education sciences

MDPI

Article

# Considering Students' Abilities in the Academic Advising Process

**Samia Loucif [1], Laila Gassoumi [2] and Joao Negreiros [1,*]**

[1] College of Technological Innovation, Zayed University, Abu Dhabi P.O. Box 144534, UAE; samia.loucif@zu.ac.ae
[2] TechnipFMC, Abu Dhabi P.O. Box 7657, UAE; laila.gassoumi@technipfmc.com
* Correspondence: joao.negreiros@zu.ac.ae

Received: 15 August 2020; Accepted: 10 September 2020; Published: 17 September 2020

check for updates

**Abstract:** Academic advising is time-consuming work. At the same time, it needs to be efficient and productive in assisting the students to choose appropriate academic courses towards the completion of their selected programs in a beneficial manner. In addition, both private and public educational institutions are, currently, operating in an extremely competitive market and are, thus, faced with various challenges. Among these are the twin challenges of student retention and the rate of success in completion of their chosen academic courses. The mentioned challenges have a direct bearing on the quality of academic advising and services provided to the students, by the individual academic institution. A number of research studies have been carried out suggesting various online academic advising systems for undergraduate and graduate programs. In this context, we develop and present, here, an academic advising system which differs from and improves upon previously suggested methodologies with the inclusion of the facility to track individual students' performance and, thus, ability in educational subjects and programs, taken in the previous academic terms. Our suggested methodology is based on the use of this facility to guide students in the selection of courses that they may register for the forthcoming academic term. We believe that the consideration of individual students' past academic preformation, in our suggested methodology, is a significant improvement and will assist students in making more beneficial choices when registering for academic courses.

**Keywords:** education; program curriculum; academic advising; course registration; Bloom's taxonomy; student's ability

## 1. Introduction

With the advances in information technology, the current generation of students is using digital technologies extensively in almost all of their daily activities [1]. In addition, information technology has overtaken education in a way that the traditional teaching approach is no longer a requirement for a successful learning experience. Yet, depending upon the student's comprehension and the nature of the subject, the role of an instructor can vary from guiding a student through every step of the education process to being a mere moderator [2]. Notwithstanding, guiding students through the education process continues to play a crucial role in their academic and professional success.

Academic advising is an interactive process involving the student and the academic advisor, with the sole purpose of enabling the student to progress through the educational program in a timely manner. The academic advisor helps students to select the most appropriate courses to enroll in and in the development of study plans, throughout the academic year [3]. The academic interests and the ability to cope with different subject areas vary from student to student; it is, therefore, important that due consideration is given to these variables, by the advisor, in the advising process. A fruitful advising process also contributes towards the quality of an educational institution through the increase in students' satisfaction and retention [4,5].

It thus follows that the quality of academic advising services, provided by an educational institution, is considered to be among the factors that contribute not only to the success of the students but to the success of an institution, itself. Student advising assumes more importance especially for freshman and sophomore students who find college environments more challenging with little or no backup support as compared to their high school experience in familial environments. It is, therefore, important that the academic advising task is assigned to professional advisors who are dedicated to the task and are accessible to students whenever help is required. For effective and successful academic advising, the academic advisors must also be knowledgeable about the institutional resources and policies, details of the programs and curricula such as course requirements, and students' performance, etc. However, the current situation in the majority of educational institutions is that the academic advising task is assigned to faculty members who, in addition to their teaching duties, committee work, and research, are assigned a group of students to advise and monitor their progress towards graduation. In some cases, the faculty members assigned with the advising task are also handicapped by the lack of knowledge and experience in the art and science of academic advising.

In every academic term, students crowd in their academic advisors' offices to seek guidance on their study plans. This could be a frustrating process for both students and advisors, particularly at the time of the new intake of students. In addition, the routine academic advising process is not an easy task. It is not only time consuming but needs to be carried out in a timely manner with due attention to details. Depending on how advisors and students are involved in making decisions with regards to the selection of courses, academic advising can be classified as prescriptive, developmental, integrated, or intrusive [6–8]. At one extreme, the prescriptive advising model refers to students who totally rely on their academic advisors to decide on the list of the courses to register whereas, on the other extreme, the academic advisor expects the students to take full responsibility for selecting courses for registration. The "integrated" and the "intrusive" approaches are located between the two mentioned extremes. The integrated model is more flexible and involves both advisor and student in making decisions upon the selected course, whereas the intrusive becomes adequate for special cases of students who need continuous monitoring such as freshmen or those who are on probation [6,9].

To ensure quality of their program curricula, all education institutions must define the associated courses' learning outcomes which must, in turn, be aligned with the program goals. It is generally agreed that the courses' learning outcomes contribute, directly, to the success and the achievement of the program itself. Bloom's taxonomy has always been the most frequently used framework in setting courses' learning outcomes [10], particularly under the cognitive domain which describes knowledge, skills, and abilities gained by students by the graduation day [4,11]. Utilizing this assumption, our proposed advising system is based on the developmental advising style. It takes into account the course requirements in making decisions about the selection of the courses a student can register for, in addition to the consideration of the knowledge, abilities, and skills used and developed by the student in courses successfully completed in previous terms. The details of the developmental approach are provided later, in this paper.

This research paper is organized as follows. Section 2 discusses the literature review while Section 3 provides details of the proposed advising system, including the criteria used for course prioritization and architecture. The suggested internal algorithm is described in Section 4, followed by an illustrative example in Section 5. Section 6 concludes this paper. Finally, the shortcomings of the proposed system are described in Section 7.

## 2. Related Work

A number of research studies have been conducted to find the most accurate and comprehensive automated academic advising system. Nagy et al. [12] developed an advising system that is based on machine learning techniques where the main selection criteria are based on students' academic performance along with the grades obtained in their first year at the university. However, the data entry to this prototype model is carried out by the students themselves, thus, adding to the likelihood of intentional or non-intentional input mistakes and the resultant quality of the recommendations.

El-Sheikh et al. [13] presented an analytical cross-sectional research design (51 academic advisors and 424 students enrolled) in the faculty of nursing at Mansoura University to develop corrective actions and to improve the academic advising process through investigating the obstacles and solutions from the perspective of both academic advisors and students. According to Jaime et al. [14], obstacles related to advisors scored higher than the obstacles from the students' point of view. Students' levels varied significantly with advisors' performance and obstacles but there was no significant correlation between advisors' performance and academic advising obstacles as observed by students. Still, their findings pointed to enhance academic staff's abilities related to academic advising through training programs.

Mueller and Meyer [15] addressed the organization interested in launching online advising and reported two variables must be addressed before moving forward. (1) What platform would work best based on technology options and student demographics? It seems that Adobe Connect is a possibility because it provides a user-friendly web and voice-enabled experience. It also does not require the installation of any software and allows for screen sharing, chat, and group advising sessions. (2) The physical space allocated for the operation of the facility in terms of quietness and confidentiality must be adequate.

Gordon [16] has studied student satisfaction within centralized advising offices with general, departmental, and individual advisors. He has used the advising scale to measure student satisfaction. The study is of the causal-comparative type and uses a one-way ANOVA where the sample was drawn from an online undergraduate and graduate program student population. The researcher found that there was a statistically significant difference in student satisfaction. The students in the individual advisor group had the highest level of student satisfaction, the students with departmental advisors scored second highest, and the students falling within the general group had the lowest student satisfaction scores.

Choudhari [17] proposed an expert system for academic advising which includes features such as the information provided to students, by the academic institutions, on the availability of programs and courses and the enrollment requirements. In addition, the system also provides information on other available facilities such as developing multi-semester study plans with updating facilities. The proposed system is linear in that it relies only on information related to course requisites without any consideration of actual student performance. In addition, to access the advisor option, for instance, a student has to go through several steps that can be skipped by simply entering the university ID.

Chakraborty et al. [18] reported a content-based mining approach which goes through all relevant institutional data storage facilities and extracts required information in order determine the most accurate recommendation list of scholars based on a comparative analysis (Latent Dirichlet Allocation, LDA, Hierarchical Dirichlet Process, HDP, Latent Semantic Analysis, LSA, and Clustering techniques: k-means and Hierarchical Clustering). In reference [19], Matulatan and Resha offered a Monte Carlo tree type style search to help a students' advisor in analyzing students' performance based on the actual academic progress records of the students. It is augmented with the inclusion of a facility to build course patterns based on the performance records of previous subject-specific students. The data mining approach uses a selective cross join for each possible permutation of pair courses with respect to course grades in order to develop a knowledge base that is used to construct a complex tree of any possible study path that might be taken by a student.

McMahan [20] suggested another proposal based on prescriptive modeling. Its focus is on the interaction between the advising system and the student based on natural dialog. The advisory session is recorded in order to identify the main phrases and information used during that process and then encoded using Artificial Intelligence Markup Language along with the dialog system (implemented in Python). The usage of natural language, for the purpose, is also suggested, by Latorre-Navarro and Harris [21], to facilitate students to get advice and interact with the system. A crucial component of this intelligent advising system is the knowledge base that encompasses the electrical and the computer engineering program requirements of Florida University. Under the context of intelligent advising platforms, Gavriushenko et al. [22] advocated the use of a clustering technique to classify students' profiles and to identify similar cases that may be compared by academic advisors to speed-up this advising process.

Machine learning techniques are helpful to develop intelligent prototype academic advising systems as pointed out in reference [12]. Hsu et al. [23] proposed a Web-based academic advising system for undergraduate programs (computer science and civil engineering) at Florida Atlantic University. They focused on the selection of the "next course to take" option which allows a student to input information related to all courses already taken in order to get a list of courses for selection in the following semester. The selection is based on students' preferences, course requirements and their availability. The system relies on students' inputs which can be improved by just inputting their university ID and, thus, minimizing errors.

Ozturan and Ayan [24] recommended a simple advising framework for management information system students at Boğaziçi University, where the system generates a list of suggested courses, considering the status of students. The framework is handicapped by the fact that it does not cover freshman. Similarly, Bansal et al. [25] presented a knowledge-based resource advising kit (KRAK) (an easy to use drag-and-drop Web-based framework) that allows users to design their own study plans. Its main tasks are degree-planning, semester scheduling, and collecting general university requirement information. The major outputs include complete degree plans with lists of offered courses with a description of each course and faculty member's information. KRAK does not have a decision-making algorithm because it depends on the student's opinion and wishes. Nevertheless, the system offers a facility to resolve conflicts between the chosen courses.

Mihali et al. [26] introduced a course scheduling and advising software framework called SKED. This software, based on the methodology, is capable of generating a list of courses that a student may register to satisfy the requirement of each program. To ensure a student's fast graduation, the algorithm prioritizes the selection of classes with higher requirements and zero-availability cost iteratively until the allowed total number of credits required per semester is reached. Another feature involves the control of special cases such as forcing students not to delay registration with low cost requirements.

Liu et al. [27] believe that the advisor–advisee relationship represents direct knowledge heritage and such a relationship may not be readily available from academic libraries and search engines. Their work aims to discover advisor–advisee relationships hidden behind scientific collaboration networks by proposing a novel model based on network representation learning (NRL). The system is named Shifu2. Shifu2 takes the information from the collaboration network as input and identifies the advisor–advisee relationship as output. In contrast to existing NRL models, Shifu2 considers not only the network structure but also the semantic information of nodes and edges with the capability to generate large-scale academic genealogy datasets.

HE-Advisor is an academic advising software introduced by Albalooshi and Shatnawi [28] that provides several options including the generation of a list of courses that a student may register for in the upcoming academic term. The displayed output includes all program courses: courses already accomplished, currently being taken, and those that a student can or cannot register for. On the other hand, Aly et al. [29] have developed a system that uses an intelligent algorithm in order to develop an expert advising application for smart phones. Their experimental results show that their system has an average root mean square error of 6.64%, thereby making it a reliable system for making high-quality correct subject-selection decisions.

Alkhoori et al. [30] have presented UniBud, a virtual academic adviser system that is specifically designed for ease of use. It uses DialogFlow, a natural language understanding platform, to build voice-based interactions with students. The interactions allow students to inquire about course information, enrollment, and other general enquiries. As indicated by the authors, UniBud is not meant to replace the traditional academic advising process. UniBud is offered to support a limited set of academic enquiries thereby freeing human academic advisers to assist students with more involved enquiries.

Gutiérrez et al. [31] have introduced a learning analytics dashboard for advisers (LADA) to support the decision-making process of academic advisers through comparative and predictive analysis. Their results indicate that LADA enables expert advisers to evaluate significantly more scenarios (median = 2), especially for more involved academic advising cases with students that failed several courses over a fairly short period. For inexperienced advisers, LADA is considered a valuable tool for more accurate and efficient decision-making as they were able to make informed decisions in a similar amount of time compared to the experts.

Other platforms propose the usage of course history such as the Academic Advisor COurse Recommendation eNgine (AACORN) suggested by Sandvig and Burke [32] and the Course Advisory Expert System (CAES) proposed by Daramola et al. [33]. AACORN is an advising system for graduate students at DePaul University and it is mainly rooted in a case-based reasoning technique. Using the student's query, the system retrieves the most similar students' cases from the case histories database for recommendation purposes. CAES, on the other hand, combines the techniques of rule-based reasoning with case-based reasoning. The former chooses the courses based on a set of rules configured and set by experts. The rules include the assigning of weights to each offered course and selecting the courses with the highest weights. The weights are then added to the current plan. The CAES, on the other hand, does not rely on rules, rather, makes recommendations based on similarity to previous cases.

## 3. The Proposed Advising System

Most academic institutions seek professional accreditation for their programs, which is considered a quality indicator of their programs. In addition, the professional accreditation bodies impose certain criteria to be fulfilled by programs for the continued retention of the accreditation. For example, the three main components comprising mathematics and basic sciences, engineering topics, and general education must be included for accreditation of an ABET engineering program curriculum [34]. The UAE Commission for Academic Accreditation (CAA) Standards [35] for Program Accreditation dictates that the program must put a main emphasis on majors, general education, and electives subjects. Another criterion is the minimum number of credit hours (CHs) for each component of the program.

We feel that the objectives of the subject components of all programs must be aligned with the objective of all sponsoring institutions. Simply stated, a set of goals need to be developed that broadly describe what the program intends to accomplish, the learning outcomes, and the details of the constituent courses. The course details must include the required syllabus, its objectives, learning outcomes, and the assessment tools. There is no doubt that the hardest part in designing a course syllabus is the development of learning outcomes and measurable statements of the knowledge and skills that students must have acquired by the end of the course.

Bloom's taxonomy is the framework that is the most frequently used in writing course learning outcomes [4]. Its six levels are related to the cognitive domain: remembering, understanding, applying, analyzing, evaluating, and creating. Somehow, the use of students' abilities in terms of knowledge and skills under the advising process for the upcoming semester should be taken into consideration. We feel it to be an important factor that needs to be considered to develop an improved academic advising system for the selection of courses for the following semester. Therefore, our present proposal recommends that for each course a weighted percentage should be assigned for each of Bloom's six categories with percentage weightage ranging from 0 to 100. The maximum total that can be allocated to a single subject is not exceeding 100. The percentage weightage assigned to each Bloom category would depend on the level of a course. For an introductory course, the learning outcome targets would be based on remembering and understanding while those the allocated weightage of advance courses would be based on application, evaluation, and creativity. For future reference, we denote the six levels of the cognitive process as abilities. It would not be out of place to mention that the changes suggested here can be generalized to suit learning outcomes for almost all types of accreditation organizations.

Under our proposal, the curriculum for any academic program is designed by selecting the abilities that a student should develop to fulfill the program requirements and by assigning a percentage weightage for each of the abilities that contributes to the program learning outcomes. Based on these numerical values, each course is developed by targeting some of these abilities, each of which with a certain percentage and whose total equals 100. For instance, the subject of English is considered to require the development of 100% communication skills whereas the successful completion of a capstone project would require the development of other abilities such as designing and creativity. By assigning the required abilities weightage to courses, as proposed, will help in (1) devising assessment methods based on the target abilities, (2) assessing the performance of students for each ability in that course, and (3) using students' performance to assess the achievement of the course learning outcomes (CLOs) and suggesting improvements regarding teaching and/or assessment methods. The proposed approach would also help in (4) analysis of graduated students' abilities to identify potential weaknesses, point out revisions of the program curriculum, and suggest improvements to remedy these issues. In addition, the proposed methodology will enable (5) the collection of students' abilities data from the courses taken. This can, then, be used to recommend students' academic areas of strength where they are more likely to succeed. The information can also be used to suggest potential transfer to other majors.

The addition of an ability component to a curriculum does not require major changes in the registration system. A simple webpage can be added to the online admission system, allowing input of the required students' abilities for each program curriculum with their respective percentage weightage. This can be updated anytime, for instance, in the case of a program revision. This would, however, require re-mapping of abilities to courses. The decision on which ability and its respective percentage weightage is assumed to be taken at the design or revision stage of the program curriculum. Therefore, more attributes related to program abilities are added to the course entity. Additionally, students' records encompass these abilities reflecting their performance from all courses previously taken.

Academic programs consist of a course sequence and where it is assumed that each course belongs to one of the three following categories: university general requirements, faculty requirements, and program requirements. Most of higher educational institutions offer credit-based programs where students must complete a maximum number of CHs in each category. If there are no courses in a specific category, then the maximum number of CHs is simply set to zero. The courses in each category can also be classified as compulsory or elective. A CH is defined as fifty (50) minutes of class contact time and two (2) hours of outside of class work per week [35]. In addition, the attributes of any course include the code, title, number of CHs, term (semester and year in the study plan), pre- and co-requisites, among others. Lastly, a course must, at least, belong to a major, a category (such as basic sciences, general education, or major), and to a group to indicate whether the course is compulsory or elective.

### *3.1. Criteria Used for Course Prioritization*

The courses are, in turn, sorted based on their prerequisite chain length (PCL). The concept of PCL is explained in Figure 1 where courses A and F show a PCL of three and one, respectively, indicating course A to be of higher priority than F. The students are, therefore, advised to take courses with the longest prerequisite chain in order to avoid graduation delays.

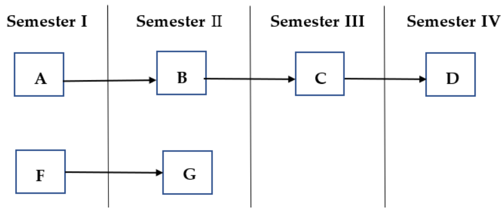


**Figure 1.** Example of prerequisite chain length (PCL) for two given courses: PCL(A) = 3 and PCL(F) = 1.

Prerequisite cost (PC) of a course *i* represents the number of courses which require that course *i* as a direct prerequisite [26]. Consequently, the greater the PC, the higher is its priority. This is shown in Figure 2, where course A is given higher priority than F because it leads into two courses while F leads into only one.

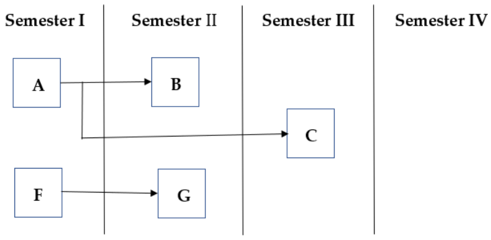


**Figure 2.** Prerequisite cost (PC) for two given courses: PC(A) = 2 and PC(F) = 1.

The ability weight is considered to be the core of the proposed advising system as it is used to analyze students' academic performance and their aptitude level. The ability weight, in itself, is estimated, by the proposed advising system, by assigning a set of abilities (communication, problem solving, creativity, or analysis, for instance) along with their respective percentage weights to each course considered essential for the successful completion of the subject course. The ability weight is, then, used to establish the difficulty level of each of the offered courses that a student is considered to be eligible for enrollment in.

### 3.2. *Advising Software Architecture*

A quality academic advising system must be able to elicit requirements and expectations of the future users of the system. In order to comply with this particular need of the proposed system, two survey questionnaires were prepared. One questionnaire was designed for the students and included questions such as whether the students knew their academic advisors, how often they refer to their academic advisors prior to actual registration on chosen courses, how did a selection of courses affect their academic performance, whether the graduation delay was due to inappropriate selection of courses, and whether the difficulty level in the selected courses had a negative impact on their performance. The second survey questionnaire was intended for faculty members to elicit information on the number of advisees assigned to them, the time allotted to each advisee and the number of meetings held with advisees per semester, whether the advisor considered students' skills acquired from previous successfully completed courses in the advising process, reasons for delays in graduating, and additional features that need to be considered and included in the proposed system. The survey was conducted at ALHOSN University by interviewing faculty members and students. In addition, the questionnaires were also floated to faculty and student membership of three universities in the Emirate of Abu Dhabi through the website SurveyMonkey.

Responses were received from 35 faculty and 110 students. Feedback received from students indicate that 87% of the students knew their academic advisors, 75% of the participants met their advisors before deciding on the courses to register for the following term, and 51% of students stated that graduation delay is due to inappropriate selection of courses. On the other hand, with regard to the advising task, 97% of the participants who were assigned the advising task were faculty members, and 56% of the participants spend fifteen to thirty minutes per meeting for each advisee while 40% of the participants spend less time ranging between ten to fifteen minutes. It shows that a considerable amount of time is required to provide academic advising service especially for the faculty members who have a large number of advisees in addition to other commitments such as teaching, committee work, and research. It, thus, proves that there is an urgent need for the development of an effective electronic academic advising system to enable faculty members in reducing stress and the workload associated with the advising process thereby enabling faculty members to devote more time to their non-advisory responsibilities. The electronic advising system is also expected to help students in registration for the courses by varying the difficulty level of these courses resulting in a consequent reduction in graduation delays. The survey also revealed that 94% of the respondents consider only pre-requisites/co-requisites when recommending courses to their advisees.

This brings us to the question of what needs to be added to the currently existent student information systems to accommodate the new features of the proposed electronic academic advising system. The data requirements, in the proposed advising, are kept aligned with the information already collected in the existent university information system except for the need for an additional entity termed "student abilities". This can be achieved by universities by deciding on the required abilities assigned to each program curriculum and those needed for the successful completion of each course offered, along with their relative percentage weightage. This can be accomplished through the assignment of a subset within the orchestra of the existing student information collection process.

The proposed advising system is web-based. It is a three-layer application composed of presentation, business, and data layers. The former contains the user interface code and web forms operated by the administration, advisors, and students in order to interact with the system. The business is a service layer. It contains the ASP.net code that helps the presentation layer to interact with the database for processing purposes and to display the right feedback to the end-users. The data layer provides the connection to the SQL database.

The proposed internal structure consists of multiple modules. The system abilities administration is supervised by an administrator with the following options: manage university abilities, manage program abilities, manage course abilities, and system configuration settings. Another module, the academic advising system, offers services to students through which they may communicate and send requests to their academic advisors and access their academic details, major plans, FAQ, and course recommendations. It is an essential component and uses PCL, PC, and ability weight criteria to rank the recommended courses

The system abilities administration module (Figure 3) allows the administrator to input and manage all possible abilities. The manage program abilities protocol assigns a subset of the university abilities to any curriculum, whereas the manage course abilities protocol consigns a set of abilities to each course of the program with their respective percentage weightage. Based on each program revision, the set of abilities with their respective percentage weightages may change and, hence, options to add/delete/view/update abilities to a course can be accomplished. Finally, the system configuration settings enable the updating of university academic rules such as the maximum or minimum number of CHs a student is allowed to register in a semester. Based on the decision made by the university, a percentage value is assigned to each of the three criteria discussed earlier indicating the importance of each in the advising process.

The other key module, in the proposed advising system, is the academic advising protocol that consists of five sub-modules: academic details, major plan, FAQ, recommended courses, and contact advisor. A student can see all the details, via the academic details, such as failed/passed courses with their respective grades, current enrollment, grade point average (GPA) earned at the end of each semester, cumulative GPA (CGPA), and academic warnings given to the student. Students may also access their majors/study plans that show information about the curricula such as courses, categories, compulsory or electives, requirements, number of CHs, term in which they are planned to be taken, the total number of CHs needed to graduate, etc. FAQ helps students to get answers to their academic-related common questions. Contact advisor allows students to communicate with their academic advisors which is, currently, implemented using email but can be extended to accommodate social media and mobile apps.

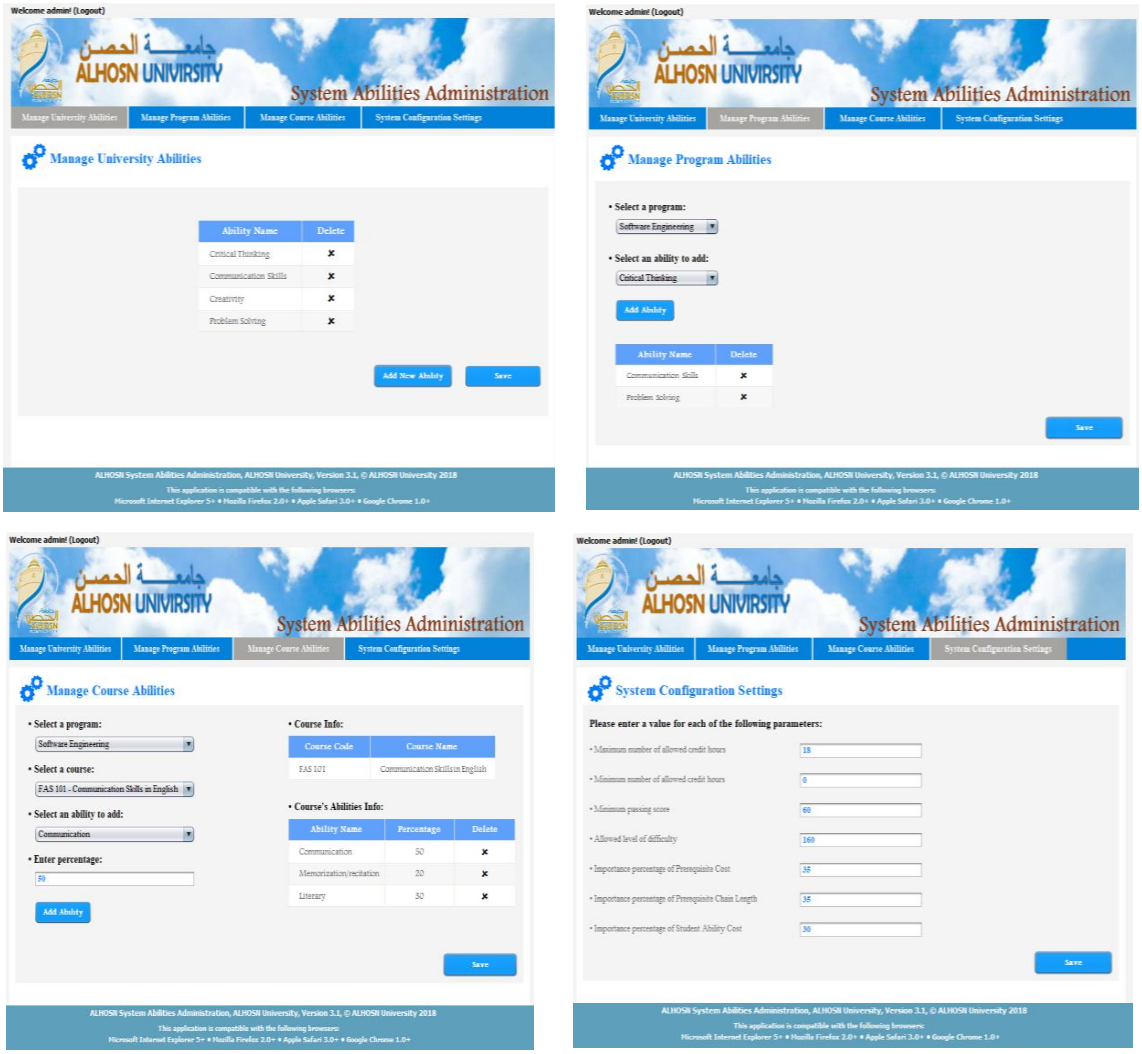

**Figure 3.** Manage university abilities (**top-left**), manage program abilities (**top-right**), manage course abilities (**bottom-left**), and system configuration settings (**bottom-right**) webpages.

## 4. Advising Algorithm

The core contribution of this research is the recommended courses sub-module of the academic advising system, which generates a list of recommended courses in which a student can register for the following academic term. This selection is and based on three criteria introduced earlier (PCL, PC, and abilities weight). In a five-step approach, the proposed advising system filters the offered courses by suggesting the ones in which a student is eligible to be enrolled. Depending upon each student's case and according to the three aforementioned criteria, these courses are ranked and prioritized (the highest-ranked course is the most recommended one). In addition, for each recommended course, an explanation justifying the importance of the recommended course showing the PCL and PC and the difficulty level of that course is provided. The proposed system takes as inputs the student's university ID and the list of offered courses for the generation of recommended courses, as shown in Figure 4, below.

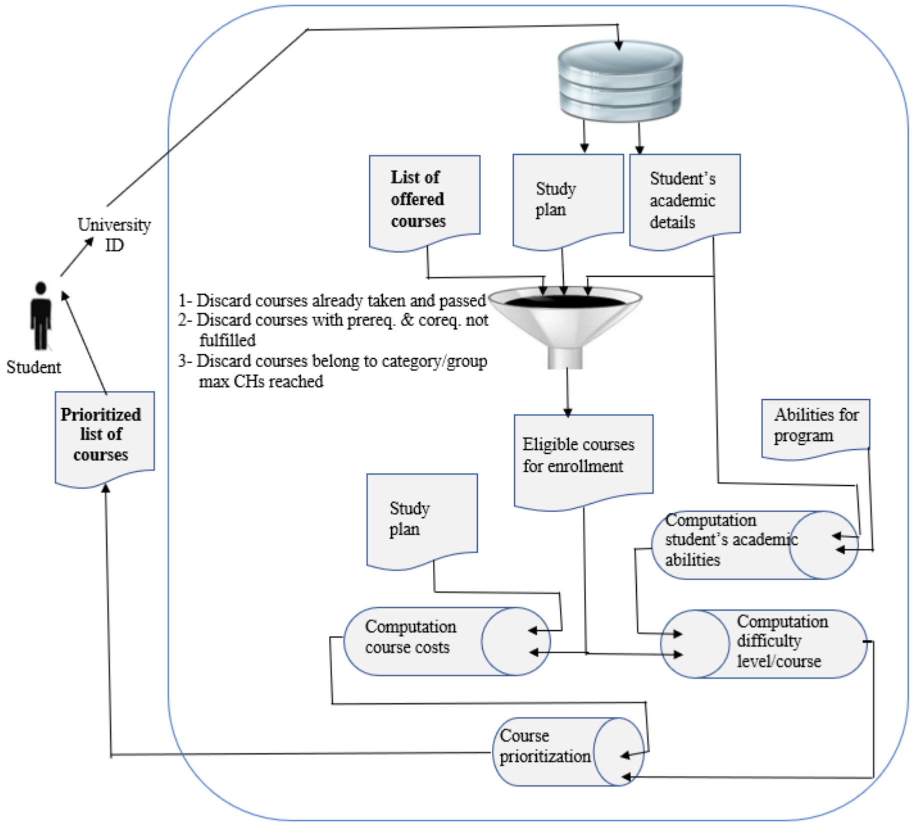


**Figure 4.** Framework of the generation and prioritization of recommended courses.

## 4.1. Phase 1

The filtration of the list of offered courses, to establish courses a student is eligible to register for, is the first step. There is, therefore, a requirement for the computation of the number of CHs of the courses, according to each category and accomplished by the student. This helps to know whether the student has reached the maximum allowed CHs in a certain category such as university general requirements or technical elective courses. This process is achieved by considering the list of offered courses and eliminating the ones already successfully completed by the student, courses which have prerequisites not yet fulfilled, and courses that belong to a group or category in which the maximum CHs have been achieved.

## 4.2. Phase 2

Computation of a student's level related to each ability $A_j$ of the given program, $SL_{A_j}$, is achieved by considering all courses that the student has already taken and successfully completed. This is given by the following equation, where $n$ equals the number of courses that the student has taken and completed. The equation also allows for the inclusion of each course $C_i$ with the corresponding grade earned by the student, $G(C_i)$, and the number of credit hours $CH(C_i)$. $P_{A_j}$ represents the percentage assigned to that ability $A_j$ in each course $C_i$.

$$SL_{A_j} = \frac{\sum_{i=1}^{n} G(C_i) \times P_{A_j}(C_i) \times CH(C_i)}{\sum_{i=1}^{n} P_{A_j}(C_i) \times CH(C_i)}. \quad (1)$$

To get a more accurate evaluation of students' abilities, it is possible to consider courses that a student has already failed. For simplicity reasons, these courses are excluded from this actual analysis. Initially and for each student, all abilities assigned to a program curriculum are set to 100; i.e., $SL_{A_j} = 100,\ \forall A_j\ with\ j \in [1,k]$ where $k$ is the number of abilities assigned to the program.

### 4.3. Phase 3

Next is the assessment of the student's difficulty level, $DL_{C_i}$, for each course, $C_i$. This is given by the equation, given below, where $k$ represents the number of abilities assigned to a course $C_i$ (in the list of courses generated in Phase 1 and using the previously calculated abilities level of the student).

$$DL_{C_i} = \sum_{j=1}^{k} P_{A_j}(C_i) \times \left(100 - SL_{A_j}\right). \tag{2}$$

### 4.4. Phase 4

Estimation of *PCL* and *PC* of each course in which the student is eligible to enroll (results of Phase 1) are calculated using the algorithm shown in Figure 5. Courses are stored in a one-dimensional array (a course $C_i$ of the study plan is taught in term $t_i$) while *pre_Co* represents a two-dimensional array holding prerequisites/co-requisites. Finally, a course $C_i$ has its prerequisites/co-requisites in *Pre_Co*[*i*][*] (*NbPre_Co* is the maximum number of prerequisites/co-requisites for a course. Typically, the numerical value of this value does not exceed three).

```
Algorithm: Computation of PCL and PC for a course Ci.

Inputs:
  A course Ci at index i in courses[NbCourses], planned in term ti

Outputs:   Ci, PCL, PC

begin
 PCL=PC=0
 for k=i+1 to NbCourses
  for l=1 to NbPre_Co
   if Pre_Co[k][l] == Ci, then
     PC++
     PCL=max(PCL,1+PCLComputaion(k+1,courses[k],term[k]))
   end if
  end for
 end for

 return Ci, PCL, PC
end.

PCLComputaion (j, c, t)
Begin
 if(j<=NbCourses) then
  for i=1 to NbPre_Co
   if(Pre_Co[j][i]==c) then
     PCL=1+ PCLComputaion (j+1,courses[j], term[j])
   else PCL= PCLComputaion (j+1,c,t)
   end if
  end for
 else PCL=0
 end if
 return PCL
end.
```

**Figure 5.** Algorithm to compute *PC* and *PCL* for a course $C_i$ and the final output will be the list courses generated in Phase 1, each with its respective *PCL* and *PC*.

The min–max method for normalization purposes of the PC and PCL values takes place on a 100 scale after the computation of PC and PCL for each course and student's difficulty level in each of these courses is established.

$$newValue = \frac{oldValue - min}{max - min} \times 100. \tag{3}$$

This framework aims at recommending courses, according to their difficulty level, by combining the easiest courses with the most difficult ones. This helps students to enhance their GPA by avoiding taking all the difficult courses, simultaneously, ensuring no graduation delay. For that, let us consider the eligible courses with the respective student's difficulty level and assuming they are stored in arrays $C$ and $DL$, respectively. These arrays will be first sorted in decreasing order. Afterward, a recommendation percentage is assigned to each course in such a way that it alternates between difficult courses and easy ones, i.e., the highest percentage will be assigned to the most difficult course followed by the next percentage assigned to the least difficult course and kept alternating until the end (algorithm shown in Figure 6). Let us assume that the recommended percentage associated with each eligible course is stored in a sorted array, *RPC*, and let $N_{ec}$ be the number of these courses. By defining the variable $step = 100/N_{ec}$, we can now compute the recommendation percentage for each course as follows.

**Algorithm: Computation of recommendation percentage for each eligible course.**

**Inputs:** List of courses the students is eligible to register for.
**Outputs**: RPC

```
begin
i=1; j=N_ec; c=0;
while (i ≤ j)
 begin
  RPC[i] = 100 – c x step
  if (j ≠ i) then
   begin
     RPC[j]= 100 –(c+1) x step
     j=j-1
   end
  i=i+1
  c=c+2
 end
 return RPC
End
```

**Figure 6.** Algorithm that assigns recommendation percentage to eligible courses (RPCs).

### 4.5. Phase 5

Prioritization of the recommended courses and ranking is based on the three criteria: PC, PCL, and student's difficulty level in each course. This stage aims to prioritize courses, according to their difficulty level, by combining the easiest subjects with the most difficult ones. As stated earlier, this helps students to enhance their GPA by avoiding taking all the difficult courses in the same academic term. Since the aforementioned three criteria are considered under this course prioritization, each of those criteria must be assigned a percentage to indicate the importance of each with regard to the others. This decision is left to the university policy whether to assign equal importance (or not) and whether all curricula obey the same policy. This can be set as shown in the following equation. Note that when the percentage of the ability weight criterion is assigned the value zero, the result of this phase is similar to that found in some previously proposed systems. Let us assume that the percentages assigned to PC, PCL, and ability weight (RPC) are $P_{PC}$, $P_{PCL}$, and $P_{RPC}$, respectively. Therefore, the final score, $FS_{C_i}$, assigned to each of the eligible courses, $C_i$, is calculated using the following equation.

$$FS_{C_i} = P_{PC} \times PC_{C_i} + P_{PCL} \times PCL_{C_i} + P_{RPC} \times RPC_{C_i}. \quad (4)$$

The courses with the highest final scores will be added to the list of recommended courses until reaching either the maximum CHs allowed for the student to register in that semester or the maximum allowed difficulty level set by the university. As mentioned earlier, the advising method adopted in this study follows the developmental model. It is left to students and academic advisors to decide once the list of recommended and prioritized courses is generated.

## 5. An Illustrative Example

Let us consider the curriculum of Appendix A (intended for software engineering students). This study plan shows all courses, pre-requisites/co-requisites and categories to which they belong, and corresponding terms when they should be taken. The minimum total number of CHs for this program equals 129. The number of terms is eight. Let us assume that the set of abilities assigned to this program is as follows: problem solving, creativity, analysis, communication skills, memorization, and literature. For the sake of simplicity, we assume that the only offered courses for the upcoming academic term are the ones presented in Figure 7, as below.

| |
|---|
| Islamic Culture |
| Programming I |
| Programming II |
| Introduction to Statistics |
| Calculus II |
| Differential Equations and Applications |
| Physical Sciences I |
| Digital Logic |
| Computer Organization and Assembly Language |
| Fundamentals of Database Systems |
| Computer Networks |
| Introduction to Discrete Structures |
| Operating Systems |
| Web Programming & Technologies |
| Introduction to Software Engineering |
| Computer Graphics |
| Multimedia Applications |
| Creative Thinking |

**Figure 7.** Offered courses for the upcoming term.

Let us consider the particular case study of a freshman and average student who has taken courses and respective grades are as Figure 8 reveals.

| Course Name | Grade |
|---|---|
| Communication Skills in English | 66 |
| Communication Skills in Arabic | 75 |
| Islamic Culture | 88 |
| Calculus I | 70 |
| Programming I | 71 |

**Figure 8.** Courses taken by the student with their respective grades.

In consonance with our proposed system, the remaining CHs for each category/group of courses are shown in Figure 9, below.

| Category | Group | Remaining Credit Hours |
|---|---|---|
| University General Req. | Compulsory | 12 |
| University General Req. | Elective | 3 |
| Faculty Req. | Compulsory | 27 |
| Faculty Req. | Elective | 0 |
| Program Req. | Compulsory | 63 |
| Program Req. | Elective | 9 |

**Figure 9.** Computation of the remaining CHs per group/category.

Figure 10 is the result of Phase 1 where courses are classified as courses already taken, courses with prerequisite/co-requisite not fulfilled yet, courses which belong to a group/category with the maximum number of CHs has been reached, and courses the student is eligible to register in for the next semester.

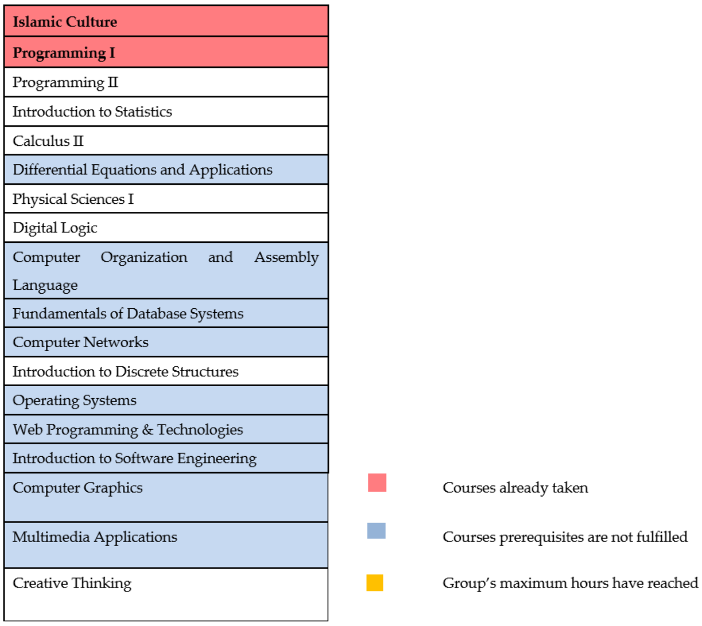


**Figure 10.** Course classification.

It, thus, follows that the only courses that the student is eligible to register for in the upcoming term are the one shown in Figure 11.

| |
|---|
| Programming II |
| Introduction to Statistics |
| Calculus II |
| Physical Sciences I |
| Digital Logic |
| Introduction to Discrete Structures |
| Creative Thinking |

**Figure 11.** Eligible courses.

To compute the student's level regarding each ability becomes central. Table 1 presents the percentage assigned to each ability, and for each course already taken or may be taken in the following term. Note that a cell with no value indicates the assignment of zero percent to the corresponding ability.

**Table 1.** Percentage of each ability for each passed and eligible course.

| | Abilities | | | | | |
|---|---|---|---|---|---|---|
| **Course** | **Problem Solving** | **Creativity** | **Memorization** | **Literature** | **Analysis** | **Communication Skills** |
| **Communication Skills in English** | - | - | 20 | 30 | - | 50 |
| **Communication Skills in Arabic** | - | - | 20 | 30 | - | 50 |
| **Islamic Culture** | - | - | 30 | 30 | - | 40 |
| **Calculus I** | 60 | - | 10 | - | 15 | 15 |
| **Programming I** | 50 | 10 | 5 | - | 20 | 15 |
| **Programming II** | 50 | 10 | 5 | - | 20 | 15 |
| **Introduction to Statistics** | 65 | - | 5 | - | 15 | 15 |
| **Calculus II** | 60 | - | 10 | - | 10 | 20 |
| **Physical Sciences I** | 50 | - | 10 | - | 30 | 10 |
| **Digital Logic** | 50 | - | 5 | - | 40 | 5 |
| **Introduction to Discrete Structures** | 70 | - | - | - | 20 | 10 |
| **Creative Thinking** | - | 25 | - | 25 | 25 | 25 |

In accordance with the first two equations, the student's ability and difficulty level for each course are given in Table 2.

**Table 2.** Student's ability level and student's difficulty level.

| **Ability ($A_j$)** | **Student's Level ($SL_{A_j}$)** | **Course ($C_i$)** | **Difficulty Level ($DL_{C_i}$)** |
|---|---|---|---|
| Problem solving | 70.5 | Programming II | 28.4 |
| Creativity | 70 | Introduction to Statistics | 28.3 |
| Memorization | 82.2 | Calculus II | 27.5 |
| Literature | 76.3 | Physical Sciences I | 27.9 |
| Analysis | 70.6 | Digital Logic | 28.7 |
| Communication Skills | 74.6 | Introduction to Discrete Structures | 29.1 |
| | | Creative Thinking | 27.1 |

Table 3 summarizes the results obtained and grounded in the three given criteria (PC, PCL, and student's difficulty level) for each eligible subject. Note that the results are shown in a percentage format as a result of Equation (3). The final recommendation percentage for each course is shown in Table 4.

**Table 3.** PC, PCL, and student's difficulty level for the given eligible courses.

| **Course** | **PC** | **PC (%)** | **PCL** | **PCL (%)** | **Difficulty Level ($DL_{C_i}$)** |
|---|---|---|---|---|---|
| Programming II | 4 | 80 | 4 | 100 | 28.4 |
| Introduction to Statistics | 4 | 80 | 2 | 50 | 28.3 |
| Calculus II | 2 | 40 | 2 | 50 | 26.9 |
| Physical Sciences I | 1 | 20 | 1 | 25 | 27.9 |
| Digital Logic | 1 | 20 | 3 | 75 | 28.8 |
| Introduction to Discrete Structures | 5 | 100 | 3 | 75 | 29.1 |
| Creative Thinking | 0 | 0 | 0 | 0 | 27.1 |

**Table 4.** Recommendation percentage for each eligible course.

| Course ($C_i$) | Difficulty Level ($DL_{C_i}$) | Recommendation Percentage |
|---|---|---|
| Introduction to Discrete Structures | 29.10 | 100 |
| Digital Logic | 28.70 | 71.43 |
| Programming II | 28.35 | 42.86 |
| Introduction to Statistics | 28.31 | 14.28 |
| Physical Sciences I | 27.91 | 28.57 |
| Calculus II | 27.52 | 57.14 |
| Creative Thinking | 27.12 | 85.71 |

The final step is the calculation of the final score and the prioritization of each recommended course. Let us assume that the allocated importance factor of PC, PCL, and ability weightage is 35%, 35%, and 30%, respectively. Then, $FS_{C_i}$ is computed for each eligible course using Equation (4) as illustrated in Table 5. Note that in this list below, the recommended courses are shown in bold according to the following restrictions: it presumes that the maximum number of CHs allowed for that student is 18 and it assumes that the maximum allowed difficulty level is 160.

**Table 5.** Final score for each eligible course.

| Eligible Courses | Final Score |
|---|---|
| **Introduction to Discrete Structures** | **91.25** |
| **Programming II** | **75.858** |
| **Digital Logic** | **54.679** |
| **Introduction to Statistics** | **49.784** |
| **Calculus II** | **48.642** |
| Creative Thinking | 25.713 |
| Physical Sciences I | 24.321 |

Figure 12 displays the outcome available to students along with the explanation of the factors that led a course to be recommended for the upcoming term.

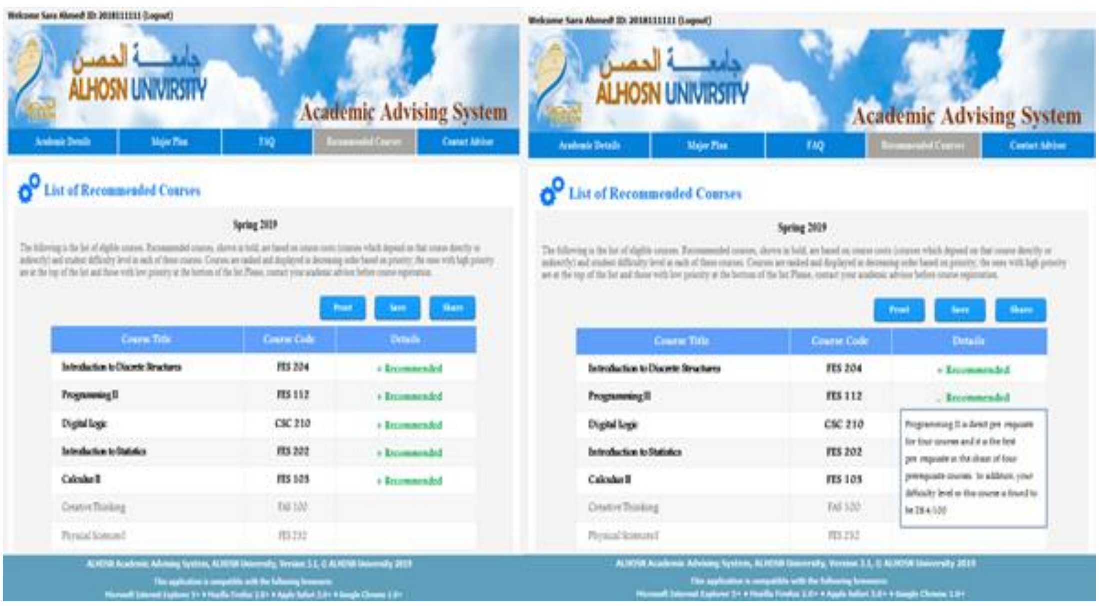


**Figure 12.** List of recommended courses displayed to students (**left**) and details about a selected course (**right**).

## 6. Conclusions

In this research paper, we proposed an academic advising framework based on the developmental advising model. This approach differs from the previously published proposals in that it does not only consider course requirements such as PCL and PC but also students' ability levels in the courses that they are eligible to register for in the following academic term. This variable factor is used to compute students' difficulty levels in each eligible course based on their academic performance in the previous terms. A list of recommended courses is, then, generated and prioritized to combine difficult and easy subjects that should be taken by a student in the upcoming term. Consequently, the system helps students to not take courses all at once which have been identified difficult which might impact negatively on their CGPAs. This also enables students to avoid unnecessary graduation delays.

The central concept of introducing an ability weight component, presented in our proposed academic advising system, can be easily be integrated into the existing university systems without requiring major changes. It is also worth pointing that the proposed schema of assigning abilities to program curricula and keeping track of students' abilities in each course may be extended, as a significant tool, to monitor students' progress and development of their abilities/skills during their studies. Finally, the findings can be further applied to assess each learning outcome achievement and to propose new teaching and assessment approaches for the continual enhancement of academic programs.

Our proposed advising system has been tested in ALHOSN University, on three bachelor programs: software engineering, architectural engineering, and industrial engineering. Several live cases have been taken from each program ranging from low-academic performance to high-academic performance. The results obtained from the system have been investigated by academic advisors from the subject-associated departments. Feedback from the academic advisors has been highly positive. The list of recommended courses generated, and the way courses have been prioritized, has been found to be aligned with advisors' expectations. The next stage in testing our proposed system would involve its sample testing in a larger sample of educational institutions. The resultant feedback is required to support the claimed system effectiveness in facilitating the timely graduation of students along with the ease with which the system can be incorporated in existing computing environments.

## 7. Limitations

The suggested algorithm does not cover technical analysis on the relevancy of any particular curriculum and associated subjects. For the prioritization of the recommended courses, the decision defining weightage ranking is based on three criteria (PC, PCL, and student's difficulty level of each course) which may potentially lead to different results. Finally, the example reported in this paper does not claim to cover all options/scenarios, particularly those related to students who were unsuccessful in completing some courses.

**Author Contributions:** Conceptualization and methodology, S.L. and L.G.; software, L.G. and S.L.; validation, S.L. and L.G.; formal analysis, S.L.; investigation, S.L., L.G. and J.N.; resources, L.G.; data curation, S.L. and L.G.; writing—S.L.; writing—review and editing, S.L. and J.N.; visualization, S.L., L.G. and J.N.; supervision, S.L. and J.N.; project administration, S.L. and J.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** Number of CHs required for each category and group of courses.

| Category | Group | Remaining Credit Hours |
|---|---|---|
| University General Req. | Compulsory | 21 |
| University General Req. | Elective | 3 |
| Faculty Req. | Compulsory | 33 |
| Faculty Req. | Elective | 0 |
| Program Req. | Compulsory | 63 |
| Program Req. | Elective | 9 |

**Table A2.** Eight-semester study plan.

| Course No | Course Title | CH | Pre-Co |
|---|---|---|---|
| | **FRESHMAN YEAR (I)** | | |
| | **First Semester (Fall)** | | |
| FAS 108 | Ethics | 3 | - |
| FAS 101 | Communication Skills (English) | 3 | - |
| FAS 102 or FAS 109 | Comm. Skills (Arabic) or Human Rights in Law and Sharia | 3 | - |
| FES 111 | Programming I | 3 | - |
| FES 102 | Calculus I | 3 | - |
| | **Second Semester (Spring)** | | |
| FAS 103 | Islamic Culture | 3 | - |
| FBA 100 | Intro to Economics | 3 | - |
| FES 103 | Calculus II | 3 | FES 102 Pre |
| FAS 120 | Scientific and Technical Writing | 3 | FAS 101 Pre |
| FES 112 | Programming II | 3 | FES 111 Pre |
| FES 201 | Matrix Algebra for Engineers | 3 | - |
| | **SOPHOMORE YEAR (II)** | | |
| | **First Semester (Fall)** | | |
| FAS 106 | History of Sciences | 3 | - |
| FES 202 | Intro to Statistics | 3 | FES 102 Pre |
| FES 232 | Physical Science I | 3 | FES 102 Pre |
| FES 204 | Introduction to Discrete Structures (Discrete Math) | 3 | FES 102 Pre |
| FES 206 | Calculus III | 3 | FES 103 Pre |
| CSC 225 | Advanced Data Structures and Algorithm Analysis | 3 | FES 112 Pre |
| | **Second Semester (Spring)** | | |
| FES 150 | Natural Science | 3 | - |
| CSC 210 | Digital Logic | 3 | FES 111 Pre, FES 204 Co |
| FES 207 | Differential Equations and applications | 3 | FES 103 Pre |
| FES 233 | Physical Science II | 3 | FES 232 Pre |
| SWE 265 | Intro to Software Engineering | 3 | FES 112 Pre |

**Table A2.** *Cont.*

| Course No | Course Title | CH | Pre-Co |
|---|---|---|---|
| | **JUNIOR YEAR (III)** | | |
| | **First Semester (Fall)** | | |
| CSC 327 | Fundamentals of Database Systems | 3 | FES 112 Pre |
| CSC 230 | Computer Organization and Assembly Language | 3 | FES 112 Pre, CSC 210 Pre |
| SWE 310 | Human Computer interfaces | 3 | SWE 265 Pre |
| FreeXXX | Free Elective | 3 | - |
| SWE 320 | Software Project Management | 3 | SWE 265 Pre |
| CSC 350 | Computer Ethics | 3 | SWE 265 Pre |
| | **Second Semester (Spring)** | | |
| CSC 330 | Computer Architecture | 3 | CSC 230 Pre, FES 202 Pre |
| SWE 362 | Software Design and Architecture | 3 | SWE 265 Pre |
| CSC 360 | Computer Networks | 3 | CSC 225 Pre, FES 202 Pre |
| CSC 371 | Web Programming & Technologies | 3 | CSC 327 Pre |
| CSC 370 | Operating Systems | 3 | CSC 330 Pre, CSC 225 Pre |
| SWE 497 | Internship | 0 | Min 84 CH |
| | **SENIOR YEAR (IV)** | | |
| | **First Semester (Fall)** | | |
| CSC 429 | Computer and Network Security | 3 | CSC 360 Pre, FES 204 Pre |
| CSC 425 | Theory of Computing | 3 | CSC 225 Pre, FES 204 Pre |
| SWE 498 | Capstone Project-1 | 3 | SWE 320 Pre, Min 90 CH |
| SWE 421 | Software Requirements and Specification | 3 | FES 204 Pre, SWE 265 Pre |
| Tech. Elective | SWE 4XX or CSC 4XX Elective | 3 | - |
| | **Second Semester (Spring)** | | |
| SWE 425 | Software Testing and Measurement | 3 | SWE 362 Pre |
| SWE 426 | Software Quality Control | 3 | SWE 362 Pre |
| SWE 499 | Capstone Project-2 | 3 | SWE 498 Pre |
| Tech. Elective | SWE 4XX or CSC 4XX Elective | 3 | - |
| Tech. Elective | SWE 4XX or CSC 4XX Elective | 3 | - |

**Table A3.** List of program elective courses.

| | **Program Requirements Electives (9 CH).** | | |
|---|---|---|---|
| **Course** | | **CH** | **Pre-Requisites** |
| CSC 455 | Computer Graphics | 3 | CSC 225 Pre, FES 201 Pre, |
| CSC 457 | Selected Topics in Programming | 3 | CSC 225 Pre, SWE 362 Pre |
| CSC 460 | Programming Languages & Compiler | 3 | CSC 225 Pre, CSC 425 Pre |
| CSC 461 | Object Oriented Programming and C++ | 3 | CSC 225 Pre, SWE 265 Pre |
| CSC 462 | Object Oriented Design Patterns | 3 | CSC 225 Pre, SWE 362 Pre |
| CSC 464 | Modeling and Simulation | 3 | CSC 225 Pre, FES 202 Pre |
| CSC 475 | Analysis of Algorithms | 3 | CSC 225 Pre, FES 204 Pre |
| CSC 480 | Numerical Methods | 3 | CSC 225 Pre, FES 201 Pre, FES 207 Pre |
| SWE 451 | Multimedia Applications | 3 | CSC 225 Pre, FES 201 Pre |
| SWE 462 | Distributed Systems | 3 | CSC 360 Pre |
| SWE 465 | Advanced Topics in Software Engineering | 3 | SWE 362 Pre |
| SWE 471 | Software Evolution | 3 | SWE 362 Pre |
| SWE 475 | Software Systems Development | 3 | SWE 362 Pre |
| SWE 484 | Artificial Intelligence | 3 | CSC 225 Pre, FES 201 Pre, FES 202 Pre, FES 204 Pre |
| SWE 485 | Pattern Recognition | 3 | CSC 225 Pre, FES 202 Pre, FES 204 Pre, FES 201Pre |
| SWE 486 | Robotics | 3 | CSC 225 Pre, FES 201 Pre |

**Table A4.** University general requirements-elective courses.

| Elective [Total Hours: 3 HRS] The Student Must Select ONE Elective Course the List of Courses That Are Outside the Major | | | |
|---|---|---|---|
| Course | | CH | Pre-Requisites |
| ELE 901 | Free Elective 1 | 3 | - |
| FAS 100 | Creative Thinking | 3 | - |
| FAS 104 | Globalization | 3 | - |
| FAS 105 | Scientific Miraculousness of the Quran | 3 | - |
| FAS 107 | Sports in our Life | 3 | - |
| FAS 109 | Human Rights in Law and Shari'ah | 3 | - |
| FAS 130 | Introduction to Sociology | 3 | - |
| FAS 210 | Politics, Government and Planning | 3 | - |
| FAS 220 | Understanding Society | 3 | - |
| FBA 102 | Introduction to Entrepreneurship | 3 | - |
| FES 160 | Research Methods | 3 | - |
| FES 280 | Landscaping & Flower Arrangement | 3 | - |
| FES 281 | Human & Health | 3 | - |
| FES 282 | Interior Design | 3 | - |

## References


1. Negreiros, J.; Oliveira, M. Contemplation on today's digital education. In *New Contributions in Information Systems and Technologies*; Rocha, A., Correia, A., Costanzo, S., Reis, L., Eds.; Advances in Intelligent Systems and Computing; Springer: Berlin/Heidelberg, Germany, 2015; Volume 353.
2. Negreiros, J.; Lei, W. Web 2.0 tool recommendations for teachers. *Int. J. Innov. Res. Educ. Sci.* **2019**, *6*, 513–525.
3. McConnell, R. Student perspectives of academic advisors and leader development at the graduate level. *J. Contin. High. Educ.* **2018**, *66*, 146–157. [CrossRef]
4. Mu, L.; Fosnacht, K. *Effective Advising: How Academic Advising Influences Students Learning Outcomes in Different Institutional Contexts*; American Educational Research Association: Washington, DC, USA, 2008.
5. Crocker, R.; Allen, C.; Kahla, M. Fixing advising: A model for faculty advising. *Res. High. Educ. J.* **2014**, *26*, 1–9.
6. Davis, J. Intrusive Advising and Its Influence on First and Second Year Students: A Formative Evaluation of a Pilot Intrusive Advising Initiative at a Hbcu in the South. Ph.D. Thesis, Florida State University, Tallahassee, FL, USA, 2015.
7. Jones, E. An Examination of the Effects of Intrusive Advising and Support Services on Academically at-Risk Students. Ph.D. Thesis, Liberty University, Lynchburg, VA, USA, 2013. Available online: digitalcommons.liberty.edu/doctoral/666/ (accessed on 15 July 2019).
8. Varney, J. Proactive advising. In *Academic Advising Approaches: Strategies that Teach Students to Make the Most of College*; Manhattan, K.S., Drake, J.K., Jordan, P., Miller, M.A., Eds.; NACADA-The Global Community for Academic Advising: Manhattan, KS, USA, 2013; pp. 137–154.
9. Donaldson, P.; McKinney, L.; Lee, M.; Pino, D. First-year community college students' perceptions of and attitudes toward intrusive academic advising. *NACADA J.* **2016**, *36*, 30–42. [CrossRef]
10. Anderson, L.; Krathwohl, D. (Eds.) *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives: Complete Edition*; Longman: New York, NY, USA, 2001.
11. Dalto, J. Teaching Knowledge: The Cognitive Domain of Learning and Learning Objectives. 2013. Available online: www.convergencetraining.com/blog/the-cognitive-domain-of-learning-and-learning-objectives (accessed on 10 July 2019).
12. Nagy, H.; Aly, W.; Hegazy, O. An educational data mining system for advising higher education students. *Int. J. Comput. Inf. Sci. Eng.* **2013**, *7*, 1266–1270.
13. El-Sheikh, O.; Mohammed, B.; Emad, H.; Zoromba, M. Developing corrective actions to improve academic advising process. *Am. J. Nurs. Res.* **2019**, *7*, 286–292.
14. Jaime, A.; Olarte, J.; García-Izquierdo, F.; Domínguez, C. The effect of internships on computer science engineering capstone projects. *IEEE Trans. Educ.* **2020**, *63*, 24–31. [CrossRef]

15. Mueller, D.; Meyer, A. Design a Sustainable Online Advising Option. NACAD. 2017. Available online: Nacada.ksu.edu/Resources/Academic-advising-Today/View-Articles/Design-a-Sustainable-Online-Advising-Option.aspx (accessed on 12 February 2019).
16. Gordon, C. The Difference in Satisfaction among Academic Advising Formats for Online College Student. Doctoral Dissertations and Projects. 2020. Available online: Digitalcommons.liberty.edu/doctoral/2494 (accessed on 10 May 2020).
17. Choudhari, R. Expert System for Student Advisement. Annual Meeting of SEDSI-Charleston South Carolina. 2009. Available online: www.sedsi.org/2009_Conference/proc/proc/p080911001.pdf (accessed on 15 May 2019).
18. Chakraborty, J.; Thopugunta, G.; Bansal, S. Data extraction and integration for scholar recommendation system. In Proceedings of the IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 397–402. [CrossRef]
19. Matulatan, T.; Resha, M. Deep learning on curriculum study pattern by selective cross join in advising students' study path. In Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 477–481. [CrossRef]
20. McMahan, B. An automatic dialog system for student advising. *J. Undergrad. Res.* **2010**, *10*. Available online: cornerstone.lib.mnsu.edu/jur/vol10/iss1/6 (accessed on 15 May 2019).
21. Latorre-Navarro, E.; Harris, J. An intelligent natural language conversational system for academic advising. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 110–119.
22. Gavriushenko, M.; Saarela, M.; Kärkkäinen, T. Supporting institutional awareness and academic advising using clustered study profiles. In Proceedings of the 9th International Conference on Computer Supported Education, Porto, Portugal, 21–23 April 2017; Volume 1, pp. 35–46.
23. Hsu, S.; Marques, O.; Ilyas, M.; Ding, X. Web-based undergraduate academic advising system. In Proceedings of the International Conference on Engineering Education, Manchester, UK, 21–18 August 2002; pp. 1–6.
24. Ozturan, M.; Ayan, Z. BUADVIS—A decision support system for student advising. *J. Appl. Sci.* **2001**, *1*, 3–5.
25. Bansal, A.; Cline, K.; Rangel, K.; Zunke, R. KRAK: Knowledge-based resource advising kit. *J. Comput. Sci. Coll.* **2003**, *18*, 245–253.
26. Mihali, R.; Sobh, T.; Vamoser, D. SKED: A course scheduling and advising software. In Proceedings of the American Society of Engineering Education (ASEE) Conference, Albuquerque, NM, USA, 24–27 June 2001.
27. Liu, J.; Xia, F.; Wang, L.; Xu, B.; Kong, X.; Tong, H.; King, I. Shifu2: A network representation learning based model for advisor-advisee relationship mining. *IEEE Trans. Knowl. Data Eng.* **2019**. [CrossRef]
28. Albalooshi, F.; Shatnawi, S. HE-Advisor: A multidisciplinary web-based higher education advisory system. *Glob. J. Comput. Sci. Technol.* **2010**, *10*, 37–49.
29. Aly, W.; Eskaf, K.; Selim, A. Fuzzy mobile expert system for academic advising. In Proceedings of the IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–5. [CrossRef]
30. Alkhoori, A.; Kuhail, M.; Alkhoori, A. UniBud: A virtual academic adviser. In Proceedings of the 12th Annual Undergraduate Research Conference on Applied Computing (URC), Dubai, UAE, 15–16 April 2020; pp. 1–4. [CrossRef]
31. Gutiérrez, F.; Seippa, K.; Ochoab, X.; Chiluiza, K.; Laeta, T.; Verberta, K. LADA: A learning analytics dashboard for academic advising. *Comput. Hum. Behav.* **2020**, *107*. [CrossRef]
32. Sandvig, J.; Burke, R. *AACORN: A CBR Recommender for Academic Advising*; Technical Report TR05-015; DePaul University: Chicago, IL, USA, 2005.
33. Daramola, O.; Emebo, O.; Afolabi, I. Implementation of an intelligent course advisory expert system cased-based course advisory expert system. *Int. J. Adv. Res. Artif. Intell.* **2014**, *3*, 6–12. [CrossRef]
34. ABET. Accreditation Board for Engineering and Technology. Available online: www.abet.org (accessed on 15 July 2019).
35. CAA. Standards for Licensure and Accreditation. Available online: www.caa.ae/caa/images/standards2011.pdf (accessed on 15 July 2019).