문서인식 서비스 개발

유아이패스코리아
산학협력프로젝트 8팀

목차

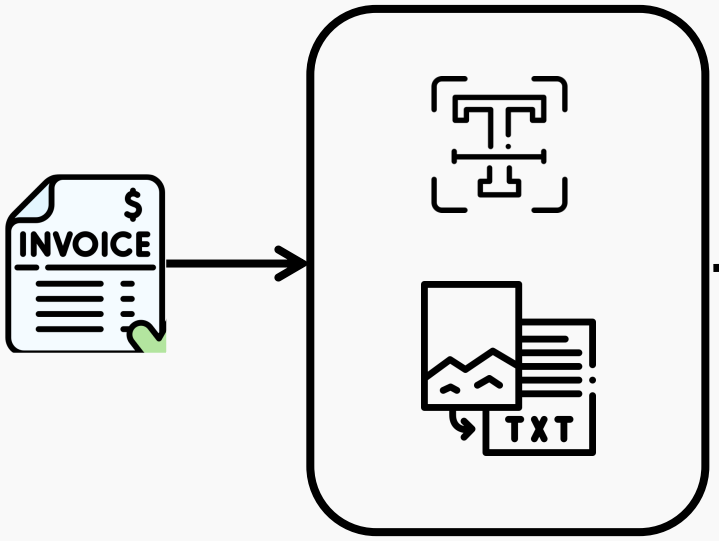
Contents

Project Overview

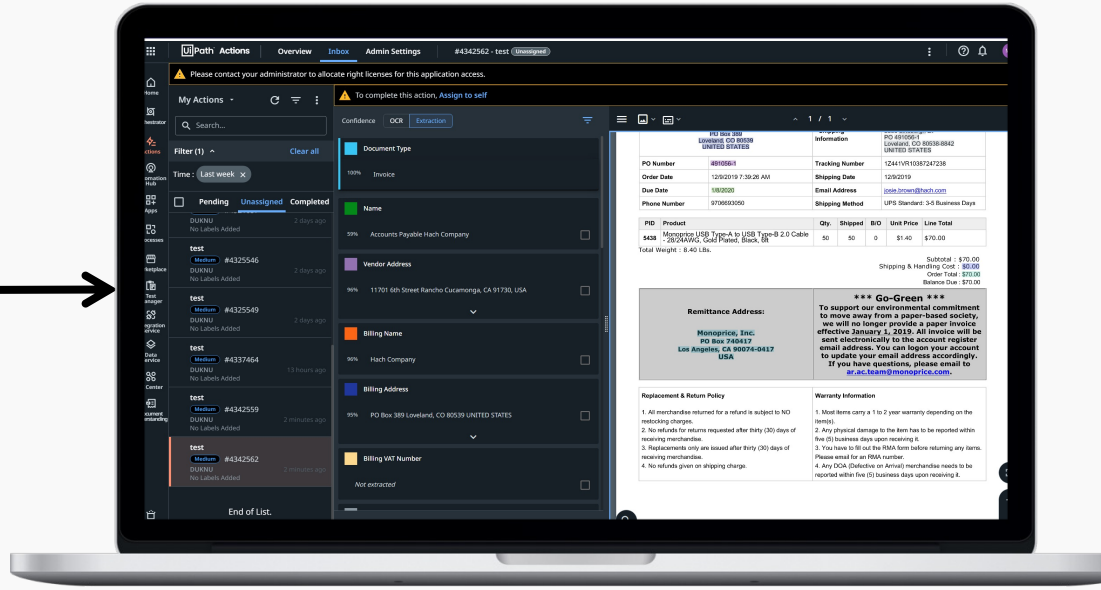
Invoice 유형의 문서 인식

Uipath RPA + Azure Api, Naver Clova Api

강력한 인지 서비스 활용



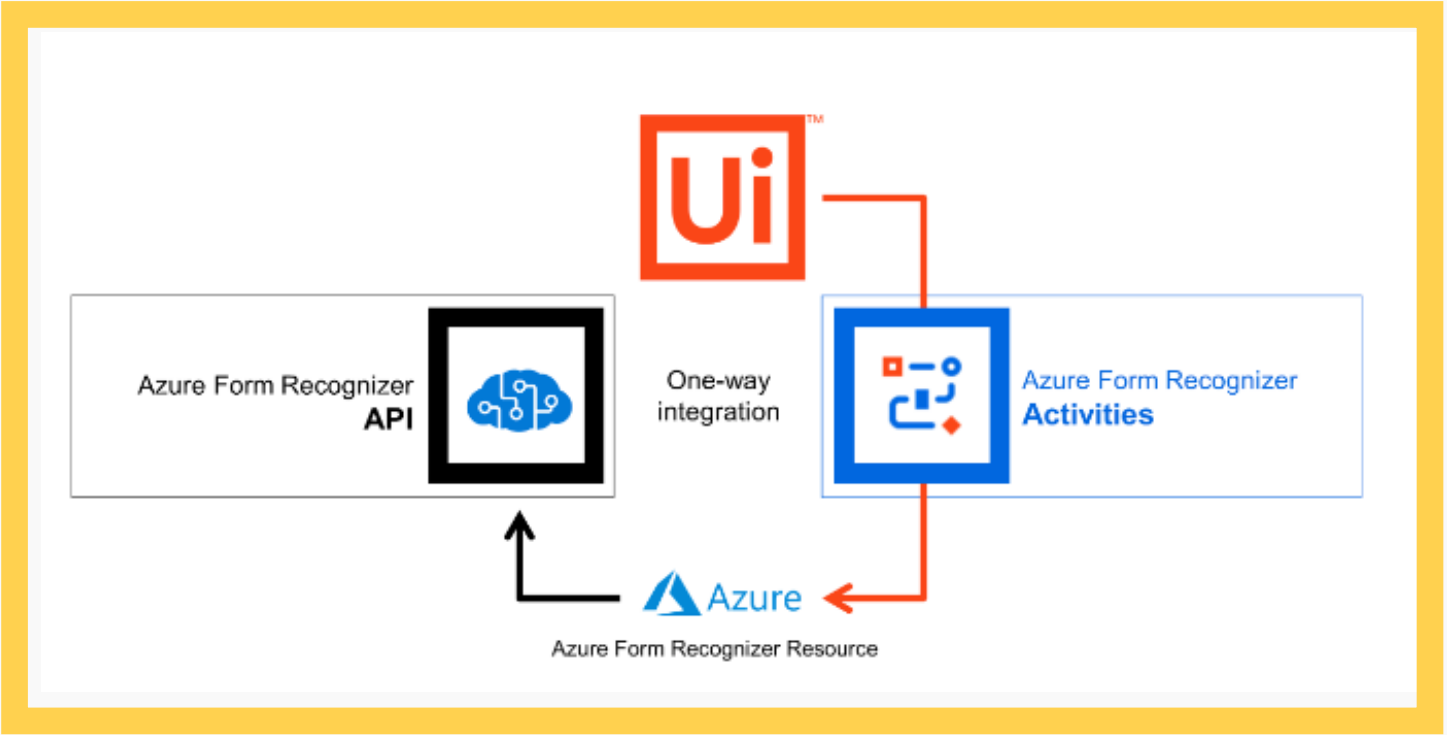
Microsoft Azure, Naver CLOVA



Project Overview (Contd.)

Azure Prebuilt Model 활용

> Data 입력을 자동화, Invoice 처리 워크플로우를 능률화하는 효과



Requirements

3) 소프트웨어 인터페이스

이름	RestAPI를 이용한 문서 추출기
목적 / 내용	RestAPI를 통해 Azure Form Recognizer 서비스로 문서를 전송하고 결과값을 생성
입력 주체 / 출력 목적지	Uipath Studio / Uipath Studio
범위 / 정확도 / 허용오차	인식하고자 하는 문서의 양식에 따른 입력 범위 / Azure Form Recognizer의 정확도를 따름 / Azure Form Recognizer의 허용 오차를 따름
단위	문서 입력
시간 / 속도	Azure Form Recognizer의 실행 속도
타 입출력과의 관계	입력 내용에 따라 클라이언트에서 처리
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식	c# 프로그램을 빌드하여 생성되는 DLL파일
명령 형식	빌드
종료 메시지	실행 후 자동 종료

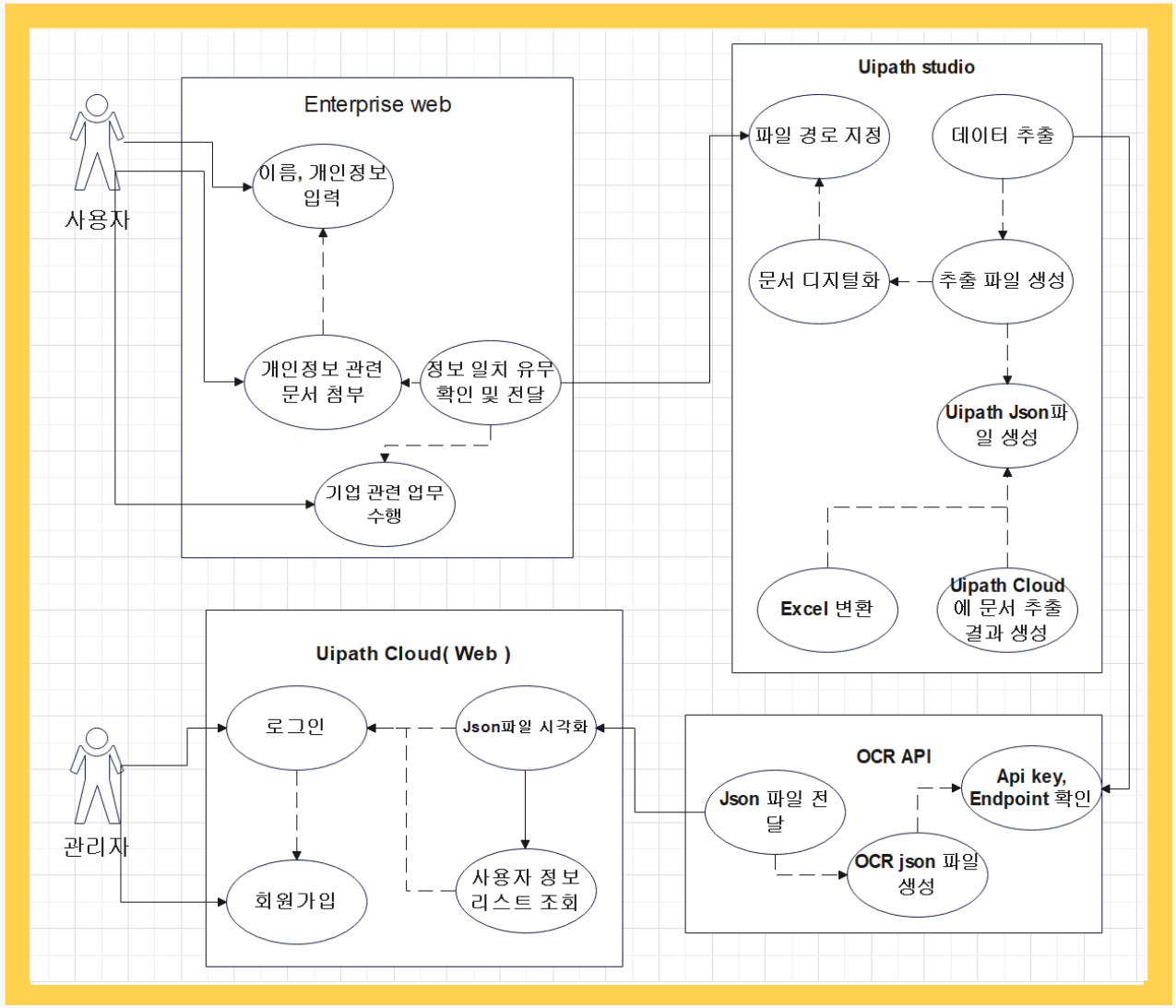
가. 외적 인터페이스 요구사항 (External Interface Requirements)

1) 사용자 인터페이스 (User Interfaces)

이름	Uipath Studio를 통한 입력 처리
목적 / 내용	Uipath사의 클라이언트가 Uipath Studio를 통하여 인식하고자 하는 문서 전달
입력 주체 / 출력 목적지	사용자 / Uipath Cloud
범위 / 정확도 / 허용오차	인식하고자 하는 문서의 양식에 따른 입력 범위 / 문서의 품질에 따른 입력 정확도 / Uipath studio의 파일 인식률에 따른 허용 오차
단위	문서 입력
시간 / 속도	즉각적인 사용자 명령 수행
타 입출력과의 관계	입력 내용에 따라 클라이언트에서 처리하거나 서버로 명령 요청
화면 형식 및 구성	IDE 형식의 구성
윈도우 형식 및 구성	그래픽 유저 인터페이스
데이터 형식	인쇄 및 손글씨 양식, PDF, 이미지 파일
명령 형식	파일 실행 명령
종료 메시지	실행 후 자동 종료

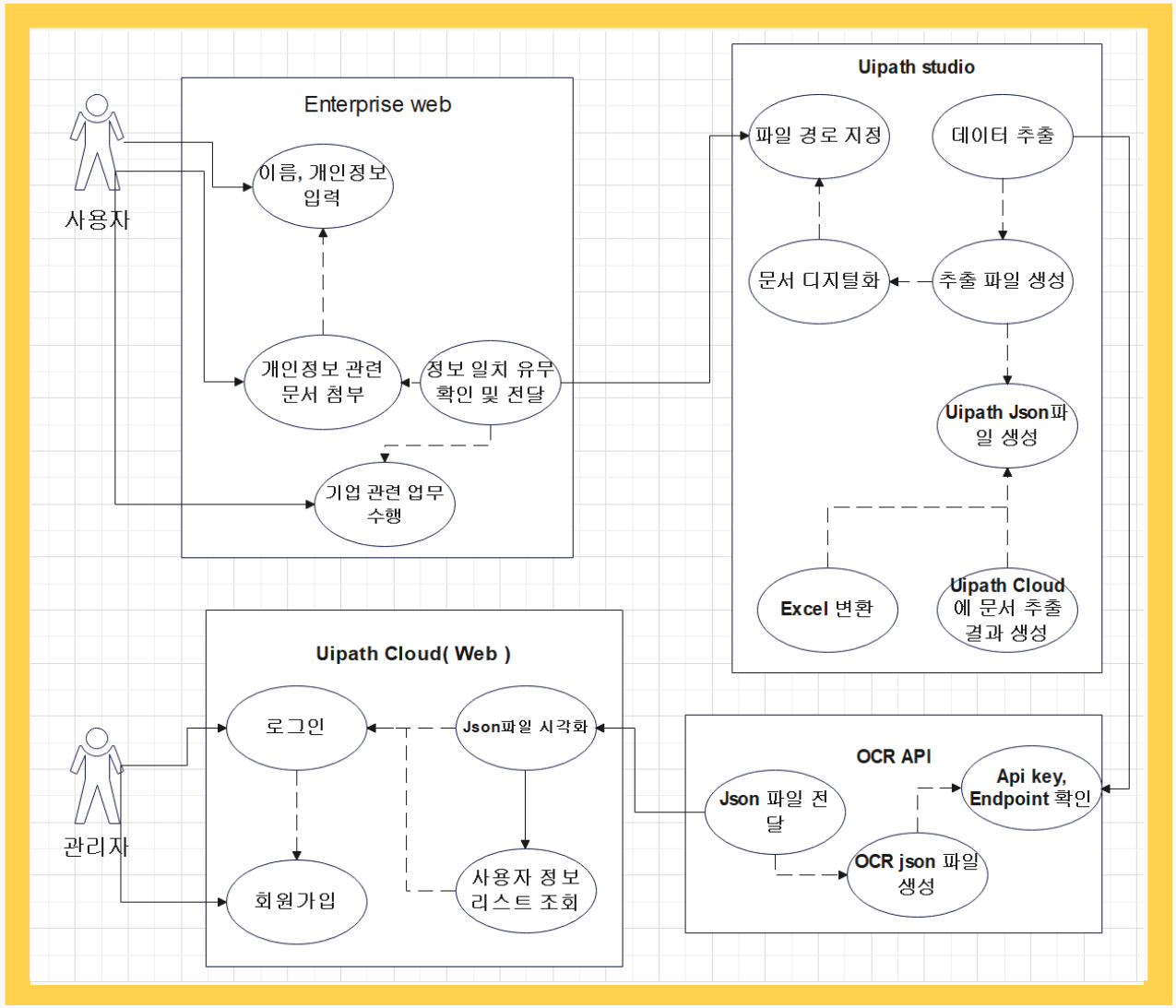
1. 시스템이 invoice 문서를 정확하게 인식
2. 인식 프로세스는 Azure API (Naver Clova API)를 사용한다.
3. Invoice 문서에서 관련된 정보를 안정적으로 추출할 수 있도록 인식 프로세스의 정확성에 대한 필요성을 강조

Use-Case Diagram



1. 시스템 경계
   Enterprise web
   Uiptath Studio 환경
   Azure API 서비스
   Invoice 문서 소스
2. Actor
   사용자
   관리자

Use-Case Diagram (Contd.)



3. Relationship

a) 사용자 – 문서 업로드

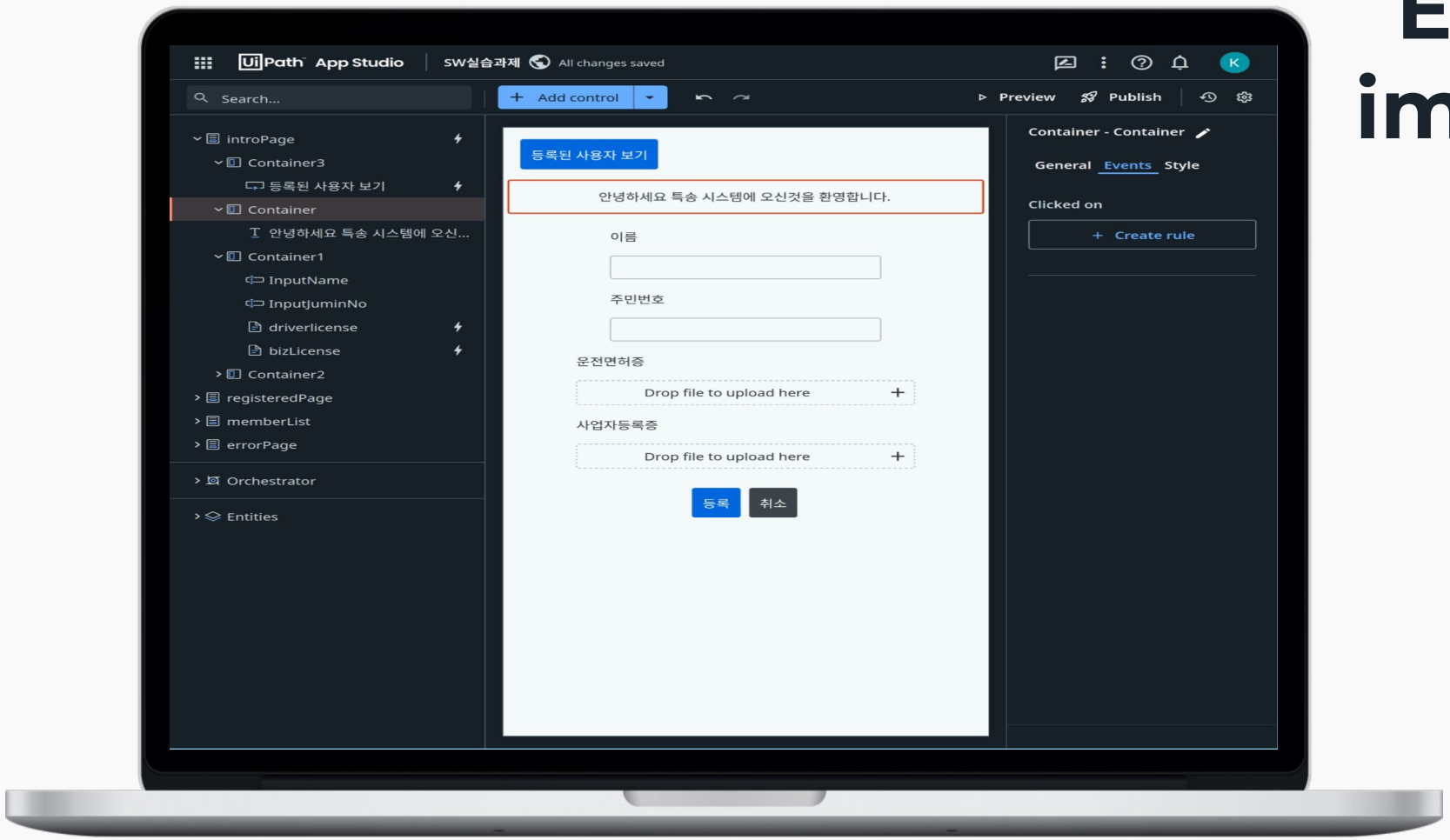
b) 문서 업로드 – API 전송

c) API 전송 – 문서 데이터 받기

d) 문서 데이터 수신 – 데이터베이스 업데이트

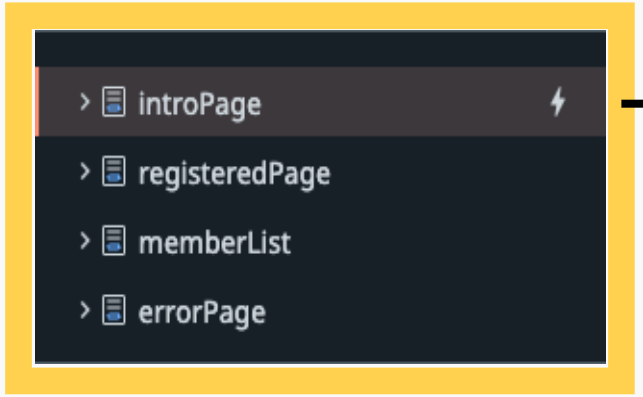
e) 데이터베이스 업데이트 – 사용자 알림

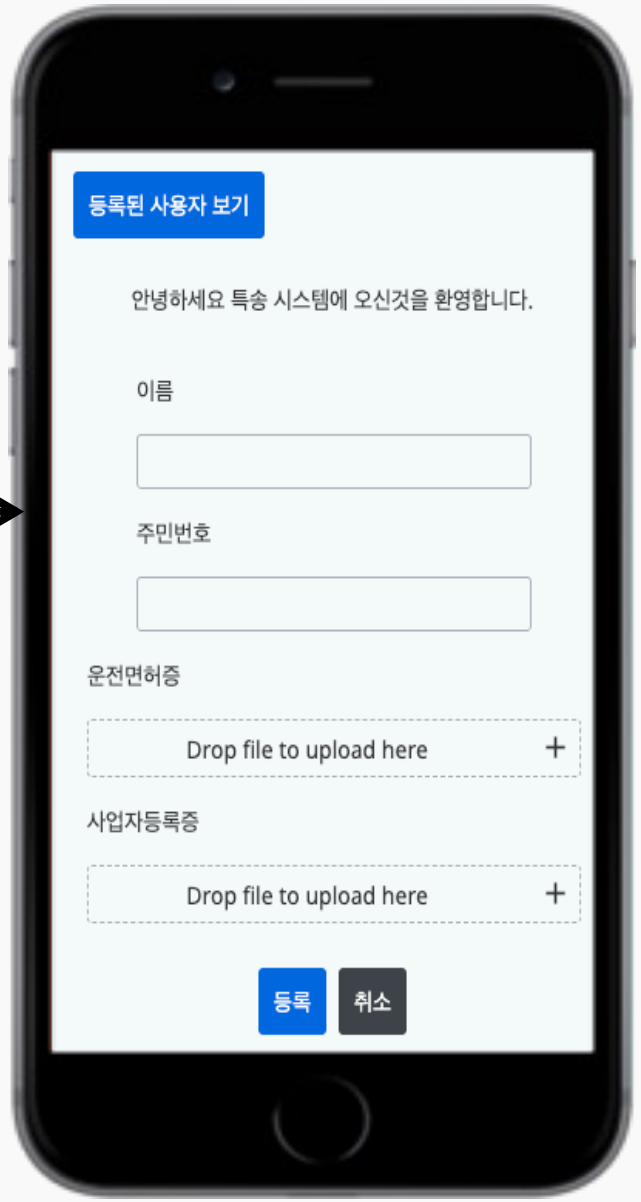
Enterprise web implementation



- Uipath Apps 라는 low-code 애플리케이션 개발 플랫폼 사용
- 지금까지 개발한 Azure Activity Code와 연동하여 Use-case scenario 시연을 위한 프로그램 작성 중

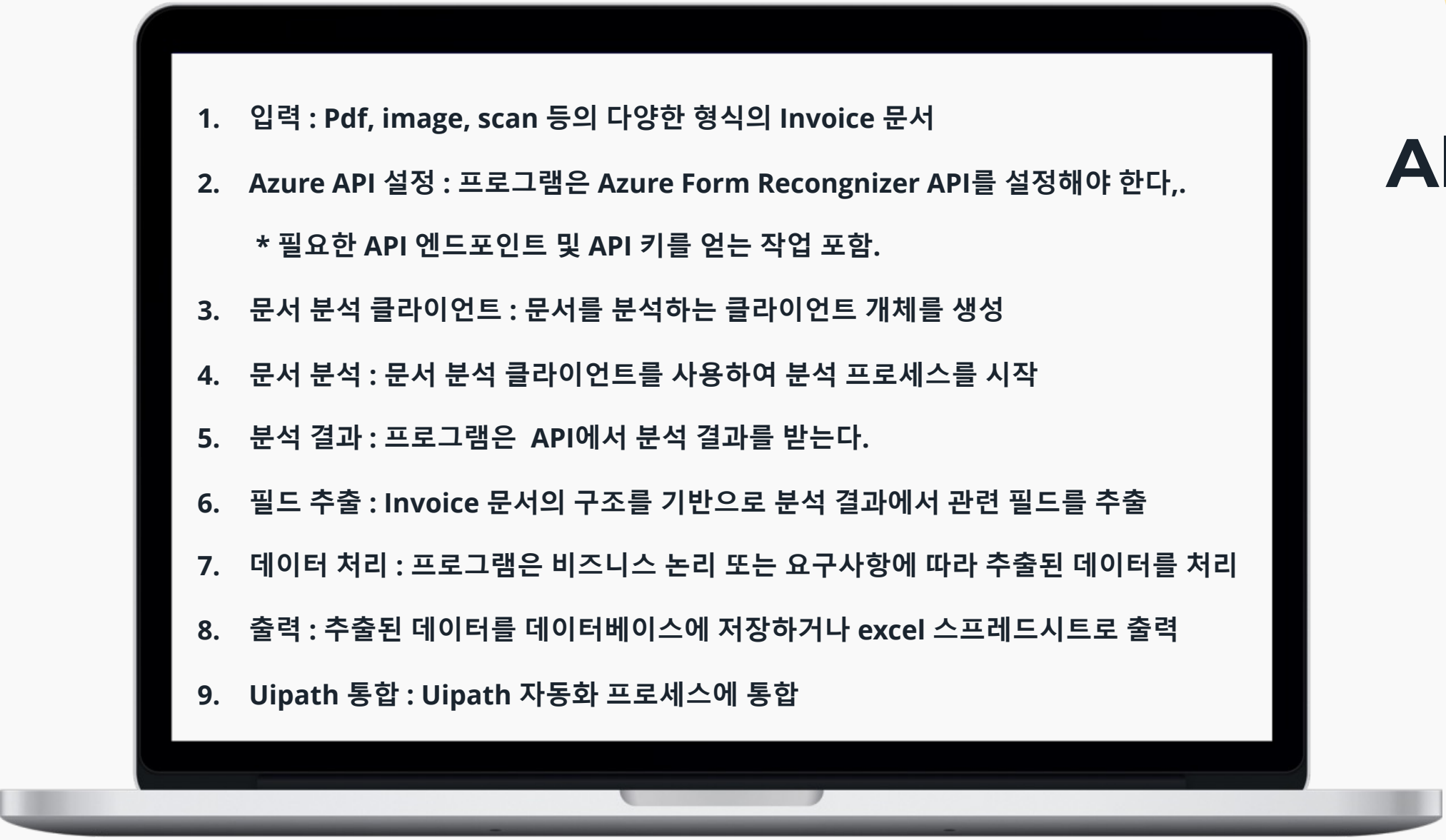
Enterprise web implementation (Contd.)



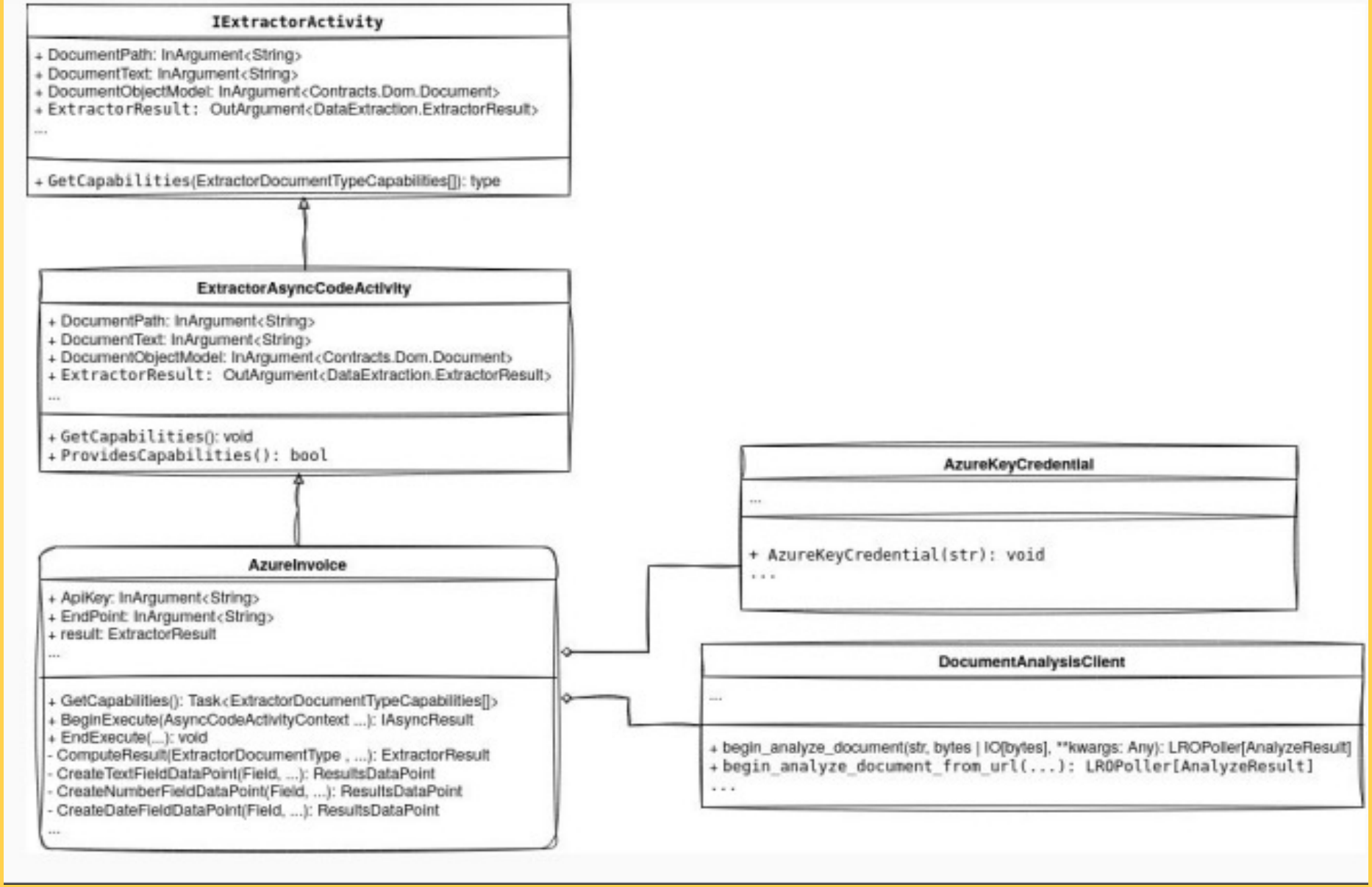


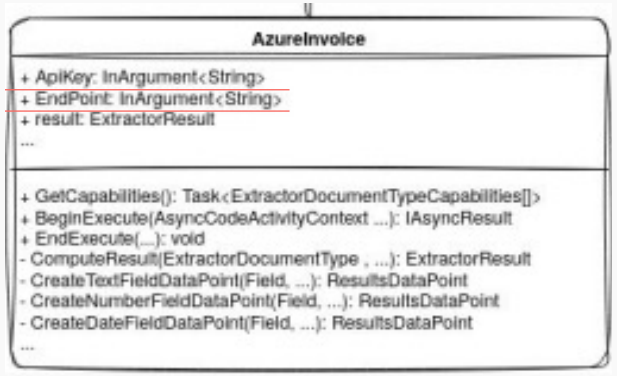
1. 사용자 정보 입력 & 문서 업로드
2. Azure Api로 전송
3. 시스템 데이터 수신
4. DB 업데이트
5. 사용자에게 알림 (정상 등록 / 오류)

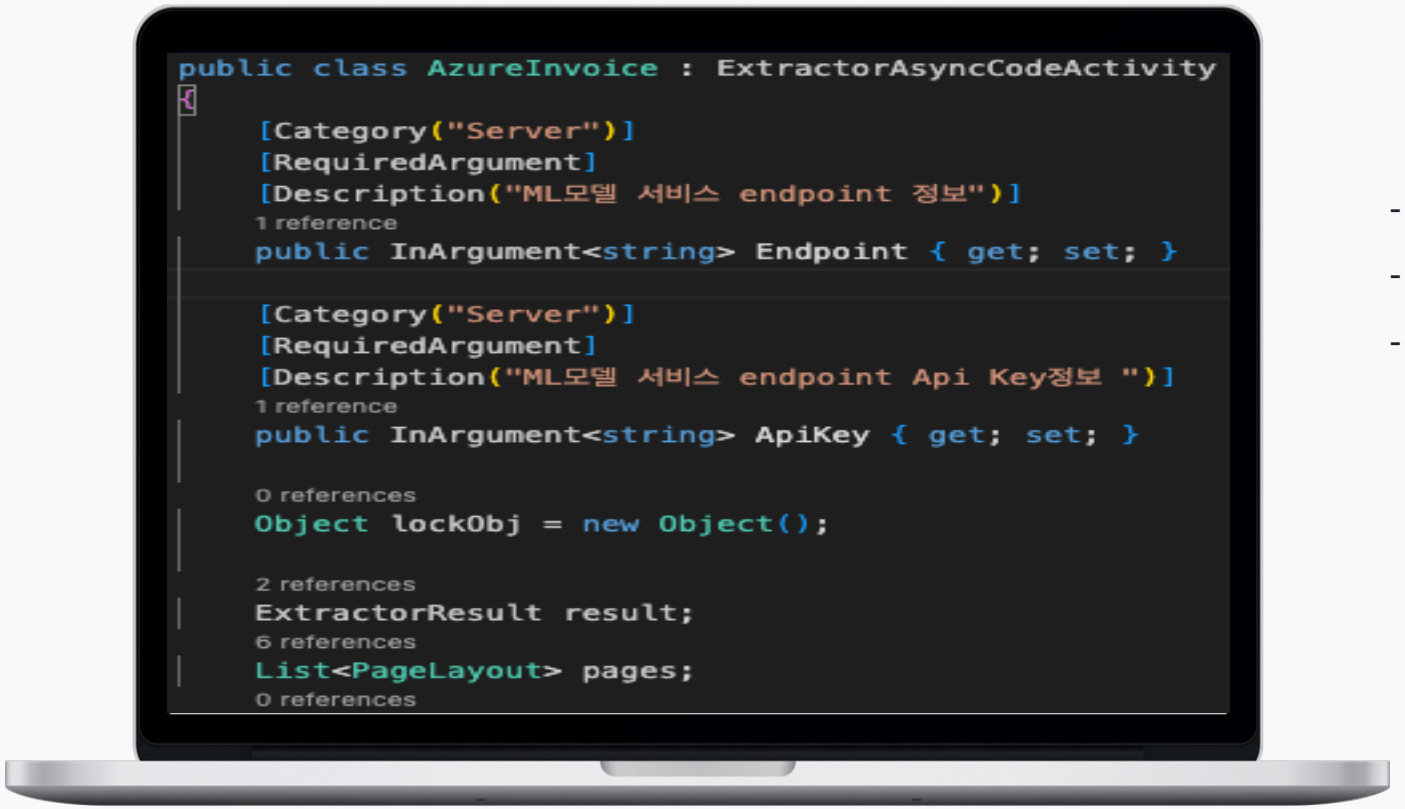
Algorithm


1. 입력 : Pdf, image, scan 등의 다양한 형식의 Invoice 문서
2. Azure API 설정 : 프로그램은 Azure Form Recongnizer API를 설정해야 한다,.
* 필요한 API 엔드포인트 및 API 키를 얻는 작업 포함.
3. 문서 분석 클라이언트 : 문서를 분석하는 클라이언트 개체를 생성
4. 문서 분석 : 문서 분석 클라이언트를 사용하여 분석 프로세스를 시작
5. 분석 결과 : 프로그램은 API에서 분석 결과를 받는다.
6. 필드 추출 : Invoice 문서의 구조를 기반으로 분석 결과에서 관련 필드를 추출
7. 데이터 처리 : 프로그램은 비즈니스 논리 또는 요구사항에 따라 추출된 데이터를 처리
8. 출력 : 추출된 데이터를 데이터베이스에 저장하거나 excel 스프레드시트로 출력
9. Uipath 통합 : Uipath 자동화 프로세스에 통합

Class Diagram

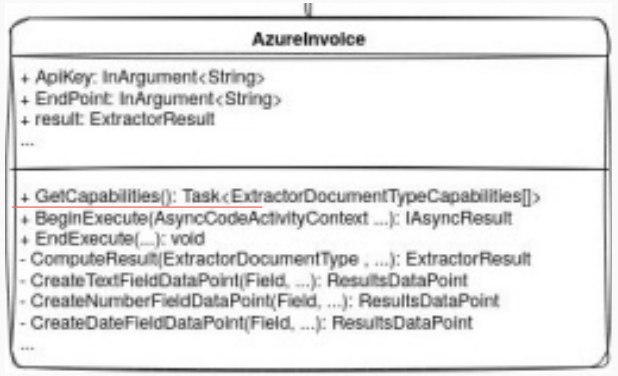




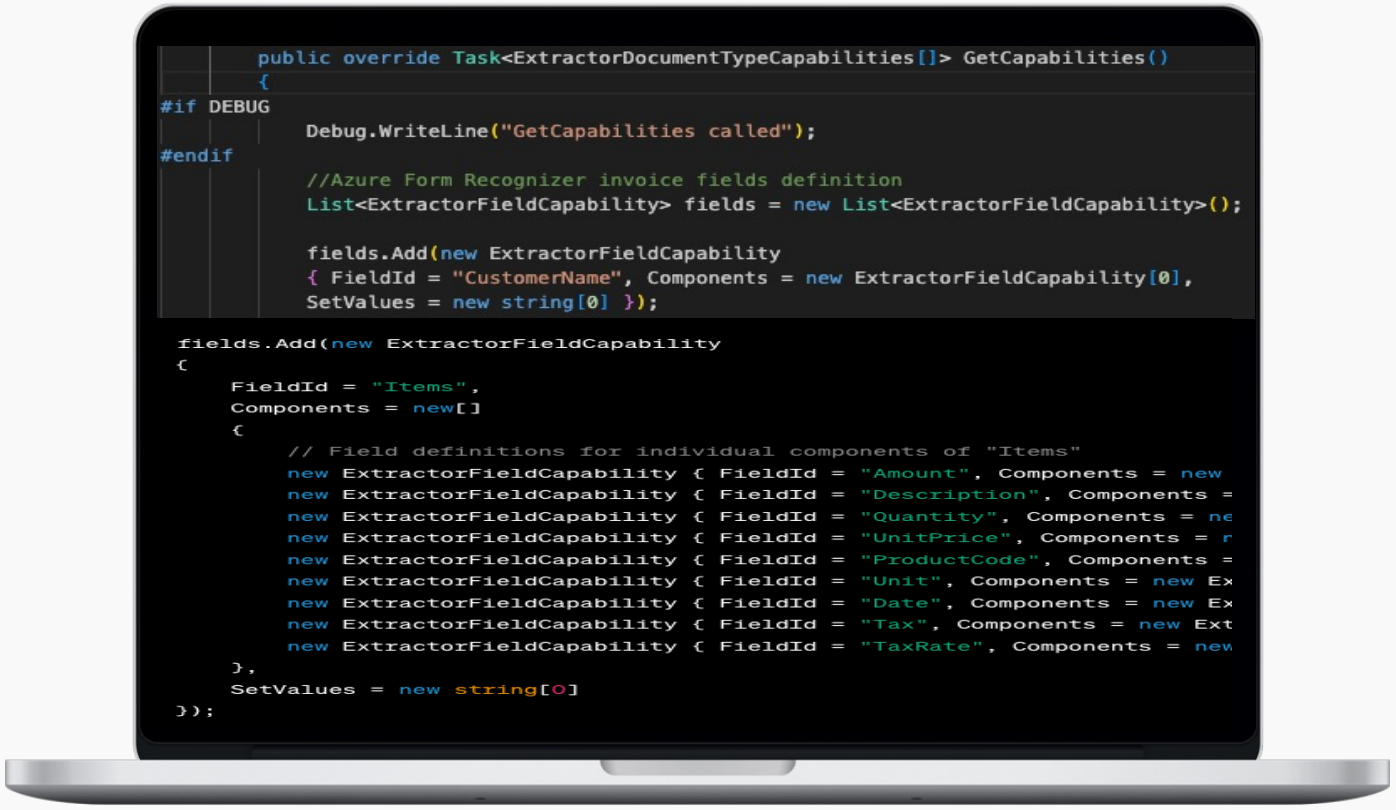


Code Description

- AzureInvoice 클래스
- “Endpoint”와 ”APIKey” 속성 정의
- ML모델 서비스와의 상호작용과 관련된 필수 정보 및 기능을 캡슐화하여 Invoice 추출 기능을 애플리케이션에 통합



Code Description (Contd.)

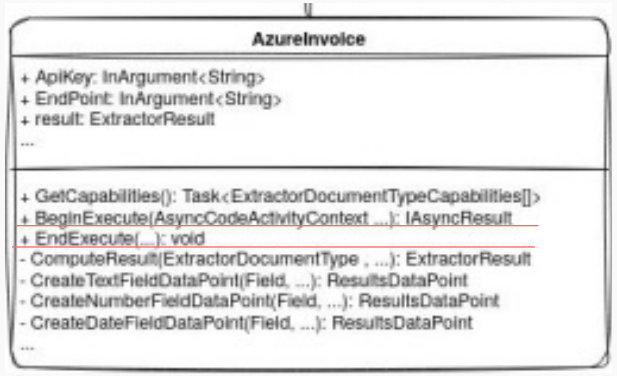


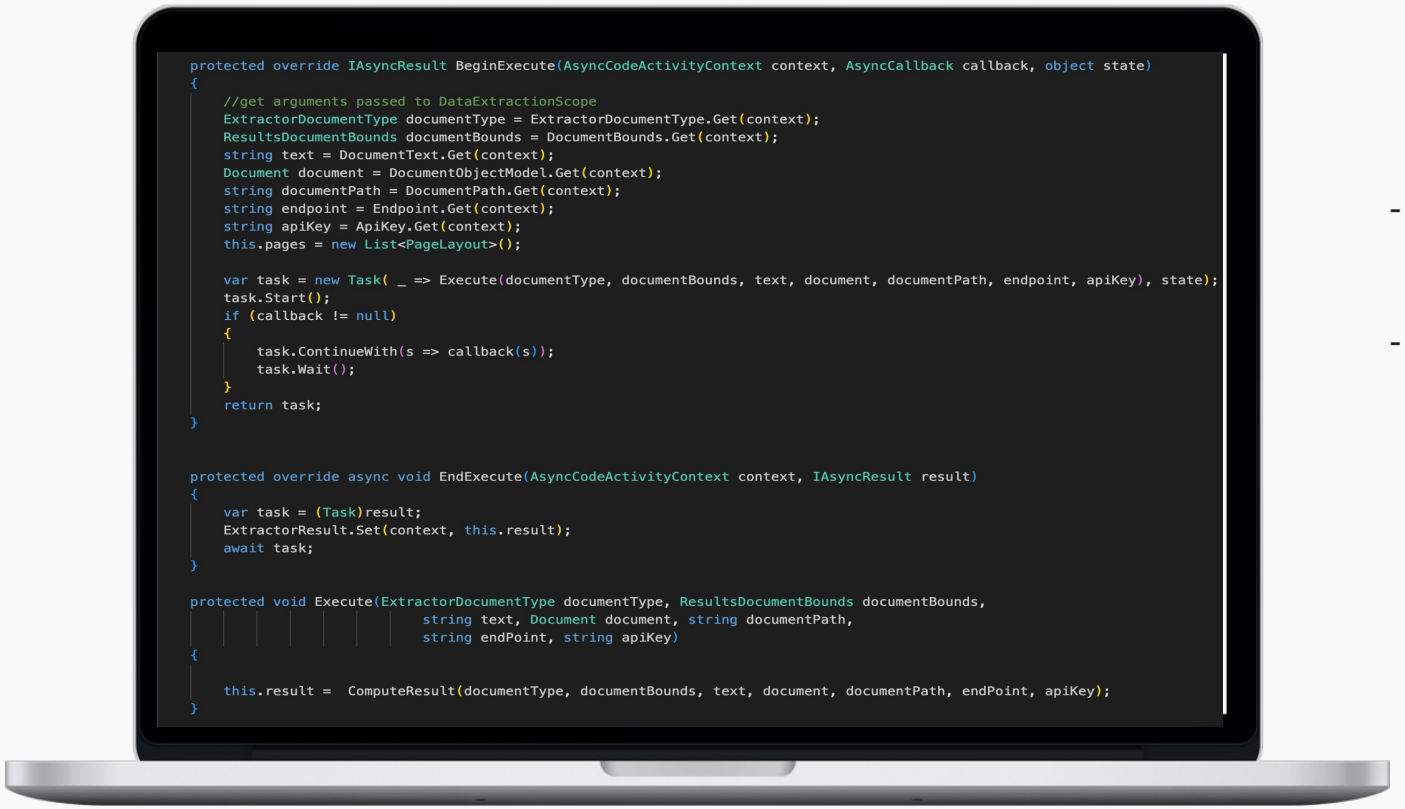
```
        public override Task<ExtractorDocumentTypeCapabilities[]> GetCapabilities()
        {
#if DEBUG
            Debug.WriteLine("GetCapabilities called");
#endif
            //Azure Form Recognizer invoice fields definition
            List<ExtractorFieldCapability> fields = new List<ExtractorFieldCapability>();

            fields.Add(new ExtractorFieldCapability
            { FieldId = "CustomerName", Components = new ExtractorFieldCapability[0],
            SetValues = new string[0] });
```

```
fields.Add(new ExtractorFieldCapability
{
    FieldId = "Items",
    Components = new[]
    {
        // Field definitions for individual components of "Items"
        new ExtractorFieldCapability { FieldId = "Amount", Components = new
        new ExtractorFieldCapability { FieldId = "Description", Components =
        new ExtractorFieldCapability { FieldId = "Quantity", Components = ne
        new ExtractorFieldCapability { FieldId = "UnitPrice", Components = r
        new ExtractorFieldCapability { FieldId = "ProductCode", Components =
        new ExtractorFieldCapability { FieldId = "Unit", Components = new Ex
        new ExtractorFieldCapability { FieldId = "Date", Components = new Ex
        new ExtractorFieldCapability { FieldId = "Tax", Components = new Ext
        new ExtractorFieldCapability { FieldId = "TaxRate", Components = new
    },
    SetValues = new string[0]
});
```

- GetCapabilites() metiod를 상속
- Method 내에서 Azure의 Invoice Field를 나타내는 ExtractorFieldCapability 개체 목록에 각각 필드 정의가 추가
- Item field의 각 구성요소는 amount, Description, Quantity 등...

Code Description (Contd.)

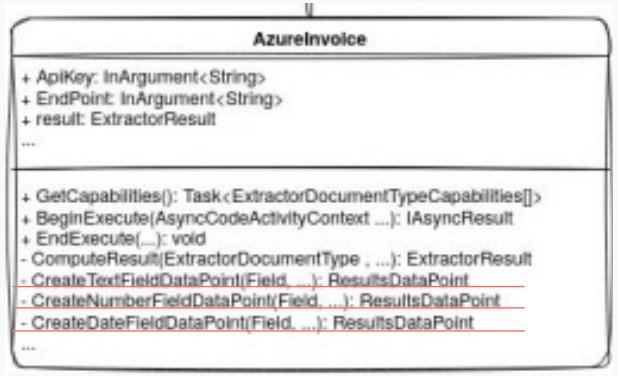


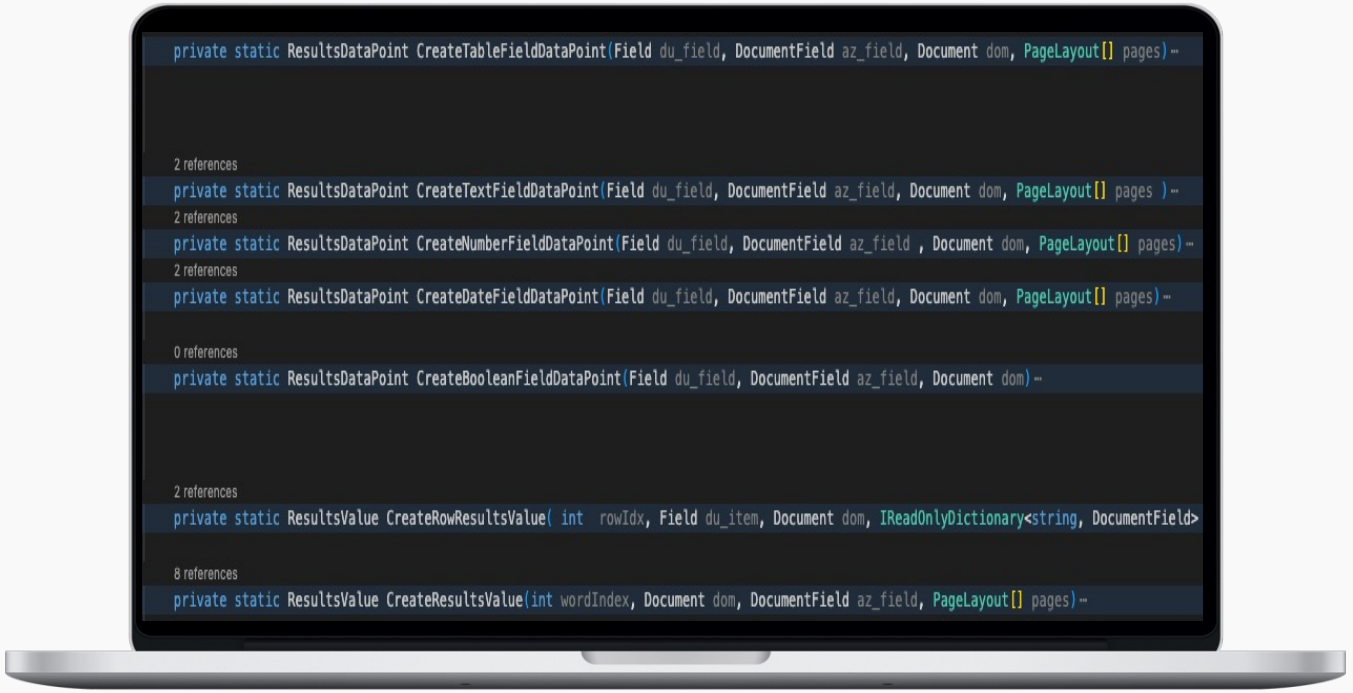


```
protected override IAsyncResult BeginExecute(AsyncCodeActivityContext context, AsyncCallback callback, object state)
{
    //get arguments passed to DataExtractionScope
    ExtractorDocumentType documentType = ExtractorDocumentType.Get(context);
    ResultsDocumentBounds documentBounds = DocumentBounds.Get(context);
    string text = DocumentText.Get(context);
    Document document = DocumentObjectModel.Get(context);
    string documentPath = DocumentPath.Get(context);
    string endpoint = Endpoint.Get(context);
    string apiKey = ApiKey.Get(context);
    this.pages = new List<PageLayout>();

    var task = new Task( _ => Execute(documentType, documentBounds, text, document, documentPath, endpoint, apiKey), state);
    task.Start();
    if (callback != null)
    {
        task.ContinueWith(s => callback(s));
        task.Wait();
    }
    return task;
}


protected override async void EndExecute(AsyncCodeActivityContext context, IAsyncResult result)
{
    var task = (Task)result;
    ExtractorResult.Set(context, this.result);
    await task;
}

protected void Execute(ExtractorDocumentType documentType, ResultsDocumentBounds documentBounds,
                       string text, Document document, string documentPath,
                       string endPoint, string apiKey)
{

    this.result =  ComputeResult(documentType, documentBounds, text, document, documentPath, endPoint, apiKey);
}
```

- **Execute 메소드는 문서 처리의 다양한 측면을 나타내는 여러 매개변수로 정의**
- **ComputeResult 메서드를 호출하여 필요한 매개변수를 전달하고 반환된 결과를 “result” Field에 할당함.**

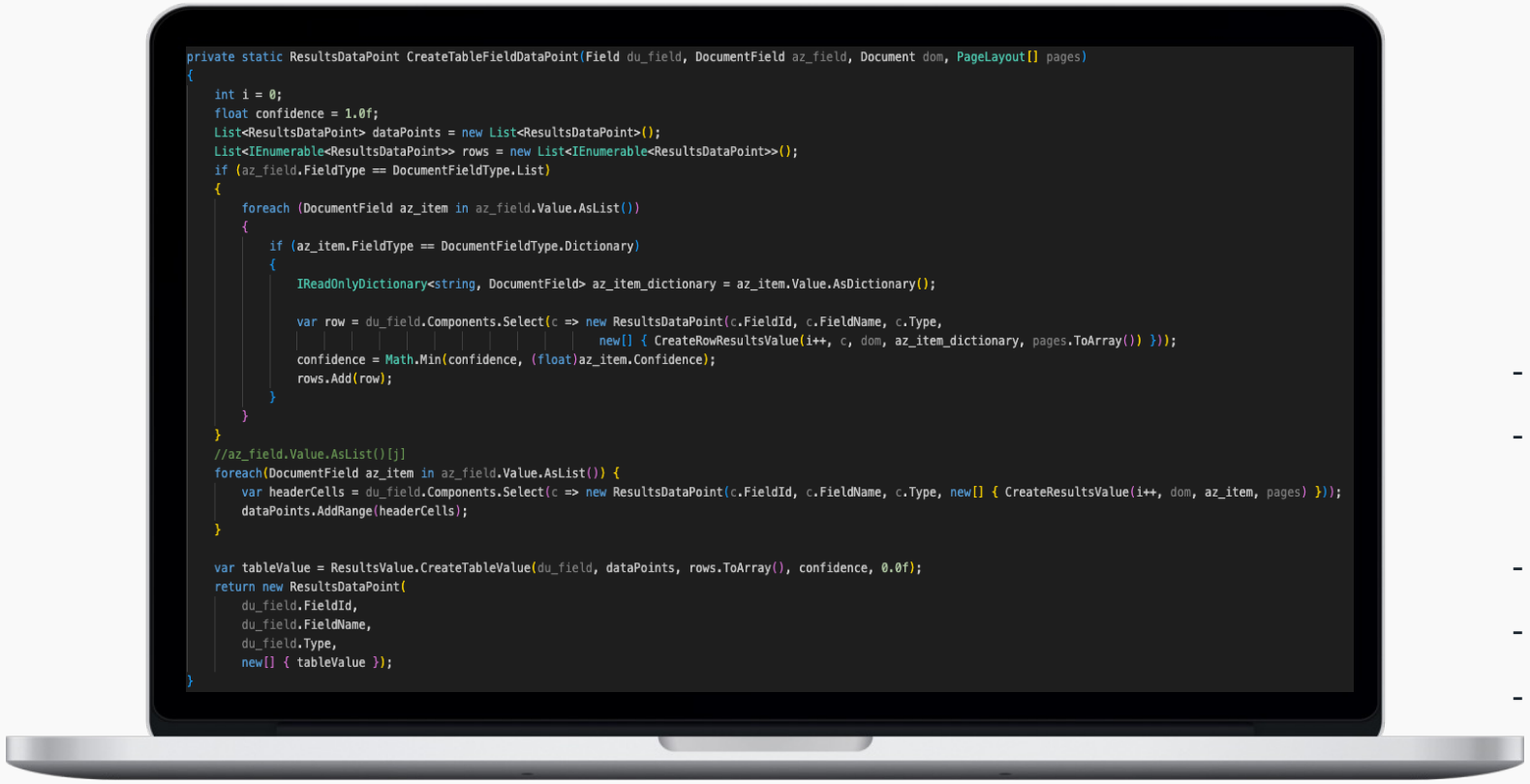
Code Description (Contd.)





- 제공된 매개변수를 기반으로 데이터 포인트를 생성
- FieldType에 따라 각각 다른 메서드 사용
- CreateTextFieldDataPoint
- CreateNumberFieldDataPoint
- CreateTableFieldDataPoint

Code Description (Contd.)



```
private static ResultsDataPoint CreateTableFieldDataPoint(Field du_field, DocumentField az_field, Document dom, PageLayout[] pages)
{
    int i = 0;
    float confidence = 1.0f;
    List<ResultsDataPoint> dataPoints = new List<ResultsDataPoint>();
    List<IEnumerable<ResultsDataPoint>> rows = new List<IEnumerable<ResultsDataPoint>>();
    if (az_field.FieldType == DocumentFieldType.List)
    {
        foreach (DocumentField az_item in az_field.Value.AsList())
        {
            if (az_item.FieldType == DocumentFieldType.Dictionary)
            {
                IReadOnlyDictionary<string, DocumentField> az_item_dictionary = az_item.Value.AsDictionary();

                var row = du_field.Components.Select(c => new ResultsDataPoint(c.FieldId, c.FieldName, c.Type,
                                                        new[] { CreateRowResultsValue(i++, c, dom, az_item_dictionary, pages.ToArray()) }));
                confidence = Math.Min(confidence, (float)az_item.Confidence);
                rows.Add(row);
            }
        }
    }
    //az_field.Value.AsList()[j]
    foreach(DocumentField az_item in az_field.Value.AsList()) {
        var headerCells = du_field.Components.Select(c => new ResultsDataPoint(c.FieldId, c.FieldName, c.Type, new[] { CreateResultsValue(i++, dom, az_item, pages) }));
        dataPoints.AddRange(headerCells);
    }

    var tableValue = ResultsValue.CreateTableValue(du_field, dataPoints, rows.ToArray(), confidence, 0.0f);
    return new ResultsDataPoint(
        du_field.FieldId,
        du_field.FieldName,
        du_field.Type,
        new[] { tableValue });
}
```

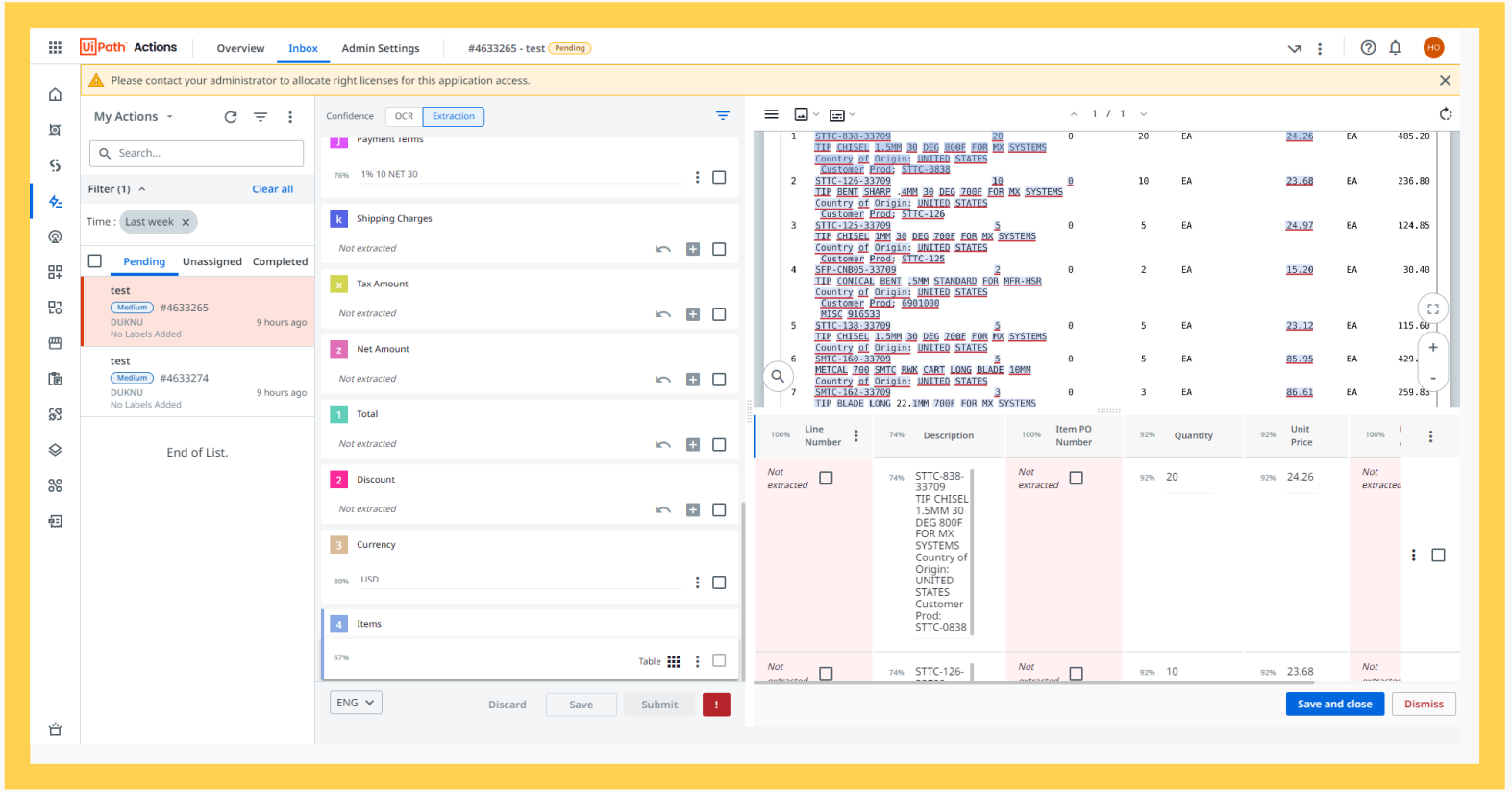
- CreateTableFieldDataPoint 메서드
- FieldType.Table 유형의 필드를 처리

- CreateRowResultValue 메서드
- az_item의 경계 폴리곤을 기준으로 사각형 생성
- 정의된 사각형에 속하는 dom 문서에서 단어를 선택하여 채움
- ResultsValueToken 개체 목록을 만듦

- CreateResultsValue 메서드
- 결과 ResultsValue 반환

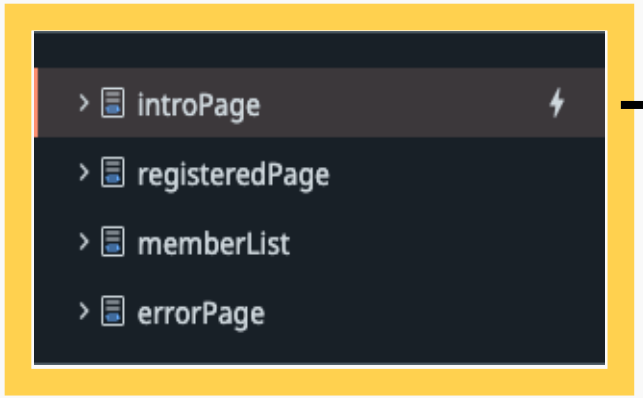
Progress

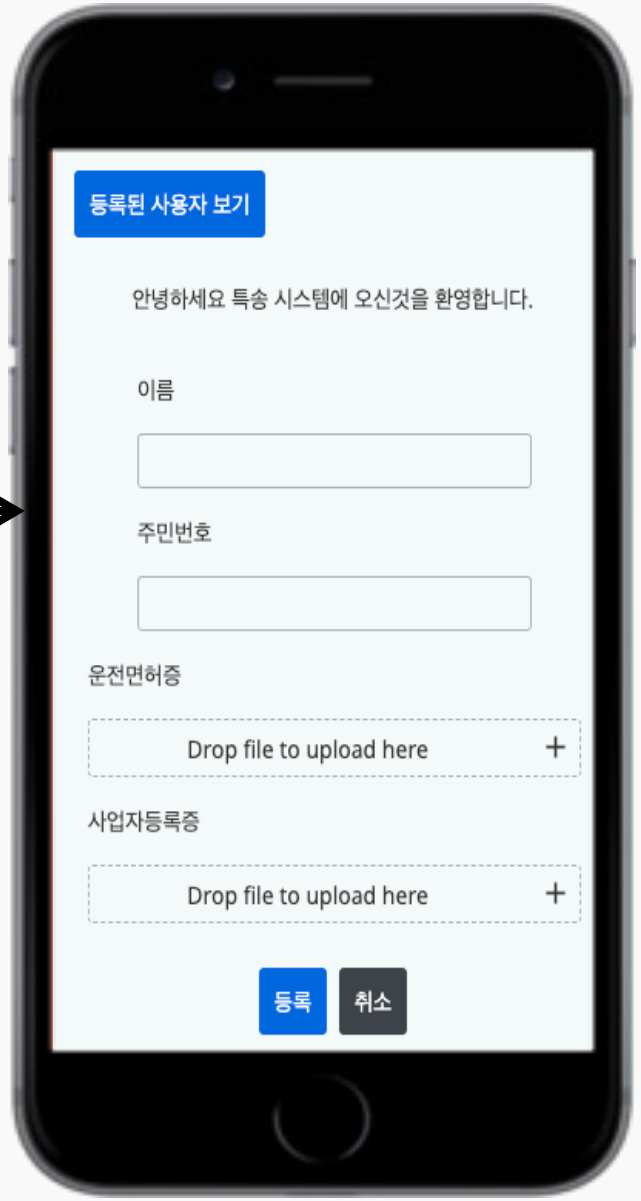
OCR Field와 Uipath Studio에서 사용하는 Field Table을 생성하는 알고리즘 개발



Progress

- Naver Clova OCR 연동 구현
- Enterprise Web 개발 진행중





Future Plan

Our Mission

유아이패스 프로젝트	시작일	작업일수	종료날짜	진행률	진행일수
개발환경 설치(UIPASS,VS)	2023-03-06	8	2023-03-14	100%	8
테스트 환경 설정-전 팀원	2023-03-11	6	2023-03-17	100%	6
Azure,네이버 클로바 API구상 및 자료조사-전 팀원	2023-03-11	17	2023-03-28	100%	17
패키지 파일 생성-김기태, 이재영	2023-03-26	5	2023-03-31	100%	5
AzureInvoice.cs 개발-명노아,이재영,김기태	2023-03-26	30	2023-04-25	100%	30
Chalres Extractor 데이터 필드 수정 작업-박경모	2023-04-09	23	2023-05-02	100%	23
제 3자 솔루션 사업자 연동-박경모,이재영,김기태	2023-04-23	12	2023-05-05	100%	12
Uipath 테스트 액티비티 작성-김기태, 박경모	2023-04-28	5	2023-05-03	100%	5
Azure Invoice 좌표 출력 기능 구현	2023-05-03	10	2023-05-13	90%	9
ClovalIDCard.cs 개발-박경모, 명노아, 이재영	2023-05-12	6	2023-05-18	100%	6
AzureInvoiceDesigner.xaml 환경 설정-이재영, 박경모	2023-05-14	10	2023-05-24	50%	5
ClovalBusineseLicense.cs 개발 - 김기태, 박경모	2023-05-14	10	2023-05-24	50%	5
AzureInvoice.cs token에러 수정	2023-05-22	7	2023-05-29	0%	0
ClovalDriverLicense.cs 개발 - 이재명, 명노아	2023-05-27	7	2023-06-03	30%	2.1
ClovalDriverCard.cs 개발-명노아,박경모, 이재영	2023-05-30	4	2023-06-03	0%	0
데모 형태의 유저 시나리오 제작-김기태,박경모,이재영	2023-05-29	6	2023-06-04	10%	0.6
네이버 클로바 좌표 중복 출력 기능 수정, 구현-명노아, 이	2023-06-05	7	2023-06-12	0%	0
기업 소프트웨어 저작권 등록-이재영,명노아	2023-06-12	1	2023-06-13	0%	0

Future Plan

Our Mission

Thank you!