A WWW INTERFACE FOR ELECTIVE COURSE PRE-REGISTRATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF SOCIAL SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

Ş. TAYFUN HIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF BUSINESS ADMINISTRATION
IN
THE DEPARTMENT OF MANAGEMENT

MAY 1997

Approval of the Graduate School of Social Science.

Prof. Dr. Bahattin Akşit
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Business Administration.

Prof. Dr. Alaeddin Tileylioğlu
Chairman of The Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope of quality, as a thesis for the degree of Master of Business Administration.

Assist. Prof. Dr. Cemal Akyel
Supervisor

Examining Committee Members

Prof. Dr. Neşe Yalabık

Assist. Prof. Dr. Zeynep Onay

Assist. Prof. Dr. Cemal Akyel

ABSTRACT

A WWW INTERFACE FOR ELECTIVE COURSE PRE-REGISTRATIONS

Hız, Ş. Tayfun

M.B.A., Department of Management

Supervisor: Assist. Prof. Dr. Cemal Akyel

May 1997, 124 pages

This thesis aims at defining an alternative system for the elective course registrations at the Department of Management of Middle East Technical University. The proposed system collects the bids from the students where they assign bid points to their preferred courses. While making the allocation these preferences of students are taken into consideration by giving priority to higher bid points. By implementing the system as a Web-based application, the problem of accurate information flow among the parties involved in the system has also been resolved.

Keywords: Auction Method, Bidding, Elective Course Registrations, Web-based Application

ÖZ

SEÇİMLİK DERS ÖNKAYITLARI İÇİN WWW TABANLI ARABİRİM

Hız, Ş. Tayfun

Yüksek Lisans, İşletme Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Cemal Akyel

Mayıs 1997, 124 sayfa

Bu tez Orta Doğu Teknik Üniversitesi İşletme Bölümü'ndeki seçimlik ders kayıtları için bir alternatifi sistem tanımlamayı amaçlamaktadır. Önerilen sistem öğrencilerin tercih ettikleri seçimlik derslere ihale miktarları atadıkları ihaleleri toplamaktadır. Daha sonra dersler dağıtılırken bu tercihler yüksek ihale miktarlarına öncelik verilmesi yoluyla dikkate alınmaktadır. Sistem Web-tabanlı bir uygulama olarak geliştirilmiş, böylece sistem içinde yeralan kullanıcılar arasında güvenli bilgi akışı sağlanmıştır.

Anahtar Sözcükler: Açık Artırma Metodu, İhale, Seçimlik Ders Kayıtları, Web-tabanlı Uygulamalar

ACKNOWLEDGEMENTS

I would like to express sincere appreciation to Assist. Prof. Dr. Cemal Akyel for his guidance and insight throughout the research.

Special thanks go to other faculty members, Prof. Dr. Neşe Yalabık and Assist. Prof. Dr. Zeynep Onay, for their suggestions and comments.

Finally, I would like to thank to all friends who had supported me throughout the study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Throughout the undergraduate education, the regular students of the Department of Management at the Middle East Technical University are required to take nine elective courses along with their must courses. One elective course must be registered on the second semester of the third year, four on the first semester of the fourth year and four on the second semester of the fourth year.

Each semester, the Department offers around 25 to 30 elective courses in various fields of management. Most of the courses are taught by the full-time faculty members, while some courses are given by part-time instructors and invited professors. The capacities of the courses are usually 30, 50 or 90 depending on the popularity of the courses. These capacities are determined by considering various factors such as the number of student that the instructor prefers, the physical capacity of the classrooms or laboratories, the popularity of the course among the students and the semesters on which the course is being offered (some courses are offered on Fall or Spring semesters only). Actually, for some courses, the initially determined capacities are modified to allow at most five to fifteen more students to take the course.

The demand for some courses is not too high when compared with the available capacity, so the students preferring that course can register it without facing a major problem. However, as it occurs in many demand-supply scenarios, when the number of students preferring this course exceeds the capacity of the course, the problem of fair allocation of the available capacity to the students arises. The

capacities of the courses should be allocated to students in such a way that the course capacity constraints are met. In the mean time, the perception of fairness in the eyes of the students should be preserved. A student's right to register the elective courses that he[1] prefers is not questionable. On the other hand, the best conditions should be provided to the instructors and students to ensure an effective learning environment.

There are several ways to assign students to courses. One method is to consider the student's success in certain courses or his overall success. But, this system does not ensure the student's right to select the electives that he really wants. This can also result in an unfair condition where the most popular courses are filled up by successful students, while those with lower levels of success will be obliged to register less popular courses. As an alternative, the instructor may select his students. In this case, the selection criteria are hard to determine and the objectivity of the final class list is questionable. Other methods can be a lottery for each course to select the course list from the willing students or the random assignment of students to courses by the Department. Again, the fairness of these methods is extremely doubtful.

The problem of allocating the students to the courses can be solved by making use of the First Come - First Served method. In the systems where this method is applied, the client that asks for a service before the others will be replied before the others. A client has to wait by giving priority to the clients who joined the system earlier than him.

This method has actually been used for a long time in The Department of Management. The details and positive/negative aspects of this system are discussed in the third chapter.

[1] Throughout this report, every time a reference is made to a person, the pronoun "he" will be used, instead of "he/she".

While registering elective courses, students try to give priority to their preferred courses. A student prefers a course because he has an interest in the topics covered in the course. Therefore, the method for placing the students to elective courses should be based on the preferences. The literature contains a number of studies that uses this approach. In Chapter 2, such systems are explained.

A study that takes the preferences into consideration has been proposed by Sezer (1996). This proposed system has actually been used for the registrations of the Fall Semester of the 1996-1997 Academic Year. The details of the system are also discussed in Chapter 3. Though the system was satisfactory enough in terms of placing the students to elective courses, the system had lacking dimensions. These deficiencies will also be discussed in the chapter.

In the fourth chapter, a new structure that integrates the system with its users (the students, the advisors, the instructors, the department, and the supervisor) is described. It also resolves the problems faced in the system used in the Fall Semester of the 1996-1997 Academic Year.

As the system is designed to be used by the Department of Management, the system proposed in the fourth chapter is also implemented as a computer software. The implementation issues of this software are explained in the fifth chapter.

The last chapter is the conclusion, which discusses the aspects of the proposed system.

CHAPTER 2

LITERATURE REVIEW

The scope of the present study is twofold: The allocation of students to elective courses by an auction method and the implementation of such systems on the Web-space. The following sections list studies that are concerned with either one or both of these features.

2.1 - Studies Concerning the Allocation Procedures

The problem of allocating scarce "resources" to a number of "tasks", where the tasks are the students and the resources are elective courses, has lead to a number of studies.

One of the studies on allocating elective courses to students is due to Macon and Walker (1966). The aim was to allocate students to course sections some of which are too popular. The course sections are determined by selecting random students from the set of students desiring to register that section.

In Washington State University, Busam (1967) developed a similar system. The main difference was that the random selection is done by a computer program rather than by hand.

Allocating students to dormitory rooms is a problem that is very similar to the problem investigated in this study. The problem is simpler in this case, because each student is assigned to a single dormitory room rather than a number of elective courses. Hylland and Zeckhauser (1979) have carried a such study. The system introduces the concept of fictitious points in order to eliminate the unfairness of using monetary measures. The students use these points to express their preferences. The system takes the points into consideration while making the allocation.

Warner (1976) resolves another similar problem again with the use of fictitious points. The management of a hospital should assign the right number of nurses to different departments so that the level of service meets the needs of different workloads. However, the perception of fairness of the this allocation in the eyes of nurses is crucial for ensuring the quality of their services. This time, the fictitious points are called as penalties. To select a certain shift in a day or a certain day in a week, the nurse promises to pay an amount of penalties. For a period of usually four weeks, the nurse distributes the given 50 points to her selected days and shifts. On the other hand, the management of the hospital determines the required capacities for different shifts and days in each department. Then, the system uses a mathematical programming model to decide on a schedule. The system has been used in a number of hospitals like University of Michigan Hospital.

A course registration system has also been developed at the University of Chicago Graduate School of Business (GSB) [Graves, Schrage and Sankaran (1981)]. The problem is slightly different than the one considered in this study, in the sense that the system allocates course schedules rather than individual courses. As the popularity of different courses and sections are not equal, a need to allocate them fairly emerges. The students are again given fictitious credit points. The system uses the auction mechanism. In a bid, students bid for four alternative schedules (four different sets of courses and sections) and assign an amount of bid points for each. The schedules of all students sorted by bid amounts, then by the position among the four alternative schedules of the student (first schedule having the highest priority), then by the time of submission. The system evaluates each schedule starting from the one at the top of the sorted list. If all of the courses specified in the schedule are still not filled (i.e. the course is not closed) the schedule is successful. Otherwise, the next schedule of the student is considered. If none of the schedules of a student is successful, the system allows him to bid for individual courses in four Drop-Add-Swap

(DAS) rounds. For the courses that are not closed, the system does not charge any bid points. For the closed courses, the system determines a closing price by using an optimization model. Any bid amounts above the closing prices are returned back to the student. For example, if the course closes at a price of 40 points, a student has a bid amount of 120 will receive 80 points back.

The differences between the system of Graves et al. (1981) and the present study are:

- The size of the problem: At GSB, the system works for 18000 students and 475 courses. The number of students of the Department of Management who will use the system will be at most 350 and the number of courses is usually around 30.
- The nature of the resource being allocated: The system at GSB allocates schedules while the problem studied in this report is concerned with allocating individual courses.
- The approach to the problem: Graves et al. (1981) resolve the problem with an optimization model while in this study an allocation approach is used.

2.2 - WWW Based Course Auction Systems

With the use of Internet, a some of the existing course auction systems have been modified to work on it, while new ones have been developed.

Online Registration Bidding System of the Graduate School of Business at the University of Chicago

One system that has been ported to work on Internet is the course registration system of the University of Chicago Graduate School of Business (1997). The registration system described in Section 2.1 is modified to run on a computer

network. However, the development team of the system preferred to implement it as a special software that runs on the network rather than using the standard WWW interfaces with the reason that their On-Line Registration Bidding system is too complicated to be implemented on the Web, unless powerful tools, such as Java programming language, becomes reliable enough. The system has been developed using the OpenRoad package of Computer Associates Inc. and runs on Windows 3.x or Windows'95.

The attempt to investigate the interface of the system was unsuccessful because the system is protected by a password mechanism. However, the combination of the help documents of the software and the study of Graves et al. (1981) gives clues about the system. The first bid for four schedules are entered by using the RBS (Registration Bidding) form. To learn the results of the system, the users log into their UNIX accounts on GSB servers and use the command "results". Similarly, "openclose" shows whether a course is open or close (i.e. the course has available capacity or not) and "checkcourse" shows the current prices for the courses. These two commands are useful in the Drop-Add-Swap (DAS) rounds. During the DAS rounds, the student bids for individual courses instead of schedules and the software has also a page for entering DAS bids.

Course Bidding and Registration System of The J.L. Kellogg Graduate School of Management at NorthWestern University

The J.L. Kellogg Graduate School of Management at NorthWestern University has also a number of Web-based applications. One of these systems is of concern for this study: "Course Bidding/Registration System" [NorthWestern University Kellogg Information Systems Department (1997-a)]. As its name suggests, this is a system where the students are bidding for courses and register those courses if successfully placed. The system is accessible with a Web-browser, however it asks for a password. Information about the system is obtained from the well-written help pages of the system.

The system gives 1000 bid points to each student for each quarter. These credits can be used for at most five courses. The students are required to bid for at least three courses. The minimum bid is 1 points and maximum is 977.

The bidding process consists of two rounds of bidding followed by a final check process. In the first round of bidding, the bids with highest points are placed to courses. When the capacity of a course exhausts, the course is said to be closed and unsuccessful bids are placed on the wait list for that course. Each wait list is also sorted from highest to lowest bid points in itself.

On a successful first round bid, students are only charged the minimum successful bid amount. For example, if a student bids 801 points for a course and the lowest successful bid is 201, then 600 points is returned. If a course does not close, only 1 point is charged. Dropping the course or removing the bid from the wait list is possible and the bid amount for that course is returned.

With the bid points that remain from the first round, the students enter the round two of bidding. In this round, the students can bid points for open courses only. No bidding is permitted to closed courses and wait lists. The course lists and wait lists are determined again as in round one. One major difference between this round and the first one is that in the first round first year students are not allowed to bid for second year courses while in the second round all students can bid for them. The aim is to give priority to second year students in second year courses.

If a course is closed, the students who are willing to take that course and who have not placed bids can add themselves to the bottom of the wait list of that course at no point cost.

Bids in the wait list are promoted up in the wait list (and finally to the course list) as other students drop the course or cancel their bids in the wait list.

After the second round of bidding, the final check process starts. In this period the students have to drop any courses that they do not want as well as any wait list positions they do not want to maintain. The bids in the wait lists have still chances to be promoted to the course lists. Also, the students can still add themselves to open courses.

The function of the system ends with the final check process. The Add/Drop transactions are carried out manually with the Student Affairs Office.

The first screen of the Kellogg's Web-Based Course Bidding System is the login screen where the students enter their Windows-NT account and password. When the password is authenticated, the menu screen is displayed. On this screen, there are four choices for a next screen. The screen displayed if the "Quick Entry" choice is selected is a page that is used by the students who know the exact information about the courses, sections and the bid amounts. The "Courses, Bids, WL" choice on the menu page of the system takes the student to the screen where already placed bids, their results, bid points remaining and the courses available for bidding are displayed. To place a bid, from the list of courses, the desired course should be highlighted and the "Place Bid" option should be selected. On the screen that appears, the bid points should be entered and the "Submit Bid" button should be pressed. To change or cancel a bid, "Revise Bid" should be selected on the "Courses, Bids, WL" page. On the page that follows, the student can change the bid amount and press "Submit Bid" or simply press "Cancel Bid" or "Delete Bid" buttons according to his desired action. The other two choices of the menu screen allow the student to see statistics about sections in previous rounds of bidding and to quit the system.

Kellogg Information System Department of NorthWestern University has another Web-based auction system developed in cooperation with Kellogg Career Management Center [NorthWestern University Kellogg Information Systems Department (1997-b)]. The system aims at arranging meetings for

students with recruiting companies. Every company that comes to the campus for recruitment has a predefined number of slots (number of students to be interviewed). Each student is given 800 bid points per academic year. They can spend these points to bid for their preferred companies in order to get an interview with those companies.

Apart from the academic applications, many commercial applications are also available on the Web. One such company is Auction Sales Inc. located at Calabasas, California [Auction Sales Inc., (1997)]. The company collects sale offers of sellers of various products from a long list of product categories ranging from computer hardware to fashioncare. Each product has a page that describes it. On that page, there is also the minimum bid amount of the product (opening price) and its sales price (the actual price). The users can bid for the products that they want to purchase. At specified intervals announced by the company, bids are evaluated and those with highest price offers can buy the products at specified amounts. The users can also buy the products, without joining the bidding process, at the sales price. To use the system, the user should identify himself, send his regular and electronic mail address and his credit card information to the company either over the Internet or by fax.

CHAPTER 3

PREVIOUS ELECTIVE COURSE PRE-REGISTRATION SYSTEMS

In the Department of Management, the method for preregistering elective courses was completely manual. The first section of this chapter explains this system and discusses its deficiencies briefly. The section that follows describes and discusses the Elective Course Registration System, proposed by Sezer (1996).

3.1 - The Manual Registration System

The parties involved in the elective registration system are the students, the advisors, the instructors, the registrars and The Department. The students need to register elective courses. The advisors help the students to register the right number of courses, and to conform the regulations of The University. The registrars register the students to the elective courses and keeps records of the students enrolled and ensures that the class limits are not exceeded. All of these activities are carried out under the control and supervision of The Department.

The registrations at METU are usually held on three consecutive days. However, here are a number of activities that should be performed before the registrations. The advisors should determine what courses each student can register and how many of them will be elective courses. The instructors submit their course proposals, which include the course description, syllabus, prerequisites and course capacity, to the Department. The Department organizes all of these activities, announces the courses offered and the registration procedure. On the first day of registrations, in order to be registered to an elective course, first thing that a student has to do is to go to his advisor and learn the number of elective courses that he is eligible to take. According to that number, the advisor signs empty Electives Cards written to the name of student (Fig. A.1 in Appendix A).

After getting the Elective Card, the student selects the elective courses that he wants to register. Then the student should get these cards signed by the registrar. The registration of all elective courses starts at a predetermined time and at the places announced by The Department. The registrar of each course waits the students at the announced meeting rooms. Therefore, the student has to go to the meeting rooms of his preferred courses. When the student comes, if there is still available capacity for him, the registrar signs the Elective Card. After getting the signature of the registrar, the student goes back to the advisor. The advisor checks the signatures for the elective courses and gives the list of additional courses that the student has to register, if there are any. If the courses are accepted by the advisor, then the student can officially register the courses and get the Registration Form (see Fig A.2 in Appendix A). Finally, the student brings the Registration Form to the advisor who will check whether the registration information displayed on the online registration system is correct or not and electronically approve the Registration. This completes the registration procedure.

Though it seems easy, practically many problems arise during this process. Each registrar carries out the registration at a different room in the building. However, there is the fact that a student usually needs to register more than one elective courses while all registrations start at the same time. To overcome this problem, students delegate each other (in other words their friends)in the queues, an action that some registrars do not accept, while some do. Another method is creating a list. The students do not have to wait all the time, they simply have to be ready when the registration starts.

The registrars, on the other hand, have to deal with dozens of students all trying the get their cards signed before the others. At the same time the registrar has also the responsibility of keeping record of the registered students.

Some students do not even try to go to the queues and directly go to the instructors and they use their abilities in bargaining and they negotiate with instructors. As a result, the instructors allow the student to register the course without really going through the steps described above (especially waiting in the queue).

Dealing with the students makes it impossible for the registrars to have the list of enrolled students updated correctly. At the end of the registrations, the list that the instructor has is usually much different than the one at The Registrar's Office. The instructor has to wait until the lists are sent by The Registrar's Office.

In this system preferences of the students were not fully reflected in the registered courses. Although the student joined the queues of his preferred courses, that he takes all of these courses depended on his position in the queue.

3.2 - The Elective Course Registration Assistant

Sezer (1996) has suggested a software, Elective Course Registration Assistant (ECRA), that has been implemented during the registrations of the Fall Semester of 1996-1997 Academic Year.

To reflect the preferences of students into the system, the Elective Course Registration System gives a fictitious credit amount to students. This amount has been determined by allocating 100 credits per course. Therefore, every student starts with a credit balance of 900 credits, because each student should take and pass nine elective courses in order to graduate.

The student is allowed to spend this credit amount freely on his will. The basic idea is that the student will bid larger amount for a course that he has a higher willingness to register. It is also possible that the student spends few credits during a semester and transfers credits to another semester for bidding more aggressively.

The preferred courses and bidden amounts of the students are collected by a module of the Elective Course Registration Assistant and are stored in a database. While bidding, the students can also use AND and XOR logical operators to select courses. If a student wants to register a course if and only if he is also allowed to register another course, then the student has to use the AND operator in his bid. Or, if a student is indifferent between two courses and would like to register only one of them (perhaps because of a timetable conflict), the student can place an XOR bid.

When the period for bidding ends, the allocation procedure is executed. The bid amounts for each course are sorted and according to the capacity of the course the students at the top of the list are allowed to register the course. The output of the execution is the course lists. Finally, the student actually registers the courses that he has been accepted and it is only the advisor's responsibility to ensure that the student registers correct courses. In other words, ECRA did nothing about checking that the courses that the students registers are the same as those he has preregistered. The system was intended to be used anytime before the registrations, it was not a tool to be used over the period of registrations.

After the registrations of Fall semester of Academic Year 1996-1997, it has been observed that the Elective Course Registration Assistant proposed by Sezer (1996) is successful in assigning the students to their preferred courses in a fair manner. Students have used the system without major problem. At the end of the semester, the number of objections of the students was low enough to be ignored. Few objections were in fact not related to the system itself, but to other parties involved in the system. Yet, it is obvious that no objection does not necessarily mean a complete system.

One major lacking dimension was that it does nothing about the information flow within the system. Before the registration period begins, the supervisor of the system collects all the information necessary for the operation of the system and

loads them into the system. Then, the students enter their bids into the system by using the computer terminals. Once the bid data of the students is collected, the system makes allocation at a single run. The course lists generated after the execution are printed and posted. The students learn the results from those postings. The course list are generated in two different formats: The first of them is the list of the students placed to courses. It is sorted by the Student ID Numbers and it is posted on the bulletin boards. The other list contains the bids of the students and it is sorted by the bid amounts of the students, so as to enable to see the ranking of bids in the lists.

The Elective Course Registration System (ECRS) does not produce anymore information. All of the remaining activities are carried out manually. However, the users of the systems need information in order to continue. For instance, the advisor needs to know what courses a specific student can register. In manual system, this could have been observed from the Elective Cards signed by registrars, while in ECRS this created a problem.

A single run of the system is far from placing all students to the available courses. It has been found that nearly 80% of all bids were successfully placed to preferred courses. However, the remaining 20% had to be assigned manually to available capacities. Moreover, though they have bidden for, some students did not really want to register the elective courses that they were assigned. All of these later placements modifications were to be resolved manually.

Another problem arouse because of the fact that the software developed for collecting bids and making placements was not running on a multi-user basis, so collection of bids had to be carried out on different databases. At the end, when time came to make the placements, the databases have to be merged.

Although no such cases have been reported, the security of the system was so weak that anybody who wants to change the bids of a student was able to access

the student's record and modify it. As a result it becomes obvious that a password protected system is necessary.

CHAPTER 4

THE ELECTIVE COURSE PRE-REGISTRATION SYSTEM

The system proposed in this study integrates all the entities of the system together and ensures that the necessary information flow takes place between the parties involved over a computer network. The algorithm for the placement of students to courses that was described by Sezer (1996) remains almost unchanged at the heart of the system.

4.1 - The Assignment Algorithm

The algorithm of Elective Course Pre-Registration System (ECPRS) is almost the same as the algorithm of Elective Course Registration Assistant designed by Sezer (1996). It tries to allocate the available seats to willing students in a fair manner. The willingness of students are demonstrated by their bids for the courses. Fairness is guaranteed by taking the bid amounts into consideration.

Before discussing the algorithm, it is important to explain the alternative states of the bids. A bid that is processed in the algorithm can either be *successful* (i.e. the student has been placed to the course list), or *unsuccessful* (i.e. the student cannot be placed to the course list), or *unevaluated* (i.e. the algorithm has not yet decided whether the bid is successful or not).

The algorithm starts by creating the list of bids of all students. The list is created by reading the student files.

One major aspect of the algorithm is that it can also handle couple bids. A couple consists of two courses and bid amounts joined by an AND or XOR operator. In this case the student is a candidate for both courses. Therefore, while creating the course lists, the bid of the student should appear in the list of

both courses. To accomplish this, the couple of each bid is added to the list of bids so that the bid appears in the list of both courses. The basis of the algorithm is sorting. First, the list of bids is sorted by the course codes. Then, within each course, the bids are sorted by the bid amounts. As a result, course lists contain bids sorted from highest to lowest bid amounts.

Once the bids are sorted within each course and capacity pointers are inserted, all of the bids are evaluated to see if they are successful or not. To keep track of the course capacities, the algorithm uses pointers. A pointer is actually a dummy bid inserted into the course list. The bids which are above the pointer are within the course capacity. But, being above the pointer is not enough for the couple bids, as the result for the bid depends on the result of its couple. Whether the bids are successful or not is determined by the evaluation procedure.

The evaluation procedure evaluates each bid one by one. If the bid is a simple one (i.e. contains no logical operators like AND or XOR), the decision straightforward: The bids above the capacity pointer are successful. In the subsequent iterations of the evaluation procedure, when some bids above the capacity pointer are declared unsuccessful, the pointer will be moved downwards in the course list to include the waiting candidates. This means that the bids that are below the capacity pointer for the time being may move to above during the future passes through the list. For this reason, the simple bids below the capacity pointer are left unevaluated until the next pass.

The bids with an AND operator are also simple to evaluate: If the bid is above the pointer, the couple course is checked. If the couple bid is also above the pointer, then both bids are successful. But if the couple bid is below the pointer or if the bid under consideration is below the pointer, then the bids are left as unevaluated until the next pass.

In the case of XOR'ed bids, if the bid and its couple are above the capacity pointers, both courses can be successful. However, XOR operator requires that

only one of the bids is successful. The tie is resolved by giving priority to the bid with higher amount of bid credits. If that rule fails, closeness to the top in the course list is the secondary tie-breaker. If the bid and its couple is below the capacity pointers, they are left unevaluated. If anyone of the bids (the bid or its couple) is above the pointer while the other one is below, then the one above the pointer is successful and the one below is unsuccessful.

By using the evaluation rules defined in the previous paragraphs, the algorithm evaluates all bids in all courses. After each pass through the lists, the capacity pointers of courses should be adjusted, because some of the bids above the pointer may be declared as unsuccessful and hence the bids below the pointer have to be included into the capacity. As the pointers are modified, new bids are moved above the pointers and evaluation of these bids becomes necessary with these new capacity figures. The iteration of the procedures for evaluating the bids and moving the pointers continues until no more capacity pointer movements can be done. At the end of these iterations, the course lists are checked to see if the number of successful bids above the pointer of each course equals its capacity. If a course capacity is exhausted, the bids waiting for reevaluation should become unsuccessful. When a bid is declared unsuccessful the result of its couple bid (if any) can also be determined. However, this operation changes the lists of the courses containing the couple bids. So a new "movement of pointer-evaluation of bids" iteration starts. As explained in the previous paragraph, this iteration ends when no bids more capacity pointer movements is possible. This meant that during the last evaluation process no more could be declared as successful. However, the algorithm makes another check: Did the last run of the evaluation procedure change the status of any bids? If the answer to this question is yes, then the algorithm resumes from the evaluation of bids once again, continues with adjusting the pointers and checking the course capacities. Finally, it comes back to the same question. If the last evaluation could not change the status of any bids, then there is no more bids left to process and the algorithm stops.

The tests of Sezer (1996) have revealed that the algorithm was efficient in doing the allocation, with the given size of the problem of allocating students to courses. Out of 625 bids collected from the students during the Fall Semester registrations of the Academic Year 1996-1997, only 88 was unsuccessful. The details of the algorithm is presented in Appendix D.

4.2 - Major Activities in The System

- Initiation: Before the system starts to run, it requires that some information about the students and courses are provided. These are the name, ID Number, year of study, and the number of elective courses that the advisor allows to register for each student, and the name, prerequisites and capacities of elective courses prepared by the advisors and course information provided by the instructors.

- Bidding: This activity is carried out during a period called the Bidding Period and it consists of collecting the bids of the students into the system. The students are allowed to access the system during this period. They place the bids for their preferred courses, and modify them. At the end of this period, the bids are automatically merged together, and made ready to be processed by the system in the Execution activity.

- Execution: The assignment of students to their preferred courses is done by this activity. The system takes the bids into consideration and places the students to the courses according to the procedure defined in Section 5.1. At the end of this activity, the course lists are generated and the student records are updated.

- Rejection: When the results of the execution are announced, it is possible that a student does not want to register a course that he has been placed. Additionally, the student can placed to more courses than he is allowed to register. In this case, the student has to reject the course. This activity can be

done during the Reject period. The capacities freed in this period are allocated to the students waiting in the reserve list of the course and the results are posted at the end of this period.

- Manual Registration: Although the primary aim of having an auction method for assigning students to courses is to eliminate the manual activities, thus making the system more stable and less difficult to handle, there is still a need to have a manual period. If a student is unable to find enough number of elective courses at the end of Executions, or if he is still unhappy with the courses that he has been placed, then with the help of the advisor, the student can be added to or deleted from the list of an elective course in the Manual Registrations period.

- Registrations for Non-Departmental Students: The courses offered by The Department of Management are very popular among the students from other departments. Handling this demand is the hardest task during the registrations. This activity is intended to assign Non-Management students to free capacities (if there are any) manually after all Management students find their courses.

- Deductions: This activity involves the deduction of the bid amounts by the supervisor of the system from the students' credit balances according to their expenditures. The rules for deductions are detailed out in the Deductions section.

4.3 - Time Based Description of The System

Registrations in METU are performed in two periods. The first one is Registrations Period and the other is the Add-Drops Period. The activities described above are implemented during these periods. Appendix B contains the flowchart which shows the flow the activities graphically. The timing for these activities depend on the choice of the supervisor of the system. Appendix C

contains a sample for timing which has been used for the registrations of the Spring Semester of 1996-1997 Academic Year. The application of the above activities to Elective Course Pre-Registration System to are described in the following sections.

4.3.1 - Registrations Period

In the Registrations Period, the students make their decisions about which courses to take under the light of the advisors' advises and register those courses. The registrations are approved by the advisors to ensure that the registrations of the students are in compliance with the rules and regulations of the university.

The Registrations Period in METU continues for three days. Therefore, the system is designed to be active for three days. During these days, the activities listed in the previous section are undertaken in a sequence with the order as they appear. However, the first three activities can be repeated, so as to maximize the students' chances of being placed to their preferred courses through the system during different Bidding-Execution-Reject periods. Manual Registrations and Non-Departmental Registrations are carried out once at the end of the registrations. Timings for these periods are flexible and can be adjusted by the supervisor of the system. It is even possible to omit some of these periods, for example the Reject period.

Bidding Periods

The tasks common to all Bidding Periods is simple: The student enters into the system and make his bids. The system stores the bids of the student in his record file. During the Bidding Period, the student has the chance to modify his bids, change the bidden courses, the bid amounts, and the operators –if he used any. Only the last state of the records will be taken into consideration in the Execution. During the Bidding Period the system is accessible to all users.

The first of the Bidding Periods starts on the first day of the registrations. There are few specific tasks that should be carried out in the first Bidding Period:

- The student should read the course descriptions, outlines, etc., and decide on a list of courses that he prefers.
- The student must go to his advisor to learn the password that he will use during the registrations.
- With this password, the student can access the system and see the number of electives that his advisor allows him to register.
- In case of objections to number of allowed elective courses, the student goes back to the advisor. If the advisor accepts the objection of the student, he can change this number.

Execution

At the time of the Execution, the system merges all bids collected from students and executes the placement procedures. When this activity is being carried out, the system is inaccessible by the users, except the supervisor. Once the placement operation is completed, the course lists resulting from that execution will be prepared. Additionally, the courses that a student has right to register are added to the record of the student.

Reject Period

After the Execution the results of the allocation process are announced. The student learns about the courses that he has been placed to. If the courses are exactly what the student wants to register, then he can complete the registration procedure by going to the registration room of the METU Computer Center. However, in two cases the student should reject courses: (i) the student may not be happy with an elective course that he has been placed, so does not want to

register that course; (ii) the number of courses that the student has been placed exceeds the number of courses that he is allowed to register. If any of these conditions takes place, the student has to reject the course that he does not plan to register.

If the reason for rejection is that the number of courses placed is larger than the number of courses that the student is allowed to register, then rejection is mandatory. If the student omits this step, the system will carry out automatic rejection and the course with the smallest bid amount will be rejected.

The purpose of the reject period is to free up the capacity that is consumed by a student who will not register that course. At the end of Reject period, if the capacity figure of any course has been changed then the bids which were unsuccessful during the execution are reevaluated to see if they can be placed in this period. The results of this process is also announced. The bids which are still unsuccessful will not be taken into consideration in the subsequent executions of the allocation procedure. The capacities that could not be filled up with the bids in the reserve list will be available during the following Bidding or Manual Registrations period.

Rejection is essential for the fairness of the system, because if the student is placed to a course and does not want to register this course, then he should not use up the space reserved for him. Instead, this place should be available to another student who still wants to take this course.

The system is accessible by all user during this period.

Subsequent Bidding-Execution-Reject Periods

A student that has not been placed to an enough number of elective courses at the First Execution, can try his chance by participating in the subsequent Bidding-Execution-Reject Periods.

During the subsequent Bidding-Execution-Reject periods, the procedures and rules are the same as explained above. It is expected that the number of bids placed in the subsequent periods will be much less than the first period.

One important rule of the system is that once the student has been placed to as many courses as the number of courses that he can register, he cannot bid anymore. This ensures that the student does not take a number of courses for granted and then chases for other ones. For providing equal opportunity, all students should try to find only the necessary number of courses.

But if the student really wants to change a course, then has to reject the course that he does not want to take. This allows the student to bid for another course that he wants to register. A course can rejected only during the reject periods. After rejecting, the student can participate in the subsequent bidding periods for a new course.

Approval of The Advisor

The elective course preregistration procedure is completed for the student who has been placed to his preferred courses, and he has rejected enough number of courses, if necessary. In this case, he can go to the Interactive Registrations Room of the METU Computer Center to enter all of the elective and compulsory courses that he is registering that semester. Then, the last step is going to the advisor and to get his approval for the registration of that semester.

According to the rules of the university, the student cannot override the prerequisite conditions and register courses that conflict in the timetable of classes and final exams. The advisor is required to check for the validity of such rules only at this point. This means that the student should not try to register courses that the advisor will not approve later on. If he does, the advisor will ask the student to go back and find a new course and this can be considered as a

penalty for such an attempt. Additionally, the student may not be able to find an elective course suitable for him after that time.

While approving the registration, besides performing other necessary actions not related to the system in consideration, the advisor accesses the elective course preregistration system and makes sure that the student has really been placed to the elective courses that the student has registered in his online registration. If this information is correct, the advisor approves the registration. The advisor also locks the record of the student. The reason for locking the record is that when the student finishes his registration, he can go back to the system and reject the preregistered courses, so that he frees up the space for making it available to his friends.

It is also possible that the student wants to change one or more elective courses that he has registered, though the Registration is officially approved by the advisor and the record of the student is locked. However, as his record is locked, the student cannot access the system to reject the course that he wants to drop, and bid for a new course. When such a case occurs, he should immediately go to his advisor, where he will get his account unlocked. After that, he can continue to use the preregistration system. An important issue is that the advisor cancels his approval for the registration of the student, because the courses will change.

Manual Registrations Period

After all Executions, the student may still be unable to find an elective course for himself. Though the system is designed so that few such transactions are necessary, in such a case the advisor of the student will place the student to a course during the Manual Registrations Period. The advisor can see the idle capacities of the courses, and can place the student to one of the available courses, with the bid price necessary for this course, and this course will cost only as much as the minimum bid price (base price) of the course for the student. The advisor will have the right to use this part of the system only during the

Manual Registrations Period. The students cannot access the system anymore starting from this period.

Registrations for Non-Departmental Students

The end of the Manual Registrations period means that everything is over for the Management students. The remaining capacities of the courses are compiled once again, this time for the registrations of the Non-Management students. These activities are carried out exactly the same way as the manual registrations, but on a FCFS basis. A specific registrar is assigned to each course for adding the students to the system. Additionally, the Elective Cards that other departments usually ask are also signed by these registrars. The reason why there is no auction method for non-departmental students is that there is no information about them beforehand and that a specific selection criteria can not be defined for them.

4.3.2 - Add-Drop Period

Usually, two weeks after the Registrations, The University initiates the Add-Drop Period. The purpose of this period is to give the student a last chance to change the registered courses, because there may be timetable modifications resulting in lecture hour conflicts, or after two weeks of lecture, the student may feel that the elective course does not really meet his needs.

Although it is possible to apply a procedure similar to Registrations, the Add-Drop operations are totally carried out by using the Manual Registrations module of the system. It is expected that the number of transactions in the Add-Drop period will always be low enough that the Manual Registration module will meet the need.

However, if the supervisor finds out that it is necessary to apply the regular Bidding-Execution-Reject period, he has to start with a Reject Period, so that the

students first reject the courses that they do not want to register. Then, they can go on with bidding for new courses. In other words, during the Add-Drop period the classical Bidding-Execution-Reject sequence will become Reject-Bidding-Execution sequence.

4.4 - Deductions

When the Registrations and Add-Drops are over, the account balance of each student is adjusted by carrying out the deduction activity. This is the Supervisor who initiates the Deduction activity. The calculation of remaining credits are done on the basis of the rules listed below and the student records are updated accordingly. A chart that summarizes the deduction policies is available in Appendix E.

- Initially, every student has 100 credits allocated for each elective course, therefore, in total every student starts with 900 credits. The student is totally free to use these credits according to his desires. However, he has to be aware that once his credits are exhausted, he cannot register any course, expect the courses with minimum credits (The minimum credits are set to 20 points).

- The amount of credits that the student specified in his bid for a certain course will be deducted from the student's balance if he is successfully registered to that course.

- If a student cannot get the right to register a course at first, but gets the right to do so during the manual phase of the system without having a chance to specify a bid amount for the course, the student will be asked to spend the minimum amount of points (base price) for that course. For example, suppose that the least bid amount for a course is 35 points. Then any student registering that course manually will also spend 35 point for the same course.

The reason for this rule is that the student should spend at least as much credits as the student who is last on the course list.

- During the Manual Registrations period or with the approval of the Instructor, it is possible that the student registers a course although he has bidden for the course and was not placed in the list. In this case, the amount to be deducted is the maximum of the student's initial bid or the base price of the course. For example, suppose that the base price of the course is 80 points and the bid amount of the student registering the course with this method is 50. The deducted amount will be 80 credits.

- If a student gets the right to register a course by bidding for it or by being placed manually, and he does not register the course officially, the bidden credits will be refunded except a constant penalty set by the supervisor, because not registering a course while having the right is hindering another student.

- If the student registers a course and then drops or withdraws that course or fails on that course, his credits will not be refunded.

- Non-departmental electives are 100 credits worth.

4.5 - MBA Students in The System

As they also consume some space in the capacities of elective courses, the MBA students cannot be exempted from the registration system. However, MBA students are not required to bid for courses. As long as there is available capacity in the list of a course, the MBA advisor can add the ID number of the student to that course in the same manner as the method explained in the Manual Registrations Period section above. With such additions, the capacity of the course is also modified, so that the capacity figure can be kept under control.

In the future, if there is a great overload created by the MBA students, they can also be included in the system.

4.6 - Processes in The System

The students, advisors and the supervisor use the system through processes. The processes are sets of tasks that allow the user to accomplish a specific task in the system. In the figures, the data flow diagrams show the processes and their data input and output. Figures 4.1, 4.2, 4.3 show student processes, Figures 4.4, 4.5 show advisor processes and Figures 4.6, 4.7, 4.8 show supervisor processes.

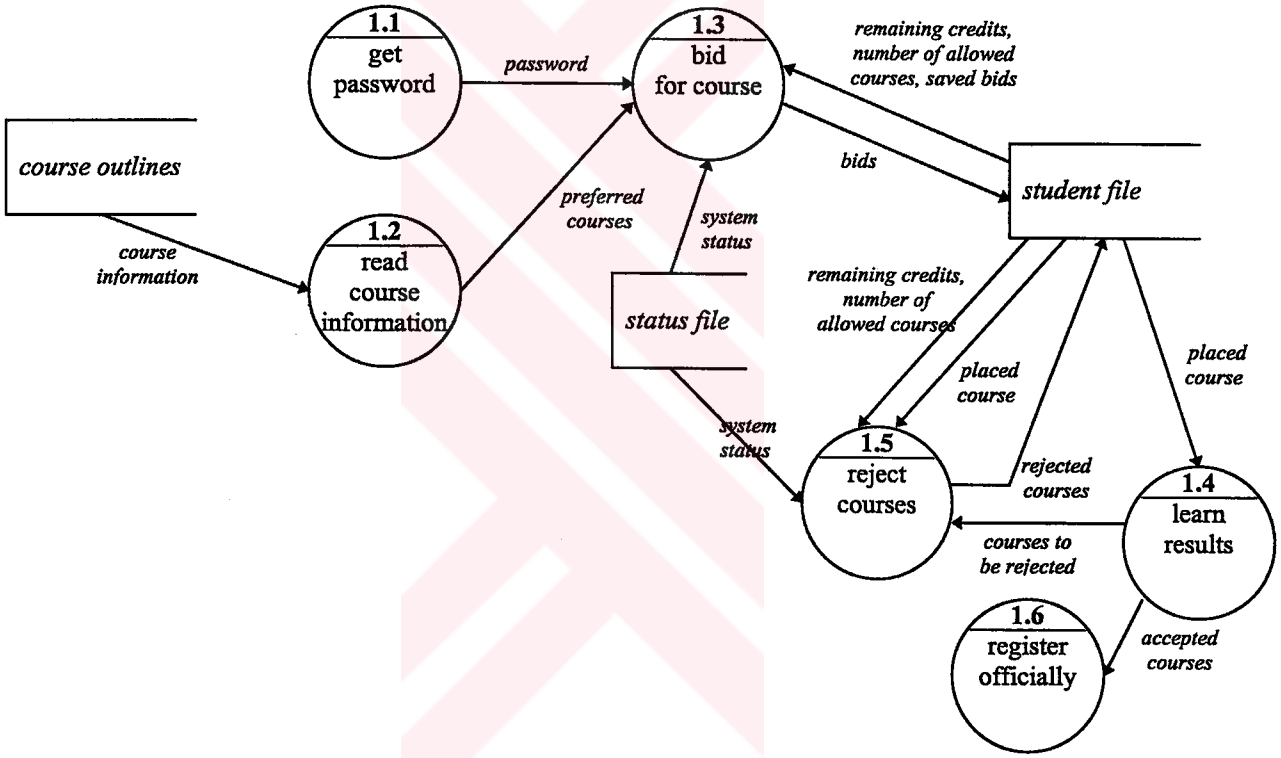


Figure 4.1 - Student Processes

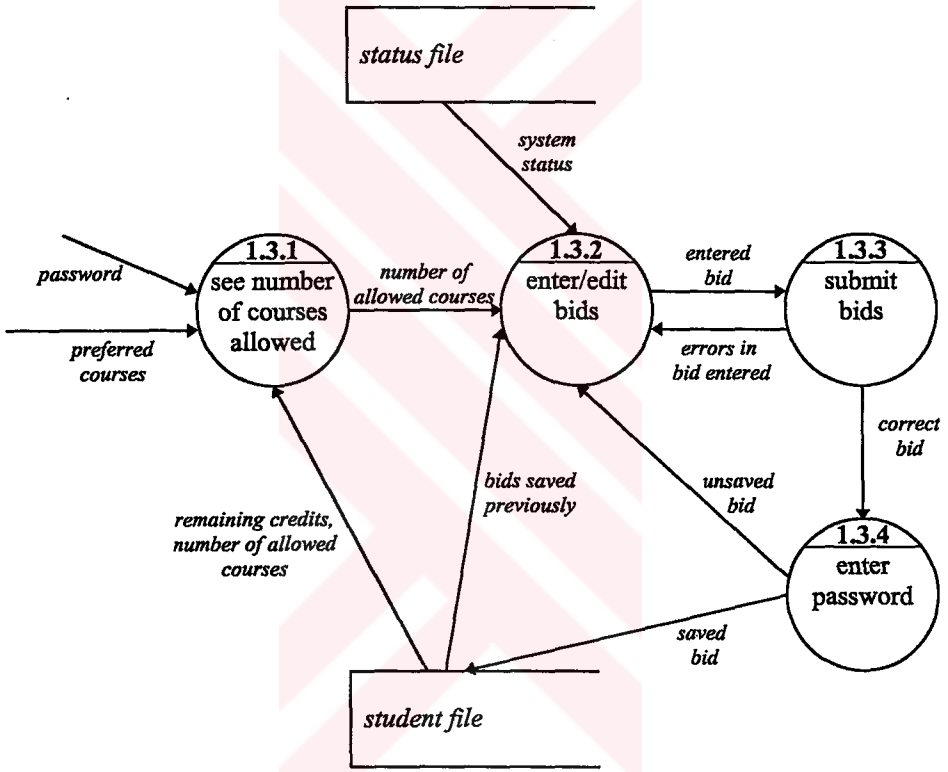


Figure 4.2 - Expansion of Process 1.3 (bid for course)

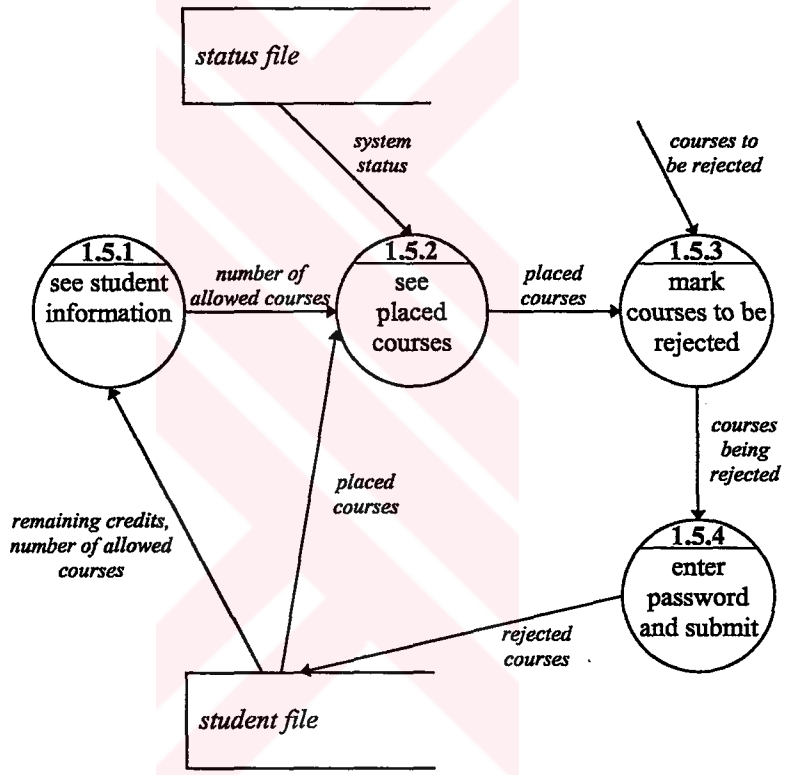


Figure 4.3 - Expansion of Process 1.5 (see placed courses)

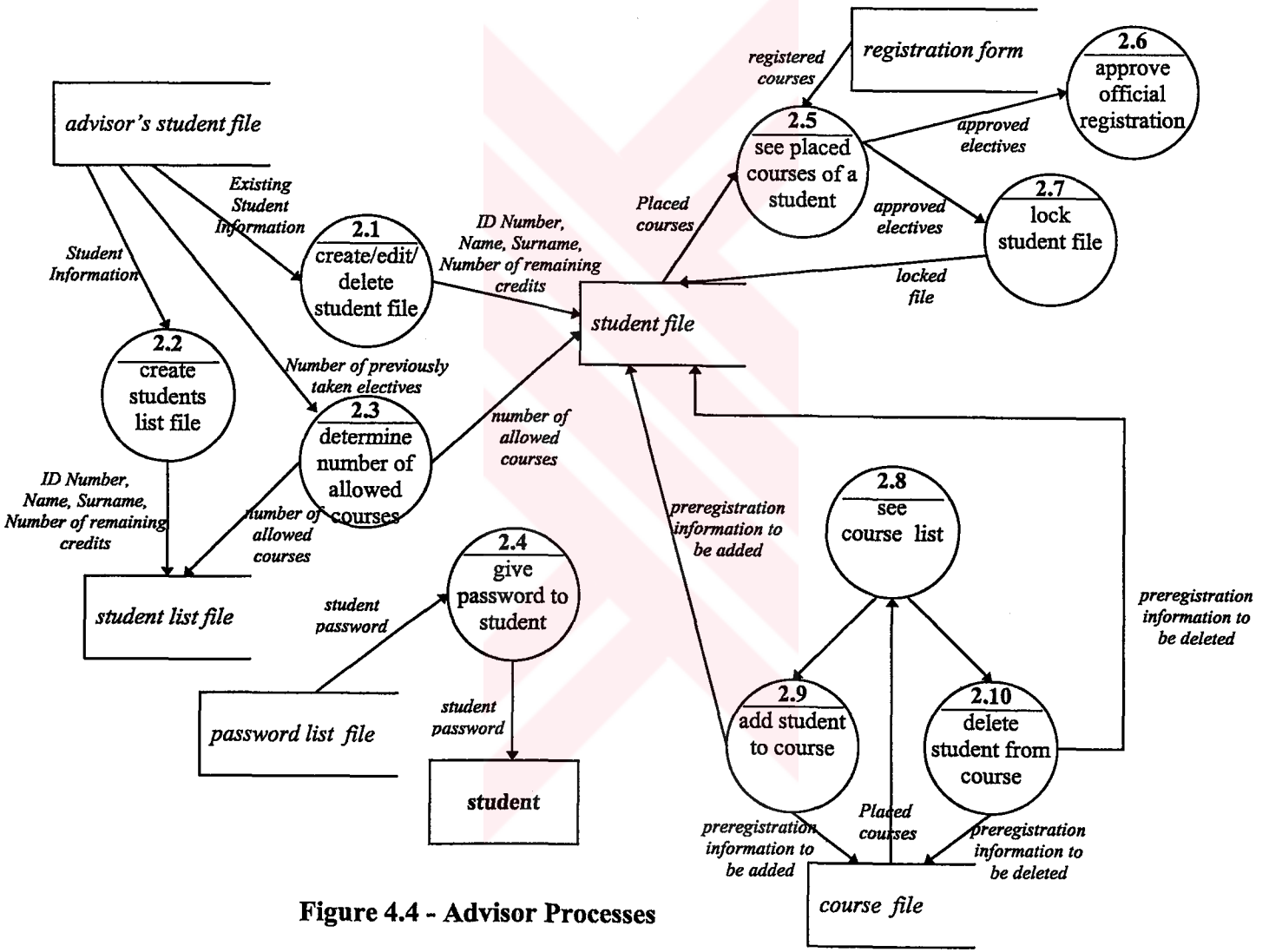


Figure 4.4 - Advisor Processes

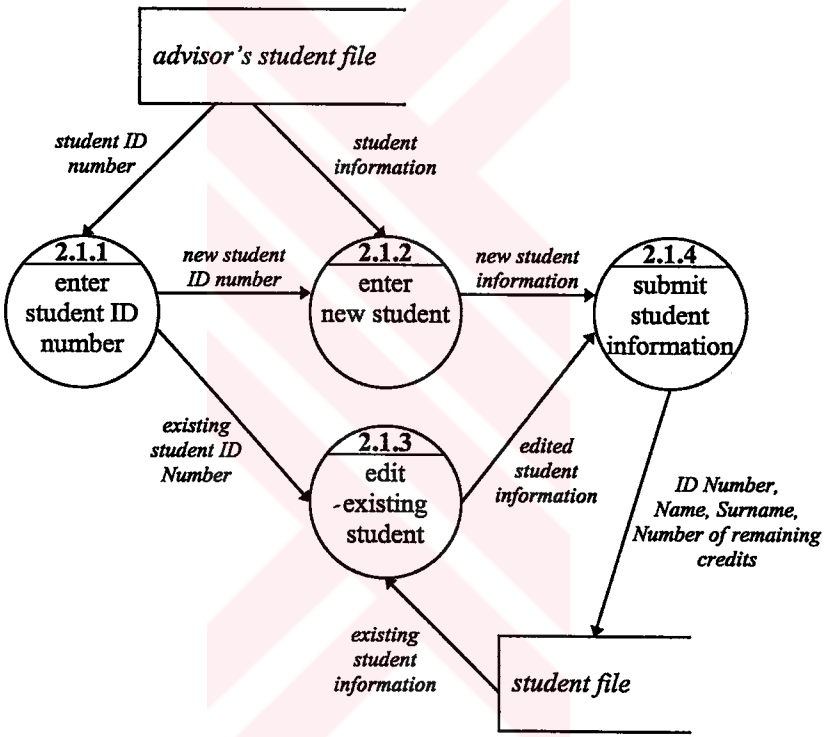


Figure 4.5 - Expansion of Process 2.1 (create/edit/delete student file)

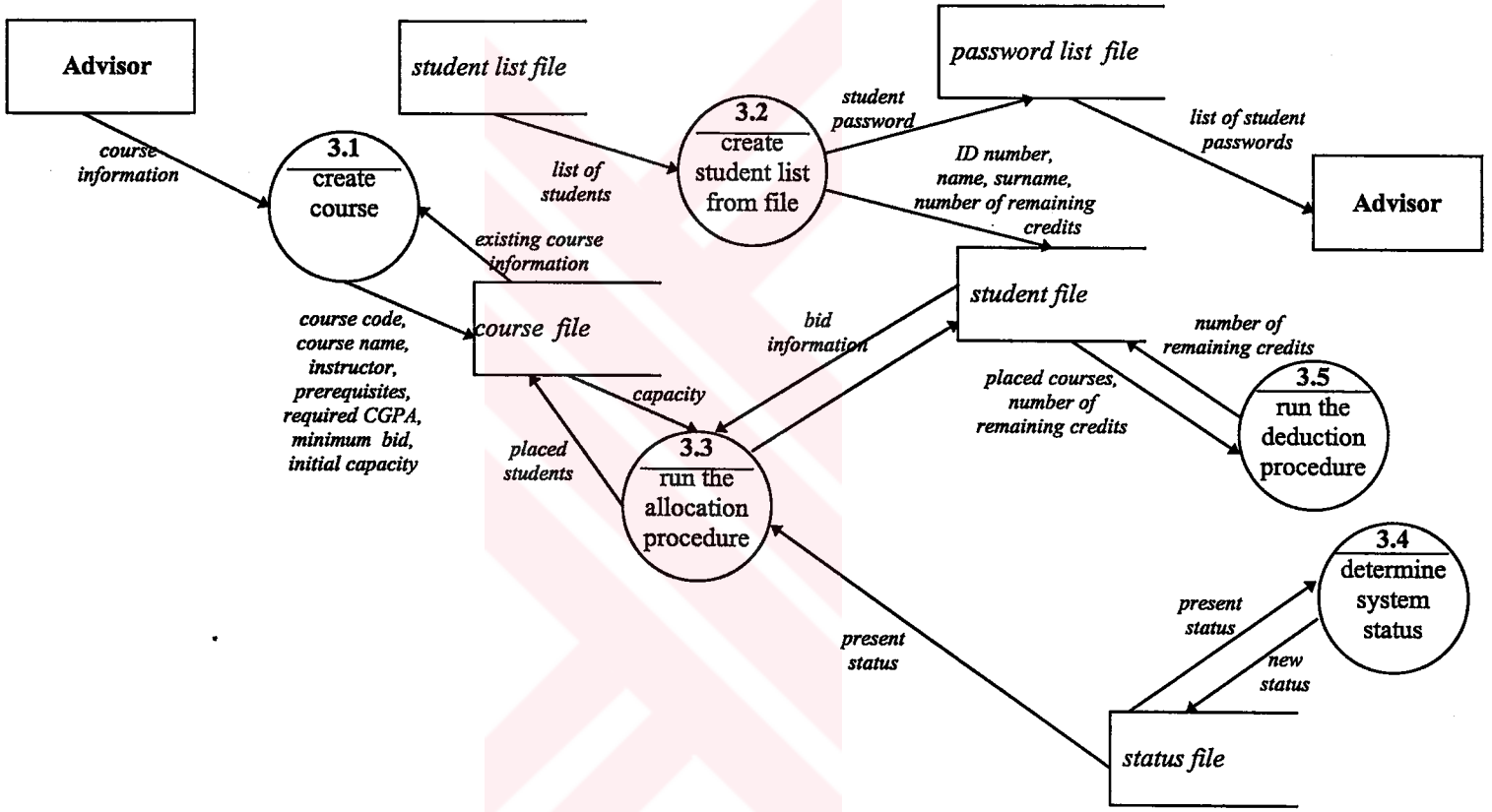


Figure 4.6 - Supervisor Processes

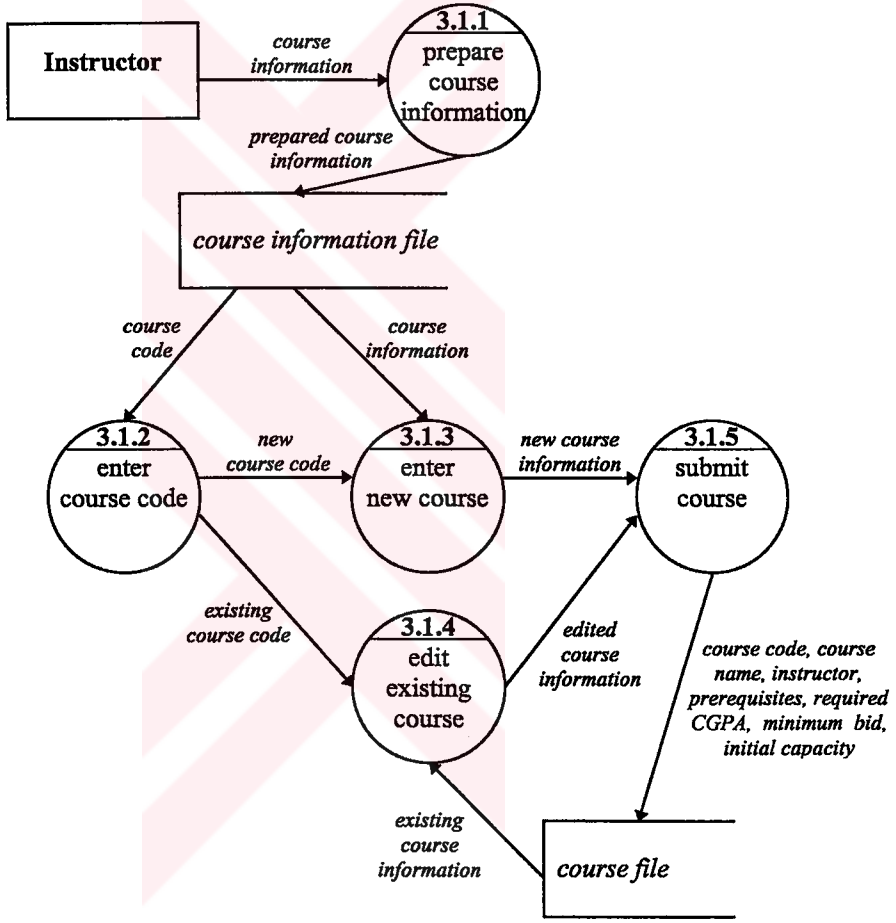


Figure 4.7 - Expansion of Process 3.1 (create course)

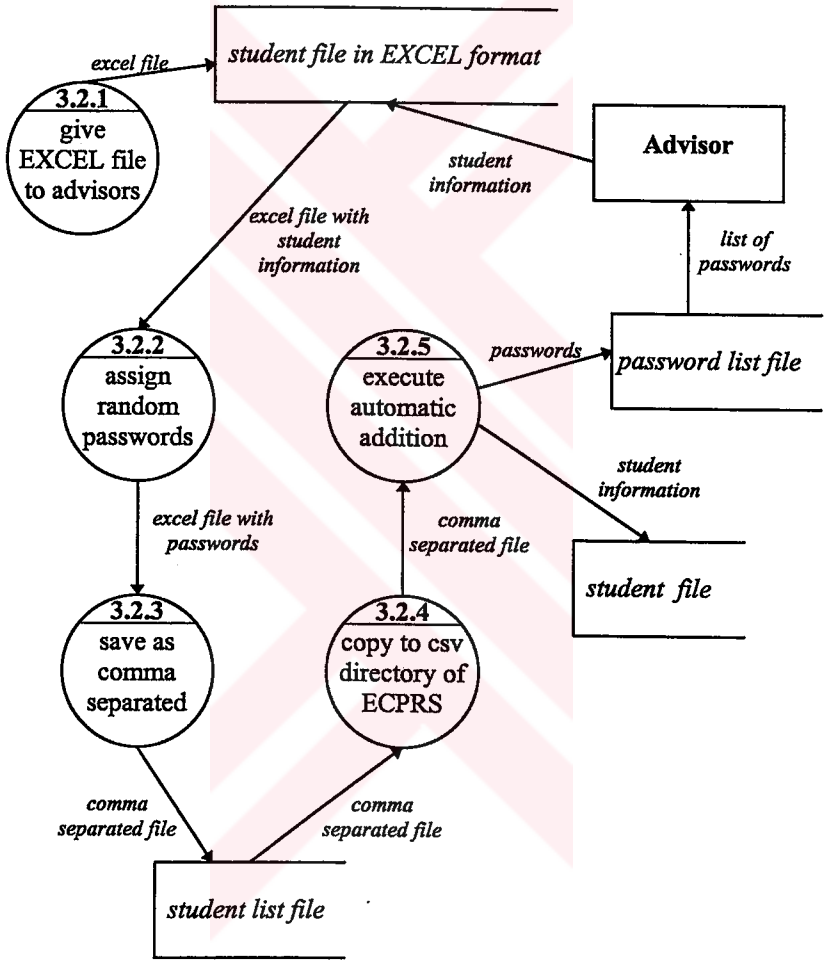


Figure 4.8 - Expansion of Process 3.2 (create student list from file)

CHAPTER 5

IMPLEMENTATION ISSUES

Within the scope of this study, the Elective Course Pre-Registration System that has been analyzed in the previous chapters has also been implemented as a software application. The first concern in the implementation is the selection of software packages and platforms to be used. Then, the format of data stored in the system should be determined. An important issue in the implementation is the security protection of the system. These points will be clarified in this chapter. The software developed has been used during the registrations of the Spring Semester of Academic Year 1996-1997. The evaluation of the results generated by the system are also discussed in this chapter.

The user manual prepared for the implemented system is available in Appendix F.

5.1 - Selection of Software and Platform

There are several ways to implement a system where the parties involved accesses the system simultaneously. A typical method is the use of a large scale database management system, which runs as a server and provides the necessary information upon requests. However, the information load of the elective course preregistration system is not heavy enough to require such software.

Another alternative for the selection of software is the use of a less sophisticated database management application. In this case, there is no server which centralizes the data storage. Instead, the data is distributed over a number of different machines. When it is time to do placement, the data is merged and then it is processed. Although this method has actually been used in Elective Course Registration System Sezer (1996), it cannot be used in the new system. The reason is that in the former system, there was a single bidding period, and editing

the data once the bids are placed was not possible and necessary. On the other hand, the basis of the new system is the continuous storage, modification and processing of data. Therefore, the data should be kept centrally, and any user of the system should be able to browse the data and modify it. The data should be easily accessible from anywhere, as long as the security is ensured.

Considering the need for a centralized database and accessibility, the simplest yet strongest alternative is the development of an interface over the World Wide Web (WWW). The WWW is accessible from everywhere over the METU campus as well as from the outside. Provided that he has a web-browser, the student can even register his elective courses without actually coming to the campus. As the formal registrations of the University can also be done over the WWW, the system developed here will complement it.

WWW applications give the developer a powerful set of tools for designing highly interactive applications with little effort. Yet, the processing power is always high. From the users' point of view, the system is easy to use. In fact, most of the potential users of this system are already accustomed to use a web-browser.

Apart from the interface, the use of WWW for implementing the system gives the chance of storing the data in a single database. The database can actually be kept on the WWW server, and accessed by the users. At this point the selection of the proper database management systems becomes important. There are many database applications which claims to be WWW-compatible, i.e. the stored data can be accessed through a web-browser. Some of these databases are commercial while there are others which are in the public domain. As the scale of the system to be implemented is not extremely large, any database application suits.

It is also possible to implement the system without the use of any database application. Instead, raw text files can be used for storing the data. All of the

information that should be stored in the system can be kept in ASCII files. In fact, this is the preferred alternative for the implementation of the Elective Course Pre-Registration System.

As the system is intended to work on the WWW, there must be a means to access and modify the data files within this environment. The best method is to use of Common Gateway Interface (CGI) programs. A CGI program is a script or a compiled executable that runs on the side of the HTTP server, i.e. on the computer that the HTTP server runs. It can get arguments from the WWW client. Once the user the necessary information in the fields of a form displayed in the web browser, he can make a call to the CGI program. The program takes the arguments and processes them on the machine where the server runs. As the output, the CGI program generates an HTML file, which is then sent back to the client where it is displayed by the browser. During the generation of the HTML file, dynamic information can be incorporated into the document according to the results of the necessary processing.

There are several alternatives for the selection of the programming language to be used to develop the CGI programs, such as C, FORTRAN, shell programming, TCL or many others. Among these programming languages, TCL has been selected, because it is easy to port between different platforms, it is simple but powerful, and that a TCL program is not compiled, it is interpreted. Another important factor is that TCL has an associated toolkit called TK, using which interactive X-Windows based activities can be generated. Therefore, with the combination of TCL and TK, the system can be extended to implement X-Windows based applications, too.

5.2 - Format of The Files

As stated in section 6.1, the data of Elective Course Pre-Registration System is kept in raw text files. The most doubtful aspect of the use of text files is that the

handling of data may not be as efficient as it would be with a database application. The most important data in this system is the bid data. During the execution of the Elective Course Registration System in Fall Semester of 1996-1997 Academic Year, it has been observed that approximately two hundred students created around 650 bids. Even at the worst case, this number is not expected to exceed 1000. This number is small enough to be handled without a sophisticated database application.

A separate file for each course and student will be used for storing the data. Accessing the record of a course or student is simply accessing the its file, which means that random access to the record of a student or course is possible.

On the other hand, not using a database has some its disadvantages. The first disadvantage is that when there is no database, the records of the students and courses are kept in raw files. However, it is possible that more than one user wants to modify a same file at the same time. Database management systems usually resolve this problem internally, so the application programmer does not have to care about it. One solution of this problem in the text file environment is avoiding large data files that holds all of the information. Instead, the files can be designed at such a length that a user uses only the necessary file and nobody else wants to write to that file. This is another reason why the data of each student and each course is kept in a separate file. Only the student himself will want to use his file, so there will be no simultaneous access to student files. For the files that are inevitably accessed simultaneously by many users, a locking mechanism is built, so that only the first user who opens the file can modify it, and the other users have to wait until that user releases the file.

When the execution time comes, the data that the allocation procedure needs is simply the Student ID Number of the student and his current bids. To create this data, all the student files are scanned and necessary information is written to the master bids file. Then, by using this file, the system places the student to their

preferred courses. The output of this process is the course files, a file for a course which contains the lists of students allowed to register that course. Additionally, the files of the students are also updated to contain the list of courses that the student has been placed. At this point, it is obvious that there is redundancy in the way the data is stored. However, if this step is not implemented, then each time a user wants to see the list courses that a student has been placed –a list the advisor will want to see many times–, all of the course files should be scanned for that student. This is certainly much costlier than having some portion of data repeated in the system.

In the system, the names of student files are in form *xxx.stu,* where *xxx* is the ID number of the student. Similarly, the course files are named as *xxx.crs*, where *xxx* represents the course code.

The format of the files are similar to UNIX style configuration files. They contain *name=value* pairs, the *name* is the name of the information variable. The *value* for that variable is recorded after the = character, and continues until the end of the line.

The student files contain the following information variables:

- Student ID Number
- Name of the student
- Year of education
- Cumulative GPA
- Name of the advisor
- The remaining amount of elective courses
- The remaining amount of credits

- The number of elective courses that the student can register that semester
- Bids of the student
- Placed courses of the student

In the file, bids of the student and placed courses of the student lines may appear more than once. The bid lines consist of the code of the course bidden for, the bid amount, the time and date of the bid. When the allocation procedure starts to execute, the bid lines of all students are read and this information is the input to the procedure. The system allows the student to cancel a bid. If a bid is canceled, a # character is added to the beginning of the line. Such lines are ignored by the system.

Placed course lines are also similar to the bid lines. However, an integer value is added to the end of the line. This value represents the result of the evaluation. As, only the successfully placed courses are recorded to the student file, this value is always 1 in the student files. In the course files this integer can have other values. Similar to the canceled bid lines, a placed course line can also be preceded by a # character, meaning that student has rejected the courses that he has been placed. The reason for preserving this line is that the system will later deduct a penalty for this course.

The course files contain the following information variables:

- The course code
- Name of the course
- Name of the instructor
- Timetable information
- Prerequisite courses

- Required CGPA
- Initial capacity
- Present number of students placed to the course
- Base bid amount of the course
- Average bid amount of the course
- Placement information lines

The placement information lines are also similar to the placed courses lines of the student files. One additional information is the ID number of the owner of the bid. An integer value is also added to the end of the line and it can have the values 0, 1 and -1. The value is 0 when the bid is done but the execution did not start. After the execution, the value will become either 1 or -1, meaning that the bid is successful or unsuccessful, respectively. If the student rejects the course or is rejected automatically by the system, then the corresponding bid line is removed from the course file, because this information is not necessary in the course file anymore.

5.3 - Security Issues

On the UNIX platforms, an HTTP server has to run with the privileges of a specific user and group, specified in the configuration file of the system. The files are created or modified by the CGI programs should be readable and writable by this user. For this reason a specific user account is created on the system and all of the files related with the system are only readable and writable by this user. This ensures that someone who is able to access the machine where the server runs is still unable to access the data. The data is accessible only if the password of the specific user is provided.

Password Detection

The users of the system can perform the activities that modify the files of the system only if they have an authenticated password. All users should have their own password. Giving the correct password allow a user to access the whole system, because different user groups such as students or advisors have different access privileges.

The passwords for each user is initially generated by the supervisor, and they are output to a file which can be used for printing. The generated passwords are then encrypted using the Data Encryption Standard (DES), the standard encryption method of passwords on UNIX operating systems. What DES does is using the password to encrypt a block of zeros iteratively (usually 25 times). Then, using a two bytes long *grain of salt* string, the encrypted password is again modified to become one of the 4096 possible outputs. The output is an eleven characters long printable ASCII string. This string is stored in a password file. Every time the user enters the password, the same process is repeated and if the resulting string is the same as the one in the password file, then the system enables the user to access.

When a password change is necessary, the user has to contact the supervisor. For the sake of security, the programs necessary for generating and modifying the passwords can only be used by the supervisor.

In the password file there is a separate line for each user. Each line consists of a triplet:

Login name of the user: For the students it is the ID number, including the check digit, and without the - character.

Password: This is the encrypted password of the user, generated by the supervisor.

User Group: There are four groups in the system. The advisors, the instructors, the students, and the supervisors. Each group has privileges to perform a specific set of tasks defined by the system.

5.4 - Evaluation of The System

The system proposed in Chapter 4 has been implemented as explained in the previous sections of this chapter. The software has been used for the elective course registrations during the registrations for the Spring Semester of Academic Year 1996-1997. The results and performance of the system are discussed in this section.

Initial Data

The system has two major sets of initial data. These are the student data and the course data. The number of students that the 3rd and 4th year advisors have reported was 342. According to the regulations of the University, the year of a student is determined by the academic standing of the student, rather than the year of study. However, the assignment of advisors is done on the basis of the year of study at the university. As a result, a student can be a sophomore although his advisor is a seniors advisor. The distribution of the students to years is shown in Table 5.1.

Year	1	2	3	4	Total
Number of Students	**4**	**43**	**136**	**159**	**342**
Total Allowed Courses	**4**	**45**	**166**	**528**	**743**
Average Allowed Courses	**1**	**1.05**	**1.22**	**3.32**	**2.17**

Table 5.1 - Distribution of Allowed Courses to Years

Among these 342 students, 39 was not allowed to register any elective courses. These students are reported to the system only because the advisors are required to submit a full list of their students. Therefore, the real number of students included in the system was 301.

There were 30 elective courses that were offered. The total amount of seats available in all of these courses was 1264.

First Bidding-Execution-Reject Period

During the first bidding period, a total number of 525 bids were collected from the students. 496 of these bids were simple (contained no operators), while 28 of them were XOR'ed and 1 was AND'ed. As the bids with operators are doubled by reversing them, the system actually processed 554 bids. One important fact that has been observed is that the students usually avoided bidding for more courses than they were allowed to register.

After the execution of the allocation procedure, 31 of 554 bids were evaluated as unsuccessful. 3 of these bids were simple, therefore they have definitely failed. The remaining 28 unsuccessful bids were XOR'ed. However further investigation of the placement results has revealed that the couple bids (reverse bids) of all of them were successful. Therefore, out of 525 bids submitted by the students, only 3 were unsuccessful.

This figure shows that the system was successful in placing the students to courses at a rate of 99.46%. However, there were two factors that yielded this high percentage of success. First, 1264 available seats were far more than the 554 bids. The other reason was that the system was misused by some students. The system asks for a password when the student want to submit his bids. If the password is not provided, the system gives an error message to tell the student to go back and enter the password. The students who did not read the instructions on the screens failed to submit their bids. As a result, 34% of the students were

not included in the first execution. If these students could also submit their bids, it could be expected that around 750 bids would be processed by the system. This amount is close to the number of total allowed courses figure in Table 5.1.

After the execution, the rejection period started. The system requires that courses should be rejected if the number of courses that the student has been placed exceeds the number of allowed courses, or if the student is unhappy with a course that he has been placed. The student has to do the rejection on his own, otherwise the system will do automatic rejection. Out of the 32 observed rejections, only 9 had to be done by the system itself.

Second Bidding-Execution-Reject Period

At the beginning of the second bidding period, with the initiative of the Chairman of the Department of Management, the capacities of some courses have been increased, and the total capacity became 1321.

The number of bids collected in this period is 211. None of them contained AND operator while 19 had XOR operators. The result of the execution was 209 successful placements. Two unsuccessful bids were simple bids. Similar to first execution, 19 XOR'ed bids were successfully placed to one of two alternative bids.

During this period, 17 rejections have taken place.

Manual, MBA and Non-Management Registrations

During the rest of the registrations period and the Add-Drops period, 119 manual placement transactions have been done by the advisors. This amount is actually high when compared to the total number of 743 allowed courses. However, no problems have been reported about these transactions, while the integrity of course lists were never lost and placement of students to specific courses was a lot easier than the previous registrations periods.

The number of MBA students added to the system was 90 and that of Non-Management students was 117. Again, these students have easily been added to courses.

Courses Lists At The End of Registrations

Table 5.2 shows the values of key variables as of the end of registrations. It is observed that among the Left Capacity values, the course capacities are exceeded by certain amounts (those expressed as negative values). This results from the fact that some instructors stretched their course capacities during the registrations.

It is observed that the average price of elective courses at the Department of Management comes out to be 67.50 credit points.

The investigation of student records reveal that the initial average of remaining credits (i.e. before registrations started) was 617.70. When the registrations ended this average became 395.33. Therefore, the average expenditure per student was 222.37 credit points.

Course	Initial Capacity	Left Capacity*	Number Placed**	Minimum Bid	Maximum Bid	Average Bid
12410	40	-15	40	20	202	87
12413	50	-3	51	39	201	127
12415	40	40	0	0	0	0
12416	40	12	30	40	190	121
12419	21	5	15	75	250	153
12423	40	26	13	20	153	77
12426	40	28	9	20	102	57
12429	40	38	0	0	0	0
12432	45	29	16	20	21	20
12433	50	23	12	20	51	27
12434	50	-1	54	26	160	78
12438	40	22	11	22	100	54
12443	40	38	1	20	20	20
12445	40	36	4	20	40	25
12448	21	-2	21	25	181	91
12449	40	40	0	0	0	0
12452	40	11	29	23	150	67
12454	32	-3	33	93	201	131
12455	52	6	46	21	134	73
12460	40	31	9	50	150	87
12461	40	17	17	40	125	73
12462	40	40	0	0	0	0
12472	40	1	39	55	165	91
12473	45	-5	50	45	202	129
12481	40	32	5	20	76	33
12482	54	-2	57	20	202	106
12484	40	21	8	20	55	29
12490	40	26	12	61	170	112
12495	86	-5	87	87	204	124
12497	95	19	54	20	208	33
Total	*1321*	*541*	*723*			
Averages	*44.03*	*18.03*	*24.10*	*30.73*	*123.77*	*67.50*

Table 5.2 - Values of Key Variables After All Registrations

* Negative capacities represent the number of students exceeding the capacity. The total of the values in this column is not exactly equal to the total amount at the bottom of the column. The presented total value is computed by considering the negative capacities as zero.

** The Number Placed column shows only undergraduate students placed to the course by bidding.

CHAPTER 6

CONCLUSION

In this thesis, the problem of allocating elective course capacities to the students at the Department of Management has been analyzed. After the analysis, a system has been designed and to resolve the problem. The designed system has then been implemented as a software. The system has been mainly based on the Elective Course Registration System of Sezer (1996), because that system had been used in the Department and therefore the policies had to be the same.

However, the main properties of the Elective Course Pre-Registration System can be summarized as:

- A system that ensures accurate information flow: The flow of information between the parties involved in the system is fast, reliable and as needed.
- A secure system: Access to different modules of the system can be restricted to different groups of users and each user has his own password.
- A Web-based system: The system can be accessed from any computer connected to Internet, without the need of installing a special software.
- A centralized system: The data necessary for the system is stored on a server and is sent to the users only when requested.
- An easily maintainable system: The programming language used for developing the system (TCL) is easy to learn and use. The data structures used in the system are also easy to use.

The results of the registrations of the Spring semester of 1996-1997 Academic Year have also shown the success of the system. In fact, as it can be observed in Figure 5.2, the number of courses where the demand exceeds capacity is only

eight. The results show that the registration procedure for an elective course is in general non-problematic unless the course is very popular among students. On the other hand, the proposed system is successful in providing the necessary information flow, a problem which is irrelevant to the individual courses.

However, future work on this subject may be concentrated on improving the allocation algorithm and comparing different algorithms under various load levels of test data. The introduction of an optimization algorithm into the system may yield more valuable results in terms of quality of allocations.

The idea of using auction based systems to allocate scarce resources can be used in similar problems. The distribution of dormitory rooms to students, and the arrangement of meetings with recruiting companies are two examples explained in the report. For example, the proctors for exams can be given credit amounts that they can use for bidding on their preferred exams hours or days.

The implementation of the system has also shown that porting network based systems to Internet is possible and advisable. In fact, there are lots of systems that are not even computerized and can be ported to Internet. It can be expected that in the near future many organizations will switch to such Web-based systems in order to avoid operating system and software version incompatibilities. There are already many companies whose regular mail address is totally useless, because they are carrying all their activities over the Internet.

Yet, there is one problem that still waits for improvement: The security of these systems. The password mechanisms available as of today are only able to provide a certain level security. On the other hand, experienced crackers can still break into the systems without much difficulty. This factor is one of the reasons why Web-based applications cannot gain the real popularity that is suggested by the conveniences they bring. However, actions are being taken against them, such as secure or encrypted document transmissions. Such subjects should be

studied in detail and added to Web-based systems like Elective Course Pre-Registration System.

A future study can also port the data of the present work to a database management system. This operation can make programming easier and increase the speed of operations in the system. A such improvement can also be the roots of a future system where not only elective course data but all data about the students of the Department can be stored. Such a supersystem can consist of many modules one of which will be this study.

Another challenge for the Elective Course Pre-Registration System is to convert it into a real registration system. In this case, as soon as a student is placed to an elective course the registration information would be added to the databases of the campuswide registration system of the METU Computer Center. This would eliminate many obligatory features on the present system such as locking the student files.

REFERENCES

1. AUCTION SALES INC., "Auction Sales - Live Auctions On Line", *http://www.auction-sales.com/*, 1997

2. BUSAM, V.A., "An Algorithm for Class Scheduling with Section Preference", *Communications of the ACM*, Vol. 10, No.9, pp.567-569 (1967)

3. GRAVES, ROBERT. L. and SCHRAGE, LINUS and SANKARAN, JAYARAM, "An Auction Method for Course Registrations", *Interfaces 23:5*, pp.81-92 (September-October 1993)

4. HYLLAND, AANUD and ZECKHAUSER, RICHARD, "The Efficient Allocation of Individuals to Positions", *Journal of Political Economy*, Vol. 87, No.2, (1979)

5. MACON and WALKER, A., "A Monte Carlo Algorithm for Assigning Students to Classes", *Communications of the ACM*, Vol. 9, No.5, pp.339-340 (1966)

6. NORTH WESTERN UNIVERSITY KELLOGG INFORMATION SYSTEMS DEPARTMENT, "Kellogg Web-Based Course Bidding System", *http://www.kellogg.nwu.edu/script_html/bid/*, (1997)

7. NORTH WESTERN UNIVERSITY KELLOGG INFORMATION SYSTEMS DEPARTMENT, "Kellogg Career Management Center: Student Menu", *http://www.kellogg.nwu.edu/career/student/stu-mnu.htm*, (1997)

8. SEZER, K. BILGIN, "An Auction Method For Elective Course Registrations", Middle East Technical University, Department of Management, MBA Thesis, (1996)

9. UNIVERSITY OF CHICAGO GRADUATE SCHOOL OF BUSINESS, "GSB Online Registration Bidding System FAQ", *http://gsbwww.uchicago.edu/comp_svcs/orbfaq.html*, (1997)

10. WARNER, D. MICHAEL, "Scheduling Nursing Personnel According to Nursing Preference", *Operations Research*, Vol.24, No. 5 (1976)

APPENDIX A

FORMS USED IN MANUAL ELECTIVE COURSE REGISTRATION SYSTEM

(This Card Should be Retained by the Instructor)
REQUEST TO TAKE AN ELECTIVE COURSE
..... SEMESTER ..... METU

Last Name : ..... Faculty : .....
First Name : ..... Dept : .....
Student Number : ..... Year : .....

Course No.: ..... Credit : ..... Group : .....
Course Description .....

Date : Advisor's name and signature Student Signature :

(This Card Should be given to Advisors)
INSTRUCTOR'S PERMISSION FOR REGISTRATION
TO AN ELECTIVE COURSE
..... SEMESTER ..... METU

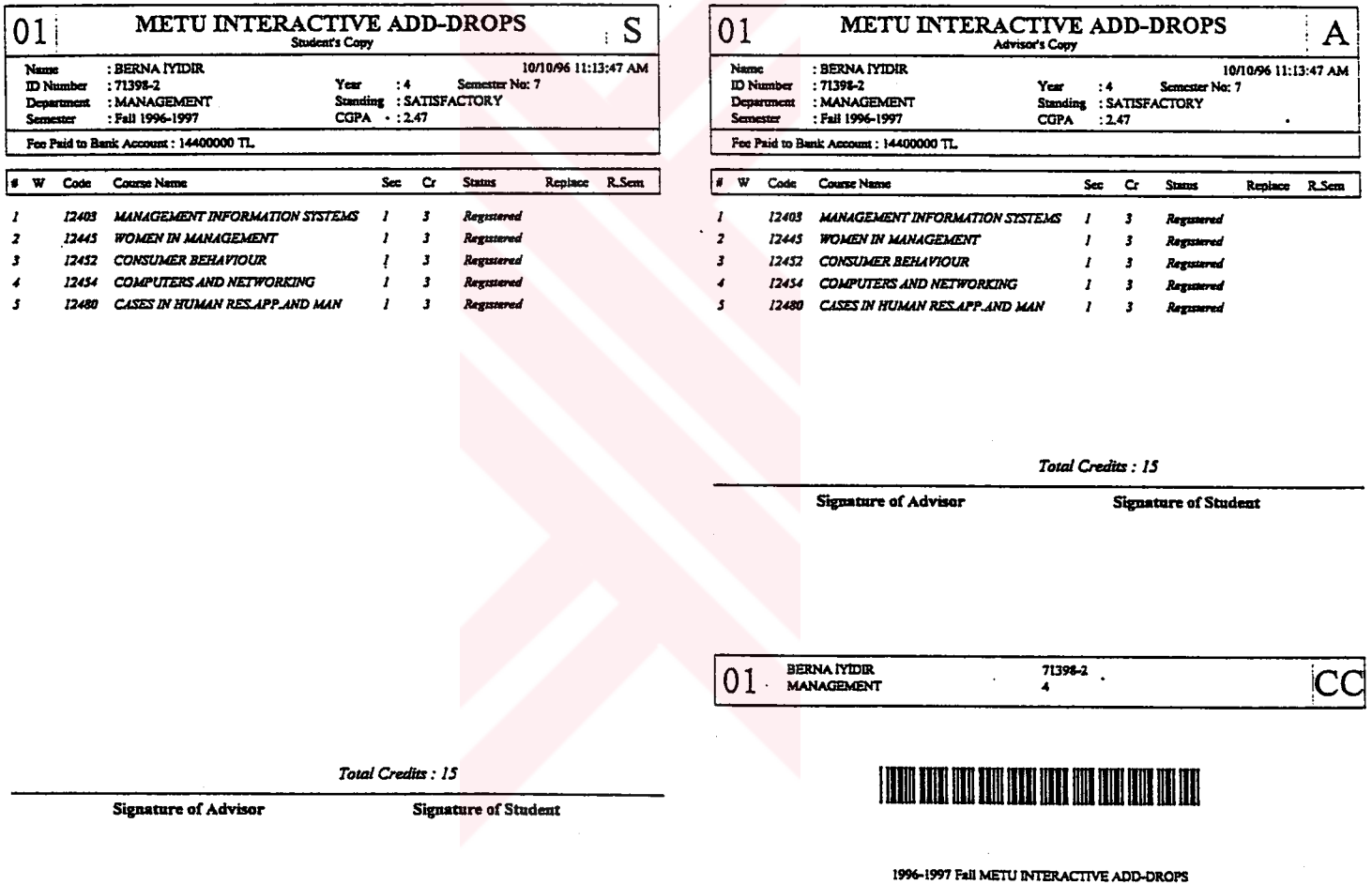
Last Name : ..... Faculty : .....
First Name : ..... Dept : .....
Student Number : ..... Year : .....

Course No.: ..... Credit : ..... Group : .....
Course Description .....

The above named student can be registered to my COURSE
Instructor's Name : ..... Signature

Date : .....

Fig A.1 - Elective Card



01 METU INTERACTIVE ADD-DROPS S
Student's Copy

Name : BERNA IYIDIR 10/10/96 11:13:47 AM
ID Number : 71398-2 Year : 4 Semester No: 7
Department : MANAGEMENT Standing : SATISFACTORY
Semester : Fall 1996-1997 CGPA : 2.47

Fee Paid to Bank Account : 14400000 TL

#	W	Code	Course Name	Sec	Cr	Status	Replace	R.Sem
1		12403	MANAGEMENT INFORMATION SYSTEMS	1	3	Registered		
2		12445	WOMEN IN MANAGEMENT	1	3	Registered		
3		12452	CONSUMER BEHAVIOUR	1	3	Registered		
4		12454	COMPUTERS AND NETWORKING	1	3	Registered		
5		12480	CASES IN HUMAN RES.APP.AND MAN	1	3	Registered		

Total Credits : 15

Signature of Advisor Signature of Student

01 METU INTERACTIVE ADD-DROPS A
Advisor's Copy

Name : BERNA IYIDIR 10/10/96 11:13:47 AM
ID Number : 71398-2 Year : 4 Semester No: 7
Department : MANAGEMENT Standing : SATISFACTORY
Semester : Fall 1996-1997 CGPA : 2.47

Fee Paid to Bank Account : 14400000 TL

#	W	Code	Course Name	Sec	Cr	Status	Replace	R.Sem
1		12403	MANAGEMENT INFORMATION SYSTEMS	1	3	Registered		
2		12445	WOMEN IN MANAGEMENT	1	3	Registered		
3		12452	CONSUMER BEHAVIOUR	1	3	Registered		
4		12454	COMPUTERS AND NETWORKING	1	3	Registered		
5		12480	CASES IN HUMAN RES.APP.AND MAN	1	3	Registered		

Total Credits : 15

Signature of Advisor Signature of Student

01 BERNA IYIDIR 71398-2 CC
MANAGEMENT 4

1996-1997 Fall METU INTERACTIVE ADD-DROPS

Fig. A.2 - Registration Form

APPENDIX B

FLOW OF ACTIVITIES DURING REGISTRATIONS

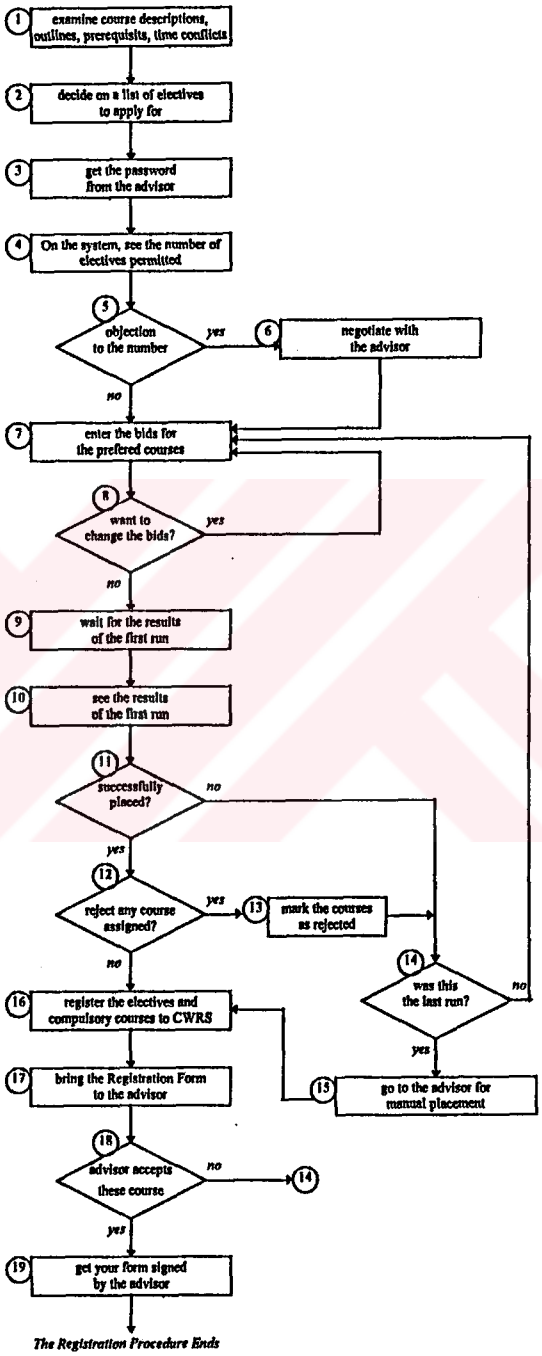


Figure B.1 - Flow Of Activities During Registrations

APPENDIX C

TIMING OF ACTIVITIES DURING REGISTRATIONS

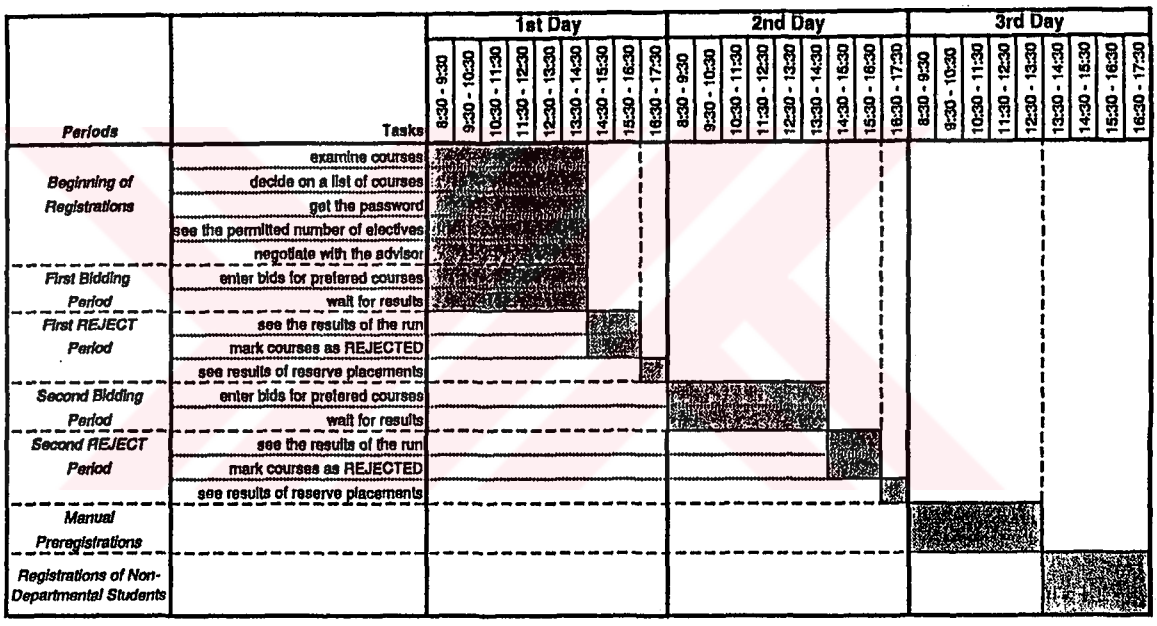


Table C.1 - Timing Of Activities During Registration

APPENDIX D

THE ALGORITHM OF EVALUATION PROCEDURE

I. Read all student files to create a list of bids

II. Add the couples of non-simple bids to the list of bids

III. sort the bids by courses, then sort the bids in each course by the bid amounts

IV. Insert a pointer to each course list so that the number of unevaluated bids above it equals the course capacity

V. For each course, starting from the bid at the top, evaluate each bid. If the bid under consideration is:

- *a simple bid*: and the bid is
  - above the pointer (within the capacity), the bid is successful
  - below the pointer (not in the capacity), the bid is unsuccessful
- *an AND'ed bid*: and the bid is
  - above the pointer: check the place of the couple bid. If it is
    - above its pointer: the bid and its couple is successful.
    - below its pointer: wait for reevaluation
  - below the pointer: wait for reevaluation
- *an XOR'ed bid*: and the bid is
  - above the pointer: check the place of the couple bid. If it is

- above its pointer: both bids can be successful, but the one selected to be successful is the one with a higher bid amount; the other one is unsuccessful

- below its pointer: the bid under consideration is successful and its couple is unsuccessful

- below the pointer: check the place of the couple bid. If it is

  - above its pointer: the bid under consideration is unsuccessful and its couple is successful

  - below its pointer: wait for reevaluation

VI. Adjust the positions of pointers so that the number of successful bids and those waiting for reevaluation equals the capacity of the course

VII. If the position of the pointer of any course has changed, go back to **V** to reevaluating and adjusting **(VI)** again; otherwise continue with **VIII**

VIII. If the last evaluation of all bids was able to able to change the status of at least one bid continue with IX; otherwise stop the algorithm

IX. Simplify the list of courses whose capacities are full (i.e. number of successful bids above the pointer equals the capacity) by declaring the bids that wait for reevaluation as unsuccessful and reevaluating their couple bids. Then continue with **V**.

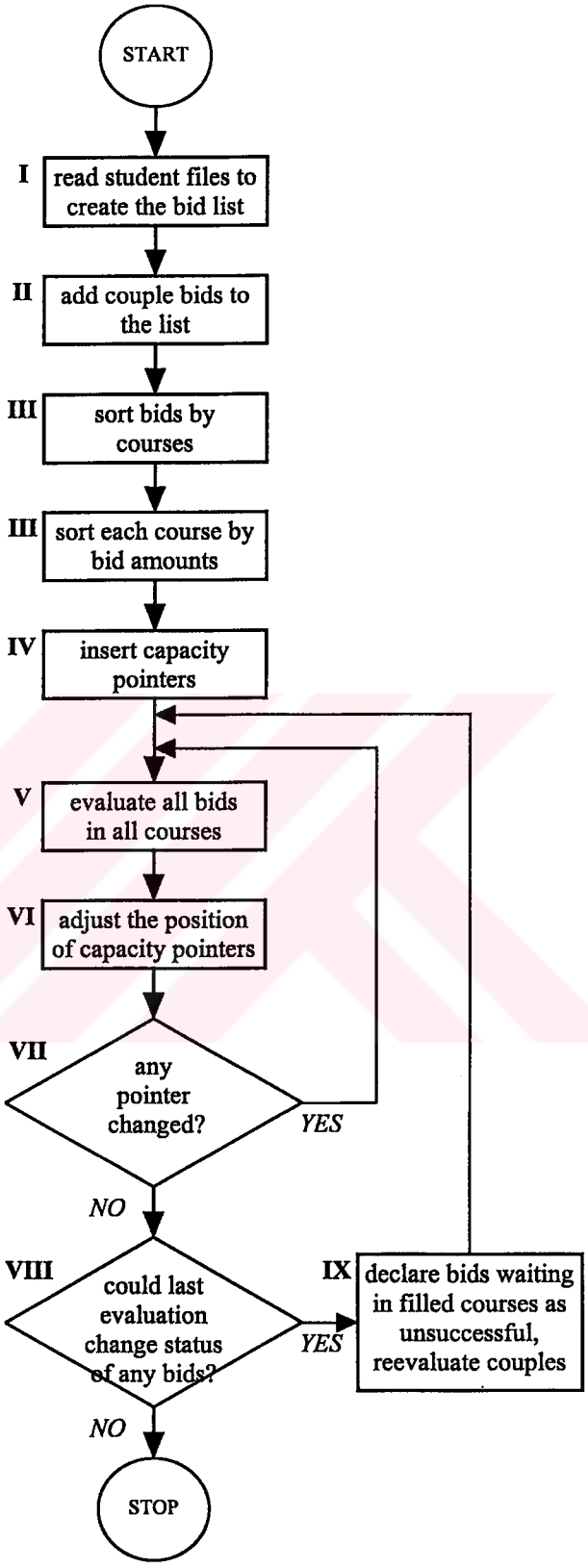


Figure D.1 - Flowchart of the Allocation Algorithm

APPENDIX E

DEDUCTION POLICIES

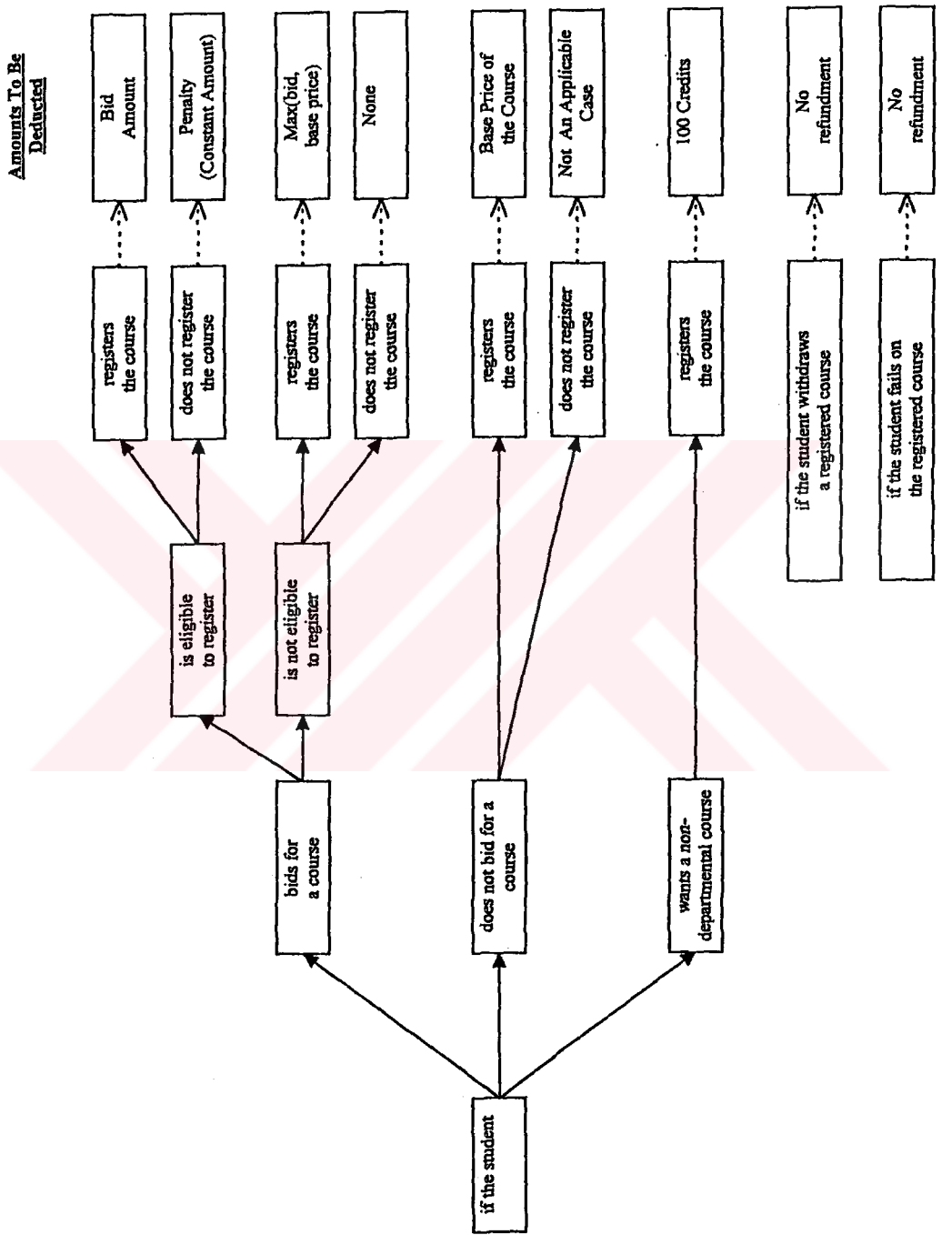


Figure E.1 - Deduction Policies

APPENDIX F

STUDENTS' MANUAL

(FREQUENTLY ASKED QUESTIONS)

Important Notice: *This document is a F.A.Q. which gives answers to specific questions. However,* ***we strongly recommend that you read the whole document*** *at least once in order to get the whole picture of the system.*

As you may observe throughout this document, most of the last minute announcements are posted on the Bulletin Boards of the Department. Pay attention to check the boards as frequently as possible during the registration days.

If you have a question that is not answered in this document, please send an e-mail containing your question to ***www@man.metu.edu.tr****. More questions will make this document richer.*

What is ECPRS?

ECPRS (abbreviation for Elective Course Pre-Registration System) is a WWW based system that aims at placing the students of the Department of Management to elective courses. The system balances the demand for courses with limited capacities. Another feature of ECPRS is that it is a pre-registration system. This means that this system does not actually register the students to the courses. It only grants them the right to register the courses. The students have to use the registration system of METU Computer Center in order to officially register the courses.

What are the advantages of using ECPRS?

First, the former pre-registration system that was a physical race is eliminated (Remember the days you were waiting in queues for getting the elective cards signed). Second, you are be able to express your choices about courses. Hence, you will register your preferred courses. Another advantage is that all of the registration information is kept by a computerized system, making the process faster, safer and more fair.

How do I express my course preferences?

In the past, you were joining the queues of your preferred courses. Now, instead of yourself, the bid points will be queued and they will wait.

What are these "Bid Points"?

Bid Points are fictitious credit amounts that you will assign to your preferred elective courses. In other words, they are the prices that you are willing to pay should you register your courses. The function of bid points is simple: Higher bid points for a course will increase your chances of registering that course, because your bid points will be larger than the bid points of more students.

Do I have any bid points? How many?

Yes, you have your bid points. Initially, every student of the department is given 100 credits per elective course that he/she has to complete in order to graduate. All students, other than a few exceptional students (like transfer students), have to complete 9 elective courses. Therefore, each regular student starts with a bid amount of 900 points available for use until your graduation.

When will I start using the system?

This answer is twofold: You will start using the system the first semester that your advisor allows you to register an elective course (this is the Spring Semester

of third year, if you are a regular student). Or, you will start using the system at the exact day and hour that is announced by the Department.

Am I free to use my bid credits?

Yes. However, there are two sets of constraints: The rules of the game and the strategies. The rules require that your bid for any single course should be within the limits of 20 to 250 bid points. Within this range you can assign any bid points to your preferred courses. But, you should keep in mind that your 900 bid credits are given to you for your whole undergraduate education. If you give 250 points for all your bids, you will certainly register those courses, but it is also certain that your credits will exhaust at your fourth elective course. Therefore, you should adjust your bids to have an average bid points of 100 per course to make the best use of your bid credits.

How do I access ECPRS?

ECPRS is a computerized and WWW-based system. Therefore, you have to use a computer with an Internet access to place your bids (All computers in our labs, or in our campus in general, can be used). The WWW address of the system will be announced to you each semester by the Department. With a web-browser like Netscape Navigator (in fact, we strongly recommend the use of it), you can access that address.

When do I bid?

The exact timing for all activities of the registrations will be announced by the Department. Check the bulletin boards frequently. Speaking generally, bids are collected during the few hours before the execution of the allocation procedure.

Is the time of bidding important?

The main reason for using ECPRS is to eliminate the timing. As long as you respect the time period reserved for bidding, the exact time that you submit your bid is irrelevant. However, this is a network based system so it is always safer to complete your work before the last few minutes of the period.

How do I place my bids for courses?

When you access ECPRS with the address announced by the Department, you will see a page where you have to enter your Student ID Number (this is your access code) and your password. Once you enter these information, you will receive a page that shows information's about you, specifically your name, surname, year, advisor, the number of your remaining elective courses, your bid balance, and the number of elective courses that your advisor allows you to register that semester. At the bottom of that page you see a button called "Place Bids". By clicking on that button, you will get an empty bid form. In that form, you can enter your preferred courses and the bid points that you assign them. Once you enter your bids in the form, for a second verification, you type your password again in the field provided. Then by clicking the submit button, you can have your bids sent to the system. However, if you forgot to enter your password, the system will warn detect it and it will ask for a correction. When your entries are error-free, the system will save your bids, and you will get a screen summarizing your bids.

What is my password? Can I change it? What if I forget it?

On the first day of registrations, you will contact your advisor. He/she will give your password. Do not let anyone learn your password. You may want to change your password especially if someone learns your password or if you forget it. In such cases, you have to go to your advisor. He/she will resolve the problem or forward you to a person who can resolve it. (Note: Do not get

ashamed of forgetting your password, we are human beings and everyone of us can forget anything.)

The information on the opening page of the system is incorrect/incomplete. What should I do?

For getting these information changed contact you advisor. It is expected that most of such arguments will actually be about the number of allowed elective courses. If so, read the answer to the corresponding question.

I don't agree with the number of elective courses that I am allowed to register. What should I do?

This is your advisor who determines that number. Therefore go to your advisor. If your advisor agrees on changing the number, then he/she will do it. Have a consensus with your advisor on this number, because ECPRS will not let you pre-register more courses than the number specified.

Can I bid for as many courses as I want?

The limit for the number of courses that you can bid is seven. But an important rule of the game says that you get right to register (i.e. if you pre-register) a course that you don't officially register in the CC's registration system, then you have to pay a penalty. And remember that ECPRS will not let you to preregister more elective courses than you are allowed to. So, the exceeding courses that you preregistered will be dropped by the system, and you still have pay penalty for them. To make it clear read the questions about penalties, deductions and rejections.

What is a "Penalty"?

The penalty is a certain amount of bid points that will be deducted from your credits balance. Usually, the reason for a penalty is that you hold a place (a seat)

in the list of a course but you don't officially register that course. But, this is hindering another student from preregistering that course.

There are so many fields in the bidding form. Do I have to fill them all?

No. The bid input form is designed for the longest combination of bids. In each row of the form, you will enter a single bid (note that a single row is not necessarily for a single course). If your bids are simple, you only use the first two fields of the row. However for the bids with operators, you fill in the whole set of fields on a row. And the number of individual rows that you will use depends on the number of courses that you prefer. Read the answer for the question **"Can I bid for as many courses as I want?".**

What is the format of a bid?

Each row of the bidding form is for a single bid (note that a single row is not necessarily for a single course). In the first field of the row you enter the optic code of the course. In the second, you enter the bid amount that you are willing to pay in order to get right to register that course. These two fields constitute a **simple bid**. However, you can extend your bid by adding a second course with an operator. By default, the operator field (third field on the row) shows "None", meaning that the line contains a single bid. If you click on that field and pull it down, you will see that your other choices are AND and XOR. Use an AND operator if you want to register the two courses specified if and only if you are allowed to register both of them together. The AND operator is useful in cases where you want to register two courses at the same time. For instance there are two elective courses about marketing, but for some reason, you decide that registering only one of these courses will not master you in that field. In this case you case use the AND operator. Our experience has shown that AND operator is seldom used (it has been used only by two students during two registration periods). The XOR operator, on the other hand, is more favorite. In this case you say that you will register either one of two courses, but not both of

them. The system will place you to only one of the courses. If your bids are high enough to be placed to both of the courses, the system will pick the course for which you gave a higher amount of bid points. This saves you from the penalty cost for one of the courses. After specifying the operator, you enter the course code of the second course (usually called as couple course) and the bid for it.

What are the operators? Should I use them?

Read the answer for the question **"What is the format of a bid?"** for a detailed discussion of the operators.

Can I bid for any course?

Yes and no. First, remember that you can bid only for the elective courses. The courses are open to all students who are allowed to register elective courses by the advisors. However, some courses have specific requirements that you have to qualify. You can learn about them by reading the course outlines. The outlines are posted on the bulletin boards by the Department, or you can access them through the WWW page of the Department. Don't bid for any courses that you don't qualify, because your advisor will not let you register. The advisor will check your courses at the end of the registrations and if any course is not accepted, you may have difficulties in finding a course to replace the refused course. And, don't forget that even if you qualify for a course, the advisor may not accept your registration if there are overlapping courses in your weekly timetable.

Do I have a chance to change the bids that I have placed?

The bids are meaningful only to the next execution of the allocation procedure. So, within the same bidding period that you made your bids, you can also access the system to modify them.

When and how will I learn about the results of a bidding round?

Once the bids are collected from the students, the system will run the allocation procedure. The time of this process will be announced by the Department. The output of the allocation procedure is the course lists. The course lists will also be posted on the bulletin boards. Check all the courses that you have bidden. The results are also available on the WWW, however the number of students using the system is so high that the laboratories may be overcrowded.

I have been placed to my preferred courses. What do I do next?

Being placed to all your preferred courses does not mean that you have finished your work on ECPRS. If you have been placed to more or less courses than the number of electives that you are allowed to register, read the answers for the relevant questions. If the number of courses that you have placed equals the number of courses that you are allowed to register, then your preregistration is complete. You can go to the interactive registrations room of the CC to register your preregistered courses along with your compulsory courses. This step is very important. Keep in mind that ECPRS is a preregistration system that only grants you the right to register elective courses. The official registration is done at the Computer Center's Registration Room.

I have been placed to less courses than the number courses I am allowed to register. What will I do?

If this is the case in the first bidding round, you can try your chance in the subsequent bidding rounds.

I am not placed to enough courses and there is no more bidding rounds. What will I do?

Check the course lists and find the courses that have left capacity and select the course that you would like to register. Go to your advisor. He/she will place you

to that course during the Manual Preregistrations period. The time of Manual Registrations period is announced by the Department. A course that you have been placed manually will cost you as much as the bid amount of the last student who has been placed to that course by the allocation procedure. During the between you select a course and the advisor tries to add you to a course, other manual placements may exhaust the capacity of the course. Therefore, select a number of alternative courses before you go to your advisor.

Can I change the courses that I have been placed?

If you are placed to a course that you preferred once, but not anymore, you have the right to cancel your preregistration for the course. But, remember that this change will cost you a penalty. To the change, you can use the system during the Reject period (see the answers to questions concerning the Reject period), or you can contact your advisor if there is no more Reject period. Then, you have to find a new course to replace it.

Should I worry if I am placed to more courses than I am allowed to register?

Certainly yes, because you will not be allowed to register the excess courses although you reserve a seat in them. And, there may other students (your friends) waiting for that space. You have to reject the course that you wont register to make it available to someone else.

What if I insist on not rejecting the excess courses?

You may not be fair, but ECPRS is. The system will compare the number of placed courses of each student and his/her allowed number of elective courses. If it discovers any exceeding courses, it will pick the courses with the least amount of bid points and automatically reject them. So, you may end up with an

unwanted course in your allowed courses while a course that you were planning to register is automatically rejected.

What is the Reject Period?

During the Reject period you specify the courses that you are placed but you will not register. There can be two reasons for not registering a course: It may be an excess course (the case of being placed to more courses than the allowed number) or it may be course that you don't want to take (a change in your preferences). You can read the answers of the corresponding questions. When you reject a course you are deleted from the students list of the course. The bid amount that you have given for this course is refunded to you except the penalty.

How can I reject a course?

During the Reject period, on the login screen of ECPRS, enter your student ID number and your password. You will be prompted the screen that contains your information. At the bottom of that page there is button called "Reject Courses". Clicking on that button will give you the page containing the list of the courses you have right to register. Next to each course there is a checkbox which is unchecked by default. Check the box of the courses you want to reject, then enter your password and click the submit button. Seeing a success message means that the courses are rejected.

This document talks about the Reject Period, whereas the announcements of the Department don't mention it. Why?

This is possible. The supervisor may prefer to skip the Reject period. The reason can be time shortages. But this does not mean that the courses that exceed the allowed number of elective courses will not be canceled. With no Reject period, all rejections will be done automatically by the system. What you have to do is to read the announcements very carefully. If there is no Reject

period, then place your bids logically so that if automatic rejection is carried out, you don't loose the right to register a course that you really want to take.

What are the deduction policies?

Deduction policies are announced by the Department at the beginning of registrations. Although the stability of the deduction policies are crucial for fairness of the system, the Department reserves its right to modify them, if necessary.

How can I learn my present bid credits available?

On the information screen of the ECPRS, you can see the amount of remaining credits as of the beginning of that semester. You can also go to your advisor to get a more accurate figure. At the end of each semester, the Department announces the deducted bids, for your review.

What should I do in the Add-Drops?

During the Add-Drops, the allocation procedure will not be executed. Therefore, you will not need to bid. Any changes that are necessary or needed will be done by your advisor manually. The bid amount given for the dropped courses will not be refunded.

Should I have a user account on any computer?

No. Your accounts on hosts like orca, olympos, or the Novell Netware are totally irrelevant to this system.

I cannot come to the campus for registrations. Can I still bid?

As ECPRS is a WWW-based application, it is possible to use it remotely. However, you have to know your password in advance. Additionally, all of the information may not be available on the system (like accurate course outlines, or

time schedule of the activities in ECPRS). It is always safer to come to the campus. If you really cannot, then keep in contact with your advisor via phone.

APPENDIX G

Elective Course Pre-Registration System

HTML Files and TCL Scripts

The Elective Course Pre-Registration System software has been implemented by using the programming language TCL and the markup language HTML. These files are described in this appendix. The figures show the flow of control between the files and programs.

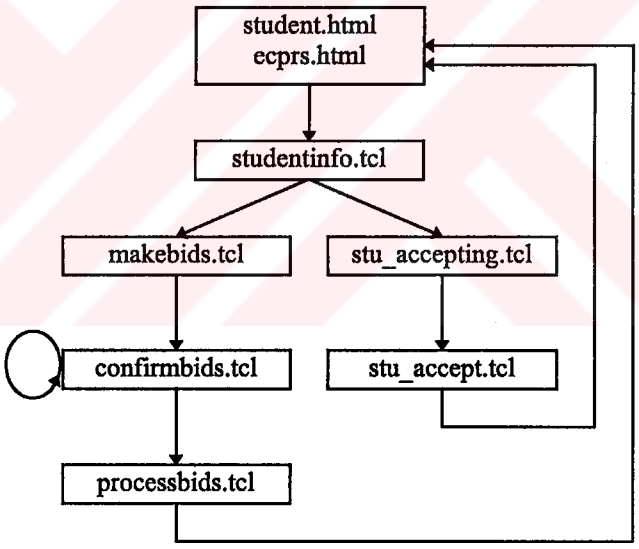


Figure G.1 - The files and programs concerning the students

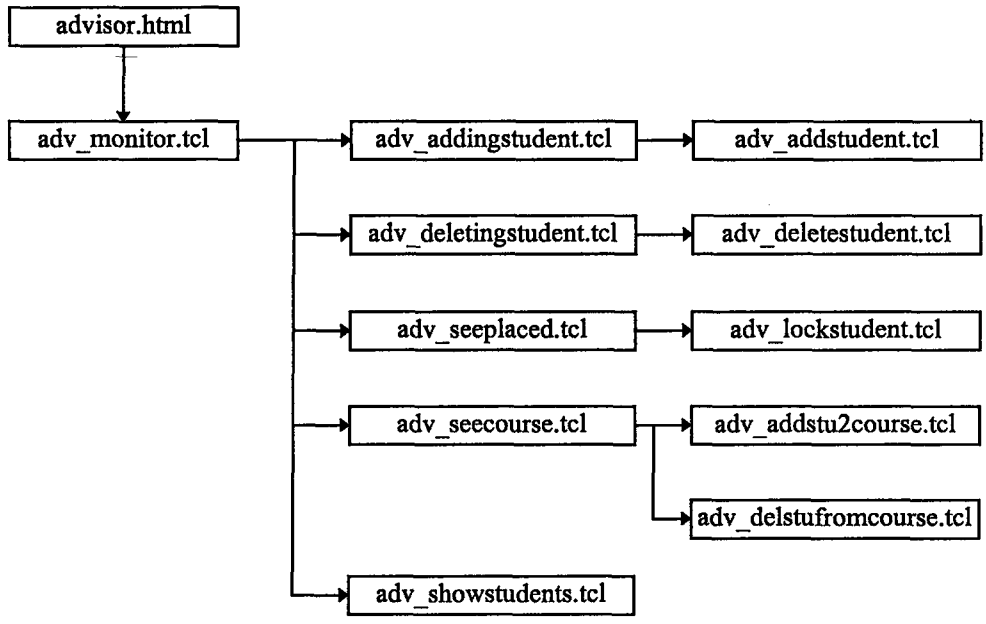


Figure G.2 - The files and programs concerning the advisors

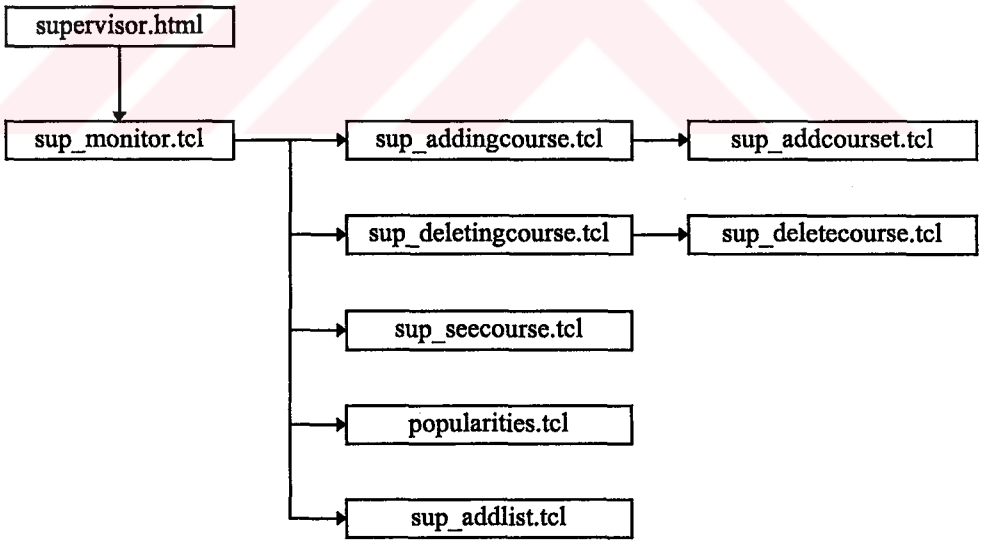


Figure G.3 - The files and programs concerning the supervisors

/home/ecrs/public_html/student.html

Collects from browser:

- student ID number
- student password

Passes control to:

/home/ecrs/tcls/studentinfo.tcl

Includes a link to:

/home/ecrs/public_html/stufaq.html

See figure:

Figure G.4

/home/ecrs/public_html/ecprs.html

File Symbolically linked to:

/home/ecrs/public_html/student.html

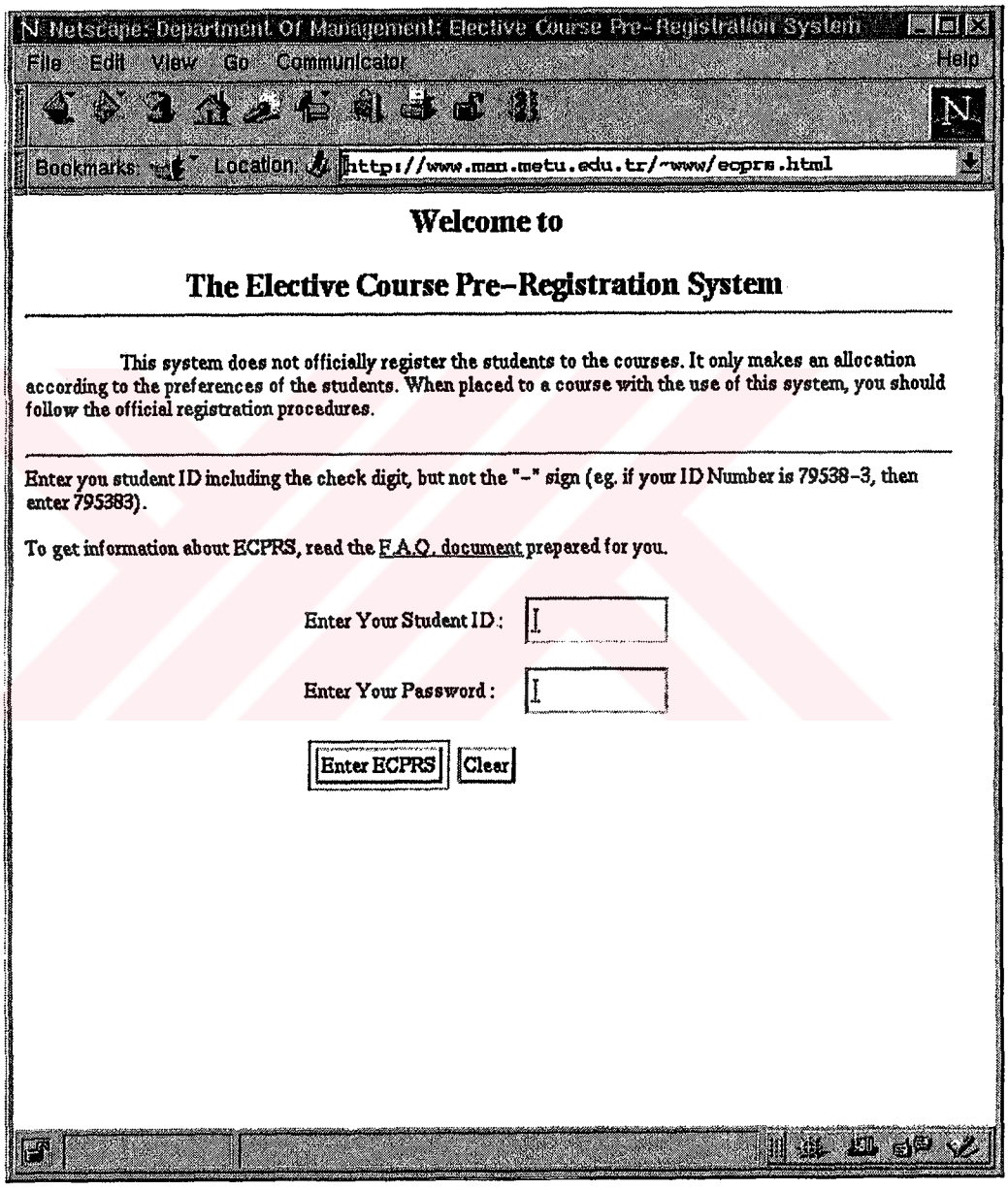


Figure G.4 - Student Opening Page

/home/ecrs/tcls/studentinfo.tcl

Inputs:

- student ID number passed as argument
- student password passed as argument
- status read from */home/ecrs/supervisor/status.sup*
- reads the information of the student from */home/ecrs/students/xxx.stu*

Checks:

status: If the status is LOCKED, NOSTUDENTACCESS or SUPERVISORONLY, the students cannot access the programs in the system.

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

ID number: Corresponding error messages are displayed in the browser when the ID number is not entered and it is incorrect. Also, the student information cannot be displayed if the student file is locked and the error message is displayed.

Passes control to:

Depending on the status of the system, the flow is as follows:

Status	**File**
BP1, BP2	*/home/ecrs/tcls/makebids.tcl*
AP1, AP2	*/home/ecrs/tcls/stu_accepting.tcl*

See figure:

Figure G.5

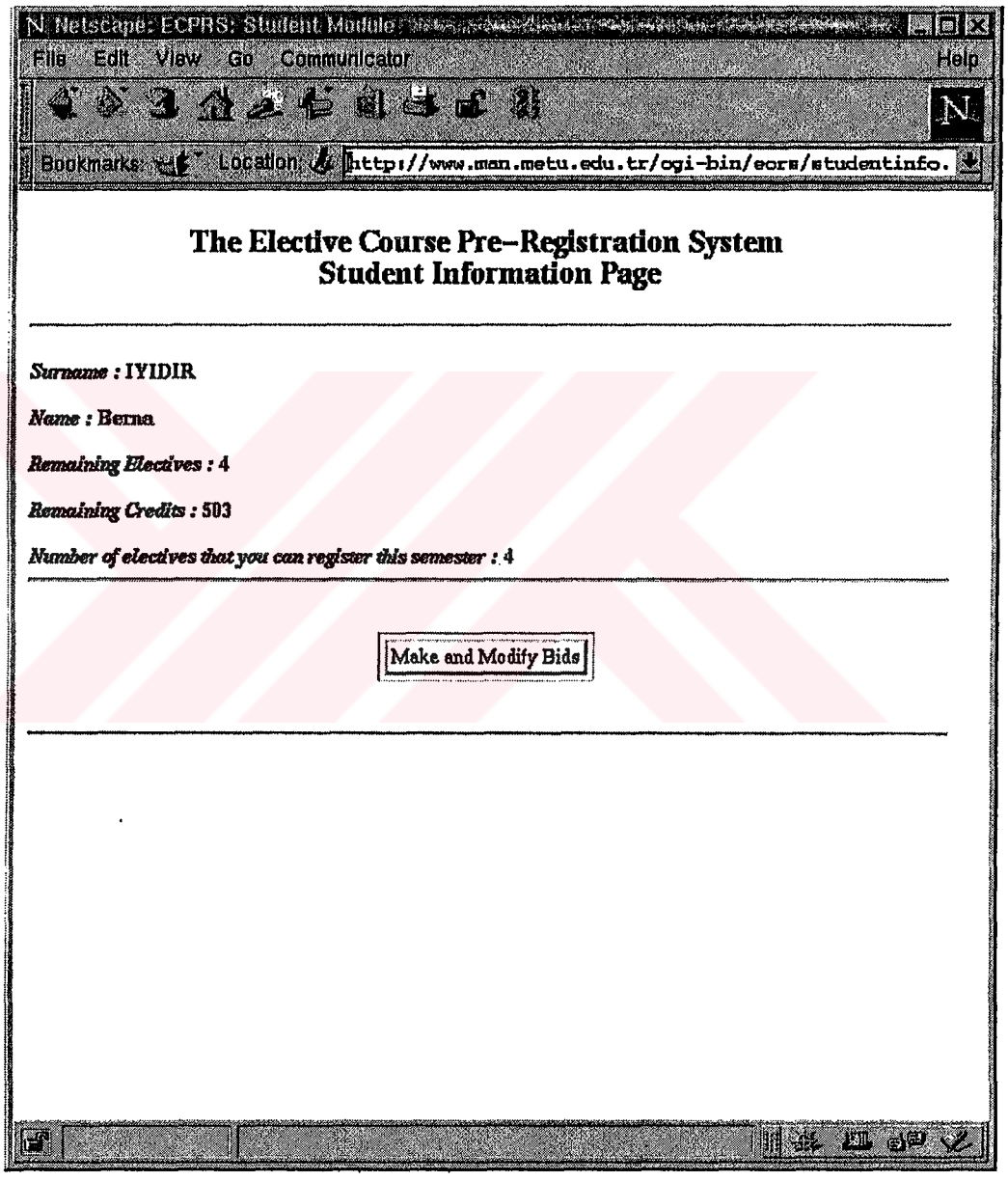


Figure G.5 - Student Information Page

/home/ecrs/tcls/makebids.tcl

Inputs:

- student ID number passed as argument
- present bids of the student read from */home/ecrs/students/xxx.stu*, if there are any

Collects from browser:

- At most seven "course code, bid amount, operator, couple course, couple bid amount" sets (called as **bids** in short)

Passes control to:

/home/ecrs/tcls/confirmbids.tcl

Includes link to:

/home/ecrs/public_html/bidhelp.html

See figure:

Figure G.6

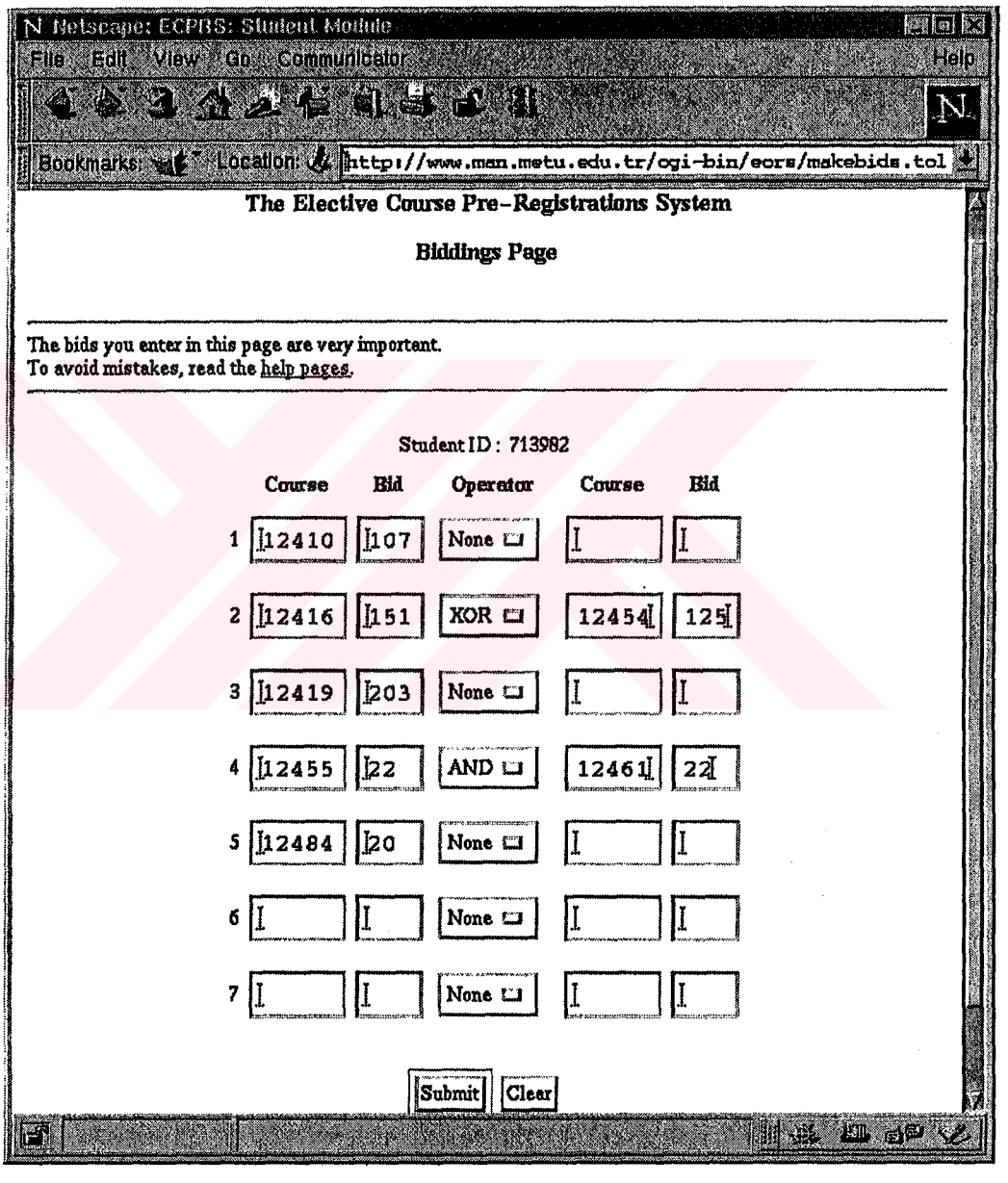


Figure G.6 - Student Bidding Page

/home/ecrs/tcls/confirmbids.tcl

Inputs:

- student ID number passed as argument
- bids passed as argument
- list of elective courses being processed in the system
- remaining credits of the student read from student file

Collects from browser:

- student password
- At most seven "course code, bid amount, operator, couple course, couple bid amount" sets (called as **bids** in short)

Checks:

The program checks the following in the bids:

- course code has been entered
- course code is valid
- course code is not repeated in more than one bids
- bid amount has been entered
- bid amount is in the range [20-250]
- operator has been entered
- couple course has been entered
- couple bid amount has been entered
- total of bid amounts is less than/equal to remaining amount of credits of student
- the number of bidden courses is less than or equal to seven

Passes control to:

Depending on the results of checks of the bids:

- */home/ecrs/tcls/confirmbids.tcl* is called again if there are any errors (to allow correction of invalid bids)
- */home/ecrs/tcls/processbids.tcl* is called if there is no error

Includes link to:

/home/ecrs/public_html/bidhelp.html

See figures:

Figure G.7

Figure G.8

Figure G.9

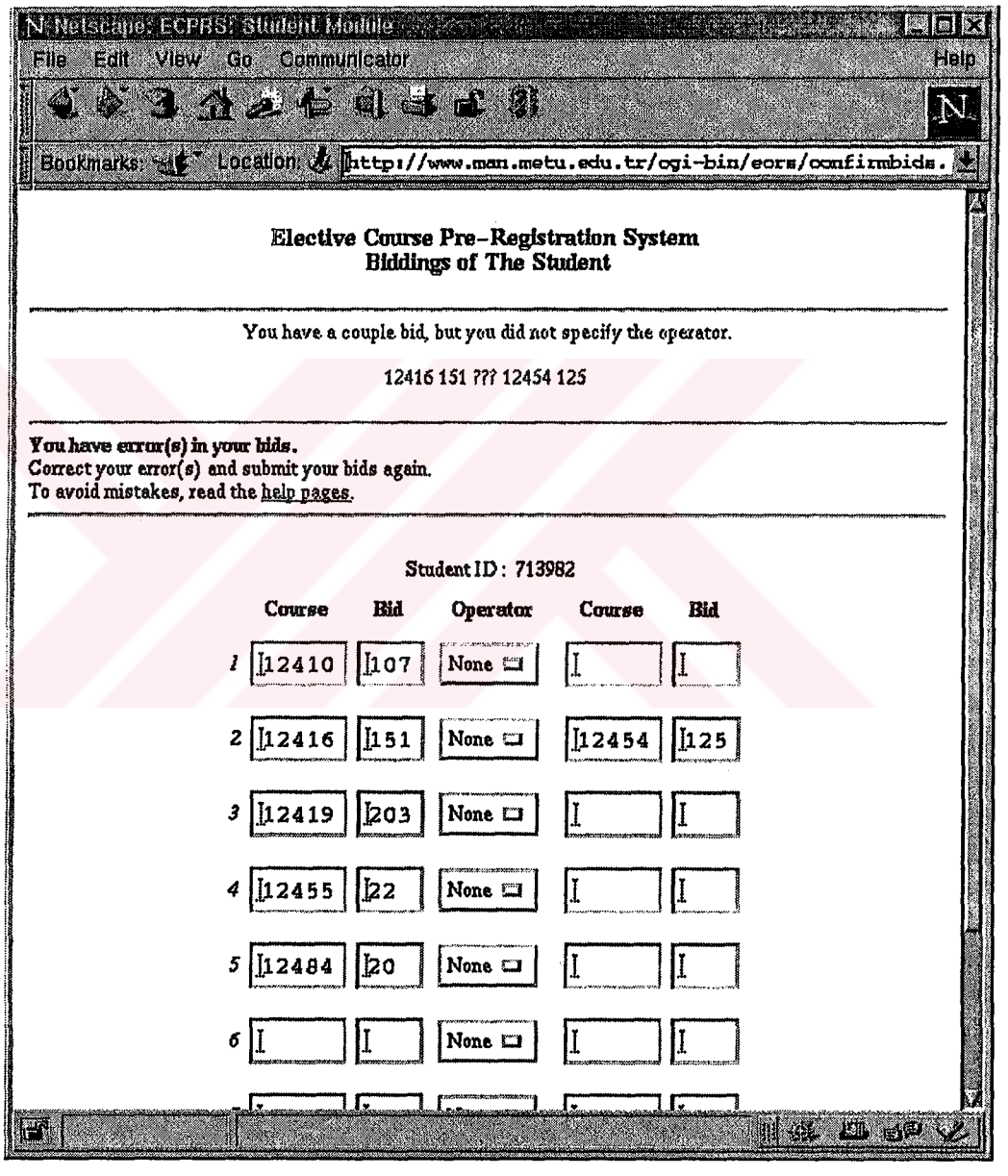


Figure G.7 - Error in Bid

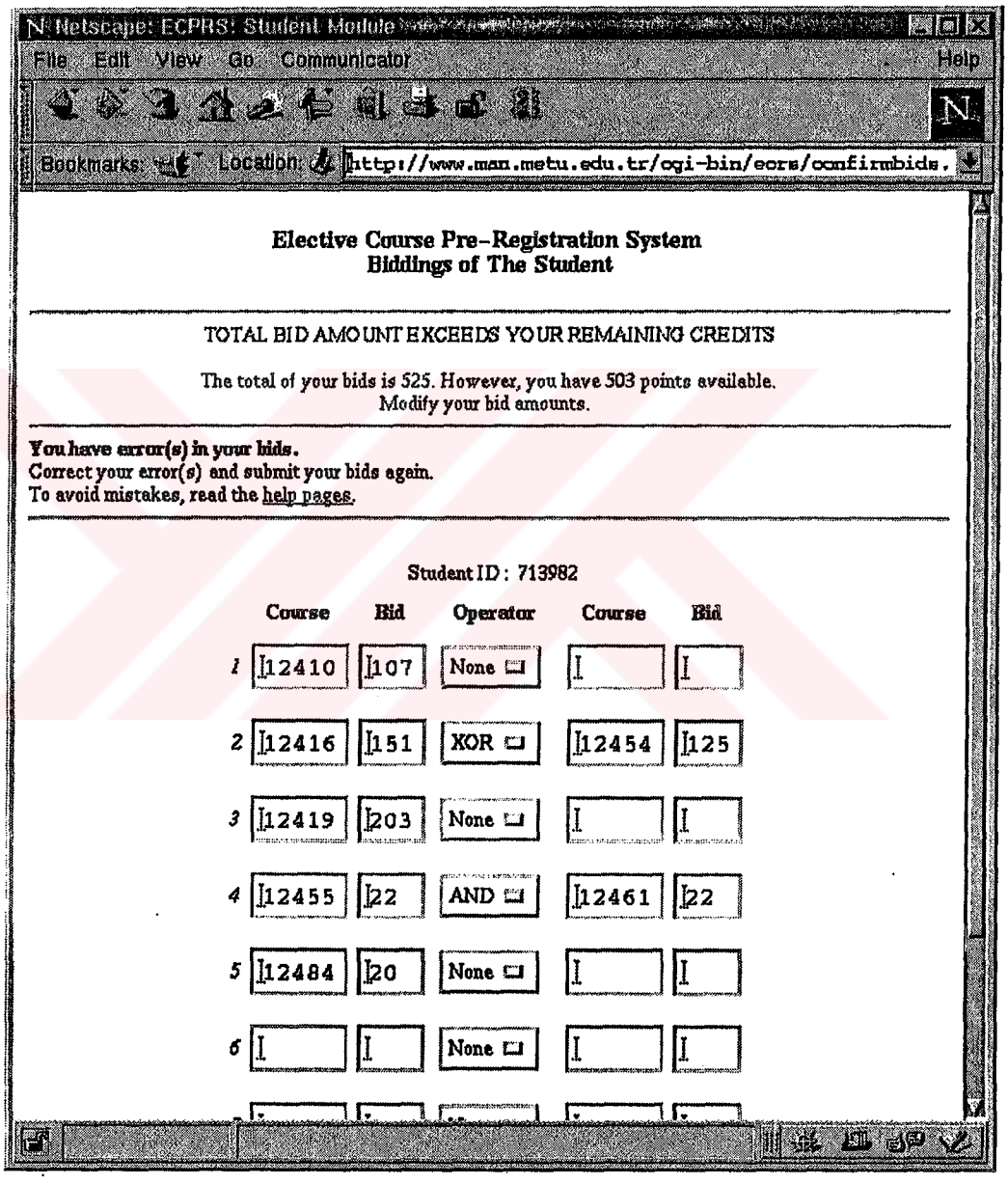


Figure G.8 - Remaining Credits Error

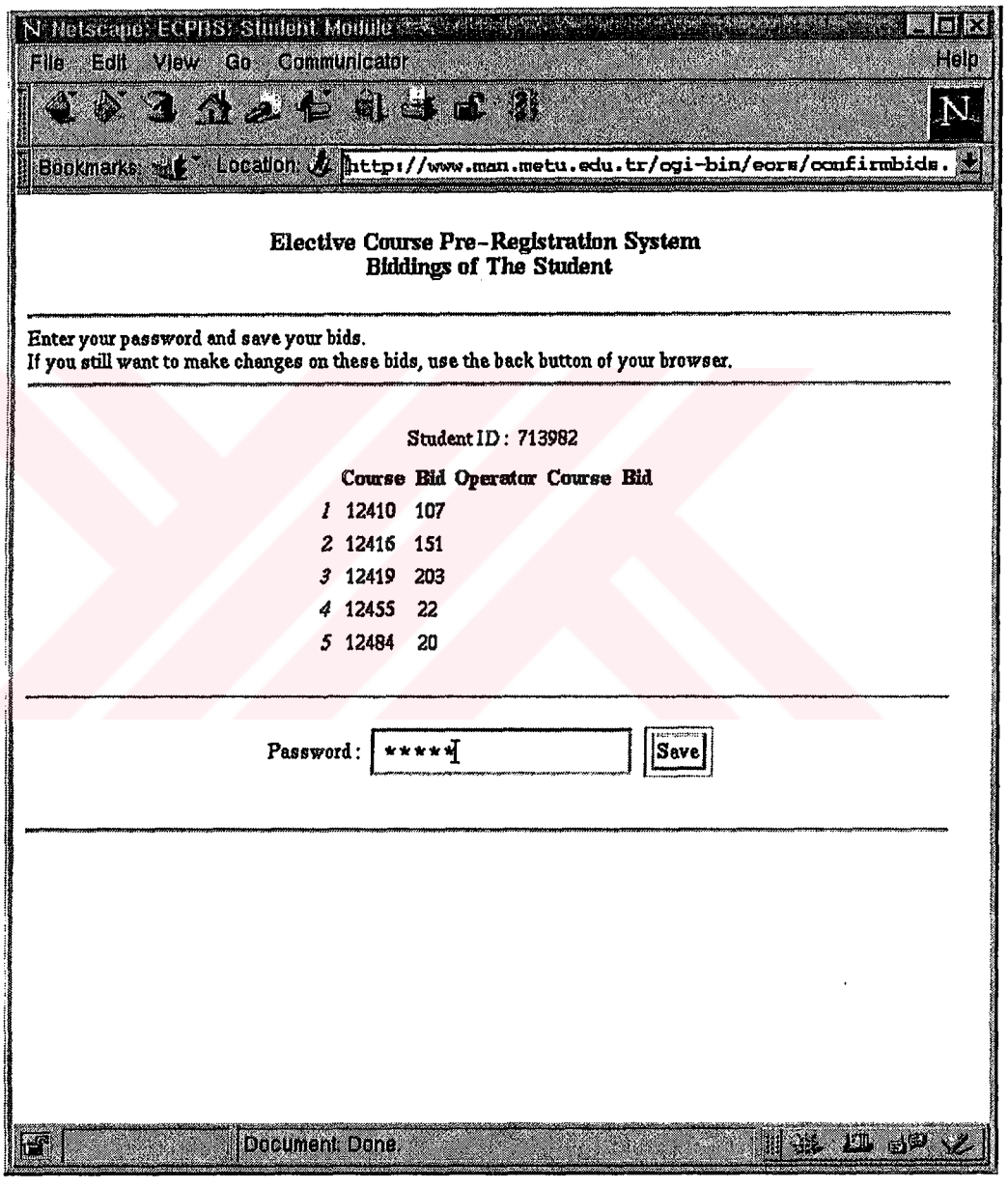


Figure G.9 - Correct Bids

/home/ecrs/tcls/processbids.tcl

Inputs:

- student ID number passed as argument
- student password passed as argument
- bids passed as argument
- present bids of the student read from */home/ecrs/students/xxx.stu*, if there are any

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

Passes control to:

/home/ecrs/public_html/student.html

See figure:

Figure G.10

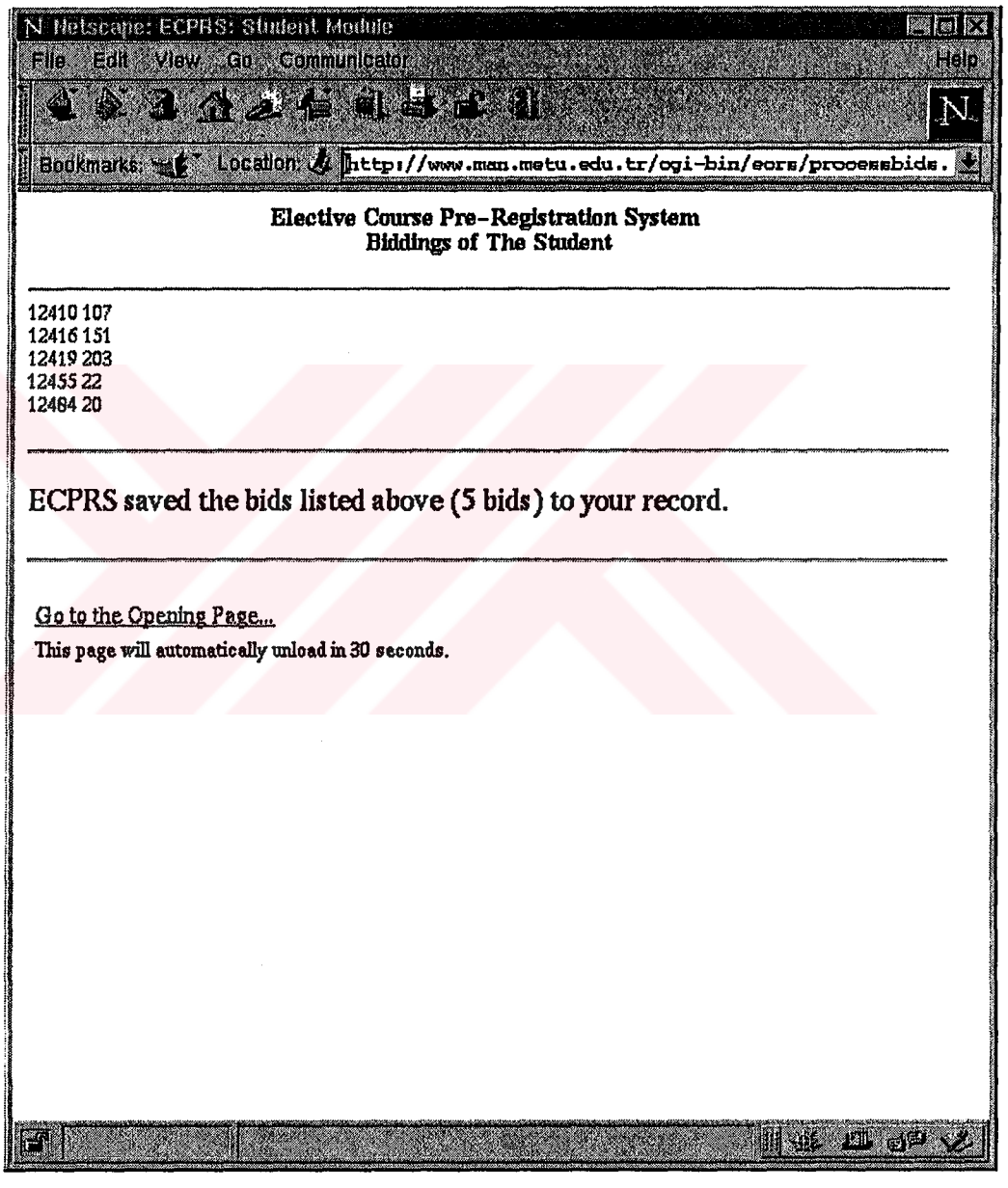


Figure G.10 - Saved Bids

/home/ecrs/tcls/stu_accepting.tcl

Inputs:

- student ID number passed as argument
- placed courses of the student read from */home/ecrs/students/xxx.stu*, if there are any
- number of allowed courses of the student read from */home/ecrs/students/xxx.stu*

Collects from browser:

- student password
- rejected courses

Checks:

If the number of allowed courses is more than the number of placed courses, a message about the necessity of rejecting courses is displayed.

Passes control to:

/home/ecrs/public_html/student.html, if the student has not been placed to any course

/home/ecrs/tcls/stu_accept.tcl, otherwise.

See figure:

Figure G.11

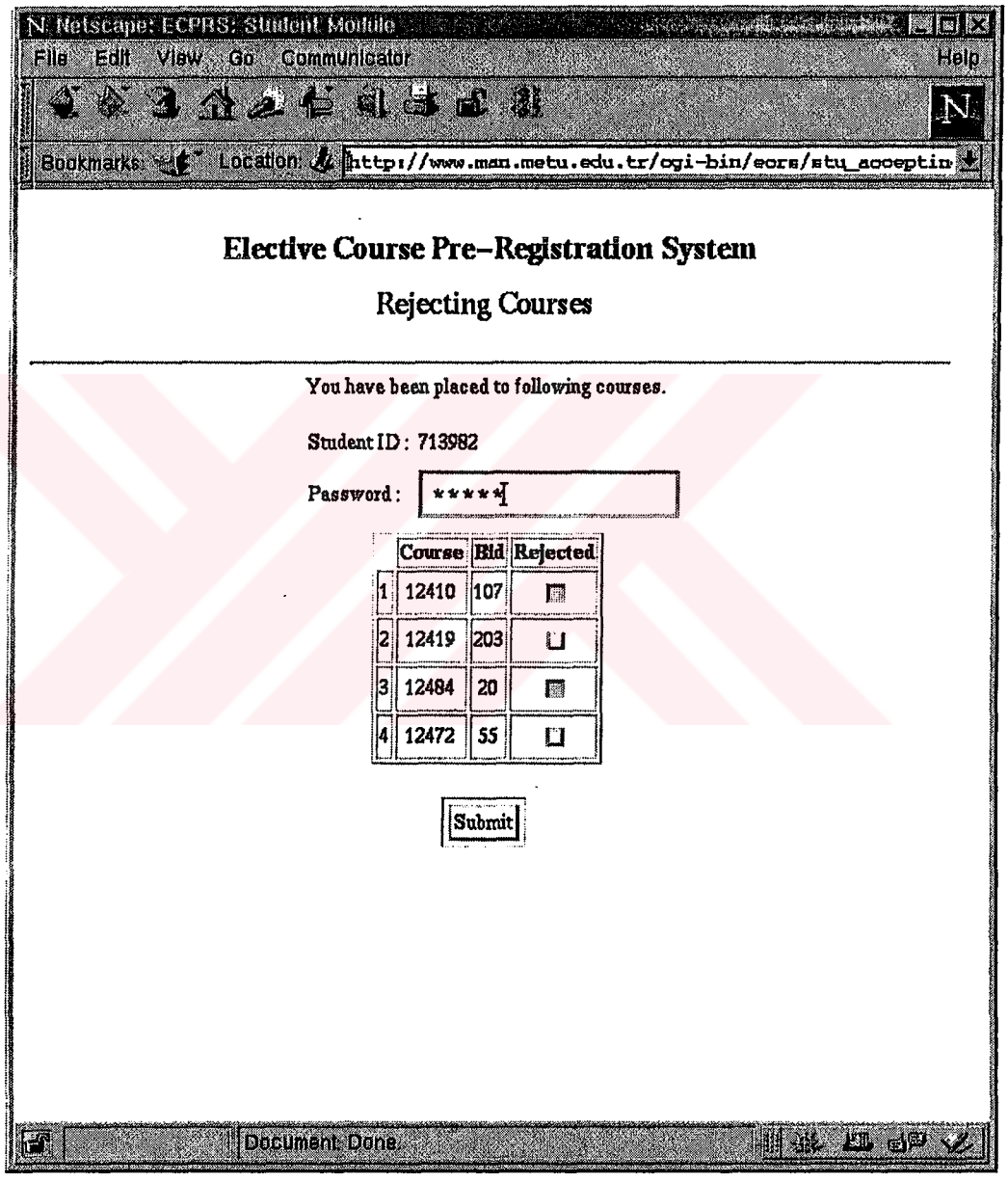


Figure G.11 - Rejecting Courses

/home/ecrs/tcls/stu_accept.tcl

Inputs:

- student ID number passed as argument
- student password passed as argument
- placed courses of the student passed as argument
- rejected courses of the student passed as argument
- placed courses of the student read from */home/ecrs/students/xxx.stu*

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

rejections: if there are any rejections, the student file is modified. The file remains the same if there are no rejections.

Passes control to:

/home/ecrs/public_html/student.html, automatically after waiting for 10 seconds.

/home/ecrs/public_html/advisor.html

Collects from browser:

- advisor ID number
- advisor password

Passes control to:

/home/ecrs/tcls/adv_monitor.tcl

Includes link to:

/home/ecrs/public_html/stufaq.html

/home/ecrs/tcls/adv_monitor.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- status read from */home/ecrs/supervisor/status.sup*
- information of the advisor read from */home/ecrs/supervisor/advisors.sup*

Collects from browser:

- student ID number
- course code

Checks:

status: If the status is LOCKED or SUPERVISORONLY, the advisors cannot access the programs in the system.

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

ID number: Corresponding error messages are displayed in the browser when the ID number is not entered and it is incorrect.

Passes control to:

Depending on the type of advisor:

if this is an MBA or NONMGMT (Non-Management) advisor the next screen is */home/ecrs/tcls/adv_seecourse.tcl*

if this is a MGMT (Management) advisor, depending on the choice one of the following

/home/ecrs/tcls/adv_addingstudent.tcl

/home/ecrs/tcls/adv_deletingstudent.tcl

/home/ecrs/tcls/adv_seeplaced.tcl

/home/ecrs/tcls/adv_seecourse.tcl

/home/ecrs/tcls/adv_showstudents.tcl

See figure:

Figure G.12

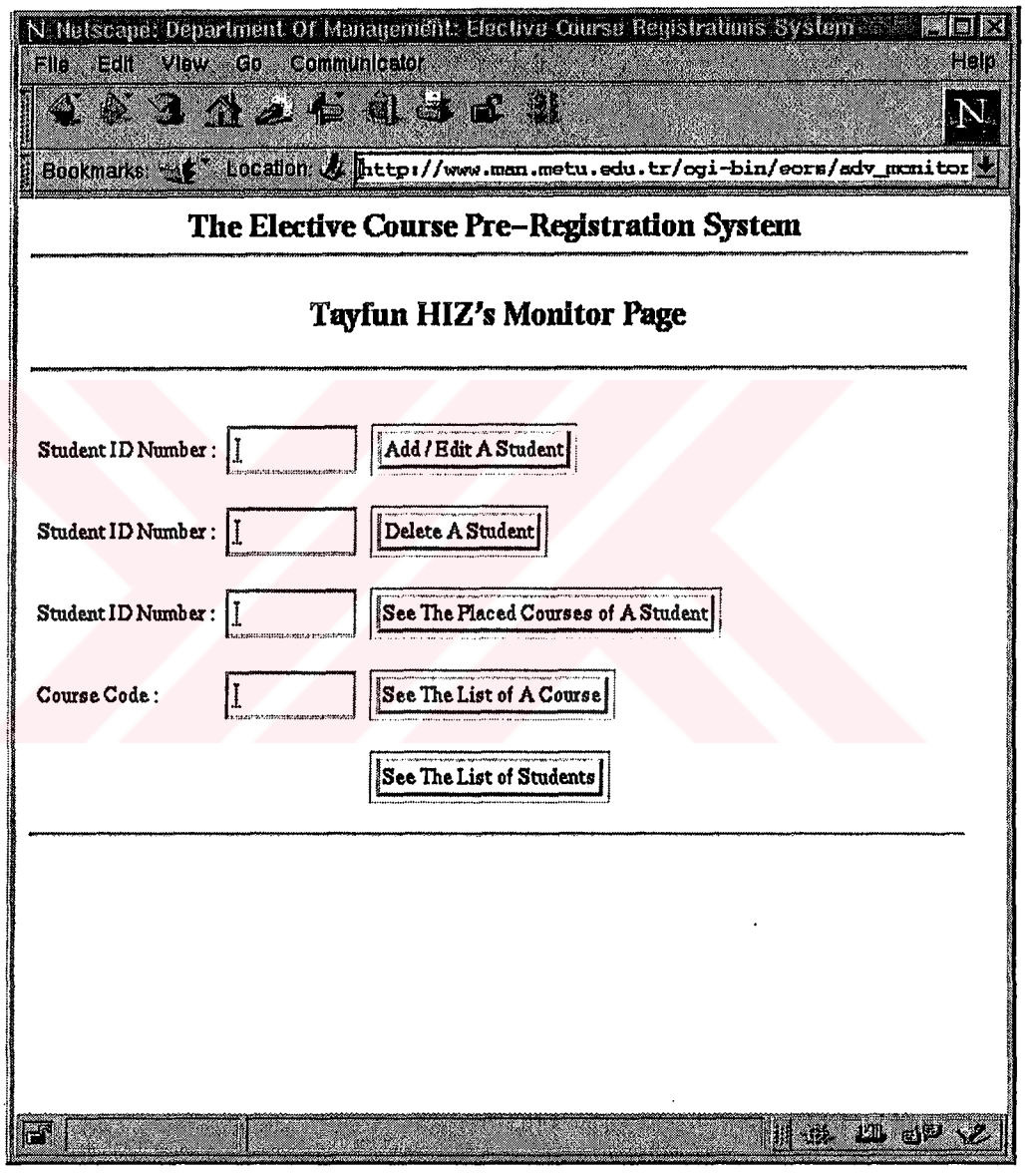


Figure G.12 - Advisor's Monitor Page

/home/ecrs/tcls/adv_addingstudent.tcl

Inputs:

- advisor ID number passed as argument
- student ID number passed as argument
- information of the student read from */home/ecrs/students/xxx.stu*

Collects from browser:

- advisor password
- information of the student

Checks:

Whether the student file exists. If it does, it allows editing. Otherwise, it allows entry of a new student.

Passes control to:

/home/ecrs/tcls/adv_addstudent.tcl

See figures:

Figure G.13

Figure G.14

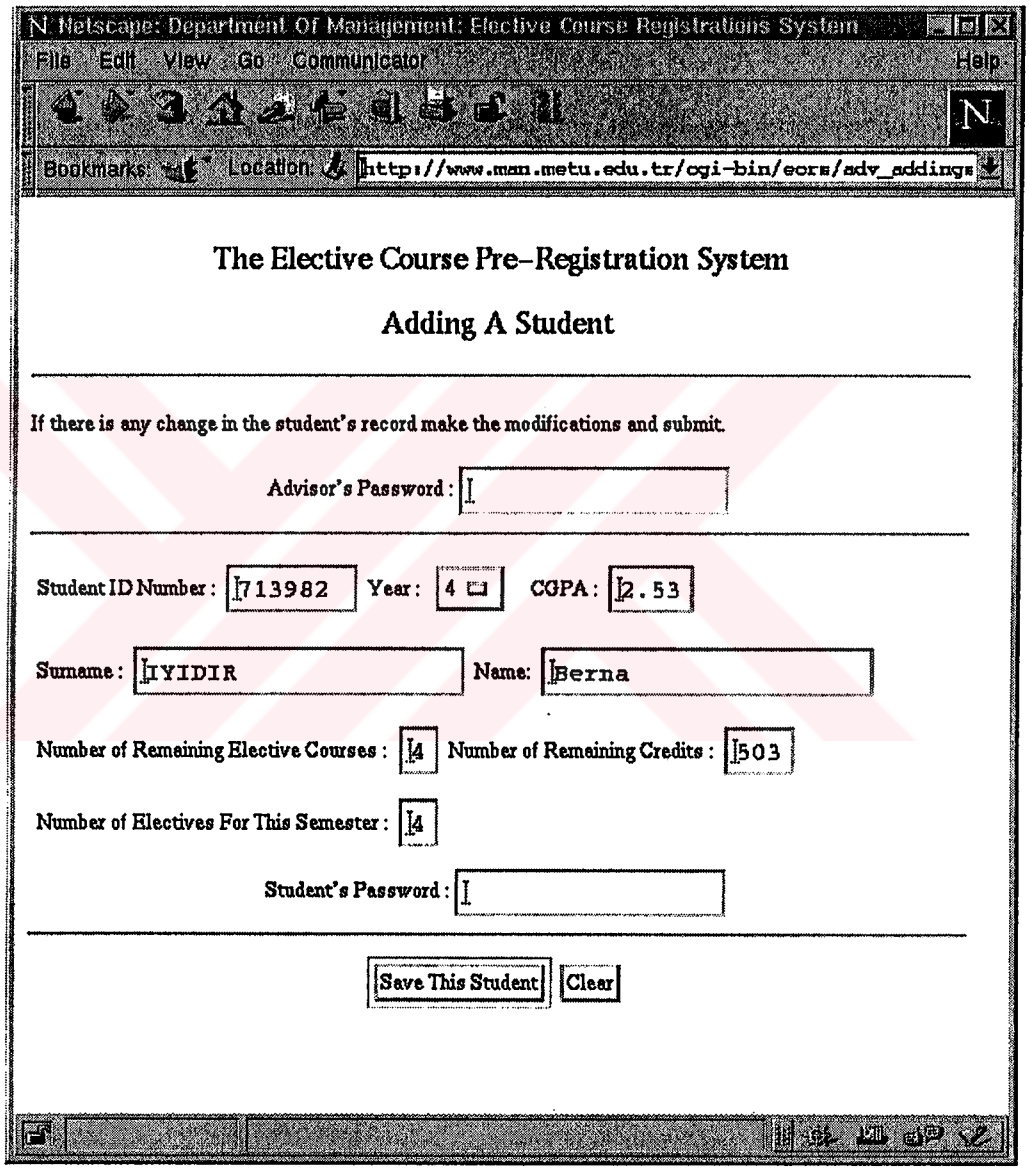


Figure G.13 - Adding a Student

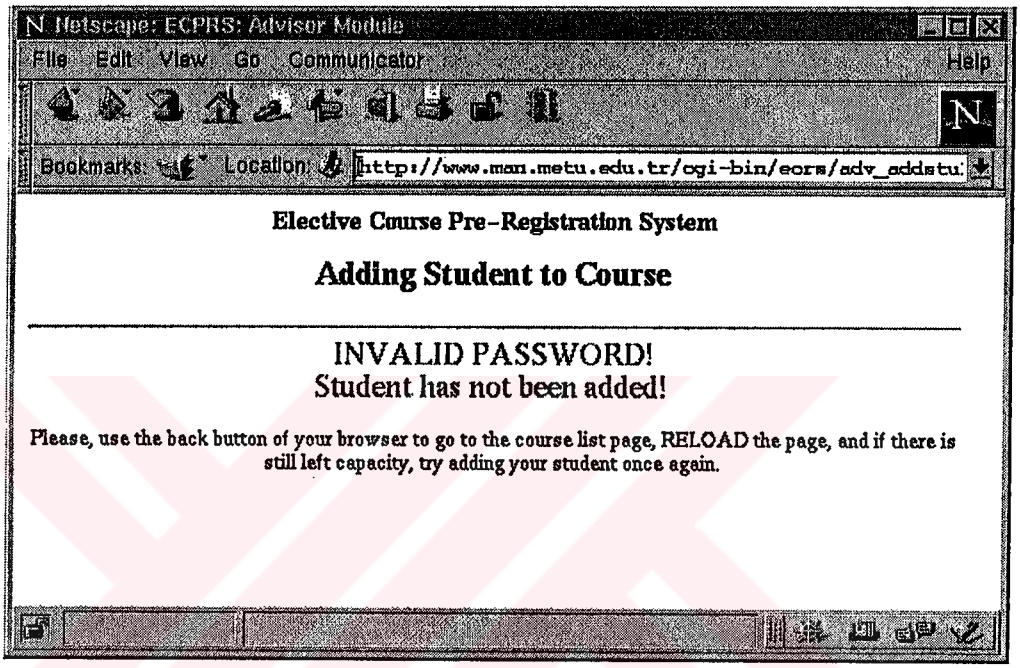


Figure G.14 - Password Error While Adding Student

/home/ecrs/tcls/adv_addstudent.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- student information passed as argument
- information of the student read from */home/ecrs/students/xxx.stu*

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

student file: if it exists and the information passed as argument is different than the contents of the file then the file is updated. If it does not exist, the student file is created with the information passed as argument.

Passes control to:

/home/ecrs/public_html/advisor.html

/home/ecrs/tcls/adv_deletingstudent.tcl

Inputs:

- advisor ID number passed as argument
- student ID number passed as argument
- information of the student read from */home/ecrs/students/xxx.stu*

Collects from browser:

- advisor password

Checks:

Whether the student file exists. If it does not, it gives an error message.

Passes control to:

/home/ecrs/tcls/adv_deletestudent.tcl

/home/ecrs/tcls/adv_deletestudent.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- student information passed as argument

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

student file: if it exists the file is deleted. If it does not exist, it gives an error message.

Passes control to:

/home/ecrs/public_html/advisor.html

/home/ecrs/tcls/adv_seeplaced.tcl

Inputs:

- advisor ID number passed as argument
- student ID number passed as argument
- information of the student read from /home/ecrs/students/xxx.stu
- course prerequisites read from multiple file /home/ecrs/courses/xxx.crs

Collects from browser:

- advisor password

Checks:

student ID number: If student ID number has not been entered, it gives an error message. If the entered student ID number does not really exist in the system, the program gives again an error message.

placed courses of the student: If the student has not been placed to any courses, it gives a corresponding error message.

student record file: If the file has already been locked, it gives a message indicating that the students file is locked. Otherwise, it displays the "Lock" button.

Passes control to:

Depending on the locking status of the student file:

/home/ecrs/tcls/adv_lockstudent.tcl, if the file is not locked

/home/ecrs/public_html/advisor.html, if the file is locked

See figure:

Figure G.15

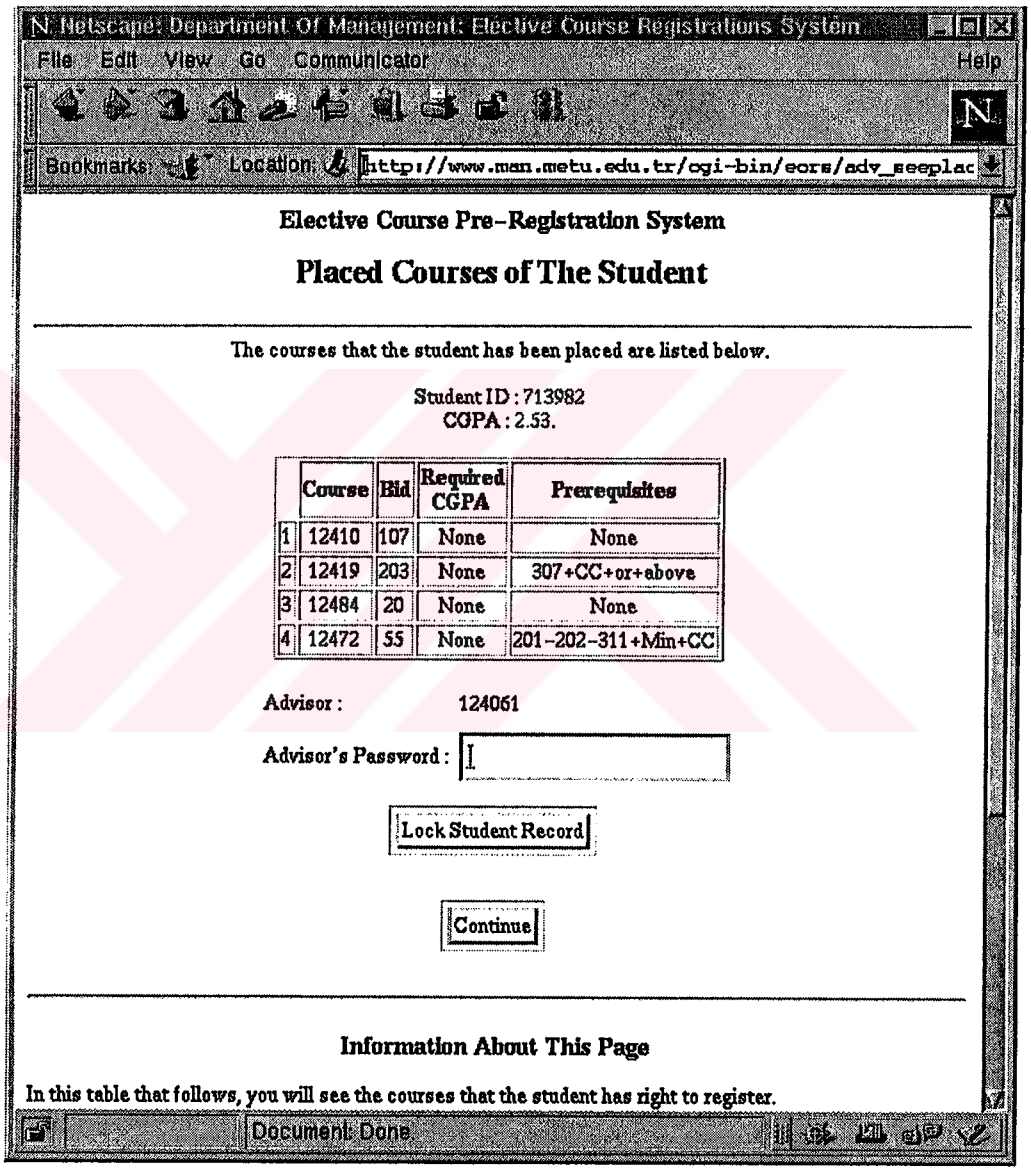


Figure G.15 - Placed Courses of a Student

/home/ecrs/tcls/adv_lockstudent.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- student ID number passed as argument

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

student file: if it is not already locked, the program locks it by moving it to the file */home/ecrs/locked/xxx.lck.*

Passes control to:

/home/ecrs/public_html/advisor.html

/home/ecrs/tcls/adv_seecourse.tcl

Inputs:

- advisor ID number passed as argument
- course code passed as argument
- information of the advisor read from */home/ecrs/supervisor/advisors.sup*
- course information read from */home/ecrs/courses/xxx.crs*

Collects from browser:

- student ID number of the student to be added to / deleted from the course

Checks:

course code: If the course code has not been entered, the error message is diploid. If the course code entered does not really exist in the system, the program gives again an error message.

course capacity: If the number of students placed to the course exceeds the course capacity then the warning message is displayed.

Passes control to:

Depending on the choice :

/home/ecrs/tcls/adv_addstu2course.tcl, if the student will be added to the course file

/home/ecrs/tcls/adv_delstufromcourse.tcl, if the student will be deleted from the course file

/home/ecrs/tcls/adv_addstu2course.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- student ID number passed as argument
- student type passed as argument
- course code passed as argument
- course capacity passed as argument
- course capacity read from */home/ecrs/courses/xxx.crs*

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

student ID number: : If student ID number has not been entered, it gives an error message.

course capacity: If the course capacity passed as argument is not the as the course capacity read from the file, then an addition has been done by someone else so the operation is canceled and an error message is given.

course file: If the course file is in use by another advisor (i.e. an addition is being done) then the operation is canceled and an error message is given.

Passes control to:

/home/ecrs/public_html/advisor.html

See figures:

Figure G.16

Figure G.17

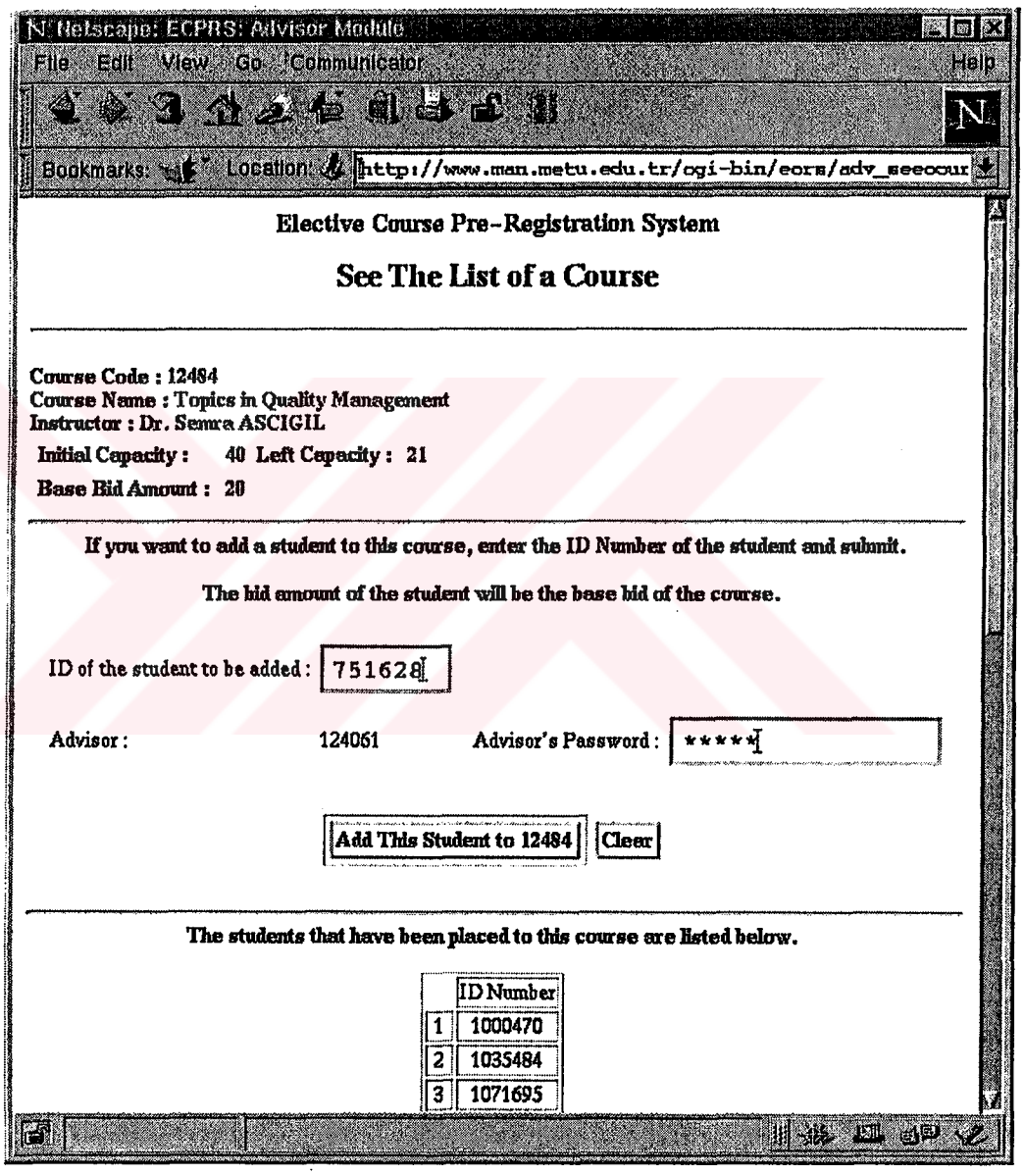


Figure G.16 - List of a Course

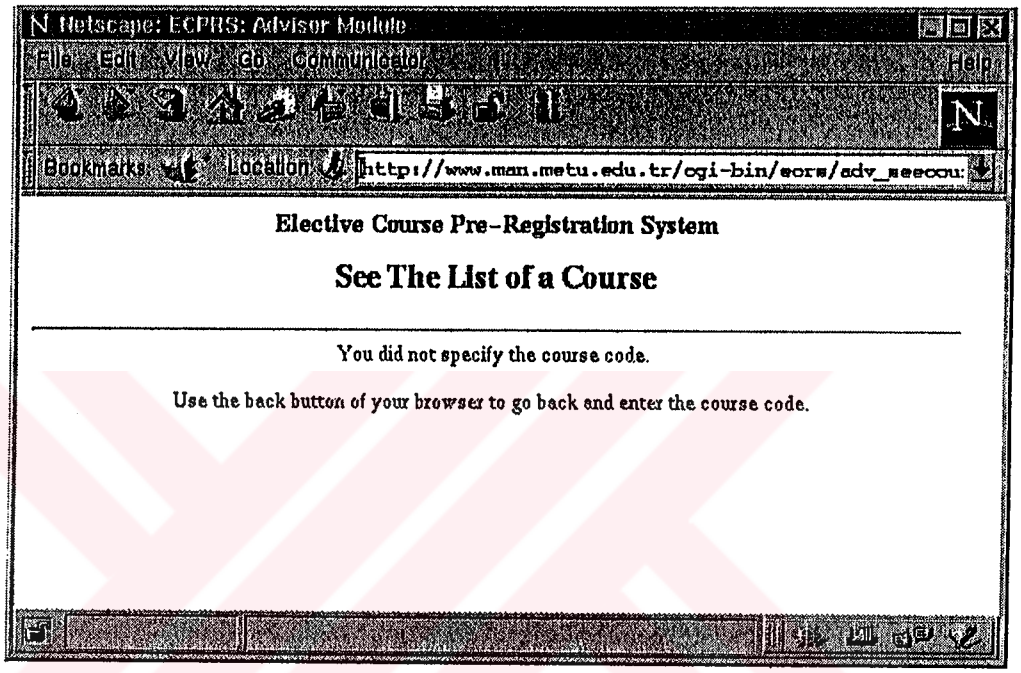


Figure G.17 - Course Code Error in List of a Course

/home/ecrs/tcls/adv_delstufromcourse.tcl

Inputs:

- advisor ID number passed as argument
- advisor password passed as argument
- student ID number passed as argument
- student type passed as argument
- course code passed as argument
- course capacity passed as argument
- course capacity read from */home/ecrs/courses/xxx.crs*
- placement information read from course file */home/ecrs/courses/xxx.crs*
- placement information read from student file */home/ecrs/student/xxx.stu*

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

student ID number: : If student ID number has not been entered, it gives an error message.

course file: If the course file is in use by another advisor (i.e. an addition is being done) then the operation is canceled and an error message is given.

placement information: A bid is deleted only if it is really in both the course file and the student file. Otherwise, the operation is canceled and an error message is given.

Passes control to:

/home/ecrs/public_html/advisor.html

/home/ecrs/tcls/adv_delstufromcourse.tcl

Inputs:

- advisor ID number passed as argument
- student information read from multiple student files in */home/ecrs/students*
- status information read from */home/ecrs/supervisor/status.sup*

Checks:

status: If the status is LOCKED or SUPERVISORONLY, the advisors cannot access the programs in the system.

student information: If the student file is not locked and the ID number of the advisor of the student is the same as the advisor ID number passed as argument, then the name, surname and ID number of the student is displayed.

/home/ecrs/public_html/supervisor.html

Collects from browser:

- supervisor login name
- supervisor password

Passes control to:

/home/ecrs/tcls/adv_monitor.tcl

/home/ecrs/tcls/sup_monitor.tcl

Inputs:

- supervisor ID number passed as argument
- advisor password passed as argument

Collects from browser:

- course code
- student type to be listed
- fields to be listed
- sort key for the list
- name of the comma separated student list file

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

ID number: Corresponding error messages are displayed in the browser when the ID number is not entered and it is incorrect.

Passes control to:

Depending on the choice one of the following

/home/ecrs/tcls/sup_addingcourse.tcl

/home/ecrs/tcls/sup_deletingcourse.tcl

/home/ecrs/tcls/sup_seecourse.tcl

/home/ecrs/tcls/popularities.tcl

/home/ecrs/tcls/sup_addlist.tcl

See figure:

Figure G.18

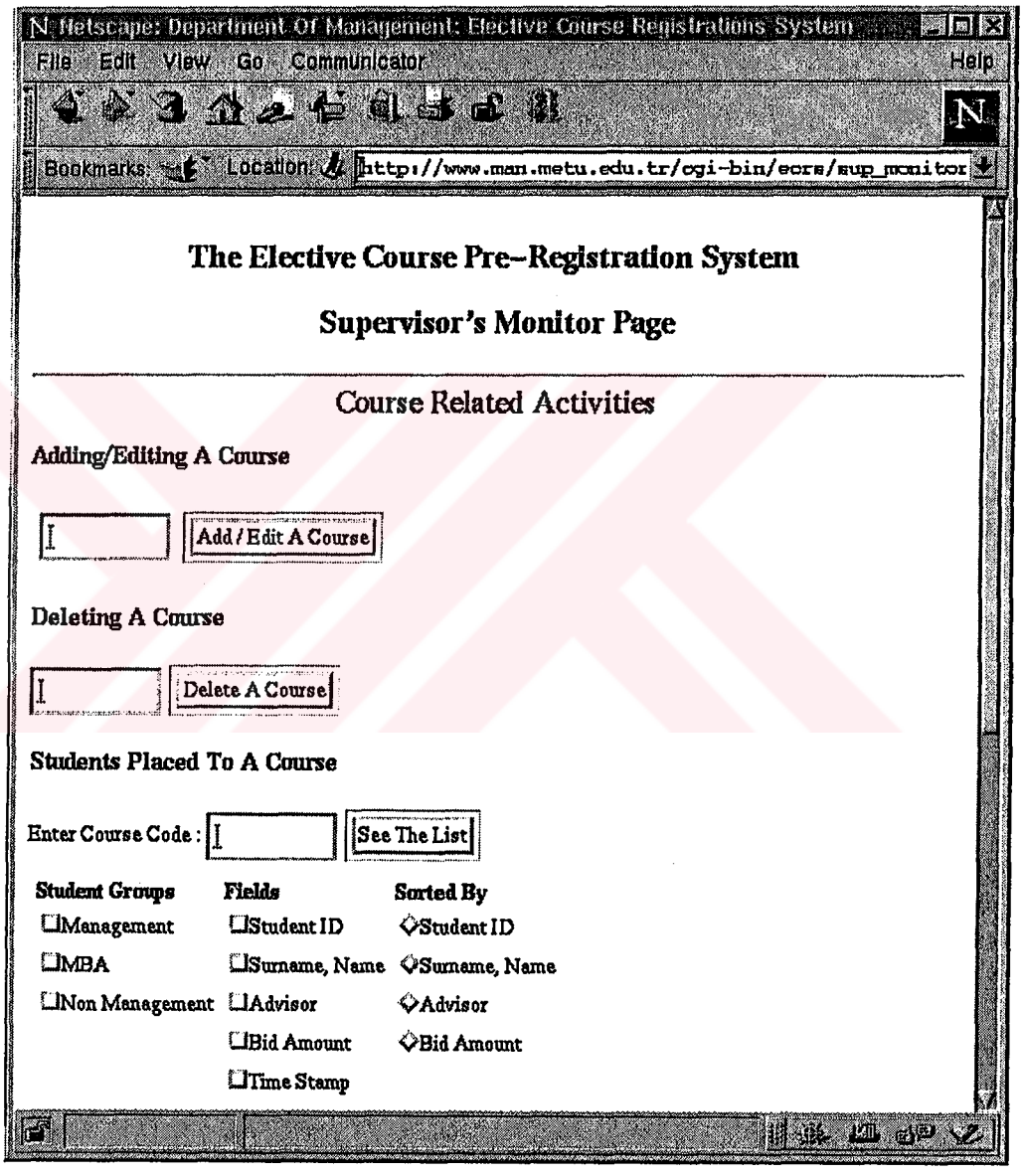


Figure G.18 - Supervisor's Monitor Page

/home/ecrs/tcls/sup_addingcourse.tcl

Inputs:

- supervisor login name passed as argument
- course code passed as argument
- course information read from */home/ecrs/courses /xxx.crs*

Collects from browser:

- supervisor password
- course information

Checks:

Whether the course file exists. If it does, it allows editing. Otherwise, it allows entry of a new course.

Passes control to:

/home/ecrs/tcls/sup_addcourse.tcl

See figure:

Figure G.19

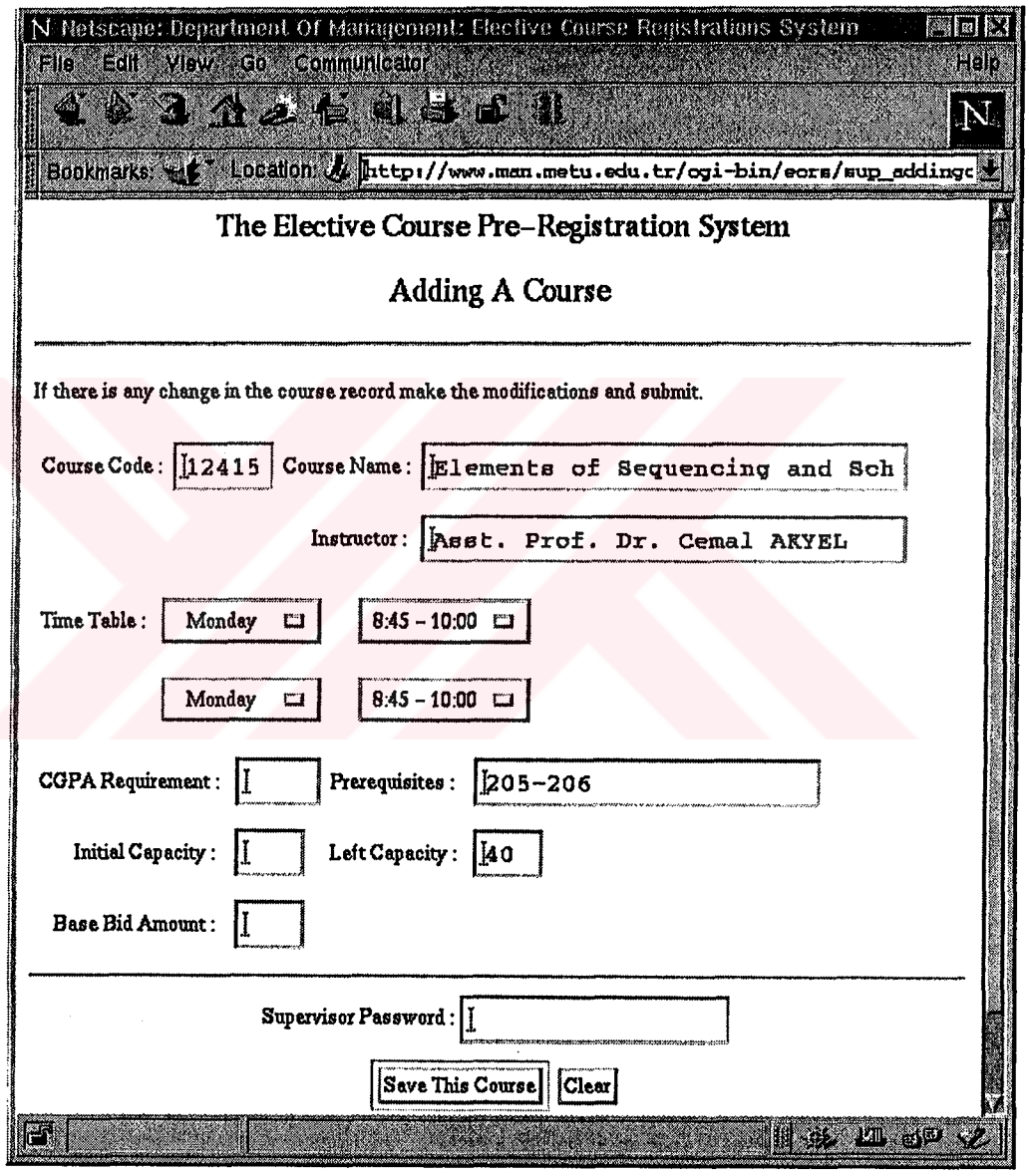


Figure G.19 - Adding a Course

/home/ecrs/tcls/sup_addcourse.tcl

Inputs:

- supervisor login name passed as argument
- supervisor password passed as argument
- course information passed as argument
- course information read from */home/ecrs/courses /xxx.crs*

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

course file: if it exists and the information passed as argument is different than the contents of the file then the file is updated. If it does not exist, the course file is created with the information passed as argument.

Passes control to:

/home/ecrs/public_html/supervisor.html

/home/ecrs/tcls/sup _deletingcourse.tcl

Inputs:

- supervisor login name passed as argument
- course code passed as argument
- course information read from */home/ecrs/courses /xxx.crs*

Collects from browser:

- supervisor password

Checks:

course file : Whether the course file exists. If it does not, it gives an error message.

Passes control to:

/home/ecrs/tcls/sup _deletecourse.tcl

/home/ecrs/tcls/sup_deletecourse.tcl

Inputs:

- supervisor login name passed as argument
- supervisor password passed as argument
- course code passed as argument

Checks:

password: ID number and password of the user are encrypted and compared with the password file entry. If they are not identical, an error message is displayed and the access to the system is restricted.

course file: if it exists the file is deleted. If it does not exist, it gives an error message.

Passes control to:

/home/ecrs/public_html/supervisor.html

/home/ecrs/tcls/sup_seecourse.tcl

Inputs:

- supervisor login name passed as argument
- course code passed as argument
- student type to be listed passed as argument
- fields to be listed passed as argument
- sort key for the list passed as argument
- course information read from */home/ecrs/courses /xxx.crs*
- student information read from multiple student file in */home/ecrs/students/*

Checks:

course code: Gives an error message if the course code is not entered.

student file: If the program cannot find the student file, it gives an error message.

student type: By checking the student type, the program creates different lists for different student types.

Passes control to:

The program is used to generate the course lists which are then printed and posted on bulletin boards. Therefore, no links have been added to the page.

See figure:

Figure G.20

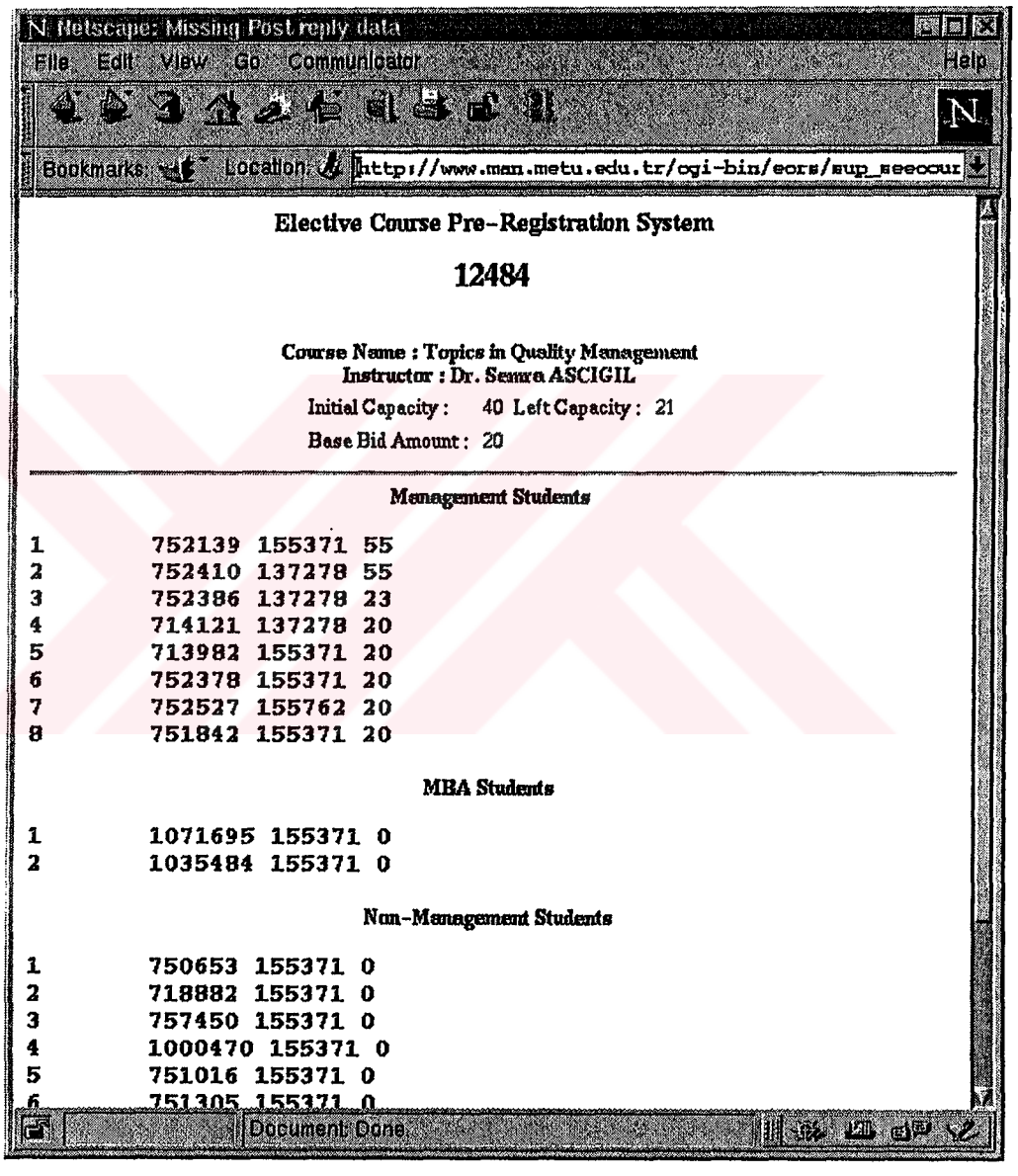


Figure G.20 - Full List of a Course

/home/ecrs/tcls/popularities.tcl

Inputs:

- course information read from multiple files in */home/ecrs/courses/xxx.crs*

Passes control to:

/home/ecrs/public_html/supervisor.html

/home/ecrs/tcls/sup_addlist.tcl

Inputs:

- supervisor ID number passed as argument
- name of the comma separated student list file passed as argument
- student information read from the comma separated student list file

Passes control to:

/home/ecrs/public_html/supervisor.html