

ELECTRA, A MULTIFUNCTIONAL EDUCATIONAL WEB PORTAL FOR TECHNOLOGICAL EDUCATION INSTITUTE DEPARTMENTS

Panagiotis Kenterlis
Department of Electronic Computer Systems
Technological Education Institute of Piraeus
P. Ralli & Thivon 250, 12244 Egaleo, Greece
P.Kenterlis@mprolab.teipir.gr

ABSTRACT

Information and knowledge in the years that are and still to follow will change the face of education from its classical view to a new and hopefully improved part of technological society. In the Greek educational system today, Technological Education Institutes (T.E.I.s) provide their students with sufficient insight on various fields of technology and science.

A considerable percentage of T.E.I. students even before graduation join the work force or live far from the institute's installations and are forced to travel long distances every day. These students face difficulties acquiring information required for their studies, since regular attendance is required. Academics on their behalf are troubled by the incommunicableness with students and the inability to provide teaching material or other information. Additionally, module registrations are a major burden both for academics wishing to justly allow student attendance, as well as for students that are required to be present at the institute for a number of hours to make sure they are placed at a module session that will fit their schedule.

In this paper, a multifunctional content management web application, project name "*Electra*", is presented. *Electra* is being used for a long period of time by the Microprocessors Laboratory and the department of Electronic Computer Systems at T.E.I. of Piraeus, Greece, allowing students and academic staff to communicate together in a uniform manner and access information resources at any time and from any place.

KEYWORDS

Web application, education portal, course management system

I. INTRODUCTION

From the classical education of ancient Greeks to the current state of Modern Greek society little has remained the same. The anthropocentric philosophical search for answers has been subsided by technological advancements and the quest for machine-centric knowledge. The prominence of the Internet and networked computing has driven much of humanity's acquired knowledge outside of thick library walls and into fancy web resources, thus rendering them more publicly available. Nowadays, many universities worldwide provide their entire curriculum freely over the internet, as well as long-distance learning courses.

The highest form of education in Greece is divided in two sectors, each one represented by its own authority. Universities pursue the advancement of science through research, while Technological Education Institutes (T.E.I.s) are primarily oriented in training students to become fully qualified professionals in the ever-changing field of technology and science. The most striking difference in the curriculums of similar departments of universities and T.E.I.s, is the presence of mandatory laboratory sessions within nearly each taught course module of the latter. Laboratory sessions are a major prospect in practical training of students over the classical theoretical only approach of universities, allowing the growth of experience through guided experimentation and module assignments.

Attributed to the level of education and training provided, a considerable percentage of T.E.I. students even before graduation are absorbed by the work industry, with job positions ranging from entry-level up to the top of the hierarchy. Additionally, a considerable percentage of T.E.I. students live far from the institute's main installations, therefore spending weekly a significant amount of time in transit, whereas this time could have been used for studying or socializing. Due to the limited physical presence at the institute, these students face difficulties acquiring information required for their studies and for their everyday life to function as members of the academic society. Academic staff on its behalf is troubled by the incommunicableness with students and the inability to provide teaching material, as well as any other course module or department related information. The lack of a uniform approach to communication is disappointing. With mass media such as radio and television, there is a requirement for continuous or time specific tracking, which can not be applied in the above cases. Contrariwise, there is need for a centralized, always available, user-centric customizable information repository that can be practical and fully featured to accommodate the needs described above. The internet is the best candidate where a solution to the described problems should be looked for. The development of such work is presented in this paper.

II. PROJECT OVERVIEW

Project *Electra* (from this point on simply referred to as *Electra* or *system*) is a complete course management web application designed to manage user and course module information. *Electra* is based on open-source software which significantly reduces installation and maintenance cost and also allows for better development support by a group of thousand code developers around the globe. The PHP (HyperText Pre-Processor) script programming language [1], has been used to develop all data processing algorithms and functions to produce the XHTML pages [2] served to the visiting users. The MySQL database server software [3] has been used to provide the required underlying facilities for storing, retrieving and processing data.

Features embedded in *Electra* enhance the staff-to-student and student-to-student communication aiming in better organization and homogeneous presentation of information and services offered to system users. *Electra* can simultaneously support many academic departments using the same hardware and software installations. Each department has its own course modules, split in theory and laboratory modules, and its own system users. System users are divided in two categories, the module administrators (*ModAdmins*) which consist mostly of academic teaching staff, and the students. A *ModAdmin* can be responsible for many modules, and a student can be registered to many modules. Even the *system* itself is a module (with a unique module code) and its associated *ModAdmins* are the system administrators (*SysAdmins*) which have full access in other departments and modules. Additionally, each department has a module code associated with its secretarial office. All department users can access information related to modules belonging to their department and the global *Electra* module.

For every course module, *Electra* includes features such as:

- Unique and department-wide user accounts for multiple web services.
- Electronic registration of students in modules and selection of attending sessions.
- Administration of electronic registrations by the *ModAdmins* and placement of registered students in module sessions according to student preferences, session capacity and demand for specific sessions.

- Facilities for electronic mutual exchange of assigned module session for requesting registered module students.
- Assignment of students in study/work groups per module session.
- Electronically posting and reading announcements.
- Publication of document or binary files with module related content.
- Publication of hyperlinks to information resources located on the internet.
- System wide and per module event calendar.
- Creation and processing of electronic records of student attendances.
- Electronic submission of student assignments with grading and feedback performed by the *ModAdmins*.
- Exchange of private messages between system users.
- Electronic discussions section with system wide and per module forums.
- Electronic real-time interactive chat room for system users only.
- Electronic message (email) notification of students for newly published module related information.

These features are analyzed in more detail in the following subsections.

A. USER AUTHENTICATION & AUTHORIZATION

For the purpose of user *authentication, authorization and access control* (AAA), whether being a student or a *ModAdmin*, a powerful centralized user database is used. This database is implemented using Lightweight Directory Access Protocol (LDAP) [4] facilities located and centrally administrated by the Institute's Network Operation Center (NOC). The LDAP server is responsible for supporting the *unified user account authentication model* (U²A²M) introduced by NOC staff and the department of Electronic Computer Systems. The U²A²M allows for authentication of users on services provided by the institute, with the use of only one set of user credentials. The user's password is neither stored, nor administrated locally at each service site.

Users connecting to *Electra* must enter their assigned U²A²M credentials to the login page form to gain access to the rest of the site. These credentials are securely authenticated against the LDAP server (Figure 1), which returns a success/fail response.

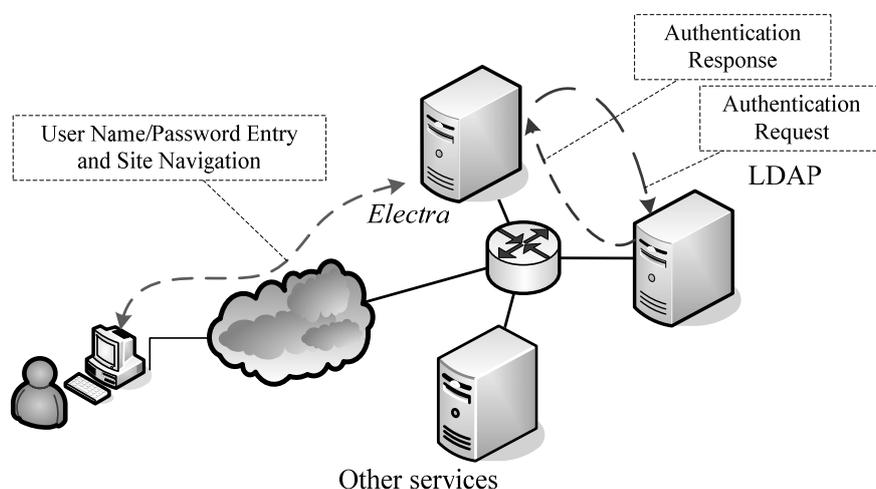


Figure 1 Site login and user authentication

Once the user is successfully authenticated, he/she is authorized and identified against a local database which holds the student list and module administrator specific user details. Other tables in the local database store information concerning the user's registered or administrated modules, thus allowing the system to provide dynamically created custom pages according to the user's access rights. The same pages offer an entirely different view of information to different users, depending on their relation to the content served by that page.

B. STUDENT REGISTRATIONS

For the most part of laboratory modules, a First-Come First-Server strategy has been adopted over the past 10-20 years. This strategy states that at the start of each semester, the first students to enter the laboratory doors will certainly be placed in their preferred laboratory session by handwriting their names (and probably the names of others) on a piece of paper. However, this strategy is entirely unfair for students attending the module's typical semester since they can be supplanted from their rightfully laboratory session placements by students of greater attending semester, which may have previously attended the same laboratory or not. As indicated, the registrations period is one of great discomfort for the students that have to be physically present in the institute for a number of hours or days to succeed at being registered in a laboratory module session that will best suit their weekly schedule. Being informed on when each laboratory would conduct its short period registrations, usually lasting a few hours, and managing time to participate in the registrations would lead to much anxiety. Originally, the development of *Electra* targeted this injustice and troubling period, but soon evolved into a larger project.

It is the responsibility of *ModAdmins*, before the start of each semester to evaluate the number of students that are likely to be registered in the laboratory and thus create the corresponding number of attending sessions that will hold these students. Typically, the number of sessions does not fluctuate more than one or two sessions compared to that of previous semesters. Every session has specific properties, such as day/time of session conduct, session capacity in number of students, number of work groups, etc. Having created the electronic equivalent of a laboratory session using *Electra's* web tools, the module administrator needs to decide when to open the module's registration period to allow students to choose the module sessions that will fit in their attendance schedule. The electronic registrations period can last from a few hours up to a few days, allowing more students to register the module and enter their session preferences at their leisure, using any computer that is connected to the internet. To overcome any problems in the case of a large number of students wishing to be placed in a specific session, students are required to choose among a variable number of sessions in order of preference. Within the open registrations period, the students can change their preferences as many times as they wish, based on the likelihood to enter a selected session, making use of information provided by the processing of live data by the registrations scenarios. If a student cannot be placed in his/her first session of choice, placement will be tested against the next session in the order of preference. Exploiting the student's known module registrations and complete session placements, the process of placing the student to a session that conflicts in time with another module is not permitted.

After the registration period has ended, the *ModAdmins* can choose whether to perform session placement manually or by using automatic placement algorithms. In both cases, the *ModAdmin* has to specify which one of the many pre-written registration scenarios to use for ordering, filtering and placing students in sessions. These scenarios can process registration requests on various parameters such as student's attending semester, previous attendance, previous grade (requires some knowledge of student grades not directly stored in the *system*),

time of registration request submission (not recommended). Additionally, *Electra* can disallow students from selecting sessions that have exceeded maximum capacity thus operating in a First-Come First-Served manner. However, this should only be used in modules where the expected number of students and available sessions will not cause a problem to arise.

Using the manual placement tools, for each module session the *ModAdmin* is presented with a series of lists, tables and information screens as in Figure 2. Selecting one of the established module sessions, a list of students that have requested this session as their first choice and the rest of their preferred sessions appears. This list is sorted and presented according to the configured registration scenario as an indicative placement solution. Students exceeding the session’s maximum capacity will appear in red colored rows at the bottom of the list. The *ModAdmin*, depending on the condition of other sessions and the proposed placement augmented by the registration scenario, can choose to place students either on one of their preferred sessions, or on another one. This can be performed individually or for several selected students in the list. In the event of the number of students registering with the module exceeds the expected figure, more sessions can be readily created during open registrations period, allowing students to change their preferences should they wish to.

Σπουδαστές χωρίς Ανάθεση					
A/A	Όνοματεπώνυμο	Εξάμηνο	Επιλ.	Πρώτη Επιλογή	Δεύτερη Επιλογή
<input type="checkbox"/>	1. ▲ ΑΛΕΞΑΝΔΡΗΣ ΑΛΕΞΑΝΔΡΟΣ	11	1 2	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	2. ▲ ΑΣΦΗΣ ΔΗΜΗΤΡΙΟΣ	7	1 2	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	3. ○ ΒΑΣΙΛΕΙΟΥ ΧΡΗΣΤΟΣ	7	1 2	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	4. ▲ ΒΛΑΧΟΣ ΠΕΤΡΟΣ	7	1 2	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	5. ▲ ΒΛΑΧΟΣ ΣΤΑΥΡΟΣ	9	1 2 X	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	6. ○ ΒΡΕΚΟ ΣΚΕΛΧΙΜ	8	1 2	Τετάρτη 20:00-22:00	Δευτέρα 08:00-10:00
<input type="checkbox"/>	7. ○ ΠΑΠΑΔΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ	9	1 2	Τετάρτη 20:00-22:00	Πέμπτη 20:00-22:00
<input type="checkbox"/>	16. ○ ΤΡΙΜΠΟΣ ΚΑΛΑΜΠΟΚΗΣ ΝΙΚΟΛΑΟΣ	7	Μαζική Ανάθεση	Μη Ολοκληρωμένη Εγγραφή	Μη Ολοκληρωμένη Εγγραφή
<input type="checkbox"/>	18. ○ ΤΡΙΤΟΣ ΑΠΟΣΤΟΛΟΣ-ΑΝΑΡΓΥΡΟΣ	9	1 2	Πέμπτη 18:00-20:00	Δευτέρα 12:00-14:00

Καμία Επιλογή Επιλογή όλων

Μαζική Ανάθεση:
 Μαζική Αποστολή Μηνύματος:

Σπουδαστές Τυπικού Εξαμήνου: 0
 Σπουδαστές Μικρότερου Εξαμήνου: 0
 Σπουδαστές Μεγαλύτερου Εξαμήνου: 5
 Σπουδαστές Μεγαλύτερου Εξαμήνου (Επαν.): 13
 Σπουδαστές με Απαλλαγή λόγω Βαθμολογίας: 0

Figure 2 Student registrations in a module session with first two student preferred sessions

Using the automatic session placement, *Electra* will attempt to assign as many students as possible in their first selected session, limited by the number of attending students per session. The algorithm implemented resembles the practical way of thinking and follows next:

1. Create session container objects with all their respective individual properties, i.e. container maximum capacity in students.
2. For each student, check all of their preferred sessions and mark those allowed to be used for processing. Selected module sessions that coincide with sessions of other modules that the student has been already placed at are marked as “Don’t Use” and are not considered in further processing steps.
3. Fill session containers with students ordered according to the module’s registration scenario. Higher priority students are placed at the bottom of the container, while students with lesser priority at the top.

4. Check for manual placements and move the respective students at the bottom of their manually placed containers. Mark these students with a “Don’t Move” flag.
5. For each container remove students exceeding container’s maximum student capacity place them in the “*unassigned students*” container.
6. Check all containers for available space and establish a counter for each one with the number of students that can be fitted.
7. Randomly check the “*unassigned students*” container to see if their second choice of session lies inside the session containers with available space. If true, move the student at the top of the container and decrement its available space counter. Perform this step until all available space is filled or all students with their second choice in these containers have been moved out of the “*unassigned students*” container.
8. If there is still available space and unassigned students, then perform the previous step for the unassigned students’ third or further choice of session in order of preference (if more than two session choices were required for registration).
9. If all students have been placed, terminate the process.
10. If no space is available and students remain unassigned, terminate the process and request from the *ModAdmin* to resolve the issue (e.g. increase the capacity of available sessions or create new ones).
11. If there is still available space and unassigned students that had not selected any of the sessions where available space is, then perform the following steps.
12. Pick one student from the “unassigned students” container. For this student’s first session of choice check from top to bottom whether there is a student with a second choice at any of the containers with available space. If such a student exists and has no “Don’t Move” flag, then move this student at the session of his second choice and the unassigned student in his place.
13. Continue the previous step until all unassigned students have been placed or processed for their first choice, or all available space is filled.
14. If space is still available and there are unassigned students, then perform the previous two steps for the rest of the unassigned students’ selected sessions in order of preference. Continue until no space is available or all preferred sessions processed.
15. If at this point there are still some unassigned students, then expect from the *ModAdmin* to resolve the issue.

Following the algorithmic steps described above, the computer-aided student placement program allows for an 80-85% of students to be placed at their first session of choice while being just in the process of placement with students of higher priority being served first and with little chance of being moved to an alternate session. The rest of the students with lesser priority are either placed at their second or at a worst-case scenario they are placed at one of their remaining choices (where applicable). If there is a session space issue, students can be contacted to change their choice of preferred sessions, in order to move to a newly created session. Experience so far has proven that through efficient communication there are a very small number of students that will be not fully satisfied.

C. ACCESS TO MODULE RELATED INFORMATION

It is important that users waste no time searching for information about their modules. For this purpose, the *system*’s main page is dynamically generated to include only modules and related information concerning the authorized system user. Virtually every page presents to the user a custom set of dynamic information. Students have little control over the module information, practically limited to viewing, while *ModAdmins* have full control of any content related to their modules, and *SysAdmins* have full site-wide access and control.

The main page contains the following sections:

- Electra's own notice board, hyperlink and file repositories.
- The department's secretarial office notice board and linked file repositories.
- A list of public forums.
- Theory modules with each containing the following subsections and links:
 - Fast access and title listing of module's notice board items with new and unread notices being graphically marked as "new".
 - Fast access to the module's hyperlink repository.
 - Fast access to the module's file repository.
 - Link access to the list of sessions and the students requesting placement to each session (during registrations period only).
 - Link access to the finalized list of sessions with their placed students (after registrations period has ended).
 - Fast link to student assignment submission page (for students).
 - Link access to submitted student assignments' lists (for ModAdmins).
 - Link to module's linked forum. If unread forum posts are present, the link is also graphically marked as "new".
- Laboratory modules with content as listed in theory modules above.
- Scheduled *system* and registered/administrated module events.

Module notice board holds chronologically sorted notices each containing the following information:

- Short descriptive title of notice.
- The full text body of the notice.
- List of module files linked to the notice.
- Publication timestamp.
- Name of publishing author (*ModAdmin*) and direct link to his/her user profile.

Module files repository presents the files related to the module in chronological or content category order. Each file is presented with the following information:

- Short descriptive title of file.
- A full description of the file's contents (not mandatory).
- A hyperlink to the download management script for this file.
- The file size in human readable format.
- File publication (upload) timestamp.
- Publishing user (*ModAdmin*) and direct link to his/her user profile.

Hyperlink repository contains a list of URLs to module information found on the internet, and are either sorted by time of publication or content category. Information presented includes a short link title and the link to the external site.

Student access to all of the above sources of information is monitored by the *system* for administrative purposes. Statistic reports are automatically generated by the *system* for use by the *ModAdmins*. It is therefore possible to track whether a certain student has accessed any of these resources and when. When a resource is published, it will appear to each individual student graphically marked as "new" and this mark will be removed only when the resource is accessed by that student (downloaded for files, read for notices, visited for hyperlinks).

D. ATTENDANCE RECORDS, STUDENT GROUPING AND ASSIGNMENT SUBMISSION

Since attending a laboratory session is mandatory, recording student attendance is essential. *Electra* allows the entry and processing of per-session attendance records in a uniform manner for all teaching staff. The need for paper records that are often lost or accidentally destroyed is kept to a minimum or eliminated. Records are accessible over the internet by *ModAdmins* at any time. Each attendance recorded holds a timestamp and can be linked to an assignment or a taught session (week of conduct).

Students within a session can be divided by the *ModAdmins* in attending or assignment groups by means of incorporated web tools, allowing for easy handling of groups at any point in time. After the groups have been formed, they are used for the display of sorted student lists throughout the rest of the module session pages. PDF or Excel readable files holding the module's registered student list with assigned group and session information can be generated by the *system*.

To further allow the reduction of paper waste and to provide *ModAdmins* with more control over the submission of student assignments, a special *system* feature has been developed. Through the use of the *student assignments* facilities, the mediation of the department's secretarial office, as practiced in most British Universities, is avoided. Additionally, the unregulated and unscheduled submission of assignments via email to specific members of the *ModAdmins* group can be banned, thus allowing all of the *ModAdmins* to have access to the submitted assignments. Students are required to submit their assignment documents as one file (possibly a compressed file tree) to a per-student *assignment slot*. The assignment slot is associated to a specific assignment title or taught session. The assignment slot is left open for submission for a period of time defined by the *ModAdmins* and can be linked to the attendance records for the student. Examples include a period of 7 days after an attendance has been recorded for a taught session, or a specific date and time for each assignment. Beyond this open submission period and with an accuracy of one second, the slot is automatically closed by the *system* and no further submission can be allowed or processed. A *ModAdmin* can choose to defer a student's assignment slot closure by crediting the student with a sufficient additional number of days for the completion of the assignment. Before or after slot closure, *ModAdmins* can examine the submitted assignment and proceed to one or more of the following actions:

- Provide the student with feedback on the assignment.
- Request a resubmission of the assignment within the open submission period.
- Request a resubmission of the assignment after the slot has closed and provide the student with sufficient time extension.
- Grade the assignment in a range of 0 to 100% (grade is not visible to the student).
- Create a compressed file with all of the student's assignments.
- Create a compressed file of selected assignments from different students.
- Generate a report of student assignments containing submission dates and grades.

E. USER-TO-USER AND INFORMATION COMMUNICATION

Electra has been especially designed to augment communication between students and teaching staff targeting to the advancement of the education process. Students can direct questions, requests for teaching material, etc, to their teachers, while teachers can talk to individual students to respond to questions, give feedback on module progress, etc. System users can communicate with each other in three ways, each one with a specific purpose. First,

users can exchange private messages using the embedded messaging facility. *Private messages* are stored encrypted inside the local database and can only be decrypted by the recipient user. Private messages can be addressed to any individual user, or a group of users, as done in email messaging. Second, embedded forums facilities are available for users to exchange ideas, opinions and other information. Forums can be related to specific course modules or be independent, owned by the *system* itself. Module forums are primarily available to registered module students, though non-registered students can post their messages if required. *ModAdmins* in the case of forums related to their administrated modules also function as moderators and have complete control over every posted message. Third, an interactive AJAX based [5] chat service has been incorporated that allows system users to talk to each other in real-time either in a public channel or in private channels. All text message facilities described above are semantically enriched with graphical representation of emotions (emoticons).

A more aggressive method of communicating information supported by *Electra* is the notification of students for newly published module related content, such as notices and files, by means of email notification messages. *ModAdmins* can choose to notify the students for which the information is intended when processing new content publication. Individual students not wishing to receive notification emails can override this function from their personal profile settings.

F. OTHER FEATURES

An number of advanced features are incorporated in *Electra*. RSS (Rich Site Summary or Really Simple Syndication) [6] which is based on XML (eXtensible Markup Language) [7] is employed to communicate summaries of available module content, such as new published notice board items and module files. Each course module has its unique dynamic RSS feed URL. Users must have an RSS client or other RSS-aware software installed in their personal computers. Currently a small number of applications have been home grown to take advantage of the offered RSS feeds. Such a Java [8] application has been developed by two students [9] of the department of Electronic Computer Systems as their degree project. This application designed for use by most modern mobile phone devices can talk to *Electra* either over Bluetooth, GPRS or 3G networks, download and present the phone user with information in the RSS feed concerning the modules of interest. Personal Digital Assistants (PDAs) having much higher processing and storage capabilities than mobile phones, can directly talk to *Electra* over any means of internet connection (Ethernet, WiFi, Bluetooth, dial-up modem, etc) and download an RSS feed (if a binary RSS client is installed). With the increasing use of internet connected mobile devices, it is technically realizable for students on the move to be constantly informed on issues related to the modules of their interest.

Recently, a site collaboration framework has been developed under PHP source code that allows external sites to talk directly to *Electra* and extract information related to modules. These sites are largely the web sites for these modules wishing to make use of registered user information as well as published module content. For security reasons, this information is limited and made available only to sites that authenticate themselves and all communication is performed over secure HTTP (HTTPS) data channels. Three such sites have so far been designed also implementing the U²A²M scheme thus allowing site access only to registered module users. The main advantage of these implementations is that there is no requirement for local databases containing user accounts for AAA or lists of registered module users. All critical information and processing are handled by the central LDAP server (user authentication) or *Electra* (authorization and access control).

III. CONCLUSIONS

Through the work presented in this paper it has become possible for the academics to effortlessly publish educational material, post notices, talk to students and respond to questions both by private messaging or publicly using embedded forums. Academics can extract useful statistic reports on their courses and student participation. Students on their part are given access to information related only to their registered course modules, thus minimizing time required to find information of their concern. Several functionalities have also been implemented that attempt to fulfil special needs of T.E.I. academic programs. In particular, this work has allowed the remote registration and placement of students in laboratory module sessions, without the requirement for physical presence in the institute. Therefore, it has become possible to remove any pressure and frustration originated by the limited number of sessions, the overwhelming number of students and the established unfair First-Come-First-Served registration practices. Algorithms and facilities developed and used to assign students to sessions according to configurable scenarios have allowed an average of 80-85% students satisfactorily assigned to their first session in order of preference. Additional developed facilities allow the reduction of paper waste by enabling students and academics to keep attendance records as well as sending and receiving any module related assignments in electronic format. Students on the move can now have access to information related to their studies by taking advantage of the *system's* features.

Overall, this work has contributed much to the development of a new information era for the department of Electronic Computer Systems at T.E.I. of Piraeus.

IV. REFERENCES

- [1] PHP HyperText Pre-Processor, Available at: <http://www.php.net/>
- [2] XHTML - The eXtensible HyperText Markup Language, Standard published by the W3C (World Wide Web Consortium), Available at: <http://www.w3.org/TR/xhtml1/>
- [3] MySQL Open-Source Database Server, Available at: <http://www.mysql.com/>
- [4] OpenLDAP Open-Source implementation of the Lightweight Directory Access Protocol, Available at: <http://www.openldap.org/>
- [5] AJAX - Asynchronous JavaScripting and XML, <http://www.ajaxmatters.com/>
- [6] RSS - Really Simple Syndication, <http://blogs.law.harvard.edu/tech/rss>
- [7] XML - The eXtensible Markup Language, Standard published by the W3C (World Wide Web Consortium), Available at: <http://www.w3.org/XML/>
- [8] J2ME - Java Micro Edition, Sun Microsystems, Inc., Available at: <http://java.sun.com/javame/index.jsp>
- [9] K. Kalotychos, C. Gkikas, P. Kenterlis, J. Ellinas, "Autonomous mobile application enabling direct briefing via RSS by networks of mobile telecommunications (GPRS, 3G)", Accepted for presentation at 1st International Scientific Conference eRA, Tripoli, Greece, 16-17 Sept. 2006.

Panagiotis Kenterlis received his first Degree in Electronic Computer Systems engineering from the Technological Education Institute of Piraeus, Greece, in 2001. He received his first M.Sc Degree in Digital Electronics from the Universities of Brighton and Sussex, UK, in 2002. He received his second MSc Degree in Data Communication Systems from the joint MS.c course of T.E.I. of Athens, Greece, and the University of Brunel, UK, in 2004. He is currently pursuing his Ph.D. Degree in the field of VLSI Testing at Athens University, Department of Informatics and Telecommunications. Since 2002, he has been with Technological Education Institute of Piraeus, Department of Electronic Computer Systems, Greece, where he is currently a Laboratory Teaching Assistant and research associate. His research interests are in VLSI testing, microprocessor systems design, embedded systems, and web applications development.