ISSN: 2473-4901
v10 n6190

Thirty-time Speed-up for Course Selecting by Using the Power of Machine

Zhouzhou Li
zli2@semo.edu

Juefei Yuan
jyuan@semo.edu

The Deparment of Computer Science
Southeast Missouri State University
Cape Girardeau, MO 63701, U.S.

Ashikur Rahman Nobel
anobel@mail.yu.edu

Department of Graduate Computer Science and Engineering
Katz School of Science and Health
Yeshiva University
New York, NY 10033, U.S.

Abstract

In the demanding environment of academic advising, faculty members face the significant challenge of assisting students with course selection and graduation planning within a tight timeframe. This paper presents a novel application of automation technology, termed Auto-Course Selection, which dramatically accelerates this process. Traditionally, constructing a tailored, error-free course and graduation plan for each student requires approximately 15 minutes, excluding further communications. The Auto-Course Selection system reduces this time to an impressive 30 seconds per student by automating the evaluation of prerequisites, checking course availability, resolving scheduling conflicts, and ensuring alignment with graduation timelines. This study details the development and operational deployment of the system, emphasizing the methodology for data acquisition, the architecture of the data structures, the logic underpinning the course recommendation algorithm, and the overall feasibility of the project. Our results indicate a thirty-fold increase in efficiency, providing a scalable and reliable solution that substantially eases the advising burden on faculty, particularly those managing large cohorts, and minimizes the potential for human error.

Keywords: Academic Advising, Course Selection, Automation, Data Acquisition, System Deployment, Educational Technology, Efficiency Improvement, Machine Learning.

1. INTRODUCTION

In the realm of academic advising, one of the perennial challenges faced by faculty members is the efficient and accurate selection of courses and planning of graduation timelines for students (Algarni et al., 2023; Gong et al., 2024). This process, traditionally manual and time-

consuming, is crucial to ensuring that students meet their academic requirements and achieve their educational goals within the stipulated time frame. Appendix A demonstrates the manual approach. However, it often results in significant time expenditure and a high potential for human error, especially when dealing with large cohorts of students.

With the rapid advancements in technology, particularly in the fields of software development and automation (Li et al., 2024; Mirzaolimovich et al., 2023), there exists an opportunity to revolutionize the way academic advising is conducted. This paper introduces Auto-Course Selection, an innovative system designed to streamline and expedite the course selection process through enhanced automation techniques. By automating key tasks such as evaluating prerequisites, checking course availability, resolving scheduling conflicts, and aligning course selections with graduation timelines, this system can reduce the time required for these tasks from an average of 15 minutes to just 30 seconds per student.

The development and deployment of the Auto-Course Selection system represent a significant advancement in educational technology (Atalla et al., 2023; Maphosa et al., 2023). This system not only enhances efficiency but also ensures a higher degree of accuracy in the advising process. By minimizing human intervention, the potential for errors is drastically reduced, providing a more reliable and scalable solution for academic institutions.

This paper provides a comprehensive overview of two Auto-Course Selection systems (Python-based and Web-based), detailing their development process, data acquisition methodologies, data architecture, and the logic behind their course recommendation algorithms. Our study demonstrates the systems' feasibility and effectiveness, highlighting a thirty-fold increase in efficiency. This significant improvement underscores the systems' potential to ease the advising burden on faculty members, particularly those managing large student populations and to enhance the overall academic experience for students.

The subsequent sections of this paper are structured in the following manner. In the 'Literature Review' section, we review the benefits and importance of auto-course selection. In the 'Methods' section, we describe the data acquisition techniques, data sources, and course recommendation logic in the Auto-Course Selection system, supported by visual workflows and examples from the Southeast Missouri State University (SEMO) course enrollment systems. In the 'Experimental Results and Analysis' section presents the details and outcomes of the web-based and Python-based auto course selection systems. The 'Conclusions and Future Work' section will offer a wrap-up of the findings and shed light on potential avenues for future research.

2. LITERATURE REVIEW

The benefits of auto-course selection are numerous, and research in this area is ongoing. One study published in the Applied Information Technology and Computer Science journal developed a Course Selection Guidance System (CSGS) (Zaidi et al., 2023). CSGS is a web-based tool developed to aid Form 5 students in selecting diploma courses at University Tun Hussein Onn Malaysia (UTHM) and University Putra Malaysia (UPM), tailored to their personality types and academic results. Utilizing Holland's Theory for personality assessment and developed with HTML, JavaScript, PHP, and SQL, the CSGS undergoes alpha and beta testing to optimize user experience. However, the CSGS system cannot be widely used by other universities.

Another study published in the Applied Mathematics and Nonlinear Sciences presents a teaching information management system that utilizes a nonlinear differential equation approach, specifically the Riccati differential equations with statistical linearization, to optimize the college course selection process. The system employs collaborative filtering based on nonlinear differentiation and student feature classification, and experiments have shown it to be correct in recommending courses 34.6% of the time. It concludes by emphasizing the system's potential to enhance teaching resource allocation and provide optimal elective plans, contributing to the development of "intelligent campus" and "humanization" in education within Chinese universities (Yangg et al., 2023). Nonetheless, the system's application range is overly broad, and its course selection accuracy remains limited.

The third study, published in the International Conference on Modern Education and Information Management developed a hybrid recommendation algorithm for a university course selection system designed to provide intelligent course recommendations based on relationships among students and between students and courses. The paper explores the development of a recommendation system that

ISSN: 2473-4901
v10 n6190

combines collaborative filtering with content-based techniques by utilizing both explicit and implicit student data. The findings indicate that the algorithm effectively addresses the randomness and lack of direction in student course selection, ensuring a better alignment of students' learning abilities with course demands (Luxiao et al., 2022). However, this system focuses more on establishing relationships between courses rather than helping students choose courses.

In conclusion, while various systems have been developed to enhance the process of course selection, each exhibits unique strengths and limitations. Systems like CSGS offer tailored recommendations but lack broader applicability across different institutions. Meanwhile, approaches using complex models like Riccati differential equations provide innovative solutions but may suffer from broad applicability and lower accuracy issues. On the other hand, hybrid recommendation systems show promise in aligning student capabilities with course demands, though they might prioritize relational data over direct course selection assistance. The ongoing evolution in this field suggests a move towards more integrated, adaptable, and student-centric course selection technologies that can potentially transform academic advising across diverse educational landscapes.

3. Methods

This section explains the implementation and deployment process of the Auto-Course Selection research project by utilizing two ways: 1) Web-based auto-course selection with *html* and *JavaScript*. 2) Python-based auto-course selection with *re* and *subprocess* modules. The primary focus are the sources of information, data acquisition - techniques and feasibility, data structure building, course recommendation logic and deployment. Appendix B shows the base workflow for auto-course selection.

Problem and Objective: To automate and speed up the course advising task for academic advisors is the primary goal here. Organized information extraction from relevant sources is an addition to this objective.

The data collection and experimental procedures detailed in this paper utilize information from the Department of Computer Science at SEMO course enrollment webpage. The methodology employed is thorough and widely applicable, as the majority of university course enrollment systems globally are web-based. This approach ensures that the findings are relevant and can be generalized to similar systems at other institutions.

Data Information Sources
The required information is split into three sets:

1. Pre- and Co- Requisite information: from SEMO Course Bulletin (SEMO BULLETIN, 2024)
2. Offered Courses (per semester): Look-up Classes
3. DegreeWorks (per student): SEMO DegreeWorks (SEMO DEGREEWORKS, 2024)

Pre- and Co-Requisite information: from SEMO Course Bulletin, which is publicly available information acquired directly by fetching the page. Offered Courses (per semester): Look-up Classes are to be copied and pasted by each client/student into a webpage field. DegreeWorks (per student): SEMO DegreeWorks is to be copy-pasted by each client/student. Appendix C visualizes the data format across the sources.

Data Acquisition Techniques
SEMO Course Bulletin: This section is to be fetched while loading the client-side web page/app.

The client-sided solution makes one single PHP call to fetch the page from the server, as shown in Appendix D.

Look-up classes:
Here is the process of providing information of Look-up classes to the client-side tool:

1. Under look-up classes tool in my.semo.edu, pick the specific semester.
2. Then choose 'Advanced Search' at the end of the page,
3. In the 'Advanced Search' page pick all of the department options by clicking the top and scrolling to the bottom and shift+click the bottom option,
4. Click search, in this search page, click anywhere, then do Ctrl+A and Ctrl+C (copy all),
5. Lastly, go to the tool page and paste it into the Look-up classes box.

Degreeworks: Same as above, but click inside the Degreeworks window before copying everything and pasting it into the Degreeworks section/textbox of the tool.

3. Experimental Results and Analysis

Web-based Auto Course Selection

Information Extraction:

The three sets of information are extracted using 'regular expression' (Regular Expression, 2024), 'jQuery' (JQuery, 2024) and vanilla JavaScript codes. The end results are the following data structures:

- scrapedPrereqData: JavaScript dictionary with the course codes as 'key' and list of prerequisites as 'value'.
- scrapedCoreqData: same as above but for co-requisites.
- offered courses: JavaScript dictionary with the course codes as 'key' and time - course name - days as 'value'.
- completed courses: JavaScript dictionary with the completed course codes as 'key' and true as 'value' (for easy and fast access).
- remaining courses: JavaScript dictionary with the 'not yet completed' course codes as 'key' and true as 'value' (for easy and fast access).
-

The aforementioned set of information would suffice in generating course selection.

Course Recommendation Logic:

The steps follow:

1. Remove all completed courses from remaining list (Completed courses end up with remaining ones due to Degreeworks formatting)
2. Only keep remaining courses in the list if it's offered in the semester
3. On each remaining courses, then, check if the pre-requisites are "None"
    - If "None", push the course to 'recommendation list'
    - Else, start checking for pre-requisites - if they are completed or not (also account for special cases - like consent of instructor/chairperson)
        - If the pre-requisites are completed (checked on each individual groups - if any group is completed) - push course into 'recommendation list', add note for special cases
        - If incomplete, do nothing
4. For each remaining course, check for co-requisites in the same way. If a course that is already in the recommendation list appears as a co-requisite, then also add the current course to the 'recommendation list' (based on the co-requisite condition)
5. Process the 'recommendation list' for final output (to make it look better)

Deployment:

Because the deployment involves a single PHP function call, installing PHP on the server side is necessary. The application is designed to be browser-friendly and features a minimalistic user interface. Therefore, the following server-side requirements are essential for deployment (No special configuration required, can be customized as needed):

1. Apache or equivalent web server
2. PHP to work alongside the web server

On the client side, all processes except data acquisition are automated, which includes generating recommendations and retrieving the bulletin without user intervention. The sole significant external resource on the client side is 'jQuery', which is sourced from the Google Web Hosted Library (Google Hosted Libraries, 2024).

Open Worldwide Application Security Project (OWASP) and Zed Attack Proxy (ZAP) testing:

Initial testing for this project was conducted using the security tool - Owasp Zap (Owasp Zap, 2024), to identify usable segments of the data sources. For this purpose, a collection of standalone scripts was developed, mirroring the workflow of the primary client-side solution.

After importing the scripts into the Standalone section of the Zap Scripts, the pages (Degreeworks, Course Bulletin, Look-up Classes) need to be loaded into the proxy history, then all scripts except the 'main' script needs to be run first and then lastly, the 'main' script needs to be executed. The result will be the recommended course list shown in the 'console' display tab. Appendix E displays the interface of the web-based auto course selection system, while Appendix F illustrates the email content automatically generated by this system, which will be sent to students advising them on which courses to enroll in for the upcoming semester.

Python-based Auto Course Selection

Information Extraction:

The three sets of information are obtained utilizing 're' (Python re Module, 2024) and 'subprocess' (Python subprocess Module, 2024) modules in Python:

- 're' module: parsing and extracting relevant information from text data, such as course codes, prerequisites, or availability from online catalogs or web pages. In addition, validating user inputs,

like ensuring course codes or identifiers match a specific format before processing.
- 'subprocess' module: Interacting with system-level scripts that might be part of the course selection infrastructure, like scripts for database updates, enrollment processes, or integration with other institutional software. In addition, automating tasks require external commands or applications, such as data backups or system checks related to course selection systems.

The steps follow:

1. Use the 're' to scrape and parse course information from the university's course catalog web pages. Regular expressions can help in extracting structured data like course names, numbers, prerequisites, and descriptions.
2. Before allowing a teacher/student to select courses, use 're' to ensure that their inputs (like course codes or names) match the expected patterns, thus avoiding errors during the selection process.
3. Use the 'subprocess' module to execute the command directly from the code. Refer to Figure 1 for the specific command.
4. After the courses are selected, an email to the student will be automatically generated and prepared for sending. Appendix G provides an example of the generated email.

Figure 1: Run Command to Execute Auto Course Selection System.

By combining these two powerful modules in Python, we can create a robust system that automates the process of course selection for students, ensuring efficiency and accuracy while interacting with existing university systems. These web-based and python-based auto course selection systems speed up the course selection advising and reduce the course selection time from 15 minutes to 30 seconds for each student. We inserted code to print out the timestamp when the tool starts to handle a student's data, and print out another timestamp when the tool completes the course selection for the student by generating the advising email. The difference between the two timestamps will be the time consumption of course selection per student. In 99.99% cases, the time consumption is less than 5s a student. The declared '30' seconds are a conservative number. Also, we would like to point out that we did not use a dedicated machine to host the tool. We simply allocated a Kali Linux VM with a 30 GB storage, 4 GB RAM, and two cores from a DELL laptop (11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 1.50 GHz, 16 GB RAM, Windows 11 Education 23H2) to run this tool. Therefore, 30 seconds are a very conservative number. Regarding the time consumption in traditional manual course selection, it is calculated from the fact that faculty reserve a 15-minute meeting per advisee for their course selection.

4. CONCLUSIONS AND FUTURE WORK

In this study, we introduced the Auto-Course Selection system, leveraging the capabilities of machine learning and automation to significantly enhance the efficiency of academic advising. By automating the critical aspects of course selection, including prerequisite evaluation, course availability checks, scheduling conflict resolution, and alignment with graduation timelines, our system reduces the advising time per student from 15 minutes to just 30 seconds. This thirty-fold increase in efficiency not only eases the workload of faculty advisors but also ensures greater accuracy and consistency in the advising process.

The deployment of both web-based and Python-based implementations demonstrates the system's versatility and robustness in interacting with existing university infrastructures. The detailed methodology for data acquisition, the architectural design of data structures, and the underlying logic of the course recommendation algorithm collectively highlight the system's comprehensive approach.

Our experimental results confirm the significant impact of the Auto-Course Selection system, providing a scalable, reliable, and error-minimizing solution for academic institutions. This innovation holds the potential to transform academic advising, allowing faculty to manage larger cohorts with reduced effort and increased precision, ultimately benefiting both advisors and students. As universities continue to seek efficient and effective solutions to meet the growing demands of academic advising, the Auto-Course Selection system represents a pioneering advancement in educational technology.

Looking ahead, there are several avenues for expanding and enhancing the Auto Course Selection system. A key area for improvement involves integrating advanced machine learning techniques. Universities offer a wide range of elective courses, both within and across different departments. By allowing students to input their research interests or career goals, the system could analyze the patterns and content of available elective courses. Consequently, it could provide personalized recommendations, suggesting the elective courses that best align with each student's individual interests and aspirations. This enhancement would not only optimize course selection but also help students make more informed decisions, ultimately supporting their academic and professional development.

Another essential aspect for future work is the improvement of the universities' curriculum. For example, by analyzing enrollment patterns and trends, machine learning can forecast demand for specific courses, helping universities to optimize course offerings and allocate resources effectively. In addition, machine learning can identify gaps in the current curriculum by comparing it with industry trends, job market requirements, and student feedback. This can inform the development of new courses or the revision of existing ones.

5. REFERENCES

Algarni, S., & Sheldon, F. (2023). Systematic Review of Recommendation Systems for Course Selection. Machine Learning and Knowledge Extraction, 5(2), 560-596. https://doi.org/10.3390/make5020033

Atalla, S., Daradkeh, M., Gawanmeh, A., Khalil, H., Mansoor, W., Miniaoui, S., & Himeur, Y. (2023). An intelligent recommendation system for automating academic advising based on curriculum analysis and performance modeling. Mathematics, 11(5), 1098. https://doi.org/10.3390/math11051098

Gong, T., Li, J., Yeung, J. Y., & Zhang, X. (2024). The association between course selection and academic performance: exploring psychological interpretations. Studies in Higher Education, 1-14. https://doi.org/10.1080/03075079.2024.2310151

Google Hosted Libraries. (2024). URL: https://developers.google.com/speed/libraries/.

JQuery. (2024). URL: https://jquery.com/.

Li, K., Zhu, A., Zhou, W., Zhao, P., Song, J., & Liu, J. (2024). Utilizing deep learning to optimize software development processes. arXiv preprint arXiv:2404.13630. https://doi.org/10.48550/arXiv.2404.13630

SEMO BULLETIN. (2024). Bulletin, URL: https://semo.edu/student-support/academic-support/registrar/bulletin/courses/bltn_data.php/ .

SEMO DEGREEWORKS. (2024). DegreeWorks, URL: https://semo.edu/student-support/academic-support/registrar/degree-works/ .

Luxiao Zhu, and Ben Wang. (2022). Course Selection Recommendation Based on Hybrid Recommendation Algorithms. Proceedings of the 2022 3rd International Conference on Modern Education and Information Management (ICMEIM 2022). Atlantis Press, 2022. Retrieved July 16, 2024 from 10.2991/978-94-6463-044-2_60

Maphosa, V., & Maphosa, M. (2023). Fifteen years of recommender systems research in higher education: Current trends and future direction. Applied Artificial Intelligence, 37(1), 2175106. https://doi.org/10.1080/08839514.2023.2175106

Mirzaolimovich, S. M., Ugli, N. R. M., & Ugli, K. E. I. (2023). Development of automated management system in technical processes. Science and innovation, 2(A4), 195-198. Retrieved July 16, 2024 from http://dspace.umsida.ac.id/handle/123456789/23686

Owasp Zap. (2024). : Proxy. URL: https://www.zaproxy.org/.

Regular Expression. (2024). URL: https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference.

Python re Module. (2024). URL: https://docs.python.org/3/library/re.html.

Python subprocess Module. (2024). URL: https://docs.python.org/3/library/subprocess.html.

Yangg, Yingfa, and Hui Zhao. (2023). Nonlinear Differential Equation in University Education Information Course Selection System. Applied Mathematics and Nonlinear Sciences

ISSN: 2473-4901
v10 n6190

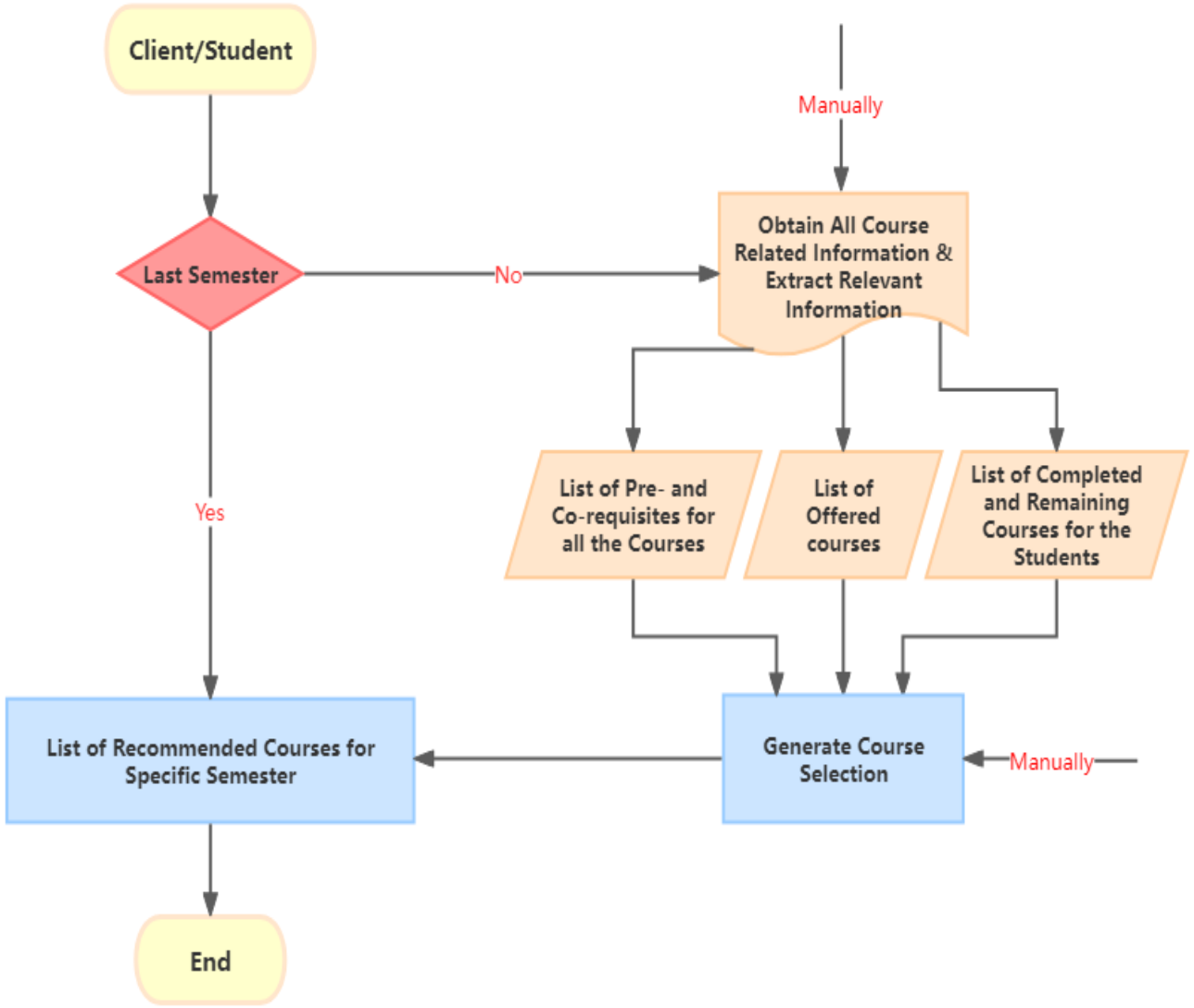
8.2 (2023): 1463-1474. https://doi.org/10.2478/amns.2023.1.00047

Zaidi, Nur Hilda, and Norhamreeza Abdul Hamid. (2023). The Development of Course Selection Guidance System. Applied Information Technology And Computer Science. 4.2 (2023): 1597-1616. Retrieved July 16, 2024 from https://penerbit.uthm.edu.my/periodicals/index.php/aitcs/article/view/12364

ISSN: 2473-4901
v10 n6190

APPENDIX A

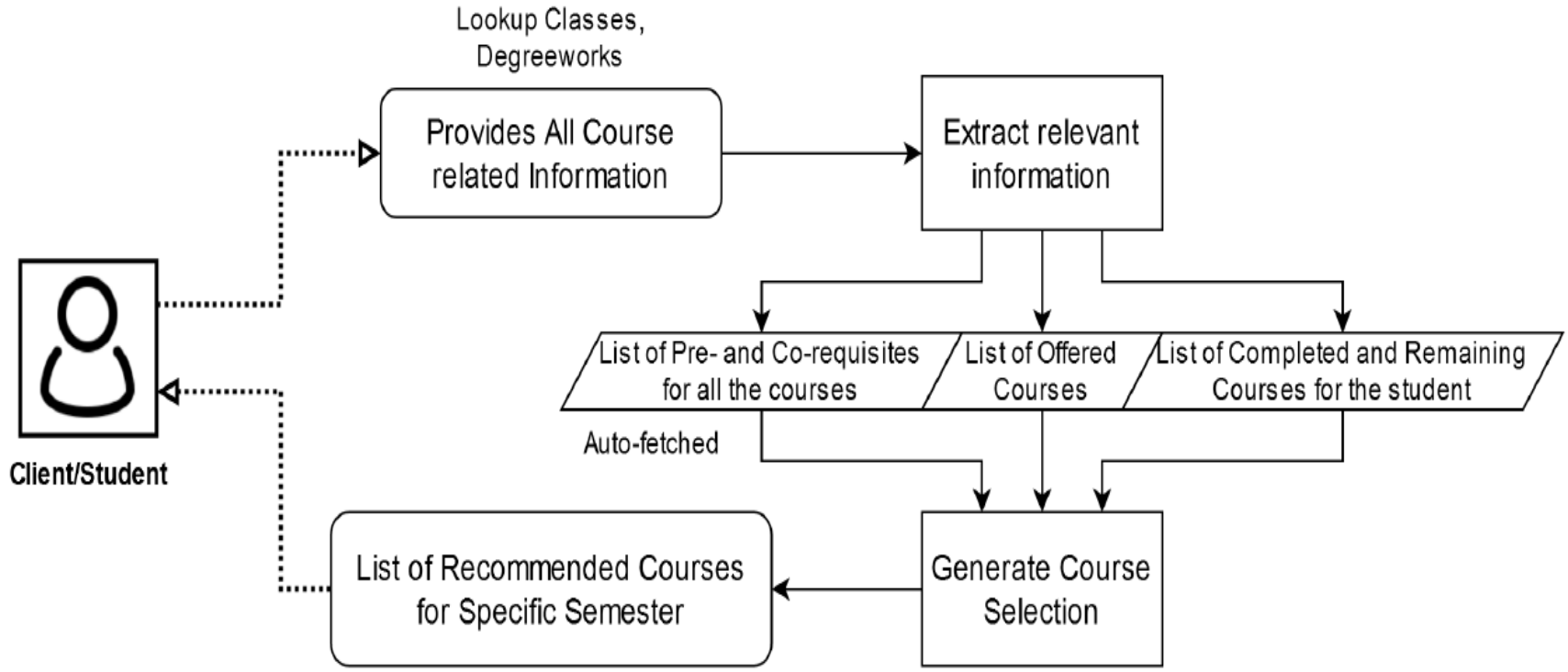
Manual Approach Course Selection



ISSN: 2473-4901
v10 n6190

APPENDIX B

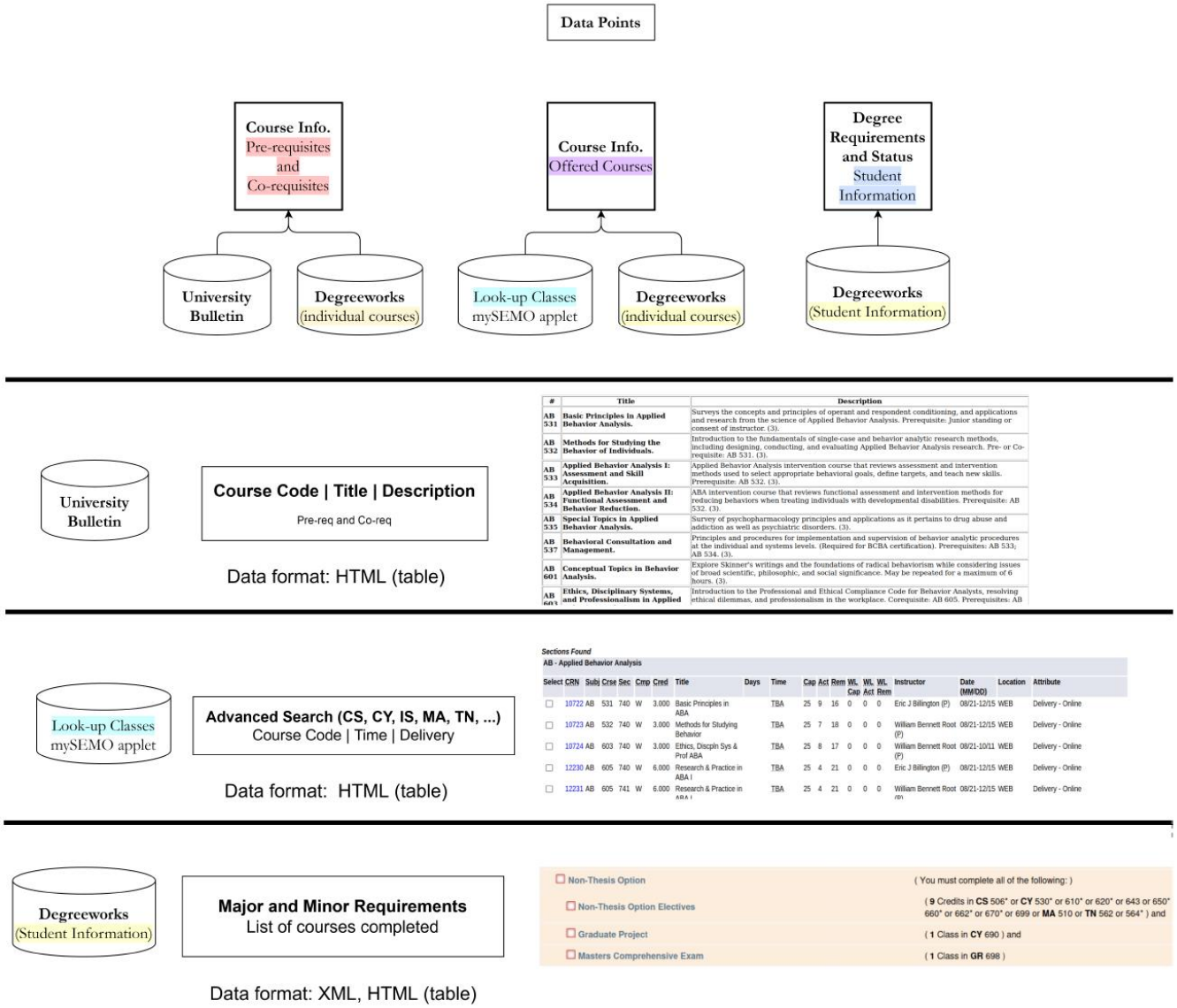
Base workflow: Auto-Course Selection



ISSN: 2473-4901
v10 n6190

APPENDIX C

Data Sources: Auto-Course Selection



ISSN: 2473-4901
v10 n6190

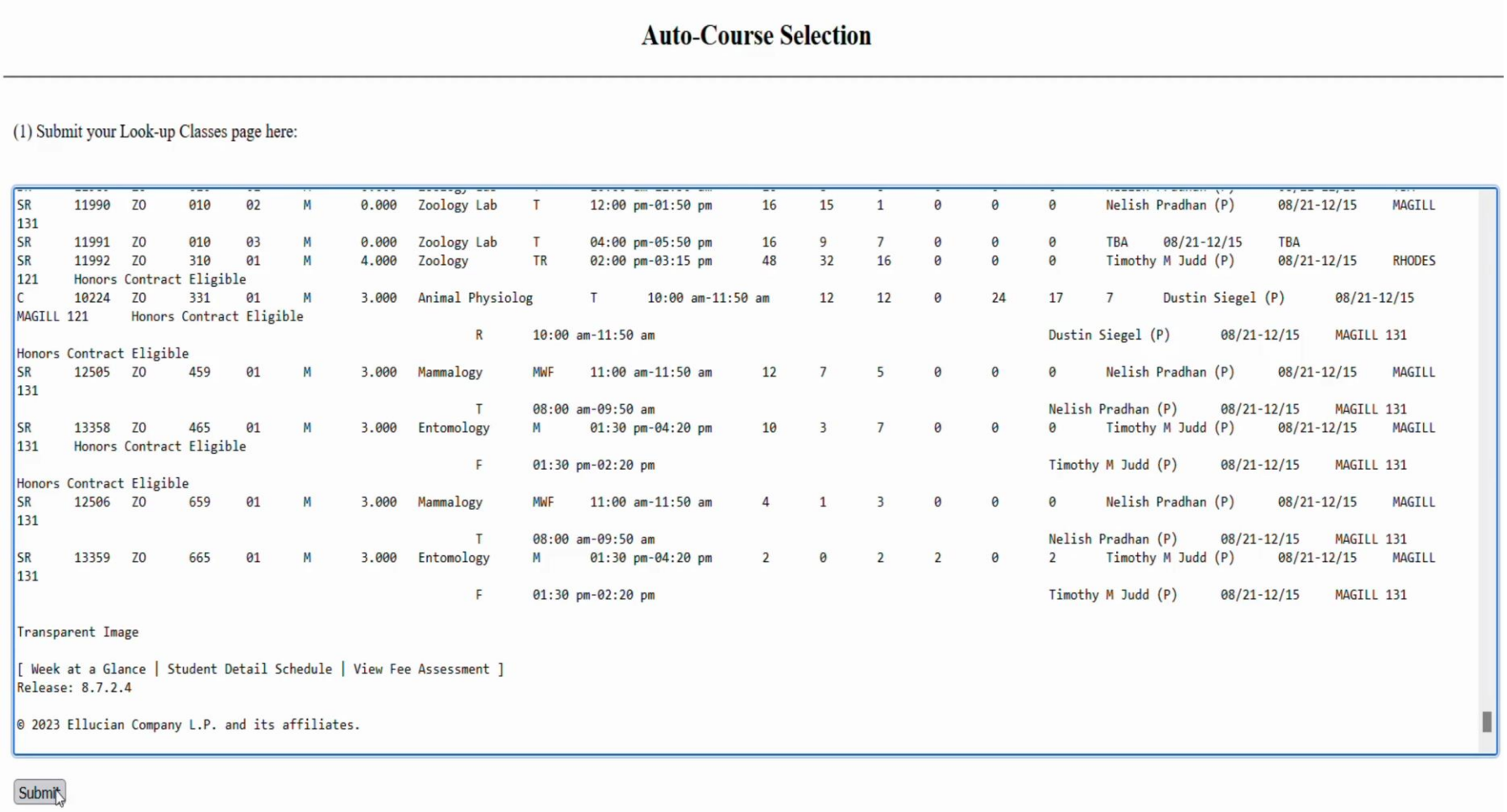
APPENDIX D

PHP call to fetch the page from the server

```
1 // Example fetch for bulletin data (php+js)
2 var bulletin_data = <?php echo json_encode(file_get_contents('https://
    semo.edu/student-support/academic-support/registrar/bulletin/courses
    /bltn_data.php'), JSON_HEX_TAG); ?>; //json_encode for escaping
    special characters
3
4 // If the tool is being deployed to semo.edu
5 // Replace the above line with the following code (js)
6 const url = "https://semo.edu/student-support/academic-support/
    registrar/bulletin/courses/bltn_data.php";
7 var bulletin_data = "";
8 fetch(url).then(function (response) { return response.text(); })
9           .then(function (html) { bulletin_data = html;});
```

APPENDIX E

Web-based Auto Course Selection System Interface



Auto-Course Selection

(1) Submit your Look-up Classes page here:

```
SR   11990  ZO   010  02  M  0.000  Zoology Lab      T    12:00 pm-01:50 pm   16  15  1  0  0  0  Nelish Pradhan (P)   08/21-12/15  MAGILL
131
SR   11991  ZO   010  03  M  0.000  Zoology Lab      T    04:00 pm-05:50 pm   16  9   7  0  0  0  TBA   08/21-12/15  TBA
SR   11992  ZO   310  01  M  4.000  Zoology          TR   02:00 pm-03:15 pm   48  32  16 0  0  0  Timothy M Judd (P)   08/21-12/15  RHODES
121  Honors Contract Eligible
C    10224  ZO   331  01  M  3.000  Animal Physiolog        T    10:00 am-11:50 am   12  12  0  24 17 7  Dustin Siegel (P)   08/21-12/15
MAGILL 121   Honors Contract Eligible
                                     R    10:00 am-11:50 am                                  Dustin Siegel (P)   08/21-12/15  MAGILL 131
Honors Contract Eligible
SR   12505  ZO   459  01  M  3.000  Mammalogy        MWF  11:00 am-11:50 am   12  7   5  0  0  0  Nelish Pradhan (P)   08/21-12/15  MAGILL
131
                                     T    08:00 am-09:50 am                                  Nelish Pradhan (P)   08/21-12/15  MAGILL 131
SR   13358  ZO   465  01  M  3.000  Entomology       M    01:30 pm-04:20 pm   10  3   7  0  0  0  Timothy M Judd (P)   08/21-12/15  MAGILL
131  Honors Contract Eligible
                                     F    01:30 pm-02:20 pm                                  Timothy M Judd (P)   08/21-12/15  MAGILL 131
Honors Contract Eligible
SR   12506  ZO   659  01  M  3.000  Mammalogy        MWF  11:00 am-11:50 am   4   1   3  0  0  0  Nelish Pradhan (P)   08/21-12/15  MAGILL
131
                                     T    08:00 am-09:50 am                                  Nelish Pradhan (P)   08/21-12/15  MAGILL 131
SR   13359  ZO   665  01  M  3.000  Entomology       M    01:30 pm-04:20 pm   2   0   2  2  0  2  Timothy M Judd (P)   08/21-12/15  MAGILL
131
                                     F    01:30 pm-02:20 pm                                  Timothy M Judd (P)   08/21-12/15  MAGILL 131

Transparent Image

[ Week at a Glance | Student Detail Schedule | View Fee Assessment ]
Release: 8.7.2.4

© 2023 Ellucian Company L.P. and its affiliates.
```

Submit

APPENDIX F

Automatically Generated Email Content by the Web-based Auto Course Selection System

Auto-Course Selection

Refresh

Hi

I am writing to you as your primary advisor with the aim of assisting you in selecting your courses for the upcoming semester, which is set to start next week.

Based on your study progress so far, I have carefully reviewed your completed and applied courses, which include:
MA115 FE200 SO102 US107 EN100 EN140 SC155 AO120 AO125 FN235 MA223 DA100 MU190 EL274 UI100 PY222 CS351 CS440 CS480 MA464 CS101 CS155 CS245 CS265 CS288 CY201 IS245 BA252

After careful consideration, I would like to suggest the following courses for your next semester:

CS380
CS445 : Senior Standing or Status
CY310
CY320
CY440

Please also select the elective, minor, and Gen-Ed courses to meet the full-enrollment requirement!

Please note that these courses have been chosen to help you achieve your academic goals and build a strong foundation for your future studies and career. I believe that these courses will provide you with the necessary skills and knowledge to excel in your academic pursuits.

Let me know if you agree with this suggestion, then I will forward the pin to you.

Southeast Missouri State University

ISSN: 2473-4901
v10 n6190

APPENDIX G

An Example of the Generated Email

Student 63: ——

n[redacted]s@semo.edu

S02[redacted]

Hi Matt,

I am writing to you as your primary advisor with the aim of assisting you in selecting your courses for the upcoming semester, which is set to start on 11/06. Based on the number of completed (rather than applied) credit hours, you can find your priority registration date here: https://semo.edu/student-support/academic-support/registrar/priority-registration-dates.html

I have carefully reviewed your completed and applied courses, which include:
{'CS101': 'A', 'CS155': 'A', 'CS245': 'A', 'CS265': 'A', 'CS288': 'A', 'CS351': 'B', 'CS380': 'A', 'CS440': 'A', 'CS445': 'IP', 'CS480': 'IP', 'CY201': 'A', 'CY310': 'IP', 'CY320': 'IP', 'CY440': 'IP', 'IS245': 'B', 'IS299': 'A', 'MA223': 'B', 'MA464': 'C', 'BA252': 'B', 'SI001': 'IP'}

After careful consideration, I would like to suggest the following courses for your next semester:
['CS499', 'CS533', 'CY410', 'CY420', 'CY450']

Please note that these courses have been chosen to help you achieve your academic goals and build a strong foundation for your future studies and career. I believe that these courses will provide you with the necessary skills and knowledge to excel in your academic pursuits.

Please remember to apply for graduation!

Simply reply to this email if you agree with this suggestion, then I will forward the pin to you.

If you think more discussion will be needed, feel free to reserve a timeslot form my office hours by the following link:
https://semo.starfishsolutions.com/starfish-ops/dl/instructor/serviceCatalog.html?bookmark=connection/54509/schedule

Regards,
George

Total 63 studetns!