Design Documentation

Mine-A-Niner

Daniel Bakhshandeh , Tanner Jarrell , Rachel Gutowski , Mohammad Naeem , Eddie Seitz , Tinsae Dejene

Table of Contents

Project Overview

Overview of Problem and Design

Faculty connections is an aggregation of information about University of North Carolina at Charlotte faculty and their field of research/interests. On the faculty connections website, researchers are categorized into their respective departments and have keywords associated with their research and interests. The current process for finding information is completed by hand and the staff at faculty connections are looking for an automated method for gathering the information into an organized format. Our software, Mine-A-Niner, aims to use web crawlers to scrape through UNCC department pages and extract information regarding UNCC staff and their fields of interest. Finally, our software organizes this information into a CSV which can be used by the faculty connections staff.

https://pages.charlotte.edu/connections/

Stakeholders

- **Faculty Connection Staff** - Faculty connections staff is our main stakeholder. They will primarily be using the CSV files to update the faculty connections website.
- **UNCC Students and faculty -** UNCC students and faculty are an indirect stakeholder in our software because they will be benefiting from a more accurate and updated faculty connections website.
- **Other individuals/entities interested in UNCC faculty -** Other individuals will also indirectly benefit because they will receive a more accurate and updated faculty connections website due to our software.

What our program does

Our software uses web crawlers to scrape UNCC department pages and extract text from the profiles of staff members at UNCC. The web crawlers are built from the Scrapy and BeautifulSoup4 framework. The software then uses rake_nltk, a natural language processing framework, to extract keywords from the text. Finally, all of this information is saved and exported as a CSV with each college and its department and the keywords being associated with staff of each department. This gives faculty connections an organized format for the faculty connections website.

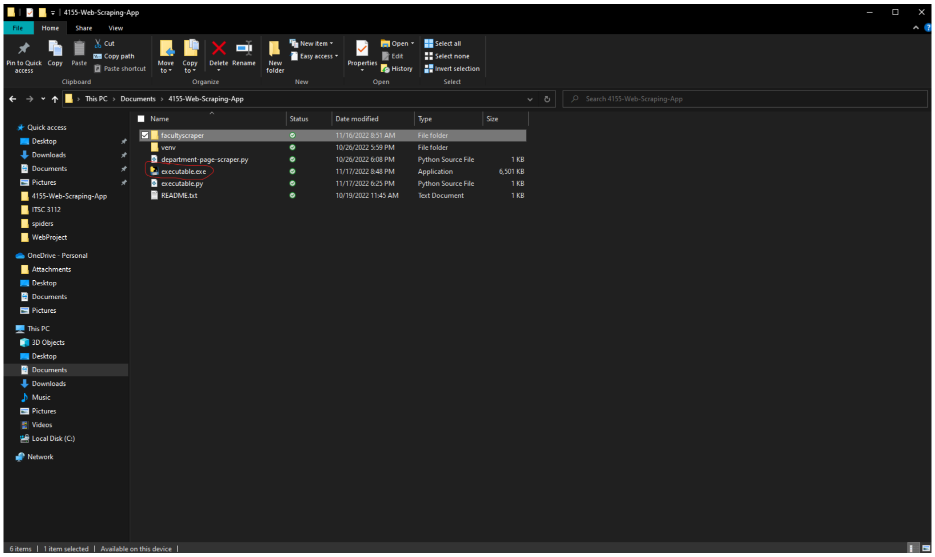
Design Rationale

The rationale behind the software is to create software that is easy to use, as non-technical as possible for the user, and accurate for our user. Our goal was to create a software that limits the interactions of the user with the software and ultimately achieves the goal of the stakeholder. In our prototypes we had a UI component that the stakeholder could use for manipulating the data, but was eventually removed as we decided the UI was not a part of the stakeholders requirements. Additionally, in sprints 1 + 2, we decided to add new frameworks such as Scrapy to ensure the software is quick and responsive for the user. Ultimately, as the software progressed, our main goals became to create efficient software that achieves the stakeholders needs.

Key User Stories

ST-1: As a user, I want the program to be an executable file, so that I can run it by opening the program.

- Software contains an executable file that allows the user to run the applications without a lot of interaction. Executable file contains the code that downloads packages onto the users machine and executes the calls for the crawlers and CSV exportation.



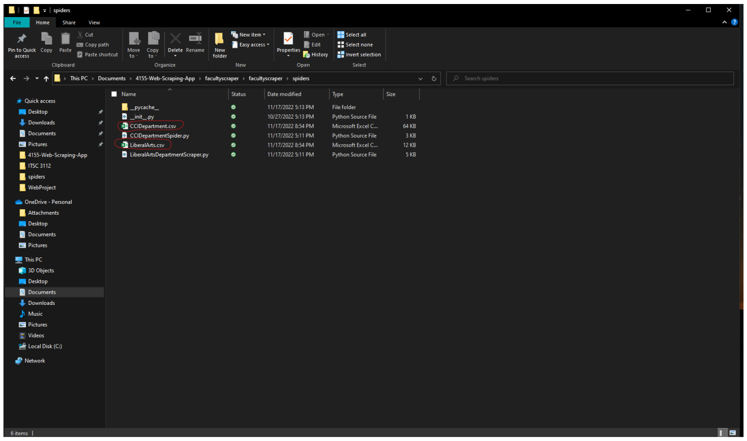
Name	Status	Date modified	Type	Size
facultyscraper	✓	11/16/2022 8:51 AM	File folder	
venv	✓	10/26/2022 5:59 PM	File folder	
department-page-scraper.py	✓	10/26/2022 6:08 PM	Python Source File	1 KB
executable.exe	✓	11/17/2022 8:48 PM	Application	6,501 KB
executable.py	✓	11/17/2022 6:25 PM	Python Source File	1 KB
README.txt	✓	10/19/2022 11:45 AM	Text Document	1 KB

ST-2: As a user, I want the data to be scraped by a bot, so that a lot of data can be provided to me without a lot of interaction.

- Program uses a bot that navigates pages and extracts data. This provides a fast and automated way to get the data needed.

ST-3: As a user, I want the scraping results to be presented in an organized and readable format.

- Software exports the scraped data as a CSV for the user to have an organized format.



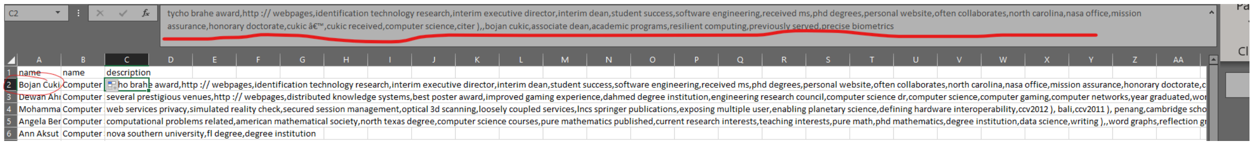
This PC › Documents › 4155-Web-Scraping-App › facultyscraper › facultyscraper › spiders

Name	Status	Date modified	Type	Size
__pycache__	✓	11/17/2022 5:13 PM	File folder	
__init__.py	✓	10/27/2022 5:13 PM	Python Source File	1 KB
CCIDepartment.csv	✓	11/17/2022 8:54 PM	Microsoft Excel C...	64 KB
CCIDepartmentSpider.py	✓	11/17/2022 5:11 PM	Python Source File	3 KB
LiberalArts.csv	✓	11/17/2022 8:54 PM	Microsoft Excel C...	12 KB
LiberalArtsDepartmentScraper.py	✓	11/17/2022 5:11 PM	Python Source File	5 KB

6 items

ST-6: As a user, I want researchers to be associated with keywords within their own field, so that I can have information that is accurate about the faculty member.

- The software pairs the keywords extracted from the department pages with the faculty. This allows the exported data to be accurate to the faculty member.



C2

tycho brahe award,http :// webpages,identification technology research,interim executive director,interim dean,student success,software engineering,received ms,phd degrees,personal website,often collaborates,north carolina,nasa office,mission assurance,honorary doctorate,cukic â€™,cukic received,computer science,citer),,bojan cukic,associate dean,academic programs,resilient computing,previously served,precise biometrics

	A	B	C
1	name	name	description
2	Bojan Cuki	Computer	ho brahe award,http :// webpages,identification technology research,interim executive director,interim dean,student success,software engineering,received ms,phd degrees,personal website,often collaborates,north carolina,nasa office,mission assurance,honorary doctorate,c
3	Dewan Ahı	Computer	several prestigious venues,http :// webpages,distributed knowledge systems,best poster award,improved gaming experience,dahmed degree institution,engineering research council,computer science dr,computer science,computer gaming,computer networks,year graduated,wor
4	Mohamma	Computer	web services privacy,simulated reality check,secured session management,optical 3d scanning,loosely coupled services,lncs springer publications,exposing multiple user,enabling planetary science,defining hardware interoperability,ccv2012), bali,ccv2011), penang,cambridge scho
5	Angela Ber	Computer	computational problems related,american mathematical society,north texas degree,computer science courses,pure mathematics published,current research interests,teaching interests,pure math,phd mathematics,degree institution,data science,writing),,word graphs,reflection gr
6	Ann Aksut	Computer	nova southern university,fl degree,degree institution

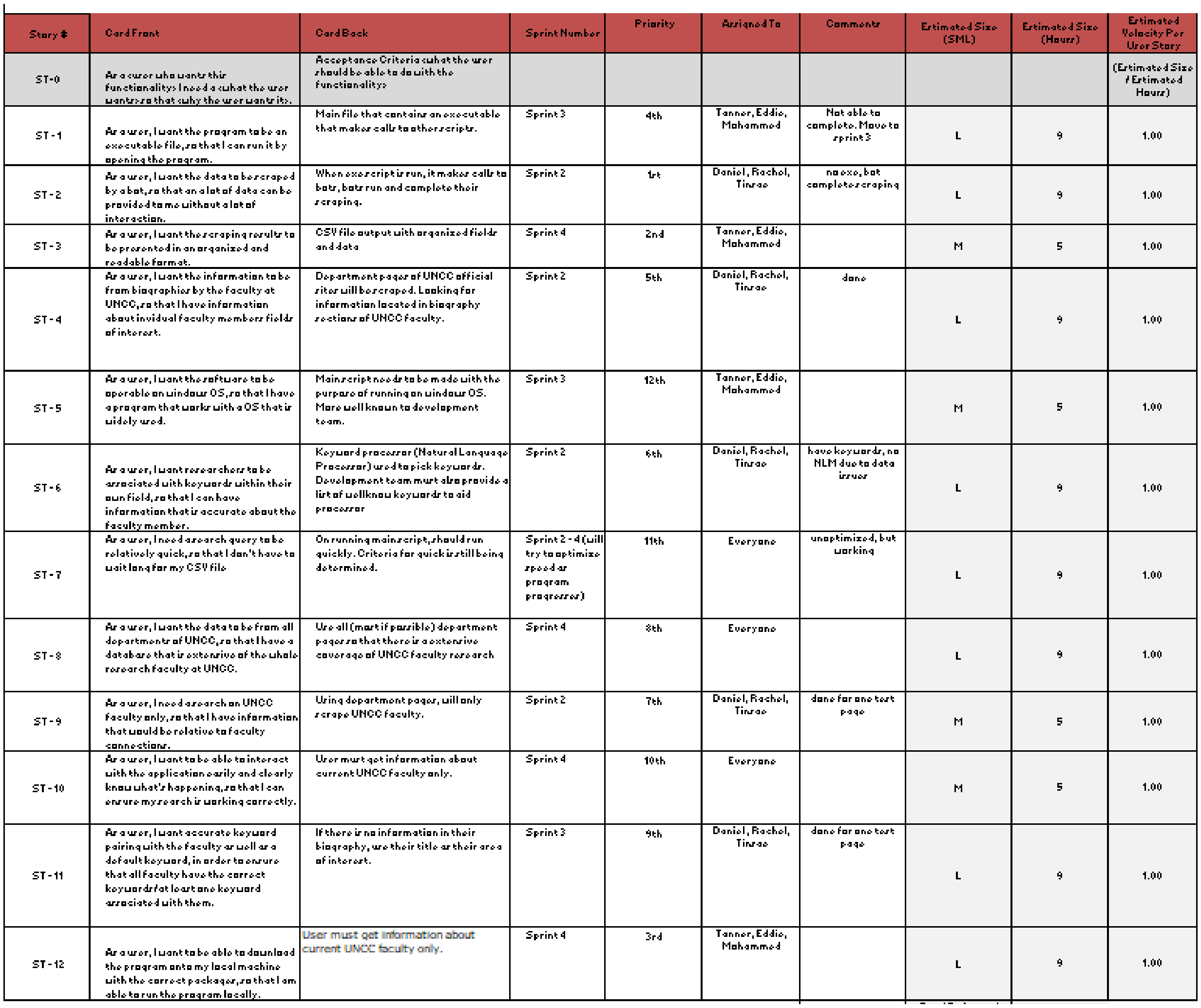
ST-8: As a user, I want the data to be from all departments of UNCC, so that I have a database that is extensive of the whole research faculty at UNCC.

- The software scrapes all scrapable departments at UNCC to ensure that a majority of the faculty connections staff are being covered.

ST-9: As a user, I need a search on UNCC faculty only, so that I have information that would be relative to faculty connections.

- The software only pulls data on UNCC faculty, ensuring that the information is beneficial for faculty connections.

User Stories

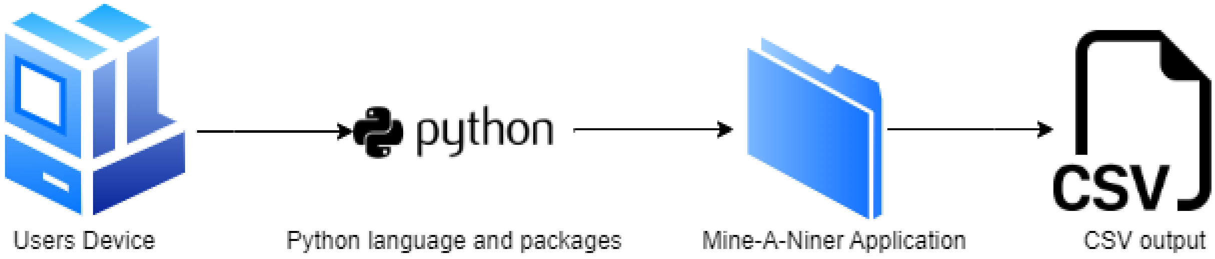


Story #	Card Front	Card Back	Sprint Number	Priority	Assigned To	Comments	Estimated Size (SML)	Estimated Size (Hours)	Estimated Velocity Per User Story
ST-0	As a user who wants this functionality: I need a <what the user wants> so that <why the user wants it>.	Acceptance Criteria <what the user should be able to do with the functionality>							(Estimated Size / Estimated Hours)
ST - 1	As a user, I want the program to be an executable file, so that I can run it by opening the program.	Main file that contains an executable that makes calls to other scripts.	Sprint 3	4th	Tanner, Eddie, Mohammed	Not able to complete. Move to sprint 3	L	9	1.00
ST - 2	As a user, I want the data to be scraped by a bot, so that an a lot of data can be provided to me without a lot of interaction.	When exe script is run, it makes calls to bats, bats run and complete their scraping.	Sprint 2	1st	Daniel, Rachel, Tinsae	no exe, but completes scraping	L	9	1.00
ST - 3	As a user, I want the scraping results to be presented in an organized and readable format.	CSV file output with organized fields and data	Sprint 4	2nd	Tanner, Eddie, Mohammed		M	5	1.00
ST - 4	As a user, I want the information to be from biographies by the faculty at UNCC, so that I have information about invidual faculty members fields of interest.	Department pages of UNCC official sites will be scraped. Looking for information located in biography sections of UNCC faculty.	Sprint 2	5th	Daniel, Rachel, Tinsae	done	L	9	1.00
ST - 5	As a user, I want the software to be operable on windows OS, so that I have a program that works with a OS that is widely used.	Main script needs to be made with the purpose of running on windows OS. More well known to development team.	Sprint 3	12th	Tanner, Eddie, Mohammed		M	5	1.00
ST - 6	As a user, I want researchers to be associated with keywords within their own field, so that I can have information that is accurate about the faculty member.	Keyword processor (Natural Language Processor) used to pick keywords. Development team must also provide a list of wellknow keywords to aid processor	Sprint 2	6th	Daniel, Rachel, Tinsae	have keywords, no NLM due to data issues	L	9	1.00
ST - 7	As a user, I need a search query to be relatively quick, so that I don't have to wait long for my CSV file	On running main script, should run quickly. Criteria for quick is still being determined.	Sprint 2 - 4 (will try to optimize speed as program progresses)	11th	Everyone	unoptimized, but working	L	9	1.00
ST - 8	As a user, I want the data to be from all departments of UNCC, so that I have a database that is extensive of the whole research faculty at UNCC.	Use all (most if possible) department pages so that there is a extensive coverage of UNCC faculty research	Sprint 4	8th	Everyone		L	9	1.00
ST - 9	As a user, I need a search on UNCC faculty only, so that I have information that would be relative to faculty connections.	Using department pages, will only scrape UNCC faculty.	Sprint 2	7th	Daniel, Rachel, Tinsae	done for one test page	M	5	1.00
ST - 10	As a user, I want to be able to interact with the application easily and clearly know what's happening, so that I can ensure my search is working correctly.	User must get information about current UNCC faculty only.	Sprint 4	10th	Everyone		M	5	1.00
ST - 11	As a user, I want accurate keyword pairing with the faculty as well as a default keyword, in order to ensure that all faculty have the correct keywords/at least one keyword associated with them.	If there is no information in their biography, use their title as their area of interest.	Sprint 3	9th	Daniel, Rachel, Tinsae	done for one test page	L	9	1.00
ST - 12	As a user, I want to be able to download the program onto my local machine with the correct packages, so that I am able to run the program locally.	User must get information about current UNCC faculty only.	Sprint 4	3rd	Tanner, Eddie, Mohammed		L	9	1.00

Architectural Overview

Type of Architectural Pattern

Our software uses a monolithic architecture. Our software is a singular, unified unit that is independent and works to achieve a specific goal. Our software is modularized into different components but under one unified application. First, this includes a script which executes the command prompt calls to download necessary packages and run the web crawlers. Second, it includes the scripts for each of the UNCC colleges and their departments web crawlers. Although there is modularization, all of these units function as a single unit and must all be present for the code to execute properly.



Rationale for Architecture

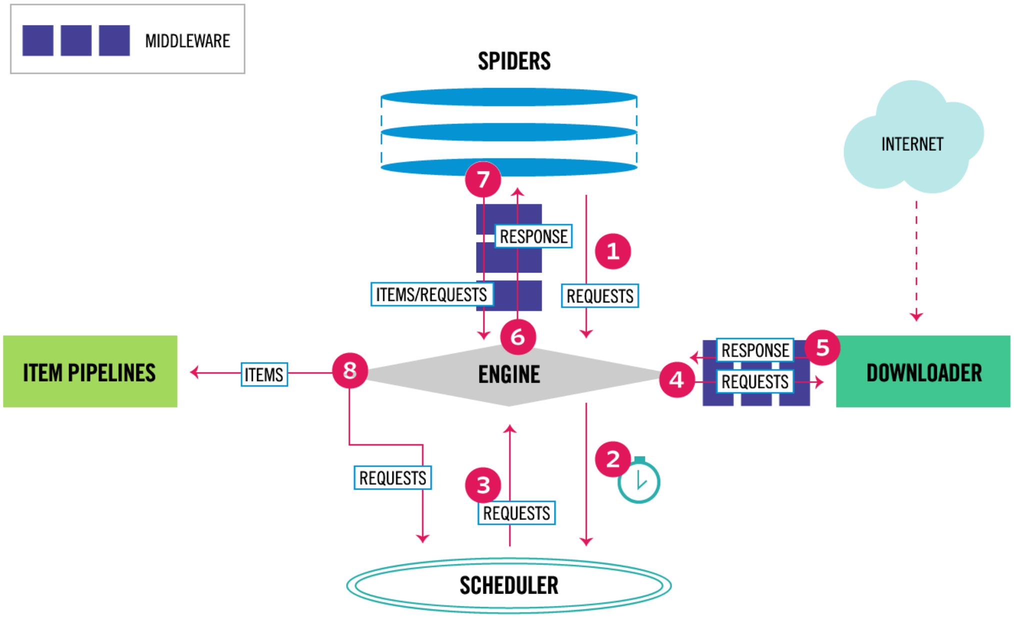
The rationale behind the choice of this architecture was that we needed a fast deployment of a software intended with a specific goal for the stakeholder. Additionally, our program is small and lightweight due to its specialized purpose. The state of the project and given our time constraint required a software that isn't intended with frequent changes and aims to serve a single purpose. Additionally, our goal was to provide good performance, and this comes from the simplicity of the code base. Finally, our goal was to make the deployment of the software easy as it's a single code base that our users can download.

Alternative Architecture Considerations

An alternative architecture that was considered was a microservice architecture. This architecture was not chosen because our software was not large enough to create a series of units of functionality. Our software has a single goal and required a software architecture that was focused with a goal of creating a single software product that achieved the stakeholders needs.

Subsystem Architecture

Due to the nature of a monolithic application, there is a limitation in the subsystem architecture. The sub architecture of our system would essentially be broken into two main components that are contained by the entire application. One component consists of the scripts that run the spiders and another is the executable script that runs the program. Additionally, our architecture used multiple frameworks to execute its functionality.



Scrapy framework data flow. *https://docs.scrapy.org/en/latest/topics/architecture.html*

Our main framework is the Scrapy framework. This resides in the applications spiders folders and is responsible for making requests to the department pages and downloading the data specified.

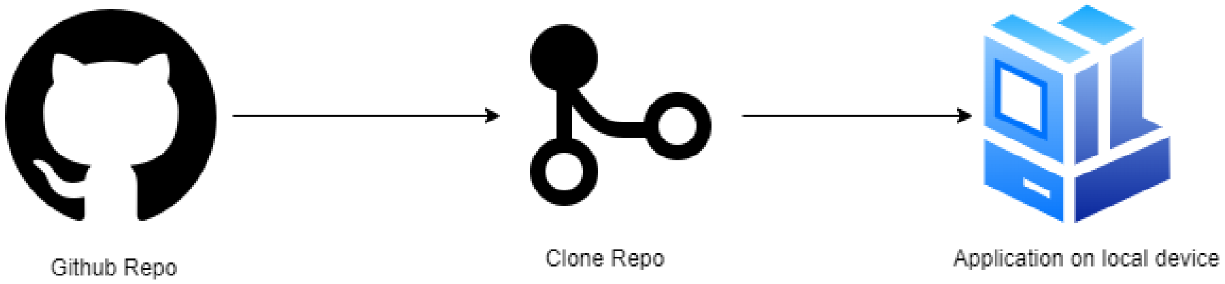
We also use the BeautifulSoup4 framework for beautifying scraped data.

Finally, we use the rake_nltk framework to process the scraped data and produce keyword lists.

Deployment Architecture

Our application is hosted on the users machine. The software is developed with the intention of running on Windows OS. Since our project is not a web based application, the user is required to directly download the software and its content onto their device. The software can be downloaded onto the user's device through two methods.

1. **Download from github repository:** they can download the software using our github repository which hosts the project.



2. **Receiving software packages directly:** they can receive the application directly by being sent a zipped folder of the application and its content.

Application sent
Download onto machine

.

Persistent Data Storage

The data will be stored locally and in real-time as the program is running. The data that will be stored are the faculty name, department name, faculty title, and keywords that are related to the faculty member. This will happen real-time as the web crawlers are scraping through the department pages and pulling the data. The data will then be stored into a CSV file, with a CSV file being created for each of the colleges at UNCC. The CSV files will be stored on the user's device inside a directory of the application.

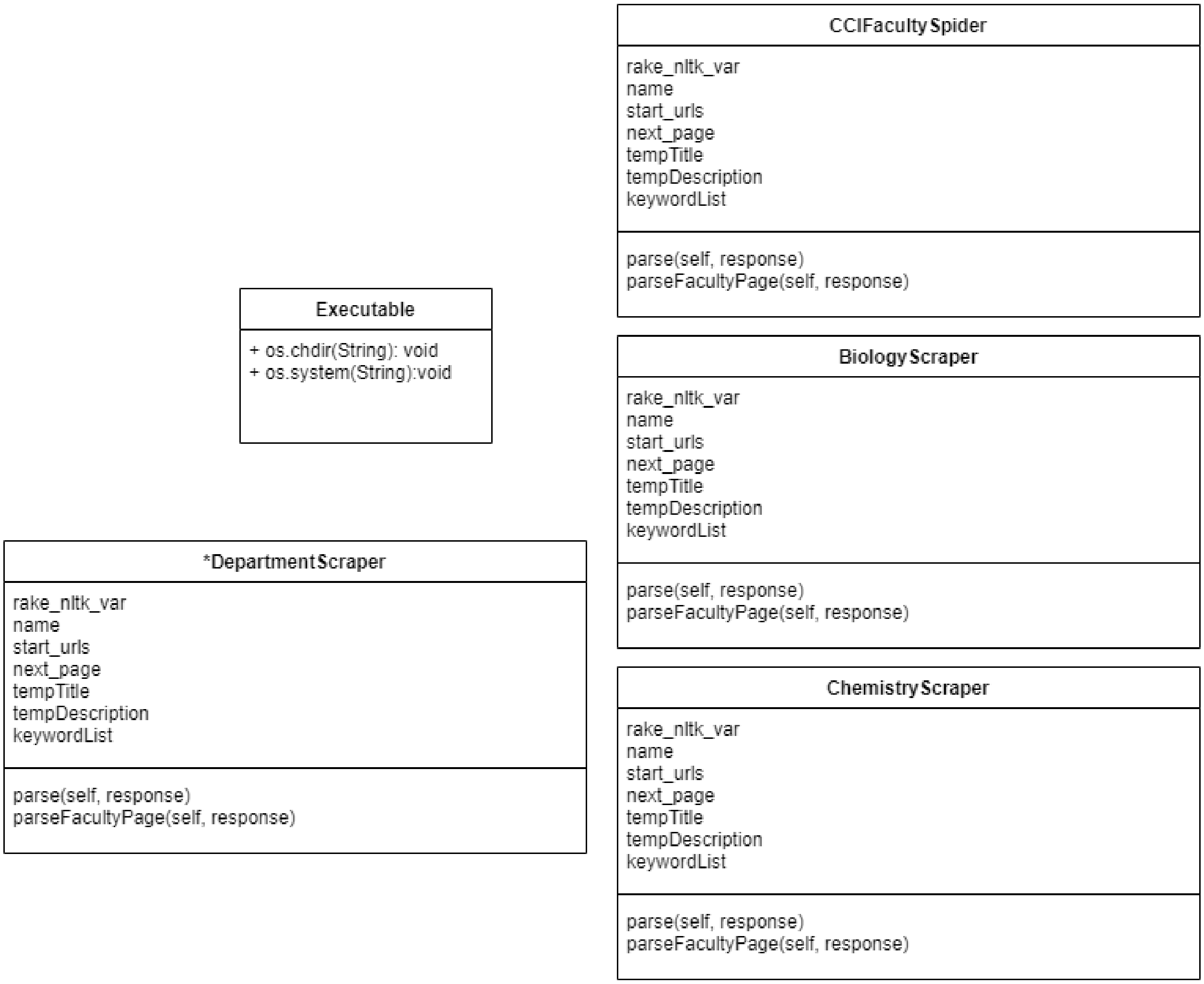
Global Control Flow

Shown in the earlier sections, our architectural pattern is monolithic. Due to the nature of the program and its specification, the entire project works as one system. This system is driven by a single executable file that makes calls to the projects scripts and downloads the systems required packages. The system follows a linear sequence of calls by moving through each script and executing the crawlers within those scripts. This design allows the system to be easily deployed to the user and simplifies the design of the project.

Our project doesn’t exhibit concurrency. Due to a single executable file being in charge of running the software, the linear nature of the program's execution does not allow for concurrency. Additionally, the frameworks used for the program do not support concurrent processing.

System Design

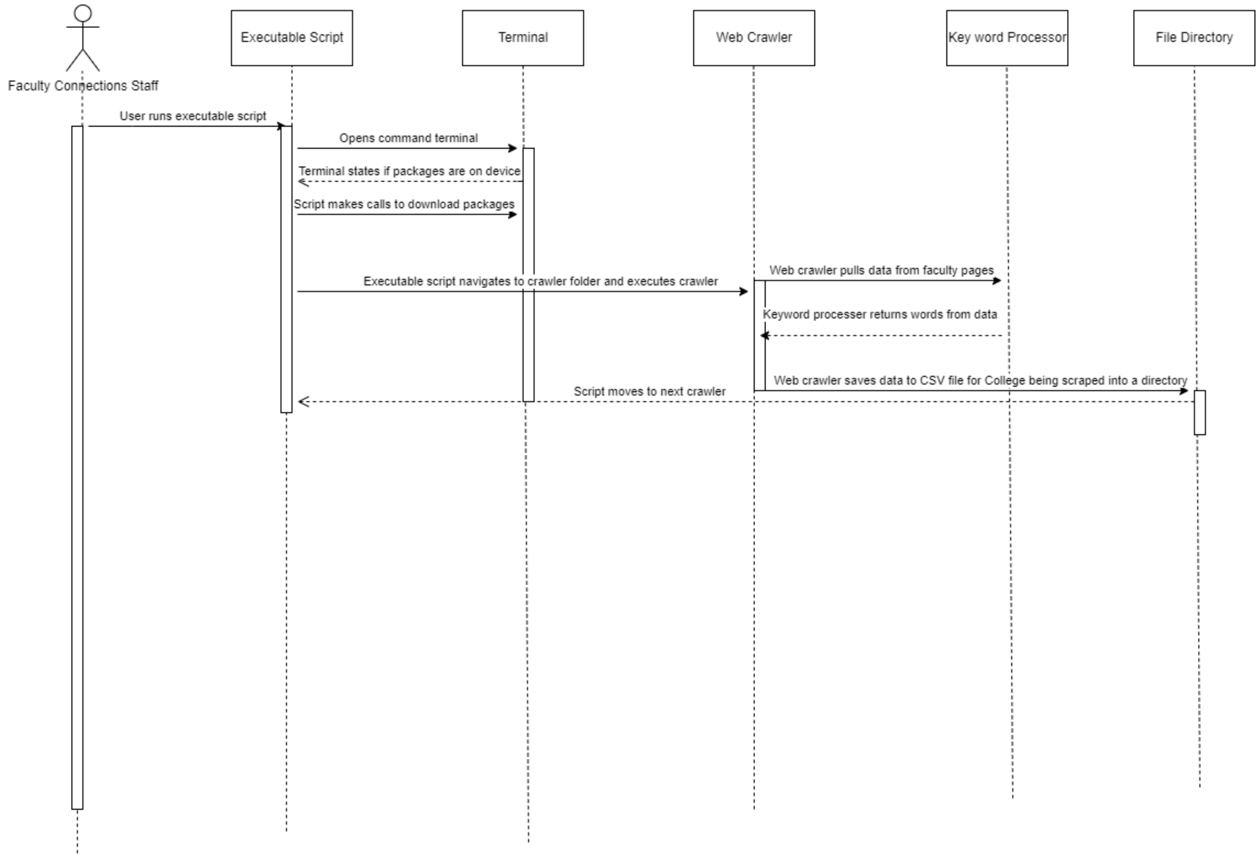
Static View



Essentially we have two unique classes. The first class is the executable class. Inside the executable class there is a python package that is imported to use command line calls. The two methods are defined and created within the python package. The other classes are our scraper classes. Each one of these classes are identical copies of each other. The only changes are small edits for the links that are used for each department's page and the CSS that the scraper uses to identify the sections to scrape the page. All of the classes aren't shown here, but there is a class for each department at UNCC.

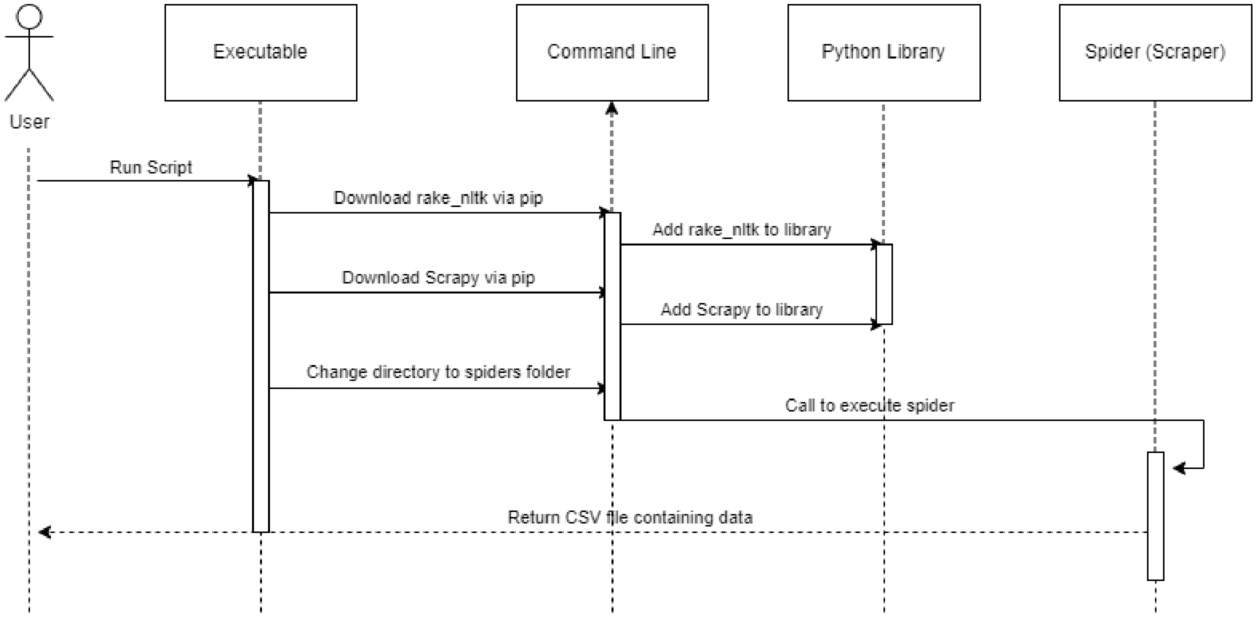
Dynamic View

Crawler Execution + CSV file output


Faculty Connections Staff
Executable Script
Terminal
Web Crawler
Key word Processor
File Directory
User runs executable script
Opens command terminal
Terminal states if packages are on device
Script makes calls to download packages
Executable script navigates to crawler folder and executes crawler
Web crawler pulls data from faculty pages
Keyword processer returns words from data
Web crawler saves data to CSV file for College being scraped into a directory
Script moves to next crawler

The execution of a crawler is a linear process that is executed identically for all of the colleges scripts. The user double clicks the executable file to run the program. The program then follows its steps for downloading any packages. Then the script navigates to our programs folder containing the crawlers and executes them one-by-one. The data scraped is then run through a keyword processor and saved to a CSV file. The executable script does this for each of the crawllers.

Executable Script Execution



The executable script is the controller of the system. Inside the executable script, the lines of code correspond with command line calls that download the required packages and execute each of the scripts containing code for each spider. When the script is run, the executable downloads the packages needed for the program. Then it proceeds to run each of the spiders and return CSV files containing the data.

System Design Rationale

The design rationale follows the architecture for a monolithic application. The backbone of our application is the main executable script that controls the flow of the application. The application is simplified to have scraper classes that are identical to one another except for the links for the department pages to be scraped and the CSS to identify sections of the pages to be scraped. This ensured that there was enough separation of concerns between the scrapers but also provided a frame of reference for creating each of the department's scrapers. This allowed our application to be linear in its sequence and made its execution easier.