Associate-Degree-Plan scheduling and Recommendation system for Virtual Academic Advisor system

Iswarya Hariharan

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

2021

Project Committee:
Erika Parsons, Chair
Marc Dupuis
Wooyoung Kim

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

Abstract

Associate-Degree-Plan scheduling and Recommendation system for Virtual Academic Advisor system

Iswarya Hariharan

Chair of the Supervisory Committee:
Erika Parsons
Computing and Software Systems

Community college students come from diverse backgrounds and experience levels. They begin their education path pursuing a degree in a major of their choice. Most students aim to get transferred to certain universities, an academic path that demands to fulfill specific requirements, which makes students eligible for the transfer. Academic advisors at community colleges help students in creating academic plans trying their best to incorporate students' interests, life constraints, and background. Being a heavily manual process that demands experience and familiarity with the process, there is a clear need to automate this process. The Virtual Academic Advisor (VAA) system aims to address the problem of automating academic plan creation for community colleges. The VAA is a research project paired with the development of an interactive software system that supports creating and displaying academic plans based on the needs and preferences of students. Work previously done by various students, focused on automated recommendation of core courses for targeted majors. However, no research or development has been done to incorporate selection of elective-course choices when generating an academic plan, nor a clear strategy on how to integrate elective-recommendation with the VAA system has been outlined. Incorporating electives opens up a whole new research aspect of automated scheduling. Furthermore, elective-course selection is crucial for scheduling associate degrees plans. Associate degrees

are offered by community colleges and students can earn such a degree before/without getting transferred to a university. In this thesis, we incorporate the logic and functionality of scheduling elective courses along with the core courses to generate associate degree schedules for the intended major and university of the student. We gather and collect the necessary data for the elective courses and test our scheduler for the associate degree schedules. This project also addresses the research and implementation necessary to generate alternative-schedule recommendations and its integration with the VAA system using APIs. This will assist students in exploring alternate academic paths.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express sincere appreciation to Prof. Wooyoung Kim, Prof. Marc Dupuis for accepting my request to be on the committee for this capstone project. I am also grateful for the support received from the Virtual Academic Advisor research team for having played a part in the completion of this project. I also thank Prof. Matthew Fuentes and the VAA research team for extending their support. Finally, I offer my very special thanks to Prof. Erika Parsons for accepting me into the research team and without whose support and guidance, this project would not have been possible.

DEDICATION

To my dear husband, Sabari without whose support I wouldn't have been able to achieve my goals. And to my family for providing support throughout this journey.

Chapter 1

INTRODUCTION

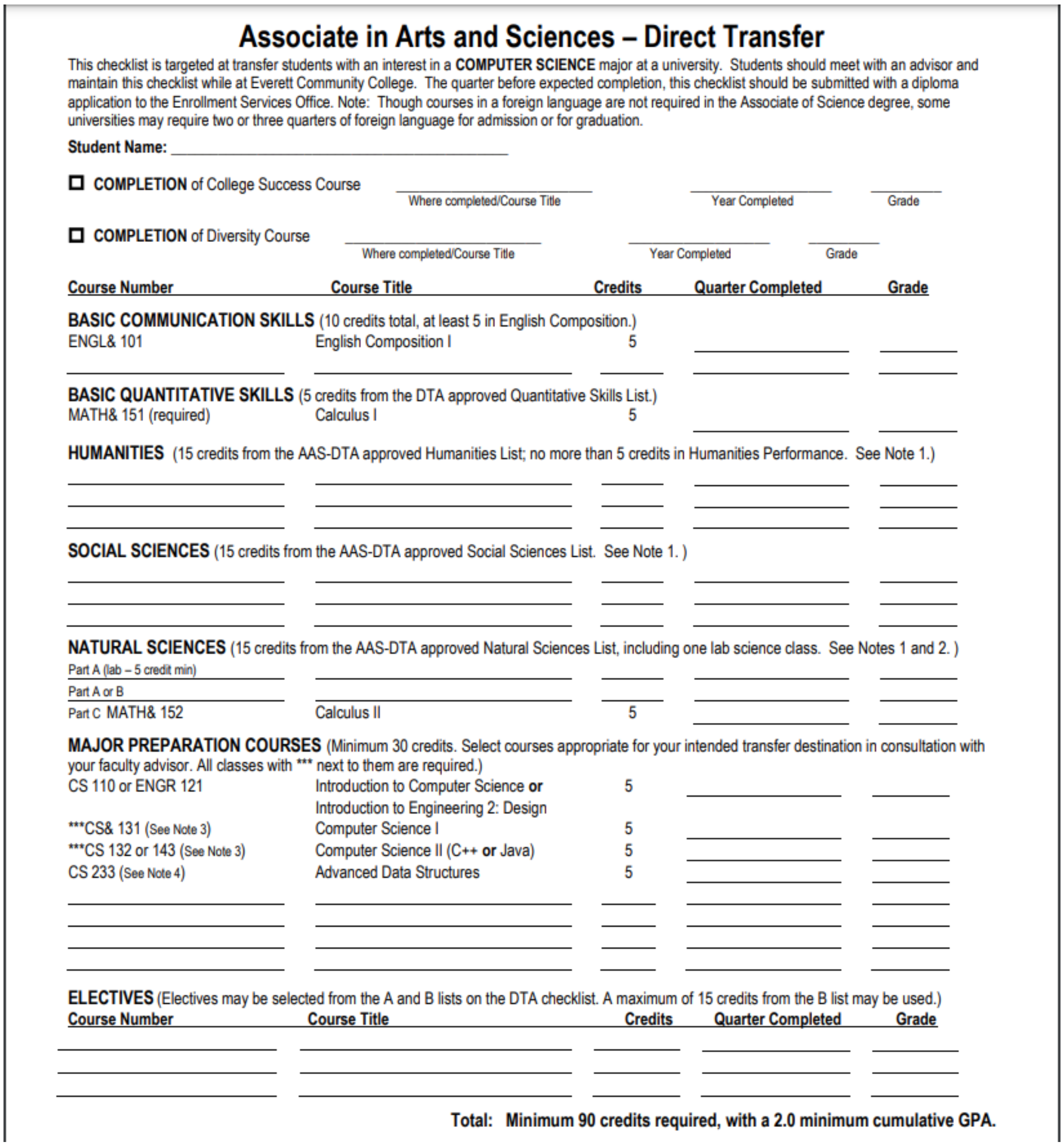
Academic planning is the process of planning and determining a student's path to achieve their academic goals by considering various options of achieving them, focusing on the target universities [51]. Academic planning is one of the essential steps to help students achieve their academic goals. This is especially true for students doing their higher education. Many students face challenges to decide their career paths when doing their higher education. Higher educational institutions have increased enrolling people from diverse backgrounds such as students restarting their education, first-generation students, new immigrants without much English language background, working students, etc. Many of these students aim to get transferred to universities or four-year institutions after doing their two-year coursework at the community college [24]. This process requires taking specific courses and number of credits to satisfy the requirements for transfer. Usually, these students in community colleges are confused with the wide range of options and ambiguity about the courses and programs offered in different schools and universities as there are various academic paths to achieve their goals [24][49]. Academic planning helps the students when they are confused with the various academic paths posed. Apart from this, various academic paths and university options confuses the students about the academic advising process [24]. Lack of sufficient resources to help them plan their degree to academic success is another challenge they face along with lack of cultural, academic, financial support, etc [55]. Understanding the academic and professional goals of the students are essential to pave path for their success, which thereby improves the quality of America Higher education[37]. Observations show that this success has been influenced by student-advisor interactions [55].

Academic advisors hold academic advising sessions with the students where the various course requirements, degree options, course selection process, transfer requirements and procedures, background interests of the students are discussed [55]. Advisors tailor academic plans for students taking into all their considerations, academic goals still ensuring the consistency of the plans [42][26]. Essentially the academic advisors bridge the gap between students' goals and the educational institutions requirements by scheduling courses such that the students' can complete them in a timely fashion [52]. A study [46] has found that students who underwent advising sessions has a higher rate of persistence and proceeded into second year of study in community colleges than the students who did not participate in the advising sessions. Attending advising sessions properly has also shown to increase the chances of a student finishing math courses and transferring to universities of their preference [50]. For the students of STEM (Science, Technology, Engineering and Mathematics), completing math and science courses are usually a major challenge. Without undergoing proper advising they are less motivated and inspired to take these courses.

Academic advising needs expertise and knowledge in the domain of different course options and academic paths to get transferred to a four-year institution. Due to this, in many community colleges, the faculty serve the job of academic advisors, which started in early 2000s. As classrooms serve as an important point of contact between the students and the college, the faculty play an important role for their educational development [40]. This idea was brought into effect with the faculty serving as advisors, which also serves the purpose beyond advising such as helping students figure out the rationale behind their education, comprehending the disconnected aspects of their plan and their past educational history, enhancing learning experiences by relating to the knowledge gained [52]. Despite serving these purposes, advisors interact with students during these sessions apart from their regular responsibilities. They find themselves overwhelmed with numerous questions and scenarios posed by the numerous students. The advisors perform a lot of work in the background to construct an academic plan for a particular student. During a session, their job doesn't end with gathering a student's background, limitations and goals. They end up spending extra hours every week to collect logistic information such as the course information, their prerequisites and offerings timings, transfer requirements etc. Even when some of the parameters remain common across some students, the advisor may have to work through the entire planning process every time to tailor a plan to a student. For instance, (see in Figure 1.1) the advisor and the student will have to fill out the courses to be considered in different quarters manually in a paper in Everett Community College (EvCC). This additional workload and time for the faculty could be used productively to engage with other students in advising sessions. Further, the students pose questions to the faculty that goes beyond creation of an academic plan. The faculty face challenging situations to choose between using the time to construct the plan or to have their questions answered [52]. During the advising sessions, most of the time is spent drafting academic plans or study plans, from the beginning using inadequate software tools. As every student, have different background, interests, life and career aspirations, restrictions and academic expectations, they need study plans tailored to their specific needs in order to follow a clear path to success. The faculty are overwhelmed with this process and the amount of additional work this involves [33].

Modern software can be used to help with the advising tasks where the academic plan creation can be automated, thereby improving the advising session quality as the focus of the conversation can be around other challenges that the students face, which would help in their retention and success [23]. Figure 1.2 shows where technology can be used to improve the efficiency of various services in the educational sector [23].

The Virtual Academic Advisor (VAA) is an end-to-end system envisioned by Dr. parsons to handle the automation of making study plans to help the academic advisors thereby reducing their time and effort. It is an interactive system to provide the community college students with academic study plans that tailor to their background, interests along with being flexible to their schedules. Its goal is to effectively improve the academic advising experience for the students and the advisors, thereby significantly reducing the amount of work that academic advisors need to do to help students develop academic plans. Though technology cannot automate the entire work during these academic sessions, this project aims to reduce the responsibilities on the advisors to just the logical part thereby, reducing



Associate in Arts and Sciences – Direct Transfer

This checklist is targeted at transfer students with an interest in a **COMPUTER SCIENCE** major at a university. Students should meet with an advisor and maintain this checklist while at Everett Community College. The quarter before expected completion, this checklist should be submitted with a diploma application to the Enrollment Services Office. Note: Though courses in a foreign language are not required in the Associate of Science degree, some universities may require two or three quarters of foreign language for admission or for graduation.

Student Name: ____________________

☐ **COMPLETION** of College Success Course ________ Where completed/Course Title ________ Year Completed ________ Grade

☐ **COMPLETION** of Diversity Course ________ Where completed/Course Title ________ Year Completed ________ Grade

Course Number	Course Title	Credits	Quarter Completed	Grade
BASIC COMMUNICATION SKILLS (10 credits total, at least 5 in English Composition.)				
ENGL& 101	English Composition I	5		
BASIC QUANTITATIVE SKILLS (5 credits from the DTA approved Quantitative Skills List.)				
MATH& 151 (required)	Calculus I	5		
HUMANITIES (15 credits from the AAS-DTA approved Humanities List; no more than 5 credits in Humanities Performance. See Note 1.)				
SOCIAL SCIENCES (15 credits from the AAS-DTA approved Social Sciences List. See Note 1.)				
NATURAL SCIENCES (15 credits from the AAS-DTA approved Natural Sciences List, including one lab science class. See Notes 1 and 2.)				
Part A (lab – 5 credit min)				
Part A or B				
Part C MATH& 152	Calculus II	5		
MAJOR PREPARATION COURSES (Minimum 30 credits. Select courses appropriate for your intended transfer destination in consultation with your faculty advisor. All classes with *** next to them are required.)				
CS 110 or ENGR 121	Introduction to Computer Science **or** Introduction to Engineering 2: Design	5		
***CS& 131 (See Note 3)	Computer Science I	5		
***CS 132 or 143 (See Note 3)	Computer Science II (C++ **or** Java)	5		
CS 233 (See Note 4)	Advanced Data Structures	5		
ELECTIVES (Electives may be selected from the A and B lists on the DTA checklist. A maximum of 15 credits from the B list may be used.)				

Total: Minimum 90 credits required, with a 2.0 minimum cumulative GPA.

Figure 1.1: Engineering transfer guide from Everett Community College given to students and advisors to keep track of courses every quarter

their overall responsibilities and making their job easier. VAA aims to reduce the amount of redundant work, thereby saving time during advising sessions, which would involve recommending study plans already available. As shown in Figure 1.2, this system will also try to reduce human-associated errors and in the long-term motivate colleges to adopt new

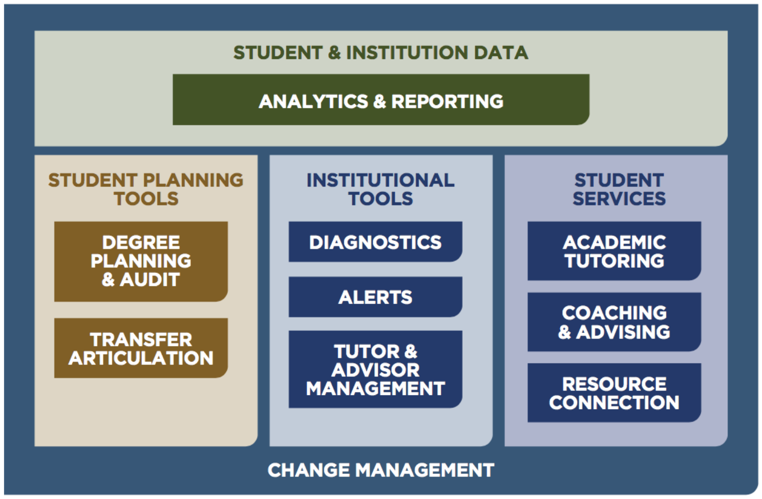


Figure 1.2: The 9 distinct categories where technology can be used to improve the efficiency of the services provided to the student [24].

technologies in neglected areas like the student services [20][53][31].

Further studies have observed that students who just complete coursework at the community colleges to get transferred to four-year institutions are less likely to get a bachelor's degree with successful completion compared to the those who obtain AD [36][29]. Associate degrees(AD) are degrees awarded by community colleges where traditional level college courses are offered to make students get prepared for the four-year bachelor's degree [1]. Community colleges also offer vocationally oriented ADs that prepare students for their careers. [36]. With more financial need for pursuing bachelor programs in the United States, young people have preferred ADs in community colleges even before 2000, which serves as a low-cost platform for the students. Many community colleges have transfer agreements with the four-year baccalaureate universities, where these institutions accept community college credits towards some of their bachelor's requirements [36]. Eventually, students who have ADs take lesser credits at the four-year institutions compared to their counterparts who don't have one. For instance, in EvCC, this is called Direct Transfer Agreement (DTA) where they have Associate degree(AD) in Business DTA with Western Washington University [1]. In addition to this, the Nationwide Commission on the Future of Community Colleges recommended community colleges to help build better communities by creating partnerships with local employers and businesses along with making facilities available for workforce training. This alliances with the local companies have served the local businesses and many corporations in the US who have relied on the locally trained workforce for a long time. This has also eased the financial burden that community college bears as their funds and standard improves when more students graduate with ADs [1]. For instance, the Lawson State Community College in Birmingham, Alabama have allied with Microsoft Corporation offering ADs and certificates accredited by the Association of Collegiate Business Schools and Programs [36]. Ever since, community colleges have started to enroll and make education more available to the students from varied backgrounds, immigrants, minorities, ethnicities, etc. the ADs have boosted confidence and a sense of satisfaction to prove their identities in the society. They have been able to pursue paid jobs with the AD even if they decided to pursue bachelors. This is evident the study found among the Hispanic students during the 1990s decade that their undergraduate enrollment in higher education increased by 98% and they started off with community colleges to enter into higher education [27]. The largest group in this study indicated that 68% obtained an AD and there was 130 percent increase in the proportion of ADs awarded to the Hispanics [36][27]. Whether a student pursues a bachelor program after obtaining an AD or not, doesn't change the number of courses or credits needed to complete the bachelor program. However, a student getting transferred with an AD would already be ahead in the coursework compared to their counterparts, which thereby also reduces the financial aid needed by the students as community colleges are largely low-cost platforms [21]. With ADs providing benefits for both students and the community college, it would be better to generate schedules during advising sessions to make the students aware of their benefits.

The scope of work described in this thesis targets the Everett Community College courses and the transfer data to Washington State's universities.

1.1 Problem Statement

As discusses in the previous section, with the number of students increasing in community colleges especially in STEM disciplines, the unnecessary workload on the faculty have also increased. With the community colleges providing low-cost platforms for students from varied backgrounds to pursue their higher education, it is essential to provide them advising sessions with proper guidance on options and pathways available. The students should also be made aware of the AD benefits when getting transferred to universities compared to not considering that option. This can be possible if software technology can efficiently help in generating academic plans for the students, thereby allowing the faculty to focus on having wholesome conversation with the students to listen to their challenges and understand their demands. This also reduces the workload on the advisors and the repetitive work for the overwhelmed faculty. Academic advising is an area that has been neglected by software development due to which the options available today for advising are outdated providing options that are hard coded and inflexible. It is necessary to build a software system that clearly provides the different options for students and advisors, which are very flexible adhering to their varying preferences. Recommendations of previously created academic plans is a part of a larger system, which reduces the redundant work for the system as well as the faculty. Usage of Machine learning(ML) for the recommendations improves the overall academic planning experience.

Terminology. This report includes a list of terminology which is widely used throughout the report. Table 1.1 lists the terminologies and their context in this project.

1.2 Existing System and Architecture

The existing VAA system comprises of the modules namely, the User Interface (VAA UI), VAA API, and the VAA database. VAA UI gets the preferences from the user and thereby displays the generated schedule. It is hosted on a local dev box. The UI interacts with the

Table 1.1: Terminology and context used in the report.

Case	Context in the project
Preference set	List of preferences of the student such as target major, target school/university for transfer, the maximum number of quarters of study, maximum courses per quarter, evening classes preference, summer quarter study preference
Schedule	List of quarters and courses every quarter to satisfy the preferences of student- major and target school for a transfer preference
Prerequisite	Course to be completed before a particular course. Example: Course English-I may be a prerequisite for course English-II
Prerequisite network	List of prerequisite courses represented in the form of a network to satisfy the major and target school preference of students
Course, Job	In the flow shop algorithm, a unit of work is termed as a job. In the job-shop scheduling algorithm used in the VAA system, a job is a course
Quarter, Machine, Job	A machine is an object that can perform some work (ie. units of work-job). In the job-shop scheduling algorithm, a machine is a quarter that can have courses scheduled, thereby the jobs are done in that machine.
EvCC	Everett Community College
VAA	Virtual Academic Advisor system
API	Application Programming Interface
Associates, Associates degree(AD)	A degree offered by a community college, which is associated with a major after completing the courses listed in its requirements.

scheduler using the Rest APIs. The APIs are hosted on Azure web apps. The overall job of the APIs is to save the preferences of the user sent from the UI into the database and initiate the process of creating the schedules. The created schedules are returned as JSON to the UI for the display. The scheduler generates a prerequisite network of all the courses, which acts as the critical part of the system to schedule the courses needed to satisfy the user's preferences. The prerequisite network is explained in detail later in section 3.3. The scheduler uses the Genetic algorithm to generate the academic plan for the students based on their preferences entered [18]. The core academic courses for transfer to the desired major and school are considered in the scheduling algorithm, which then generates the academic plans. This system does not contain any provision or implementation of scheduling elective courses for generating the schedules. The deterministic recommendation engine has been tested with a limited dataset. This system doesn't integrate the recommendation module with the system to recommend alternate plans to the advisors. The VAA database stores the data in the relational form in the MSSQL server hosted in the HostTek [54] hosting account. Figure. 1.1 shows the overall architecture of the existing system as described previously. The UI is used for getting the user preferences and to display the generated schedule. It is deployed in the local development box. The VAA API is a web application deployed in Azure web apps used for leveraging the data between the UI and the backend system. The backend comprises a scheduler module with the scheduling algorithms and the prerequisite network builder. It is deployed in Azure web apps. The database is hosted in a third-party server in HostTek hosting account. It is a relational database that stores the user preferences, schedules, courses, and various schools' data in the form of tables in the MSSQL server application. The overall workflow of the system [52] is as shown in Figure 2.1. It significantly reduces the amount of redundant work done by the advisors for creating the academic plans. The algorithms used in the backend implementation tries to deterministically generate study plans based on the student's input preferences along with building prerequisite networks for all the involved courses. With all the interacting phases in the VAA system, it aims to refine the recommendations of plans using a feedback-loop approach. The user preferences entered

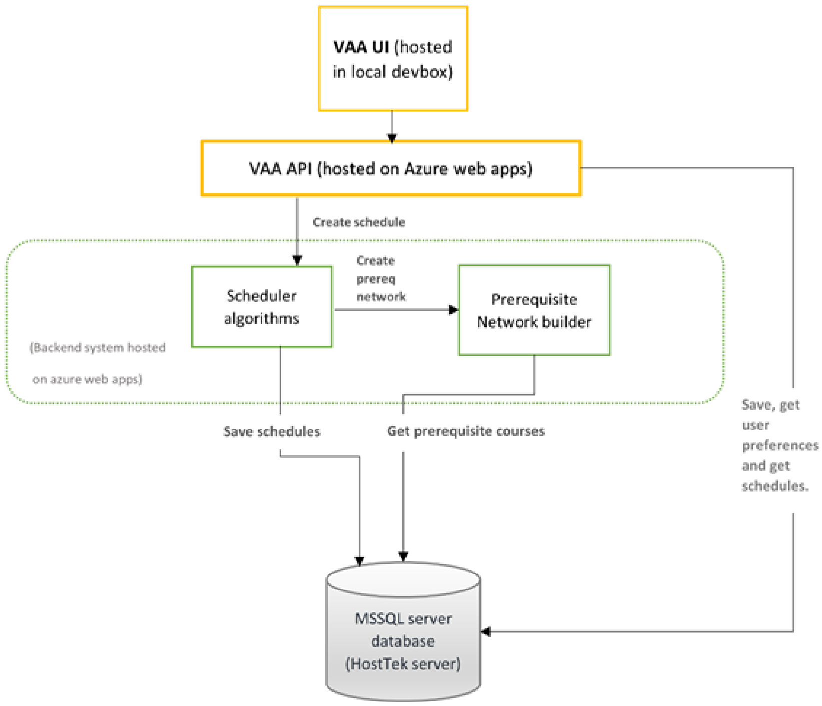


Figure 1.3: The overall architecture of the existing VAA system for scheduler and API

are utilized by all the phases to generate the study plans. Ratings for the study plans are used to improve the recommendation engine framework [52][18].

Limitations of the existing system The limitations of the previous system include the following: The scheduling algorithms used in the previous system do not consider electives for generating schedules. They just consider the required courses to complete to get transferred to the target university.

- Elective courses scheduling: The existing scheduling algorithms don't have provisions to include elective choices based on the students' preferences for generating academic plans.

- Lack of data: Even though the database has the provision to store the electives data,

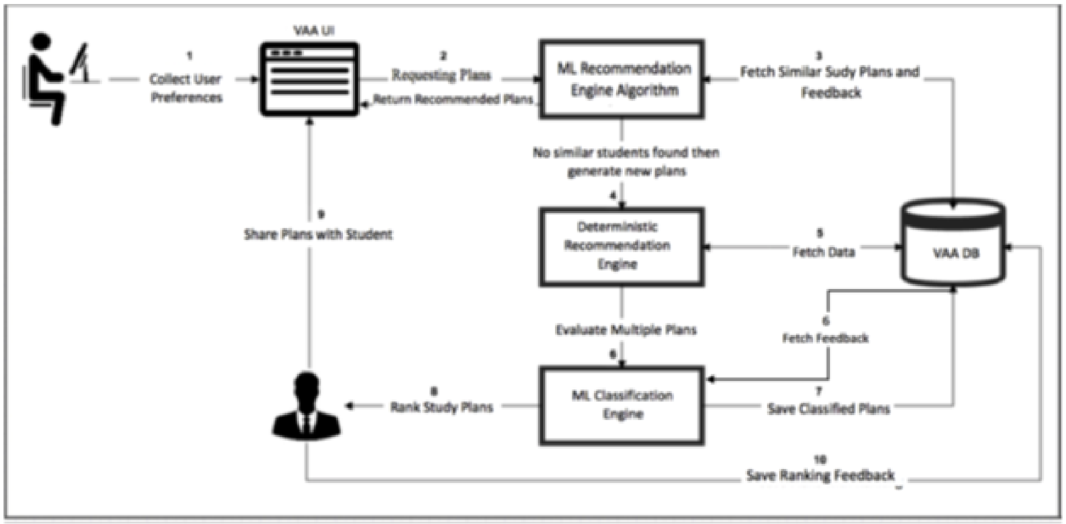


Figure 1.4: The workflow of VAA system [52]

there is very limited data for the course timings of these electives. Out of the various elective courses available, the course timings data is partially available only for certain elective courses. There is also no data present for three AD majors in their respective tables, which is needed for generating AD plans.

- Recommendation system: The existing system doesn't have the recommendation engine integrated into the VAA system to recommend suitable alternate schedules based on appropriate parameters. Previous work on the recommendation framework doesn't include appropriate research on determining the metrics for implementation using the existing dataset [33][18].

- SDLC: The database hosted in the HostTek account needs proper maintenance and monitoring. The APIs hosted in the Azure web account need proper integration with the backend and database for the schedule generation functionality to be used by the UI efficiently. The existing system doesn't have the necessary documentation for regular monitoring of the database on the third-party hosting server [54].

1.3 Proposed Solution

The VAA system targets students at community colleges who aim to transfer to four-year institutions after completing two-year at the community college. It also aims to encourage students to complete AD along with ensuring their transfer based on their preferences, limitations, and background. All this is automated to ensure the workload on the advisors is reduced. The proposed solution consists of improving the VAA system with the addition of new features and functionality. The goals of the work discussed in this paper is discussed below.

Schedule generation for AD majors. The existing system uses the students' preferences to generate academic plans or schedules, which is dependent on the core courses to complete for getting transferred to the desired school and major. AD requirements specified by a community college involves specific courses under every category to be completed to obtain the degree. Some of these requirements get satisfied by core courses based on the student's preferences. Our primary goal focuses on generating plans that can enable a path for the students to get transferred to the universities of their choice, along with obtaining an AD in the related major. In the proposed work, we have incorporated the elective courses along with the core courses to generate plans providing an option for students to consider pursuing AD along with the transfer requirements.

Adding elective courses data. The existing system contains data only for certain number of student preferences of major and school combinations. These are specifically core courses, which have their offered timings data available in the database. However, the system doesn't have any provisions for considering elective courses for scheduling. Hence, the elective courses data and their offering timings throughout the year is not available in the database. In our proposed work, as there are several elective course options available, we have collected data for some of their actual offerings timings and entered into the database. Due to the scope of this thesis, we have generated elective courses timings based on the insights from the academic advisors on the course box slots and entered into the database. Course box slots are standard timings during which any course in a community college is offered. This data is necessary for generating AD schedules, which considers elective courses along with the core courses for transfer.

Adding AD requirements data. In the previous work [35], the different requirements and elective course options to satisfy the requirements were collected and entered into the database only for a few limited AD majors. This limits the pool of options to test and generate plans. In our proposed work, we have collected data for a few other AD majors and entered them into the database. This opens different options of AD majors to choose based on the closest match of the preferences of the students for desired school and major to get transferred.

Recommendation Module Improvement and Integration. In the previous work to explore ML strategies for the recommendation engine in the VAA system [33][18], only limited dataset was used. Due to the inadequate availability of study plans, the k-means clustering approach used for the ML model did not effectively compute the optimal k-value to cluster all the dataset. This is essential to ensure that the recommendations are close to the preferences of the student considered. Also, the recommendations were not integrated into the VAA system. The proposed work, firstly analyses the dataset for the optimal k-value when performing k-means clustering by varying the value of k. The selected k-value is used to perform k-means clustering and the generated model is saved for future recommendations request. When a request for recommendations for a particular schedule is obtained, two nearest plans from the model is recommended. These recommendations are integrated into the VAA system using API.

1.4 Scope

The scope of the work in this thesis targets Everett Community College (EvCC) focusing on the courses related to STEM and transfer data to Washington state universities. The outline of thw work includes the following:

- Dataset is composed of preferences and data provided by Everett Community College (EvCC) which will be used for generating plans using scheduling algorithms from previous work. Relevant features using the weak labelling engine [52][18] will be used for rating the plans.

- Data collected from EvCC academic advisors for the AD specifications and elective courses will be used in combination with the core courses data in the system.

- Current existing implementations will be extended and improved to make recommendations of relevant alternate plans. These results will be integrated into the VAA system.

- Strategies to determine the optimal value of 'K' in K-means clustering will be explored and analyzed on the existing dataset.

- Academic plans for AD majors will be generated, which also focuses on enabling students to get transferred to school and major of their preference. These results will also be integrated into the VAA system using API.

- The Software development Lifecycle process (SDLC) of the existing VAA system will also be maintained and monitored regularly by making certain improvements.

1.5 Outline

The rest of this thesis is organized as follows. Chapter 2 focuses on the previous related works to the problem statement along with the previously implemented functionality in the system. Chapter 3 addresses the methodology of the proposed work along with the SDLC process used in the system. Chapter 4 focuses on the implementation of the proposed work and its results. Chapter 6 focuses on the overall implementation discussion, limitations, and the possible future work.

Chapter 2
RELATED WORK

In this chapter we discuss about the existing work related to the academic advising that supports our research and thesis. We present the current state of academic advising along with the software systems that support advising. The systems following different approaches for the recommendation system is also discussed. The previous work done in the VAA system that supports our work is also discussed in detail. Finally, we discuss how our work differs from the previous work done in VAA.

2.1 Current State of Academic Advising and VAA system

The non-technical efforts undertaken we see that in 2017, Washington state have begun academic advising efforts online using Guided Pathways [11], which clarifies and addresses four pillars of students concerns. They include clarifying the academic paths, helping the students get on to the suitable path, helping the students stay on the chosen path and ensure that students keep leaning throughout [33]. Currently, technology is being used to support this effort and also has been identified to best support this effort in the long run. In [24]. The importance of advising in higher education is discussed clearly. In their research, they have discussed that close to a billion dollars are spent on the services pertaining to student success and retention every year, but the graduation rates of the students on the expected time hasn't improved in the last decade. Their studies show that less than 20% of the total students finish two-year degrees within three years of time. It also discusses the burden advisors carry to attend several students, which leads to inefficient advising with human errors. This clearly shows the need for a better and more precise advising system where technology can solve this problem.

The information and knowledge available for generating college-level academic plans for students are very limited. From a technical viewpoint, there isn't much effort in this area for tools developed for educational purposes specifically. No proper software applicants are in use for educational settings [33]. Due to this, many schools have developed internal tools for their specific use in an ad-hoc manner, which could also be extended to other schools to solve similar problems. However, these don't have capacity to handle flexibility with the academic plans. For instance, Starfish tool from Hobsons [9] has hardcoded information and academic paths to make a decision, which also have simple and deterministic approaches. EvCC currently uses tools [8] as simple as notepad webforms to create academic plans and also provides least scheduling functionality [33]. Figure 2.1 shows an example of a schedule used by EvCC in their tool's UI.

In section 1.2, we point out that there is an increase in the number of tools that are being developed to address the deficiencies in the current advising process in the US. Such tools address different academic advising areas such as caseload management, performance management, measurement [52][24]. For instance, Salesforce advisor link used for caseload management by the advisors streamlines the advising operations and maintains student records [39]. Aviso retention services [25] identifies the students at risk by analyzing their performance tries to improve their retention rate. Signal Vine Blended [10] has solutions for direct interactions with the students and also provides them with notifications when configured. Edunav's optimize [14] predicts the demand for the courses based on the aggregations of the degree plans made. Loud Cloud [24] also provides user-centric solutions considering the student's financial ability. Even though these many tools deliver certain functionality related to the students needs and advising, none of them actually address the fundamental problem of advising. The VAA system addresses this problem by being interactive with the students by taking their preferences and thereby, generating the schedules. These schedules are generated deterministically. Dynamically generated plans help students to focus on courses step by step, which also allows advisors to focus on interactions with students on the challenges they face and improve their learning and retention rate [46][48]. The VAA system lacks extensive ML-based approaches to add variety to the data and help reduce the repet-

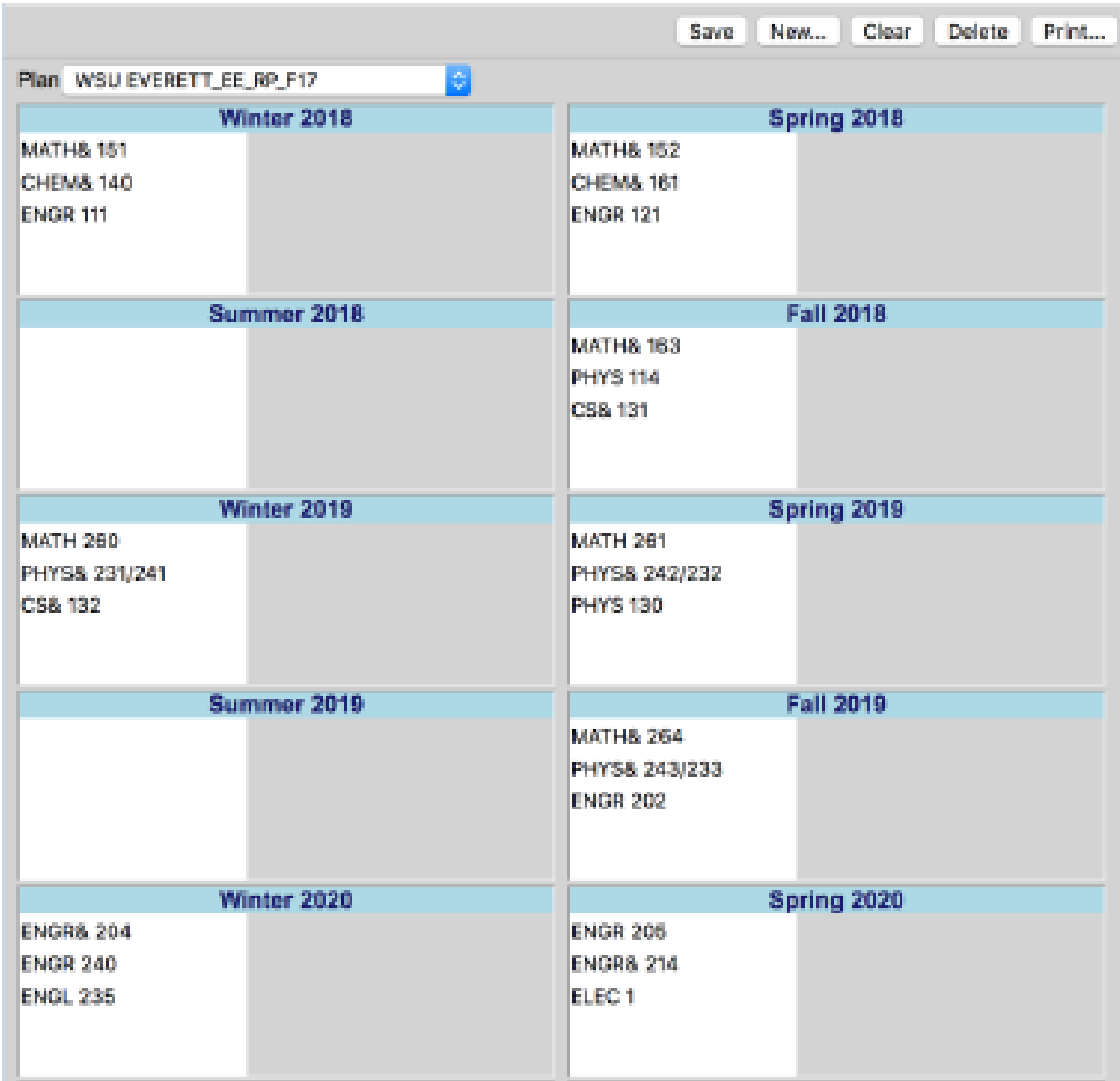


Figure 2.1: Study plan example for a student who wants to get transferred to Washington State University in Electrical Engineering as generated by the advisor

itive work to generate schedules deterministically for preferences set by the students every time. Our proposed approach is to upgrade and integrate the ML-based recommendation system with the VAA to recommend similar academic plans.

Collaborative filtering based Recommendation systems

Collaborative filtering is widely used in recommendation systems due to its simplicity in the logic to compute the recommendations using the distances and is available in different areas including e-commerce, entertainment, technology, knowledge management, etc. This is effectively used in applications like movie recommendations, which are generated for users based on global user similarity [38]. GroupLens generates news articles recommendations which is also based on collaborative filtering [44]. In [47], personalization of tag-based searches have been done using collaborative filtering. Even though it is widely used in real-time and also researched, education sector has not seen any system effectively using it especially in academic advising. Clustering is an algorithm that is based on collaborative filtering, which has been proposed to be used in our work and integrated into the VAA for receiving effective recommendations.

2.2 Previous work in VAA system

The previous VAA system has effective implementations of schedule/academic plans generation based on the preferences of the student including the intended school and major for transfer. The scheduling algorithms generate the plans. The plans are machine-rated based on the criteria and heuristics collected from advisors[52]. This reduces the amount of work required by the advisors to rate these plans. As discussed in chapter 1, advisors also welcome students to consider AD plans related to their preferred major before getting transferred to their school of choice. The specifications data for these degrees have also been stored in the database in the VAA. This previous work effectively used in the system have been discussed in detail in this section. We will be using this work to perform our research and achieve the goal of our work. The primary goal of this research is to implement the functionality to cater to students to obtain a degree called the AD provided by community colleges even before they get transferred which is discussed in this section.

2.2.1 Associate Degree

ADs have specific requirements to meet with a variety of courses catering to satisfy these requirements. Each requirement has a minimum and a maximum number of credits to satisfy to complete the AD. In the process of academic career planning with the academic advisor, once a student decides the target school and major, the scheduling algorithms in the existing VAA system pulls the required core courses to get transferred/enter the intended major and school from the database. Some of these core courses may satisfy certain requirements for the AD but not all of them. The rest of the courses to satisfy these requirements are treated as electives by the scheduling algorithms, which can vary according to the specifications of the degree and the community college.

Figure 2.2 is an example of AD requirements and their specifications. The AD requirements are the specific headings with credit requirements such as Basic Communication skills (10 credits), Basic quantitative skills (10 credits). The possible elective courses that satisfy the requirements are listed along with the credits to satisfy each of them just next to their title. The possible electives to satisfy the requirements along with their description are stored in tables with appropriate primary key and foreign key relationships as detailed in the next section 2.2.1. Consider Figure 2.3 shows a particular requirement: "Basic Communication Skills" from Figure 2.2. Here, the first entry in this requirement provides the option of the courses ENGL 101 or ENGL 101D (the prefix ENGL stands for English in course names) to complete the English composition-I elective requirement of five credits. Both these two courses are offered for five credits and completing either of them would satisfy the elective requirement.

Relational database structure for AD requirements. Previously, information was gathered for two of the AD majors namely, Associate in Business DTA and Associate in Arts & Sciences and their requirements and electives data were entered in the database [41]. The relational structure for these tables is shown in Figure 2.4 below. Explicit entries for all AD majors were made in the Major table to identify the AD. The Requirement table

ASSOCIATE IN BUSINESS (DTA) DEGREE **REQUIREMENTS** ☐ **Completion of Diversity Requirement**

***Must earn a C grade (2.0) or better in all required courses. Courses may be subject to prerequisites.**

Course	Course Title	Credits	Grade	Quarter	Year
BASIC COMMUNICATION SKILLS (10 credits)					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
*ENGL& 101 or ENGL& 101D	English Composition I	5			
*ENGL& 102, ENGL& 102D or CMST& 220	Composition II or Public Speaking (CMST& 220 required at EWU)	5			
BASIC QUANTITATIVE SKILLS (10 credits)					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
*MATH 138 or &141, &142, or &144 or &148 or &151 (or higher)	Applied Algebra/Precalculus Business Calculus Calculus I	5			
*MATH& 148 or &151 or higher	Business Calculus Calculus I	5			
HUMANITIES (15 credits with no more than 10 credits from any one discipline on the AAS DTA Humanities distribution list.) No more than 5 credits of foreign language and performance arts credits can be listed. Two quarters at EvCC or two years in high school of the same world language is required for admission to all UW campuses. Students interested in an international business major should consult with the specific transfer institution regarding foreign language requirements.**)**					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
		5			
		5			
		5			
SOCIAL SCIENCE (15 credits; 10 credits in economics; **5 credits** other than economics from the AAS DTA Social Science distribution list. BUS &101 recommended as a social science distribution course.**)**					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
*ECON& 201	Micro Economics	5			
*ECON& 202	Macro Economics	5			
BUS& 101 (recommended)	Introduction to Business	5			
NATURAL SCIENCE (15 credits; 5 credits in statistics; 5 credits each from the AAS DTA Natural Science distribution lists Part A and Part B. No more than 10 credits from any one discipline on the AAS DTA Natural Science distribution list.**)**					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
*MATH& 146	Introduction to Statistics	5			
Part A (lab course)		5			
Part A or Part B		5			
REQUIRED BUSINESS-SPECIFIC ELECTIVES (20 credits)					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
*ACCT& 201	Principles of Accounting I	5			
*ACCT& 202	Principles of Accounting II	5			
*ACCT& 203	Principles of Accounting III	5			
*BUS& 201	Business Law	5			
OTHER ELECTIVE (5 credits; course numbered 100 or above. WSU, Gonzaga, PLU, SPU, and WWU have additional requirements for admission that may be met as elective credit. See Notes below.**)**					
Course	**Course Title**	**Credits**	**Grade**	**Quarter**	**Year**
		5			

- To earn a degree, the program must be completed with a cumulative GPA of 2.0 (C) or better.
- Gonzaga requires a course equivalent to its BMIS 235, Management Information Systems.
- PLU requires a course equivalent to its Computer Applications CSCE 120 or MOS certification; CL 101 may suffice to fulfill this requirement.
- SPU requires a course equivalent to its BUS 1700 or MOS certification (MOS 77-420); BT 242 may suffice to fulfill this requirement.
- WWU requires a course equivalent to its MIS 220 Introduction to Computer Systems; CL 101 may suffice to fulfill this requirement. The WWU Manufacturing and Supply Chain Management program requires additional coursework, some of which may also be taken as elective credit at EvCC. Management program web site is www.wwu.edu/node/731/.
- WSU requires either MIS 250 or EvCC's BUS 250.

Figure 2.2: Associates in Business degree specifications in EvCC [2]

holds the description of each of the requirements for the AD major identified by the MajorID field, which is a foreign key to the Major table. The electives for each requirement are listed in the Elective table with ElectiveID as the primary key. The RequirementID field is

BASIC COMMUNICATION SKILLS (10 credits)					
Course	Course Title	Credits	Grade	Quarter	Year
*ENGL& 101 or ENGL& 101D	English Composition I	5			
*ENGL& 102, ENGL& 102D or CMST& 220	Composition II or Public Speaking (CMST& 220 required at EWU)	5			

Figure 2.3: Elective requirements for Basic Communication skills requirement in Associates in Business degree DTA [2]

the foreign key in this Elective table that identifies the electives for each requirement. According to Figure 2.3, "English Composition-I" would be an entry in the Elective table for the RequirementID that identifies the entry, "Basic Communication skills" in Requirement table. The CourseID of courses, ENGL101 and ENGL101D (the prefix ENGL stands for English in course names) would be entries in the ElectiveClass table, which are identified by their respective ElectiveClassID. Their ElectiveClassID are listed for the ElectiveID that identifies "English Composition-I" in ElectiveToClass table. In ElectiveToClass table, both ElectiveID and ElectiveClassID are foreign keys. Hence, this means that courses, ENGL101 and ENGL101D are the elective course options that can satisfy the elective requirement of 'English Composition-I' for the Associate in Business degree requirement of "Basic Communication skills". The existing tables for the electives don't have entries for all the elective courses that satisfy all the requirements for the AD majors. The previous system just utilized the required core courses to schedule for generating academic plans for students with an intended major and school. The elective courses were not utilized and most of their course timings are not present in the database as well. One of the primary goals of this capstone project is to research and gather information about the elective courses and their timings from EvCC advisors and implement the functionality to treat courses as electives and choose from elective courses pool to generate AD academic plans.

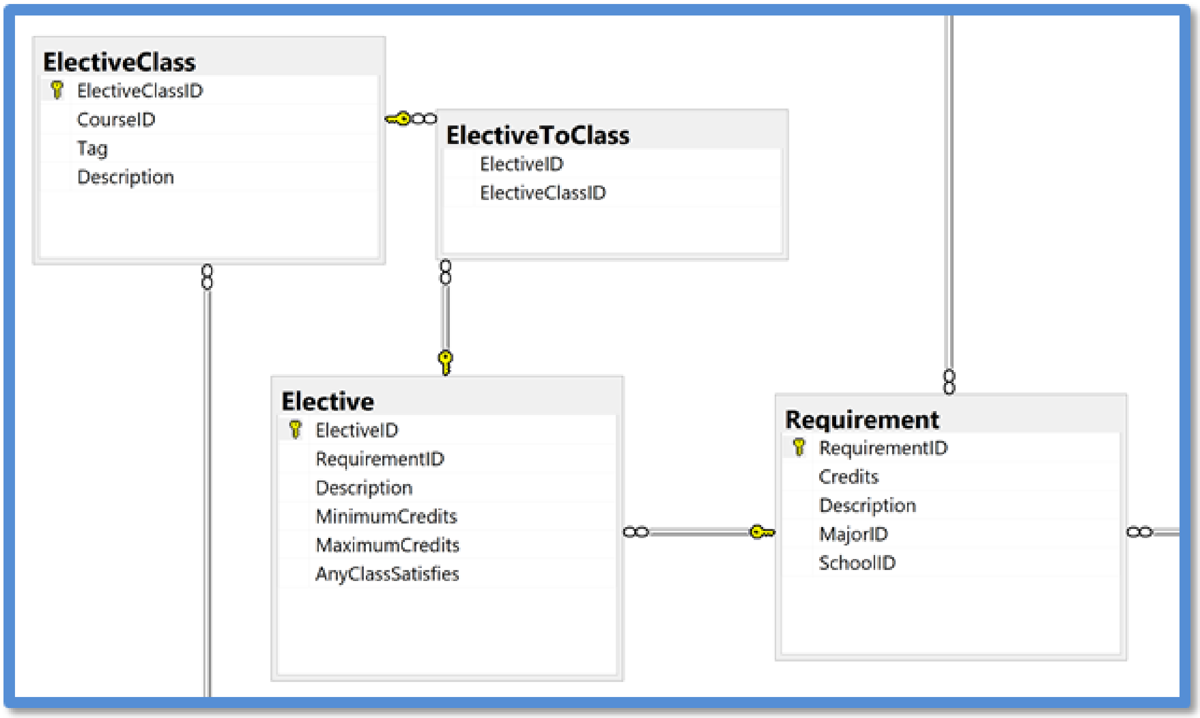


Figure 2.4: Relational structure of the requirement and electives table for the associate degree [41]

2.2.2 Schedule Generation

The schedule generation is the core of the VAA system. The scheduling algorithms used can generate schedules meeting the student's set of preferences targeting their dream school of choice and major. The schedules in the system are generated purely based on the preferences of the student and possible course availability and timings in their current school where they are pursuing education. The recommendation system also relies on schedule generation for its ability to generate schedules. The existing VAA system allows the preferences such as the choice of major and school to be entered through the User Interface.

Preference Set. Both the scheduling algorithms and recommendation engine function are based on the set of preferences provided by the student, which is thereby used for schedule generation. These preferences are treated as constraints by the scheduler to generate the optimal schedule. The preferences provided by the student are treated as constraints when generating the schedules. Some constraints such as number of quarters, number of credits, choice of class timings in a day (evening class preference) can be changed from the values provided by the student to generate the most optimal schedule that satisfies all the preferences. Some constraints cannot be modified when generating the schedule as the core of the scheduling system works to satisfy these constraints. Such constraints are called hard constraints, which include the choice of major and target school to get transferred. The list of preferences obtained from a student to generate a schedule is displayed in the Figure 2.5. Figure 2.5 shows a sample of the preferences obtained from a student to generate an academic schedule plan on the VAA UI.

Scheduling algorithms. In the existing system, there are four scheduling algorithms implemented, which have their own advantages and disadvantages when generating schedules. All of them rely on the prerequisite network to ensure that precedence courses are first scheduled before scheduling the courses required to take for the major and school, given the student's preferences. A prerequisite network is a graph data structure used to represent the courses and their prerequisites, which is also discussed in detail later in chapter 3. The following sections describe the various scheduling algorithms implemented in the existing system and the limitations of each of them. Finally, we also determine which algorithm would suit our existing problem.

Job Shop scheduling algorithm. The Job Shop scheduling algorithm uses the prerequisite network and treats every quarter as a machine and uses a depth-first search to schedule the courses recursively in a free machine focusing on the shortest path to finish a prerequisite. It is a combinatorial optimization problem that assigns "jobs" to available "machines" that can perform the jobs [22]. However, this algorithm doesn't take into account the courses that don't have dependencies on other courses. It schedules the non-dependent course only

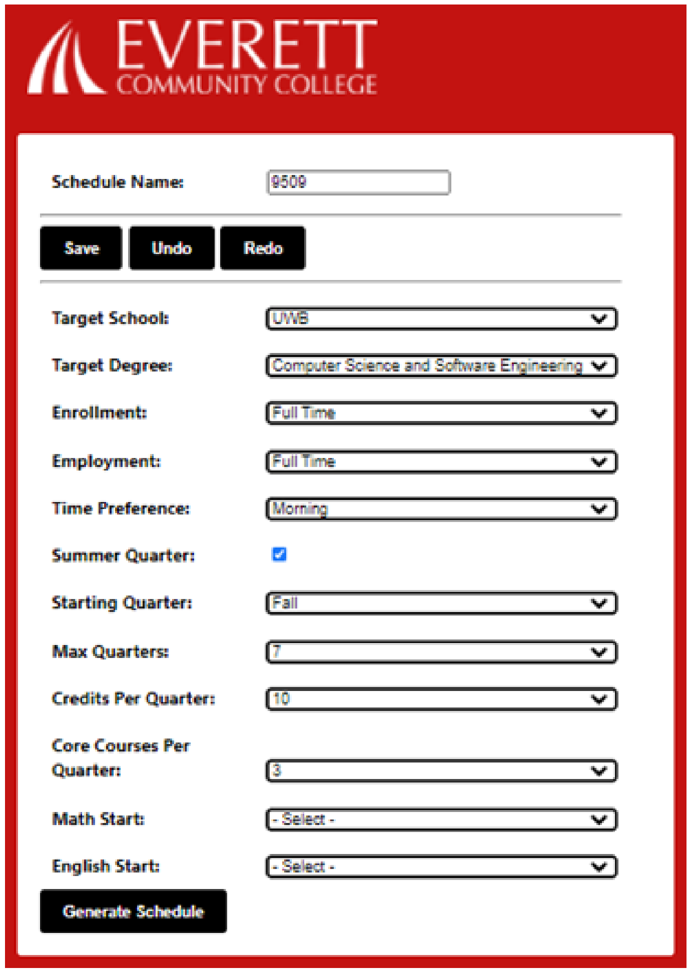


Figure 2.5: Student input preferences from VAA UI

after scheduling the other course and its prerequisites. This causes the scheduler to skip scheduling courses in some quarters/machines, creating breaks in the generated schedules.

Open Shop scheduling algorithm. The Open Shop algorithm schedules courses using the breadth-first search approach on the generated prerequisite network. This algorithm focuses on optimizing the jobs to be performed in a given amount of time [32]. This overcomes the drawback of the job shop scheduling algorithm described above, by scheduling courses that don't have prerequisites among each other along with the courses having prerequisites. A course having a precedence relationship can have more than one group of prerequisite courses to satisfy the precedence relationship. Choosing either one of these groups and scheduling all the courses in that group can satisfy the precedence relationship. This algorithm also chooses the prerequisite group randomly to schedule focusing on generating a schedule with the least number of courses rather than focusing on locally optimizing to select prerequisite groups with the least number of courses. Thus, this overcomes the drawback of the job shop algorithm by minimizing the empty quarter gaps in the generated schedules. Due to this, it sometimes generates schedules having more courses than the maximum number of courses preference per quarter of the student.

Longest path scheduling algorithm. Inspired by the Shortest Job First scheduling CPU algorithm, the Longest path scheduling algorithm prioritizes and schedules prerequisites having the longest length of sequence from the prerequisites priority queues it creates. To minimize sequence breaks with maximum continuity, it is achieved by prioritizing and scheduling prerequisites having long sequence length or long length. Due to this, it doesn't prioritize the prerequisites, which should be given importance when scheduling certain courses. For example, sometimes it doesn't schedule the English prerequisites first, which would unlock certain other courses to be scheduled.

Genetic scheduling algorithm. The Genetic scheduling algorithm [18] finds the optimal schedule generated for the given set of preferences utilizing all the above scheduling algorithms. Each of the scheduling algorithms discussed above is good at producing optimized schedules for a specific set of preferences. The genetic scheduling algorithm was implemented previously [18] to find the optimal schedule from all the schedules generated by these algorithms. It works by creating a population set of all the fittest schedules generated by the job shop, open shop, and longest path scheduling algorithms; it then performs mutation on one-course order in the schedule and generates the crossover schedules. The crossover schedules become the population set for the next generation to mutate. This continues until the top 30% of the schedules have the same rating [7]. This is when the global optimum is reached and the algorithm terminates. This algorithm works on the principle of the genetic algorithm by considering the schedules as chromosomes [7]. Currently, this algorithm stops if maximum number of generations is reached or is the top 45% of schedules remain the same as the previous generation [18]. As the scheduling algorithms are good at scheduling courses according to user preferences, this method reaches optimum at an early stage. Hence, with simulated annealing approach, some least fit schedules are seeded again to the population set during the mutation process, which then using the genetic algorithm is used to find the global optimum. Figure 2.6 shows the overall steps of the existing genetic algorithm used for schedule generation.

Figure 2.6: Genetic algorithm workflow using simulated annealing [18]

Since the genetic scheduling algorithm works to find the best schedule out of all the optimal schedules from all the implemented scheduling algorithms in the system, we will be using it to generate schedules for the AD majors.

2.2.3 Schedule Evaluation

A weak labeling engine or Weak supervision rating system was developed [18] with high-level heuristics to label the data and then evaluate the schedules based on these. Since every schedule generated by the system cannot be rated by an advisor, high-level heuristics to label the generated data were gathered from EvCC academic advisors to rate the schedules. In order to determine the score (or rating) of a schedule, we need to first define several terms involved in this calculation. The following definitions are based on those described in [52]

Definition 1. Let S be the set of all possible valid academic plans within the scope of this work, including both, generated by human and synthetically generated plans. Any study plan can be represented as $s_i \in S$ where and $i = \{1, 2, \ldots, |S|\}$. [52]

Definition 2. Let $\vec{v}$ be a vector or tuple of input parameters to the VAA system. Such parameters include preferences and academic requirements (e.g., “evening classes”, major, target school, etc.). In this document, we will refer components of a vector $\vec{v} \in V$ as parameters or criteria, so we will use these terms interchangeably.

Furthermore, $v_i \in V$ is a vector representing a specific combination of parameters, where V is the set of all possible vectors of input parameters. A list of criteria and weights corresponding to each criterion was obtained to calculate a schedule's score or rating. A schedule's rating is calculated using the formula:

$$Rating_i = \frac{\sum_{j=1}^{n} weight_{ij} \times score_{ij}}{Rating_{max}} \tag{2.1}$$

Where n is the length of a $v_i \in V$, i.e., the total number of criteria considered as input parameters, $i = 1, 2, \ldots, |S|$ is the index of a preference vector (the schedule unique plan identifier), and j is the index of a parameter in $\vec{v}_i$, i.e., the index of each criteria. Then, v_{ij}, would represent the j^{th} input criteria of the i^{th} schedule. Every schedule $\vec{v}_i$ receives a rating computed based on heuristics obtained from the community college, this is denoted as $Rating_i$. Every criteria j in $\vec{v}$, has a score and weight, which will be denoted as $score_{ij}$and

$weight_{ij}$, for the j^{th} parameter of the i^{th} schedule. $Rating_{max}$ is the highest possible rating based on all the criteria and weights. Tablẽreftab:EvalCriteria shows the list of criteria and their corresponding weights used for evaluating generated schedules. The total sum of all the product of all criteria and weights of an academic schedule when it satisfies all the criteria is the value of $Rating_{max}$. Hence, the rating of any schedule, $Rating_i$ will be normalized and would be between the range of 0 and 1.

Table 2.1: Schedule evaluation criteria and their weights [18]

Criteria	**Description**	**Weights**
All prerequisites included	Plan includes all the required core courses and their prerequisites	3
Number of core credits per quarter	Plan meets preference of number of core credits per quarter	1
Number of core courses in last year	Plan meets the preference of number of core courses in last year	1
Time of Day	Plan meets the time preference of classes	1
Maximum number of quarters	Plan meets maximum number of quarters preference of student	2
Order of prerequisites	Prerequisite courses are scheduled in the right order	3
English start	English prerequisites are scheduled early in the plan	3
Math Breaks	Plan doesn't have breaks in Math courses	3

The genetic scheduling algorithm as described in section 2.2.2, uses the schedule evaluation as described above using the criteria and weights listed in Table 2.1. For generating AD schedule plans, we will be utilizing the genetic scheduling algorithm and this schedule evaluation to select the optimal schedules.

2.2.4 Recommendation system

ML is used in the recommendation system for making future decisions taking into consideration, the ratings of the users or the users' preferences [56][28]. The patterns in the data are observed in the ML model and then recommendations are made. Recommendation systems can be classified into Content-based systems and collaborative filtering [3]. Content-based systems examine the user's history data and then recommendations are made. Collaborative filtering works by finding similarities between users combined with their preferences, can be used to create a list of suggestions that can be ranked [45]. In the VAA system, the users' preferences are obtained, and their history is not known. Hence, a content-based system is not a suitable approach for the VAA recommendation system. The VAA recommendation system was proposed with some initial research on the implementation of using collaborative filtering and k-means clustering [33] for finding the similarity between students. The collaborative filtering approach identifies a group of data points that are similar to an instance in the dataset. The similarity between data points is computed using Euclidean distance due to which the complexity is high when the number of schedules and students increases. K-means clustering [5] is an unsupervised ML algorithm that groups datapoints into clusters taking their preferences. It doesn't require extensive computation of similarities between every set of data points. The appropriate number of clusters, 'K' to group the data points needs to be determined for an effective and accurate recommendation system. This was done by using the elbow method and silhouette analysis, as proposed in [33]. The proposed K-means clustering was implemented in [18] using a certain number of schedules as data points. However, research on the usage of the appropriate value of 'K' was not performed and the recommendation approach was not integrated into the VAA system. The aim of the recommendation system implementation in this capstone involves research on determining the optimal value of 'K' on the current dataset and integrating the results with the VAA API for the UI to consume in the future. We will be using the proposed K-means clustering algorithm[33][18] for the recommendation system approach.

In this thesis work, we have addressed the limitations of the previous work in the VAA and table 2.2 shows the functionalities proposed to be added in the new system compared to the existing system.

Table 2.2: Feature comparisons of the previous and new implemented system

Features	**Previous system**	**New system**	**Benefits**
Elective courses scheduling	No	Yes	Elective courses scheduling functionality along with the core courses adds more real-time scheduling options for
Associate degree schedule generation	No	Yes	Students can opt for associate degree schedules which can help them with job opportunities and also improve community college standard
Elective courses data	Very Limited	Yes, added	Elective course options add variety of options to choose for generating schedules along with the core courses
Associate degree data	Limited	Yes, added	Adding more associate degree specifications data increases the pool of degree options to generate schedules according to students' preferences
Integration of recommendation system into VAA system	No	Yes, using API	Students can receive recommendations already in the system based on their individual preferences
Research on appropriate value of K in K-means clustering	No	Yes	Optimal k-value ensures right amount of clusters are created and recommendations are close enough for the given student preferences along with the generated schedule

Chapter 3

METHODS

This chapter describes in detail the architecture of the system, concepts and methods used in the design, and the implementation of the AD scheduling and Recommendation module in the VAA. The VAA system is implemented on the principles of service-oriented architecture based web application. We discuss the architectural structure and the software development life-cycle process involved in each of the components in the VAA system.

3.1 System architecture

A high-level representation of the architecture of the system is shown below in Figure 3.1. The overall architecture involves the VAA UI which is used to display schedules and obtain the student preferences. The VAA API is the middle layer that is used to send and receive responses to the scheduler and the recommendation system. The scheduler and recommendation engine first obtains schedule and course data from the VAA database, then implements algorithms to generate schedules, and finally it saves the results to the database.

Software Development Lifecycle Process. The Software Development LifeCycle (SDLC) is an approach consisting of six phases happen during the development and deployment of a software application, these are: Requirement gathering and analysis, Design and Implementation, Testing, Deployment, and Maintenance [15]. Currently, any development in the VAA system is done using the SDLC model adopting the agile methodology, Scrum [19]. Team members address problems of varying levels of complexity during productive Scrum meetings. The backlogs are used and monitored using GitHub's Kanban board [19], which is used to host the VAA backend code repository. The product leader and owner

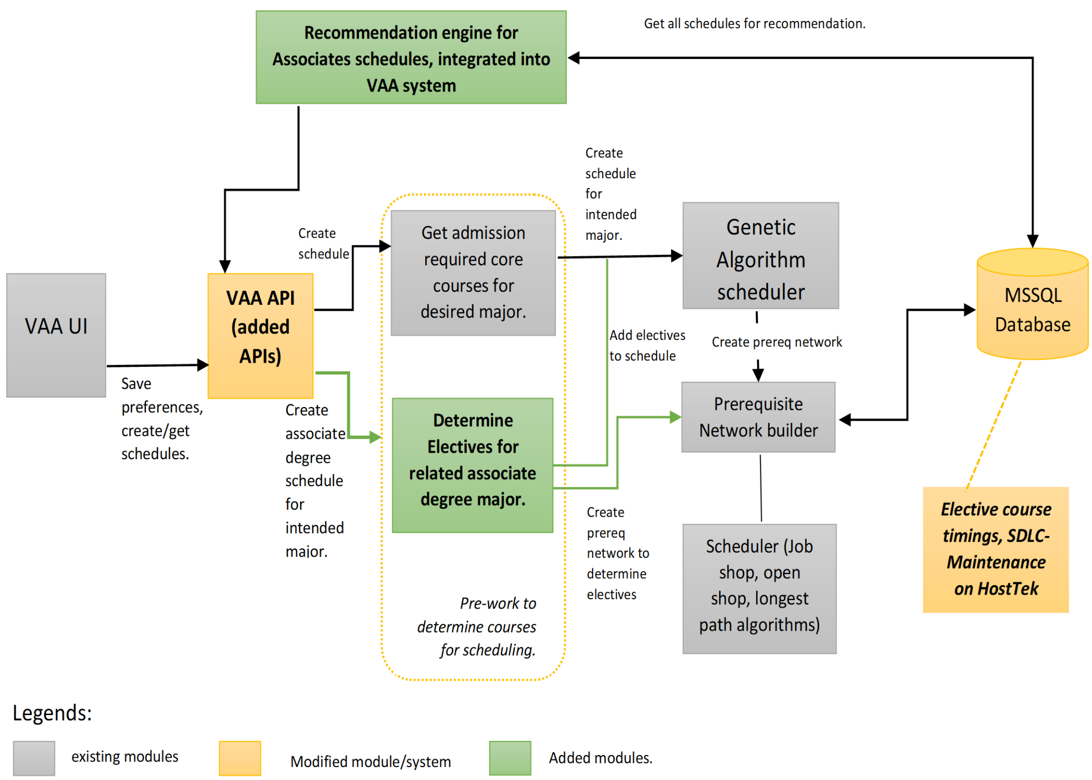


Figure 3.1: Overall system architecture with the newly added features and modules

is required to continuously update and track the backlog before and after every meeting. The following sections give a brief idea of how all the phases of SDLC were used in the implementation of this project.

Requirements Gathering. Gathering the requirements was performed by interacting with the EvCC academic advisors and getting the AD requirements and details from their official websites. Conversations about the advantages of AD for students as well as the community college took place which also involved a detailed understanding of their expectations on how the electives have to be scheduled.

Design and Implementation. The existing VAA system has the necessary modules for UI, APIs, and database (storage). The scheduling algorithm to be used for the AD schedule was determined to be the Genetic algorithm with simulated annealing [3] as it obtains the most optimal schedule using all the scheduling algorithms already implemented. The design of the project was discussed with the team and then the best approach was determined. The design and implementation are detailed in section 3.3.

Testing and deployment. The APIs used in the existing VAA system are tested on a local system using the Swagger hub framework before deploying in the Azure web apps [17]. The backend system with the scheduler is tested connecting to the development database hosted in the HostTek [54] account before any implementation changes are added to the main code repository in GitHub. Every coding implementation is ensured to have proper documentation before adding to the master branch in the Git repository. The VAA UI code is also hosted in a GitHub repository. It is tested in the local system before adding the changes to the GitHub repository. The database is hosted on a remote server in the HostTek. When the changes to the development database are approved by the product owner, the changes are then pushed to the production instance. The database is built and maintained using MSSQL server [13].

Maintenance. The GitHub repositories are maintained with proper documentation of the code, with Prof. Parsons being the owner of the code repositories. Any new code push to the master branch is verified if it meets the coding standards and documentation before the merge with the master branch.

3.2 Data collection for Elective courses and Associate Degrees.

Due to challenges in data collection, the VAA system database has had a lack of entries for certain data such as preferences, generated schedules, suitable elective courses (elective courses that have valid course timings). Certain AD majors and their requirements data have also not been available. This section details the work involved in collecting all this data for performing our experiment. This section refers to the terms involving AD specifications and relational structure as described previously in section 2.2.1 and section 2.2.1.

3.2.1 Data collection for Associate Degree requirements.

In order to generate Associate Degree (AD) academic plans, information for such degrees is necessary (e.g., types of courses, number of credits). At the beginning of this thesis work, the VAA database contained information only for the 'Associate in Business DTA' and 'Associate in Arts & Sciences' degrees, along with their requirements and elective options. However, this information was insufficient to test the AD schedule-generation algorithm with all the known majors. Information was missing for three of the five AD majors namely, 'Associate of Science' - Computer and Electrical engineering, Associate of Science - Pre-Engineering for Mechanical, Aeronautical, Industrial, Material science, and Associate of Science - General Engineering transfer. As part of this work, this missing information was collected and entered into the database; it was gathered by engaging the EvCC academic advisors/faculty as well as the EvCC "Associate Degree Curriculum Structure" web pages[2][4][1].

3.2.2 Data Collection and generation for Electives and Student preferences.

Besides lacking some types of information in quantity, the VAA data had other forms of data limitations, for instance, some of the existing academic plans and preferences combinations were had been created by EvCC academic advisors/faculty and were manually entered by the team into the database. The main reason such kind of data is so limited is because it is very hard to obtain and parse. For instance, on an academic year a single advisor may advise 80 students, if this advisor didn't follow any formatting rules and using their current software, then their resulting academic plan may be close to gibberish that cannot be easily parsed or even interpreted. In an initial effort the team was able to collect only about 100 useful examples of schedules and associated preference settings. Furthermore, this kind of examples have little statistical variability of schedules and combinations, making it very challenging to test the different combinations of input preferences of a random student especially if they don't fit any of the few existing examples. Furthermore, we acknowledge that way more examples (data points) are needed for ML strategies.

To address the issue of limited examples, previous work developed implementation strategies to generate synthetic data [18]. However, there has not been enough preferences stored in the system to generate schedules. This lack of information impacts the functionality of the recommendation system. The data for AD requirements and electives for AD majors are very limited in comparison to those for core courses. So far, there had not been any schedule generated and stored in the database that considers electives. In this research, we worked on strategies to generate synthetic data for elective courses and their course time and day offerings. Data generation for all the elective courses with their course timings required communication with EvCC academic advisors/faculty to confirm correctness before storing them in the database. Even though ML strategies can be used to generate the course time offerings data, there are some conditions to be met from the academic plan perspective. For example, for course-time offerings, EvCC and other institutions use standard course box slots, which ensure avoiding overlapping of certain courses, as well as staff availability. Course box slot is a standard time window such as a one-hour slot, two-hour slot during which any course would be offered. Schools ensure that all the courses are offered in standard box slots with consistent start and end timings. Such checks need to be done before they can even be entered into the database.

Due to the lack of information about course offerings, i.e., days of the week and time of the day, it was not possible to fully populate the VAA database with such information required for scheduling. For this reason, we "forge" certain course offerings times with varied hypothetical days and times so that we can test and generate trial schedules. While some of this data is not real, we can have a reasonable approximation of what actual offerings would look like; we can achieve some of this, by making certain assumptions such as the "box schedule". A box schedule is used by various community colleges and universities for scheduling courses, classrooms, and allocating staff. A box schedule consists of well-known course box slots and days when a specific course is usually offered, e.g., in UWB, CSS535 is always offered on T/Th from 5:45pm to 7:45pm. Examples of data generated for elective courses with box schedules are discussed later in section 4.1. To have a more realistic set of data for the course timings for all the elective courses, we reached out to the EvCC academic advisors to get the course timing blocks and their offerings schedule along with the staff availability. For the current experiments and target scenarios, the data is limited to EvCC's information since we can work closely with their academic advisors on validating this data. After reaching out to the advisors, we concluded that the elective courses are offered during the course box slots in the afternoon and evenings. Usually, the core courses to achieve transfer credits for the major are offered in the morning course slots. There are a variety of elective courses offered which would satisfy the different requirements. Schools decide on the set of elective courses to offer every year by planning aspects such as the set of courses to offer for satisfying all the minors and those that satisfy all the credit requirements for every elective-requirement group in the AD.

We encountered that the existing VAA database does not contain any data for almost 95% of the elective course offering timings. There are a few elective courses for which limited offering timings are present in the database. However, these don't fully cater to the satisfiability of all the requirements in the AD. Since electives are flexible, they get offered during different timings during every quarter, unlike the core courses that get offered during the same time box slots throughout all the quarters every year. We have identified the existence of what we have dubbed *elective-requirement groups* (see Definition 10, each of which contains a large number of possible electives to satisfy specific credit requirements; this concept will be explained in detail in section 3.3 as well as its relevance in creating AD schedules. Due to the large number of electives, it was not possible to enter all of their time offerings into the database since this has to be done manually. Hence, for testing purposes, we have picked a random 30% of them from different possible minors to enter their offering timings in the database.

Lastly, possible combinations of input student preferences that are used to generate the schedules for AD majors were not present in the database. We have used *weak supervision* data generation approach as in [18] to generate possible combinations, and thereby generate the schedules for testing. The generated labeled data is then used for training the scheduling engine.

3.3 Associate Degree Schedule Generation

Scheduling core courses in the existing system. The existing system is dependent on the scheduler for generating schedules for the provided major and target school preference of the student. The preferences of the student include the major, target school of their choice, core courses per quarter, maximum credits per quarter, summer quarter preference, and evening classes preference. Hard constraints on these preferences include the target school and major which cannot be changed by the scheduler to generate the academic schedule. The number of credits per quarter, number of core courses per quarter, evening class preference are soft constraints that can be modified to some extent by the scheduler to generate the optimal schedule. Once the preference set of the student is passed to the API, it creates the prerequisite network which has the graph structure of prerequisite courses for all the courses in the database. The scheduling algorithms are dependent on the prerequisite network to determine the courses and their prerequisites to be scheduled. The prerequisite network is explained in detail below. In the scheduler module of the existing system, only the required core courses to be scheduled for entering the major and target school of choice are retrieved from the database and stored in a list. All these required courses for admission/transfer, which need to be scheduled for getting transferred are treated with equal priority by the scheduling algorithms. The prerequisites for all these courses are determined from the prerequisite network and are then ordered with the previous list of required core courses to be scheduled by the genetic algorithm-based scheduler [18]. Hence, the generated schedule only contains the required core courses and their prerequisites.

Prerequisite Network. In the VAA system, all the scheduling algorithms use the prerequisite network built earlier in [33] for determining the courses that need to be scheduled. The prerequisite network is a directed acyclic graph (DAG) where the nodes in the graph denote the courses and directed edges represent the dependency of the courses. From now on, we will denote the Prerequisite Network as PN. Figure 3.2 below shows a partial prerequisite network used for a student's preference to enter Chemistry (Bachelor of Science- B.S.) at the University of Washington.

Electives scheduling. When a schedule for an AD is to be generated, the major of the AD that is similar to the intended major of the student needs to be identified. For example, the Associate in Business degree can be applied to a student intending to get transferred to a school in a business major. Similarly, "Associate of Science – Pre-Engineering" degree can be applied to students whose major preferences are in any of the following engineering fields: "Mechanical, Civil, Aeronautical, Industrial, Materials Science". The list of all ADs in EvCC and to what major preferences of the student they can be applied are listed in Table 3.1. Each of the ADs has separate entries in the 'Major' table.

As previously discussed, in the current scheduler, only the required core courses for the intended major and school are obtained from the database and scheduled by the algorithm. For scheduling AD plans, the electives need to be scheduled along with the required core courses for the intended major. Since the students aim to get transferred to a university with the core courses, the electives cannot be treated with the same priority as the core courses when scheduling them. To determine how to schedule the electives along with the required core courses, we identified two approaches as listed below.

Approach I. Schedule electives after the core courses with lower priority: From the student's perspective, they aim to obtain a schedule that paves an easier and shorter path to complete their courses at the community college to get transferred to their intended school. In that case, the electives cannot be treated equally when scheduling them along with the core courses. We need to schedule them after the core required core courses are scheduled.

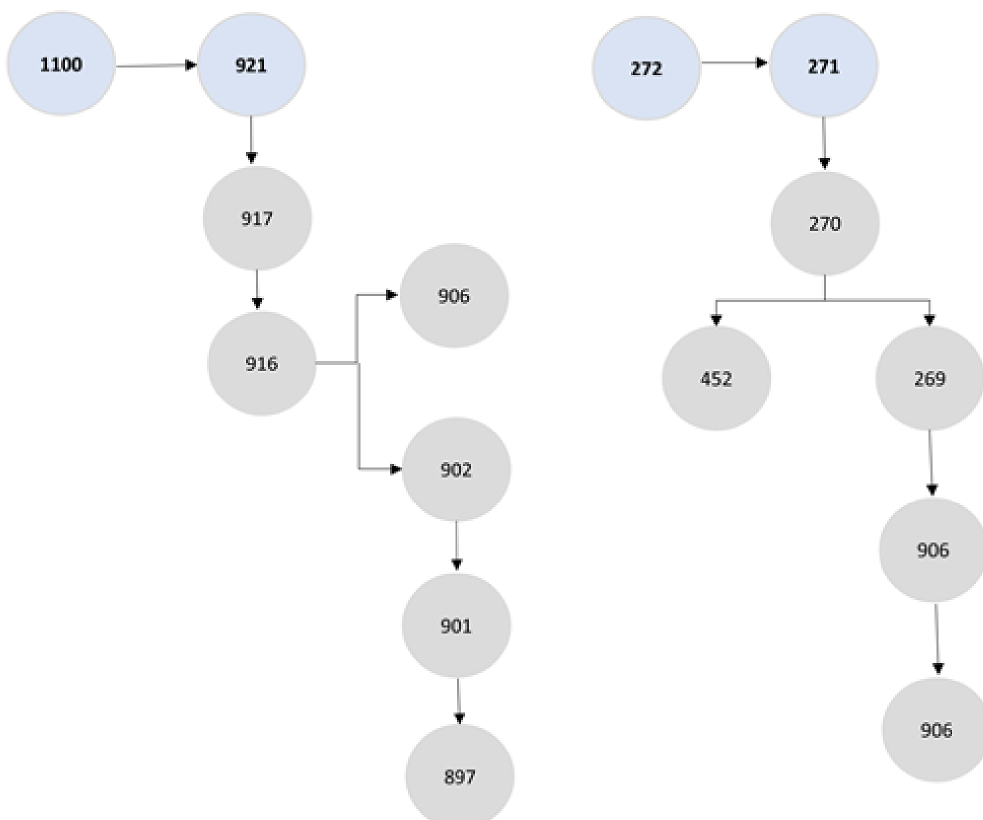


Figure 3.2: Partial prerequisite network for entering Chemistry (B.S.) at University of Washington Seattle: four required courses (blue colored nodes) required to enter the Chemistry B.S. at the University of Washington are shown. Their respective prerequisites are shown in grey color nodes. The prerequisite network path shows the dependency chain as these required courses for transfer has multiple prerequisites.

Table 3.1: Associate degree majors compatible with specific major preferences

Associate degree major ID (as in the database)	**Associate degree major**	**Specific Majors**
36	Associate in business	Business, Business DTA (Direct Transfer Agreement)
37	Associates in Arts & Sciences	Computer Science
17,18,19	Associate of Science - Pre-Engineering	General Engineering fields such as Mechanical, Civil, Aeronautical, Industrial, Material science, Computer, and Electrical Engineering

This takes a toll on the scheduler's flexibility on the soft constraints of the students' preferences. Even though it gives some flexibility to choose electives from a long list of the possible options and is similar to how academic advisors would schedule an AD plan, it is relatively difficult to incorporate this in the current scheduling algorithm.

Approach II. Schedule electives along with the core courses: From the perspective of scheduling courses for an AD, this involves treating all the courses including the core courses and the electives with equal priority. Hence, we generate the list of all the core courses and the elective courses that can fulfill the AD requirements to generate the schedule. This is relatively easier to implement by incorporating into the existing scheduling algorithm, however, it may not generate the optimal schedule with the optimal courses that can yield the best-rated schedule satisfying all the preferences of the student. This is because finding the exact elective courses along with core courses from the huge pool of possible electives to satisfy the requirements leads to a combinatorial problem. However, this approach would be useful for any scale-up in the future.

After carefully analyzing these two approaches, we decided to implement a solution that combines both of them. As we discussed previously in section 3.2.2, the elective courses are predominantly offered in the evening course box slots, which don't overlap with the core course timings in the mornings. However, there may be a few elective courses that can be offered in some afternoon slots that may overlap with the core course timings. We also know that core courses are offered in fixed course slots planned during every year, which usually don't change as the number of students enrolling in these courses is usually more or reaches the maximum limit. However, there can be some flexibility provided to some elective courses in terms of the timings if the enrolled students' count is relatively low. Keeping this in mind, we implemented an approach by first determining the required core courses for the student's intended major and target school of transfer, and then determining the remaining elective courses from each pool of elective options that would satisfy all the requirements for the ADs. This will also include the list of all prerequisite courses of the core courses and the elective courses obtained from the built prerequisite network. The following describes the definitions for the algorithm along with the algorithm in detail.

Abstractions and Algorithm. Here, we present and discuss the several relevant concepts and their formal definitions. These are essential to understand how schedule recommendations are created based on credits, degree definitions and various types of courses. We use set representation and operations to represent concepts, to better abstract and formalize their definitions and interactions, and how they can be used to create a general algorithm to create recommendations including electives and considering ADs. Figure 3.3 shows the mapping of different course types relevant in the VAA into a set diagram.

Definition 3. Let D be the set of all the degrees offered in a community college. A degree $d \in D$ is a specific combination of courses corresponding to a unique combination of target major and school. For instance, the set of courses required to transfer WSU to the Mechanical

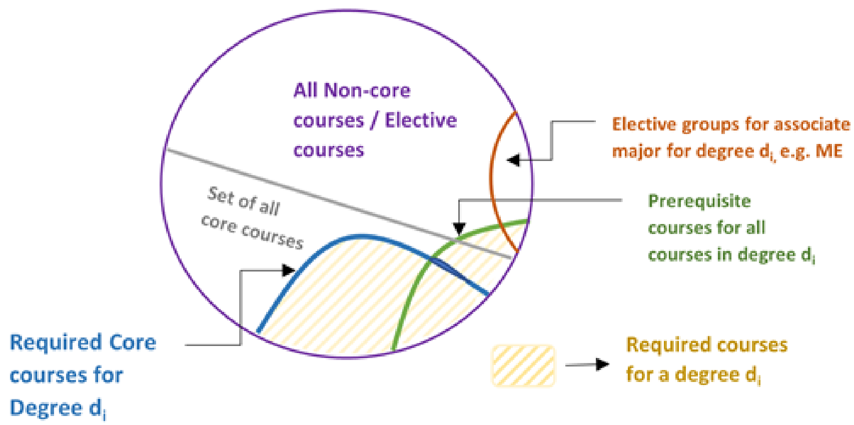


Figure 3.3: Set diagram showing the different types of courses in EvCC. This diagram is used to define and explain abstractions used in the formalization of the VAA system's concepts as well as the implementation of the algorithm.

Engineering major. In addition, we must consider ADs, which are not transfer courses, but instead required to obtain a degree at the community college.

Definition 4. Let D_A be the set of all the ADs offered by that same community college, and $D_A \subset D$.

As discussed in previously Section 3.2.1, for any plan $d \in D$ created for a degree for transfer, there would be an AD. We can denote as $d' \in D_A$ the AD closest to d.

Definition 5. Let $\mathbf{C}$ be the set of all courses offered at a community college. Out of all the courses in $\mathbf{C}$, some of them are core courses, which means that they are required to complete a degree, or for a transfer to a university. We will denote the set of all core courses as C_K, where $C_K \subset \mathbf{C}$.

Definition 6. The set of courses that are not part of C_K are known as electives and will be denoted as C_E, where $C_E \subset \mathbf{C}$ and $\mathbf{C} - C_E = C_K$, i.e., $C_E = \overline{C_K}$ In addition, there are some sequences of courses where all the courses in the sequence are prerequisites except for the last one.

Definition 7. Let $C_P \subset \mathbf{C}$ be the set of all prerequisite courses at a community college, i.e., all courses whose completion is required to take another course.

A degree can be defined as an instance $d_i \in D$ consisting of a specific list of required courses that fulfill the number of credits to be completed towards that degree. All such courses come from C_K and C_E, so generally speaking, all the courses required in all degrees can be defined as $R = C_K \cup C_E$, and $R \subset \mathbf{C}$, i.e., the set of all core courses offered at a community college that appear in transfer requirements for degrees in universities. Now, in the context of the degree d_i, there is a set of specific major-related courses required to transfer to a certain major at a specific university.

Definition 8. We denote the of required courses for degree d_i as $R_i \subset R$, and let $r_i \in R_i$ be the set of core courses required to transfer to a specific major and university $d_i \in D$, where $i \in \{0, 1, 2, \dots, n\}$ and n is the number of total possible major-university combinations.

Credit Definitions In addition to specific courses, degrees also require a number of credits an every course accounts for a precise number of credits. We will denote the total number of credits required by a degree $d_i \subset D$ as N_i. The number of credits for course c_{ij} in degree d_i will be denoted as n_{ij}. If, m is the total number of courses in degree d_i, in other words:

$$N_i = \sum_{j=1}^{m} n_{ij} \tag{3.1}$$

Figure 3.4 depicts the relationship between degrees an courses as a matrix. Every degree can be seen as a sequence of required courses. In this figure, a degree is represented as a row (vector), i.e., each row is a degree and each column is a course belonging to that degree. Furthermore, each course accounts for a number of credits. Note that not all degrees have the same number of courses, so some course elements of this matrix will be empty (accounting for zero credits).

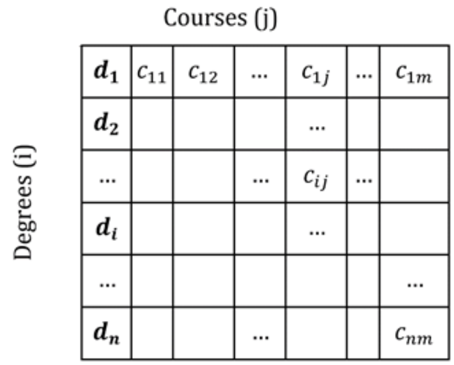


Figure 3.4: Relationship between degrees and courses.

Using this logic, we can represent a degree as a vector of courses with their corresponding number of credits, e.g., a degree d_i having m number of courses would be represented as:

Courses	c_{i1}	c_{i2}	...	c_{ij}	...	c_{im}
Credits	5	5		3		5

In other words, the total number of credits for d_i can be calculated as follows:

$$N_i = \sum_{j=1}^{m} n_{ij} \tag{3.2}$$

Where n_{ij} is the number of credits for each course in d_i.

In practice, an academic plan has more courses than the required number of courses to satisfy the degree. Now take for instance an automatically generated schedule that besides fulfilling requirements of d_i, also fulfills those required by the related AD, we will call this degree $d_i^{'}$ with a total number of courses of $m^{'}$ then we have:

$$N_i^{'} = \sum_{j=1}^{m^{'}} n_{ij}^{'} \tag{3.3}$$

And it is also true that $N_i^{'} > N_i$, meaning that the recommended schedule has more credits.

Requirements Degrees have different types of requirement and different courses fulfill certain number of credit requirements of one or more types.

Definition 9. Let T_{req} denote a type of requirement for a degree $d_i \in D$. The set of requirement types is: $T_{req} = \{$'*basic communication skills*', '*social science*', '*humanities*', '*natural sciences*'$\}$. We use req_k to represent a specific type, where $req_k \in T_{req}$.

In addition, several number of credits (from different categories), are needed to satisfy requirement req_k, for instance, the degree "Associate in Business" needs 10 credits of '*basic communication skills*', so in this case, $req_k = 10$, where $k =$ '*basic communication skills*'. Such different categories are exemplified in degrees' specification guides for students. For instance, in Figure 2.3 in Section 2.2.1, we can see this scenario under "Basic Quantitative Skills".

Definition 10. We formalize the term elective group C_{EG} as a set of courses, that is needed to fulfill a requirement type. To satisfy a type of requirement type req_k, at least one elective group C_{EG} will be needed.

Every elective group C_{EG} is formed of a list of elective courses, from which a student can choose in order to satisfy a requirement type $req \in T$. In addition, each requirement type req_k, has a very specific minimum and maximum number of credits to be satisfied from each of its elective groups. For example, in Figure 3.5, a degree that has the requirement type '*natural sciences*', has an elective group named "Part A Lab", which needs between 5 and 10 credits to be satisfied.

Functions. Let ϕ be a function that filters a set of electives L_E for a given course elective C_{EG} group, note that L_E is formatted as a list, $C_{EG} \subset C_E$ and $L_E \subset C_E$. Elective courses in

NATURAL SCIENCE - 15 credits. 5 credits in Math&146 -Statistics and 10 credits from the AAS DTA Natural Science distribution list. At least 5 of the Natural Science AAS DTA credits must be Part A lab credits and the remaining 5 credits can be either 5 credits from the Part A lab courses or 5 credits from the Part B non-lab courses.

Course	Course Title	Credits	Grade	Quarter	Year
*MATH& 146	Introduction to Statistics	5			
Part A (lab course)		5			
Part A or Part B		5			

REQUIRED BUSINESS - SPECIFIC ELECTIVES (20 credits)

Course	Course Title	Credits	Grade	Quarter	Year
*ACCT& 201	Principles of Accounting I	5			
*ACCT& 202	Principles of Accounting II	5			
*ACCT& 203	Principles of Accounting III	5			
*BUS& 201	Business Law	5			

Figure 3.5: Partial requirements specifications from Associates in Business degree DTA

L_E with the least number of credits in C_{EG}, that have available course timings in database and are compatible with other selected electives timings and those that have the least number of extra prerequisite courses to be scheduled and are stored in the list of electives courses L_E. The following functions will be used in Algorithm 1.

Definition 11. Let Q be a function that maps from one or more input sets into a single output set, based on specific criteria selection, i.e., a query. For example, such function can typically be implemented as a SQL query in a database such as:
'select CourseID from AdmissionRequiredCourses where MajorID=22 and SchoolID=6'. This SQL query retrieves the required core courses for a combination of Major and School preference specified by a student.

Algorithm 1 Associate degree schedule generator

Inputs:

The vector of user inputs (preferences) $\vec{v} \in V$

The prerequisite network **PN** (see Section 3.3)

Output:

Schedule S_i for Associate Degree(AD) d_i

Begin:

From $\vec{v}$ obtain the desired major-school combination i, and the corresponding AD, d_i

$r_i \leftarrow Q(d_i)$ obtain the list of all required core courses $r_i \in R_i$

For every course in r_i, use **PN** to determine all specific prerequisite courses $c_{pi} \in C_P$

Let $R_i \leftarrow r_i + c_{Pi}$ is the set of all the required transfer courses for degree d_i

$req_k, C_{EG\_k} \leftarrow Q(d_i)$ where Q obtains the list of all requirements $req_k \in T_{req}$ for "degree" d_i as well as the courses corresponding for each elective group $C_{EG\_k} \in req_k$

for every requirement type req_k **do**

Compute the credits to satisfy N_k

for every $C_{EG_l} \in req_k$ **do**

Compute credits n_{kl}

end for

end for

for every course $c_{ij} \in R_i$ that satisfies each requirement type req_k and $C_{EG_l} \in req_k$ **do**

Compute remaining credits $N_{k_{rem}}$ to be satisfied using:

$N_{k_{rem}} \leftarrow N_k - \sum_{j=1}^{m} n_{ij}$ and $n_{kl\_rem} \leftarrow n_k - \sum_{j=1}^{m} n_{ij}$, where m is the total number of courses that satisfy each req_k or C_{EG_l}

end for

Obtain elective courses $L_E \leftarrow \phi(C_{EG_l})$ for every course elective group $C_{EG_l} \in req_k$ whose $min\_credits > 0$ and $\forall$ requirement types req_k whose $N_(k_r em) > 0$, where $L_E \in C_E$

For every course in L_E, use the **PN**, to determine all the specific prerequisite courses and add them to the list $cpi \in C_P$

For every course $c_{ij} \in L_E$ that satisfies req_k and $C_{EG_l} \in req_k$, compute their remaining credits to be satisfied using $N_{k_{rem}} \leftarrow N_k - \sum_{j=1}^{m} n_{ij}$ and $n_{kl\_rem} \leftarrow n_k - \sum_{j=1}^{m} n_{ij}$ where m is the total number of courses that satisfy each req_k or C_{EG_l}

repeat

Obtain elective courses $L_E \leftarrow \phi(C_{EG_l})$ for any $C_{EG_l} \in req_k$ for all req_k whose $N_{k_{rem}} > 0$, where $L_E \in C_E$

For each course in L_E, use the **PN** to determine all the specific prerequisite courses and add it to the list $c_{Pi} \in C_P$

until $N_{k_{rem}} <= 0$ for all req_k

Call scheduling algorithm for courses in the final set $r_i \cup c_{Pi} \cup L_E$ for degree d_i

Figure 3.6 shows the steps for determining the elective courses to generate an AD schedule. When any of the electives selected for scheduling is not compatible with scheduling with the other courses due to its course timings, such as cases where the prerequisite courses or the core courses are already scheduled by the scheduling algorithm during those timings, the algorithm looks for other elective courses in the "selected electives" list and schedules them to satisfy the requirements.

Figure 3.7 shows a sample schedule generated by our algorithm for Associate in Business DTA degree with preference of entry into University of Washington Bothell and Business major. The schedule is displayed on the VAA UI, which uses and consumes the API we developed for generating the AD schedule as discussed below.

API for associate degree schedule. The AD schedule plans are generated by mapping them to the majors already present in the database. The mappings of the majors are stored in a table in the database, which is used for determining the AD major when the preferences of the student are passed from the VAA UI. Once, the AD major is determined, the electives to schedule along with the core courses are determined as explained in the previous section 3.3. The generated AD schedule is then saved in the database. The VAA UI interacts with the

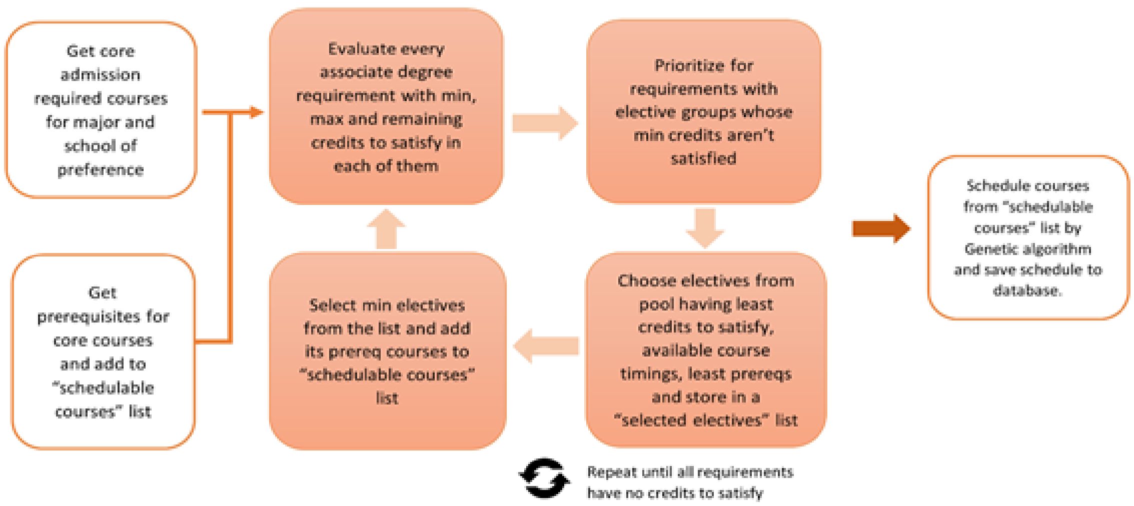


Figure 3.6: Associate degree electives scheduling approach

VAA scheduler and database using the API. We have designed and implemented an API that performs the above tasks when the student's preferences are passed. There are two APIs created for this purpose. Section 4.1 in the next chapter details the POST API call that is used for sending the preferences of the student for generating the AD schedule. The API responds with the schedule ID of the AD schedule created.

3.4 Alternative schedule Recommendation Framework

Using the schedules generated as discussed in section 3.3, we have a good amount of data along with their ratings to apply ML techniques such as k-Means clustering to cluster similar schedules together and recommend them [33]. Clustering can be used to identify neighbors that have similar courses in their schedule [18]. We have integrated the results of the clustering model using API into the VAA system, which could be consumed by the UI in the future to help students by recommending other similar highly rated schedules. The main aim of this research involves identifying the appropriate "K" value in the k-means clustering

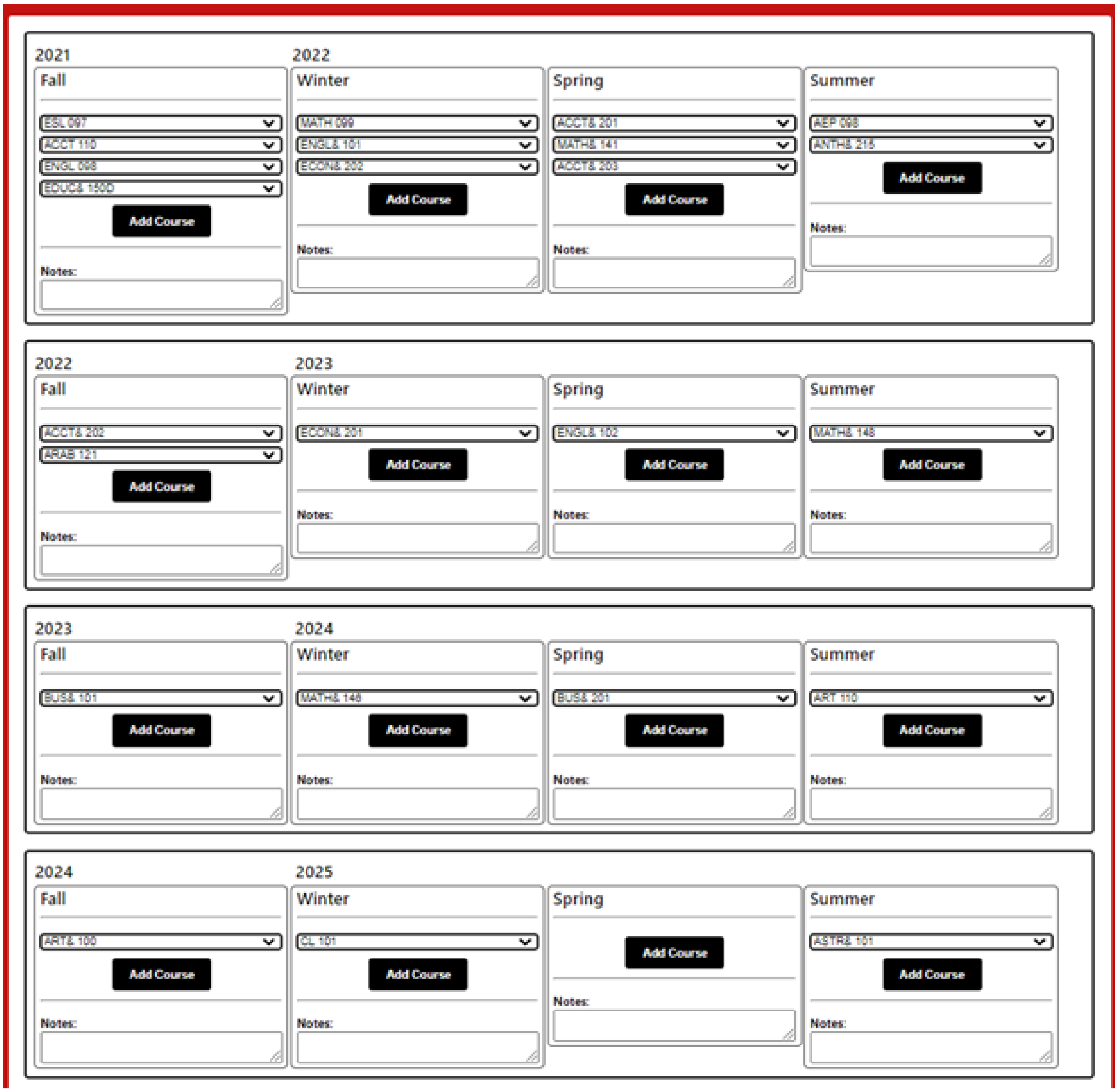


Figure 3.7: Sample schedule generated for Associates in Business DTA major on VAA UI

for grouping the schedules into "k" numbers of clusters depending on their features.

3.4.1 Determination of K in K-Means Clustering

In this scope of the project, we have researched the elbow method, silhouette analysis, and the Davis-Bouldin score on the existing set of schedules to determine the appropriate k-value for the k-means clustering approach. Based on the results of all the scores plots, we created a K-means clustering model which recommends the nearest neighbor (schedule) for the given test instance (schedule ID) through the VAA API.

Elbow method. This provides a simple visual method to estimate the optimal number of clusters, 'K', in the K-means clustering algorithm. The score represents the variance as the sum of squared distance between every datapoint and its closest centroid. This score is computed for all values of K. This score decreases as the K value increases because the samples would be closer to the centroids of the cluster they are assigned to. The rate of change in the sum of squared distance slows down for a specific value of K where it would form an elbow on the graph. The value of K can be determined by identifying where the score rapidly decreases before it reaches a plateau.

Silhouette Analysis. Silhouette Analysis (SA) uses silhouette coefficients or scores that measure the average similarity of data points in a cluster and the distance to other data points in other clusters. The silhouette plot can be used to visualize the closeness between the clusters [6]. SA scores range between -1 and 1 [16], where,

- -1: data point is assigned to an incorrect cluster.
- 1: data point assigned to the correct cluster.
- 0: data point lies on the boundary line between two clusters

SA is done by generating the silhouette plot for a range of K values on the dataset. Through SA we can look for a value of K that has high average silhouette scores for determining the

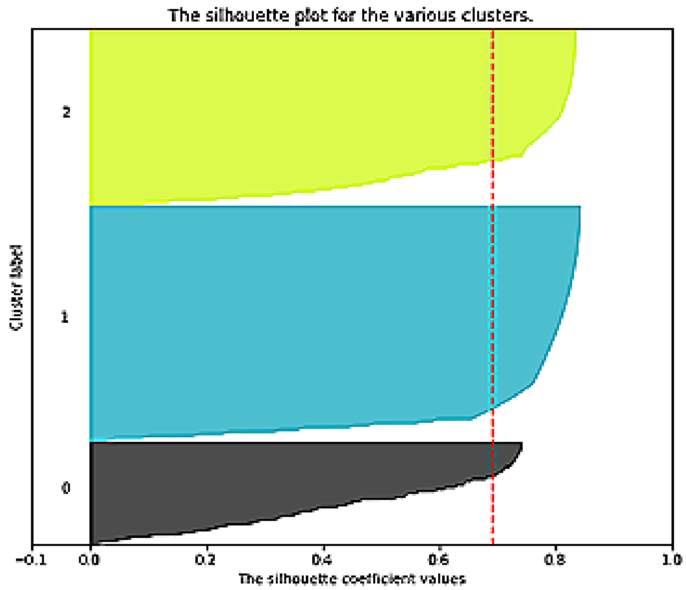


Figure 3.8: Example of silhouette plot in SA for k=3[16].

appropriate value of K. A silhouette plot is used to visually represent the silhouette scores of all the points and the clusters they are assigned specifically. An example of a silhouette coefficient plot for k=3 is shown below in Figure 3.8. The x-axis represents the silhouette coefficient value, and the y-axis represents the cluster label. The average silhouette score is represented by the red dotted line. Each cluster is represented using different colors. The number of data points in a cluster determines the height of the box of that cluster label.

Davies-Bouldin index. The Davies-Bouldin (DB) index is based on a ratio between the intra-cluster and inter-cluster distance of all points between two clusters, called the "within-to-between cluster ratio". The DB index for a value of K is the maximum value of the within-to-between cluster ratio between any two clusters. A large value of DB index denotes that two clusters are close together but have a large spread indicating that the clusters are not significantly defined. In our case, the data points must be assigned to clusters that are closely packed so that the recommendation of the nearest neighbor would be more accurate.

The DB index plot shows the DB indices for a range of K values based on which the optimal K can be determined.

3.4.2 Recommendation framework approach

The recommendation framework involves all stages of ML beginning from data pre-processing till the generation of model with the given dataset and then testing and predicting the closest recommendations for the given test data (academic plan).

Data preprocessing. The feature set includes the preferences of the student from the preference set as mentioned in section 2.2.2 and the courses in all the schedules in the VAA database. Currently, the required core courses are limited and the electives that we have added are also still limited compared to the entire set of electives possible for the courses. We have 296 different courses that can be considered as features in the dataset taking all the 9800 schedules in the database. A partial feature set for a schedule is shown below in table 3.2.

The dataset had missing values for certain preferences such as the *CreditsPerQuarter*, *CoreCoursePerQuarter* specifically for some manually generated schedules during the initial phase of development of the VAA project as they were recently introduced in the project [52][33]. These missing values can affect the model's conclusions significantly. We have identified values to replace the missing values for these features and implemented them. The categorical feature, *SummerPreference* has been converted to numerical value form to denote the preference of the student to take classes in summer. All the courses considered as features are assigned values using one-hot encoding denoting the presence of the course in the schedule using binary value (0 or 1).

Certain students' preferences and schedules generated may be the same. In the data cleaning process, such entries are duplicate rows where the value in every column is the same. Removing duplicate data is an important step that ensures accurate usage of the data [43]. We have removed 832 rows that were identified as duplicates. The model is created on

Table 3.2: Feature set (partial) of a schedule in the dataset

Feature	**Value**
Major	20
School	1
MaxQuarters	12
CoursesPerQuarter	10
CreditsPerQuarter	10
summerPreference	1
DepartmentID	5
WeakLabelScore	0.874
452	1
550	1
509	0
69	1
32	0
913	1

this data in which the duplicate rows are removed.

Feature Reduction. We used elimination with cross-validation for providing relevant features to the clustering algorithm. Otherwise, the model may take more time to train and the accuracy of the model may decrease. We noticed that some courses were present for most of the combinations of majors and schools. We started with 296 features for all the courses in the schedules and using elimination with cross-validation method, we reduced the total number of courses in the feature set to 190 courses. This method was first introduced in [33] when the dataset and features were initially lesser. This method was also implemented in [18] by considering only the major-specific courses and having lesser combinations of majors and schools.

Clustering. K-means clustering algorithm was performed on a limited dataset and the research recommended k=3 clusters [33]. The same approach was used on 3700 schedules previously [18] using k=3 and grouping of schedules was done. However, currently, our dataset has grown to 9800 schedules and the feature set with the AD majors and elective courses. We need to carefully analyze the dataset for selecting the appropriate value of K [30]. We performed the Elbow method and Silhouette analysis along with a Davis-Bouldin score plot, which gives a visual representation for easier interpretation of the results.

With the ML algorithm finalized to be k-means clustering, the overall approach of recommending schedules for the given schedule ID (test data) through the VAA API is shown below in Figure 3.9. The proposed recommendation framework approach is implemented in two phases. Phase-I involves the building of the model using the existing data in the database involving the methods as described in the previous section 3.4. The model is then saved locally in the system using the appropriate command. Phase-II involves the API request for recommendation with the recently generated *ScheduleID*. The saved model is retrieved and the data for the *ScheduleID* and the student's preferences are obtained from the database, which is then used as test data to perform clustering in the retrieved model. The model then

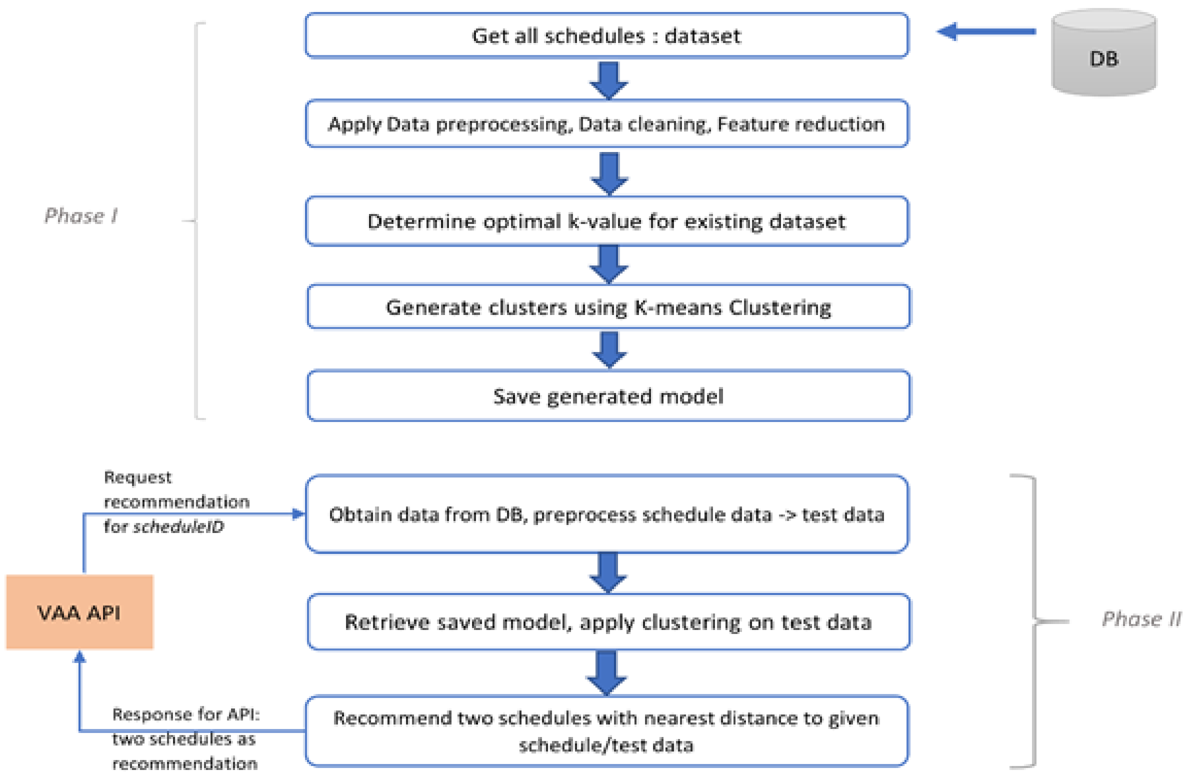


Figure 3.9: Steps and approach for K-means clustering and recommendation framework

recommends two nearest schedules based on the nearest distance, which is sent as a response in the API. To increase the response time for the recommendation, the model is generated previously on the existing data and saved in the system.

Chapter 4

EXPERIMENTS AND RESULTS

In this chapter, we have discussed in detail the experiments and results based on the methods described in Chapter 3.

4.1 Associate degree schedule plans

Experiments. We have generated AD schedules for five majors. The database only has AD requirements and elective requirements data for two majors namely, Associate in Business DTA and Associate in Arts & Sciences We collected and entered the data for the other three AD majors, Associate of Science- Computer and Electrical engineering, Associate of Science - Pre-Engineering for Mechanical, Aeronautical, Industrial, Material science and Associate of Science - General Engineering transfer. We generated these schedules by generating students' preferences data for corresponding major and school for transfer. Table 4.1 shows the majors and school IDs for which, the preferences data was used to generate the appropriate AD schedules. The existing database doesn't have required core courses data for all possible combinations of majors and schools. Hence, the valid combinations are limited as shown in table 4.1.

We used all these combinations along with the student preferences to generate the corresponding AD schedules by varying the other preferences such as maximum credits per quarter, maximum core credits per quarter, summer quarter enrollment preference, etc. Using this, we were able to generate 450 schedules. The distribution of the generated AD schedules according to the students' major preferences is shown in the below Fig 4.1.

We have also collected data for elective courses course timings for some courses and entered their data along with some synthetic data for some elective courses in the database.

Table 4.1: Major and school combinations available for transfer in VAA database

Major ID	**Major description**	**School ID**
22	Business	6
6	Computer Science	3
1	Computer Engineering	1
1	Computer Engineering	2
16	Mechanical Engineering	1
16	Mechanical Engineering	2
20	Electrical Engineering	1
20	Electrical Engineering	2
29	Aeronautical Engineering	1
13	Civil Engineering	1
8	Chemical Engineering	1
14	Construction Engineering	1

We have entered the course timings data for 150 elective courses in the database. These elective courses are distributed among all the five AD majors. Some of the elective courses are utilized by more than one AD major as they are options used by requirements across the different majors. Table 4.2 shows an example of synthetic course timings data generated for the course ID-508, 'ENGR 201'. The title of the course is "Fundamentals of Material science" and is an elective offered in the "Associate of Science-Pre-Engineering" major. Here, we can see that this five-credit course is offered in two different timings in quarter:4 differentiated with the *SectionID*. The start time and end time are course box slots where their IDs denote the exact time of the session. For example, here 75 is 12:20:00 hrs, 91 is 15:00:00 hrs, 93 is 15:20:00 hrs and 105 is 17:20:00 hrs.

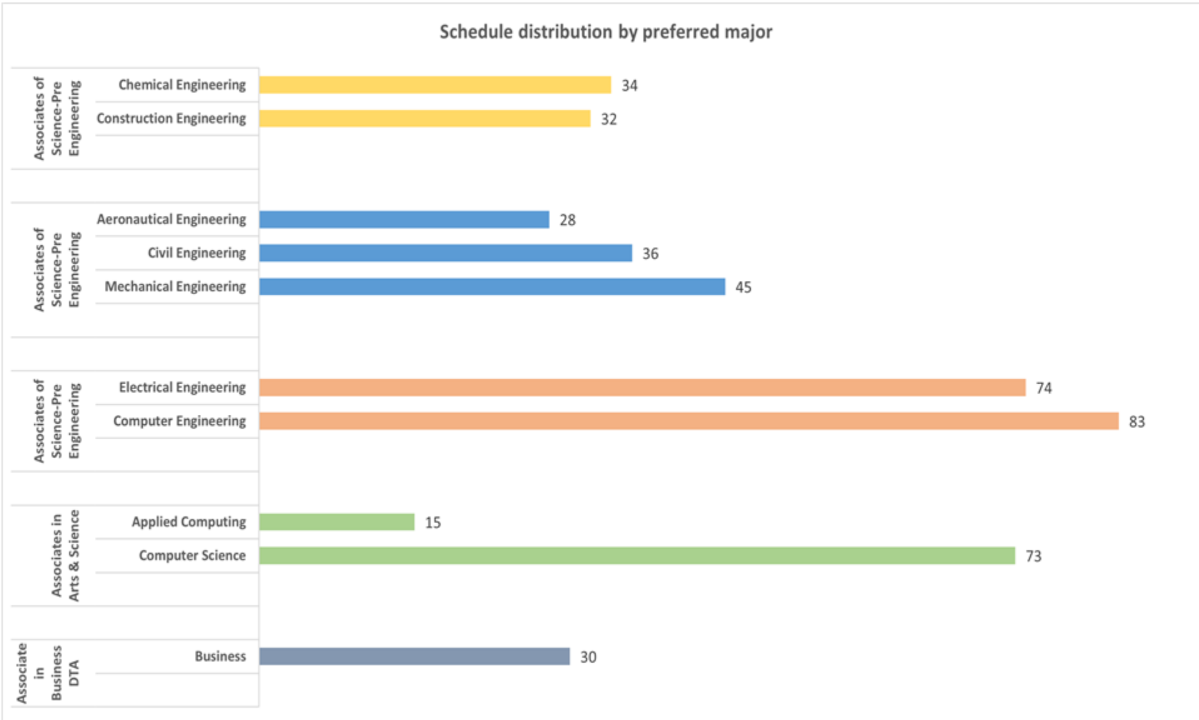


Figure 4.1: Distribution of generated AD schedules by preferred transfer major

Associate degree schedules. In this section, we look at two AD schedules generated using our algorithm and their correctness. Table 4.3 shows a generated AD schedule for Business major – Associate in Business DTA degree. The algorithm generates the schedule by taking the required core courses for the Business major and then determines the electives to be scheduled for satisfying the "Associates in Business" degree requirements. The required core courses and the electives in the generated schedule are specified in this table.

From the schedule generated in Table 4.3, we have indicated the courses that satisfy the respective AD requirement in Figure 4.2. The courses that satisfy the Associate in Business degree requirements in the generated schedule also include the prerequisite courses of the other courses and the required core courses for transferring to the major and school choice of the student. Also, in Figure 4.2, the AD requirements as specified by EvCC [2] are shown on the left-hand side, and the corresponding elective courses and their credits are shown in

Table 4.2: Example entry of elective course timing added in the VAA DB

Course ID	**Course Number**	**StartTime ID**	**EndTime ID**	**DayID**	**QuarterID**	**SectionID**
508	ENGR201	75	91	1	4	20
508	ENGR201	75	91	3	4	20
508	ENGR201	93	105	2	4	20
508	ENGR201	93	105	4	4	20

a table on the right side. Here, we can see that all the requirements for the “Associate in Business DTA” degree have been satisfied with the courses and their credits.

Table 4.4 shows the schedule generated for Associate of Science-Pre-Engineering [4] generated for a student’s preferences of ”Mechanical Engineering” major and ”University of Washington Seattle” for the target school of transfer. We can see that the courses that satisfy the AD requirements are comprised of the required core courses for transfer according to the student’s preferences of major and school and some specific elective courses that only satisfy the AD requirements.

The courses from the schedule generated in Table 4.4 that satisfy the AD requirements are indicated in the Figure 4.3. The left-hand side is the official checklist page from EvCC for the “Associate of Science degree - Pre-Engineering” requirements [4] and the right-hand side is a table that indicates the courses in the generated schedule that satisfy each requirement along with the total credits. We can see that all the requirements and the total required credits have been satisfied.

From the Tables 4.3 and 4.4, we can observe that our algorithm explained in Section 3.4 is able to generate AD schedules by determining the least number of elective courses, apart from the required core courses or the required core courses, R, to get transferred to the desired school and major. We have seen examples of two schedules of two different ADs

Table 4.3: Associates in Business degree schedule along with the plan for Business major transfer

Year	Quarter 1	Quarter 2	Quarter 3	Quarter 4
1	ACCT 110 ESL 097 ENGL 098 **EDUC& 150D**	MATH 099 **ENGL& 101** **ECON& 202**	**ACCT& 201** **ACCT& 202** **MATH& 141**	AEP 098 **ANTH& 215**
2	**ACCT& 203** **ARAB 121**	**ECON& 201**	**ENGL& 102**	**MATH& 146**
3	**BUS& 101**	**MATH& 148**	**BUS& 201**	**ART& 100**
4	**ART 110**	**CL 101**		**ASTR& 101**

Legend
- Associate in Business degree electives exclusively
- Required core courses that satisfy Associate in Business degree requirements
- Prerequisite courses that satisfy Associate in Business requirements

namely the "Associate in Business DTA" and the "Associate of Science – Pre-Engineering" in tables 4.3 and 4.4.

Rest APIs Experiment setup. The APIs were designed to take the preferences of the student and generate the appropriate AD schedule along with the required courses for transfer to the school of choice. The APIs were developed and tested in the Swagger hub framework. The APIs are hosted in Microsoft Azure web apps. The next section shows the AD plan generated for the preferences of the student.

API for Associate Degree plans. The API calls and their responses are shown below in Figures 4.4 and 4.5. Figure 4.4 shows the POST API call, which takes the preferences of the

ASSOCIATE IN BUSINESS (DTA) DEGREE REQUIREMENTS ☐ Completion of Diversity Requirement

*Must earn a C grade (2.0) or better in all required courses. Courses may be subject to prerequisites.

Course	Course Title	Credits	Grade	Quarter	Year
BASIC COMMUNICATION SKILLS -10 credits					
*ENGL& 101 or ENGL& 101D	English Composition I	5			
*ENGL& 102, ENGL& 102D	Composition II	5			
BASIC QUANTITATIVE SKILLS - 10 credits					
*MATH 138 or &141, &142, or &144 or &148 or &151 (or higher)	Applied Algebra/Precalculus Business Calculus Calculus I	5			
*MATH& 148 or &151 or higher	Business Calculus Calculus I	5			
HUMANITIES - 15 credits from the AAS DTA Humanities distribution list, with not more than 10 credits from the same discipline, up to 10 credits of foreign language with not more than 5 credits at the 100 level, and not more than 5 restricted Humanities credits. Two quarters at EvCC or two years in high school of the same world language is required for admission to all UW campuses.					
CMST& 220 (recommended)	Public Speaking	5			
		5			
		5			
SOCIAL SCIENCE - 15 credits. 10 credits in required economics classes and 5 credits other than economics from the AAS DTA Social Science distribution list.					
*ECON& 201	Micro Economics	5			
*ECON& 202	Macro Economics	5			
BUS& 101 (recommended)	Introduction to Business	5			
NATURAL SCIENCE - 15 credits. 5 credits in Math&146 -Statistics and 10 credits from the AAS DTA Natural Science distribution list. At least 5 of the Natural Science AAS DTA credits must be Part A lab credits and the remaining 5 credits can be either 5 credits from the Part A lab courses or 5 credits from the Part B non-lab courses.					
*MATH& 146	Introduction to Statistics	5			
Part A (lab course)		5			
Part A or Part B		5			
REQUIRED BUSINESS - SPECIFIC ELECTIVES (20 credits)					
*ACCT& 201	Principles of Accounting I	5			
*ACCT& 202	Principles of Accounting II	5			
*ACCT& 203	Principles of Accounting III	5			
*BUS& 201	Business Law	5			
OTHER ELECTIVE 5 credits; courses must be numbered 100 or above.					
		3			
COLL 101	College Success	2			

Courses that satisfy requirements	
Course Code	Credits
Basic Communication skills	
ENGL& 101	5
ENGL& 102	5
Basic Quantitative Skills	
MATH& 141	5
MATH& 148	5
Humanities	
ARAB 121	5
ART& 100	5
ART 110	5
Social Sciences	
ECON& 201	5
ECON& 202	5
BUS& 101, BUS& 201	5,5
Natural Sciences	
MATH& 146	5
ANTH& 215	5
ASTR& 101	5
Required business specific electives	
ACCT& 201	5
ACCT& 202	5
ACCT& 203	5
BUS& 201	5
Other electives	
MATH& 146	5
CL 101	2
Diversity course	
EDUC& 150D	3

Figure 4.2: Associate in Business degree [2] requirements and credits satisfaction in the generated schedule

student in JSON format and responds with the ScheduleID of the AD schedule generated. These API calls can be utilized by the VAA UI in the future for providing the display functionality of the AD schedules.

The schedule generated is obtained using the Get API call, which responds with the

Table 4.4: Associates of Science degree (Pre-Engineering) schedule along with the plan for Mechanical Engineering major transfer [4]

Year	Quarter 1	Quarter 2	Quarter 3	Quarter 4
1	MATH 080 ESL 097 PHYS& 114 MATH 091	MATH 092 ENGL 098 **MATH& 163**	MATH 142 MATH& 144 **ENGR 111** **PHYS& 231**	**MATH& 151** **ENGR& 114** **PHYS& 233** **ENGL& 101**
2	**MATH& 152** **MATH 260** CHEM& 140 **ENGR 121**	**CHEM& 161** **ENGR& 214** **ENGR 201**	**ENGR& 215** **MATH 261**	**CHEM& 162** **MATH& 264**
3	**ENGR& 225** **ENGL& 230** **ENGR 220**	**ENGR 240** **PHYS& 241**	**ARAB 121**	**ART 110**
4	**CS& 131**	**ECON& 201**	**EDUC& 150D** **PHYS& 242**	**PHYS& 243**

Legend	
	Associate in Business degree electives exclusively
	Required core courses that satisfy Associate in Business degree requirements
	Prerequisite courses that satisfy Associate in Business requirements

courses scheduled in each year and quarter in JSON format. Figure 4.5 shows the response for obtaining the schedule with schedule ID - 9509, which is the AD schedule generated as shown in Figure 4.2.

Associate of Science – Pre-Engineering

Mechanical, Civil, Aeronautical, Industrial, Materials Science

This checklist is targeted at transfer students with an interest in one of the above engineering majors at the University of Washington University. Students should meet with an advisor and maintain this checklist while at Everett Community College. The quarter bef checklist should be submitted with a diploma application to the Enrollment Services Office. Note: Though courses in a foreign lang Associate of Science degree, some universities may require two or three quarters of foreign language for admission or for graduation

Note: Prior to starting some or all of the following courses, students should:

- ☐ Complete ENGR 101 (formerly 109) recommended for all students considering an engineering major
- ☐ Complete ENGL 098 or earn a placement score into ENGL& 101
- ☐ Complete MATH& 144 or MATH&142 or place into MATH& 151
- ☐ Complete PHYS& 114 or physics placement test
- ☐ Complete PHYS 130 before PH
- ☐ Complete CHEM& 140 or place
- ☐ Complete ENGR 121 and PHYS
- ☐ Complete ENGR 111 and MAT

Student:__________________

❑ **COMPLETION of Diversity Course** __________ **(Where Completed/Course Title)** ______ **(Year Complet**

Course Number	Course Title	Credits	Quarter Completed
COMMUNICATIONS SKILLS (5 credits) [1]			
ENGL& 101	English Composition I	5	
MATHEMATICS (Pre-requisite Math courses may also be required.)			
MATH& 151	Calculus I	5	
MATH& 152	Calculus II	5	
MATH& 163	Calculus 3	5	
MATH 260	Linear Algebra	5	
MATH 261	Differential Equations	5	
HUMANITIES AND SOCIAL SCIENCE (15 credits, in three different disciplines. One course must be selected from Humanities Sciences. The third course may be from Humanities or Social Sciences. For acceptable courses, see course list for the Associate of S See Notes 1 and 2.)			
SCIENCE AND ENGINEERING			
CHEM& 161	General Chemistry I	5.5	
CHEM& 162	General Chemistry II	5.5	
ENGR 111 (see Note 3)	Intro to Engineering I	5	
ENGR& 214	Statics	5	
ENGR& 215	Dynamics	5	
ENGR& 225	Mechanics of Materials	5	
PHYS& 241/231	Engineering Physics I	5.5	
PHYS& 242/232	Engineering Physics II	5.5	
PHYS& 243/233	Engineering Physics III	5.5	
SPECIALIZATION COURSES (minimum 16 credits; select minimum **four** courses as appropriate for intended major and transfe page of this guide for course recommendations by intended transfer institution.)			
CS& 131	Computer Science I	5	
ENGR& 114	Engineering Graphics	4	
ENGR 121	Intro to Engineering 2: Design	5	
ENGR 201	Fundamentals of Materials Science	5	
ENGR& 204	Electrical Circuits	5	
ENGR 216	Integrated Computer Aided Design	4	
ENGR 220	Breaking Lab	2	
ENGR& 224	Thermodynamics	5	

Courses that satisfy the requirements	
Course	Credits
Communication Skills requirement	
ENGL&101	5
Mathematics	
MATH&151, MATH& 152, MATH& 163, MATH 260, MATH 261	25
Humanities	
ARAB 121, ART 110, ECON& 201	15
Science and Engineering	
CHEM&161, CHEM&162, ENGR 111, ENGR&214, ENGR& 215, ENGR& 225, PHYS& 241, PHYS& 242, PHYS& 233, PHYS& 231	53
Specialization courses	
CS& 131, ENGR& 114, ENGR 201, ENGR 240, MATH& 264, ENGR 220, ENGL& 230	28
Diversity course	
EDUC& 150D	3
Total credits	129

Figure 4.3: Associate of Science Degree - Pre-Engineering[4] requirements and credits satisfaction in generated schedule for Mechanical Engineering major

4.2 Alternate schedule Recommendation

Experiment setup. We implemented the K-means clustering model on 8968 schedules after doing the data cleaning process as discussed previously in section 3.5.1. These schedules include the previously imported schedules from Everett College and also those generated for AD majors in this project[18][33]. We have integrated the results of the recommendation framework with the VAA API, which recommends the top two schedules for the newly

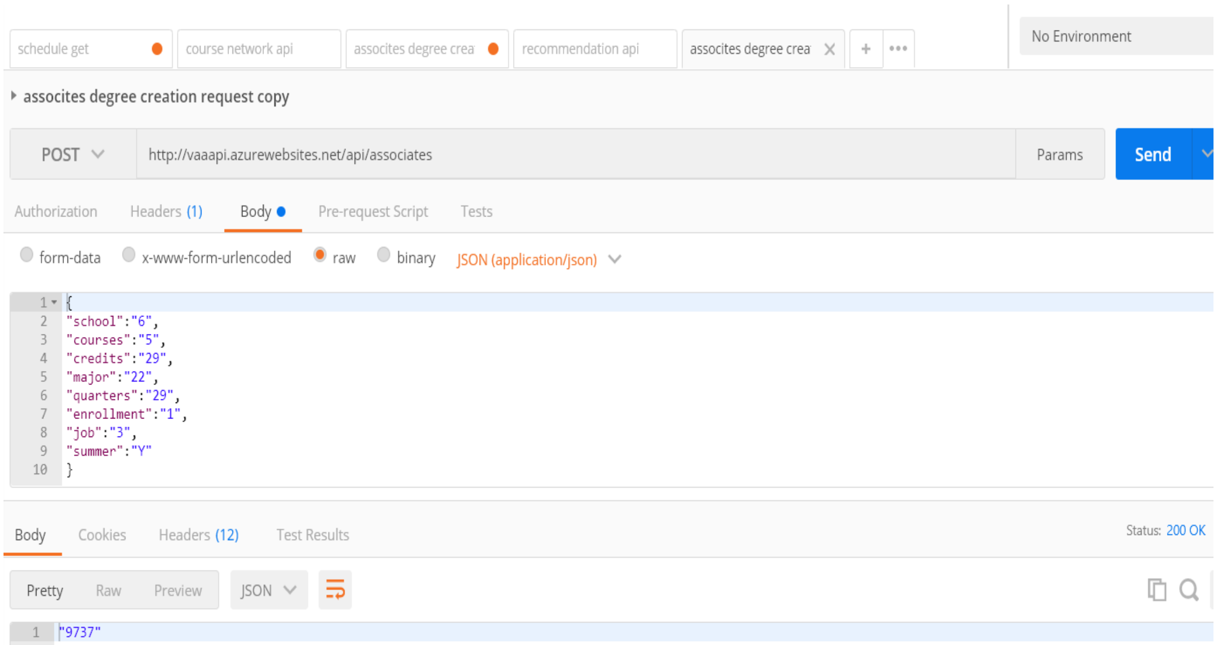


Figure 4.4: Post API call for sending the preferences of the student for generating AD schedule

generated schedule for a student. In this section, we also discuss the research and experiments done on determining the appropriate 'K' value for the dataset.

Number of clusters determination. We have used the elbow method, silhouette analysis, and Davies-Bouldin score plot to determine the closest appropriate value of 'K' in k-means clustering. We have decided the optimal number of clusters based on all the plots and their analysis. We ranged the values of 'K' between 2 to 9 and did our experiments. The description of these methods is discussed previously in section 3.4.1.

Elbow method. This is the most common approach to select the optimal number of clusters. The sum of squared distances (SSE) between the data points and the closest cluster centers are plotted by varying the K value. The idea is to identify the value of K where the SSE scores rapidly decline and produce an elbow or curve before it becomes a plateau and

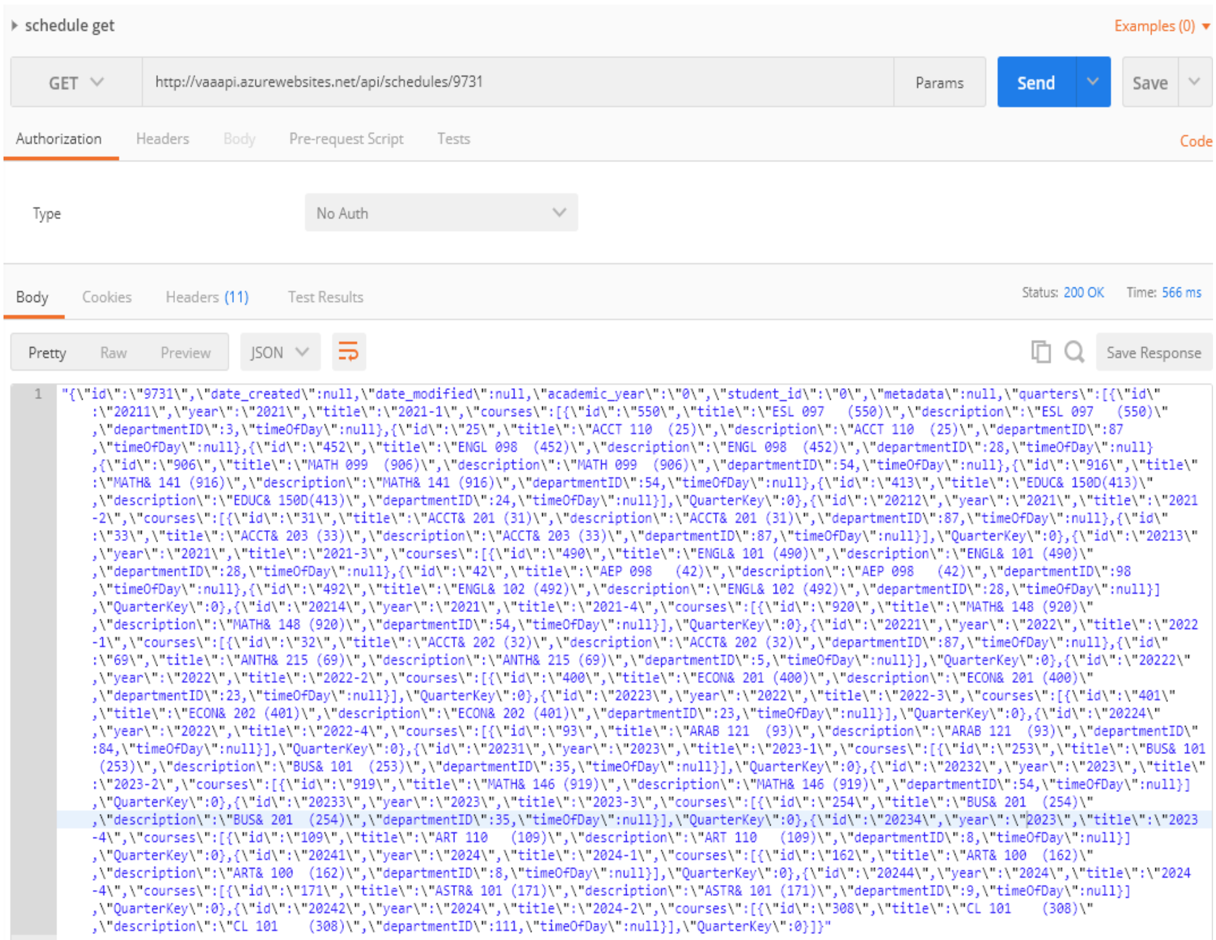


Figure 4.5: Get API call for obtaining the schedule details.

consistent [30]. Figure 4.6 shows the results obtained for the elbow method. We can see that between the values of k=3 to k=5 the curve has an elbow bend. However, we need more analysis to decide on the exact value of K between 3 and 5.

Silhouette Analysis. Silhouette Analysis is used to measure the closeness of every point in a cluster to the other points in another cluster. The ideal silhouette score is 1, which means that the points are assigned to the right cluster. When points are close to cluster decision boundaries, their score becomes 0. When points are assigned to the incorrect cluster, their

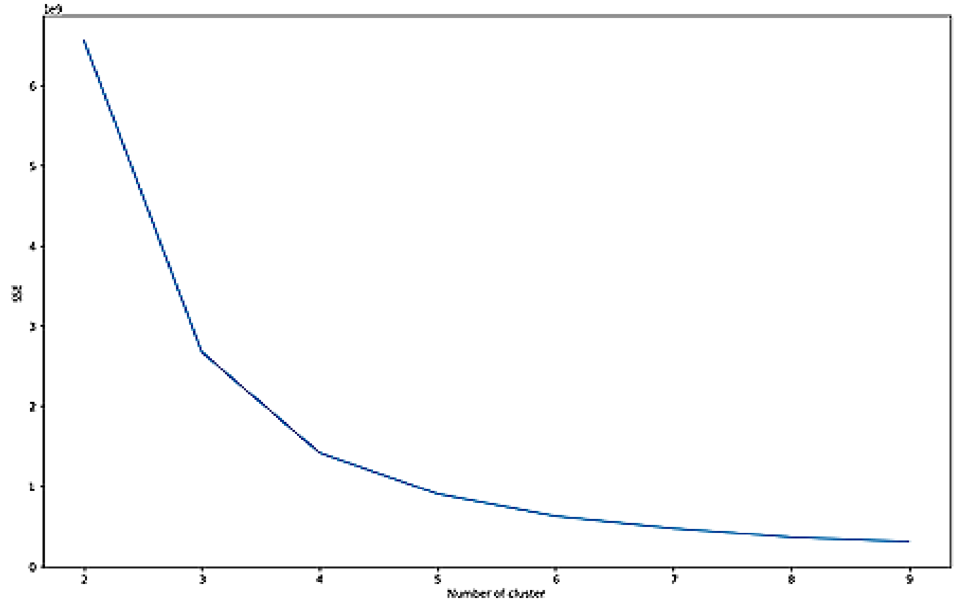


Figure 4.6: Elbow method results where the x-axis represents the number of clusters, the y-axis represents SSE values

silhouette score is -1. Figure 4.7 shows the silhouette plots for K=3, K=4, and K=5 (left to right). The dotted line in every plot shows the value of average silhouette scores. The cluster labels are on the y-axis and the silhouette coefficient scores are on the x-axis. Every cluster is differentiated by its color. A cluster's size determines the number of data points assigned to the cluster. In silhouette analysis, we look for a value of K where there are high average silhouette scores, no clusters with the maximum silhouette scores lesser than the average silhouette score, well-clustered data points where the silhouette scores of data points in every cluster are similar [30]. In Figure 4.6, we see that the K=3 has the highest average silhouette score of 0.692, however, not all data points in *cluster-2* and *cluster-0* have the same silhouette scores. For K=4, even though the average silhouette score is 0.690 the data points in all the clusters have almost similar scores. For K=5, three *clusters-0,3,4* have data points whose silhouette scores are dissimilar. Dissimilarity in a cluster here means that data points in that cluster have varying silhouette scores. Based, on this, we can say that, K=4 satisfies our requirements for optimal clustering.

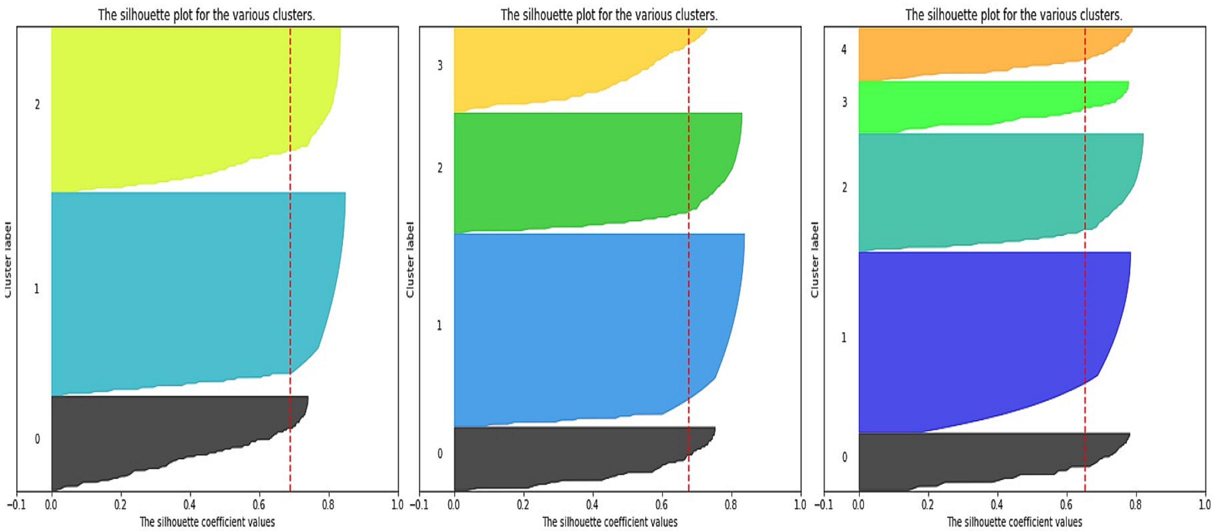


Figure 4.7: Silhouette analysis plot for k=3, k=4, and k=5.

Davies-Bouldin index plot method. The Davies-Bouldin index is the ratio between within-cluster and between-cluster distances. As discussed previously in Section 3.4.1, the optimal number of clusters has the smallest Davies-Bouldin index value[30]. Figure 4.8 shows the Davies-Bouldin index plot for a varying number of clusters. We can see that for K=4, the Davies-Bouldin index is the least. Based on the results from plots and analysis of elbow method, silhouette analysis, and Davies-Bouldin index, the optimal number of clusters in our case is K=4.

K-means clustering results. The K-means clustering model was built by generating four clusters. The existing dataset after the data preprocessing and cleaning process with 8968 schedules from various majors was used to run the clustering model. Table 4.5 shows the results of the majors that were clustered together in each clustering label. The AD majors clustered in every cluster label are also listed. From table 4.5 we see that Electrical engineering and Business majors have clustered together as some Math courses are similar

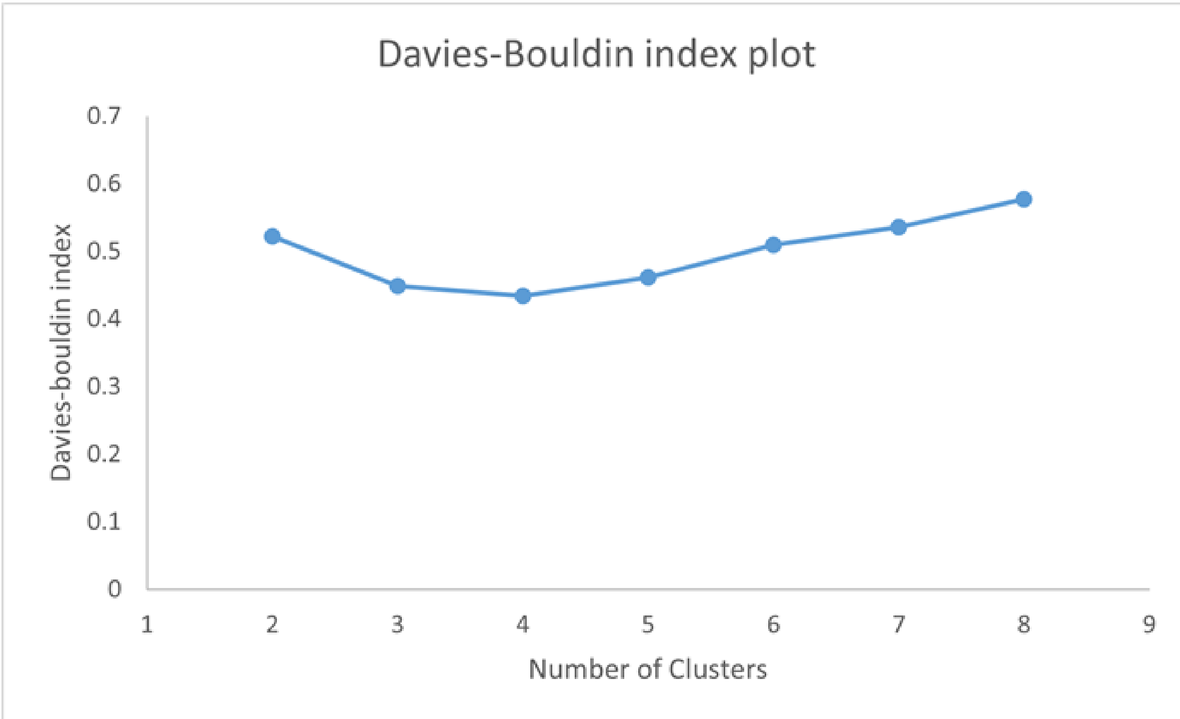


Figure 4.8: Davies-Bouldin index plot for different 'K' values.

in both these majors. Similarly, some elective courses especially Math related courses are similar in Associates of Science-Pre-Engineering (Computer and Electrical Engineering) and Associates of Business DTA due to which we notice that these schedules have clustered together.

4.3 Summary of Results

Apart from the AD academic plan generation and the recommendation framework, we also ensured the end-to-end functioning and maintenance of the VAA system. The below sections detail the work that was involved during this project. Every situation was handled by answering the industry standard "Precision Q&A" to improve communication efficiency and critical thinking[35]. These discussions were done during the team scrum meetings, where ideas and suggestions were shared among the VAA team and the decision was then finalized.

Table 4.5: Feature comparisons of the previous and new implemented system

Cluster label/ID	Majors	Associate degree majors
1	Computer Engineering, Applied Computing, Computer science, and Software engineering	Associates of Science-Pre-Engineering (Computer and Electrical Engineering), Associates in Arts and Sciences-DTA
2	Pre-Engineering, Electrical Engineering, Business	Associates of Science-Pre-Engineering (Computer and Electrical Engineering), Associates of Business DTA
3	Civil Engineering, Mechanical Engineering, Construction Engineering, Aeronautical Engineering, Chemical Engineering	Associates of Science-Pre-Engineering (Mech, Civil, Aero, Industrial, Materials Science)
4	Nursing, General Major, Business, Construction Engineering, Chemical Engineering	Associates of Business DTA, Associates of Science-General Engineering Transfer

Database Maintanence. The MSSQL server database on the HostTek hosting account has backups scheduled every two weeks, which runs usually fine. However, we also experienced a situation when the database took up most of the space in the server and crashed. This was probably because the database was being actively used and consumed a lot of RAM space, which was 2GB and considerably lower for the amount of data that was stored in the database. The usual backups also did not happen properly for a few weeks and this went unnoticed. Since, this is a paid service with HostTek, a third-party server, they were responsible for handling this issue. Even though they brought the server up, they could not start the MSSQL database. This was because the backup that was effectively saved in the system dated back two months and the current state of the database was not known to have been saved properly. In this situation, we came up with the idea of using the .mdf and .ldf files, which contain the main data of the database and the transactional logs of the changes to the database respectively. We decided to migrate these files to a new server instance with the same configuration and start the MSSQL server database using these files. Further, we proceeded with the set-up of DNS propagation with the static IP address for the new server. We proceeded with these steps after brainstorming and analyzing the possible ideas with the VAA team for mitigation. Also, as a mitigation plan, we took a local copy of the .mdf and .ldf files along with the latest backup of the database available so that we can recover the database in an unexpected event during the migration.

Chapter 5
CONCLUSION AND FUTURE WORK

This section discusses the achievements of this work and our research contributions as well as identified and known limitations in the system, and possible areas for future work.

5.1 Conclusion

The overall goal of the VAA recommendation system is to improve the academic advising experience of university entry-level students like those in community colleges. The way to do this is by automatically generating schedules and study plans for the academic advisors/faculty and their students; by reducing the time spent in this time-consuming task, the advising conversation can be more personal and meaningful. The VAA is a complex software system, consisting of various independently-implemented modules. The primary function of the VAA software system is the schedule generation functionality.

Working with real users

Academic advisors at community colleges are one of the primary target users of this project, the other –longer term– target being students. One of the most intricate yet important tasks in terms of academic planning is the creation of an Associate Degree (AD) schedule, making the automation of this process a valuable and highly anticipated feature of the VAA system. This functionality will also help community colleges to better serve the students by encouraging them to complete the AD, which is a degree that can be granted by the community college. Hence, when students complete ADs, the community colleges are directly benefited because granting such degrees give them better ranking, more recognition and thus, budget. Given this motivation, one of the major goals of this project was to implement the schedule generation functionality for AD majors and to incorporate elective courses (which are critical for AD schedules).

Throughout this work, we were able to generate 450 academic schedule plans for the AD majors. This achievement required in-depth research to investigate and understand requirements and credits satisfiability conditions for ADs as given by a community college like EvCC. In-depth research was required to understand the Direct Transfer Agreements (DTA) between community colleges and some universities with certain majors. As part of the development aspect of this work, we were able to interact directly with our main users (academic advisors) and gather data regarding the AD requirements and credit satisfiability.

A critical aspect of this work was the implementation of the functionality to incorporate electives into the schedule generation process. It's worth mentioning that this feature was not even considered in early design and implementation versions of the VAA due to it's inherent complexity. While the VAA database already had table structures to support entries for electives and associated requirements, none of the associated functionality to incorporate and use electives (critical to DA generation) was in place. Existing implementations only considered required core courses for transferring to a target major/university (the student's fundamental preferences) for generating schedule recommendations.

We were also able to interact with EvCC academic advisors/faculty and get insights regarding course box slots of the electives in comparison to the core courses and generate the elective course timings to enter into the database. We also researched and collected data for three AD majors and entered them across the appropriate tables in the database. Using this, we were able to generate valid AD schedules for five majors based on students' preferences of major and transfer school.

Working with previous efforts

In this work, were able to leverage the existing Genetic Algorithm (GA)[18] scheduling strategy to incorporate the use of elective courses to satisfy all the requirements of the ADs for schedule generation. In addition, UI functionality was enabled to obtain a schedule from the database and render it to the UI, this was done through he development of REST API on Azure Web Apps[12]. As mentioned before, even though the database had the structure to store the elective courses to satisfy requirements for the AD, it was lacking various types and amounts of data, in particular, the database lacked course timings data, critical to test the scheduler module. We have created and made available documentation about system's previously existing functions as well as the newly implemented functionality for generating AD schedules.

Summary of Contributions

The intended goals to research, design an implement an algorithm to incorporate the use of elective courses and generate AD academic plans have been achieved. The implementation of a corresponding recommendation module and its results have incorporated into the VAA system.

A critical research aspect of this work has been to formalize abstractions of various concepts that were only known and communicated as "tribal knowledge" in the past. This has been an obstacle not just in the context of the community college when ramping up new academic advisors and describing processes, but it has also complicated the design and development of decision making aspects of the algorithms in the VAA recommendation engine module. By mapping such ideas and processes into a set diagram, and formally defining them using set notations and operations, we were able to create a general algorithm that formalizes the process of creating academic schedules, not only for ADs but also in general. This algorithm can be used as a building block in the future to implement strategies, other than the ones presented here, to create schedule recommendations or to formalize the plan-creation process by humans (advisors) in the community college.

Another paramount aspect of this research was the analysis, evaluation and search for an appropriate value of 'K' in the K-means clustering model in the recommendation framework. This module is in charge of recommending alternate schedules that allow students to explore other degree and school options based on their original preferences, giving them the option to navigate and understand the requirements for their desired major/school combination compared to other similar options (for instance, ”Computer Science” vs. ”Software Engineering”), thus better understanding the expectations of different educational paths and then make educated decisions about their academic choices. The results of the recommendation module have been integrated with the VAA system using the aforementioned APIs.

Limitations and Opportunities for improvement

Apart from the contributions discussed above, certain limitations were identified in the existing system. The following list is a summary of such limitations:

- The database only contains required core courses to get transferred for six majors and four schools.
- The data is heavily skewed for the school option, “University of Washington” AD schedules. This data becomes narrower when considering University of Washington along with the major options available in the database. School and major options are preferences of the student based on which schedules are generated.
- Elective course timings are incomplete, not all elective course options are available, which narrows down the possible combination of courses that can be scheduled when generating AD schedules. Even though we have generated and entered the data for some of the elective course timings, it is not sufficient for generating AD schedules.
- The required core courses are only available for twelve combinations of majors and schools. Implemented scheduling algorithms can only use these combinations regardless of input preferences. These twelve major and school combinations are used in the recommendation framework to create schedules. Since the data is limited, the clustering model is skewed.
- The recommendation framework model should be tested on schedules with a variety of major and school combinations.
- It is currently unclear how the model would adapt to the use of a bigger dataset and

its fluctuations.

Moving forward, the VAA system can be enhanced and improved by considering these limitations for further development and testing.

5.2 Future Work

Moving forward, there are plenty of opportunities and aspects to improve the VAA system. In this section, we discuss some of the possible areas of improvement.

Data Collection. In the area of data collection and generation, currently there is very limited data for core and elective courses across various major and school combinations. Due to this, testing on data using various majors and school preferences is restricted. Gathering and collecting data from various community colleges apart from EvCC can be used to improve the efficiency of the scheduling algorithms implemented in the system.

Smart Elective Selection. The generated elective courses data needs to be validated before generating schedules for students, which would improve the accuracy of generated schedules in real-time.

In our research and implementation of AD scheduling, we have chosen elective courses to generate schedules randomly, however, it would be more beneficial for automated scheduling if the electives can be determined by using ML based on the history of AD schedules completed by the other students. A better choice of electives can also be determined by looking for courses having shorter distances in the prerequisite graph/network, or by considering the minor areas of interest of the student as a preference. Courses related to the preferences of a chosen major and minor preference could also be better choices.

Alternative ML Models. Apart from AD scheduling, ML and natural language processing can be used to choose a suitable schedule based on the preferences of the student and priority of other input features beyond student preferences, such as Math sequence break, Math starting point, etc. Regarding the recommendation framework, apart from k-means clustering, other unsupervised learning models and methods could be explored by focusing on feature engineering and the right combination of features to ensure better recommendations. Thus, the VAA system has areas of modifications and improvement to provide a better academic advising experience for students and their advisors.

Software Engineering. To establish an end-to-end system, APIs will have to be tested with the appropriate framework to ensure proper interactivity of the VAA system backend with the user interface. Currently, APIs don't have integration testing implemented for developers to test them before deploying. This can help with refactoring to handle errors and exceptions during development and before deployment. The APIs can be further consumed and utilized by the UI to give more interactivity features for the users.

BIBLIOGRAPHY

[1] Associate Certificates and Degrees, Everett Community College, WA. url="https://www.everettcc.edu/programs/certificates-and-degrees".

[2] Business DTA Transfer, Associate in Business, Everett Community College, WA. "url = https://www.everettcc.edu/files/programs/busadmin-transfer.pdf".

[3] Chapter 9 Recommendation Systems, Stanford. "url=https://infolab.stanford.edu/ ullman/mmds/ch9.pdf".

[4] Engineering Transfer, Associate of Science-Pre-Engineering, Everett Community College, WA. url="https://www.everettcc.edu/files/programs/engineeringtransfer.pdf".

[5] K-Means clustering. "url=http://scikit-learn.org/stable/modules/clustering.html.2018".

[6] Selecting the number of clusters with silhouette analysis on kmeans clustering — scikit-learn 0.19.1 documentation.

[7] Silhouette method - better than elbow method to find optimal clusters. medium.

[8] Peoplesoft Degree Audit. "url=https://www.sbctc.edu/resources/documents/colleges-staff/it-support/sms/smsoverview.pdf", 2012.

[9] Starfish, Hobsons Starfish Retention Solutions. "url=https://www.starfishsolutions.com/", 2015.

[10] Southwest Tennessee Community College: Texting to Increase Retention and Enrollment. "url = https://www.signalvine.com/case-study/southwest-tennessee-community-college-texting-to-increase-retention-and-enrollment", 2018.

[11] Washington State Guided Pathways. "url=https://www.sbctc.edu/colleges-staff/programs-services/student-success-center/guided-pathways-advisory-council.aspx", 2018.

[12] Microsoft Azure. "url=https://azure.microsoft.com/en-us/services/app-service/web/", 2019.

[13] Microsoft data platform-SQL server. "url=https://www.microsoft.com/en-us/sql-server/sql-server-2019", 2019.

[14] Optimize : Plan for Future Demand Term by Term. "url : https://www.edunav.com/optimize/", 2019.

[15] Phases of the Software Development Life Cycle. "url=https://harness.io/blog/devops/software-development-life-cycle/", 2021.

[16] Silhouette (clustering). "url=https://en.wikipedia.org/wiki/Silhouette_(clustering)", May 2021.

[17] SwaggerHub-API Design and Documentation with OpenAPI Access. "url=https://swagger.io/tools/swaggerhub/", 2021.

[18] Rigdha Acharya. Optimal Community College Academic Schedule Recommendation Framework using Genetic Algorithm with Simulated Annealing and k-Means Clustering. June 2020.

[19] Qaif Amaan Shaikh. System Reengineering and Continuous Integration of the Virtual Academic Advisor System. June 2020.

[20] B. Beaudin and J. Breiner. Academic Advising at a Distance: Student Communication Preferences. *Int. J. Continuing Engineering Education & Lifelong Learning*, 11(1-2):128–134, July 2001.

[21] Clive Belfield. The economic benefits of attaining an associate degree before transfer: Evidence from north carolina. 2013.

[22] Jacek Błażewicz, Wolfgang Domschke, and Erwin Pesch. The job shop scheduling problem: Conventional and new solution techniques. *European journal of operational research*, 93(1):1–33, 1996.

[23] Gates Bryant. The Supplier Landscape. *Driving Towards Degree - The Evolution of Planning and Education in Higher Education*, March 2015.

[24] Gates Bryant. The Evolution of Planning and Advising in Higher Education. *Part 2: Supplier Landscape*, March 2019.

[25] Krista Carver. Student Success Best Practices: The Perfect Combination. "url = https://www.avisoretention.com/why-aviso-retention", May 2019.

[26] Dianne T. Castor. Challenges for Two-Year College Advisors, June 2005.

[27] Brent D Cejda and Jeff H Rhodes. Through the pipeline: The role of faculty in promoting associate degree completion among hispanic students. *Community College Journal of Research and Practice*, 28(3):249–262, 2004.

[28] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[29] Peter Michael Crosta and Elizabeth Mary Kopko. Should community college students earn an associate degree before transferring to a four-year institution? 2015.

[30] Georgios Drakos. Silhouette analysis vs elbow method vs davies-bouldin index: Selecting the optimal number of clusters for kmeans clustering, Mar 2020.

[31] Pamela M. Golubski. Online Academic Advising. *Encyclopedia of Information Technology Curriculum Integration*, February 2008.

[32] Teofilo Gonzalez and Sartaj Sahni. Open shop scheduling to minimize finish time. *Journal of the ACM (JACM)*, 23(4):665–679, 1976.

[33] Rashi Goyal. Building a Machine Learning Based Recommendation Engine for the Virtual Academic Advisor System. June 2018.

[34] Thomas J. Grites, Marsha A. Miller, and Julie Givans Voler. *Beyond Foundations: Developing as a Master Academic Advisor.* John Wiley & Sons, September 2016.

[35] Jd, By, and Jd. Precision Questions and Precision Answers: The Most Powerful Exec Communication Framework. "url=https://sourcesofinsight.com/precision-questions-and-precision-answers/", Sep 2020.

[36] Henry T Kasper. The changing role of community college. *Occupational Outlook Quarterly*, 46(4):14–21, 2003.

[37] George D. Kuh, Stanley O. Ikenberry, Natasha Jankowski, Timothy Reese Cain, Ewell, Pat Hutchings, Jillian Kinzie, and Timothy Reese Cain. *Using Evidence of Student Learning to Improve Higher Education.* December 2014.

[38] Heng Luo, Changyong Niu, Ruimin Shen, and Carsten Ullrich. A Collaborative Filtering Framework based on both Local User Similarity and Global User Similarity. *Machine Learning*, 72(3):231–245, September 2008.

[39] Nathalie Mainland. Introducing Salesforce Advisor Link, A New Salesforce.org Solution Designed for the Art of Advising. December 2017.

[40] Ronald C. McArthur. Faculty-Based Advising: An Important Factor in Community College Retention. *Community College Review*, 32(4):1–18, March 2005.

[41] Andrei Morozov. Virtual Academic Advisor implementation of Corequisites and Electives in database. August 2020.

[42] Jane E. Pizzolato. Complex Partnerships: Self-authorship and Provocative Academic-Advising Practices. *NACADA Journal*, 26(1):32–45, March 2006.

[43] Tye Rattenbury, Joseph M Hellerstein, Jeffrey Heer, Sean Kandel, and Connor Carreras. *Principles of data wrangling: Practical techniques for data preparation.* " O'Reilly Media, Inc.", 2017.

[44] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *ACM Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994.

[45] Toby Segaran. *Programming collective intelligence: building smart web 2.0 applications.* " O'Reilly Media, Inc.", 2007.

[46] Alan Seidman. The Evaluation of a Pre/Post Admissions/Counseling Process at a Suburban Community College: Impact on Student Satisfaction with the Faculty and the Institution, Retention, and Academic Performance. *College and University*, 66(4):223–32, 1991.

[47] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized Recommendation in Social Tagging Systems Using Hierarchical Clustering. In *ACM Conference on Recommender Systems*, pages 259–266. ACM, 2008.

[48] M Steele. The challenges of advising science technology, engineering, and mathematics students. *Academic Advising Today, 31 (3)*, 2008.

[49] Margaret Steele. technology for academic and drop-in advising (project tada!).

[50] Margaret Steele. The Challenges of Advising Science, Technology, Engineering and Mathematics Students, September 2008.

[51] Jennifer Varney. Intrusive Advising. September 2007.

[52] Sri Venkata Bhavani Likitha Vijjapu. Machine Learning based Recommendations to aid Educational Planning and academic advising through the Virtual Academic Advisor System. June 2019.

[53] E. Vance Wilson. A Standards Framework for Academic e-Advising Services. *International Journal of Services and Standards*, 2004.

[54] Hostek www.hostek.com. Hostek - shared, vps, & reseller hosting provider, Mar 2018.

[55] Yi (Leaf) Zhang. An Overlooked Population in Community College: International Students' (In)Validation Experiences With Academic Advising. *Community College Review*, 44(2):153–170, April 2016.

[56] Tianqi Zhou, Lina Chen, and Jian Shen. Movie recommendation system employing the user-based cf in cloud computing. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 2, pages 46–50. IEEE, 2017.