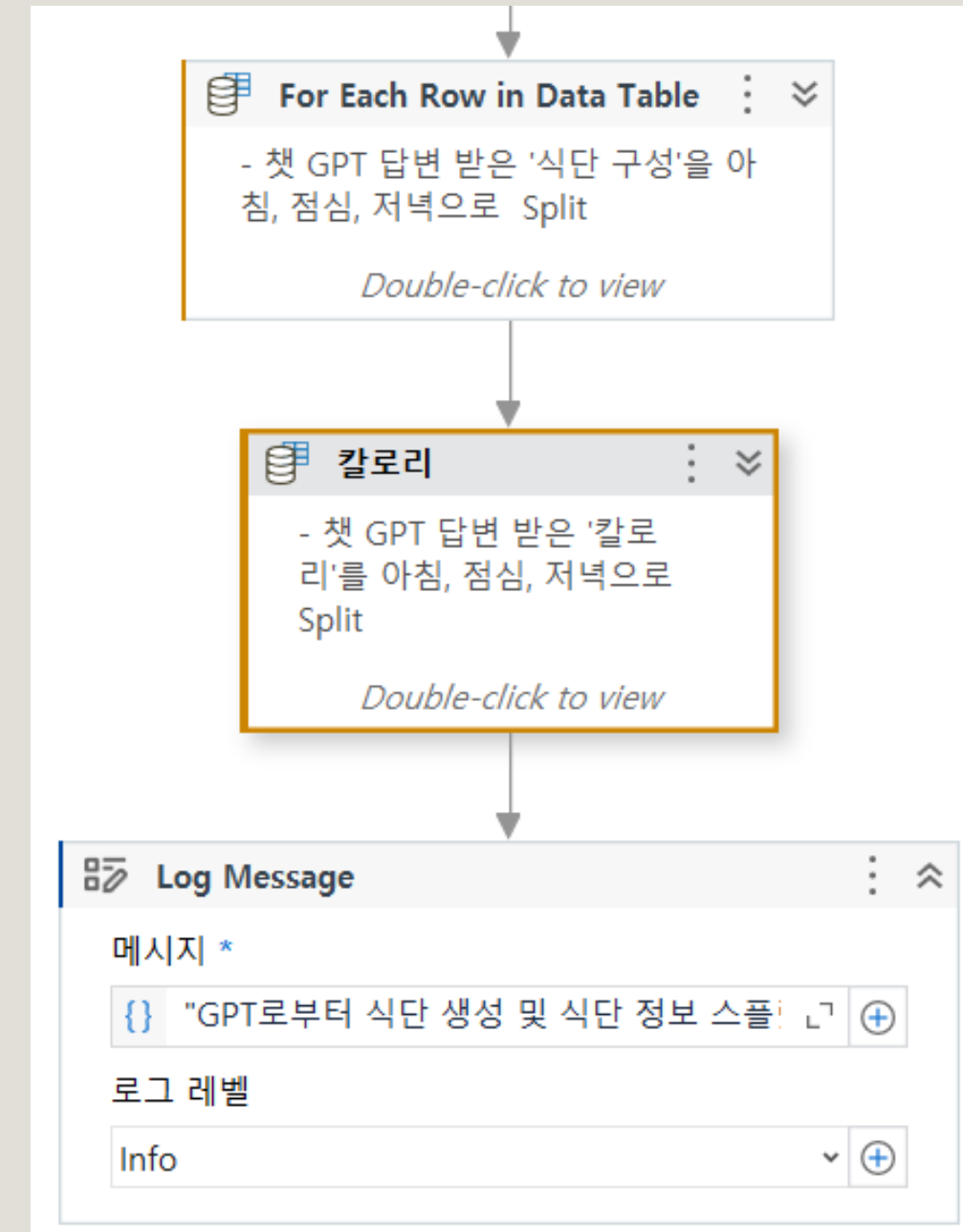
03. 수행 절차 및 방법



<챗 GPT에게 질문하고 답변받는 로직>

- 챗 GPT에게 식단 구성 요청하여 Json형태의 답변을 Str형태로 변경하여 변수에 저장

- 일주일 식단 구성 로직



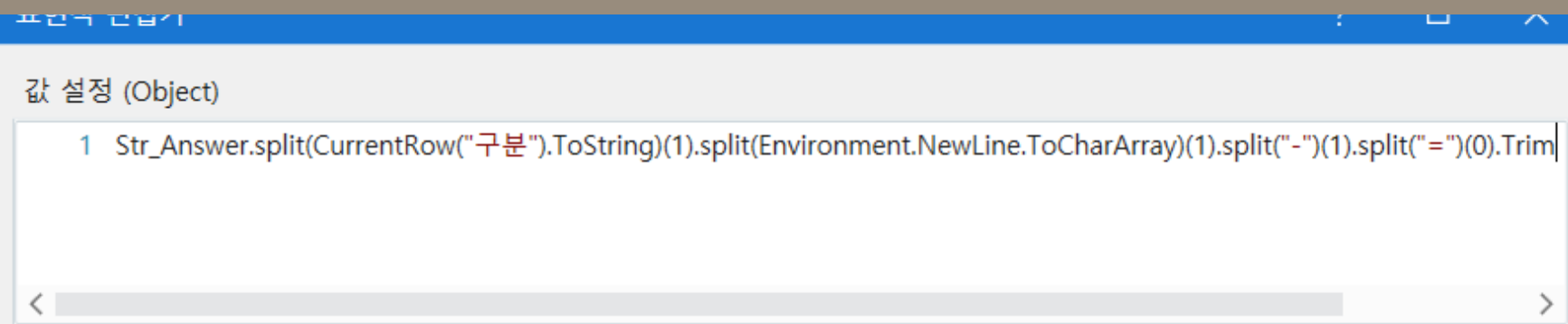
<챗 GPT에게 답변의 양식이 고정적으로 나오도록 질문>

- 챗 GPT에게 답변 받은 '식단 구성'에 Split 진행
- 시간대 별 식단 -> 일주일 및 일일 식단 구성에 사용
- 칼로리 -> 시각화에 사용

03. 수행 절차 및 방법

구분	아침	점심	저녁
월요일			
화요일			
수요일			
목요일			
금요일			
토요일			
일요일			

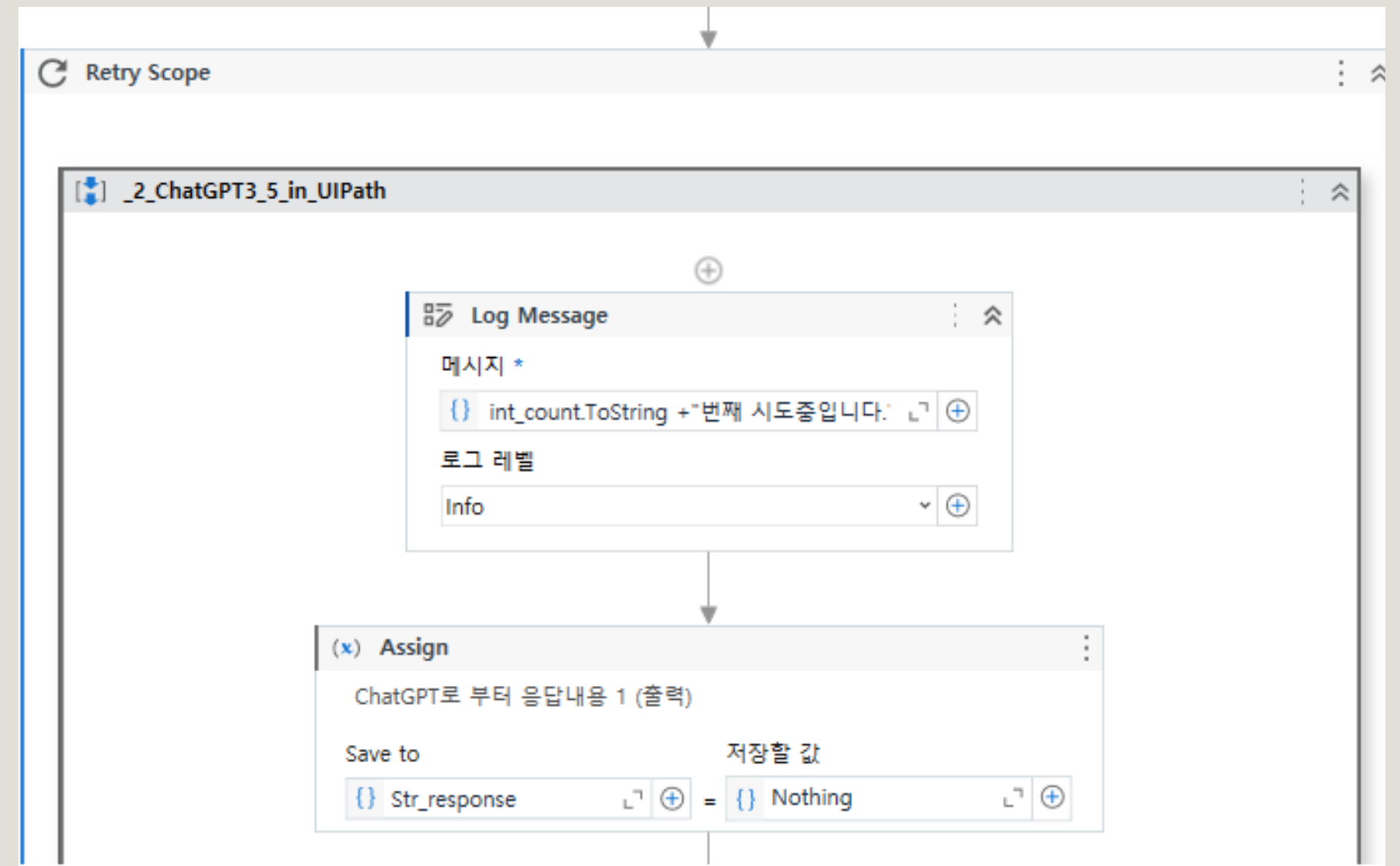
<식단, 칼로리 Template Form>



<Split 진행 로직>

- Template 형식에 맞게 요일별 및 식사 시간 별로 Split 진행해서 추후 Excel 에 담을 수 있게 데이터 저장

- 일주일 식단 구성 로직

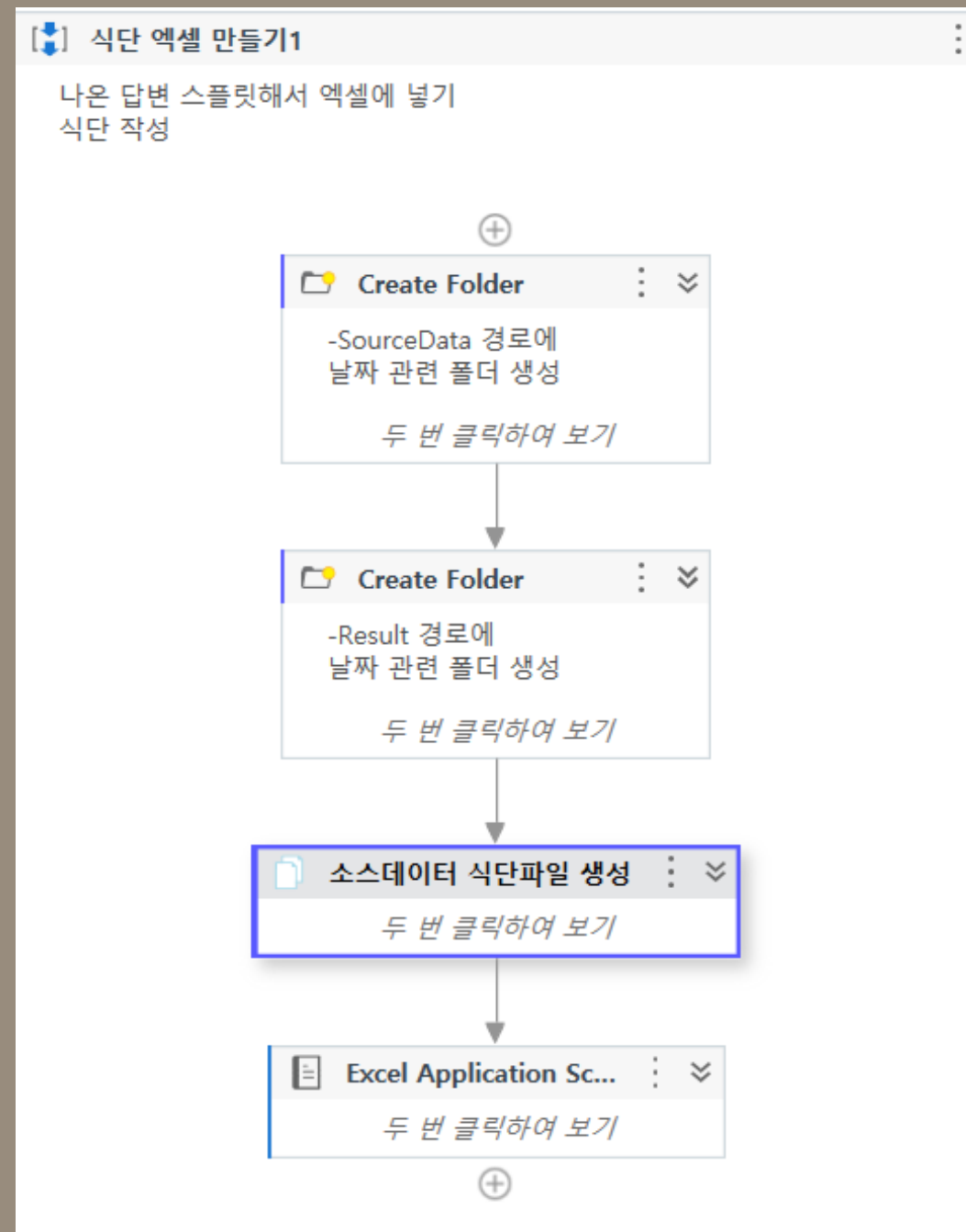


<안정성을 높이는 과정>

- 간혹 원하는 형태로 답변이 나오지 않아 스플릿과정에서 에러가 나는 경우가 있어 Retry로 안정성을 높임

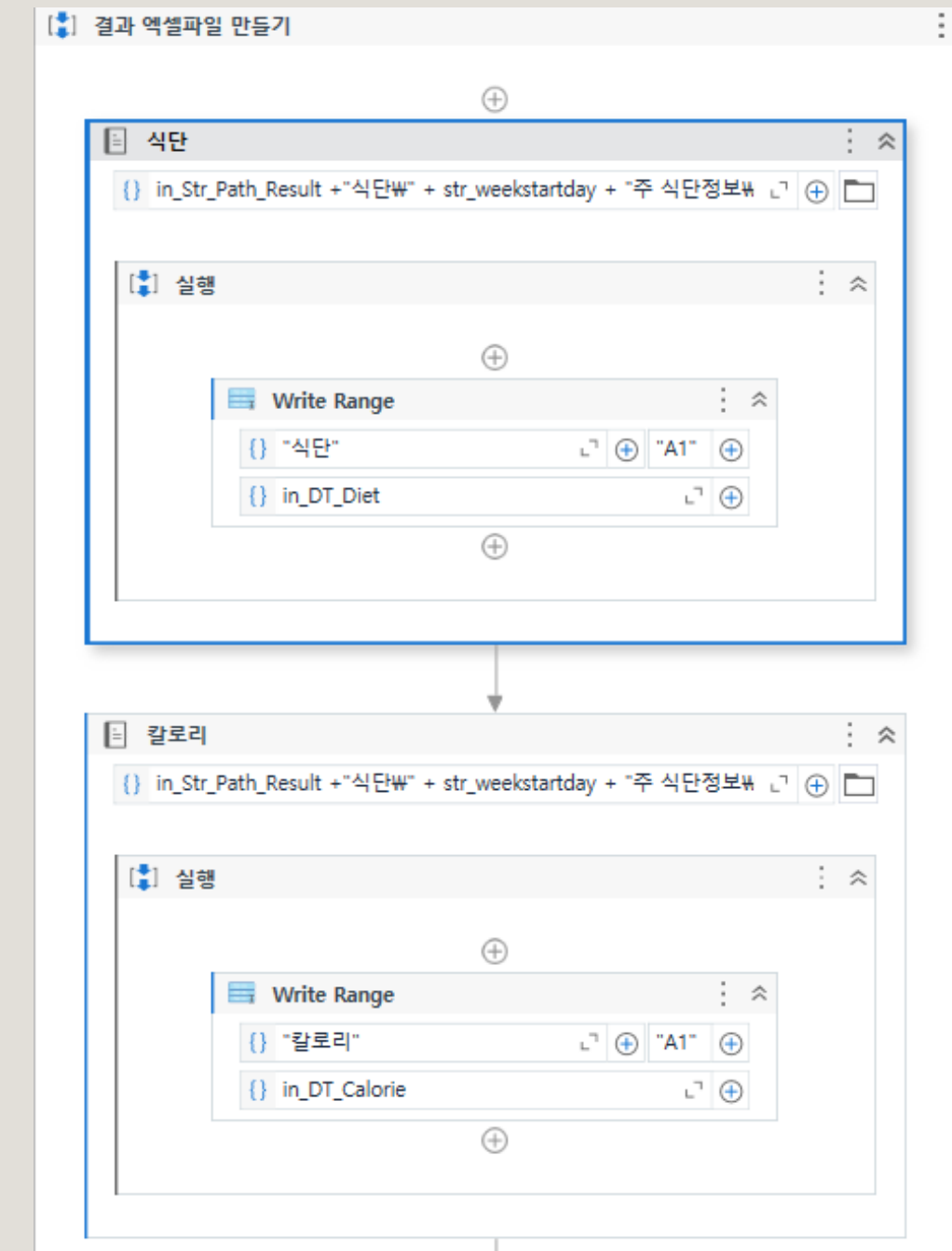
03. 수행 절차 및 방법

- 일주일 식단 구성 로직



<챗 GPT가 추천해준 식단을 저장하는 로직>

- SourceData/Result 파일내 날짜 폴더 생성
Template 복사

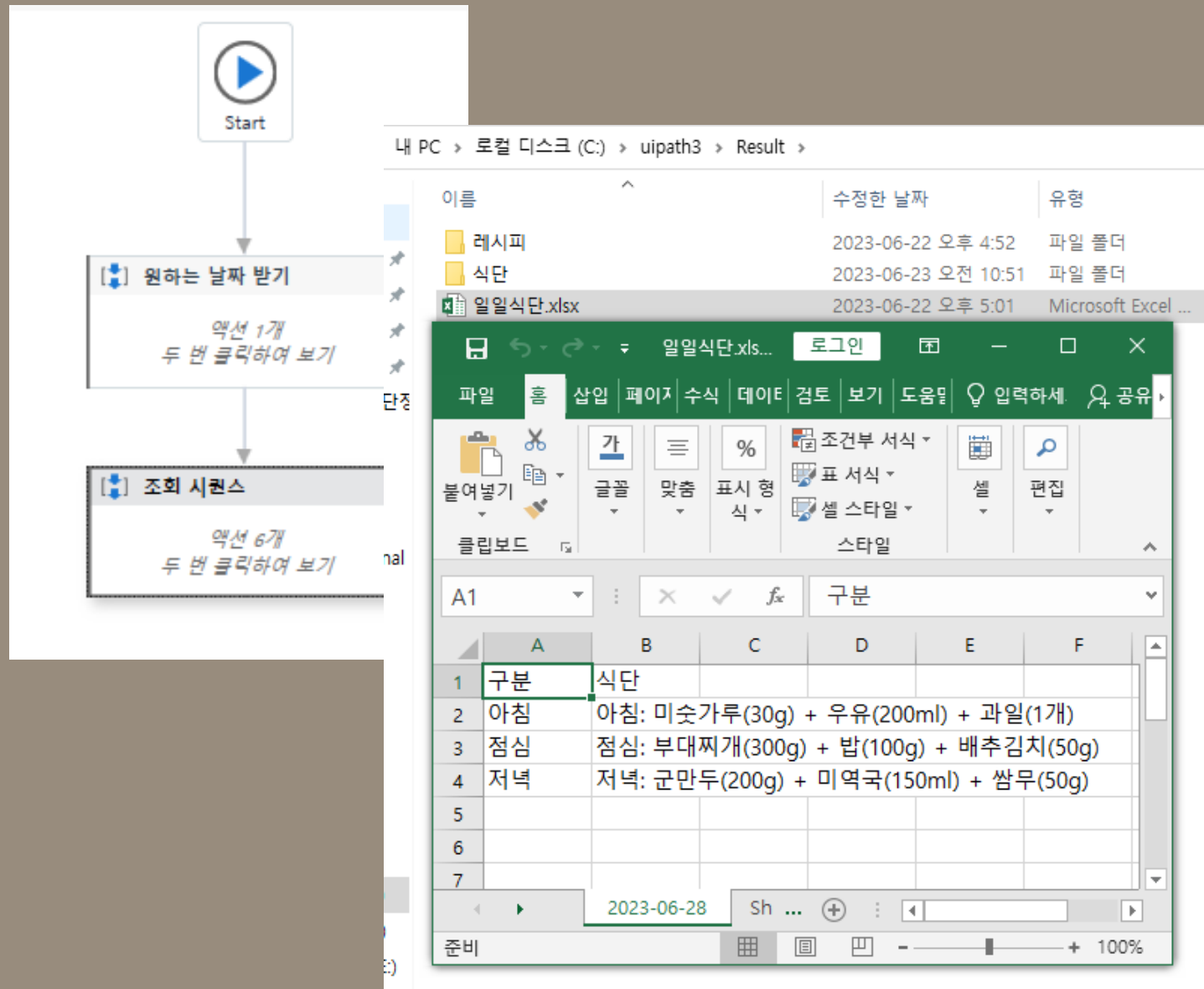


<정리된 내용을 .xlsx파일에 저장하는 로직>

- 변수를 활용한 경로로 자동분류 저장 가능

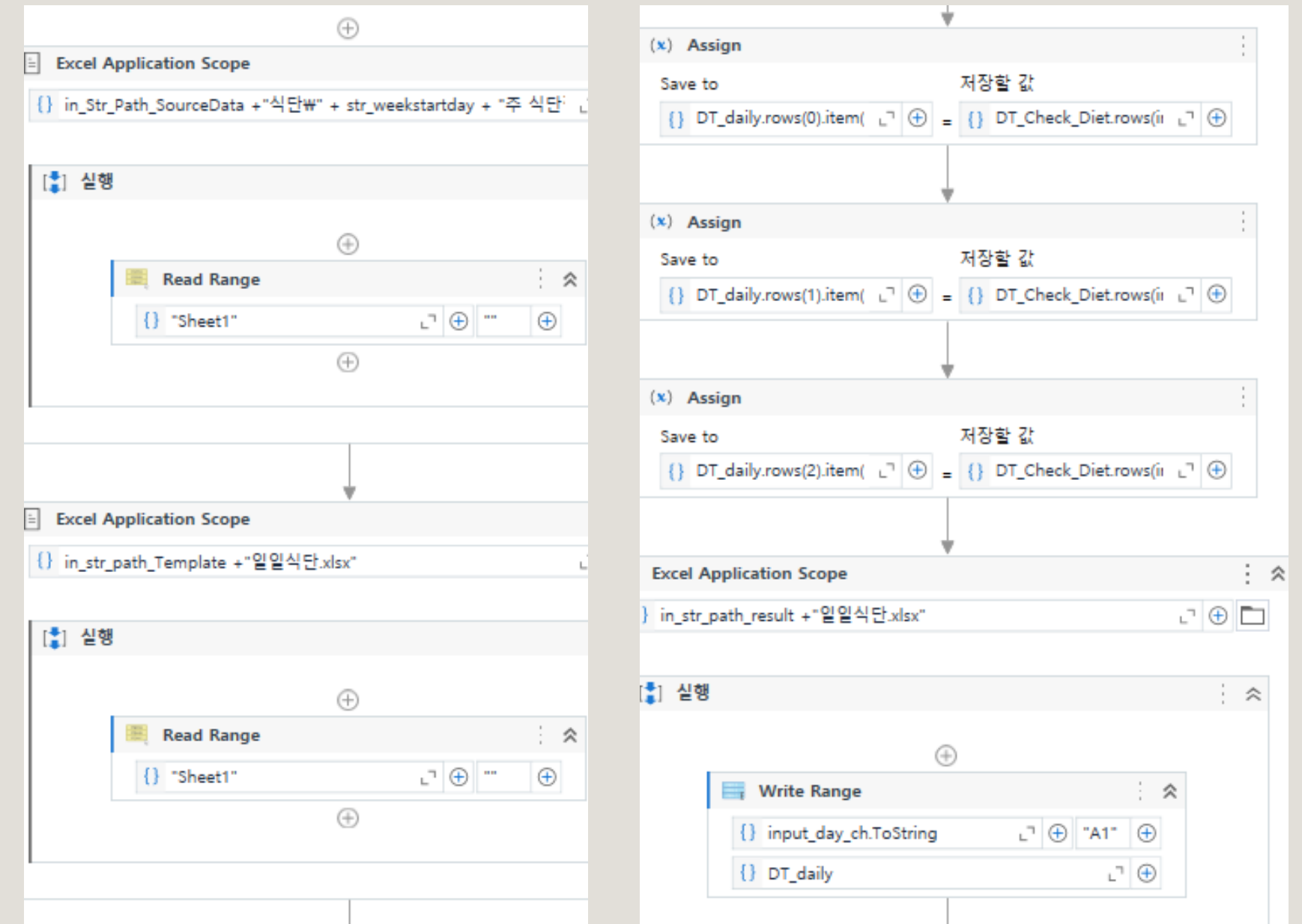
03. 수행 절차 및 방법

- 일일 식단 확인 로직



<일일식단을 엑셀에 저장>

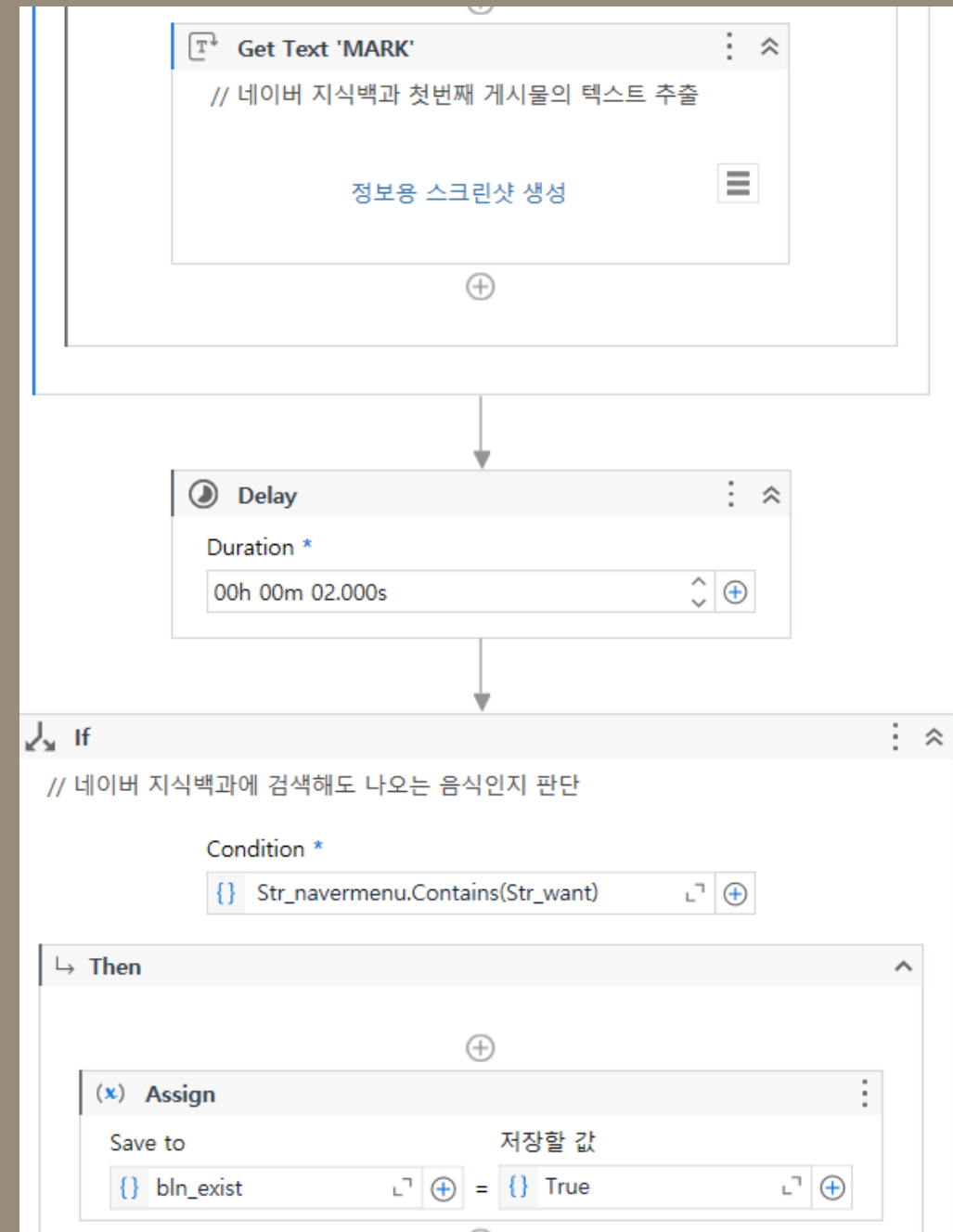
- 원하는 날짜의 식단을 조회해서 result 폴더의 일일식단 엑셀에 입력된 시트별로 저장



<원하는 날짜의 식단정보를 작성>

- 팜플렛과 원하는 날짜가 포함된 주의 식단을 DataTable 형식으로 불러오고 아침 점심 저녁의 셀 정보를 그대로 받아옴

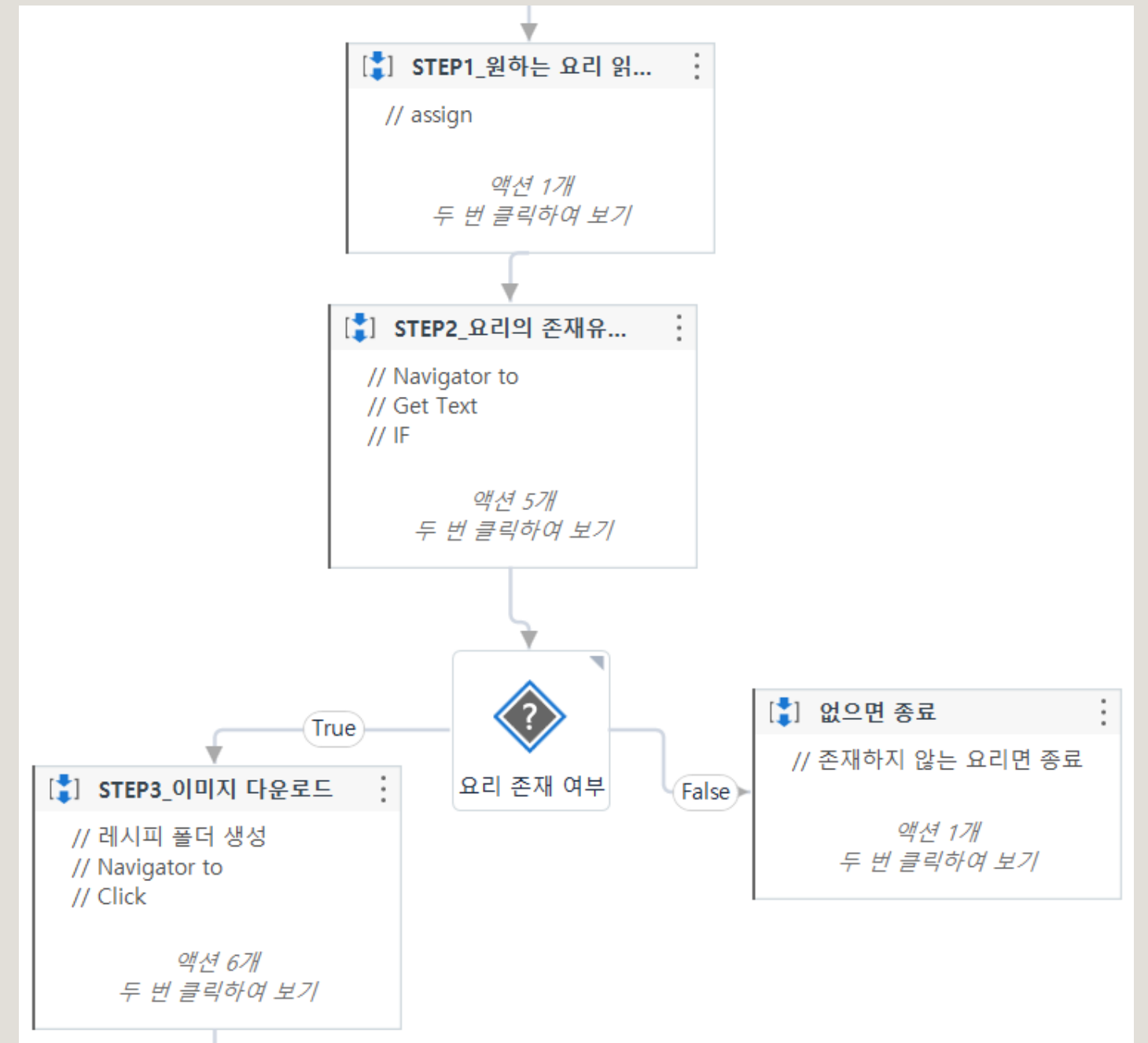
03. 수행 절차 및 방법



<IF문을 활용한 판별 로직>

- 없는 요리도 챗 GPT는 있는 것처럼 답변을 해주기 때문에 IF문으로 요리의 존재 유무 판단

- 레시피 검색



<flow decision을 활용>

- 존재 하지 않는 요리라 판단되면 로직 종료

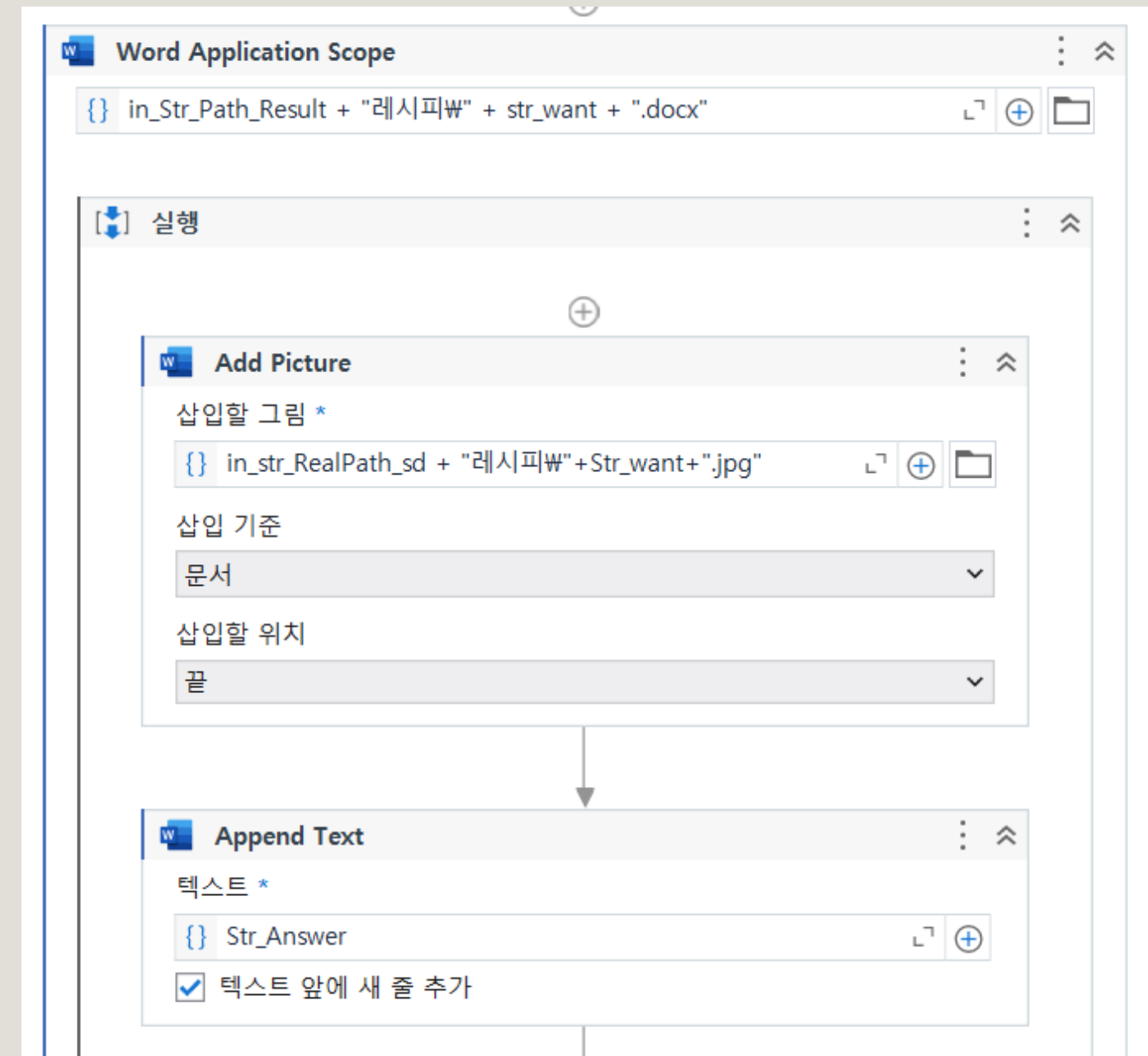
03. 수행 절차 및 방법



<이미지 필터링 설정 로직>

- 다운 받을 요리 이미지의 정확도 향상을 위해 검색 옵션을 설정

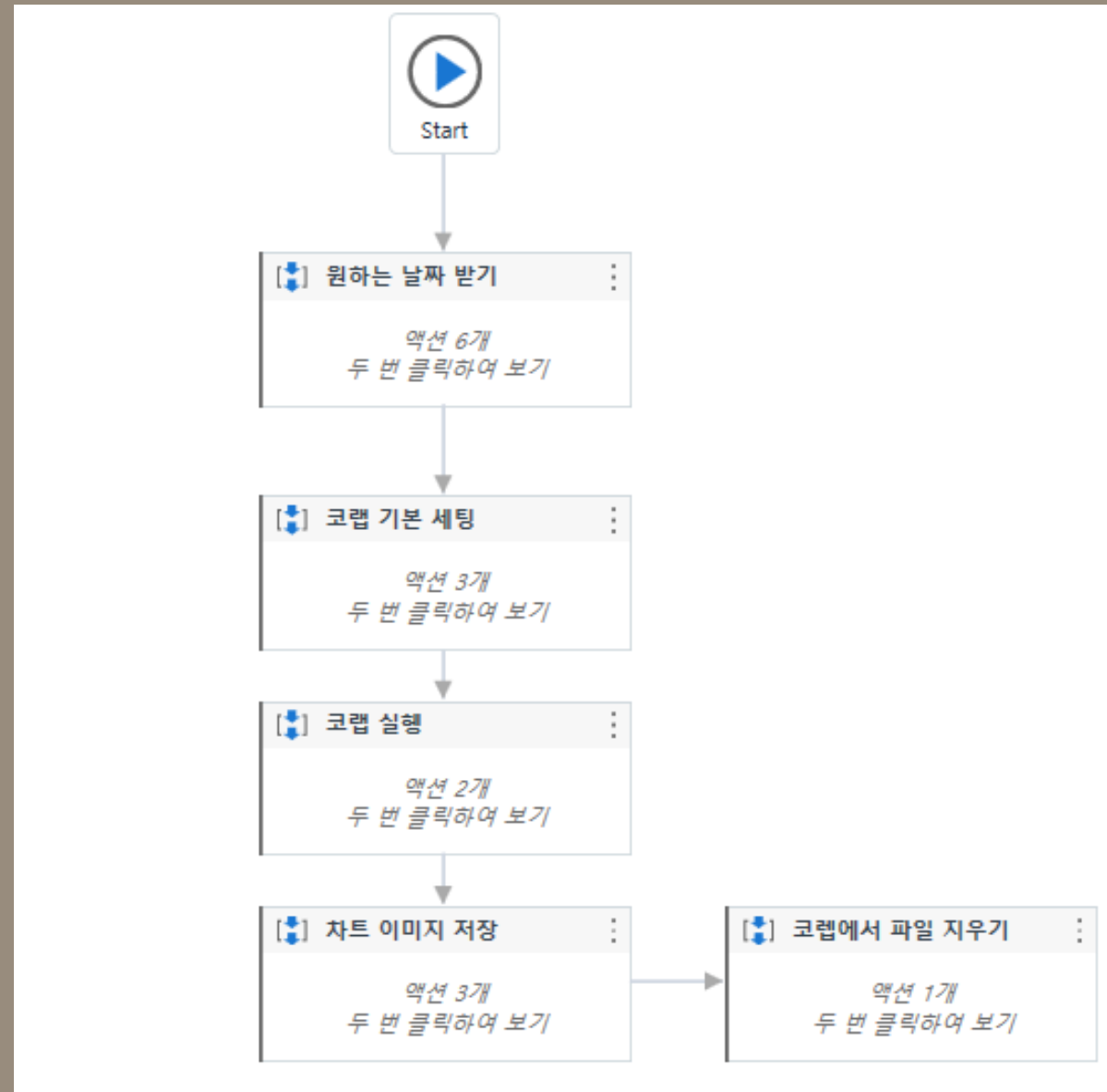
- 레시피 검색



<Word 파일 생성하는 로직>

- 다운 받은 이미지와 챗 GPT에게 전수 받은 레시피를 Word로 저장

03. 수행 절차 및 방법



<일별 칼로리의 전체적인 플로우>

- 원하는 주를 선택 받은 후, 세팅되어 있는 CoLab을 접속하여 차트 및 그래프를 다운 받음.
- 그 후 반복작업시 오류방지를 위해 업로드 된 엑셀 및 그림 삭제

- 칼로리 비교

```
1 # 일별 칼로리
2 df_melted = df.melt(id_vars='구분', var_name='시간대', value_name='값')
3
4
5 fig = go.Figure()
6 for 시간대 in ['아침', '점심', '저녁']:
7     시간대_df = df_melted[df_melted['시간대'] == 시간대]
8     fig.add_trace(go.Bar(x=시간대_df['구분'], y=시간대_df['값'], name=시간대))
9 fig.update_traces(text=시간대_df['값'], textposition='auto')
10
11 fig.update_layout(barmode='stack', title='일별 칼로리')
12 pio.write_image(fig, "/content/일별 칼로리.PNG", width=400, height=800)
13 fig.show()
14
15

1 #요일별 총 칼로리
2
3
4 df['일일 총 칼로리'] = df[['아침', '점심', '저녁']].sum(axis=1)
5 fig = px.bar(df, x='구분', y='일일 총 칼로리', color='구분', title='요일별 총 칼로리')
6 pio.write_image(fig, "/content/요일별 총 칼로리.PNG", width=800, height=600)
7 fig.show()
8

1 #요일별 평균 칼로리
2
3 df['일일 평균 칼로리'] = df[['아침', '점심', '저녁']].mean(axis=1)
4 fig = px.bar(df, x='구분', y='일일 평균 칼로리', color='구분', title='요일별 평균 칼로리')
5 pio.write_image(fig, "/content/요일별 평균 칼로리.PNG", width=800, height=600)
6 fig.show()
7

1 #시간대별 칼로리
2 df=pd.DataFrame(calorie_df)
3
4
5 df['아침합계'] = df['아침'].sum()
6 df['점심합계'] = df['점심'].sum()
7 df['저녁합계'] = df['저녁'].sum()
8
9 df = pd.melt(df, id_vars='구분', value_vars=['아침합계', '점심합계', '저녁합계'], var_name='시간대', value_name='칼로리')
10 fig = px.pie(df, values='칼로리', names='시간대', title='시간대별 칼로리 합계')
11 pio.write_image(fig, "/content/시간대별 평균 칼로리.PNG", width=800, height=600)
12 fig.show()
13
```

<그래프 작성시 필요한 코드가 미리 세팅되어 있다.>

- 컴퓨터 영향을 적게받고, 크롬 확장자 등록을 통해 간편하게 이용할 수 있는 CoLab을 파이썬 프로그램으로 선택
- 로직을 통해 작성된 엑셀을 가져와서 이미지를 다운로드 받는 방식

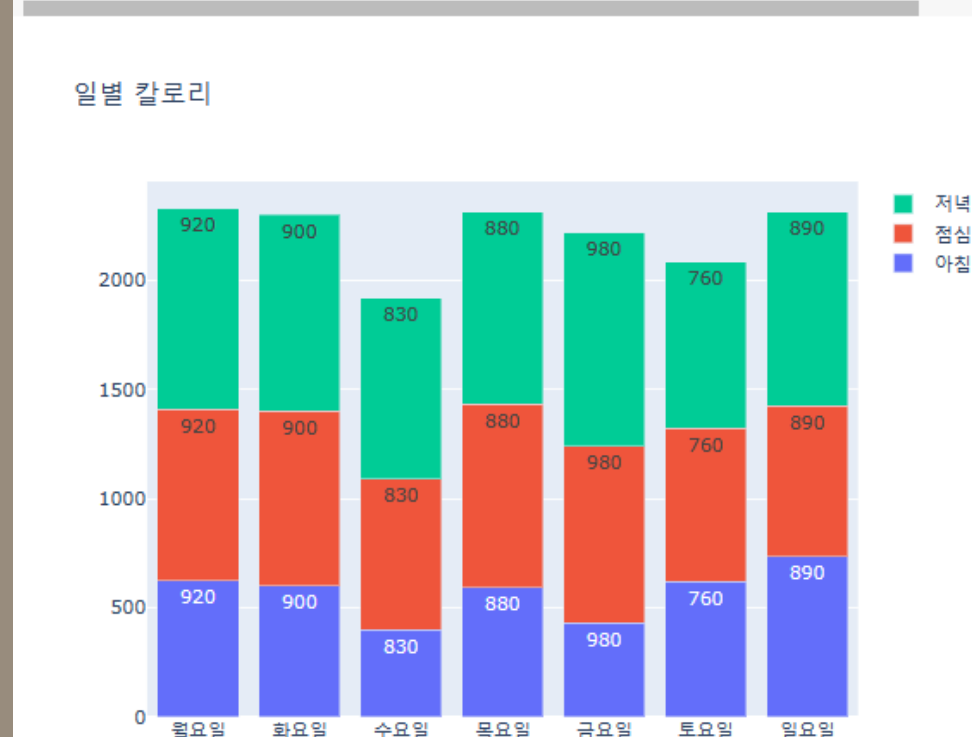
03. 수행 절차 및 방법

- 칼로리 비교

```
# 일별 칼로리
df_melted = df.melt(id_vars='구분', var_name='시간대', value_name='값')

fig = go.Figure()
for 시간대 in ['아침', '점심', '저녁']:
    시간대_df = df_melted[df_melted['시간대'] == 시간대]
    fig.add_trace(go.Bar(x=시간대_df['구분'], y=시간대_df['값'], name=시간대))
fig.update_traces(text=시간대_df['값'], textposition='auto')

fig.update_layout(barmode='stack', title='일별 칼로리')
pio.write_image(fig, "/content/일별 칼로리.PNG", width=400, height=800)
fig.show()
```



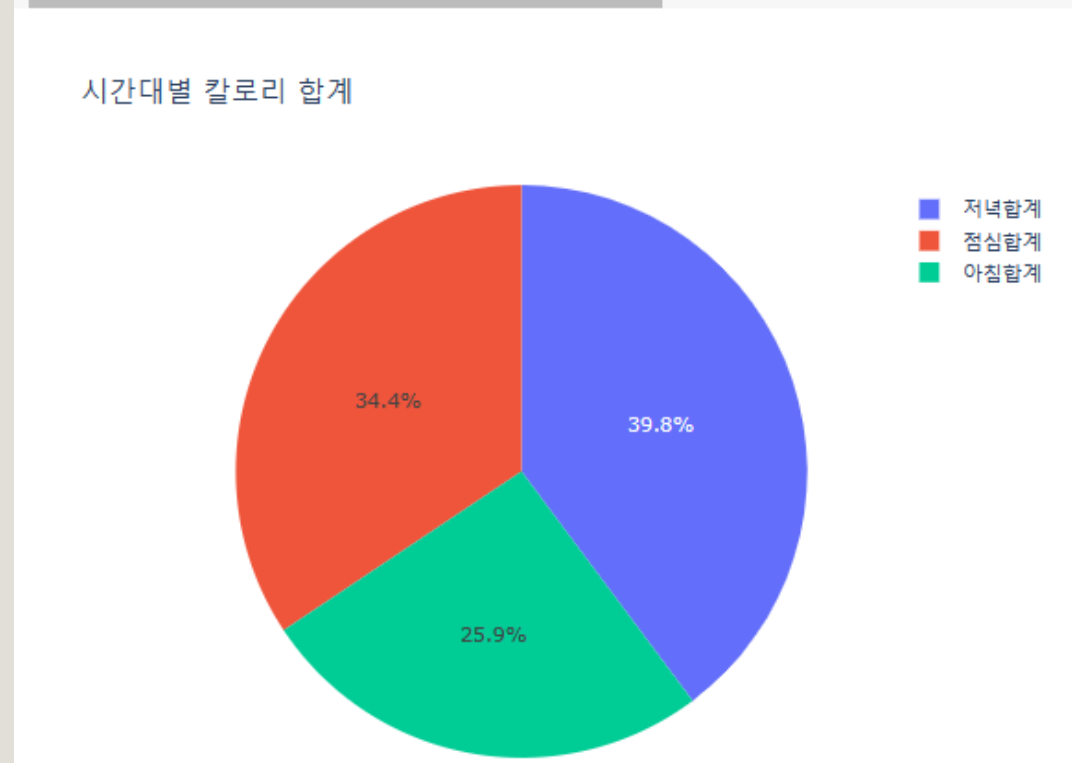
<일별 칼로리의 누적합 막대그래프>

- melt 를 활용해 데이터프레임을 '구분' 열을 기준으로 '시간대'와 '값' 열로 데이터가공

```
#시간대별 칼로리
df=pd.DataFrame(calorie_df)

df['아침합계'] = df['아침'].sum()
df['점심합계'] = df['점심'].sum()
df['저녁합계'] = df['저녁'].sum()

df = pd.melt(df, id_vars='구분', value_vars=['아침합계', '점심합계', '저녁합계'])
fig = px.pie(df, values='칼로리', names='시간대', title='시간대별 칼로리 합계')
pio.write_image(fig, "/content/시간대별 평균 칼로리.PNG", width=800, height=600)
fig.show()
```



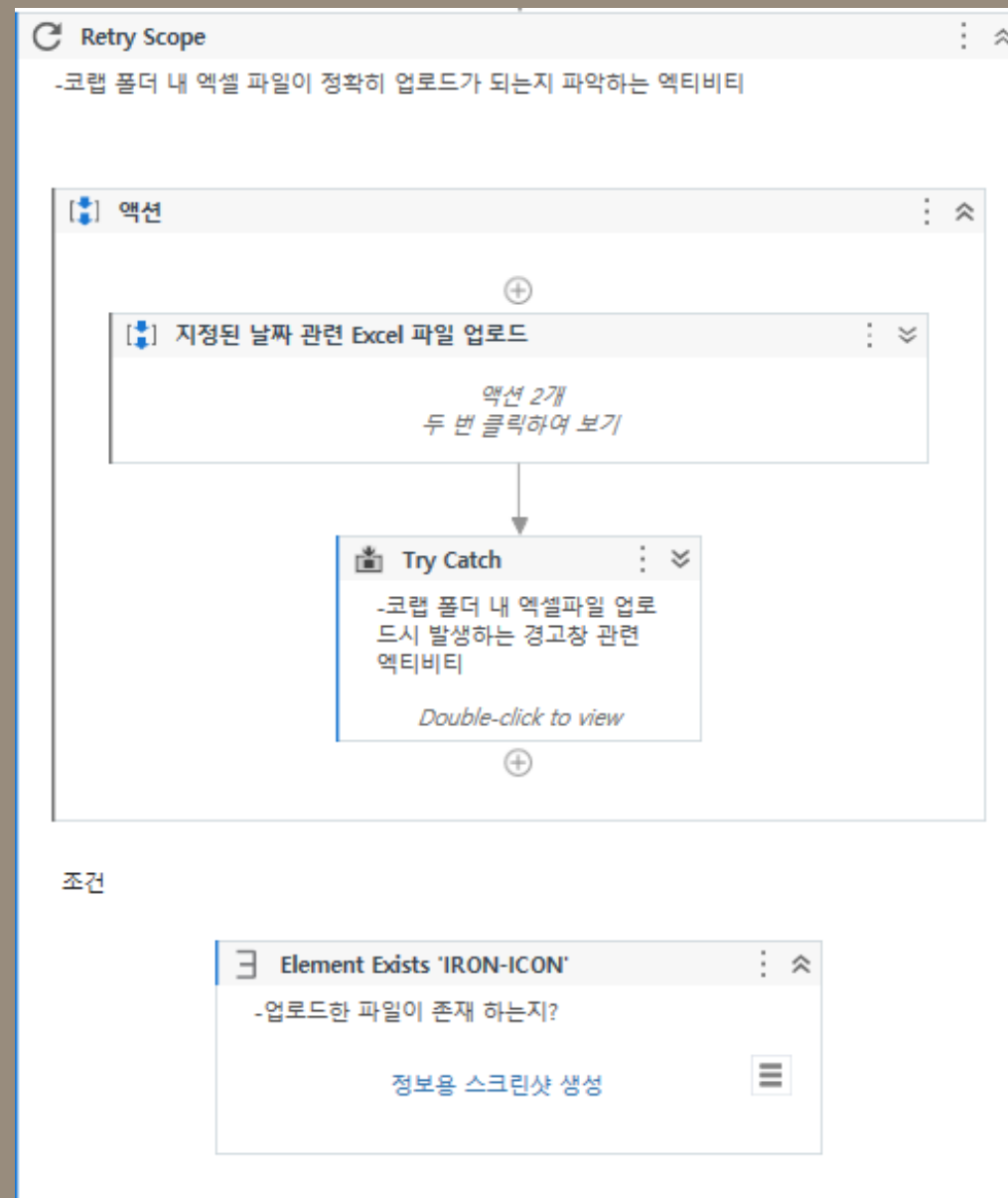
<아침/점심/저녁 칼로리 합의 파이차트>

- 각 열의 합계를 구해 새로운 열로 저장
- melt 를 활용해 새로운 데이터 열로 데이터가공

Plotly를 활용해서 칼로리 정보를 시각화하고 write_image를 활용해 이미지 자동저장

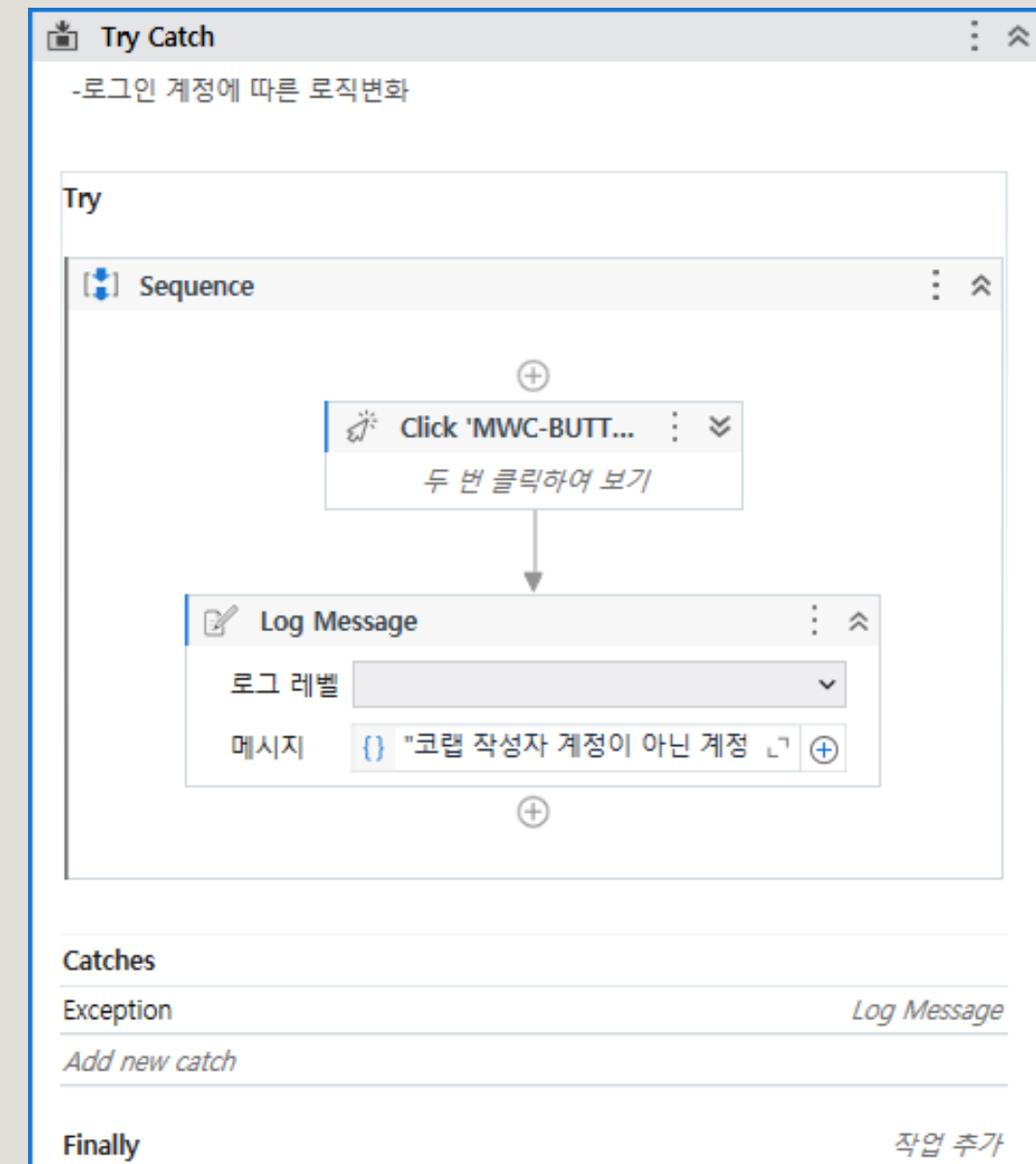
03. 수행 절차 및 방법

- 칼로리 비교(안정성 강화)



<retry 속에 trycatch를 사용한 로직>

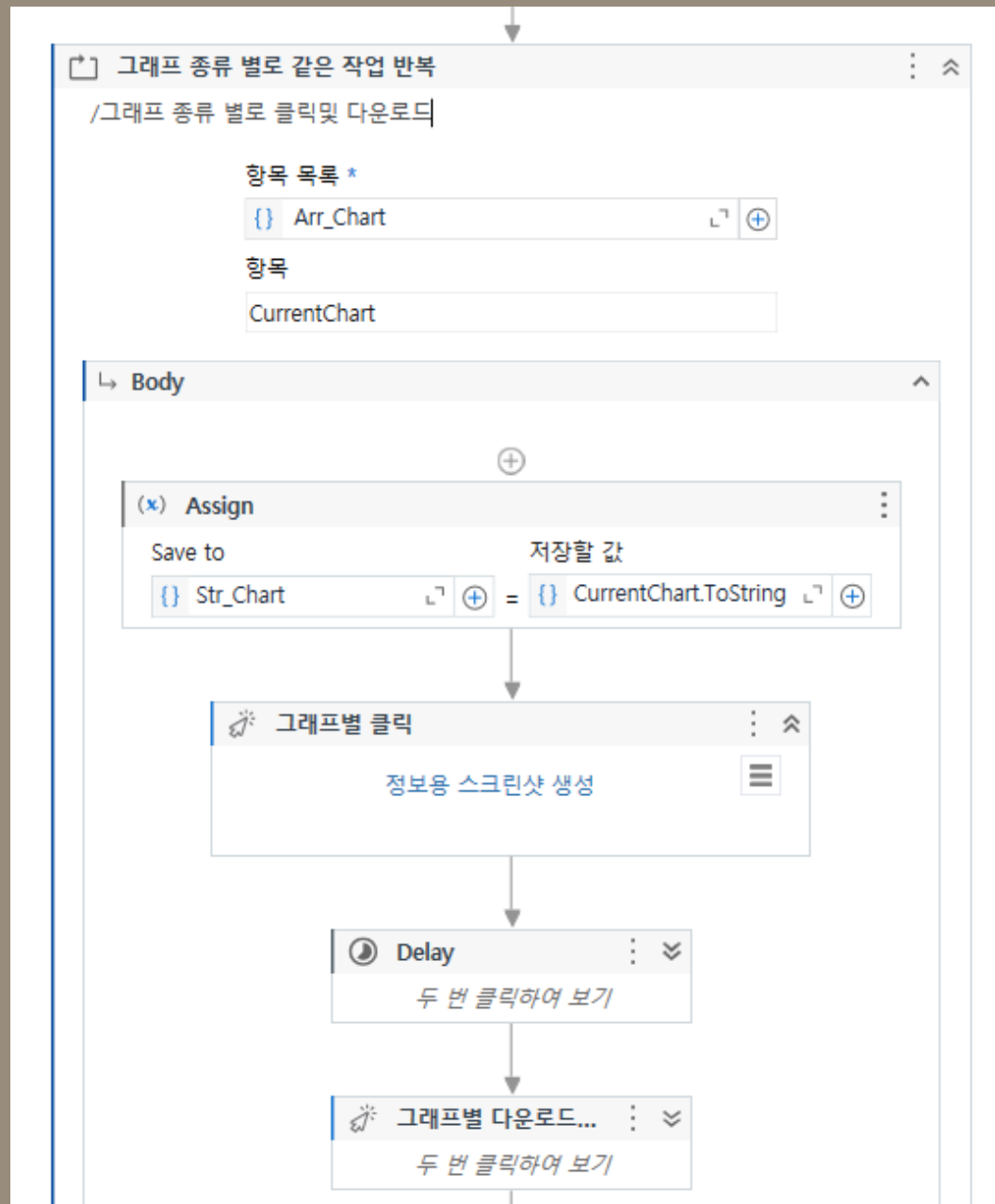
- 정확하게 파일이 업로드 위해 retry를 활용
- retry 진행 중에도 사용자 환경에 따라 경고창이 발생하기에 trycatch를 추가 활용



<try와 catch-LogMessage를 사용한 로직>

- 사용자 환경에 따라 경고창 변수가 존재하여 try catch를 활용
- 진행된 로직에 따른 LogMessage를 활용하여 진행 상황 파악 가능

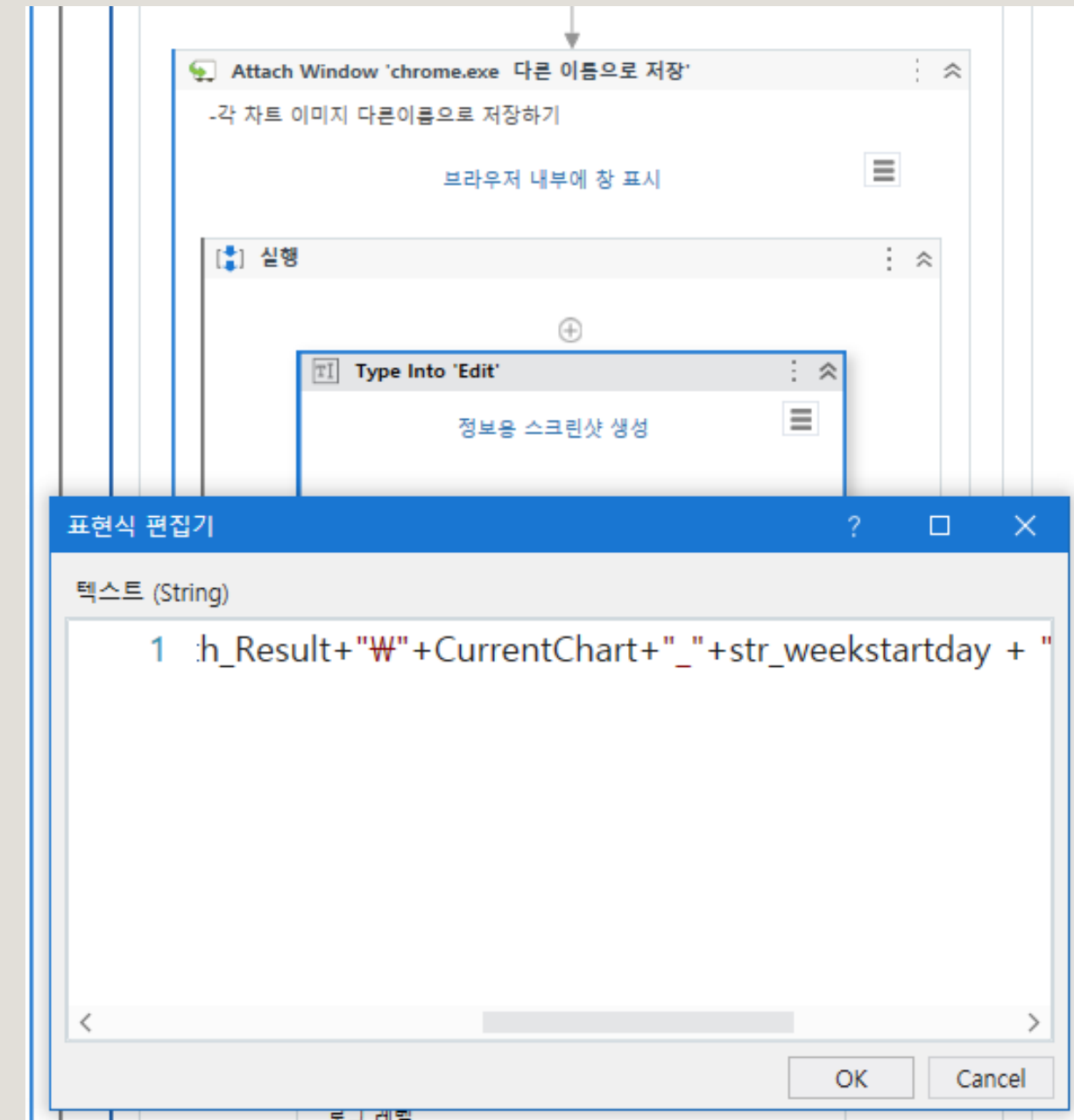
03. 수행 절차 및 방법



<반복문을 활용한 그래프 다운로드 로직>

- 그래프별로 변수에 담아 반복문으로 진행
- Selector에 변수를 활용해 반복문에 맞게 클릭

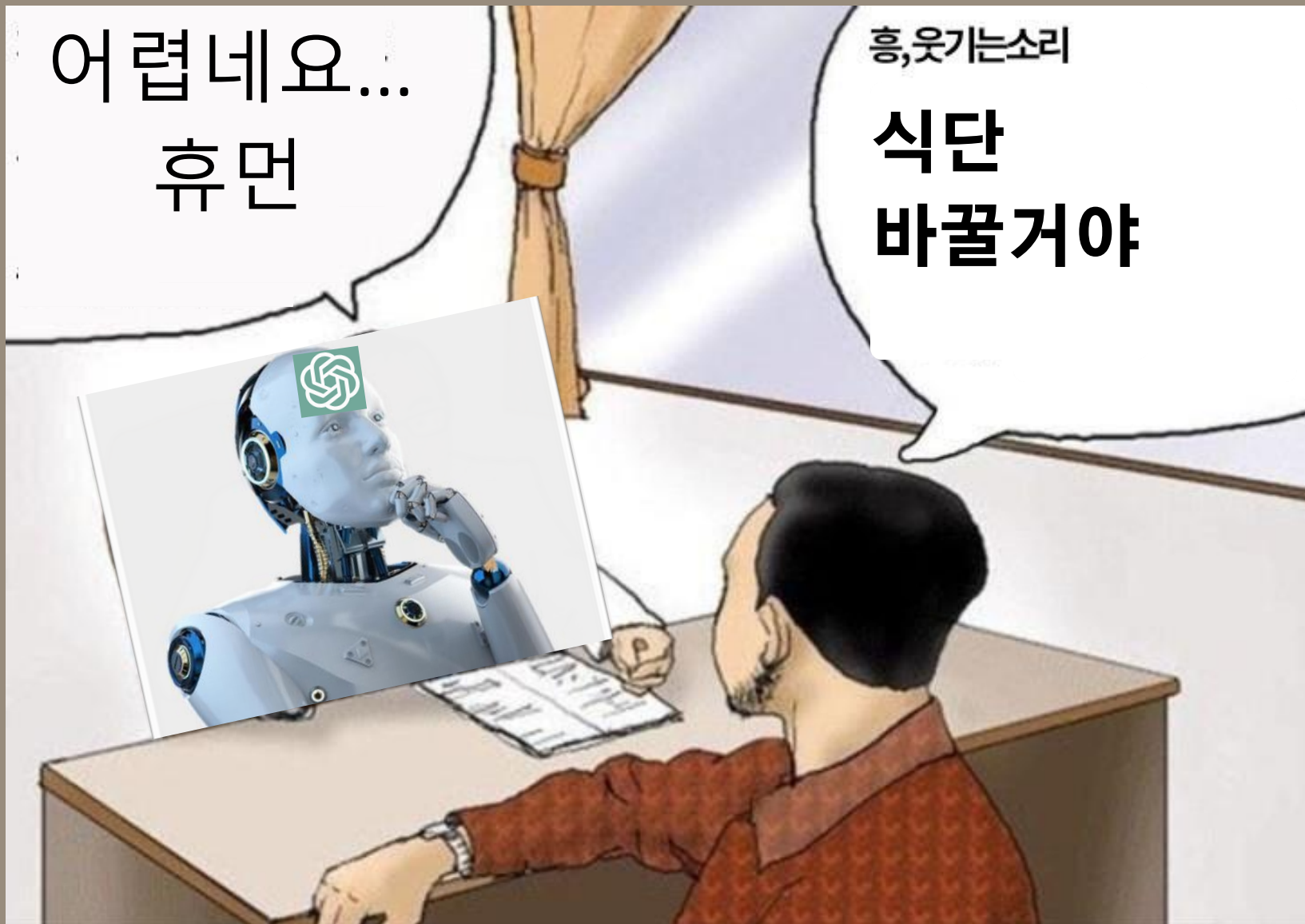
- 칼로리 비교(그래프 저장)



<다른 이름으로 저장 로직>

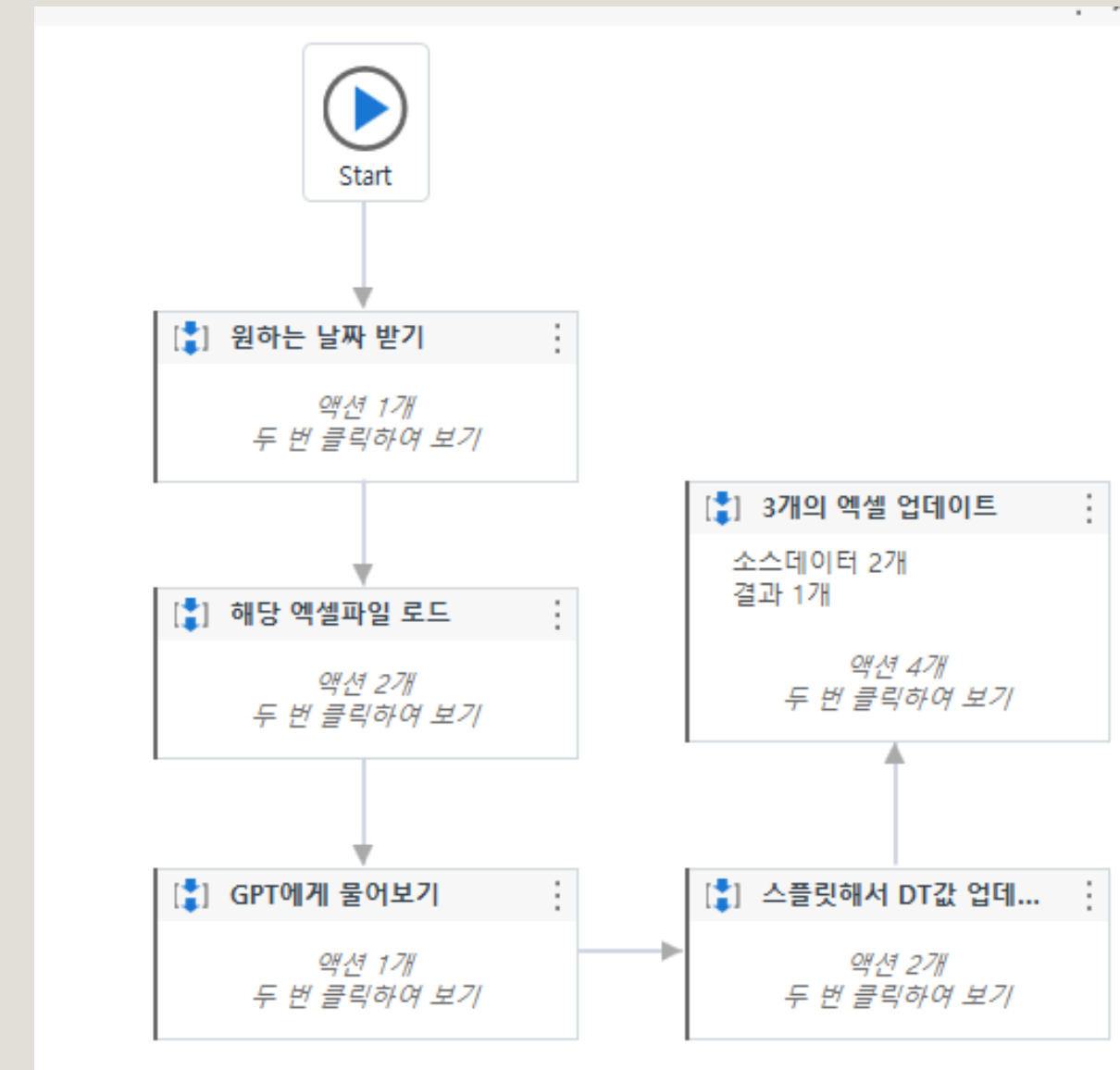
- 파일명에 날짜변수를 적음으로써 덮어쓰기 방지 및 구분

03. 수행 절차 및 방법



<집밥헛선생과 참여하게 대립하는 휴먼>

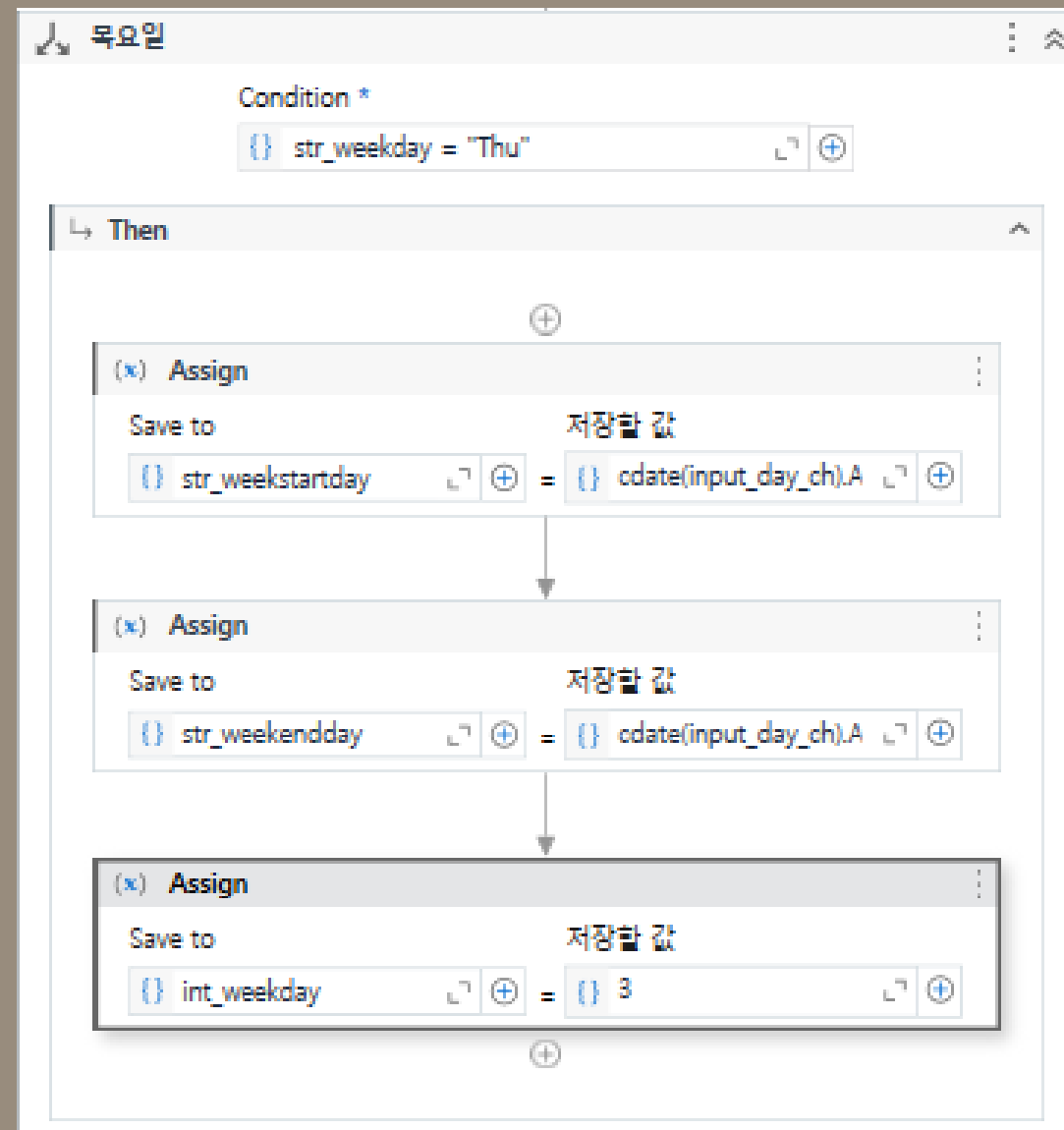
- 식단변경 로직



<식단 변경 로직>

집밥헛선생이 추천해준 식단이 맘에 들지 않을 경우 재추천을 요구할 수 있다.

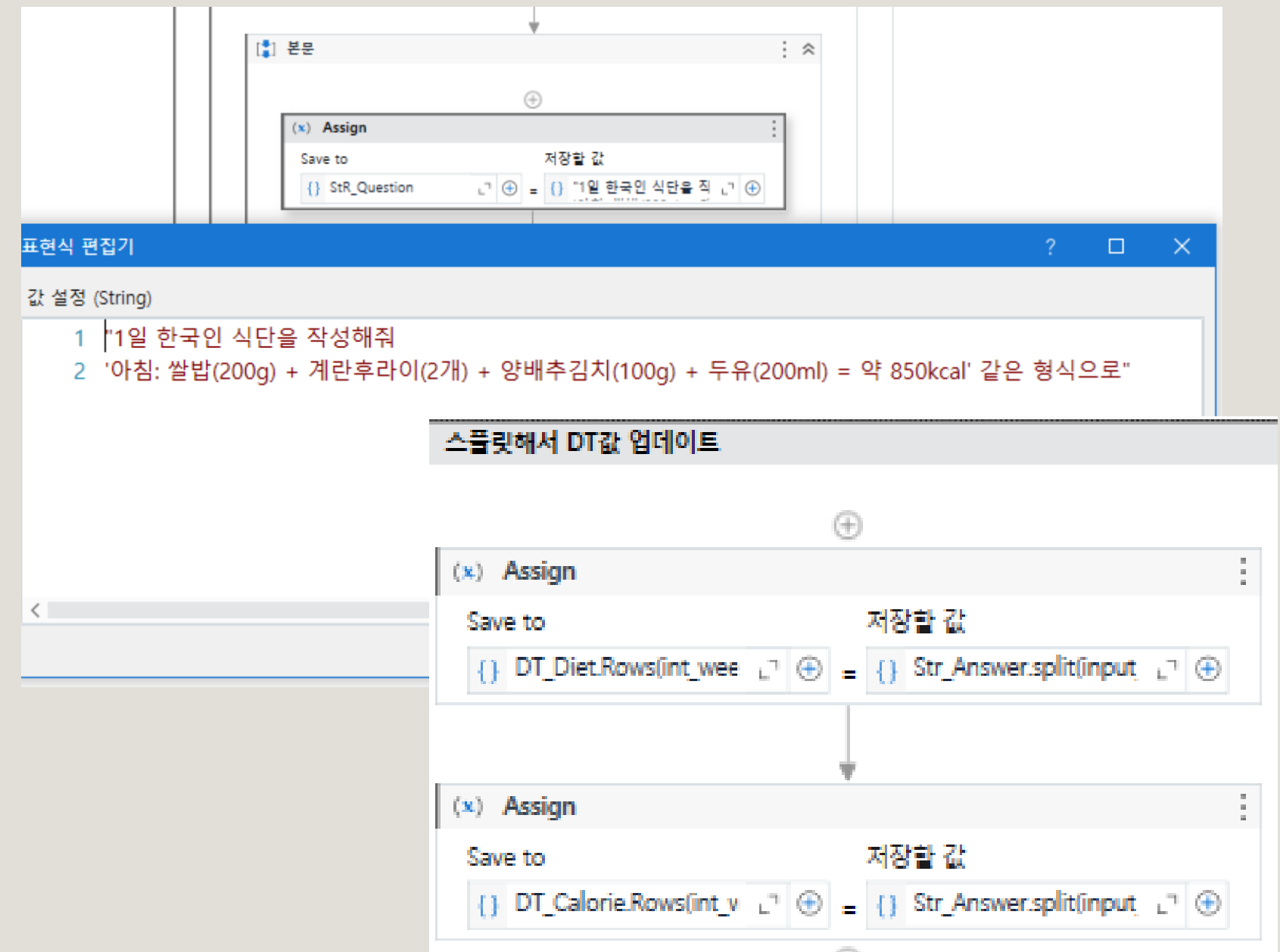
03. 수행 절차 및 방법



<날짜 입력시 월요일(0)~일요일(6)까지 숫자로 받는 Int_weekday 변수를 선언을 추가함>

- 주 단위로 생성했던 다른 로직과는 다르게 1일 단위의 선택이 필요하여 해당 변수 추가

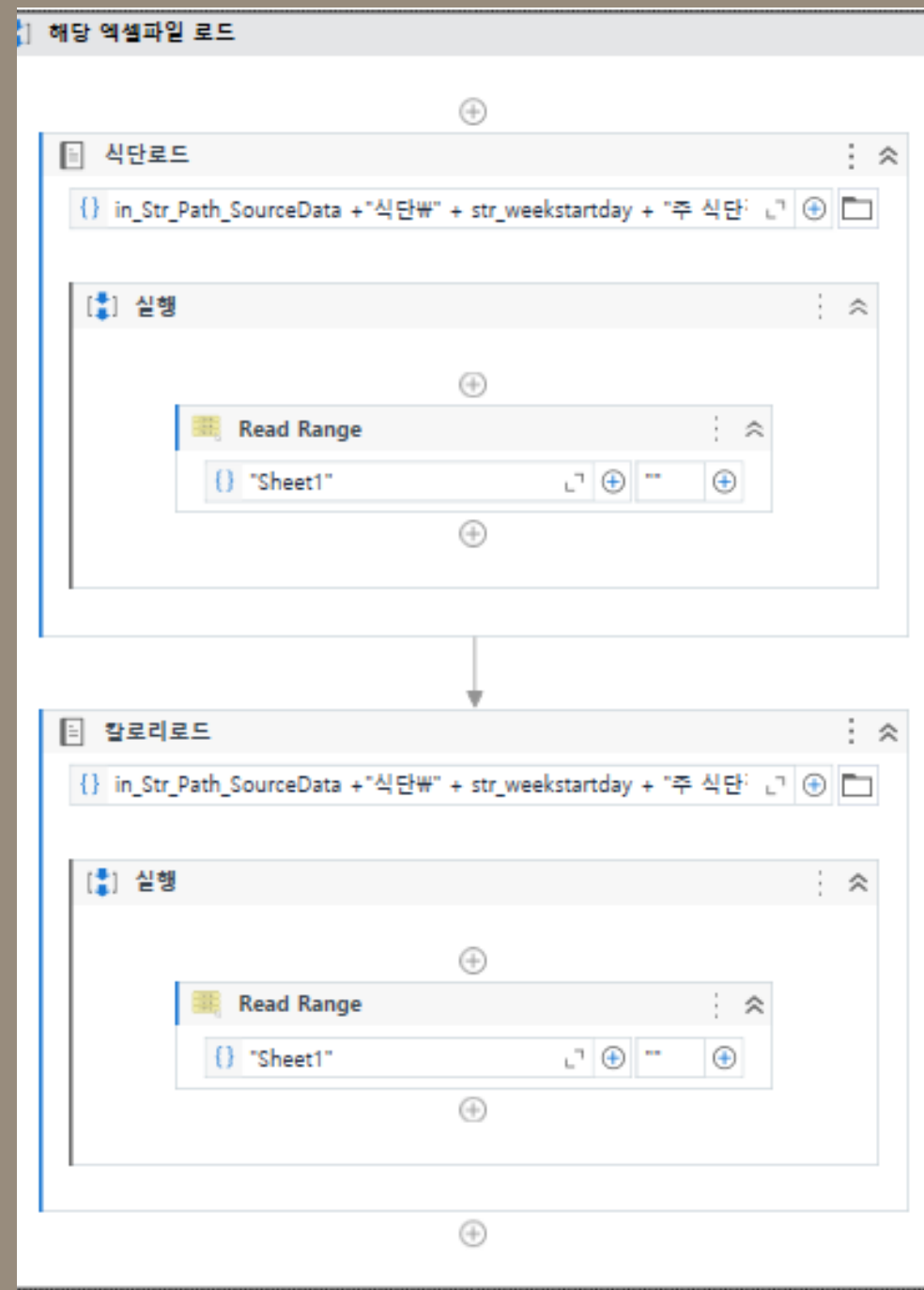
- 식단변경 로직



<챗 GPT 질문 형식과 Split 진행 로직>

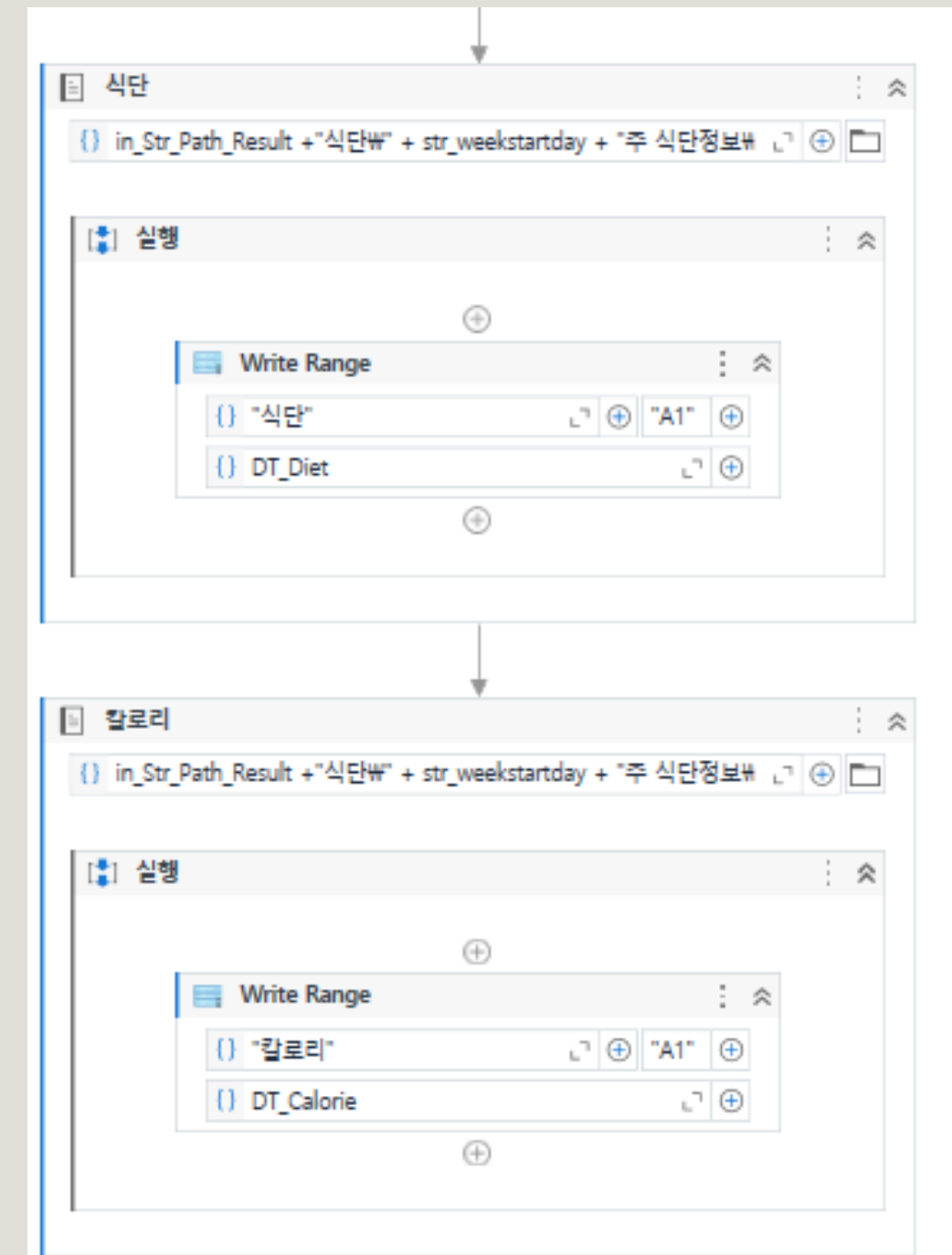
- 챗 GPT에게 1일 식단을 받은 후 스플릿 하여 불러온 DataTable에 입력

03. 수행 절차 및 방법



<원하는 날짜가 포함된 엑셀을 DataTable로 불러옴>

- 식단변경 로직

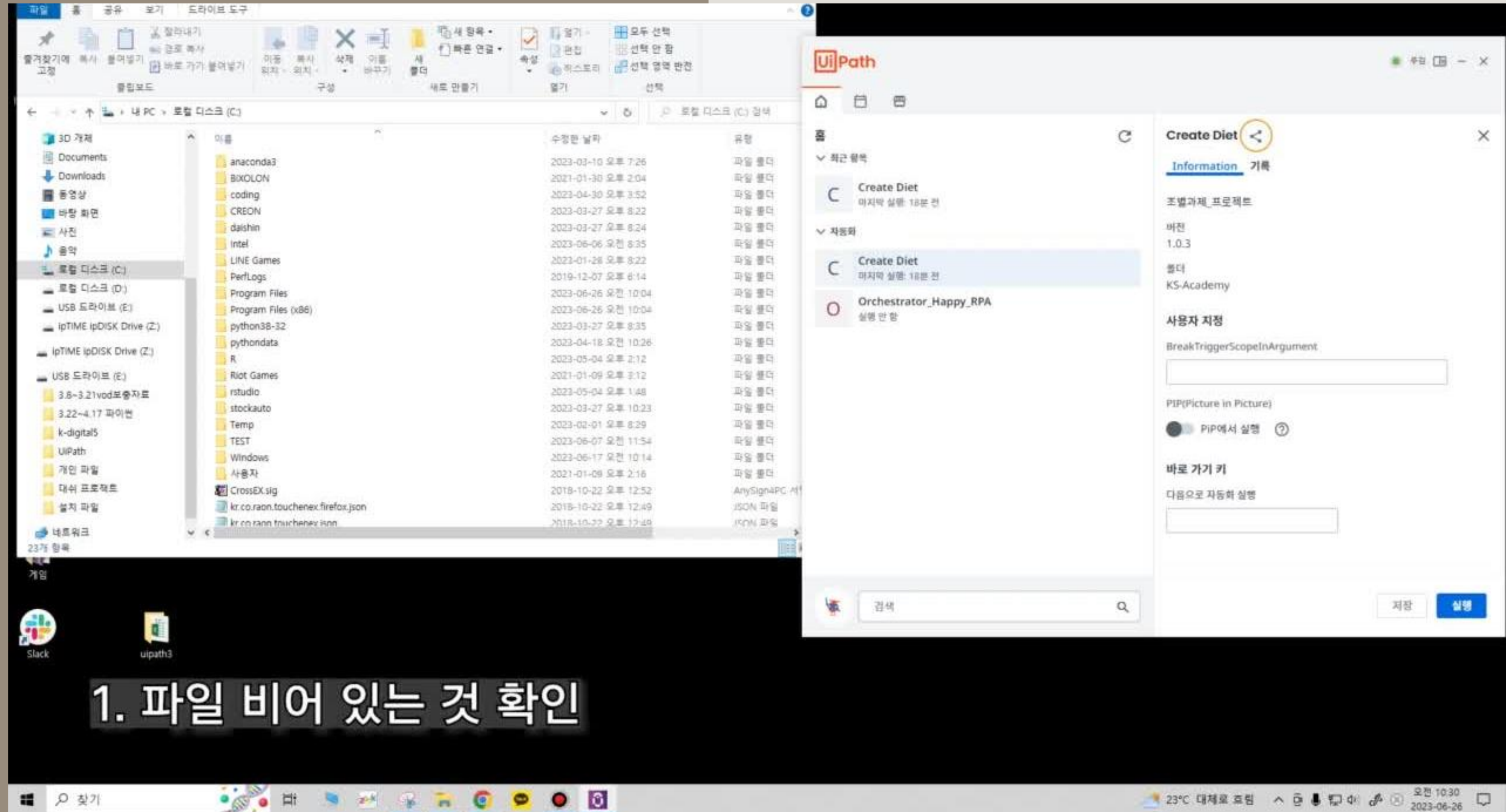


<SourceData와 Result 파일의 엑셀파일에 해당 DataTable 업데이트>

04. 프로젝트 수행 결과

- 동영상

04



04. 프로젝트 수행 결과

- Result

04

내 PC > 로컬 디스크 (C:) > uipath3 > Result			
이름	수정한 날짜	유형	크기
레시피	2023-06-22 오후 4:52	파일 폴더	
식단	2023-06-23 오전 10:51	파일 폴더	
일일식단.xlsx	2023-06-23 오후 12:57	Microsoft Excel ...	10KB

내 PC > 로컬 디스크 (C:) > uipath3 > Result > 식단			
이름	수정한 날짜	유형	
20230626주 식단정보	2023-06-23 오전 10:37	파일 폴더	
20230703주 식단정보	2023-06-23 오후 1:29	파일 폴더	
20230710주 식단정보	2023-06-23 오후 1:30	파일 폴더	
20230717주 식단정보	2023-06-23 오후 1:33	파일 폴더	
20230724주 식단정보	2023-06-23 오후 1:56	파일 폴더	
20230731주 식단정보	2023-06-23 오후 1:57	파일 폴더	
20230807주 식단정보	2023-06-23 오후 1:59	파일 폴더	
20230814주 식단정보	2023-06-23 오후 2:00	파일 폴더	
20230821주 식단정보	2023-06-23 오후 2:01	파일 폴더	
20230828주 식단정보	2023-06-23 오후 2:03	파일 폴더	

<각 주마다 식단정보 폴더가 생성되게 된다>






	A	B	C	D
1	구분	아침	점심	저녁
2	월요일	625	750	750
3	화요일	610	800	750
4	수요일	650	800	750
5	목요일	650	800	750
6	금요일	700	800	750
7	토요일	750	800	750
8	일요일	700	800	750

	A	B	C	D	E
구분	아침	점심	저녁		
월요일	아침: 쌀밥(200g) + 계란	점심: 김치찌개 + 돼지고기	저녁: 불고기 + 청경채무침 + 밥(200g)		
화요일	아침: 현미밥(200g) + 달	점심: 수육 + 된장국 +	저녁: 장조림(오리, 돼지, 닭 중 택) + 찜(상추, 깻잎)		
수요일	아침: 보리밥(200g) + 계	점심: 제육볶음 + 오이냉	저녁: 갈비찜 + 콩나물국 + 깻두기 + 밥(200g)		
목요일	아침: 코다리조림 + 쌀밥	점심: 제육덮밥 + 미소된	저녁: 돈가스 + 찜(상추, 깻잎 등) + 밥(200g)		
금요일	아침: 닭죽 + 김밥(2롤)	점심: 불고기비빔밥 + 맑	저녁: 된장찌개 + 계란탕 + 청경채나물		
토요일	아침: 누룽지탕(돼지고기	점심: 냉면(비빔 또는 나	저녁: 양념치킨 + 오이냉국 + 깻두기		
일요일	아침: 팔죽 + 만두(5개)	점심: 김치찜 + 동태찌개	저녁: 소고기무국 + 찜(상추, 깻잎 등) + 밥(200g)		

<추천 받은 일주일식단과 칼로리 시트>

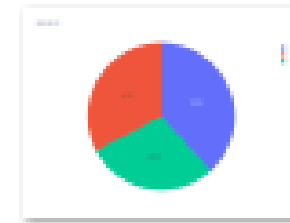
04. 프로젝트 수행 결과

- Result

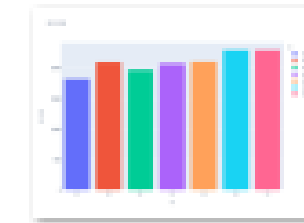
 갈비찜
 된장찌개
 시금치나물
 잡채
 평양냉면
 회냉면

<챗 GPT가 전수해준 요리 레시피들>

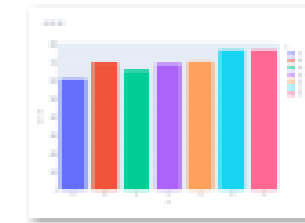
- 레시피는 하나의 Result 폴더로 저장 되어 추후 확인 가능



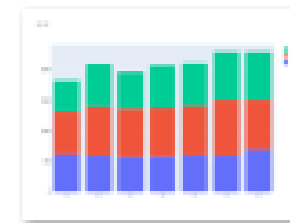
시간대별 평균
칼로리
_20230710~2
0230716



요일별 총 칼
로리
_20230710~2
0230716



요일별 평균
칼로리
_20230710~2
0230716



일별 칼로리
_20230710~2
0230716



일주일식단
_20230710~2
0230716

<추천 받은 일주일식단.xlsx 와 그것을 시각화한 자료>

- 일주일식단.xlsx 와 그 시각화 자료를 각각 기간별로 저장

04. 프로젝트 수행 결과

- 일주일 식단 구성

구분	아침
월요일	쌀밥(200g) + 계란후라이(2개) + 양배추김치(100g) + 두유(200ml)
화요일	오이, 당근 샐러드 + 햄(50g) + 브레드(1조각) + 커피(150ml)
수요일	유부초밥(2개) + 멸치볶음 + 김가루(1큰술) + 우유(200ml)
목요일	찜밥(200g) + 메추리알조림 + 시금치나물 + 콩나물국
금요일	토마토, 오이 샐러드 + 치즈(20g) + 김밥(1롤) + 우유(200ml)
토요일	구운계란 2개 + 롤케익(1조각) + 커피(150ml)
일요일	참치샐러드(참치캔 1캔, 마요네즈 1큰술) + 브레드(1조각) + 커피(150ml)

점심
된장찌개(250g) + 돈육볶음(100g) + 배추김치(100g) + 밥(200g)
불고기(100g) + 청경채나물(100g) + 삼치조림(100g) + 밥(200g)
김치찌개(250g) + 노랑진어묵(50g) + 비빔면(1인분)
순두부찌개(250g) + 김치전(50g) + 김치(100g) + 밥(200g)
잡채덮밥(1인분) + 미역줄기볶음(100g) + 김치(200g)
불고기덮밥(1인분) + 고추장쌈무(200g)
제육볶음(100g) + 깍두기(100g) + 미역국(250ml) + 밥(200g)

저녁
삼겹살구이(200g) + 상추쌈(100g) + 마늘소스(30g) + 된장찌개(200ml) + 밥(200g)
돼지갈비찜(200g) + 깍두기(100g) + 떡국(200ml) + 밥(150g)
소고기무국(200ml) + 임연수구이(200g) + 참나물무침(100g) + 밥(200g)
된장찌개(200ml) + 장어구이(200g) + 콩나물무침(100g) + 밥(200g)
불고기볶음밥(250g) + 수제라면(200ml) + 김치(100g)
쌀국수(250g) + 열무국(200ml) + 사천식잡채(100g)
돈까스(200g) + 고구마샐러드(100g) + 깍두기(100g) + 밥(200g)

구분	아침	점심	저녁
월요일	625	785	920
화요일	600	800	900
수요일	400	690	830
목요일	595	840	880
금요일	430	810	980
토요일	620	705	760
일요일	735	690	890

일주일식단.xlsx 에 요일별 '아침/점심/저녁 식단'과 '칼로리 정보' 생성

04. 프로젝트 수행 결과 - 일일 식단

구분	식단
아침	아침:オート밀(100g) + 요거트(150g) + 샐러드(50g) + 블루베리(50g)
점심	점심: 김치볶음밥(250g) + 돼지갈비찜(150g) + 계란말이(2조각)
저녁	저녁: 간장게장(150g) + 부추호박전(2조각) + 콩나물국(150g) + 쌀밥(200g)

2023-07-06 2023-06-28 Sheet1 (+)

구분	식단						
아침	아침: 유부초밥(5조각) + 두유(200ml) + 삶은김(50g)						
점심	점심: 감자짜글이(200ml) + 오므라이스 + 맛살새우볶음(100g)						
저녁	저녁: 닭볶음탕(100g) + 동태찌개(200ml) + 무생채(100g) + 백미밥(200g)						

2023-07-06 2023-06-28 Sheet1 (+)

input 엑셀 파일에 입력된 날짜에 맞는 일일 식단 생성

04. 프로젝트 수행 결과 - 레시피



제목: 된장찌개 레시피

<재료>

- | | |
|---------------|--------------|
| - 돼지고기 (200g) | - 물엿 (1 큰술) |
| - 된장 (2 큰술) | - 국간장 (1 큰술) |
| - 물 (4 컵) | - 후추 (적당량) |
| - 무 (1/2 개) | |
| - 양파 (1/2 개) | |
| - 청양고추 (1 개) | |
| - 고춧가루 (1 큰술) | |
| - 마늘 (2 쪽) | |
| - 참기름 (1 큰술) | |

<조리법>

1. 돼지고기는 국자에 담아 끓는 물에 살짝 데친 후 찬물에 헹궈 불순물을 제거한다.
2. 무와 양파는 굵게 채 썰고, 청양고추는 어슷하게 썰어 준비한다. 마늘도 다져 놓는다.
3. 냄비에 참기름을 넣고 마늘을 볶다가 돼지고기를 넣어 볶는다.
4. 물 4 컵을 넣고 끓인 후 된장, 고춧가루, 물엿, 국간장을 순서대로 넣어준다.
5. 무와 양파를 넣고 끓인다.
6. 끓이다가 청양고추를 넣고 후추로 간을 맞춘다.
7. 불을 끄고 된장찌개가 끓어난 물기를 제거한 후 바로 먹는다.

다운 받은 이미지와 챗 GPT가 제공한 레시피로 Word 생성

05. 자체 평가 의견

이름	역할	자체 평가 의견
김 주 성	팀 장	Work Flow을 생각하고 작성하는 과정이 재밌었다. 중간중간 어려움이 있었지만, 해결했을 때 쾌감이 좋았다.
이 혁 재	팀 원	하나의 로직을 짤 때마다 변수가 너무 많이 발생해서 최대한 안정성을 강화하기 위해 노력했 고, 그만큼 안정적이게 되어서 상당히 뿌듯했다. 처음 RPA를 배울때 trycatch 쓰기도 싫어했었던 나를 생각해보면 왜 써야하는 지를 이해하게 되어서 기쁘다.
차 아 란	팀 원	수업, 과제 중 혼자 로직을 짜다가 여러 사람의 로직을 보고 합치는 과정에서 같은과제에 접 근하는 방식이 사람 마다 다르다는게 인상적이었고 이를 통해 많은걸 배웠다
남 진 식	팀 원	RPA로 어떤 편리함을 얻을 수 있을지 팀원들과 주제를 자유롭게 생각하는 과정이 흥미로 웠고, 생각한 알고리즘과 실제로 구현 과정에서 차이가 있어 어려움이 있었지만 해결해 나 가는 과정에서 많은 것을 배웠다.

QNA



감사합니다