Sri Lanka Institute of Information Technology

MOOCs Recommender Based on Learning Style

Project ID: 18-036

IT14140280

Hasith M. G. S

Software Requirements Specification - CDAP-I

MOOC Scraping, Crawler and Monitoring and Schedule the process

Bachelor of Science Special (Honors) in Information Technology Specializing in Software Engineering

Title : MOOCs Recommender Based on Learning Style - MOOC Scraping, Crawler and Monitoring and Schedule the process

Project ID : 18-036

PROJECT GROUP MEMBER DETAILS:

	STUDENT NAME	STUDENT NO.	CONTACT NO.	EMAIL ADDRESS
1	Saugat Aryal (GROUP LEADER)	IT14146602	+94766948707	Sam.arnav10@gmail.com
2	Porawagama A. S.	IT14142024	+94717039769	axewilledge123@gmail.com
3	Hasith M. G. S.	IT14140280	+94717484684	sa.hasith@gmail.com
4	Thoradeniya S. D.	IT14138232	+94768401185	Chloethora@gmail.com

……………………………………..
Mr. Nuwan Kodagoda
Supervisor

……………………………………..
Ms. Kushnara Suriyawansa
Co - Supervisor

Declaration

I hereby declare that the project work entitled “MOOCs Recommender Based on Learning Styles” (MOOC Scraping, Crawler and Monitoring and Schedule the process) submitted to the Sri Lanka Institute of Information Technology, is a record of original work done by our group under the guidance of Mr. Nuwan Kodagoda (Supervisor) and Ms. Kushnara Suriyawansa (Co- Supervisor), and this project work is submitted in the fulfillment for the award of the Bachelor of Science (Special Honors) in Information technology Specialization in Software Engineering. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma. The diagrams, research results and all other documented components were developed by myself and I have cited clearly any references I have made.

…………………………………………

Hasith M. G. S.

Table of Contents

1. Introduction

1.1 purpose

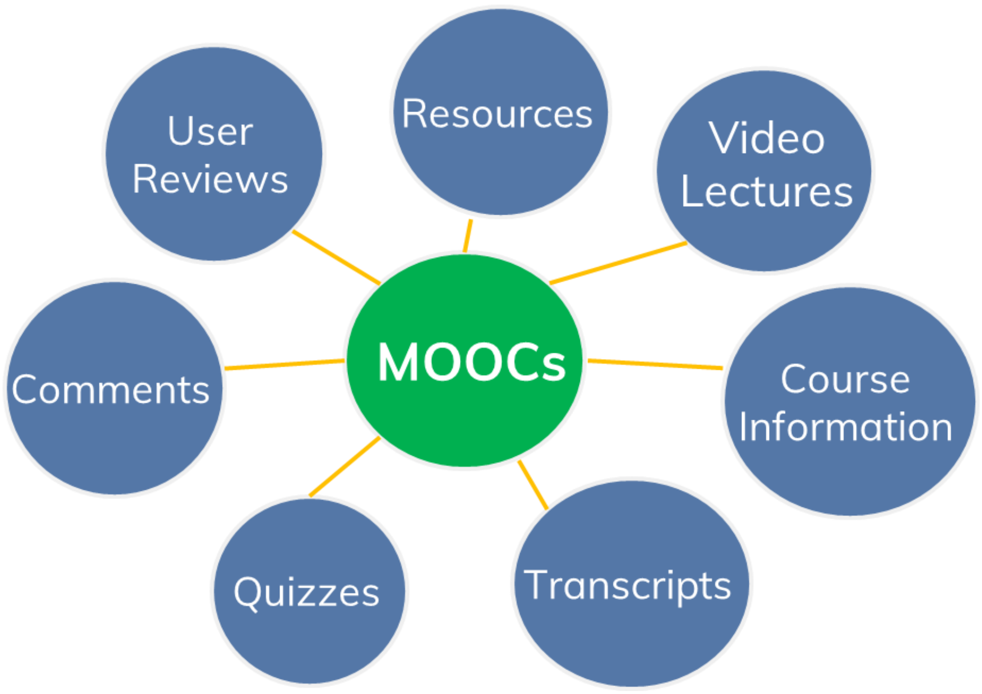
The purpose of this SRS document is to outline the requirements and present a detailed description of the process needed for MOOC Scraping, Crawler and Monitoring and Schedule**.** The document contains the necessary requirements of the system, as well as the process to create and discover them. It will explain the functional and non-functional requirements, purpose and features of the component, the interfaces of the component, design constraints, project approach, what the component will do, the constraints under which it must operate and how the component will interact with other external applications. The information is organized in such a way that the developers will not only understand the boundaries within which they need to work, but also what functionality needs to be developed and in what order. This document is intended to be proposed to a customer for approval and also this document is targeting the, designers, developers and other stakeholders as its audience.

1.2 Scope

Extracting information from three different MOOC platforms: edX, Coursera, Futurelearn and implementing a crawler to periodically look for new courses data. After that categorize that data (text, videos, pdf etc..) and download those data to our cloud storage. Depend on data type storage can be document database or hard drive. We implement an incremental web crawler on the basis of the crawler architecture named “Scrapy”. It crawls the information of news on the website successfully and implements incremental crawling. we need write different crawling rules for different MOOC platforms. A hybrid approach is in which employs the meta-search engines and a semantic-structure-based Web page analysis algorithm. The design goal of the focused crawler is to collect web pages probably related to a specific topic as many as possible and reduce irrelevant pages as few as possible.

An intelligent web crawler is presented in making the rule settings dynamic, at the same time, TF-IDF method is used to calculate the Web document correlation, and the automatic acquisition of data extraction rules is realized, which reduces the development cost and maintenance cost and improves the development efficiency of the crawler. Before download the course content we calculate the download complexity of the course (size of the videos and content). According to that complexity datamined what is the technic will be using to download the content.

After download the content call the internal system API and continue the system process. We will monitor the process and if some fail detect stop the process and send the report to responsible person via email automatically. Also provide admin dashboard to view and maintain these process. We delete all data after all process happened.



1.3 Definitions, Acronyms, and Abbreviations

MOOCs	Massive Open Online Courses
CNN	Convolutional Neural Network
OS	Operating System
RAM	Random Access Memory
SRS	Software Requirement Specification
GPU	Graphics Processing Unit

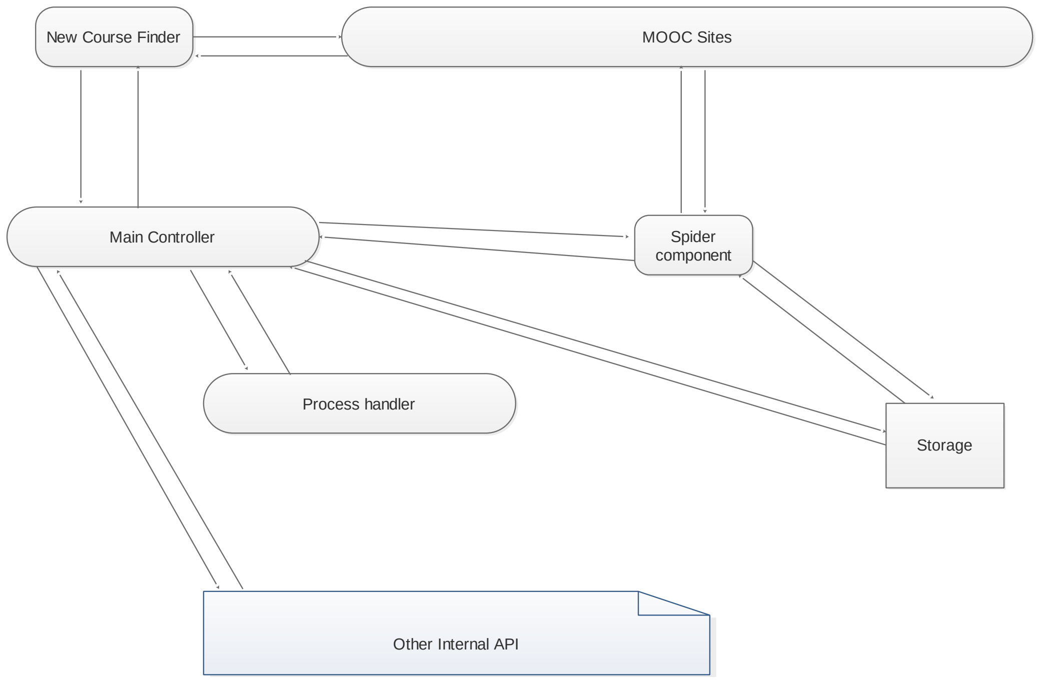
1.4 overview

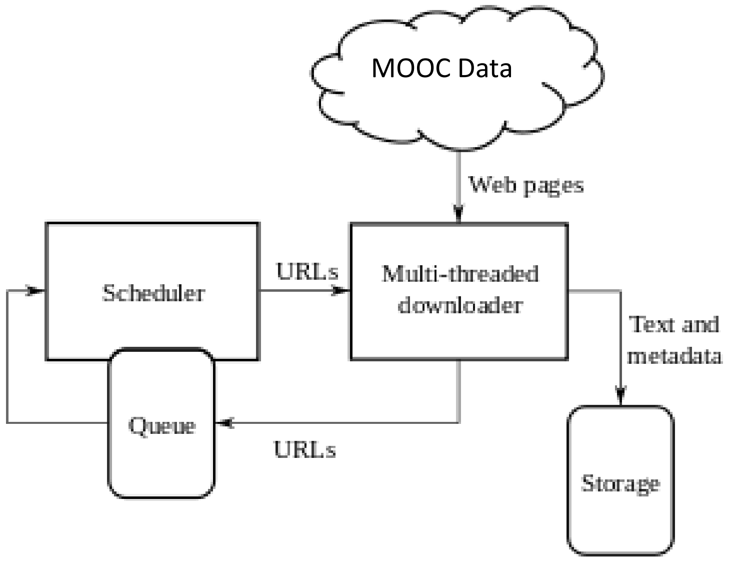
The remainder of this SRS document includes three sections and appendixes. The second section provides an overall view of the component functionality and interaction with other components. This section also discusses the specific requirements such as functional and nonfunctional requirements, design constraints and various approaches. Furthermore, this section also mentions the system constraints, User characteristics and assumptions about the product.
The third section provides the requirements specification in detail and a description of the different interfaces. Different specification techniques are used in order to specify the requirements more clearly for different audiences.
The rest of the sections that organized this document are Project perspective and descriptions, Different interfaces that the system consists, Requirements of the system, Summary of major functionality, Users and characteristics of the system and the background of the general factors affect the system.

2. Overall Descriptions

- MOOC Scraping and Crawler

  Web scraping is a software program that extracts information form websites. These programs are able to simulate human web surfing behavior by implementing either low level Hypertext Transfer Protocol (HTTP) or embedding a web browser. Web scraping concentrates on transforming unstructured data found online, into structured data that can then be stored and analyzed in a database or internal storage. Our web crawler execute five different functions. First, we undergo the task of web interaction. This will happen when we detect a new course in our one of MOOC sites. Secondly, programs known as wrappers need support for generation and execution. This point we collect overall information about download content and size. Scheduling is the third function that allows wrappers to be applied repeatedly to their respective target pages. Next, data transformation occurs which involves the filtering, mapping, refining, and integrating of data from one or more sources. The result is then structured to create the desired output format. Finally, we download the content from course. According to site (course) code base, web scrapper changes his scrapping methods. Be course of that most of the failures and speed issue will be avoid. We are going to use some deep learning algorithms for this task. We collated the common class names and html tags that are used on job posting pages on millions of sites to feed this Auto Extract engine. Then used this as training data to ensure that the engine was capable of handling any new site being added to it with minimal manual intervention. With time the engine is becoming more and more stable to extract clean job data


New Course Finder
MOOC Sites
Main Controller
Spider component
Process handler
Storage
Other Internal API


MOOC Data
Web pages
Scheduler
URLs
Multi-threaded downloader
Queue
URLs
Text and metadata
Storage

- Monitoring and Schedule the process

  All process can be monitor in admin view and this all process can automate or manually can through the system. If some failure detects, system automatically stop the task and informed to the responsible person.

2.1 Product perspective

There are several applications available around the world for scrapping and crawling and all the search engines are using same technic for gathering information from the web sites.

Mozenda [1]

Mozenda helps companies collect and organize web data in the most efficient and cost effective way possible. Its cloud-based architecture enables rapid deployment, ease of use, and scalability. If a company needs to collect data from the web, Mozenda is the best way to do it. It is quick to implement and can be deployed at the business unit level in minutes without IT involvement. A simple point and click interface helps users build projects and export results quickly—on demand or on a schedule. It is easy to integrate, users can publish results in CSV, TSV, XML or JSON format.

Automation Anywhere **[2]**

Automation Anywhere Enterprise comprises of a group of experts focused on providing a complete end-to-end cognitive and flexible Robotic Process Automation tools to easily build bots to digital functioning bots, powerful enough to automate tasks of any complexity, but at the same time is user-friendly. Automation Anywhere Enterprise is the only RPA platform designed for the modern enterprise that is capable of creating software robots to automate any process end-to-end. Advance with cognitive bots with learning ability for semi-structured processes that need expert decision-making and transforming analytics that will promote operations. Automation Anywhere Enterprise offers three types of bots, each bot working.

WebHarvey [3]

WebHarvey is a visual scraper which automatically scrapes texts, URLs, and images from websites and saves the extracted data in different formats. It scrapes data from websites within minutes, and it is easy to use because it contains a built in scheduler and proxy support which allows it to scrape anonymously hence avoiding blocking from servers. The inbuilt browser allows the user to scrape data without codes hence access and scrape data from multiple pages. The scraper allows for categorical scraping allowing the user to access links which lead to listings of the same data within a website.

2.1.1 System interfaces

- Video classification interface

  Call this interface to classify the course video according to special technic

- Topic Modeling interface

  This interface use for calculating the text and content complexity

2.1.2 User interfaces

- Admin user dashboard

  This use for monitoring and controlling the all system process. This contain all logs details. This will really helpful for maintains and controlling the system automatically and manually. This UI can access only several responsible persons.

2.1.3 Hardware interfaces

No special hardware devices are needed other than the usual PC or Laptop.

2.1.4 Software Interfaces

- Google Colaboratory
- Jupyter Notebook
- Google Tensorflow
- Keras – Deep Learning Library
- AWS EC2 instant
- Visual Studio Code
- Python 2.6 +

2.1.5 Communication interfaces

The user’s laptop/desktop should support high speed data communication methods such as 3G (HSPDA), fiber optic connection or 4G (LTE).

2.1.6 Memory constraints

Not applicable this all functionalities are backend process.

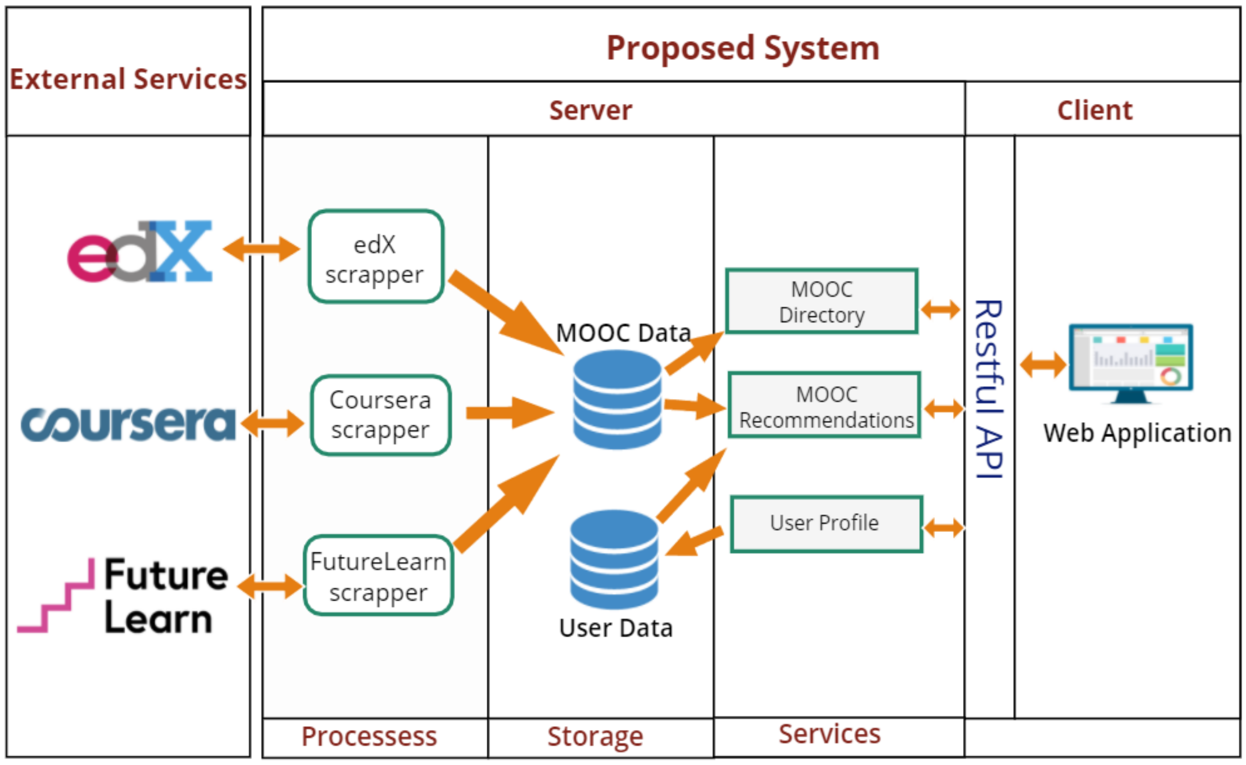
2.1.7 Operations

Functionalities of the Scrapping and Crawling Component are carried out in the backend and hence, no operations are required by the user.

2.1.8 Site adaptation requirements

Since the user can be of any nationality the user interface must be created for English language.

2.2 product functions



2.2.1 Use Case Scenarios

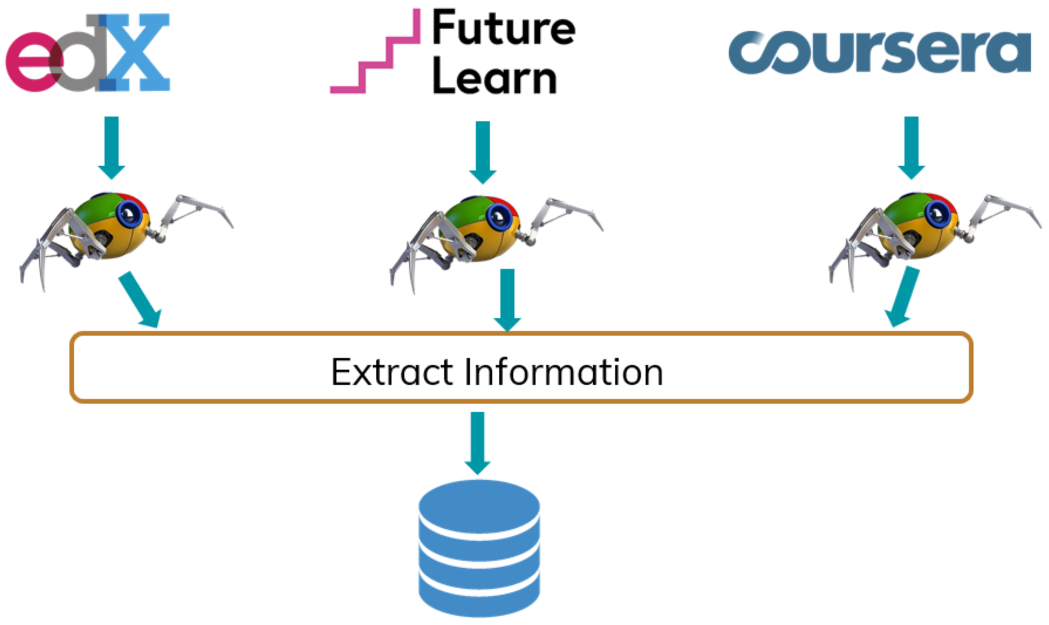
Use case Name	Find new courses
Pre –Condition	System working properly
Post-Condition	New course URL list
Actor	Backend System
Main Success Scenarios	1. Find new course with compeering already analyzed course 2. Detect newly available course 3. Put new course data into our database 4. use case end with send new courses data to main controller.

Extension	1a. New course selected.

Use case Name	Run Spider into Course
Pre –Condition	System working properly Selected a new course
Post-Condition	Store the all new course data in cloud storage
Actor	Backend System
Main Success Scenarios	1. Scrapper run in new course 2. indexing all URL and details 3. Downloading the all content in the course 4. use case end with send success message to main controller.
Extension	1a. Skip some course content.

Use case Name	Run System API for new course
Pre –Condition	System working properly Downloaded all content of the course
Post-Condition	Completed all process in system related to one course
Actor	Backend System
Main Success	1. Call internal API 2. Map course data with API

Scenarios	3. Monitoring the process 4. Delete all the content related to course after completing the process.
Extension	1a. Skip some course content.



2.3 User characteristics

The web scrapping and scrolling component deals with the backend process and involves developing a python “Scrapy” framework and deep learning neural network for the data collection and download parts. For the optimization we will be use simple machine learning technic. Process monitoring and scheduling we use cloud logs and monitoring tools. Developers should have knowledge about those area for the develop the system.

2.4 Constraints

The backend process of Scrapping, web scrawling and schedule and monitoring process involves heavy computational tasks. The constraints of the process are mentioned below:

- System should be run in AWS EC2 instant with cloud storage.
- Backend system must have minimum of 8GB RAM. Additional 1GB + GPU.
- Linux base Operating System.
- Python is the implementation language.
- Document database instant (like MongoDB).
- We shouldn`t keep any courses data after our categorization.
- Cloud base email services.

2. 5 Assumptions and Dependencies

- The operating system for the component is selected to be Linux.
- The internet connectivity can be established with ease on request.

2. 6 Apportioning of Requirements

Essential Requirements:

1. Identify the new courses and stored inside database (course details).
2. Indexing the URL.
3. Categorize the URL page content and download.
4. Delete all content after completed the process.

Desirable Requirement:

1. Data categorize and add index to the data for easily access and read it.

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Web Scrapping and Crawler is a backend process for that we don’t have any interface. We implement very simple dashboard for use who maintain the system after deployed. That contain logs and monitoring details about system processes. Admin user can also change the default process schedule and customize the process through the interface.

3.1.2 Hardware interfaces

Server-Side Hardware Interfaces

- For the server side there should be a server space with minimum 8+ GB Ram and high capacity CPU and 2+ GB GPU. It should be AWS EC2 instant and it should have cloud base storage which can be automatically scale. Also as the database instant must installed in the server.

Hardware Interfaces for the Development team

- Development team must have a computer that has 3.0 GHz processing power in order to do this project. Minimum 8 GB RAM and 2 GB GPU space and 5400 RPM hard disk drive also can consider as hardware requirements.

- For the testing purposes development team requires a cloud base environment with AWS EC2 instant include MongoDB.

3.1.3 Software interfaces

- Google Colaboratory: It is free cloud service offered by Google that provides free GPU. That distinctive feature makes it different than other cloud services. Using this service, users can develop deep learning applications using popular libraries such as, Keras, TensorFlow, OpenCV etc.

- TensorFlow: It is a computational framework for building machine learning models. It provides wide variety of toolkits that allows the users to construct models at the preferred level of abstraction. It comes in-built with Google Colab.

- Keras: It is a very popular deep learning library written in Python. We can quickly build and test a neural network with minimal lines of code. It is capable of running on top of TensorFlow and also available in Google Colab.

- Python 2.6 + supported and install in to cloud platform.

- MongoDB instant for storage data

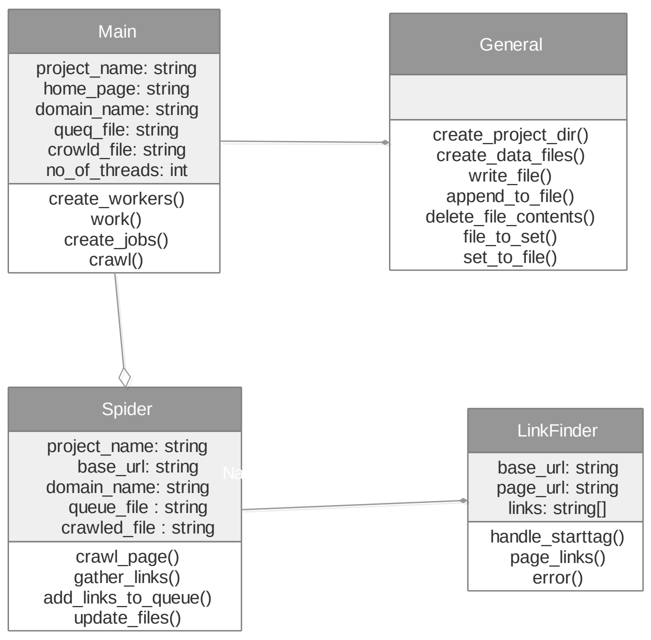
- Elastic Storage management software.

3.1.4 Communication interfaces

The backend service along with the database is hosted on the cloud. Hence, the communication between them requires internet connectivity. The connection can range from minimum of 3G to fast 4G (LTE).

3.2 Classes/Objects

- Spider class diagram



3.3 Performance requirements

It is expected that the scrapping, web scrawling and process monitoring, handling component will perform all the requirements stated under the product functions section. Some performance requirements identified are listed below:

- Time taken to download a one course must be 24 hours.
- Only one time can download and store a limited resource (around 80 GB). It is depending on available space.
- After process complete data must be deleted.
- One course not taken a more than one time. We should keep checked courses list.

3.4 Design constraints

- MOOC data only from three platforms: edX, Futurelearn and Coursera are considered.
- Courses should focus on one domain (programming, computer science)

3.5 Software system attributes

In this section, we are explaining the attribute that we are going to offer through the system. There are some explanations in follow.

3.5.1 Reliability

Reliability of a system is the ability to perform its normal operations with minimum failures over a specified time in a given environment.

- The reliability of the web scraping and web scrolling depend on minimum failure. We user dynamic web scroller and if one failure detect, we stop the task and revive about that failure. Anyway this process is a backend related then it won’t show any failures to the users.
- Process handling and monitoring can use for reliability handling in entire system. It will provide overall summary about the system process. We can use that information for make system more reliable.

3.5.2 Availability

Availability of a system is the possibility that a system will work as required during the point of time and it should be able to deliver the requested service. That is a service should be available with minimal system down time or without having any system failures for longtime.

- This all task is related to internal processes and this process not effect to availability.

3.5.3 Security

Security of a system is an attribute which reveal ability to resist unauthorized usage while still providing its services to legitimate users and it can protect itself from external assaults. In this component any authorize user should be able to use the internal process or access

- We will be use AWS Cognito Authenticate service for authentication and authorization in admin dashboard
- The system must use HTTPS protocol. It will give more secure data transaction by considering to other protocols.
- data should be stored in database using an encryption method
- Maintains strong server-side controls

3.5.4 Maintainability

Maintainability of a system means handling the system with new requirements in order to enhance the performance and capabilities of the application and making sure that new errors shall not be prone in the system because of the changes. That means the proposed system can be maintained easily if there is some modification without happening any damage or interrupt to other system functionalities. As well as modifications can be done through the low-cost solutions. It is also a somewhat important feature to having high maintainable system

In case of a failure, a re-initialization of the program is recommended.

- Our system does not have a particular client or a customer. Therefore, the maintenance requirements are handled by the development team by considering the future potential use of the application.

- The application is designed in a way that it assists for updates of the software in future. The code will comment wherever it is necessary, especially in critical and complex code segments. This will help the developers or maintaining team for modifications in future.

4. References

[1] Banerjee, Ritu. "Website Scraping." Happiest Minds. N.p., Apr. 2014. Web. 11 Apr. 2016.

[2] Chen, Hsinchun, Roger H. L. Chiang, and Veda C. Storey. "Business Intelligence and Analytics: From Big Data to Big Impact." MIS Quarterly 36.4 (2012): 1165-188. Web. 8 Apr. 2016.

[3] Data Toolbar. Computer software. Data Toolbar. Vers. 3.1. DataTool Services Inc, 2013. Web. 8 Apr. 2016.

[4] Diebold, Francis X., On the Origin(s) and Development of the Term 'Big Data' (September 21, 2012). PIER Working Paper No. 12-037. Web. 13 Apr. 2016.

[5] Huynh, David, Stefano Mazzocchi, and David Karger. "Piggy Bank: Experience the Semantic Web Inside Your Web Browser." The Semantic Web – ISWC 2005 Lecture Notes in Computer Science (2005): 413-30. Web. 13 Apr. 2016.

[6] Laender, Alberto H. F., Berthier A. Ribeiro-Neto, Altigran S. Da Silva, and Juliana S. Teixeira. "A Brief Survey of Web Data Extraction Tools." ACM SIGMOD Record SIGMOD Rec. 31.2 (2002): 84. Web. 11 Apr. 2016.

[7] McAfee, Andrew, and Erik Brynjolfsson. "Big Data: The Management Revolution." Harvard Business Review. Hank Boye, 01 Oct. 2012. Web. 08 Apr. 2016.

[8] Vargiu, Eloisa, and Mirko Urru. "Exploiting Web Scraping in a Collaborative Filtering- Based Approach to Web Advertising." Artificial Intelligence Research AIR 2.1 (2012): 44-54. Web. 13 Apr. 2016.