Lab 1 – Product Description

Kevin Carr

Old Dominion University

CS411W

Professor Thomas Kennedy

11 October 2023

Version 1.0

Table of Contents

List of Figures

List of Tables

1 Introduction

The wherewithal that students must undergo in order to obtain a degree in higher education is a very demanding task depending on the student's circumstances. Some may have more support of friends and family than others do. Some may have to work full- or part-time jobs and still are required to hold onto their full-time academic status in order to keep their financial aid. 81% of students in the U.S. work jobs while attending college (Wan, 2022). No matter the circumstances the student may or may not have, the task of finding a comfortable, well-rounded work, education, and life balances can be very challenging. Before the beginning of every semester, students are required to register for their courses, choose their instructor, and time slot that best coincides with their personal or work schedule.

Sometimes this task can be stressful for the student because they lack the necessary information required to create a class schedule that fits around their personal or work schedule. Here at Old Dominion University (ODU), students cannot simply view the syllabi until after they have already registered for the course and, in some cases, have already paid the tuition fees. Currently, ODU does not have a tool that faculty and students can utilize to preemptively view the course syllabus before registering for a course. When students are registering for their courses, they can see the time slots, who the instructor is, and what medium the course is being taught, but what ODU's Banner Registration does not tell them is the amount of work that is required for the course.

Students need to know what the course is all about:

- How large is the amount of coursework?
- Is there a lot of writing involved?

- What is the total number of quizzes/exams during the semester?

- How will the overall work week be for the course?

ODU lacks a centralized tool that engages faculty and students, as well as assist them in cultivating a more comprehensive, well managed, customizable, and successful learning experience. They lack a way to communicate how classes are taught, as well as detailed information about what is taught, leading to poor feedback for professors and a heavy loss of time and money for students.

When it comes to faculty feedback, faculty are limited in the feedback they receive from students, where most times feedback is only being sent from the most passionate of students. Most of those students represent the opposites of a spectrum of whether the students liked the professor or not. Professor review platforms, such as "RateMyProfessor", are the only places where students can get detailed information on a professor, they are unfamiliar with. However, professors could rate themselves with no student verification (Boccaccio, 2018). Most instructors view the topic of accessibility to the syllabi as a necessity only to those registered for the course during the first week of class.

Here in *Figure 1*, this displays the normal process an ODU student would currently go through when choosing their classes.

Figure 1

Current Process Flow Diagram
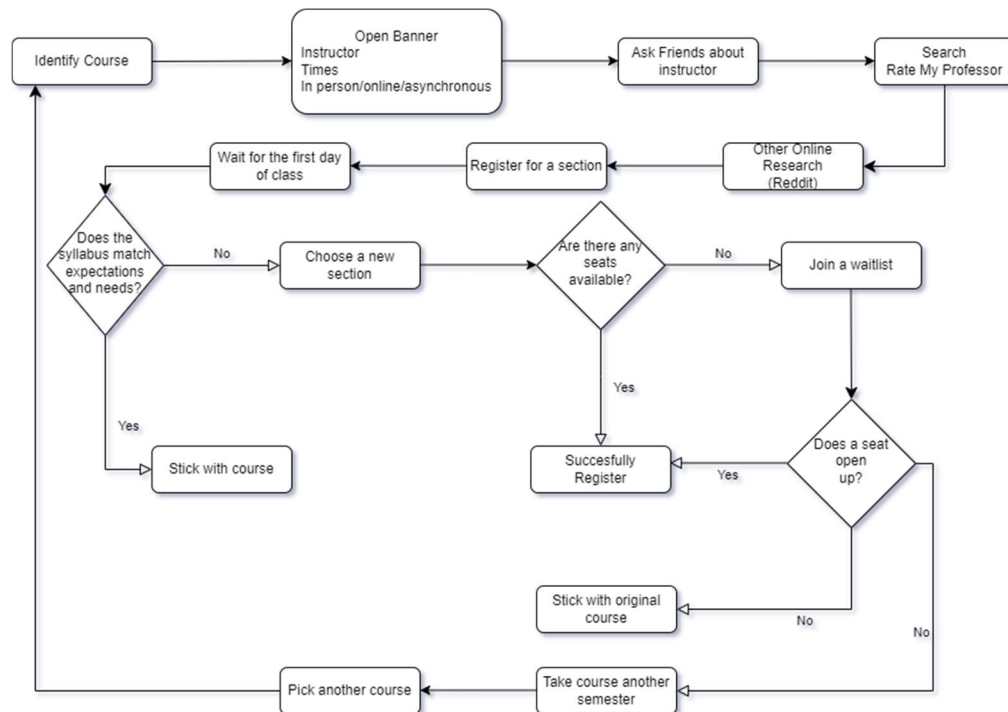
Note that even when the student believes that the course, instructor, time slot, and teaching medium have been selected; there is still that chance that the course that was chosen is not the best fit for them. After the first day of class, the student would have less of chance to find a course section that fits them due to limited availability now.

2 Monarch Course Explorer Product Description

Monarch Course Explorer (MCE) is a moderated platform that will be able to correct this problem. This platform will serve as a supplement for the advising services here at ODU. Students will have the ability to provide and view constructive feedback about courses and instructors that can be sorted by the semester. The faculty can view, comment, and utilize the feedback that is received by making changes to the courses as they see fit. An integrity check system will also be in place to ensure that poor quality feedback is removed. Monarch Course

Explorer will also serve as a central repository of course syllabi where the faculty can upload their course syllabi to make it more accessible to the students. Additionally, student advisers will also have access to assist the students that are unfamiliar with the platform or just to advise students in general with their educational needs. Monarch Course Explorer is tailored to create a more customizable and productive learning experience.

Here in *Figure 2*, displays the proposed process of a student choosing their courses for the semester while utilizing Monarch Course Explorer.

Figure 2

Proposed Solution Flow Diagram
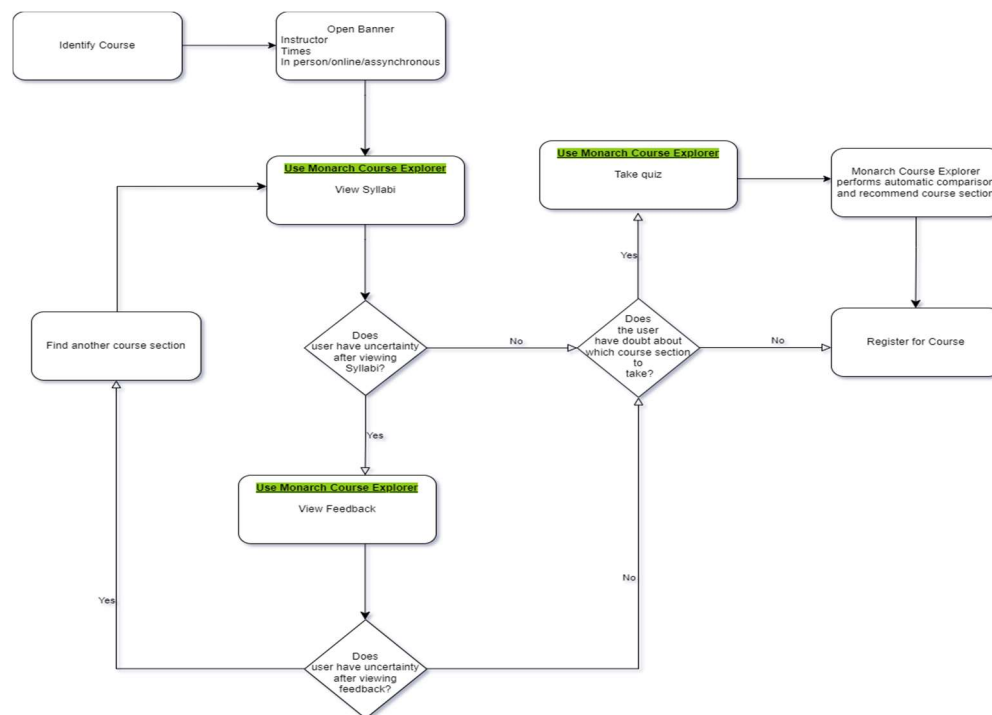
Note that even though the student must still choose the instructor, time slot, and the teaching medium; MCE now simplifies the process by providing the student with a preview of the course's syllabus. They will have all the needed information to make an informed decision.

2.1 Key Product Features and Capabilities

The first feature of Monarch Course Explorer is to provide access only to those with a verified ODU email address to ensure security, integrity, and privacy. Similarly with logging into Canvas or ODU Portal, the student will have to go through the two-step verification process using MIDAS. Once the credentials are verified, the user will be granted access to the platform – whether it is the student, faculty, or advisor capabilities of the platform. Once logged in, the menu options for the platform consists of:

- Viewing Course Syllabi
- View Feedback
- Give Feedback
- Report Feedback
- Upload Syllabi
- Compare Syllabi

Another feature of Monarch Course Explorer is the user's ability to review and compare course syllabi. This will provide students with more information about the course to fit their learning style and personal/work schedules. Just like a normal view of a course syllabus, it will display the detailed information including:

- Course description
- Instructor information
- Any textbook information
- Grading and course policies
- Learning medium (Online or On-Campus)
- Attendance policy
- Typical work week for the course

Comparing various syllabi will be automatic between semesters or faculty. It will also give customized recommendations for the user depending on their preferences and the filters they have chosen.

The next feature is that it creates a platform for verified ODU students to provide timely feedback on courses and instructors. Instructors will also be able to view the feedback they have received and respond accordingly. Students and faculty will be authorized to view, rate, provide, and filter feedback. When students are providing feedback, the website will perform integrity checks to ensure that users have thoroughly and thoughtfully given feedback. If the check feels as though it is of low or poor quality, it will not be submitted or even removed.

The last key feature that Monarch Course Explorer has pertains to the student advisor role. They will be given a resource to create a more personalized advising experience for their students. Advisors will only have a read-only functionality. They are authorized to view and compare syllabi, filter, and compare sections, and view and filter feedback.

2.2 Major Components (Hardware/Software)

Monarch Course Explorer utilizes 3 major components. The first component being the Graphical User Interface (GUI). The GUI will help support the front-end when users want to utilize the syllabi and feedback features. The second component of MCE are the algorithms that will scrape, normalize, analyze, and store in the database after they have been returned from the NLP Model API. The final component of MCE is MIDAS where the security and authentication aspect of the website is held. With MIDAS, users will be authenticated with their login credentials.

See *Figure 3* for an illustration of MCE's functional components.

Figure 3

Major Functional Component Diagram
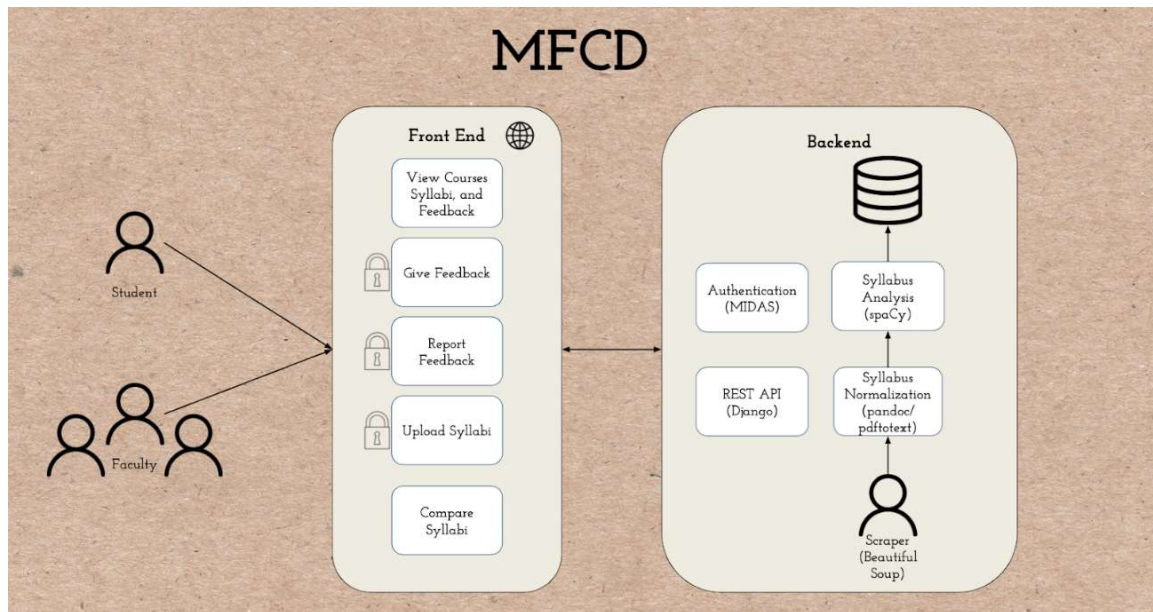
Note that MCE has a front-end and back-end. The front-end will handle all the GUI requirements while the back-end handles the algorithms that are needed.

There are two hardware requirements for a user to operate Monarch Course Explorer. The first is a device that is capable of connecting to the Internet. This device could be a smart phone, tablet, or desktop computer. The second hardware requirement is a connection to the Internet that the device can access.

The programming languages used to develop Monarch Course Explorer are Python, HTML, CSS, and JavaScript. HTML, CSS, and Javascript were chosen to create the website. HTML was utilized for the user display. CSS was used for formatting and design. JavaScript was chosen for the user interaction. Python was chosen for the programming language to implement all the algorithms along with the libraries that support the functions. The libraries are Beautiful Soup, spaCy, and Django. Beautiful Soup was used for web scraping, spaCy was used for NLP

transformation of syllabi, and Django was used for the rest of the API. For database management, PostgreSQL, was chosen. As stated before, MIDAS will be applied for authentication.

Programming tasks for Monarch Course Explorer were performed with an Integrated Development Environment (IDE), Microsoft Visual Studio Code (VSC). GitHub gave version control services and kept a record of all changes that were applied. The MCE development team utilized Trello for the agile story board. Primarily, Discord and Zoom were used for communications and coordination throughout the development process.

3 Identification of Case Study

The primary users for Monarch Course Explorer are ODU students, faculty, advisors, and curriculum committee. This case study will mostly focus on a diverse student population, including non-traditional students and online distance learning students. This website will be an aid to all students and not just CS majors. Future development opportunities will be to expand Monarch Course Explorer to other universities or higher education institutions.

4 Monarch Course Explorer Product Prototype Description

Monarch Course Explorer's main goal is to support students in their efforts to achieve their goals in higher education while also further improving the learning experience through proper feedback to the instructors. Implementing Monarch Course Explorer here at Old Dominion University can also result in higher degree completion rates and higher GPAs due the customizability of the various syllabi along with student's schedules. MCE's prototype will have some limitations where features will be eliminated and/or partially implemented due to time

constraints. Though MCE will be limited, the development team will still demonstrate its usefulness as a solution to the proposed problem.

4.1 Prototype Architecture (Hardware/Software)

The MCE prototype will feature most of the real-world product's capabilities and features. The hardware for the prototype only requires a desktop computer or any device that has Internet access. As far as software requirements, the website platform will be built using HTML for user displays, CSS for the format and design, and JavaScript for user interaction. The back-end of MCE will mostly be constructed using Python libraries. Beautiful Soup will be utilized for web scraping of pre-defined courses and feedback submitted by other users on existing platforms. To obtain information from uploaded syllabi, spaCy will use an NLP transformation to convert files viewable within the webpage. Django will exist for the rest of the API allowing connection and redirections. Authentication through MIDAS will be required to ensure security and privacy of the ODU faculty, staff, and students.

4.2 Prototype Features and Capabilities

As stated in *Section 4.1 Prototype Architecture*, a substantial amount of real-world product capabilities will be implemented in the prototype of MCE. Only a few features will be partially implemented. Refer to *Table 1, Table 2, Table 3,* and *Table 4* to view all real-world v. prototype comparisons.

Table 1

Syllabi Features Functionality

Features	Student	Faculty: Professor	Faculty: Advisor	Faculty: Curriculum Committee Member	Real World Product	Prototype
Upload Syllabi		✓		✓	✓	✓
Scrape Syllabi		✓		✓	✓	✓
View Syllabi	✓	✓	✓	✓	✓	✓
Side-By-Side View	✓	✓	✓	✓	✓	✓
Analyze Syllabi in Different Formats					✓	Partial

As shown in *Table 1*, most of the features will consist of both real world and prototype.

Due to time constraints, analyzing the syllabi in different formats will be partially implemented.

Table 2

Syllabi Features & Functionality

Features	Student	Faculty: Professor	Faculty: Advisors	Faculty: Curriculum Committee Member	Real World Product	Prototype
Verify Inclusion of Required Sections				✓	✓	✓
View Report of Missing Sections				✓	✓	✓
Fitler by Sections	✓		✓	✓	✓	✓
Compare sections	✓		✓	✓	✓	✓

All the features shown in *Table 2* here will be implemented in both the real world and prototype.

Table 3

Feedback Features & Functionality

Features	Student	Faculty: Professor	Faculty: Advisor	Faculty: Curriculum Committee Member	Real World Product	Prototype
View Feedback	✓	✓	✓		✓	✓
Rate Feedback	✓	✓			✓	✓
Provide Feedback	✓	✓			✓	✓
Filter Feedback by Semester	✓	✓	✓		✓	✓

Every feature shown in *Table 3*, will be implemented in both the real world and prototype.

Table 4

Other Features & Functionality

Features	Student	Faculty: Professor	Faculty: Advisor	Faculty: Curriculum Committee Member	Real World Product	Prototype
Authentication	✓	✓	✓	✓	✓	Partial
Take Quiz	✓				✓	✓
View Course Recommendation	✓				✓	✓

All the features shown in *Table 4*, will be implemented in some fashion with the exception of the authentication through MIDAS.

One of the main features of MCE will be its syllabi uploading feature. This where instructors can upload their course syllabi and the platform is able to take the bits and pieces of data and make a comprehensive report and filter out the information that's needed. As shown in *Figure 4*, the prototype will be able to follow the process of uploading syllabi and what occurs in the back-end of the software when it initializes.

Figure 4

Syllabus Algorithm Flow Chart
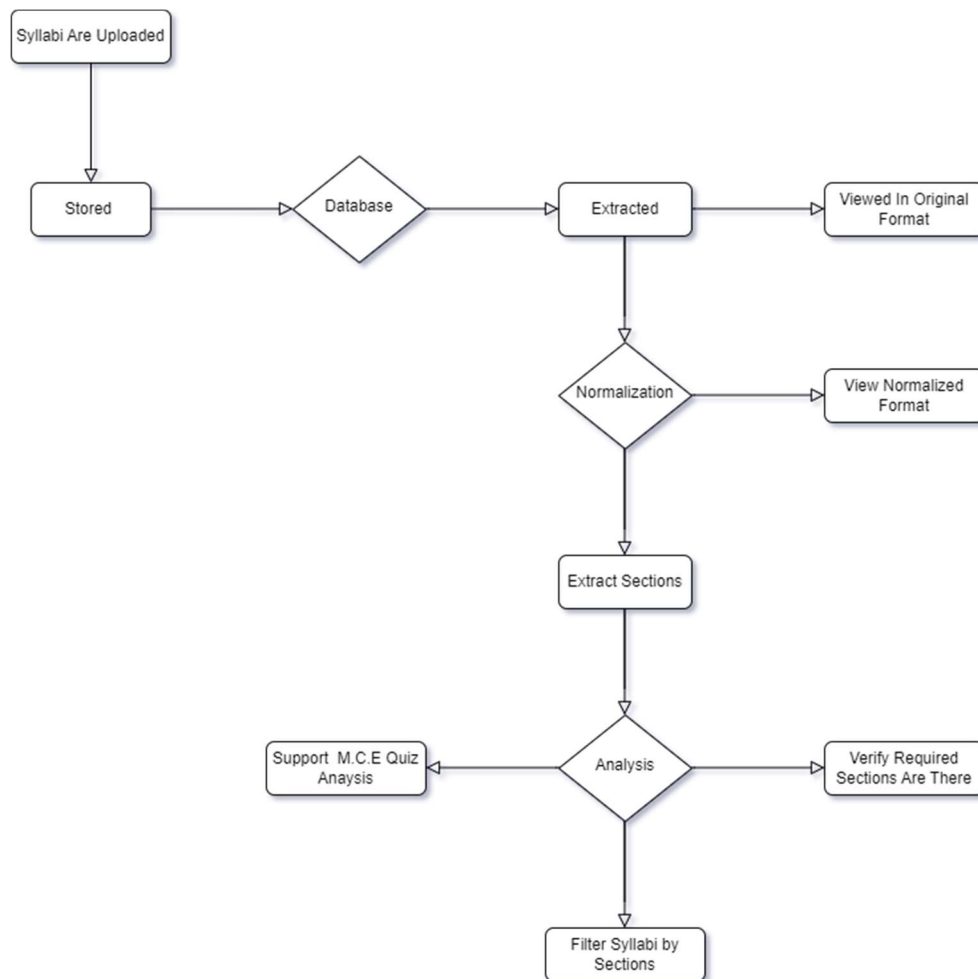
On the back end of the software, syllabi will be uploaded, standardized, analyzed, and saved to the database using PostgreSQL. This will allow secure and efficient database storage. Syllabi information will be stored securely, and remotely to avoid possible attacks and threats against the data.

The second main feature of MCE is the course/instructor feedback. This will allow students provide feedback on courses and instructors as well as filter the received feedback. The feedback options for instructors will allow them to view, rate, respond to feedback, and filter feedback. Also, the advisors will have feedback options as well that include viewing and filtering feedback. Here in *Figure 5*, this chart displays the back-end feedback algorithm the MCE will manage.

Figure 5
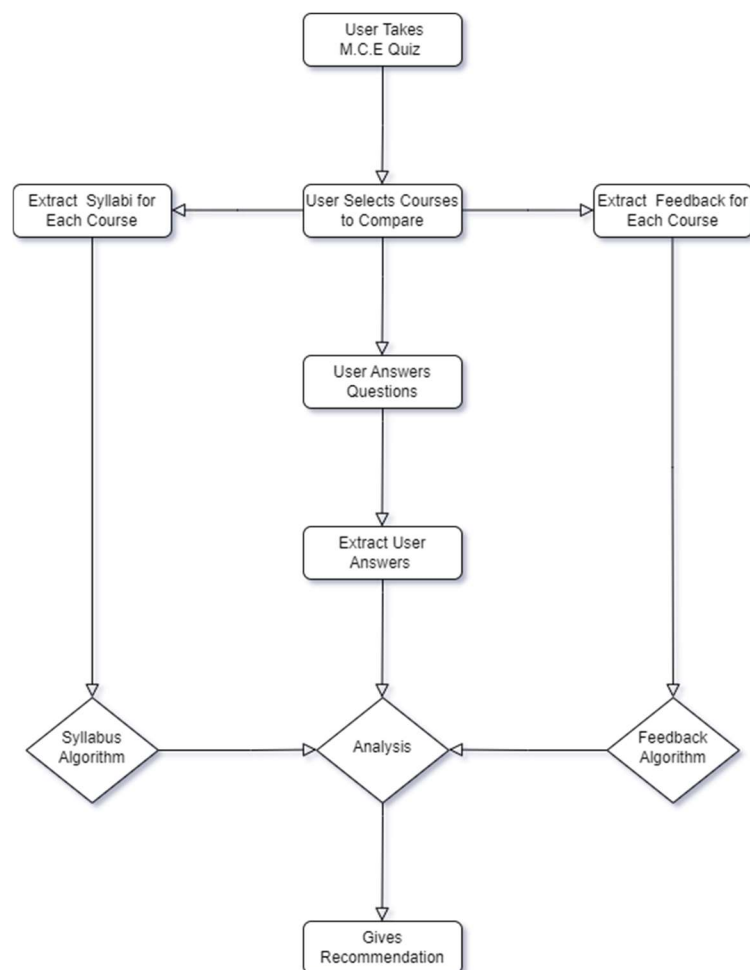
Feedback Algorithm Flow Chart
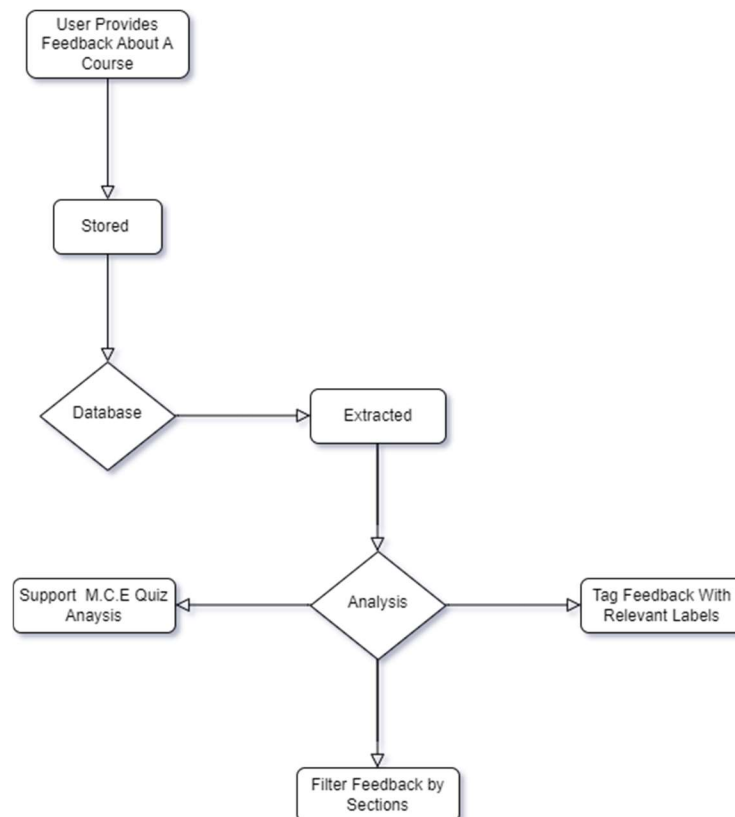
Figure 5 shows the user provides the feedback, its stored in the database, it is then extracted and analyzed. There will be various integrity checks, tags, and filters that coincide with the analysis as well.

One of the final features that MCE will be capable of are recommendations for the students. This will assist students in finding courses that fit their learning style and personal schedules. This is also where the already submitted syllabi and feedback are provided for the students to review and make their decisions. Here, in *Figure 6*, demonstrates the algorithm further.

Figure 6

Recommendation Algorithm Flow Chart

As described in *Figure 6*, the user will take the quiz to further filter out certain criteria that user is looking for in regard to what they're looking for. The syllabus and feedback algorithms will work in tandem with each other to give the user a recommendation.

4.3 Prototype Development Challenges

The prototype challenges that come with the platform is the accuracy of the NLP. When scraping and analyzing the syllabi and feedback, the results may be inaccurate when outputted. Another challenge is that the development team might face is data collection. Obtaining syllabi that are not already available online and motivating students to provide the feedback needed for their courses. There may be a need to add incentives for doing so. Lastly, normalization of the syllabi may cause some vital information to be left out as well as the amount of research that is required for these tasks.

In addition to the amount of research that is needed make this software, there is a smaller than normal time period in which to have this developed. There are many moving parts and components that may initially be in the prototype, but eventually will be moved to the real-world product version instead.

5 Glossary

API: Application Programming Interface is a software intermediary that allows two applications to communicate with each other.

Beautiful Soup: A Python library for parsing structured data.

CSS: Cascading Style Sheets are a means of describing the presentation of a document written in a markup language. They are typically used to support the display of HTML on web pages.

Discord: A Voice over Internet Protocol (VoIP) and instant messaging social media platform that allows the users to communicate with voice calls, text messages, and sharing files.

Django: A free and open-source, Python-based web framework that follows the model–template–views architectural pattern.

GitHub: An online software development platform that is used for storing, tracking, and collaborating on software projects.

HTML: Hypertext Markup Language, standard markup language for documents designed to be displayed in a web browser.

IDE: Integrated Development Environment is a software application that helps programmers develop software code efficiently.

MIDAS: Monarch Identification and Authorization System, Old Dominion University's log-in and password management system.

NLP: A subfield of computer science and artificial intelligence (AI) that focuses on the interaction between computers and humans in natural language.

PostgreSQL: A free and open-source relational database management system emphasizing extensibility and SQL compliance.

RWP: Real World Product that will be developed and used.

spaCy: An open-source software library for advanced natural language processing, written in the

programming languages Python and Cython.

6 References

Anderson, K., & Chinowsky, G. (2020, January 30). Students should have access to course syllabi before classes begin. The GW Hatchet. Retrieved January 23, 2023, from https://www.gwhatchet.com/2020/01/30/students-should-have-access-to-course-syllabi-before-classes-begin/

Boccaccio, Eric. "Debunking Myths about RateMyProfessors.com and Course Evaluations." Medium, 18 April 2018, https://medium.com/@green4172/debunking-myths-about-ratemyprofessors-com-and-course-evaluations-dd91453535aa. Accessed 14 February 2023.

Cartwright, S. (2016, September 28). Syllabi to be available online for students to preview before enrolling in classes. The Lantern. Retrieved January 22, 2023, from https://www.thelantern.com/2016/09/syllabi-to-be-available-online-for-students-to-preview-before-enrolling-in-classes/

Park, Y., & Sprung, J. M. (2013). Work-School Conflict and Health Outcomes: Beneficial Resources for Working College Students. Journal of occupational health psychology, 18(4), 384-394. https://doi.org/10.1037/a0033614

Wan, M., Feng, L., Meng, X., Zhai, M., & Konopaske, R. (2022). Working College Students' Time Pressure and Work-School Conflict: Do Boundary Permeability and Dispositional Mindfulness Matter? Psychological reports, 125(6), 3100-3125. https://doi.org/10.1177/00332941211029621