MOOCS RECOMMENDER BASED ON LEARNING STYLES

Software Requirement Specification
Project ID: 19-089

Liyanage A.Y.K. | IT 16 0327 98

De Silva W.A.T.P. | IT 16 0519 80

Pathirana P.H.P.S.L. | IT 16 0043 82

Hilmy S.B.M. | IT 16 0053 72

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

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

Date of Submission: 2019-05-13

MOOCS RECOMMENDER BASED ON LEARNING STYLES

Project ID: 19-089

Liyanage A.Y.K. | IT 16 0327 98

De Silva W.A.T.P. | IT 16 0519 80

Pathirana P.H.P.S.L. | IT 16 0043 82

Hilmy S.B.M. | IT 16 0053 72

Supervisor: Mr. Nuwan Kodagoda

Co-Supervisor: Ms. Kushnara Suriyawansa

Date of Submission: 2019-05-13

Declaration

We hereby declare that the project work entitled "MOOCs Recommender Based on Learning Styles" 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 us and we have cited clearly any references we have made.

Name	ID	Signature
Liyanage A.Y.K.	IT16032798	
De Silva W.A.T.P.	IT16051980	
Pathirana P.H.P.S.L.	IT16004382	
S.B.M. Hilmy	IT16005372	

Supervisor: Mr. Nuwan Kodagoda Signature:

Co-supervisor: Ms. Kushnara Suriyawansa Signature:

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This document presents a comprehensive set of all the major requirements associated with the 4th year research project titled "MoocRec V2" which is aimed to provide better MOOC recommendations based on user preference and engagement. This document is intended to be referred and understood by research supervisors, research team and the CDAP panel. The document also contains details such as constraints that the system should adhere to both design wise as well as implementation wise. It is also worth noting that a higher focus is given to software related aspects of the solution. Even so, hardware related details are mentioned where necessary to give a broad idea about the system.

1.2 Scope

All the major software related requirements of "MoocRec V2" application is covered in this document. The research on which "MoocRec V2" is based is about how e-learners choose which MOOCs to follow or to drop out of, based on the video style they are produced in. As such, "MoocRec V2" can be considered as a MOOC recommender, albeit with a different approach to search criteria. By understanding the user's preferred video production style, more relevant MOOCs will be recommended. At the same time, user is free to search MOOCs across 3 major MOOC providers through the system.

1.3 Definitions, Acronyms and Abbreviation

MOOC	Massive Open Online Course
SRS	Software Requirements Specification
NLP	Natural Language Processing
ML	Machine Learning
HCI	Human Computer Interaction
Containerization	Packaging applications to run independently on any host OS, similar to virtual machines but faster and smoother
Docker	A popular framework for containerizing
ILS	Index of Learning Style
DOM	Domain Object Model
CNN	Convolutional Neural Network

1.4 Overview

"MoocRec V2" is intended to be easily used by anyone interested in consuming online courses and MOOCs. Therefore, this document covers all the major functional and non-functional requirements essential to achieve the said easy to use platform. The document is structured as a 3-chapter document.

The first chapter provides an outlook about the document itself and what information it contains and what audience the document is intended for.

The second chapter will provide a comprehensive outlook of all the functionalities from the point of the end-user. Details such as user-interfaces and backend processes needed to support the said user-interfaces and their functionalities. Furthermore, it highlights the characteristics of the expected end-user and the constraints that should be adhered to, in order to ensure proper use of the system.

While the second chapter provide an outlook inclined more towards the end-user, the third chapter provides an in-depth view, primarily aimed at the developers. Constraints relevant to design and implementation are explained in detail. Similarly, non-functional requirements are described along with the reasons for their importance. Other technical details such as database structures and compliances are also mentioned in the third chapter.

To summarize, both the second and third chapters provide two different perspectives of the same system.

2 Overall Descriptions

2.1 Product Perspective

Massive Open Online Courses also known as MOOCs for short, has become a popular medium for learners of different fields to learn new things but the completion rate of MOOCs happens to be low. It is identified that one of the main reasons behind this is the mismatch between how a person learn and how the MOOC is delivered in terms of the video style. Several learning models such as Felder-Silverman Learning Model and Kolb Learning Model indicate a set of different learning styles that a person might have [1]. However, none of the existing MOOC recommenders take users' preferences and engagement to account when recommending MOOCs. Therefore, this solution aims to accurately identify a user's preferred video production style based on active user inputs as well as passive user interactions during a one-time-only interactive video session. Furthermore, this solution will analyze major online discussion spaces associated with MOOCs to provide better, more relevant and updated recommendations.
Below is a brief outlook on existing products.
There are many tools in the internet that let the user search MOOCs based on a range of categories and filters and some even recommend more MOOCs based on the history of a user. Apart from these similar tools, MOOCRec V1 is the only tool that takes a person's learning style into account when recommending a MOOC.

- MOOCRec (referred to as MOOCRec V1 for differentiating our proposed solution, MOOCRec v2)

Figure 2.1: MOOCRec Logo

A MOOC recommender that identifies the learning style of a user through a lengthy Questionnaire. It indexes MOOCs from Coursera, Edx and Futurelearn. MOOCRec also has user profiling.

- Class Central

Figure 2.2: Class Central Logo

Class Central is one of the most popular MOOC search engines that also makes recommendations based on the fields of studies and institutions that a user would prefer. But Class Central does not recommend MOOCs based on the individual learning style of a person.

- My MOOC

Figure 2.3: My MOOC Logo

Another popular MOOC search engine that focus facilitating users to rate MOOCs and review them to help others find a MOOC that suites them. But it is up to the user to go through these reviews and find out they would like to follow.

2.1.1 System Interfaces

- Docker Daemon
- Docker Network
- Host Network
- Disk Mount

2.1.2 User Interfaces

- MOOCRec V2 introductory video launch page.

Figure 2.4: Interactive Video Session

- MOOCRec V2 video content rating page.

Figure 2.5: MOOCRec V2 MOOCs recommender page

- MOOCRec V2 MOOCs recommender page.

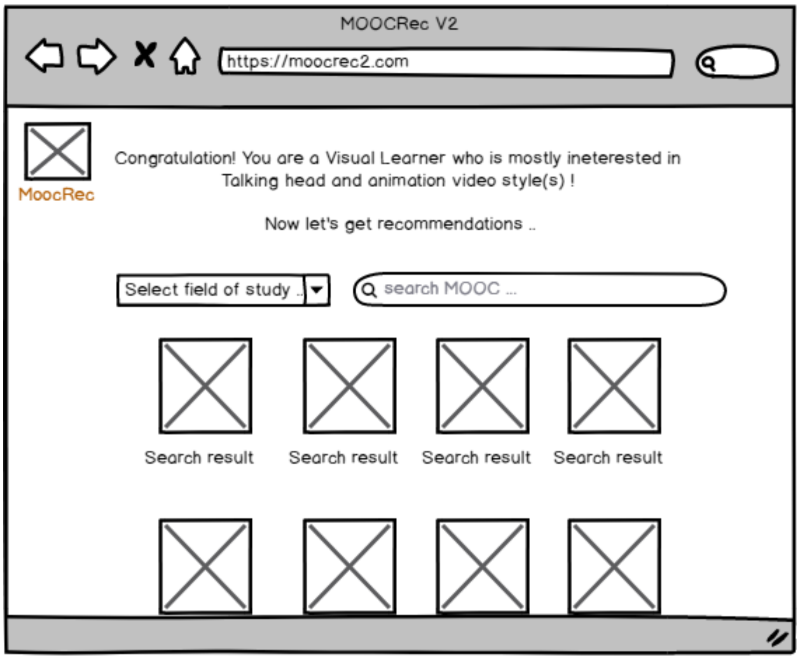

Figure 2.6: Selecting the Field of Study for MOOCs

2.1.3 Hardware Interfaces

- Linux Based Server
  A Linux based (preferably CentOS 7) server is key for running Docker. Since this component is fully a backend process, a CLI based Linux server can give maximum head room for processing.

2.1.4 Software Interfaces

- Python 3.7
- Docker
- RabbitMQ
- Browser DOM
- Keras
- Google TesorFlow

2.1.5 Communication Interfaces

- A 100Mbit or higher network interface to download video files fast enough such that it will not end up being a bottleneck.

2.1.6 Memory Constraints

- 8GB per server at minimum to efficiently run multiple instances of a Docker image.
- $GB or higher RAM for client side.

2.1.7 Operations

In order for a MOOC user to identify their learning style and the learning preference, the following operations should be done;

- User should have an active internet connection.
- User can follow the interactive video session after logging in, to identify the preferred video production style and the learning dimension.
- User can select the field of study in order to narrow down MOOCs relevant to that field.

2.1.8 Site Adaptation Requirements

- The site must be in English to cater for a wide range of audience.
- Interactive video session must contain English audio along with an English transcript.

2.2 Product Functions

- Record user activity and interactions.
  During the interactive video session, both active and passive user interactions are recorded. Interactive visual elements will be used to facilitate active user interactions highlighted below.
    - Skimming through the video.
    - Skipping sections of the video.
    - Rating sections of the video
    - Showing/hiding visual elements such as the video player itself, the transcript.

Furthermore, the passive user interactions such as mouse movement, mouse clicks and keyboard activity will be recorded. The flow of activity will be mapped to the length of the interactive video session in order to identify the passive activity of a user at a given point of time.

- Analyze user engagement.
  Both active and passive interactions mentioned above will be used as inputs to analyze the user engagement during the interactive video session. Passive interactions are represented as a set of heat maps. This will be used to find the sections of the interactive video clip that had highest activity as well as areas of the screen that had highest activity.
  Furthermore, the active interactions of the user will be processed through a model that is defined with all the possible interactions to determine the engagement level of the user at each point in time. Together, the two outputs given based on the two inputs will be used to draw a conclusion as to which section of the interactive video session that user showed most engagement.

- Map learner style and make MOOC recommendations
  Based on the video style that the user showed most engagement, a learning dimension from Felder-Silverman model will be mapped to the user. The said video style will be used as the key search filter when recommending MOOCs that are indexed in the database. When making recommendations, the user's preferred video style, learning dimension will be used. Also, the forum activity associated with the recommended MOOCs will be considered when finalizing the said recommendations. The forum activity data will be provided by another component

2.2.1 Use Case Diagram

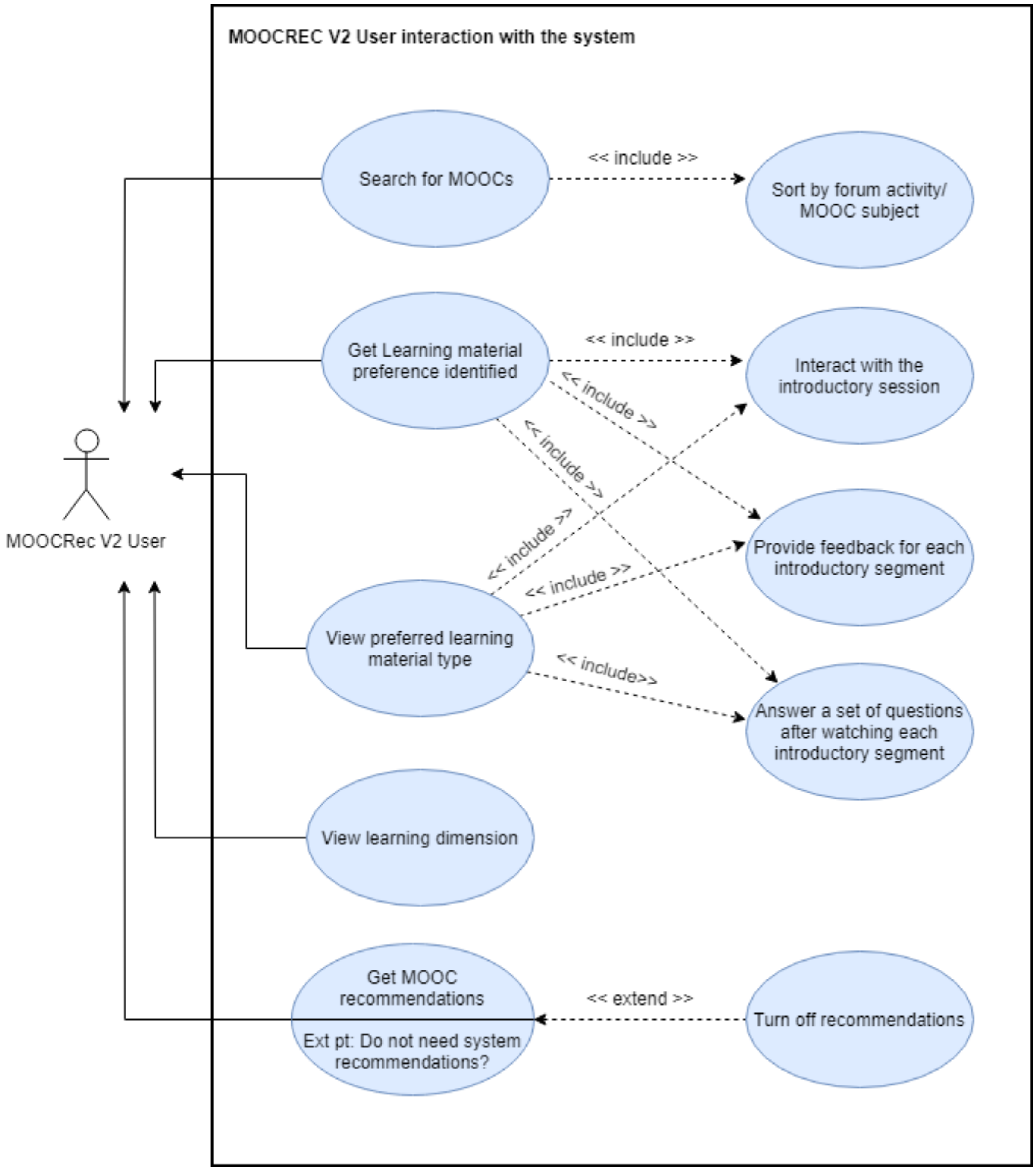

Figure 2.7: MOOCREC V2 User Interaction with the System Use Case Diagram

2.2.2 Use Case Scenarios

Table 2.1: Use case Scenario 1

User Case Name	View preferred learning material type
Pre-Condition	User should be logged in
Post-Condition	User should have completed watching the interactive session
Actor	MOOCRec V2 User
Main Success Scenario(s)	1. Watch the interactive introductory video provided by MOOCRec V2 2. Actively participate throughout the video 3. Rate each video segment watched. 4. Answer the given questions after each video segment. 5. Proceed till the end of the interactive session.

Table 2.2: Use case Scenario 2

User Case Name	Re-watch the interactive video to change preferences when needed
Pre-Condition	User should have already taken part in the interactive session or have skipped the session during initial sign in.
Post-Condition	No post condition
Actor	MOOCRec V2 User
Main Success Scenario(s)	1. User navigates to account settings page. 2. User chooses 'change my preference' link 3. User participates to the interactive session.

Table 2.3: Use case Scenario 3

User Case Name	Interact with the video player
Pre-Condition	User should be logged in.
Post-Condition	No post conditions.
Actor	MOOCRec V2 User
Main Success Scenario(s)	1. User watches the interactive introductory video provided by MOOCRec V2 2. User performs view transcript/ skim through video/ skip video segments/ pause video/ mute sounds/ maximize the video player/ watch subtitles operations.

Table 2.4: Use case Scenario 4

User Case Name	View learning dimensions
Pre-Condition	User should be logged in.
Post-Condition	User should have completed watching the interactive session and actively participated throughout the session.
Actor	MOOCRec V2 User
Main Success Scenario(s)	1. User watches the interactive introductory video provided by MOOCRec V2 2. User rates each video segment watched. 3. User answers the given questions after each video segment. 4. User proceeds till the end of the interactive session.

Table 2.5: Use case Scenario 5

User Case Name	Logically mark video file into chunks
Pre-Condition	A MOOC course with compatible video files has been downloaded
Post-Condition	Message queue contains messages where each message represents a logical video chunk.
Actor	Orchestrator (Backend System)
Main Success Scenario(s)	6. Analyze the video's length and select a suitable video chunk duration. 7. Create messages that contains the MOOC name, video file name, start and end times that match the chunk duration chosen in previous so that all the chunks cover the whole video. 8. Push the messages to message queue.

Table 2.6: Use case Scenario 6

User Case Name	Classify a logical video chunk
Pre-Condition	A message queue crowded with messages that represent logical video chunks.
Post-Condition	A message that contains the prediction for the logical video chunk should be sent to the message queue.
Actor	Containerized Classifier (Backend System)
Main Success Scenario(s)	1. Retrieve a message that represent a logical video chunk from message queue. 2. Extract the part of video that the start and end time indicates from the actual video file in the shared storage device. 3. Run the extracted video through the classifier. 4. Push the prediction of the video style of the extracted video to the message queue.

Table 2.7: Use case Scenario 7

User Case Name	Provide final prediction for a video file
Pre-Condition	A logical video chunks that makes up the video file have been classified and the prediction data is available in the message queue.
Post-Condition	Prediction for the video file must be stored in the database.
Actor	Analyzer (Backend System)
Main Success Scenario(s)	1. Retrieve messages that represent the predictions given for all logical video chunks of an individual video file. 2. Calculate the percentage of unique predictions. 3. Store the unique prediction with highest percentage in the database.

Table 2.8: Use case Scenario 8

Use Case Name	Gather forum data regarding a MOOC
Pre-Condition	Forum data about the MOOC • does not currently exist in the database • which currently exists in the database is outdated
Post-Condition	Data about the MOOC • should exist within the database • should not be outdated
Actor	Web Crawler - Backend System
Main Success Scenarios	1. Visit the MOOC platform 2. Login using the user credentials 3. Access the specific MOOC 4. Go to the forums section 5. Traverse the forum threads and store the require data in the database
Extension	1a. Connection issues 3a. The MOOC no longer exists 5a. The connection with database has been severed

Table 2.9: Use case Scenario 9

Use Case Name	Calculate Forum Activity Score using Metadata Analysis
Pre-Condition	Forum data about the MOOC • should exist within the database • should not be outdated Metadata sub score not should be present
Post-Condition	Metadata sub score should be present
Actor	Forum Processor - Backend System
Main Success Scenarios	1. Retrieve the attribute data from the database 2. Analyze forums using the attributes available 3. Save information in database
Extension	1a. Data does not exist in the database 3a. Connection with database has been severed

Table 2.10: Use case Scenario 10

Use Case Name	Calculate Forum Rating using Sentiment Analysis and Existing Ratings
Pre-Condition	Data about the MOOC • should exist within the database • should not be outdated Sentiment sub score not should be present
Post-Condition	Sentiment sub score should be present
Actor	Forum Processor - Backend System
Main Success Scenarios	1. Retrieve forum thread posts from database 2. Process posts to get sentiment sub score 3. Save information in database
Extension	1a. Data does not exist in the database 3a. Connection with database has been severed

Table 2.11: Use case Scenario 11

Use Case Name	Normalize sub scores and calculate final score (rating)
Pre-Condition	The following information must be available • Forum Metadata sub score • Forum Sentiment sub score
Post-Condition	Final forum activity score should be available
Actor	Forum Processor - Backend System
Main Success Scenarios	1. Get sub scores from the database 2. Normalize the sub scores 3. Calculate final score 4. Save final score in the database
Extension	1a. Data does not exist in the database 4a. Connection with database has been severed

Use Case Name	Extracting Videos from the source
Pre-Condition	Videos should be available in the source
Post-Condition	Extracted Video should be available in the database
Actor	Web Crawler - Backend System
Main Success Scenarios	1. Go to desired webpage. 2. Login to the page using credentials. 3. Gather video content. 4. Save content in the database.
Extension	1a. Connection Issues. 1b. Invalid content accessed.

Use Case Name	Splitting video files in to Image frames
Pre-Condition	Videos should be available in the database
Post-Condition	Image frames should be available after splitting
Actor	Video splitter – Backend System,
Main Success Scenarios	1. Select the desired video.

	2. Split the video into consecutive image frames from the beginning to the end of the video.
Extension	1a. Processing an invalid video

Use Case Name	Classification of Image fragment into a style
Pre-Condition	Image frames should be available after fragmentation
Post-Condition	Image frames should be categorized into the correct video production style
Actor	Image classifier – Backend System,
Main Success Scenarios	1. Select the image frame. 2. Pass the image through CNN 3. CNN automatically classifies the image into a video production style
Extension	1a. Processing an invalid image frame 1b. Processing a corrupted file

2.2.3 Activity Diagram

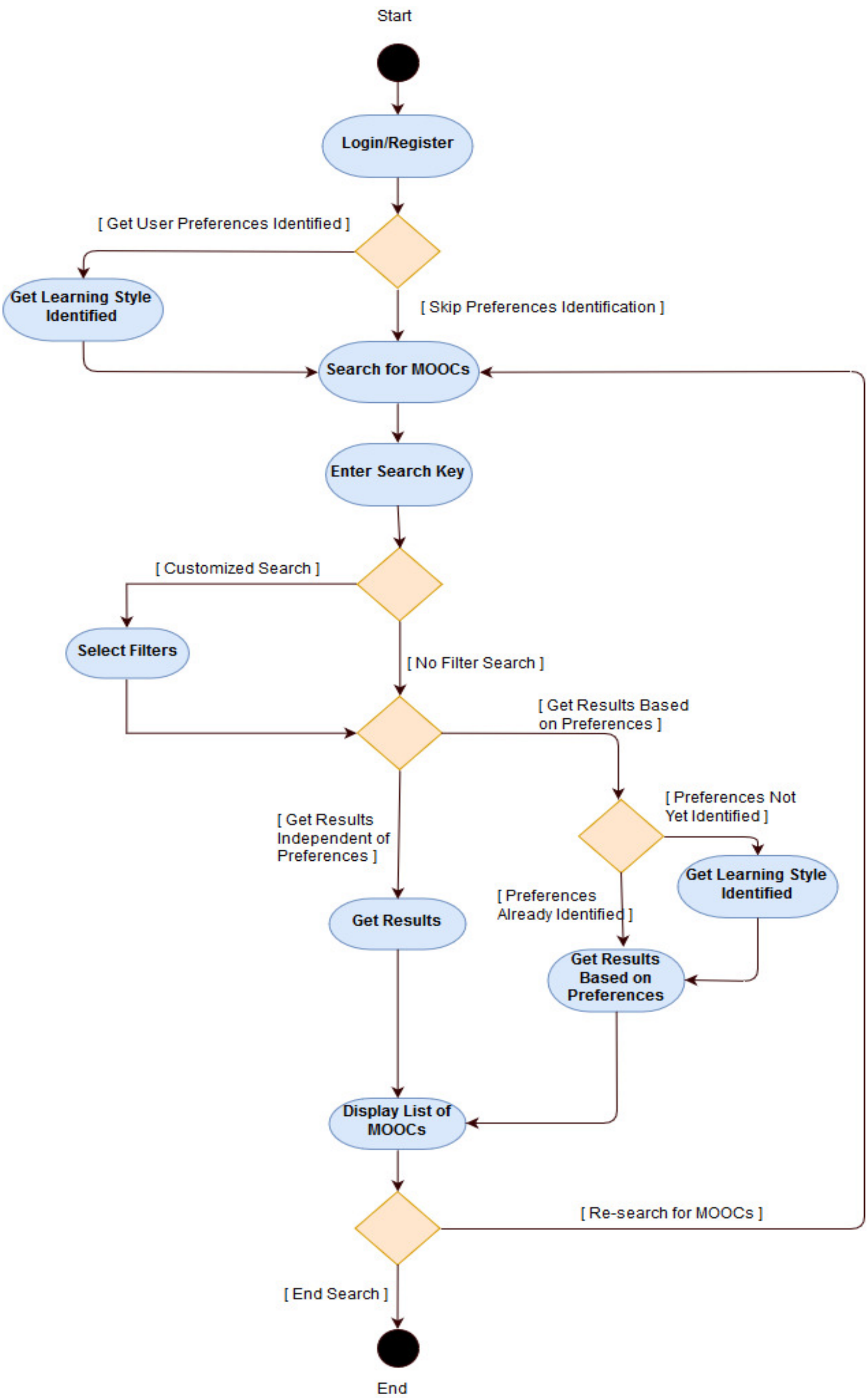

Figure 2.8: Activity Diagram

2.3 User Characteristics

- The end-user is an eager e-learner who is looking for MOOCs that suit his or her preferences. As such, any user with basic computer literacy can navigate through the system effortlessly.
- However, the bulk of the backend processes are comprised of many new technologies. As such, the developers who will interact with the said processes should have a good understanding about Python, MongoDB, JSON data structure, distributed computing concepts including message queues and containerization. Furthermore, they should be comfortable with machine learning aspects of the system such as dealing with TensorFlow and NLP modules.

2.4 Constraints

The system must adhere to a set of constraints in order to achieve its distributed and efficient processing power.

- The system must be cloud-ready and should ideally be implemented in AWS Cloud.
- Each server node must contain at least 8GB of RAM.
- Each server node must be connected to a shared storage device.
- Each server node must run CentOS 7 and if the system is implemented in AWS, the EC2 machine image should be Amazon Linux 2 which is based on CentOS 7.
- Python 3.7 must be used for this component.
- Containerization has to be done by using Docker.
- Messages must be JSON data.
- A NoSQL database must be used, preferably MongoDB.
- Video files must be deleted after classification of each video file.
- Only 1 message queue must be maintained.
- Each server node should have access to internet and the message queue.

2.5 Assumptions and Dependencies

- It is assumed that the user follows through the initial interactive process in order to receive more definite results.
- It is assumed that the user has a well-established internet connection.
- It is assumed that the servers will have a stable internet connection 24/7.
- It is assumed that the internet speed is at least as fast as 100Mbit.
- It is assumed that least 250GB of storage.
- It is assumed that Google Chrome or Mozilla Firefox or Microsoft Edge or Opera browser is used by the end-user.
- Depends on Chrome Driver (Standalone Application).
- Depends on AWS Cloud Infrastructure.
- Depends on GCP (Google Cloud Platform)'s NLP services.

2.6 Apportioning of Requirements

2.6.1 Essentials Requirements

- User should be able to rate, skip and skim through each segment of the interactive video session.
- User should be able to view transcript relevant to the intro video segment.
- Browser activity should be recorded for the duration of the interactive video session.
- Engagement of the user during each segment should be predicted based on both user input as well as the browser activity.
- System should display user's learner type depending on the learning material preference.
- MOOC recommendations must be based on the video production style the user showed most engagement with.
- Classify video files parallelly by splitting them into logical video chunks of smaller sizes.
- At the end, arrive at a single prediction for a given video file.
- Increase or decrease number of containerized classifier instances based on number of video files remaining.
- Gathering forum related data from the MOOC platforms Coursera, Edx and FutureLearn
- Analyzing the sentiments of review-type forum posts and calculating the sentiment sub score
- Analyzing the metadata of forums and calculating the metadata sub score
- Normalize the sub scores to compensate for the different types of data used to calculate them
- Use the sentiment sub score along with the normal ratings of MOOCs to get a new improved rating
- Use the two scores when the learner is being recommended MOOCs to improve the accuracy of recommendation

2.6.2 Desirable Requirements

- Users should be recommended MOOCs for video production style that showed the 2nd highest engagement.
- User should be able to re-interact with into video to update preferences in future as they need.
- Download lower resolution videos given that the resolution is still good enough for the video classifier to run effectively.
- Compress the data to improve the efficiency of storing forum related information in the database.
- Implement additional web crawlers to gather data from more external web sites like Reddit and Quora.
- Enable multiple web crawler instances to be able to work simultaneously which would significantly increase the scalability of this component.
- Enable the forum analyzer instances to be able to work simultaneously which would significantly increase the scalability of this component.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

While the figures 2.4, 2.5 and 2.6 provides how the user interfaces should look like, in order to capture the user inputs as well as user's browser activity during the interactive video session, this sections provides an overview as to how data and functions involved with aforementioned user interfaces come together from the perspective of a developer.

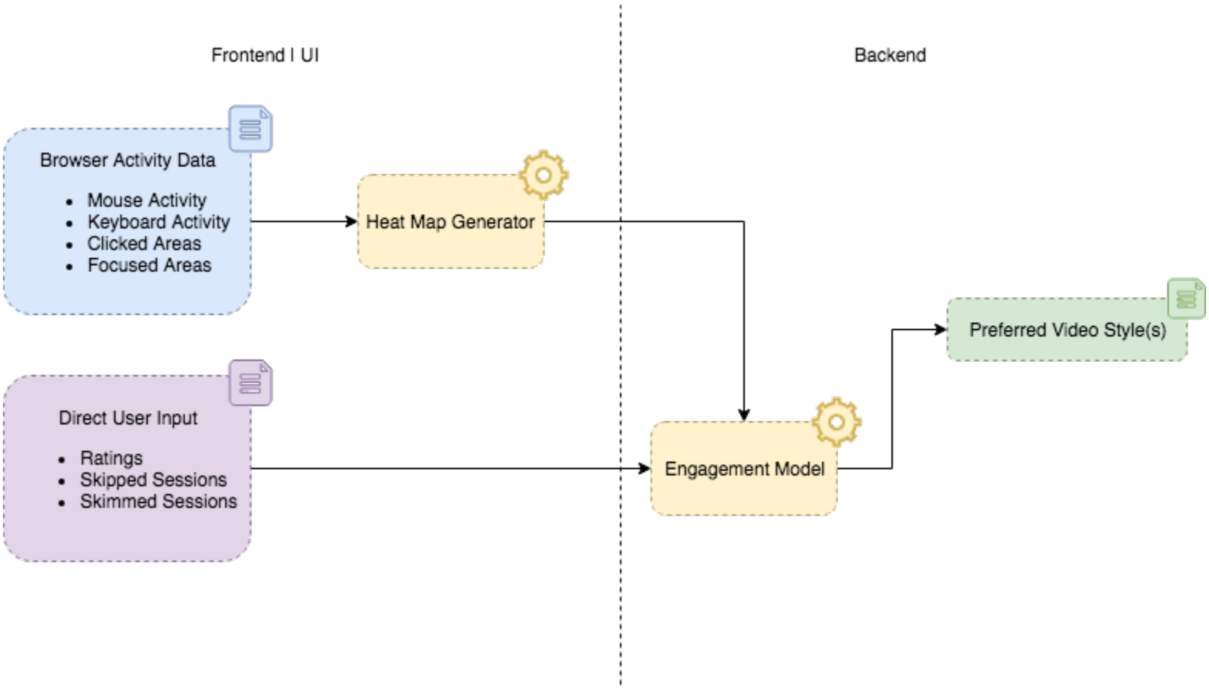

Figure 3.1: Data Flow within the User Interface

As shown in figure 3.1, browser activity data and user's direct input should be captured. This is achieved via user interfaces shown in figure 2.4 and 2.5. The responsibility of the user interfaces is to capture the above two sets of data.

Moreover, each user interface related to video session has to be verified using HCI techniques to see whether following goals are achieved.

- User is aware of the interactive elements
- Interactive elements represent the input needed from the user clearly
- Interactive area is smooth, and responsive
- Interactive area adapts to the video style that is playing at the moment

3.1.2 Hardware Interfaces

- 100Mbit network interface.
- SSD based EBS volume of 50GB or more.
- At least two EC2 instance with below specifications.
    - 8GB RAM
    - 8GB internal storage for the operating system
    - 04 or more vCPUs
- Another EC2 instance for running the DB with below specifications.
    - 16GB RAM
    - 25GB SSD based storage
    - 04 or more vCPUs

3.1.3 Software Interfaces

- Amazon Linux 2 operating system
- Python 3.7
- Docker
- RabbitMQ
- TensorFlow
- MongoDB
- JRE 8
- Node.js
- Google Chrome, Mozilla Firefox, Microsoft Edge, Opera
- Jupyter
- GCP API
- AWS API

3.1.4 Communication Interfaces

Since server sided infrastructure is created in AWS Cloud, all communication interfaces are managed by AWS and created when the servers are created.

3.2 Classes/Objects

Both user input as well as browser activity are represented in their own classes since the nature of two inputs are different. At the same time, the model/algorithm used to predict the engagement level of the user during each segment of the interactive video session is represented as its own class to enable reusability. By doing so, we can use the said model in predicting the engagement using the user input as well as the browser activity separately.

Finally, a recommender class is used to coordinate between all the above classes and their objects in order to analyze the database and recommend suitable MOOCs based on the output of above classes. This association of classes is depicted in figure 3.2.

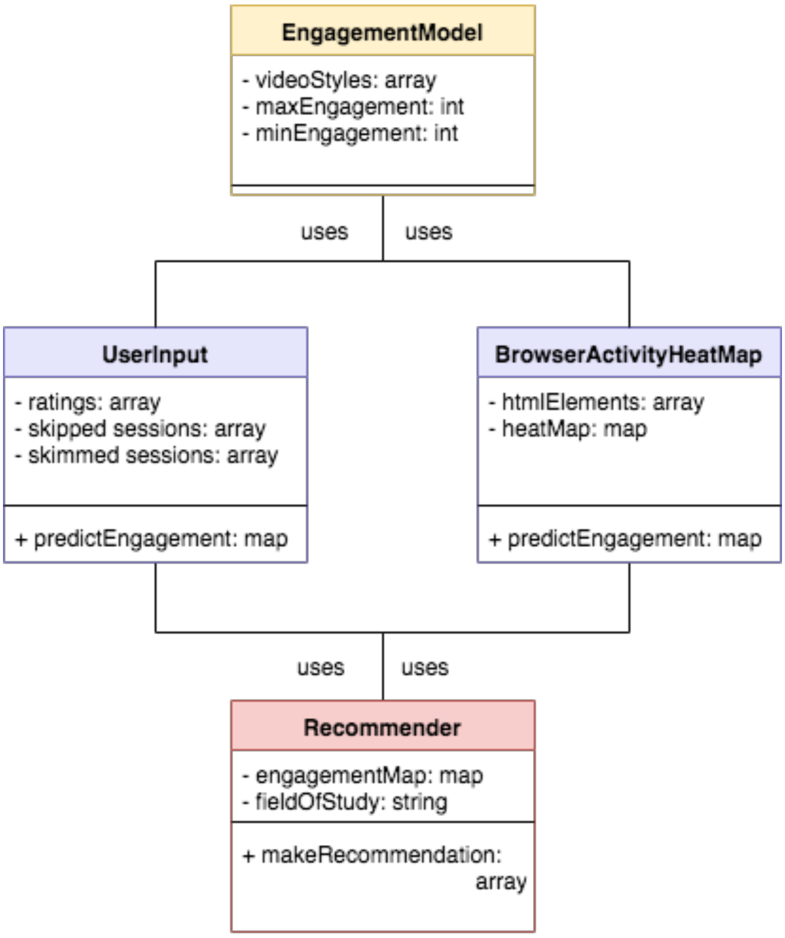

Figure 3.2: Association between Classes

3.3 Functions

3.3.1 Orchestrator

3.3.1.1 The orchestrator should mark a single video file into logical video chunks.

Table 3.1: Functions of Orchestrator

Description	The orchestrator should logical break down a large video file into small sections so that multiple service-workers can work on the large video file at different sections of it.
Sequence of Operations	1. Orchestrator goes through a video file and calculate its length 2. Orchestrator then calculates the start and end times of each logical video chunk in order to divide the large video file into small parts 3. Orchestrator then pushes messages containing details of each logical video chunk to the message queue.
Validations	Video file format should be supported by the classifier within the service worker.
Input	Video file

Output	Messages (in JSON format), each containing details about a logical video chunk.
	{
	“ParentFile”: “.....”,
	“Start”: “…..”,
	“End”: “…..”,
	“Chunk”: “…..”
	}

Exception/Error Handling	If the video file format is unsupported, log the error and notify the system administrator via an email.

3.3.2 Service Worker

3.3.2.1 Service worker should intercept a message in message queue.

Table 3.2: Functions of Service Worker in Message Queue

Description	The service worker should subscribe to the message queue and intercept a message in the message queue.
Sequence of Operations	1. Service worker subscribe to the message queue 2. Callback function within the service worker is called when a message is available 3. Message is read, and is converted to a Python dictionary for internal use (Even though the message is read, it is not acknowledged until the classification is successfully finished)
Validations	1. Message should contain a valid JSON string. 2. Message should contain all the necessary data.
Input	RabbitMQ Message
Output	A Python dictionary representing the JSON data within the message. It should contain following keys. [ParentFile, Start, End, Chunk]
Exception/Error Handling	If the message does not contain a valid JSON string, push the message back to message queue with an error message attached to it.

3.3.2.2 Service worker should trigger the classifier and retrieve its prediction.

Table 3.3: Prediction Function of Service Worker

Description	The service worker should call the classifier function after intercepting a message from the message queue. In return, the classifier will analyze the logical video chunk represented in the message and provide a prediction/classification.
Sequence of Operations	1. Service worker invokes the classifier with the Python dictionary created in *function 3.2.3.1* as an input 2. Classifier returns the prediction back to the service worker 3. Service worker pushes prediction details to the message queue 4. Service worker should acknowledge the initial message that was used to create the Python dictionary
Input	Python dictionary containing following keys. [ParentFile, Start, End, Chunk]

Output	A message containing following JSON data as a string.
	{
	"ParentFile": ".....",
	"Start": "…..",
	"End": "…..",
	"Chunk": "…..",
	"Prediction": "...."
	}

Exception/Error Handling	If the classifier crashes without providing a prediction, service worker should not acknowledge the initial message.

3.3.3 Analyzer

3.3.3.1 Analyzer should predict the video style of the original video file.

Table 3.4: Functions of Analyzer

Description	The orchestrator should logical break down a large video file into small sections so that multiple service-workers can work on the large video file at different sections of it.
Sequence of Operations	4. Orchestrator goes through a video file and calculate its length 5. Orchestrator then calculates the start and end times of each logical video chunk in order to divide the large video file into small parts 6. Orchestrator then pushes messages containing details of each logical video chunk to the message queue.
Validations	Video file format should be supported by the classifier within the service worker.
Input	Video file

Output	Messages (in JSON format), each containing details about a logical video chunk.
	{
	“ParentFile”: “.....”,
	“Start”: “…..”,
	“End”: “…..”,
	“Chunk”: “…..”
	}

Exception/Error Handling	If the video file format is unsupported, log the error and notify the system administrator via an email.

3.3.4 Gathering Data from Forum Threads

3.5: Gathering data from forum threads

Description	The web crawlers will gather data from forum threads regarding a MOOC
Sequence of Operations	1. Initialize and start platform specific crawler (e.g. Coursera Crawler if MOOC exists on Coursera) 2. Initialize and start external forum site crawler if topic related data does not exist within the database Below operations are common to both web crawlers 1. Retrieve the list of threads present 2. Iteratively traverse the threads and collect forum posts data 3. Store the collected data in the database 4. Notify main process that the task is complete

Validity Checks	It should have been more than a day (24 hours) since data was last gathered from forum threads regards that specific MOOC.
Input	MOOC Details (URL, Title, ….)
Output	Forum Thread Data
Error Handling	If collecting data from a certain forum thread fails, the operation will keep retrying for a set amount of time. If the time threshold exceeds the forum thread is considered deleted (non-existent)

3.3.5 Calculate Forum Activity Score using Metadata Analysis

3.6: Calculate forum activity score using metadata analysis

Description	Analyzing the metadata of posts of all forum threads regarding a MOOC and calculating the forum activity score
Sequence of Operations	1. Analyze forum metadata of each forum thread to get a thread specific score 2. Get average of all the scores of all the threads regarding the MOOC which is currently in analysis 3. Normalize the data to compensate for the different data attributes available in each MOOC platform 4. Notify main process that the task is complete
Validity Checks	Check whether all the required data is present and exists in the proper format
Input	Forum Thread Data
Output	Forum Activity Score (Numerical Value)
Error Handling	If task fails it will be considered unprocessed and a log record will be produced so that the maintenance personnel are notified.

3.3.6 Calculate Forum Rating using Sentiment Analysis and Existing Ratings

3.7: Calculate forum rating using sentiment analysis and existing ratings

Description	Analyzing sentiments of review type of forum thread and using existing ratings to come up with a more accurate rating
Sequence of Operations	1. Analyze sentiments of every post to get a score 2. Get average of all the post scores 3. Use existing rating along with the calculated sentiment score to get a new rating 4. Notify main process that the task is complete
Validity Checks	Check whether all the required data is present and exists in the proper format
Input	Forum Thread Data

Output	Forum Rating (Numerical Value)
Error Handling	If task fails it will be considered unprocessed and a log record will be produced so that the maintenance personnel are notified.

3.4 Performance Requirements

3.4.1 Network Performance Requirements.

- Network latency must be kept to a minimum to ensure messages are exchanged among the orchestrator, analyzer and the containerized classifier instances via the message queue optimally.
- Messages must only contain the most crucial information to keep the size of a message down to a minimum.
- Since the system is distributed, the components may need to communicate over the network, therefore it is important that the network speed and latency remain optimal to ensure each component is able to function optimally.

3.4.2 Disk Performance Requirements

- Video files should be read off of a solid-state shared storage device to ensure no bottleneck is created when transferring files between the classifier and the storage device.
- Each MOOC course must be classified only once.

3.4.3 Browser Performance Requirements

- Browser activity must be recorded with a minimum hit on browser performance.
- Browser should not be tasked with performing calculations/predictions.

3.4.4 Database Requirements.

- Proper indexes must be present to optimally analyze the database to retrieve results that belong to a certain video production style.
- Database must be able to handle concurrent write requests to different collections since a single DB can be used by multiple components simultaneously.

3.4.5 Web Crawling Requirements.

- Gathering data from a single forum thread related to a MOOC should not take more than 30 seconds
- Analyzing a single forum thread should not take more than 30 seconds
- Normalizing the sub scores should not take more than 10 seconds
- Usage of these new attributes should not add an overhead of more than a single second when recommending MOOCs to learners
- The web crawlers must collect data at least once a day

3.4.6 UI Performance Requirements.

- Since the user spends a considerable time with the interactive video session, which is crucial in finding the user's preferred video style, it is important that the user interface portion performance fluidly and efficiently.
- Browser activity must be recorded with a minimum hit on browser performance.
- Browser should not be tasked with performing calculations/predictions.

3.4.7 Service-worker Performance Requirements.

- No two containerized classifier instances should classify the same logical video chunk.

3.5 Logical Database Requirements

3.5.1 Data Format

- All data should be stored as JSON documents. MongoDB must be used in a cluster configuration to facilitate this.

3.5.2 Data Criteria

- All crucial details related to a MOOC must be stored in a separate database. Therefore, following aspects are considered crucial.
  - MOOC Name
    Name of the MOOC/Course

  - MOOC URL
    Direct URL to view the course at the course providers' site.

  - Video Style Classification
    A prediction as to what video production style the MOOC belongs to overall.

- The logical database requirements include the retention of the following data elements. The below mentioned data structures and lists are not complete and are only meant to be a starting point for development.

  - <u>Forum Threads</u>

    1. MOOC Platform
    2. URL
    3. Section
    4. Type
    5. Title
    6. Author
    7. Description
    8. Created Date/Posted Date
    9. Posts – Array of objects with the following attributes
       a. Author
       b. Post
       c. Date Posted
    10. Is Archived?
    11. Last Active Date
    12. Is Closed?
    13. Up votes
    14. Views

- The data records/documents/objects will be uniquely identified using an ID so that it will be linked to the MOOC collection which is separate from this. Multiple records of the above-mentioned collection can be linked with a single MOOC record (Many-to-One relationship) as a single course can have multiple forum threads regarding it.

- For the functionality of this component the MOOC collection will have the following attributes in addition to the ones which already exist and from the other components of the research project.

  - <u>MOOCs</u>
    1. Date of last forum analysis
    2. Activity Score
    3. Rating

3.5.3 Indexing

A single JSON document represent a single unique MOOC. Therefore, all documents should be indexed by the MOOC URL in order to uniquely identify courses with similar names provided by different course providers.

3.5.4 Data Accessibility

No user has direct access to data. Only the MOOC Recommender component should have access to data.

3.5.5 Data Availability

To ensure that classification data is not lost, MongoDB must be run in a cluster configuration with 1 master node and 2 slave nodes. This is done to prevent the need to re-classify MOOCs in case of a data loss.

3.6 Design Constraints

3.6.1 Standard Compliance

- The overall design must be compliant with containerization of applications. Furthermore, the Docker-images must be compliant with Docker standards and best practices to ensure high performance and security.
- Introductory video only contains MOOC learning material types which are identified by MOOCRec V2 such as talking head, animation, code/tutorial, presentation slides, khan academy writing and forum discussions.
- JSON must be used to communicate between components, and APIs.

3.7 Software System Attributes

3.7.1 Accuracy

- The system should predict the engagement level of the user for each segment of the interactive video session with a high accuracy since this is the main factory we consider when recommending MOOCs.
- The result we get by letting a set of people fill the ILS questionnaire to identify their learning style and the result we get by allowing the same set of people to interact with our intro video should be approximately the same in order to validate the developed component

3.7.2 Reliability

The ability of the system to keep classifying videos while keeping failures to a bare minimum. Given that the overall architecture is based on distributed computing and all sub-components can run independent of each other, a high level of reliability can be achieved.

- A message queue is used to convey tasks to containerized classifiers and every message will remain in the message queue until a response has been given by a containerized classifier instance. This ensures that another instance can take up the work if a given instance crashes during the classification process.
- By using a suitable auto-scaling mechanism, the number of containerized classifier instances can be kept at a desire number. This ensures that a new instance is started as soon as one crashes in order to keep the processing power balanced.

3.7.3 Performance

Since the user's preferred video production style along with the learning dimension are shown in a short period of time after the conclusion of the interactive video session, the backend service must perform efficiently and fast in order to avoid making the user wait.

3.7.4 Security

Since we are dealing with copyrighted content it is important that the information about MOOC courses as well as their video files remain securely without being exposed to outside.

- All EC2 instances will be created within the same Virtual Private Cloud (VPC) provided by AWS. This provides the ability to route all communications across servers internally instead of over the internet.

3.7.5 Maintainability

Give the importance of this component along with its performance sensitive nature, the system, its architecture, codebase and connected infrastructure must be maintainable to ensure future improvements as well as easy troubleshooting and seamless execution.

- Develop the orchestrator, analyzer and containerized classifier as de-coupled and independent modules so that each of them can be maintained without affecting the functionality of another.
- Create and maintain cloud infrastructure via code using Cloud Formation or Terraform to maintain consistency and transparency.
- Enforce Python coding standards in the code base.
- Functions which are used to traverse the MOOC platforms in the platform specific web crawler might have to be changed when the corresponding MOOC platform page routing structure changes.

3.7.6 Scalability

Since the component is aimed at achieving high levels of performance and efficiency, the containerized classifier instances must scale up or down based on the number of video files that have to be classified. This allows the system to use most of its resources when needed while additional server nodes can be shut down when the demand is low.

- Use EC2 auto scaling policies to scale the number of server nodes (EC2 instances) based on resource usage.
- Use container scaling policies to utilize most of the processing power in a single server node by running multiple containerized classifiers parallelly.

3.7.7 Availability

The system should show the user's preferred video production style along with the learning dimension within a short period of time after the conclusion of the interactive video session. Therefore, it is paramount that the backend service that analyze the data inputs perform efficiently

In order to keep an updated list of MOOC recommendations, it is important that the system keeps classifying new MOOC videos as they come.

- By using EC2 auto scaling policies, even if a server node crashes, a new one will be spun up right away to make the system highly available.
- By utilizing availability zones provided by AWS, server nodes can be scattered across different geographical locations to ensure continuous availability.

3.8 Organizing Specific Requirements

3.8.1 System Mode

The backend components should be designed to run by itself without any manual intervention from time to time.

4 References

[1] C. Kent, E. Laslo, and S. Rafaeli, "Interactivity in online discussions and learning outcomes," *Comput. Educ.*, vol. 97, pp. 116–128, 2016.

5 Appendix