

과목정보	과목명	창의공학기초설계		과목코드	30000(06)
	담당교수	노창배 교수님		제출일자	2021년 11월 25일
과제수행형태	팀 프로젝트				
팀 명	Wiser Wise				
과제명	대학행정정보시스템(wise)를 개선하는 방안 설계하기				
팀구성원	성명	학번	역할	분담내용	
	이명규	2020540023	팀장 (개인)	문제점 파악, 자료수집, 설계 및 코딩	
과제개요	수업계획서를 찾아보거나 다른 작업을 하기 위해, wise를 사용하다보면 여러 크고 작은 불편함을 느끼고는 했다. 이런 불편함을 해결하기 위해 직접 wise의 단점을 보완하고, 개선하는 방법을 설계한다.				
수행방법	<p>아래의 과정을 따라 과제를 수행한다.</p> <p>1. wise를 사용하면서 겪을 수 있는 문제점, 불편한 점, 아쉬운 점 등을 조사해본다.</p> <p>2. 해당 문제들을 해결할 수 있는 방안들을 떠올려본다.</p> <p>3. 각 방안들을 수행하기 위해 알아야 하는 개념들을 조사해보고, 여러 제한 사항들을 고려했을 때 현실적으로 가능한 방안들을 선택한다.</p> <p>4. 해당 방안에 대한 구체적인 설계를 진행하고, 이를 진행한다.</p> <p>5. 진행 과정에서 발생하는 문제점들을 해결하면서, 중간 결과물을 만든다.</p> <p>6. 이에 대한 피드백 과정을 진행한 후, 최종 결과물을 만든다.</p> <p>아래와 같은 평가 기준을 통해 과제가 잘 되었는지 판단한다.</p> <p>1. 기능성(50): wise를 사용하면서 느끼던 불편함 중 기능적인 부분들을 보완하고 개선하여 이용에 편리함을 제공한다.</p> <p>2. 심미성(50): 기존의 wise가 제공하던 것보다 이용하기 더 편하고 직관적인 UI를 제공한다.</p>				
수행효과	<p>과제를 통해 이루고자 하는 목표는 다음과 같다.</p> <p>서울시립대학교 대학행정정보시스템(wise.uos.ac.kr)를 사용하면서, UI/UX의 측면에서 불편함과 아쉬움을 많이 느껴왔다. 예를 들어, 강의계획서를 검색하는 과정에서는 아래와 같은 불편함이 있었다.</p> <p>1. 강의계획서 외의 여러 메뉴를 보고 싶어도, wise 내의 프로그램 탭은 4개를 초과할 수 없다.</p>				

	<p>2. 강의 하나를 검색하려면 교과목명이나 과목코드 만으로는 부족하며, 해당 교과목의 연도, 학기, 단과대학, 학과명 등에 대한 입력이 모두 이루어져야 한다. 과목명만 알고 어느 학과의 과목인지 모르는 경우에는 검색이 힘들다. 혹은 평소에는 사용할 일이 없는 교과번호를 알고 있어야 한다.</p> <p>3. 2의 과정을 거쳐서 검색을 하더라도, 강의계획서를 볼 때 창(브라우저) 안에 창(1에서 말한 4개의 탭) 안에 창(강의계획서)이 뜨는 형식으로 되어 있다.</p> <p>4. 3의 문제점에 더하여, 강의계획서 자체는 각 섹션에 대한 영역의 크기가 고정되어 있어서 내용이 좀만 길어도 스크롤을 통해 글을 읽어야 하며, 한 브라우저 창에서 창 안에 창 형식이 여러 번 반복되면서 내가 원하는 스크롤을 내리기조차 힘들다.</p> <p>5. 결국 강의계획서를 다 보고, 마침내 강의계획서 창을 닫으려고 하여도 '닫기'를 누르기까지 앞서 말한 불편함을 다시 한번 겪어야 한다. 강의계획서에 대한 이야기만을 하였지만, 다른 메뉴를 이용할 때도 이와 유사한 경험을 하리라고 쉽게 예상할 수 있다.</p> <p>따라서 강의계획서 하나를 보더라도 편하게 찾아보고자 wise의 일부 기능을 기존의 방법보다 더 개선된 방법으로 사용가능한 프로그램을 설계하고, 이를 앱 혹은 웹 등의 방식으로 설계 및 제작하는 것을 목표로 한다.</p>
수행환경	<p>제한조건과 고려사항은 다음과 같다.</p> <p>1. wise에서 제공하는 OPEN API 이외의 기능들은, 구현을 하는 데 있어서 어려움이 있을 수 있으므로 API에서 제공하는 기능을 사용한다.</p> <p>2. 실제로 앱 혹은 웹으로 제공한다면 그 과정에서 전기비나 서버 운용 비용 등의 비용이 발생할 수 있다.</p> <p>3. 앱과 웹 같은 형태의 서비스를 제공하기 위해서, 새로운 분야의 지식을 익히고 이를 적용하는 연습이 필요하다.</p>

- 간트차트 (수정)

항목	세부내용	10월	11월	12월
문제정의	wise의 불편사항들 확인			
정보수집	API, 파싱 등의 개념과 방법 조사			
해결책 생성	여러 해결 방안 수립			
분석과 선택	최적의 해결책 선택			
프로토타입 만들기	기본적인 기능을 구현			
테스트와 성능 개선	불편사항이 해결되었는지 테스트			
설계의 구현	실제 프로그램 혹은 서비스 완성			

- 목차

1. 문제 정의

2. 정보수집

-1. API

1.1 api란

1.2 서울시립대학교의 OPEN API?

-2. XML

2.1 XML이란?

2.2 실제 모습

-3. 파싱

-4. 필요한 파이썬 라이브러리

4.1 requests 라이브러리

4.2 xml 라이브러리

-5. 정보 출처

3. 해결책 생성

4. 분석과 선택

5. 프로토타입 만들기

5.1 requests 라이브러리만을 사용하여 만든 프로그램

5.2 requests 와 xml.etree.ElementTree 을 사용하여 만든 프로그램

5.3 pyinstaller을 통해 CUI로 구현

5.4 flask를 통해 간단한 웹 서버 구현

5.5 카카오톡 챗봇 완성

6. 테스트와 성능 개선

1. 문제 정의

위의 표에서 언급했듯이, 과제를 통해 해결하고자 하는 주된 불편함들은 다음과 같다.

1. 강의 하나를 검색하려면 교과목명이나 과목코드 만으로는 부족하며, 해당 교과목의 연도, 학기, 단과대학, 학과명 등에 대한 입력이 모두 이루어져야 한다. 특히나, 과목명만 알고 어느 학부(과)의 과목인지 모르는 경우에는 검색이 힘들다.

[illegible]

위의 사진은 창의공학기초설계 교과목을 검색하기 위해서, wise에서 검색을 하려고 하지만 어느 학부(과)의 강의인지를 몰라서 검색에 어려움을 겪는 상황을 가정하고 캡처한 사진이다. 사진에서 확인이 가능하듯이 어느 단과대의 강의인지, 어느 학부(과)인지 고르는 항목이 필수이기 때문에 모든 학과를 하나하나 검색을 해보는 수밖에 없다.

2. 앞선 과정을 거쳐서 검색을 하더라도, 강의계획서를 볼 때 창(브라우저) 안에 창(1에서 말한 4개의 탭) 안에 창(강의계획서)이 뜨는 형식으로 되어 있다.

총 2 건														합계		장영					
NO	학부(과)	교과구분	채우영역	교과번호	분	교과목명	학 년	학 년	당 당	주 어	강의유형	강의시간및강의실	학년별 수강인원	학년별 수강인원	전제 수강인원	전제 수강인원	타교 수강인원	복수 전공	부정전	유형비율	수업 계획
1	컴퓨터과학부	전공필수	전공필수	30000	06	창의공학기초설계	1	3	노현아	주강		목10,11,12,13/37-LI-602	34	35	34	35	Y	Y	Y	N	계획
2	컴퓨터과학부	전공필수	전공필수	30000	06	창의공학기초설계	1	3	노정환	주강		금10,11,12,13/37-LI-602	31	35	31	35	Y	Y	Y	N	계획

검색에 성공하였지만, 계획서를 보기 힘든 화면이다. 캡처 실수가 아니라 실제화면이 저기까지만 나오기 때문에 계획서를 보려면 아래에 있는 스크롤을 마우스로 클릭하여 화면을 이동해야 한다. (횡 방향이라서, 마우스 스크롤도 사용할 수 없다.)

3. 이에 더하여, 강의계획서 자체는 각 섹션에 대한 영역의 크기가 고정되어 있어서 내용이
 좀만 길어도 스크롤을 통해 글을 읽어야 하며, 한 브라우저 창에서 창 안에 창 형식이 여러
 번 반복되면서 내가 원하는 스크롤을 내리기조차 힘들다.

학부(과) 컴퓨터과학부

2021 학년도 2학기

컴퓨터과학부

노창배

실제로, 창의공학기초설계 교과목의 강의계획서를 검색하면 4개의 스크롤바(사진의 오른쪽에 위치한, 진한 회색의 막대 4개)가 있는 것을 확인할 수 있다. 오른쪽부터 각각 브라우

저, wise의 자체 탭, 강의계획서 탭, 강의계획서 내부의 표를 컨트롤하기 위한 탭임을 알 수 있다.

따라서 강의계획서 하나를 보더라도 편하게 찾아보고자, wise의 일부 기능(특히 강의 검색과 강의계획서 조회)을 기존의 방법보다 더 개선된 방법으로 사용가능한 프로그램을 설계하고, 이를 앱 혹은 웹 등의 방식으로 설계 및 제작하는 것을 목표로 한다.

2. 정보수집

1. API(Application Programming Interface)

-1.1 API란?

응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다. 주로 파일 제어, 창 제어, 화상 처리, 문자 제어 등을 위한 인터페이스를 제공한다. (위키백과)

API를 사용하면 구현 방식을 알지 못해도 제품 또는 서비스가 서로 커뮤니케이션할 수 있으며 애플리케이션 개발을 간소화하여 시간과 비용을 절약할 수 있습니다. 새로운 툴과 제품을 설계하거나 기존 툴과 제품을 관리하는 경우 API는 유연성을 제공하고 설계, 관리, 사용 방법을 간소화하며 혁신의 기회를 제공합니다. (레드햇, <https://www.redhat.com/ko>)

-1.2 서울시립대학교의 OPEN API?

서울시립대학교 대학행정정보시스템(<https://wise.uos.ac.kr/>)에서 기본적인 신상정보(이름과 소속, 휴대전화번호, e-mail)와 사용목적(개인학습, 과제제출, 서비스제공, 기타) 등을 입력하여 신청하면 쉽게 이용이 가능하다. 아래 사진은 실제 신청시 보는 화면이다.

실제로 제공되는 OPEN API 목록은 아래와 같다.

총 15 건

NO	프로그램ID	메뉴명	개발명세 파일
1	ApiCoursePlanView	수업계획서	보기
2	ApiBuildingRoomList	건물/강의실	보기
3	ApiDeptList	부서검색	보기
4	ApiMainBd	공지사항/학사일정	보기
5	ApiSubjectList	교과목검색	보기
6	UcsCourseTimeTableRoomList	강의실별 시간표	보기
7	UcrMjTimeInq	전공시간표조회	보기
8	UcsNotPermitSubjectStd	수강신청금지과목조회	보기
9	UcrCultTimeInq	교양시간표조회	보기
10	UcrConvTimeInq	융복합시간표조회	보기
11	UcsCourseTimeTableAll	종합시간표조회	보기
12	UcsFromToEmptyRoom	기간별빈강의실조회	보기
13	UrsRegPaidDayInq	등록금납부일정보	보기
14	UcsSameSubjectList	동일대체과목조회	보기
15	UcsLecturerEstimateResultInq	수업평가결과조회	보기

특히 수업계획서와 교과목검색을 이용하여 강의를 검색하고, 강의계획서를 보는 기능을 구현할 계획이고, 추가적으로 다른 기능도 필요하다면 추가해볼 계획이다.

상세 개발 명세를 확인해보면 아래와 같이 요청 URL과 요청 변수가 주어진다.

▶ 요청 URL(request url)

<https://wise.uos.ac.kr/uosdoc/api.ApiApiCoursePlanView.oapi>

▶ 요청 변수(request parameter)

요청변수키	유형	요청변수명	필수여부	설명
apiKey	문자	인증키	Y	Open API 신청시 발급받은 17자리 문자 Key
year	문자	학년도	Y	년도를 입력(예: 2013)
term	코드	학기	Y	학기 코드를 입력(예: A01) * A10 : 1학기 A20 : 2학기
subjectNo	문자	교과번호	Y	교과번호를 입력(예: 71005) * 교과목 API 개발명세 참조
classDiv	문자	분반	Y	분반을 입력(예: 01)

또한, 입력에 따른 출력 결과의 출력변수명과 설명을 확인할 수 있다.

▶ 출력 결과 필드(response field)

출력변수	출력변수명	설명
subject_no	교과번호	교과번호 정보입니다.
subject_nm	교과목명	교과목명 정보입니다.
prof_nm	성명	담당교수 정보입니다.
tel_no	연락처	담당교수 연락처 정보입니다.
score_eval_rate	성적평가	성적평가 정보입니다.
book_nm	교재	교재 정보입니다.
lec_goal_descr	교과목설명	교과목설명 정보입니다.
curi_edu_goal_nm	수업목표	수업목표 정보입니다.
week	주	(주별) 주 정보입니다.
class_cont	수업내용	(주별) 수업내용 정보입니다.
class_meth	수업방법	(주별) 수업방법 정보입니다.
week_book	교재	(주별) 수업교재 정보입니다.
prf_etc	준비물, 과제, 기타	(주별) 준비물, 과제, 기타 정보입니다.

2. XML

-2.1 XML이란?

OPEN API를 통해 강의계획서 정보를 받는 것이 목표인데, 해당 정보는 XML이라는 형태로 주어진다. 그렇다면 XML은 무엇일까?

XML(eXtensible Markup Language)은 W3C에서 개발된, 다른 특수한 목적을 갖는 마크업 언어를 만드는데 사용하도록 권장하는 다목적 마크업 언어이다. XML은 주로 다른 종류의 시스템, 특히 인터넷에 연결된 시스템끼리 데이터를 쉽게 주고 받을 수 있게 하여 HTML의 한계를 극복할 목적으로 만들어졌다. 기계는 인간의 언어를 읽거나 이해할 수 없는 계산기에 불과하므로 XML과 같은 구조화된 마크업 언어들은 인간의 읽고 분석하여 이해하는 능력

과 컴퓨터의 단순한 계산적인 판독 능력 사이에 타협점을 만들어 줄 수 있다. (위키백과)

-2.2 실제 모습

This XML file does not appear to have any style information associated with

```
<?xml version="1.0"?>
<mainlist>
  <list>
    <year>
      <![CDATA[ 2021 ]]>
    </year>
    <term>
      <![CDATA[ 20 ]]>
    </term>
    <subject_no>
      <![CDATA[ 30000(06) ]]>
    </subject_no>
    <subject_nm>
      <![CDATA[ 창의공학기초설계(Introduction to Creative Engineering Design) ]]>
    </subject_nm>
    <prof_nm>
      <![CDATA[ 노창배 ]]>
    </prof_nm>
    <tel_no>
      <![CDATA[ 010-2706-2567 ]]>
    </tel_no>
    <score_eval_rate>
      <![CDATA[ □ 출석 (20)% □ 학생포트폴리오 (20)% □ 참여도 (0)% □ 수시과제 (40) ]]>
    </score_eval_rate>
    <book_nm>
      <![CDATA[ - 창의공학설계, 생능출판사, 2021년, 김대수 지음 ]]>
    </book_nm>
    <week>
```

URL을 입력하여 웹 브라우저를 통해 창의공학기초설계 교과목의 강의계획서 정보를 확인하는 모습이다.

3. 파싱

해당 정보가 XML이라는 마크업언어로 제공된다는 사실은 알았다. 그렇다면 내가 원하는 정보, 예컨대 주차별 학습 내용이나 성적 평가 정보 등을 알고자 한다면 어떤 과정을 거쳐야할까? 이런 과정을 파싱이라고 한다.

파싱은 어떤 페이지(문서, html 등)에서 내가 원하는 데이터를 특정 패턴이나 순서로 추출해 가공하는 것을 말한다.(사이언스올)

4. 필요한 파이썬 라이브러리

파이썬을 통해 해당 작업들을 하게 될 것이며, 이러한 기능을 가능하게 해주는 라이브러리가 존재하기 때문에 각 라이브러리에 대해 알아볼 것이다.

-4.1 requests 라이브러리

Python 프로그래밍 언어용 HTTP 라이브러리이다. 목표는 HTTP 요청을 더 간단하고 인간 친화적으로 만드는 것이다. (위키백과)

학교에서 제공하는 OPEN API의 경우, url을 통해 어떤 정보가 필요한지 요청을 보내면 xml로 정보를 보내주는 형태의 서비스를 제공한다. 이때 url을 통해 어떤 정보가 필요한지 요청을 보내는 역할을 해주는 것이 이 라이브러리이다.

-4.2 xml 라이브러리

제공받은 xml 형태의 정보를 사용하기 편한 형태로 파싱을 해주는 라이브러리(패키지)가 xml 패키지이며, xml 패키지 중에서 특히 xml.etree.ElementTree 모듈을 알아볼 것이다.

xml.etree.ElementTree 모듈은 XML 데이터를 구문 분석하고 만들기 위한 단순하고 효율적인 API를 구현한다. "XML은 본질적으로 위계적(hierarchical) 데이터 형식이며, 이를 나타내

는 가장 자연스러운 방법은 트리를 사용하는 것입니다. ET에는 이 목적을 위한 두 가지 클래스가 있습니다 - ElementTree는 전체 XML 문서를 트리로 나타내고, Element는 이 트리에 있는 단일 노드를 나타냅니다. 전체 문서와의 상호 작용(파일 읽기와 쓰기)은 일반적으로 ElementTree 수준에서 수행됩니다. 단일 XML 엘리먼트와 해당 서브 엘리먼트와의 상호 작용은 Element 수준에서 수행됩니다.” (공식문서)

5. 정보 출처

-API

위키백과, <https://ko.wikipedia.org/wiki/API>

레드햇, <https://www.redhat.com/ko/topics/api/what-are-application-programming-interfaces>

-XML

위키백과, <https://ko.wikipedia.org/wiki/XML>

- 파싱

사이언스올, <https://www.scienceall.com/%ED%8C%8C%EC%8B%B1parsing/>

- requests 라이브러리

위키백과, [https://en.wikipedia.org/wiki/Requests_\(software\)](https://en.wikipedia.org/wiki/Requests_(software))

공식 문서, <https://docs.python-requests.org/en/latest/>

- xml 라이브러리

공식 문서, <https://docs.python.org/ko/3/library/xml.html>

공식 문서, <https://docs.python.org/ko/3/library/xml.etree.elementtree.html>

3. 해결책 생성

우선 어떤 형식으로 문제를 해결하더라도 기본적으로 WISE의 OPEN API를 통해 정보를 제공받아서 이를 사용자가 보기 편하게 가공하여 보여준다는 공통점을 가진다.

하지만 제공받은 정보를 어떤 식으로 제공할지에 대해서는 다양한 방법이 있을 수 있다. 예를 들어, 몇몇 불편점이 있던 wise보다 좀 더 보기에 편한 웹사이트를 제작하여 강의계획서를 보여줄 수도 있고, exe 파일로 만들어 CMD 창에서 실행하듯이 CUI 형태로 제공할 수도 있으며, 카카오톡 챗봇을 이용하여 카카오톡에서 대화를 하듯이 정보를 찾아볼 수도 있을 것이다.

아래의 표는 각 방법의 장단점을 간략히 정리한 표이다.

	웹	CUI	카카오톡 챗봇
장점	1. 카카오톡과 마찬가지로, 모바일로 이용이 가능할 것으로 예상됨. 2. 3가지 방법 중 심미적으로 가장 아름다운 디자인이 가능함.	1. 구현이 간단할 것으로 예상됨. 2. 디자인이 호불호가 갈리지 않게 매우 깔끔함.	1. 모바일로 이용이 가능하며, 복잡한 디자인을 따로 해줄 필요가 없음. 2. 다른 방법에 비해 딱딱하지 않고, 이야기하듯이 서비스를 이용할 수 있음.
단점	1. 웹에 대해 추가적으로 공부해야 하기 때문에, 구현이 쉽지 않을 것으로 예상됨. 2. 구현을 제대로 못하면, 디자인이 CUI만큼 멋있지 않을 수 있음.	1. 디자인이 너무 단순하여 밋밋하며, exe 파일을 실행할 수 있는 환경에서만 이용가능함. 2. Apikey를 함께 넘겨주어야 하기 때문에 보안상의 위험성이 존재함.	1. 웹과 마찬가지로, 구현을 하는 방법에 대해 공부해야 함. 2. 디자인의 측면에서 제약이 있음. 말풍선, 사진, 카드 리스트 정도의 한계를 지님.

4. 분석과 선택

위의 3가지 방법들을 분석해보고 가장 좋다고 판단되는 방법을 선택하여 진행한다. 평가기준은 1페이지의 표에서 설명했던 것과 같이 '기능성 50 + 심미성 50'으로 채택하였다.

기능성의 경우, 각 방법들의 난이도와 관련이 있다. 예컨대, 제한된 시간 동안 상대적으로 어려운 웹 사이트를 만들려고 하면, 관련 배경 지식을 익히기 위해서 시간을 많이 써야하기 때문에 기능적인 구현을 제대로 못할 수도 있을 것이다. 반면에 상대적으로 쉬운 CUI의 경우 추가적으로 익힐 지식 등이 적기 때문에 기능의 구현에 집중할 수 있을 것이고, 결국 기능적으로 구현을 잘 할 수 있을 것이다.

또한, 심미성의 경우 위의 표에서 서술했듯이 웹의 경우 다른 두가지 방법들에 비해 자유도도 높고, 이미 만들어지고 사용가능한 사이트들도 많이 있기 때문에 심미적으로 완성도가 높은 사이트를 만들 수 있다. 챗봇의 경우에는 자유도가 낮다는 단점이 있고, 말풍성이나 사진, 카드 리스트 등의 제한이 존재한다. CUI의 경우에는 cmd 창에서와 같이 자유도도 낮고, 심미성도 떨어진다.

이들을 복합적으로 고려했을 때, 웹사이트에 비해 상대적으로 어렵지 않으며, CUI에 비해 적당히 심미성을 보장할 수 있는 카카오톡 챗봇을 선택하였다. 물론 난이도가 낮은 편은 아니지만 시간을 충분히 활용한다면 가능할 것이라고 생각하였고, 위의 평가기준 외에도 카톡만 키면 확인이 가능하다는 접근성 등의 특징을 고려하였다.

가 조금만 달라도 올바른 정보를 추출하지 못하는 것을 확인할 수 있다.

```
검색을 원하는 과목의 교과번호를 입력해주세요: 30000
검색을 원하는 과목의 교과번호를 입력해주세요: 06
Traceback (most recent call last):
  File "C:/Users/이명규/Desktop/파이썬/수업계획서 조회/시립대_api_수업계획서_1.py", line 16, in <module>
    if 'CDATA' in html.split('>')[i][3:-2]:
IndexError: list index out of range
```

프로그램 실행 도중 오류가 발생하여, 정보를 받아오는 것에 조차 실패한 것을 확인할 수 있다.

5.2 requests 와 xml.etree.ElementTree 을 사용하여 만든 프로그램

```
import requests
import xml.etree.ElementTree as ET
year, term, classDiv = '2021', '2', '01'
apiKey = '00000000000000000000000000000000'
#year = input('검색하고자 하는 연도를 입력해주세요(예: 2021): ')
#term = input("검색하고자 하는 학기를 '숫자만' 입력해주세요(예: 1): ")
subjectNo = input('검색을 원하는 과목의 교과번호를 입력해주세요(예: 71005): ')
classDiv = input('검색을 원하는 과목의 분반을 입력해주세요(예: 01): ')
print('-----\n출력')

url = 'https://wise.uos.ac.kr/uosdoc/api.ApiApiCoursePlanView.oapi'
option = f'?apiKey={apiKey}&year={year}&term=A{term}0&subjectNo={subjectNo}&classDiv={classDiv}'
xml = requests.get(url + option).text

common = dict()
each_week = [dict()] # [{}, {week:~, class_cont:~, ...}, {week:~, class_cont:~, ...}, ~~~]
each_week_tag_set = {'week', 'class_cont', 'class_meth', 'week_book', 'prjt_etc'}
translate = {'subject_no': '교과번호', 'subject_nm': '교과목명', 'prof_nm': '담당교수 성명', 'tel_no': '담당교수 연락처',
             'score_eval_rate': '성적평가 정보', 'book_nm': '교재', 'lec_goal_descr': '교과목 설명', 'curi_edu_goal_nm': '수업 목표',
             'week': '주', 'class_cont': '수업내용', 'class_meth': '수업방법', 'week_book': '수업 교재',
             'prjt_etc': '준비물, 과제, 기타 정보'}

try:
    root = ET.fromstring(xml)
    subject_tree = root[0]
    for week_num in range(16):
        week_info = subject_tree[week_num]
        week = week_num + 1
        week_dict = dict()
        for information in week_info:
            if information.tag not in each_week_tag_set and information.tag not in common:
                common[information.tag] = information.text
            elif information.tag in each_week_tag_set:
                week_dict[information.tag] = information.text
        each_week.append(week_dict)

    for element in common:
        if not common[element] is None:
            no_enter_list = common[element].split('\n')
            no_enter = ' '.join(no_enter_list)
            no_tab_list = no_enter.split('\t')
            no_tab = ' '.join(no_tab_list)
            if element in translate:
                print(f"{translate[element]}은(는) {no_tab.strip()} 입니다.")
            print()
```

```

for i in range(1, 17):
    print(f"{i}주차의")
    week_inform = each_week[i]
    for element in week_inform:
        if element != 'week' and week_inform[element] is not None:
            no_enter_list = week_inform[element].split('\n')
            no_enter = ' '.join(no_enter_list)
            no_tab_list = no_enter.split('\t')
            no_tab = ' '.join(no_tab_list)
            if element in translate:
                print(f'{translate[element]}은(는) {no_tab.strip()} 입니다.')
    print()
except:
    print('입력이 올바른지 확인해주십시오.')

```

requests를 통해 정보를 받는 것은 동일하지만, xml.etree.ElementTree을 사용한 부분이 달라졌다. 해당 모듈을 사용할 경우, 트리 형태로 자료를 변환해주기 때문에 실제로 사용하면서 더욱 안정적이다. 또한, try~ except~ 문을 사용하여 입력이 올바르지 않다면, 입력을 확인해달라는 문구를 추가하였다.

- 실행결과

```

-----
출력
교과번호은(는) 30000(06) 입니다.

교과목명은(는) 창의공학기초설계(Introduction to Creative Engineering Design) 입니다.

담당교수 성명은(는) 노창배 입니다.

담당교수 연락처은(는) 010-2706-2567 입니다.

성적평가 정보는(는) □ 출석 (20)% □ 학생포트폴리오 (20)% □ 참여도 (0)% □ 수시과제 (40)% □ 수
교재은(는) - 창의공학설계, 생능출판사, 2021년, 김대수 지음 입니다.

교과목 설명은(는) 공학설계 능력 배양에 필수적인 창의적 사고, 팀 활동, 의사소통능력을 증진시키고, PB
수업 목표은(는) 국가 산업발전에 이바지 할 수 있는 능력과 소 양을 배양하여 컴퓨터 전문가를 육성하는 것

1주차의
수업내용은(는) ?창배 노이(가) 예약된 Zoom 회의에 귀하를 초대합니다. 주제: 1주차 오리엔테이션 시간:
준비물, 과제, 기타 정보는(는) ?창배 노이(가) 예약된 Zoom 회의에 귀하를 초대합니다. 주제: 1주차 오

2주차의
수업내용은(는) - Team조직 강의 - Brain Storming - Ice Breaking 입니다.

3주차의
수업내용은(는) - 팀별 1차 과제 아이디어 창출 입니다.

4주차의

```

5.3 pyinstaller을 통해 CUI로 구현

pyinstaller (<https://www.pyinstaller.org/>) 라는 라이브러리를 통해 간단히 exe파일로 만들 수 있다고 하여 진행하였다. 특징은 해당 파이썬 프로그램에 외부 라이브러리가 쓰여졌더라도, 외부 라이브러리도 자체적으로 통합하여 exe로 만들어주기 때문에 사용자가 다른 라이브러리를 설치할 필요가 없다는 것이다. 즉 exe 파일만으로 실행이 가능하게 해준다.

아래는 각각 exe 파일로 만들어진 것과 실행 화면을 캡처한 사진들이다.

수업계획서.exe



5.4 flask를 통해 간단한 웹 서버 구현

웹 서버를 만들어주기 위해서 아래와 같이 구름ide에서 다음 코드를 작성하였다.

```
from flask import Flask, request, jsonify
import re
import sys, requests
import xml.etree.ElementTree as ET
application = Flask(__name__)

@app.route("/hello")
def hello():
    return "Hello goorm!"

@app.route("/requests_test")
def testt():
    response = requests.get("https://wise.uos.ac.kr/uosd")
    return str(response.status_code)
```

위의 사진은 사용할 라이브러리들을 import하고 Flask 객체 인스턴스를 생성하는 코드이다.


```

@application.route("/subject_search", methods=['POST'])
def subject():
    req = request.get_json()

    params = req['action']['params'] # 사용자의 발화로부터 추출한 파라미터 정보 / 엔티티
    subjectNm, year, term = params['subject_name'], params['year'], params['term']
    year = re.sub(r'^0-9', '', year) # 숫자만 추출: https://codechacha.com/ko/
    term = re.sub(r'^1-2', '', term)

    url = f"https://wise.uos.ac.kr/uosdoc/api.ApiApiSubjectList.oapi?apiKey=202008
    response = requests.get(url).text

    common = dict()
    each_subj = [dict()] # [{}, {week:~, class_cont:~, ...}, {week:~, class_cont:~
    each_subj_tag_set = {'subject_no', 'subject_nm', 'class_div', 'subject_div', '
    translate = {'subject_no': '교과번호', 'subject_nm': '교과목명', 'class_div': '분
                'prof_nm': '담당교수 성명', 'tel_no': '담당교수 연락처', 'score_eval
                'lec_goal_descr': '교과목 설명', 'curi_edu_goal_nm': '수업 목표', 'w
                'class_meth': '수업방법', 'week_book': '수업 교재', 'prjt_etc': '준
    message = ''

    try:
        root = ET.fromstring(response)
        subject_tree = root[0]
        length = len(subject_tree)
        if length:
            for num in range(len(subject_tree)):
                info = subject_tree[num]
                subj_dict = dict()
                for information in info:
                    if information.tag not in each_subj_tag_set and information.ta
                        common[information.tag] = information.text
                    elif information.tag in each_subj_tag_set:
                        subj_dict[information.tag] = information.text
                each_subj.append(subj_dict)

```

```

        for element in common:
            if not common[element] is None:
                no_enter_list = common[element].split('\n')
                no_enter = ' '.join(no_enter_list)
                no_tab_list = no_enter.split('\t')
                no_tab = ' '.join(no_tab_list)
                if element in translate:
                    message += f"{translate[element]}은(는) {no_tab}
                    message += '\n'

        for i in range(1, len(subject_tree)+1):
            message += f"{i}번째 검색결과입니다\n"
            subj_inform = each_subj[i]
            for element in subj_inform:
                if subj_inform[element] is not None:
                    no_enter_list = subj_inform[element].split('\n')
                    no_enter = ' '.join(no_enter_list)
                    no_tab_list = no_enter.split('\t')
                    no_tab = ' '.join(no_tab_list)
                    if element in translate:
                        message += f"{translate[element]}은(는) {no
                    message += '\n'
            else:
                message = "검색 결과가 존재하지 않습니다."
    except:
        message = '입력이 올바른지 확인해주세요!'
    print(message)
    res = {
        "contents": [
            {
                "type": "text",
                "text": f"{message}"
            }
        ]
    }

    # 전송
    return jsonify(res)

```

위의 두 사진은 강의를 검색하고자 할 때 실행되는 코드이다. 기존의 코드와 거의 동일하지만 검색 내용을 문자열로 생성하고 json의 형태로 만들어서 반환한다. 강의계획서를 조회하는 코드 역시 이와 유사하게 수정하여 작성하였다.

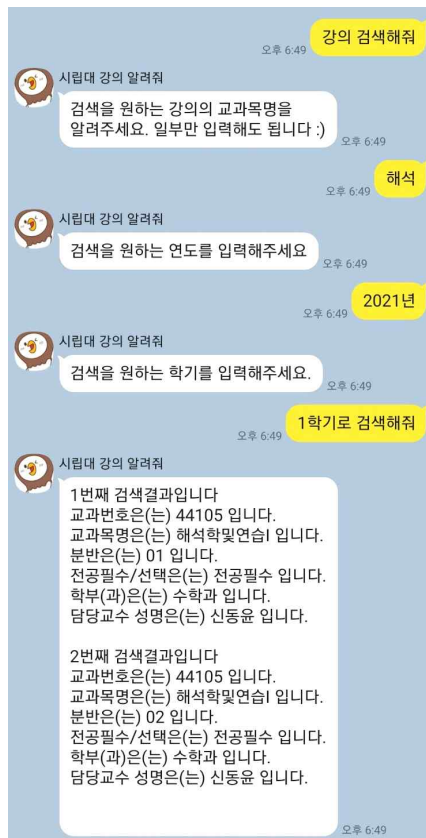
5.5 카카오톡 챗봇 완성



위의 사진과 같이 기본적으로 실행할 기능(여기서는 강의 검색과 강의 계획서 조회)을 정하고, 각 기능을 실행하기 위한 패턴 발화라는 것을 정의한다. 특징은 사용자 발화에서 중심적인 의미를 갖는 엔티티를 추출하여 의도를 파악한다는 것이다. (<https://i.kakao.com/docs/key-concepts-utterance>)



이후 스킬 탭에 들어가서, flask로 구름 ide에서 만들었던 웹 서버와 연결한다. 이때 강의 검색과 강의 계획서 조회 모두 연결한다. 이외에도 파라미터 설정 등을 하고, 미리 만들어두었던 카카오톡 채널과 연결하면 아래 사진과 같이 실행이 되는 것을 확인할 수 있다.



‘강의 검색해줘’라고 하여 강의 검색 기능을 실행시키고, 검색할 키워드와 검색 연도, 학기를 입력받은 뒤 실제로 검색 결과를 잘 출력하는 것을 확인할 수 있다.

6. 테스트와 성능 개선



위의 사진은 강의 계획서를 조회하기 위해서 교과번호와 분반 번호, 검색 연도와 학기를 입력하고 계획서를 확인하는 모습을 캡처한 것이다. 즉 앞에서 살펴본 것과 같이 강의 검색 기능 이외에도, 강의 계획서 조회 역시 잘 이루어진다는 것을 확인할 수 있다.

따라서 원하는 기능들(강의 검색과 강의 계획서 조회)이 잘 구현된 것을 확인할 수 있다. 다시 말해서, 처음에 해결하고자 했던 문제인 '강의 검색'과 '강의계획서 조회'의 불편성을 어느 정도 해결했다고 생각된다. 특히 강의를 검색하고 교과 번호를 확인한 후, 계획서 조회를 할 때 활용하면 더 편리하게 할 수 있다.

단점으로는 챗봇을 사용하기 위해서 웹 서버를 켜야 한다는 단점이 있다. 계속 컴퓨터를 켜두거나, 구름 ide에서처럼 클라우드 ide를 사용하는 경우 해당 사이트에서 제공하는 유료 서비스를 결제하여 24시간 켜두는 기능을 사용하여야 한다.

추가적인 개선 사항으로는 다른 추가적인 기능을 생각해 볼 수도 있을 것이다. 예컨대, 공지사항 확인이나 수업평가결과 조회 등 OPEN API로 제공되는 몇가지 유용한 기능들을 구현할 수도 있을 것이고, 꼭 wise와는 관련이 없더라도 도움이 될 만한 간단한 기능을 구현해볼 수도 있을 것이다.