[Future Computing and Informatics Journal](https://digitalcommons.aaru.edu.jo/fcij)

Volume 6
Issue 2 (2021) *Issue 2*

Article 4

2021

Enhancing Academic Advising in Credit Hours System using DSS

Alaa Salah ElDin Ghoneim
Canadian International College, Egypt, alaa_salah@cic-cairo.com

Recommended Citation

Ghoneim, Alaa Salah ElDin (2021) "Enhancing Academic Advising in Credit Hours System using DSS," *Future Computing and Informatics Journal*: Vol. 6: Iss. 2, Article 4.
DOI: http://doi.org/10.54623/fue.fcij.6.2.4
Available at: https://digitalcommons.aaru.edu.jo/fcij/vol6/iss2/4

Future Computing and Informatics Journal

Volume 6
Issue 2 (2021) *Issue 2*

Article 4

2021

Enhancing Academic Advising in Credit Hours System using DSS

Alaa Salah ElDin Ghoneim
Canadian International College, Egypt, alaa_salah@cic-cairo.com

Recommended Citation
Ghoneim, Alaa Salah ElDin (2021) "Enhancing Academic Advising in Credit Hours System using DSS," *Future Computing and Informatics Journal*: Vol. 6 : Iss. 2 , Article 4.
DOI: http://doi.org/10.54623/fue.fcij.6.2.4
Available at: https://digitalcommons.aaru.edu.jo/fcij/vol6/iss2/4

Future Computing and Informatics Journal

Homepage: https://digitalcommons.aaru.edu.jo/fcij/

doi: http://Doi.org/10.54623/fue.fcij.6.2.4

Enhancing Academic Advising In Credit Hours System Using Dss

Alaa Salah ElDin Ghoneim[1, a], **Salah ElDin Ismail Salah ElDin** [1, b], **Mohamed Sameh Hassanein** [2, c]

[1] Business Information Systems, Faculty of Commerce and Business Administration, Helwan University, Egypt, [2]Faculty of Commerce and Business Administration, Helwan University

[3]Integrated Thebes Institutes for Computing and Management Science

[a] alaa_salah@cic-cairo.com, [b] salahialy@gmail.com,
[c] mohamed.sameh@thebes.edu.eg

ABSTRACT

Academic advising plays a vital role in achieving higher educational institution's purposes. Academic advising is a process where an academic advisor decides to select a certain number of courses for a student to register in each semester to fulfil the graduation requirements. This paper presents an Academic Advising Decision Support System (AADSS) to enhance advisors make better decisions regarding their students' cases. AADSS framework divided into four layers, data preparation layer, data layer, processing layer and decision layer. The testing results from those participating academic advisors and students considered are that AADSS beneficial in enhancing their decision for selecting courses.

Keywords: Decision Support Systems; Rule based systems; Advising Decision Support Systems.

1. INTRODUCTION

Nowadays, a robust advising system is a backbone for any higher educational institution using a credit-hours system. The Credit-hours system is done by achieving a number of hours for any student for graduation requirements. The usual advising process is done by the presence of the student and his/her academic advisor in order to take the best decision for the student. In any academic institution now, there is a little number of advisors comparing to the number of enrolled students, and regarding that, the advisor's decision is a human decision. The advisors are overwhelmed with advising requests, because of these numbers of students waiting for a list of courses to register in a semester.

The academic advising process in many institutions today has much significant inefficiency, which leads to high levels of student dissatisfaction (Choudhari 2009). The quality of the advising process plays a vital role in a student's academic life and in the education process as a whole. Accessing the needed information becomes a major obstacle to give the right advice. That means that the advisor and the student need to communicate well to reach out to the right decision, as it is a face-to-face consulting process. By the time, the academic advising process could be complex and time-consuming for both students and advisors, due to the difficulty of obtaining the needed information that will result in the final decision for the upcoming semesters.

Decision Support Systems (DSS) are developed and implemented in many business areas to enhance the decision-making process. The decision-making process is a process of recognizing a decision, gathering data, and assessing alternative decisions. The most common areas where DSS implemented is medical and business areas. Moreover, in the late decades' researchers found that DSS can be developed and implemented in the academic area to help the students, teachers or managers in supporting their decisions.

2. RELATED WORK

Youssef and Abdleati, 2019 developed a system in form of rules and constraints using Prolog programming language. In this research, the used framework was able to conduct a study plan for all students and course schedule, even if there are complex cases. It also responds and answers all questions and queries from the staff, advisors and students.

Afify, 2017 proposed a web-based model to perform an advising system. This advising system includes the processes of complaint, evaluate and suggest. The complaint process enables the student to complain about his intuition for the next semester, this intuition goes to the advisor to solve it. If it is not solved by the deadline, this complaint directly sent to a staff member and head of the department. The evaluation process is a report filed by students and these reports are sent directly to the head of the department. Suggestion process where a student can add any suggestion about any attended course, the suggestions are viewed by the advisors and the head of the department and they have the ability to respond for it.

Another model was proposed Mohamed, 2016 using DSS to ensure the quality of academic advising. The target was to increase the quality of advising to become more evident during freshman year, for students with bad academic records, besides students who may be at risk for many reasons. The study based on two previous studies where the one was in (Young-Jones et al., 2013), the other one in (Beal and Noel, 1980; Cuseo, 2003). This interactive DSS course scheduling is a three-phase method that allows decision-makers to interactively participate and have more control over the process of course advising.

While Amin and Fekry, 2015 created a new framework for E-academic advising to help the enrolled students and advisors make better use of the student information system by taking into consideration the importance of face-to-face contact between advisor and students. The researchers had two main problems which are the advisors unknowledgeable and the perceived differences between what the students tend to have and what the advisor recommend courses. The authors find that this framework won't replace human advisors, but will support academic decisions. They found also, after testing on two consecutive academic years that in the second year the percentage of using this framework is higher.

Yeh, 2015 built an advising system using rule-based technology to provide each student with a course recommendation list. The proposed framework is divided into three layers, a data layer stores all relevant data to be used for processing. In the processing layer, two modules are executed, an interest analysis module to store and analyze student's transcript and a comparison module to predict course grades based on past grades. The solution layer guides the student by generating recommendation courses, then the student evaluates them to identify the efficiency of the system. The finding after testing this framework is that it achieved about 80% satisfaction from students towards courses recommended and its rank in the recommendation list.

Laghari, 2014 developed a system using Java programming language to guide students in selecting courses that are suitable to be registered. Courses selection is based on the knowledge area for each course, how many courses are dependent on the courses, course status and how many courses will be offered next semester depend on it. All the fields are prioritized in the system with the first field has the highest priority. Backstage the system ranks all remaining courses and the highest courses ranks are chosen for the student to register.

AbdulWahab, et al, 2014 tried to improve academic advising by building a new intelligent DSS system. The used framework is built to avoid manual errors that are resulted from the process of academic advising. The proposed system has four units, Plan Chart Unit (PCU), Study Plan Unit (SPU), Intelligent Mark Presumptions (IMP) and Decision Course Unit (DCU). The PCU provides all data needed by the advisor related to the student. SPU improves the academic advising process and helps students and advisors to do their tasks. The IMP represents the expected marks for the registered courses to reach the required GPA. The DCU a decision tree algorithm is used to take decisions for dropping the courses. An empirical test is done on 20 students, and it was found that the ICD system can be applied to help both the students and the advisors.

Almutawah, 2014 designed a DSS for the academic advisors to support the advising system and improve the decisions of the academics based on student's historical data. The researcher used an incremental model to build the framework by collecting information and knowledge using interviews with advisors, then analyzed these interviews to create multiple models and give each model a weight. After applying the new system, it was found that there is a rational increase in completing the requirements and fulfilment of courses that lead to an increase in improving student's GPA.

3. RESEARCH METHODOLOGY

Design Science Research (DSR) is a rapidly evolving methodology where research method or technique depends on designing and developing the solutions through building artifacts and finally performing an evaluation to improve their functional performance (Vaishnavi, et al, 2019) (Hevner, et al., 2004). DSR model consists of five steps:

1. Awareness of the problem
2. Suggestion
3. Development
4. Evaluation
5. Conclusion

3.1 Awareness of the problem:

A number of interviews were conducted through two aspects, the first aspect is the problems related to advising process facing the students. The second aspect is related to different students' cases handled by experienced advisors, such as academic probation, and potential graduate students' challenging cases. These interviews also, helped us to extract some rules from the advisors' accumulated experience in dealing with academic advising rare cases that are not directly stated in the school's bylaw.

A literature review was conducted to induce the approaches and methods used in academic advising process and the problems facing the students and academic advisors. The findings resulted from the literature review led to create a tentative design framework that enhances the advising decision-making process. AADSS is proposed to facilitate taking the right decision through helping the advisors and the students in courses registration process.

3.2 Suggestion:

DSS architecture is composed of sub-systems that include Database Management Sub-system (DBMS) where establishing, updating, and querying a database, Model Management Sub-system (MBMS) where managing a DSS model base, User Interface Sub-system that allows bidirectional communication between the system and its user, and Knowledge Base Sub-system (KBMS) supports and enhances other sub-systems' components operations (Turban et al, 2007).

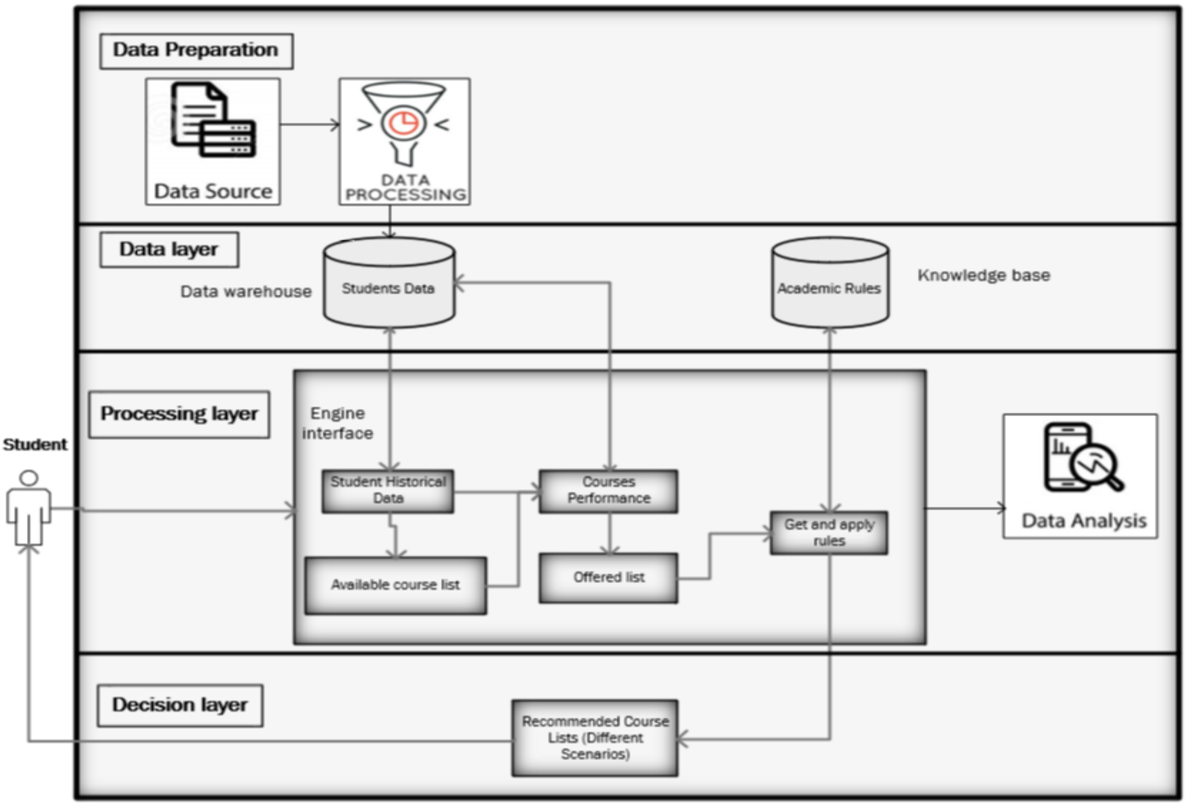


Figure 1: AADSS framework

3.3 Development:

An artifact is developed and implemented as a desktop solution to support the required capabilities for a DSS solution such as enhancing advisors' decisions, being user friendly, fostering high-demand decision making, facilitating the implementation of decisions, and supporting in decision making (Aina, 2015). Visual Basic (VB) programming language is used to implement this tool besides using MS SQL Server where the student's data is stored on. The AADSS framework is divided into four layers as shown in Fig 1. Theses layers are:

1. **Data preparation layer**

Data is collected and loaded into a data warehouse.

2. **Data layer**

Student historical data are stored in data warehouse and academic rules are stored in KB.

3. **Processing layer**

Engine interface will analyze student historical data and extract the appropriate rules based on student academic status.

4. **Decision layer**

The implemented framework generates a number of scenarios, where each scenario contains a recommended list of courses to help the advisor to pick one of the suggested scenarios that suits the student advising case.

3.3.1 Data preparation layer

This layer focuses on three main processes: extraction, transformation and loading. Extraction is the process of identifying and collecting relevant data from a real students' dataset of Canadian International College (CIC), Business School, Zayed Campus including 30182 academic records belong to 1144 students between the academic semesters Fall 2013 and Fall 2018, academic rules for the concerned school, plus the acquired knowledge from interviews with experts in the educational sector, experts in advising process, literature review, etc. The extraction process was important to overcome obstacles found in the data collected such as data integration, data incomplete, and duplication.

Transformation is the process of converting data into consistent formats to be latterly used for reporting and analysis by the implemented framework. Moreover, Data transformation process was used to define the applied business logic for data mapping in order to ensure consistency across the implemented framework.

Loading is the last phase of the ETL process, where the data are loaded into target warehouse.

3.3.2 Data layer

In this layer, students' transcripts are stored in the DW, and the academic rules are stored in the KB. So that, all relevant information is extracted from DW to be furtherly used in the processing layer. The information related to the processed student registration is collected from:

1. Student's transcript which is a form that contains student previous taken courses with its grade and points in each year and semester.

2. Advising rules that advisor go through such as indicates how many hours should a student take per semester.

3. Student's major sheet that shows courses prerequisite for each course.

After analyzing this information, the following are the main pillars to construct the appropriate advising decision to the student: Know current student status (progressing, potential graduate, …. etc.)

1. Review student passed and failed courses
2. Know the remaining courses in student's major sheet
3. Know the available courses for the current semester
4. Know the minimum and maximum number of hours to be registered.

Academic registration knowledge is obtained from the school bylaw and meeting notes explaining clearly applied school bylaw, while knowledge related to challenging cases is obtained from meeting notes with experts who face stumbling or under probation cases that need human interference.

In this framework, the knowledge is implemented using IF-Then rules, here are some examples for these rules:

R1: Allow overload credit hours for a student

If student's CGPA is greater than 3.0, Then up to 21 hours are recommended.

R2: Priority to failed courses

If a course code is in failed list and exist in offered list, Then this course code will be added in recommended list.

R3: Under academic probation status

If student's CGPA is below 2.00 and count of consecutive semesters = four, Then flag of academic probation in ON.

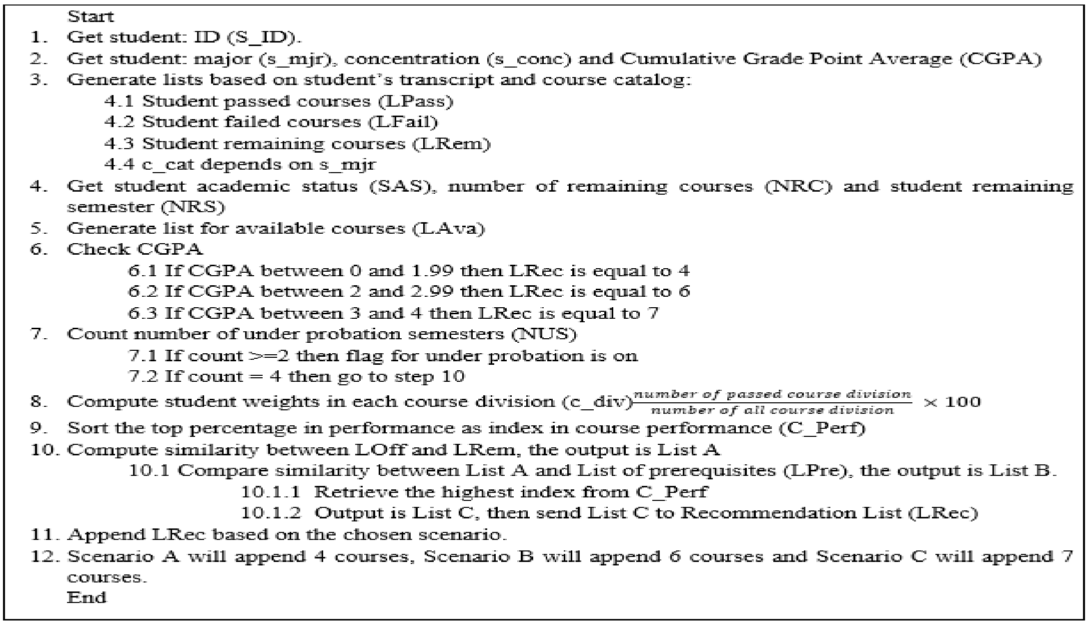
R4: Priority to course division and prerequisites.

If performance index is INFO, Then First course in recommended list is INFO course.

R5: Number of semesters restriction for registering Graduation Project

If student is in semester eight, then the student can register graduation project.

Using the extracted knowledge and implemented rules in KB, resulted in the formation of the following algorithm shown in Fig. 2.



```
     Start
1.   Get student: ID (S_ID).
2.   Get student: major (s_mjr), concentration (s_conc) and Cumulative Grade Point Average (CGPA)
3.   Generate lists based on student's transcript and course catalog:
         4.1 Student passed courses (LPass)
         4.2 Student failed courses (LFail)
         4.3 Student remaining courses (LRem)
         4.4 c_cat depends on s_mjr
4.   Get student academic status (SAS), number of remaining courses (NRC) and student remaining
     semester (NRS)
5.   Generate list for available courses (LAva)
6.   Check CGPA
            6.1 If CGPA between 0 and 1.99 then LRec is equal to 4
            6.2 If CGPA between 2 and 2.99 then LRec is equal to 6
            6.3 If CGPA between 3 and 4 then LRec is equal to 7
7.   Count number of under probation semesters (NUS)
            7.1 If count >=2 then flag for under probation is on
            7.2 If count = 4 then go to step 10
8.   Compute student weights in each course division (c_div)$\frac{number\ of\ passed\ course\ division}{number\ of\ all\ course\ division} \times 100$
9.   Sort the top percentage in performance as index in course performance (C_Perf)
10.  Compute similarity between LOff and LRem, the output is List A
            10.1 Compare similarity between List A and List of prerequisites (LPre), the output is List B.
                        10.1.1  Retrieve the highest index from C_Perf
                        10.1.2  Output is List C, then send List C to Recommendation List (LRec)
11.  Append LRec based on the chosen scenario.
12.  Scenario A will append 4 courses, Scenario B will append 6 courses and Scenario C will append 7
     courses.
     End
```

Figure 2: AADSS Algorithm

3.3.3 Processing layer

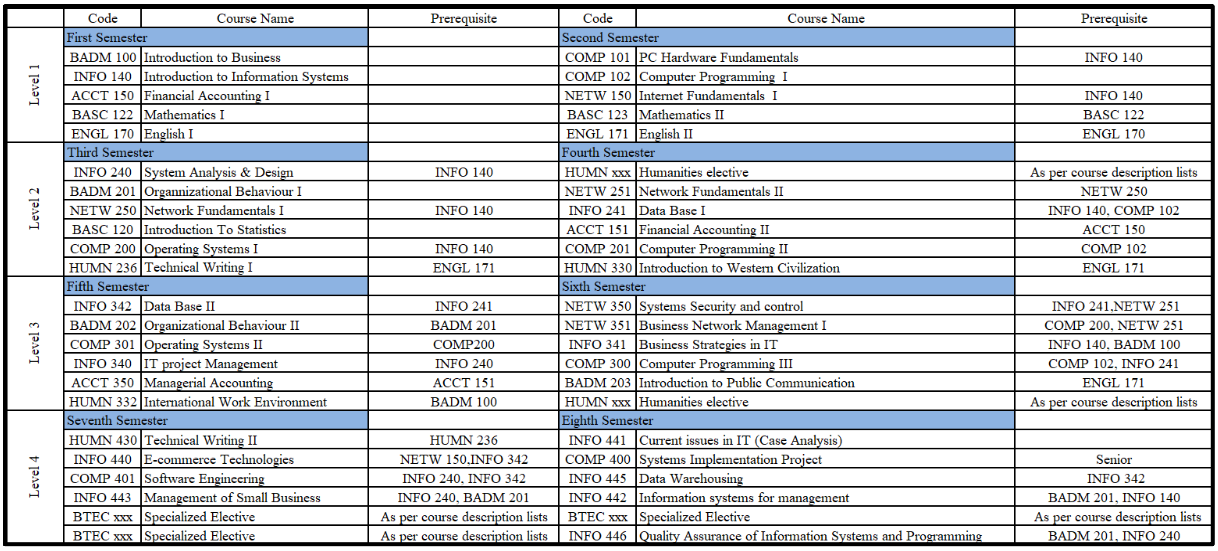
In this layer, the students' transcripts are analyzed and all passed and failed courses are retrieved. Performance Analysis calculation for any student is done through equation 1. Each major sheet is divided into number of divisions, each division contains number of courses. Then, the following equation counts the division courses and student passed courses in this division then computes the percentage as follows

$$\frac{number\ of\ passed\ course\ division}{number\ of\ all\ course\ division} \times 100 \qquad (1)$$

After calculating the percentage, it sorts the divisions percentage in a descending order. The higher percentage division will be shown first in the recommendation lists.

The engine interface output is three scenarios for each student case as recommendation lists for the user. The analyzed data that this framework used and created could be summarized as follows:

1. List of passed courses by the student and its grades.
2. List of the failed courses in the previous semesters.
3. List of all remaining courses for the student.
4. The prerequisites of the courses. Below is a Figure shows the major sheet courses and their prerequisites for BIS department.
5. List of available courses student can register after bypassing prerequisites requirements.
6. Check student's CGPA to decide how many credit hours the student should register.
7. Check student's academic status (progressing, stumbling, potential graduate or under probation student)
8. List of the offered courses in the current semester.
9. Calculate performance percentage for each division.
10. Prioritize courses in List C based upon percentage acquired from course division.
11. Recommendation lists are generated with three scenarios.



	Code	Course Name	Prerequisite	Code	Course Name	Prerequisite
Level 1	First Semester			Second Semester		
	BADM 100	Introduction to Business		COMP 101	PC Hardware Fundamentals	INFO 140
	INFO 140	Introduction to Information Systems		COMP 102	Computer Programming I	
	ACCT 150	Financial Accounting I		NETW 150	Internet Fundamentals I	INFO 140
	BASC 122	Mathematics I		BASC 123	Mathematics II	BASC 122
	ENGL 170	English I		ENGL 171	English II	ENGL 170
Level 2	Third Semester			Fourth Semester		
	INFO 240	System Analysis & Design	INFO 140	HUMN xxx	Humanities elective	As per course description lists
	BADM 201	Organnizational Behaviour I		NETW 251	Network Fundamentals II	NETW 250
	NETW 250	Network Fundamentals I	INFO 140	INFO 241	Data Base I	INFO 140, COMP 102
	BASC 120	Introduction To Statistics		ACCT 151	Financial Accounting II	ACCT 150
	COMP 200	Operating Systems I	INFO 140	COMP 201	Computer Programming II	COMP 102
	HUMN 236	Technical Writing I	ENGL 171	HUMN 330	Introduction to Western Civilization	ENGL 171
Level 3	Fifth Semester			Sixth Semester		
	INFO 342	Data Base II	INFO 241	NETW 350	Systems Security and control	INFO 241,NETW 251
	BADM 202	Organizational Behaviour II	BADM 201	NETW 351	Business Network Management I	COMP 200, NETW 251
	COMP 301	Operating Systems II	COMP200	INFO 341	Business Strategies in IT	INFO 140, BADM 100
	INFO 340	IT project Management	INFO 240	COMP 300	Computer Programming III	COMP 102, INFO 241
	ACCT 350	Managerial Accounting	ACCT 151	BADM 203	Introduction to Public Communication	ENGL 171
	HUMN 332	International Work Environment	BADM 100	HUMN xxx	Humanities elective	As per course description lists
Level 4	Seventh Semester			Eighth Semester		
	HUMN 430	Technical Writing II	HUMN 236	INFO 441	Current issues in IT (Case Analysis)	
	INFO 440	E-commerce Technologies	NETW 150,INFO 342	COMP 400	Systems Implementation Project	Senior
	COMP 401	Software Engineering	INFO 240, INFO 342	INFO 445	Data Warehousing	INFO 342
	INFO 443	Management of Small Business	INFO 240, BADM 201	INFO 442	Information systems for management	BADM 201, INFO 140
	BTEC xxx	Specialized Elective	As per course description lists	BTEC xxx	Specialized Elective	As per course description lists
	BTEC xxx	Specialized Elective	As per course description lists	INFO 446	Quality Assurance of Information Systems and Programming	BADM 201, INFO 240

Figure 3: BIS major sheet

3.3.4 Decision layer

In this layer, the framework displayed three scenarios, based on the student's academic status. The academic status used in this framework are progressing, stumbling, under probation and potential graduate. The academic progressing status occurs when a student is passing all the registered courses and did not fail any course so far. The academic stumbling status occurs when a student has some passed and failed courses in his transcript and cannot handle the registered courses. These student's cases should be handled carefully to avoid a drop in the student's CGPA that may lead to a change in the student's academic status. The potential graduate status occurs when a student is nearly to finish all required courses for graduation. These students' cases should be handled carefully to support and help them to graduate. The under-probation status occurs when a student's CGPA is below 2.00 for four consecutive semesters. these students' cases should be handled carefully to avoid the dismissal of the student.

The generated scenarios are Caution, Less caution and Optimal scenarios. Caution scenario is for under probation student case and for any student wants to improve his/her CGPA to change his academic status. In other meaning, this scenario is only displayed for students whose CGPA is below 2.00. Less caution scenario is for stumbling students' cases, progressing students and maybe for potential graduates based on the student's remaining courses. In other words, this scenario is displayed for students whose CGPA is between 2.00 and 3.00 and is most frequently used scenario by the advisors and students because it is the regular load any student could register based on bylaw rules. The optimal scenario is for progressing and potential graduates' students' cases only whose CGPA is above 3.00. Each scenario contains a recommendation list that suits the student's advising case.

4. AADSS EVALUATION:

We used the same dataset to evaluate the AADSS and check the accuracy of the framework results. For illustrating how this evaluation stage is done, two test cases will be presented in depth in this section. These cases are selected from the dataset as one of the two cases is under academic probation status. The second case study is a stumbling status and how AADSS helped the student to be graduated in the planned time and how to generate a graduation plan for this case.

Test case scenario 1:

This case scenario is under academic probation case, which means that the student has CGPA below 2.00 for four consecutive semesters; which means the advisor will have to handle this case with caution. This student will be allowed only to take 4 courses to fulfil his study plan and graduation requirements in an effort to change his academic status. The following steps for this scenario based on the algorithm defined in Fig.2.

Step 1: Getting student ID.

Step 2: Major, Program and CGPA is retrieved from DW as shown in Table 1.

Table 1: Student's Basic Information

Algorithm Step	Output	Description for output/process
2	Major	Business Technology (BTEC)
	Program	Information Systems (INEM)
	CGPA	1.76

Step 3: Retrieve from student's transcript Lists of passed courses, failed courses, and remaining courses as shown in Table 2.

Table 2: Course codes in LPass, LFail and LRem

Algorithm Step	Output		Description for output/process							
		Retrieve passed, failed and remaining courses from student's transcript								
3	LPass	Course Code	ENGL170	INFO140	COMP102	NETW150	ACCT150	BASC120	HUMN230	INFO240
			BADM100	BADM201	COMP101	BASC122	COMP200	INFO241	ENGL171	BASC123
	LFail	Course Code	ACCT151	COMP301	NETW250					
	LRem	Course Code	ACCT151	ACCT350	BADM202	BADM203	BTEC410	BTEC411	BTEC418	COMP201
			COMP300	COMP301	COMP400	COMP401	HUMN130	HUMN236	HUMN330	HUMN332
			HUMN430	INFO340	INFO341	INFO342	INFO440	INFO441	INFO442	INFO443
			INFO445	INFO446	NETW250	NETW251	NETW350	NETW351		

Step 4: Check student's status and then count how many courses and semesters remaining in order to graduate.

Step 5: Check the available courses in current semester as shown in Table 4

.

Table 3: Student Academic Status

Algorithm Step	Output	Description for output/process
4	SAS	Under Academic Probation
	NRC	30 courses
	NRS	6 semesters

Table 4: List of all available courses in current semester (LAva)

Algorithm Step	Output		Description for output/process							
		Retrieve all available courses								
5	LAva	Course Code	ACCT150	ACCT350	BADM100	BADM201	BADM202	BADM203	BASC120	BASC122
			BASC123	BTEC415	BTEC418	COMP200	COMP201	COMP301	COMP400	COMP401
			ENGL170	ENGL171	HUMN130	HUMN131	HUMN230	HUMN235	HUMN236	HUMN237
			HUMN238	HUMN330	HUMN332	HUMN430	INFO140	INFO240	INFO241	INFO340
			INFO341	INFO342	INFO440	INFO441	INFO442	INFO443	INFO445	INFO446
			NETW150	NETW250	NETW251	NETW350	NETW351	NETW450	NETW451	

Step 6: check student's CGPA in order to know how many courses student will be able to register in the current semester.
Student's CGPA is below 2.00. Therefore, student will be able to register only 4 courses.
Step 7: Compute number of semesters that student's CGPA is below 2.00

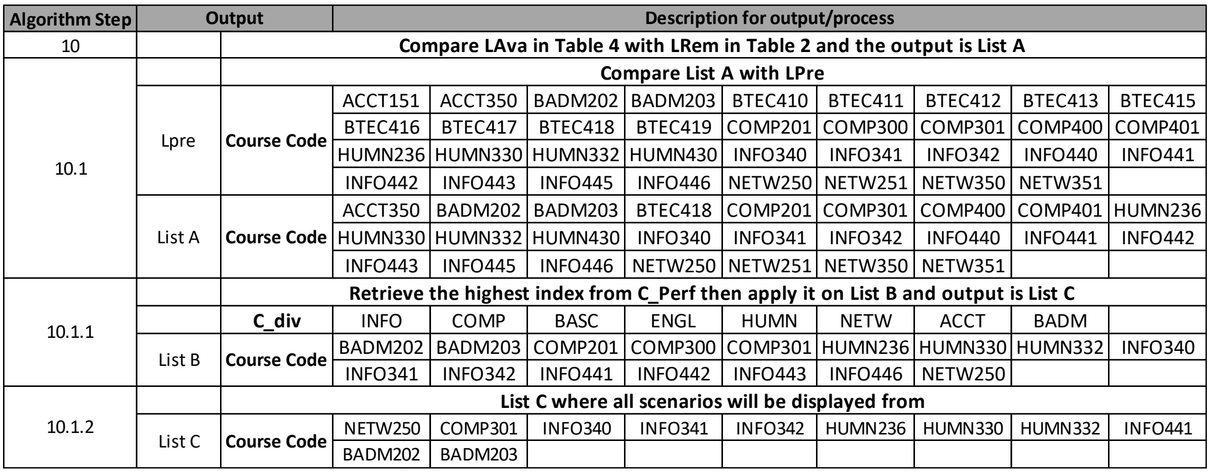
When computing number of under probation semesters, it was found that this student has four consecutive semesters where CGPA is below 2.00. Therefore, flag for under probation status is ON for this student.
Step 8: Compute student weights in each course division.

Table 5: Computed and sorted Course Divisions

Algorithm Step	Output		Description for output/process							
8	Unsorted	Compute course division weights								
		C_div	ENGL	INFO	COMP	NETW	ACCT	BASC	HUMN	BADM
		weight	20%	30%	30%	10%	10%	30%	20%	10%
9	Sorted	Sort Computed weights								
		C_div	INFO	COMP	BASC	ENGL	HUMN	NETW	ACCT	BADM
		weight	30%	30%	30%	20%	20%	10%	10%	10%

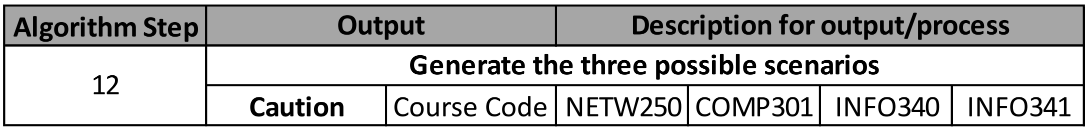
Step 9: Sort these weights in descending order as shown in Table 5.
Step 10: Compare list of available courses with list of remaining, the output is List A, then compare it with List of prerequisites, the output is List B, then compare it with courses performance, the output is List C.
Step 11: Create LRec
Step 12: Compute three scenarios for the student to choose from.

Table 6: Steps to get all possible courses for decisions



Algorithm Step	Output		Description for output/process								
10		Compare LAva in Table 4 with LRem in Table 2 and the output is List A									
10.1			Compare List A with LPre								
	Lpre	Course Code	ACCT151	ACCT350	BADM202	BADM203	BTEC410	BTEC411	BTEC412	BTEC413	BTEC415
			BTEC416	BTEC417	BTEC418	BTEC419	COMP201	COMP300	COMP301	COMP400	COMP401
			HUMN236	HUMN330	HUMN332	HUMN430	INFO340	INFO341	INFO342	INFO440	INFO441
			INFO442	INFO443	INFO445	INFO446	NETW250	NETW251	NETW350	NETW351	
	List A	Course Code	ACCT350	BADM202	BADM203	BTEC418	COMP201	COMP301	COMP400	COMP401	HUMN236
			HUMN330	HUMN332	HUMN430	INFO340	INFO341	INFO342	INFO440	INFO441	INFO442
			INFO443	INFO445	INFO446	NETW250	NETW251	NETW350	NETW351		
10.1.1		Retrieve the highest index from C_Perf then apply it on List B and output is List C									
		C_div	INFO	COMP	BASC	ENGL	HUMN	NETW	ACCT	BADM	
	List B	Course Code	BADM202	BADM203	COMP201	COMP300	COMP301	HUMN236	HUMN330	HUMN332	INFO340
			INFO341	INFO342	INFO441	INFO442	INFO443	INFO446	NETW250		
10.1.2		List C where all scenarios will be displayed from									
	List C	Course Code	NETW250	COMP301	INFO340	INFO341	INFO342	HUMN236	HUMN330	HUMN332	INFO441
			BADM202	BADM203							

Table 7: All scenarios given for test case 1



Algorithm Step	Output		Description for output/process			
12	Generate the three possible scenarios					
	Caution	Course Code	NETW250	COMP301	INFO340	INFO341

Scenario 1: Caution decision:

As shown in Table 7 this scenario is the chosen one because the student's CGPA is below 2.00 and based on the student's academic status he/she could not be offered more than four courses to register. For the chosen courses, the student has three failed courses and only two courses are available, therefore, these failed courses are the first two courses then the other two are chosen based on his/her prioritized course division calculated in steps eight and nine.

Test case scenario 2:

This case is a stumbling case. The student's CGPA is between 2.00 and 2.99, therefore the maximum number of registered courses is six. The following steps for this scenario based on the algorithm defined in Fig.2.

Step 1: Getting student ID.

Step2: Major, Program and CGPA is retrieved from DW as shown in Table 8.

Table 8: Student's basic information

Algorithm Step	Output	Description for output/process
2	Major	Business Technology (BTEC)
	Program	Information Systems (INEM)
	CGPA	2.07

Step 3: Retrieve from student's transcript List of Passed courses, failed courses, remaining courses as shown in Table 9.

Step 4: Check student's status and then count how many courses remaining to be graduated.

Step 5: Check the courses that are offered in the semester by the department shown in Table 11.

Step 6: check student's CGPA in order to know how many courses he/she will be able to register in this semester.

Student's CGPA is between 2.00 and 2.99. Therefore, student will be able to register six courses.

Step 7: Compute number of semesters that student's CGPA is below 2.00

When computing number of under probation semesters, it was found that this student has no consecutive semesters where CGPA is below 2.00. Therefore, flag for under probation status is OFF for this student.

Step 8: Compute student weights in each course division.

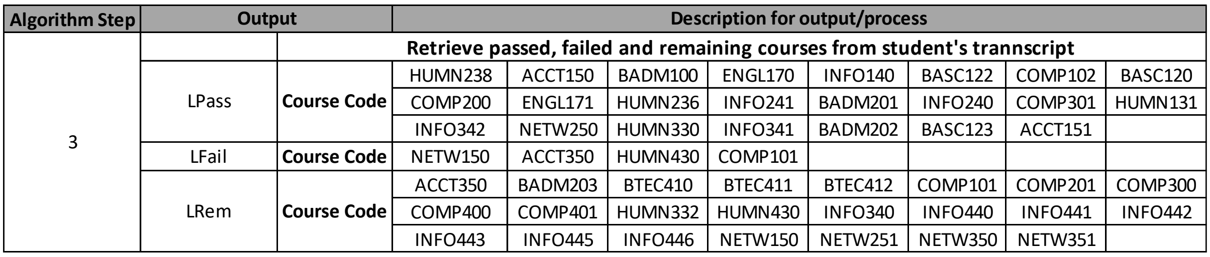
Step 9: Sort these weights in descending order.

Step 10: compare list of available courses with list of remaining, the output is List A, then compare it with List of prerequisites, the output is List B, then compare it with courses performance, the output is List C.

Step 11: Create LRec

Step 12: Compute the three scenarios for the student to choose from.

Table 9: Course codes in LPass, LFail and LRem



Algorithm Step	Output		Description for output/process							
3		Retrieve passed, failed and remaining courses from student's trannscript								
	LPass	Course Code	HUMN238	ACCT150	BADM100	ENGL170	INFO140	BASC122	COMP102	BASC120
			COMP200	ENGL171	HUMN236	INFO241	BADM201	INFO240	COMP301	HUMN131
			INFO342	NETW250	HUMN330	INFO341	BADM202	BASC123	ACCT151	
	LFail	Course Code	NETW150	ACCT350	HUMN430	COMP101				
	LRem	Course Code	ACCT350	BADM203	BTEC410	BTEC411	BTEC412	COMP101	COMP201	COMP300
			COMP400	COMP401	HUMN332	HUMN430	INFO340	INFO440	INFO441	INFO442
			INFO443	INFO445	INFO446	NETW150	NETW251	NETW350	NETW351	

Table 10: Student Academic Status

Algorithm Step	Output	Description for output/process
4	SAS	Stumbling
	NRC	23 courses
	NRS	4 semesters

Table 11: List of all available courses in current semester (LAva)

Algorithm Step	Output		Description for output/process							
		Retrieve all available courses								
5	LAva	Course Code	ACCT150	ACCT350	BADM100	BADM201	BADM202	BADM203	BASC120	BASC122
			BASC123	BTEC415	BTEC418	COMP200	COMP201	COMP301	COMP400	COMP401
			ENGL170	ENGL171	HUMN130	HUMN131	HUMN230	HUMN235	HUMN236	HUMN237
			HUMN238	HUMN330	HUMN332	HUMN430	INFO140	INFO240	INFO241	INFO340
			INFO341	INFO342	INFO440	INFO441	INFO442	INFO443	INFO445	INFO446
			NETW150	NETW250	NETW251	NETW350	NETW351	NETW450	NETW451	

Table 12: Computed Course Division Percentage

Algorithm Step	Output	Description for output/process								
8	Unsorted	**Compute course division weights**								
		C_div	ENGL	INFO	COMP	NETW	ACCT	BASC	HUMN	BADM
		weight	20%	50%	30%	10%	10%	30%	40%	30%
9	Sorted	**Sort Computed weights**								
		C_div	INFO	HUMN	COMP	BASC	BADM	ENGL	NETW	ACCT
		weight	50%	40%	30%	30%	30%	20%	10%	10%

Table 13: Steps to get all possible courses for decisions

Algorithm Step	Output		Description for output/process								
10		**Compare LAva in Table 11 with LRem in Table 9 and the output is List A**									
10.1		**Compare List A with LPre**									
	Lpre	**Course Code**	ACCT151	ACCT350	BADM202	BADM203	BTEC410	BTEC411	BTEC412	BTEC413	BTEC415
			BTEC416	BTEC417	BTEC418	BTEC419	COMP201	COMP300	COMP301	COMP400	COMP401
			HUMN236	HUMN330	HUMN332	HUMN430	INFO340	INFO341	INFO342	INFO440	INFO441
			INFO442	INFO443	INFO445	INFO446	NETW250	NETW251	NETW350	NETW351	
	List A	**Course Code**	ACCT350	BADM203	COMP201	COMP400	COMP401	HUMN332	HUMN430	INFO340	INFO440
			INFO441	INFO442	INFO443	INFO445	INFO446	NETW150	NETW251	NETW350	NETW351
10.1.1		**Retrieve the highest index from C_Perf then apply it on List B and output is List C**									
		C_div	INFO	HUMN	COMP	BASC	BADM	ENGL	NETW	ACCT	
	List B	**Course Code**	ACCT350	BADM203	COMP201	COMP401	HUMN332	HUMN430	INFO340	INFO440	INFO441
			INFO442	INFO443	INFO445	INFO446	NETW150	NETW251			
10.1.2		**List C where all scenarios will be displayed from**									
	List C	**Course Code**	NETW150	ACCT350	HUMN430	INFO340	INFO440	INFO443	INFO441	INFO442	HUMN332
			COMP201	COMP401	BADM203	NETW251					

Table 14: All scenarios given for test case 2

Algorithm Step	Output		Description for output/process						
12	**Generate the three possible scenarios**								
	Caution	Course Code	NETW150	ACCT350	HUMN430	INFO340			
	Less Caution	Course Code	NETW150	ACCT350	HUMN430	INFO340	INFO440	INFO443	

Scenario 1: Caution decision.

As shown in Table 14, this scenario would be chosen if the student wants to reduce his/her load to improve student's CGPA.

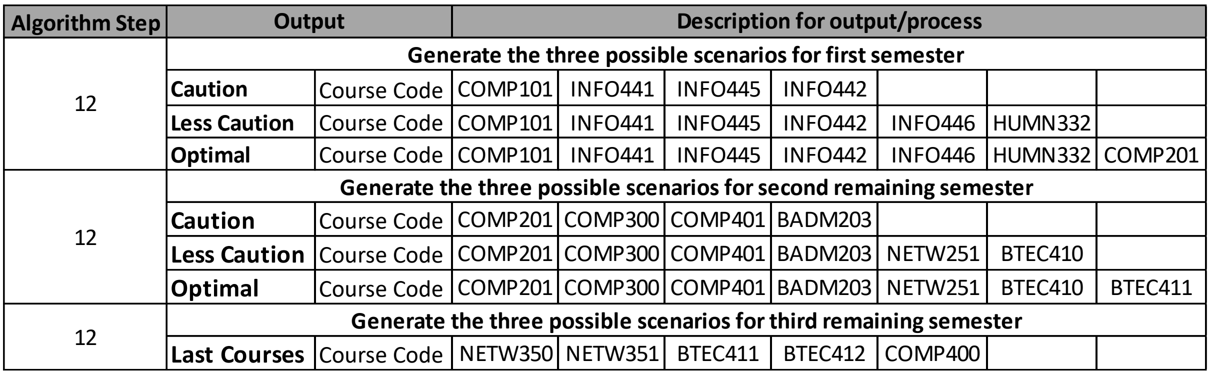
Scenario 2: Less Caution decision.

As shown in Table 14 this scenario is chosen the chosen one as student's CGPA is between 2.00 and 2.99. How the courses are chosen, the student has four failed courses and only three courses are available in this semester, therefore, these failed courses are the first three courses then the other three are chosen based on his/her prioritized course division calculated in steps eight and nine.

Finally, in this case scenario, the advisor or the student when registering courses in the current semester should pick up "Less caution" scenario for the student to stick with his graduation plan.

Assuming that the student pass in Less caution courses, three semesters will be remaining for him/her to be graduated. These semesters will be planned as shown in Table 15 and each semester the student will choose Less caution scenario because his CGPA is still between 2.00 and 2.99.

Table 15: A complete graduation plan for the student



Algorithm Step	Output		Description for output/process						
12	Generate the three possible scenarios for first semester								
	Caution	Course Code	COMP101	INFO441	INFO445	INFO442			
	Less Caution	Course Code	COMP101	INFO441	INFO445	INFO442	INFO446	HUMN332	
	Optimal	Course Code	COMP101	INFO441	INFO445	INFO442	INFO446	HUMN332	COMP201
12	Generate the three possible scenarios for second remaining semester								
	Caution	Course Code	COMP201	COMP300	COMP401	BADM203			
	Less Caution	Course Code	COMP201	COMP300	COMP401	BADM203	NETW251	BTEC410	
	Optimal	Course Code	COMP201	COMP300	COMP401	BADM203	NETW251	BTEC410	BTEC411
12	Generate the three possible scenarios for third remaining semester								
	Last Courses	Course Code	NETW350	NETW351	BTEC411	BTEC412	COMP400		

5. CONCLUSION AND FUTURE WORKS

This paper provides an AADSS system for enhancing the academic advising process in higher educational institutions using credit-hours system. AADSS is proposed to reduce the errors of manual advising, while saving the advisor's time and optimizing their efforts. The AADSS provides different scenarios for every student advising case so that the student or advisor can pick the applicable scenario based on each academic student status. This paper adopted DSR methodology, where a suggested tentative design for the framework was demonstrated in section 3.2. The AADSS framework is developed after applying the school bylaw and the academic registration rules discussed in section 4. Finally, this framework is evaluated in section 5 using two case scenarios obtained from CIC.

After testing AADSS by a number of students and advisors, the following findings could be summarized that the AADSS can reduce cost, time, and effort. AADSS also helps advisors make better decision for their students. Moreover, AADSS would enhance the advising process by limiting human errors caused by their interference.

This framework can be enhanced by using machine learning techniques to predict the grades for each registered course based on students' grades. We can also add more scenarios is the decision layer and the best scenario for each student will be chosen in an automotive way without any human interference either from the student or the advisor.

REFERENCES

1. AbdulWahhab, R., & Abdul Wahhab, S. (2016). Improving Academic Advising System by Using iAMA. Journal of Teaching and Education, 5(01), 191-208.

2. Afify, E., & Nasr, M., 2017, "A Proposed Model for a Web-Based Academic Advising System". International Journal of Advanced

Networking and Applications, 9(2), 3345-3361.

3. Aina, A.A.M. (2015). Developing Decision Support Capabilities through Use of Management Information Systems. International Journal of Management Science and Business Administration, 1(9), 46-51.

4. Al-Ansari, A., El Tantawi, M., AbdelSalam, M., & Al-Harbi, F. (2015). Academic advising and student support: Help-seeking behaviors among Saudi dental undergraduate students. The Saudi dental journal, 27(2), 57-62.

5. Almutawah, K. A. (2014). A decision support system for academic advisors. International Journal of Business Information Systems, 16(2), 177-195.

6. Beal, P.E. and Noel, L. (1980), What Works in Student Retention, American College Testing Program, Iowa City, IA.

7. Cuseo, J. (2003), Academic Advisement and Student Retention: Empirical Connections and Systemic Interventions, National Academic Advising Association (NACADA),

8. Hevner AR; March ST; Park J & Ram S., 2004. Design science in information systems research. MIS quarterly, 28 (1), pp. 75–105.

9. Laghari, M. S. (2014). Automated course advising system. International Journal of Machine Learning and Computing, 4(1), 47.

10. Mohamed, A., 2016, "Interactive decision support for academic advising". Quality Assurance in Education, 24(3), 349-368.

11. Noaman, A. Y., & Ahmed, F. F. 2015, "A new framework for e academic advising". Procedia Computer Science, 65, 358-367.

12. R. M. Choudhari, "Expert System for Student Advisement", [Online] Department of Mathematics and Computer Science, South Carolina State University, Orangeburg, SC 29117,2009.

13. Turban, E., Sharda, R., & Delen, D. (2007). Decision support and business intelligence systems. Boston: Prentice Hall.

14. Vaishnavi, V., Kuechler, W., and Petter, S. (Eds.) (2004/19). "Design Science Research in Information Systems" January 20, 2004 (created in 2004 and updated until 2015 by Vaishnavi, V. and Kuechler, W.); last updated (by Vaishnavi, V. and Petter, S.), June 30, 2019.

15. Yeh, C. W. (2015). CRIS: A rule-based approach for customized academic advising. In WMSCI 2015-19th World Multi-Conference on Systemic, Cybernetics and Informatics, Proceedings (Vol. 2, No. 2, pp. 213-218).

16. Yousef, K. A., & Abdleati, R. N. An Intelligent Electronic Academic Advisor. International Journal of Computer Applications, 975, 8887.

17. Young Jones, A. D., Burt, T. D., Dixon, S., & Hawthorne, M. J. (2013). Academic advising: does it really impact student success? Quality Assurance in Education, Vol. 21, pp. 7-19.