Thesis Dissertation

A tool for student preference registration of restricted elective courses of the Computer Science Department of the University of Cyprus

Demetris Shimitras

UNIVERSITY OF CYPRUS

COMPUTER SCIENCE DEPARTMENT

May 2024

UNIVERSITY OF CYPRUS

COMPUTER SCIENCE DEPARTMENT

A tool for student preference registration of restricted elective courses of the Computer Science Department of the University of Cyprus

Demetris Shimitras

Supervisor

Dr Elpida Keravnou-Papailiou

The Thesis Dissertation was submitted in partial fulfilment of the requirements for the award of the degree of Bachelor in Computer Science at the University of Cyprus

May 2024

Acknowledgments

I want to explicitly thank my thesis supervisor, Professor Elpida Keravnou-Papailiou, for her guidance and support during the completion of my Bachelor thesis. I am grateful to this opportunity and help for reaching the end of this academic journey.

I would also like to thank my family for their support and motivation they have been giving me during this time.

Abstract in English

The aim of this project is to describe the implementation of a solution of a persisting problem that is being faced during the course registration period of undergraduate students of the Department of Computer Science in regard to the restricted elective courses. The issues specifically being faced are that students may wish to attend particular restricted elective courses, to qualify for a specialisation or simply out of interest in the course, for which not enough available slots are offered to satisfy all or an accepted percentage of the demanding students. This results in students registering for other courses that they are not interested in and not being able to achieve their goals upon their graduation.

The concept of this solution, is the implementation of a new pre-registration system that will be provided to the students, to which they will input their preferences of the restricted elective courses they would like to attend in the next few semesters. This could give an insight of what are the most demanded courses by the students, for the administration to decide which would be more fitting to be offered then. Ideally, the system could also prepare a proposed course schedule for the next semesters, based on the demand and overview of the most preferred courses.

Table of Contents

Chapter 1

Introduction

1.1 Problem - Motivation and Goal

Undergraduate students of the Department of Computer Science race to select the restricted elective courses they prefer when their registration is opened up. Due to low seat offerings of these courses and the priority of students' registration based on their ECTS, students become unable to register to the ones they want.

Students are forced to select other restricted-choice courses which would not benefit their goals, including taking specific courses they need in order to qualify for a specialty upon graduating.

1.2 Solution concept

A new 'pre-registration' web application system, as with current Bannerweb, in which students would express their preferences of the restricted-choice courses they want to attend in later semesters, in advance.

This will give insight to which courses are more demanded by the students and which are not, thus ensuring more students to be satisfied with their selection in future semesters. It can also enable better scheduling of offering courses, like being offered in a different semester than usual, in both semesters in an academic year, or two classes offered in one semester with high demand, if other conditions also make it possible.

1.3 Scope

The problem research shows different aspects of the restricted elective course registration. The current thesis does not investigate all of them. As a consequence, the implementation of a new pre-registration system as a solution, so far, does not solve all of them.

Overall, the scope of the current thesis covers the courses preferences input by the students for the few next semesters and a simple report with the aggregated preferences for the semesters, along with a list of the students for every preference, which are the most significant features in the implementation.

Chapter 2

Requirements Analysis

2.1 Development process

The process followed for the development of this solution started with the research and design phase. In this phase, the problem that the solution intends to solve was investigated. This research was initiated by a meeting with Dr Elpida Keravnou-Papailiou, during which the most important problems and requirements were discussed. The analysis of this discussion revealed the business goals, as well as the user goals.

Once the business and user goals were defined, there was a design phase which intended to create the concept of the system, the different screens and functionality that would be included in each. The user experience was also taken into account.

The design phase was followed by the implementation in increments. At first, the main functions and requirements were implemented. After a few more functionalities were completed in other increments, the developed solution was reviewed in order to catch points for improvement. Then, based on that review, some user experience details were improved, as well as other complementary functions were added to the solution. Lastly, any points that could not be implemented have been noted as suggestions for future work.

2.2 Requirements

The following requirements are the result of the analysis for the described problem, covering the main business and user goals.

2.2.1 Generic requirements

- The system is implemented as a web-based application.
- A user can have the role of a student or an admin (faculty member).
- A user can register to, log in and log out from the website.
- A user can view and change personal details.

2.2.2 Student role

- A user can view courses and related information.
- A user can add, view and edit course preferences for semesters.
    - Adding and editing preferences are only allowed for future semesters.
    - Up to five restricted elective courses per semester.
    - Preference priority of 1 to 5, with 1 indicating the highest priority.
    - Provide a reason for the selected course as a preference.
- A user can update their personal information and academic details.

2.2.3 Admin role

- A user can view the student preferences for restricted elective courses:
    - Total preferences for a specific semester.
    - Total preferences for a specific course.
    - Students’ information of a preference.
- A user can get insights for the registered course preferences of the students for the next semesters.
- A user can add, view and edit information about courses.
- A user can add, view and edit information about semesters.

2.2.4 Instructor role

- A user can view information about courses.
- A user can edit information about their courses only.
- A user can view information about semesters.

2.3 Constraints

During the analysis of the needs for the purpose of this solution, and during the development of this application, some constraints were encountered, which are mainly related to the retrieval of real user data.

2.3.1 User authentication

Since the application is account-based and each user can interact with their own account and data, there was a need for an authentication mechanism for logging into the system and retrieving the user's data. Initially, the idea for this mechanism was to use one that could authenticate the user's credentials with the University's or Department's internal user data. This would provide a more secure authentication, since no password would be stored locally in the system. Additionally, it would offer a centralised authentication process for the users to log in with the same credentials used in existing University systems. This would also avoid the need of a registration process from the users.

2.3.2 User data

Another constraint faced during the development of this project is about the user data. Some information regarding users' data, such as personal and academic details, were needed for some specific functionalities of the system. Therefore, the retrieval of this information from the university database would be helpful, making sure that the data is always accurate and up to date. However, since it is a great concern of allowing access of sensitive data to external systems, the solution is currently built without the integrated data retrieval. Therefore, for the purposes of the developed tool, users are required to manually provide this information themselves.

Chapter 3

System Specifications

3.1 Architecture

The architecture of the web application is based on a server-client model. A backend application (the server) is responsible for managing most of the business logic, processing data by interacting with a database to retrieve, store and update it. The backend server solely provides APIs to handle client requests and does not serve web pages to the client.

The frontend application (the client), on the other hand, is responsible for serving the interface layer and presenting the content to the users. It is built as a Single-Page Application[6] (SPA) and it communicates with the backend via the APIs that are exposed, to retrieve data and perform actions. It can also generally handle aspects of the business logic that focuses on visual presentation and user interaction before the data is persisted to the database through the backend.

3.2 Technology and tools

It had been decided for the web application to be developed using the MEVN stack, which is a modern and evolving open-source JavaScript software stack for building powerful and dynamic web applications. **MEVN** stands for ***M**ongoDB*[1], ***E**xpress.js*[2], ***V**ue.js*[3] and ***N**ode.js*[4].

3.2.1 MongoDB

The database layer of the web application uses MongoDB, a NoSQL database for flexible and scalable data storing and management. It stores data as documents in JSON-like format, in collections. MongoDB is schema-less which enables for flexible document structures in the same collection, adding, omitting or modifying fields and their type without affecting other already stored documents.

3.2.2 Express.js

Express.js is a minimal and flexible Node.js web application framework designed for building robust and scalable server-side applications and APIs. It provides a set of features for handling HTTP requests, routing, middleware and serving static files. Express routing framework makes use of middleware functions when processing requests which can perform other tasks before reaching the route handlers. A wide range of third-party middleware modules can be installed via npm[5], which can extend the Express application with additional functionality.

3.2.3 Vue.js

Vue.js is a progressive JavaScript framework used for building user interfaces and single-page applications (SPAs). It provides a template syntax to bind data to the DOM (Document Object Model) with simple and intuitive directives that offer easy manipulation of HTML elements with minimal boilerplate code, and follows a component-based architecture for reusable and encapsulated components. A Vue component is made of its template, the HTML part declaratively binding with the component's data, the script section where JavaScript logic can be applied to the instance's data. Lastly, the style section allows to add CSS design to the component's HTML template. Vue's reactivity system tracks dependencies between data and the interface, which allows for automatic update of the elements' data. An ecosystem of external libraries and plugins are available to use seamlessly with Vue for better overall development of a frontend application.

3.2.4 Node.js - NPM (Node Package Manager)

Node.js is a cross-platform runtime environment for executing JavaScript code outside of a web browser and allows developers to create servers, web apps, command line tools and scripts. It is an event-driven, asynchronous, non-blocking I/O runtime environment that makes it lightweight and efficient, suitable for building real-time applications, APIs, and server-side applications. Node.js uses CommonJS modules for code structuring, which encapsulate functionality and can be imported and exported throughout the application for modular and reusable code.

NPM is the default package manager for Node.js, providing a vast number of open-source libraries and modules easily integrated into the applications, simplifying dependency management, version control, and code reuse for more efficient development.

3.2.5 Libraries

During the development of this web application as a solution of the problem, along with the previous technologies mentioned earlier, many libraries and plugins, installed through NPM, have been leveraged in both backend and frontend parts to ease implementation, as well as provide better productivity, efficiency and a higher quality of functionality and experience for the users. Some libraries used that had significant contribution to the development include:

For backend:

- **body-parser**[7]: an Express middleware that parses the body of incoming requests in a more convenient format, based on user's input, available under 'req.body' property. The format used in this development is JSON.
- **mongoose**[8]: an Object-Data Modeling (ODM) library for MongoDB and Node.js that provides a straightforward schema-based solution for modelling application data. Simplifies interactions with MongoDB databases, allows for schema definition with typed data structures with features for data validation, querying, and middleware hooks.

- **express-session**[9]: an Express.js middleware facilitating session management in web applications, with a variety of options for session storage configuration, cookie management, session expiration and regeneration.
- **passport.js**[10]: Passport.js is an authentication middleware for Node.js applications that provides a flexible and modular approach to implementing authentication strategies. For the implementation, ‘passport-local’ strategy is used, which is designed to authenticate based on a username and password using a local database or data source.
- **pino**[11]: a lightweight and fast logging library for Node.js applications.

For frontend:

- **Vue Router**[12]: the official client-side routing library for Vue.js applications, providing seamless routing capabilities for building single-page applications (SPAs) for dynamic content rendering and navigation without full-page reloads, It provides navigation guards for access control and validations before navigating to a route.
- **Vuex**[13]: a state management library for Vue.js applications for globally managing the state of all components in the application maintained in a centralised store. Changes to the state are performed using synchronous functions called Mutations, whereas asynchronous state operations, called Actions, commit the mutations.
- **Primevue**[14]: a UI component library for Vue.js applications, offering a rich set of customizable and responsive UI components seamlessly integrated into Vue.js projects, built on top of PrimeFaces. Primevue offers forms, data displays, navigations, overlays and more, always opting in for responsiveness. The components are highly customizable, supporting custom themes through CSS and SASS.
- **Bootstrap**[15]: a popular front-end framework for building responsive and mobile-first web applications. Its classes have been widely used for the development of this project.
- **Axios**[16]: a popular JavaScript library used for making HTTP requests from browsers or Node.js applications, with a simple API for asynchronous requests and handling responses. It uses JavaScript promises that allows asynchronous

code, provides error handling and response validation, request and response interceptors for centralised logic and a set of customization request options.

- **Vue-i18n**[17]: a localization library for Vue.js applications, designed to support multi-language content and interface translations.

Chapter 4

Functionality

4.1 Entities

The application as a solution for the registration of restricted elective courses problem has been developed incorporating four (4) primary entities: Student users, Faculty users including the teaching staff but focusing more on the administrative staff, Courses and academic Semesters. For each one of these entities, there exists a corresponding collection in the database. Course and Semester are entities mostly used in relationship with the Student entity to represent the students' courses preferences for a semester.

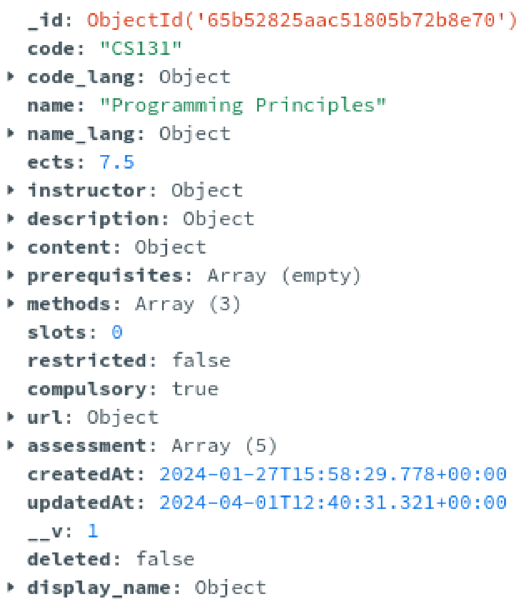
The collections in the database are created with Mongoose by specifying the schema and creating a data model. Mongoose provides various options for schemas. One of the options is to set timestamps, which automatically adds two Date properties, 'createdAt' when a document is first inserted and 'updatedAt' whenever the document is updated. Moreover, the '_id' property is set by default, which is a unique identifier given to a document when inserted, serving as the primary key of a document. The _id property is of type 'ObjectId', a class that represents a 12-byte identifier, to be used as a unique reference for MongoDB collections. Below are described the entities and the schemas defined for the development of this application, with given examples of documents inserted in the database for each one.

4.1.1 Course

The Course entity consists of general information of the courses that are offered by the Computer Science Department, such as the code and name of the course, the content taught for the course, their prerequisite courses, and more.

The schema of the Course collection is presented below (fields not fundamental to the understanding of the entity are omitted for clarity):

- code: String, Unique index, required. Represents the course's code.
- name: String, required. The name of the course.
- ects: Number. Credits earned for succeeding the course.
- description: Object of two strings (English, Greek translations)
- prerequisites: Mongoose Course ObjectIds Array. Prerequisites for the course
- restricted: Boolean. Default: false. Indicates if the course is restricted elective in the curriculum.
- compulsory: Boolean. Default: false. Indicates if the course is compulsory in the curriculum.
- specialisations: Array. Specialisations that can qualify for with the course



```
  _id: ObjectId('65b52825aac51805b72b8e70')
  code: "CS131"
▸ code_lang: Object
  name: "Programming Principles"
▸ name_lang: Object
  ects: 7.5
▸ instructor: Object
▸ description: Object
▸ content: Object
▸ prerequisites: Array (empty)
▸ methods: Array (3)
  slots: 0
  restricted: false
  compulsory: true
▸ url: Object
▸ assessment: Array (5)
  createdAt: 2024-01-27T15:58:29.778+00:00
  updatedAt: 2024-04-01T12:40:31.321+00:00
  __v: 1
  deleted: false
▸ display_name: Object
```

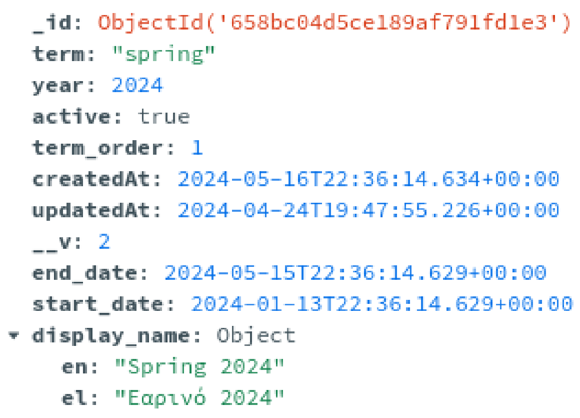
Figure 4-1. Course document example in MongoDB

4.1.2 Semester

Semester entity is a simple data model used to provide a relationship with the students' preferences, that is to set their preferred courses for a specified future semester.

Each semester is defined by its term and the year:

- term: String, required. Specifies the semester term - Spring, Fall, Summer
- year: Number. Year of the semester. Default is set the current year.
- term_order: Number. Semester order in the academic year. Defined by term field
- start_date: Date, required. Specifies the date that the semester begins.
- end_date: Date, required. Specifies the date that the semester ends.
- active: Boolean. Indicates if the semester is currently being progressed through.



```
_id: ObjectId('658bc04d5ce189af791fd1e3')
term: "spring"
year: 2024
active: true
term_order: 1
createdAt: 2024-05-16T22:36:14.634+00:00
updatedAt: 2024-04-24T19:47:55.226+00:00
__v: 2
end_date: 2024-05-15T22:36:14.629+00:00
start_date: 2024-01-13T22:36:14.629+00:00
▾ display_name: Object
    en: "Spring 2024"
    el: "Εαρινό 2024"
```

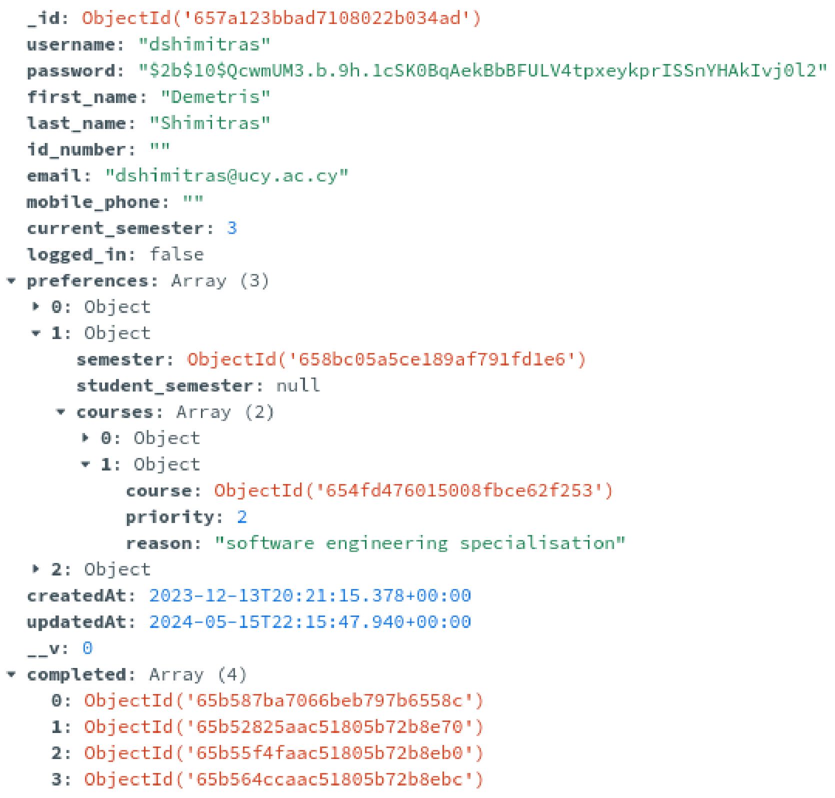
Figure 4-2. Semester document example in MongoDB

4.1.3 Student

Student entity represents the students of the Department. Other than the personal information of a student, the important data that exist in this collection in regard to the purpose of this project are the preferences that a student registers in the system of restricted elective courses they would like to attend in future semesters for their degree. The preferences of a student is a list of items composed of the semester they want to attend some courses in, and the list of those courses.

Another field ‘student_semester’ also exists, which refers to the number of the semester for which the student has set their preferences. It is associated with the current number of the semester that the student is studying in (‘current_semester’ field); currently, there is no existing logic to populate this field. Subsequently, the courses list consists of objects containing the course (as a Mongoose ObjectId) of the course preference, the priority of the preference, that is, the amount of interest the student has for taking the specified course, and a short description of the reason for making this preference select.
The schema of the Student model is presented below:

- username: String, Unique index, required. The username of the student.
- password: String, required. Encrypted string of the student’s password.
- first_name: String, required. Student’s first name.
- last_name: String, required. Student’s last name.
- id_number: String. Student’s ID number.
- email: String, required. Student’s email.
- mobile_phone: String. Student’s mobile phone number.
- current_semester: Number, required. Indicates which semester the student is currently at. (This should be updated by the student at the moment.)
- completed: Mongoose Course ObjectIds Array. Courses that the student has already succeeded. (This should be updated manually by the student at the moment.)
- preferences: Objects Array. Represents the student’s course preferences for the semester.
  - semester: Mongoose Semester ObjectId, required.
  - student_semester: Number. Semester number according to their semester and the semester setting their preferences for (not currently in use)
  - courses: Objects Array. Student’s course preferences for the ‘semester’.
    - course: Mongoose Course ObjectId, required.
    - priority: Number. Indicates level of interest for the preference.
    - reason: String. Descriptive reason for preference selection.
- logged_in: Boolean. Default: false. Login status of user.



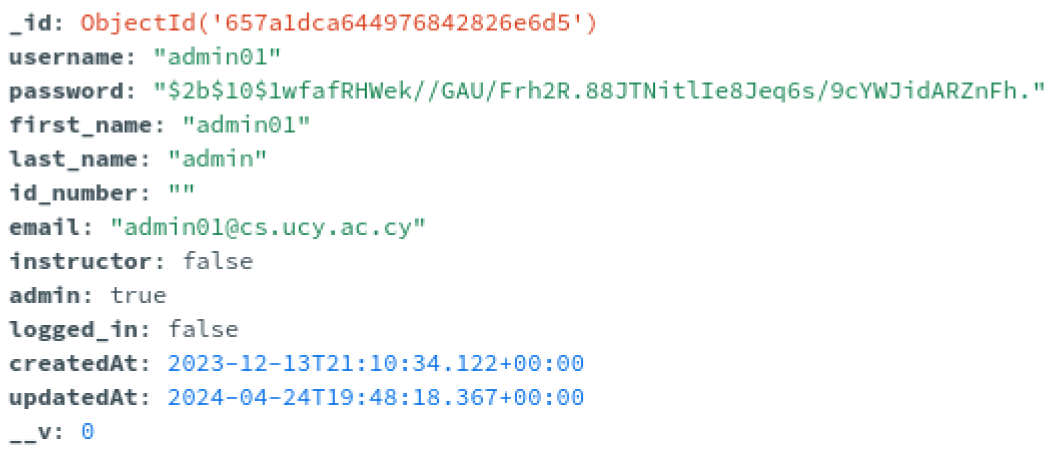
```
  _id: ObjectId('657a123bbad7108022b034ad')
  username: "dshimitras"
  password: "$2b$10$QcwmUM3.b.9h.1cSK0BqAekBbBFULV4tpxeykprISSnYHAkIvj0l2"
  first_name: "Demetris"
  last_name: "Shimitras"
  id_number: ""
  email: "dshimitras@ucy.ac.cy"
  mobile_phone: ""
  current_semester: 3
  logged_in: false
▾ preferences: Array (3)
  ▸ 0: Object
  ▾ 1: Object
      semester: ObjectId('658bc05a5ce189af791fd1e6')
      student_semester: null
    ▾ courses: Array (2)
      ▸ 0: Object
      ▾ 1: Object
          course: ObjectId('654fd476015008fbce62f253')
          priority: 2
          reason: "software engineering specialisation"
  ▸ 2: Object
  createdAt: 2023-12-13T20:21:15.378+00:00
  updatedAt: 2024-05-15T22:15:47.940+00:00
  __v: 0
▾ completed: Array (4)
    0: ObjectId('65b587ba7066beb797b6558c')
    1: ObjectId('65b52825aac51805b72b8e70')
    2: ObjectId('65b55f4faac51805b72b8eb0')
    3: ObjectId('65b564ccaac51805b72b8ebc')
```

Figure 4-3. Student document example in MongoDB

4.1.4 Faculty

The Faculty entity is divided into two roles to represent administrative and teaching staff. Administrator users have the capability to configure the system in order for the students to submit their preferences, which includes adding the courses to the system, arranging the semesters, viewing the submitted preferences of the students, and other. On the other hand, faculty users that do not have administrator privileges have limited configuration properties throughout the system. The role of the user is determined by the two fields of the schema ‘admin’ and ‘instructor’. The rest of the fields of the Faculty model are listed below:

- username: String, Unique index, required.
- password: String, required. Encrypted string of the password.
- first_name: String. First name of the faculty user.
- last_name: String. Last name of the faculty user.
- id_number: String. ID number of faculty user.
- email: String, required. Email of the faculty user.
- instructor: Boolean. Default: false. Indicates if the faculty user is teaching staff.
- admin: Boolean. Default: false. Indicates if the faculty user is administrative staff.
- logged_in: Boolean. Default: false. Login status of user.



```
_id: ObjectId('657a1dca644976842826e6d5')
username: "admin01"
password: "$2b$10$1wfafRHWek//GAU/Frh2R.88JTNitlIe8Jeq6s/9cYWJidARZnFh."
first_name: "admin01"
last_name: "admin"
id_number: ""
email: "admin01@cs.ucy.ac.cy"
instructor: false
admin: true
logged_in: false
createdAt: 2023-12-13T21:10:34.122+00:00
updatedAt: 2024-04-24T19:48:18.367+00:00
__v: 0
```

Figure 4-4. Faculty document example in MongoDB

4.2 System Roles

As it is derived from the definition and description of the entities, the primary roles that exist in the system are the students and faculty users. Faculty users can be further distinct into two more roles, the administrators and instructors (teaching staff), but they are not limited to only one of the two properties.

The students are able to set the restricted elective courses they prefer to attend in the next few semesters. They can view information about the courses which could help them decide what courses they are interested in based on a specialisation they want to qualify for, or the content that will be taught in them. Managing or reviewing their profile is also possible, so that it is kept up to date and to also help to provide a better experience for the students with additional system functionality which uses the semester they are studying in and the courses they have succeeded in until that moment.

Administrator faculty users have the capacity to configure the system so that the students can submit their preferences, which includes setting up the (future) semesters and adding the courses in the system. Administrators are able to explore the submitted preferences of the students collectively for the semesters and other results that can be exported from them, presented as Reports. This information will be intended to provide an insight of the restricted elective courses demanded the most by the students, which can contribute to a better scheduling and overall satisfaction of the students' interests. Instructor-only faculty users, on the other hand, are not able to configure the system to this extent, only the courses they are set as instructor, for latest changes and updates on a course and to lighten the administrators' tasks.

4.3 Student Functionality

As previously described a student can use this system mainly to submit their preferences regarding the restricted elective courses they want to take in the next semesters. Moreover, courses information can be provided, for the students to instantly view the content each course offers, the specialisations a course can contribute to acquire, and other. In the following subchapters are listed the actions that a student can perform throughout the system.

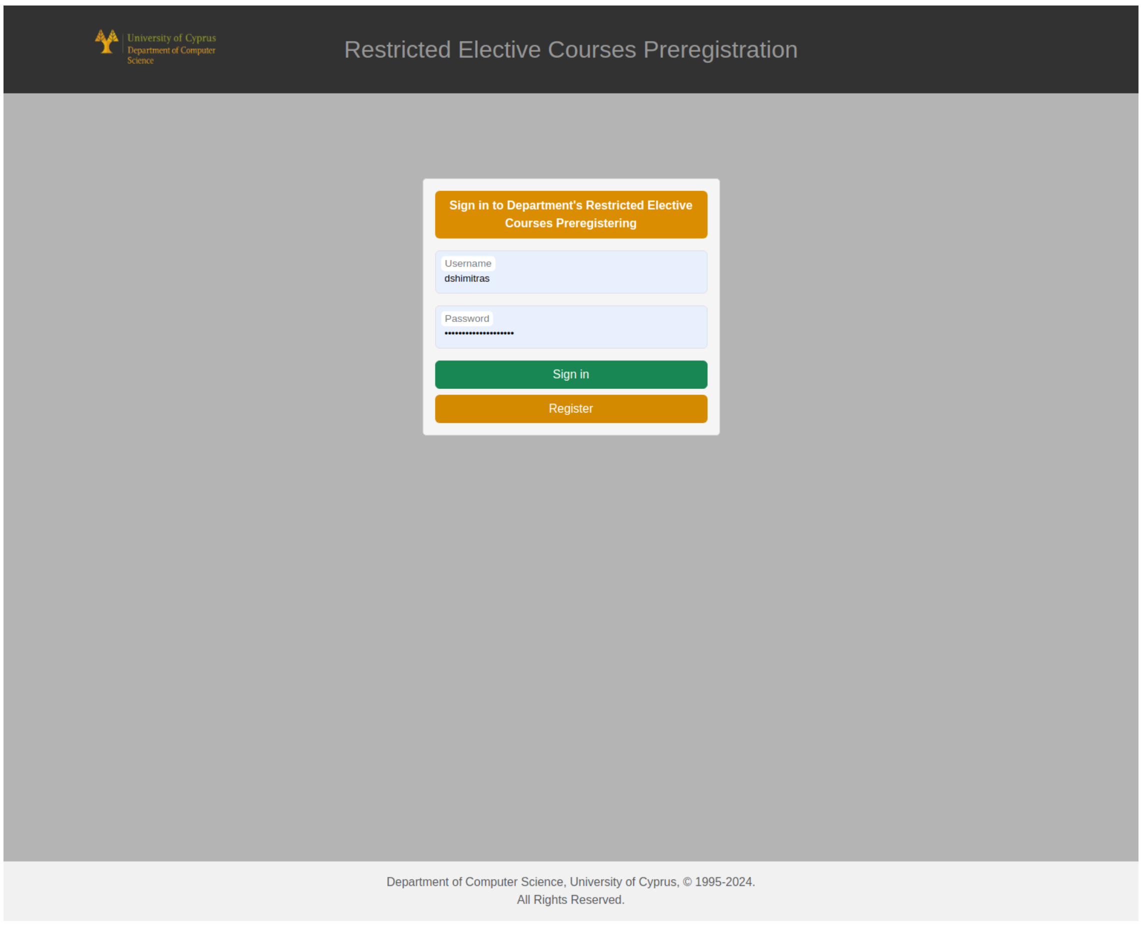
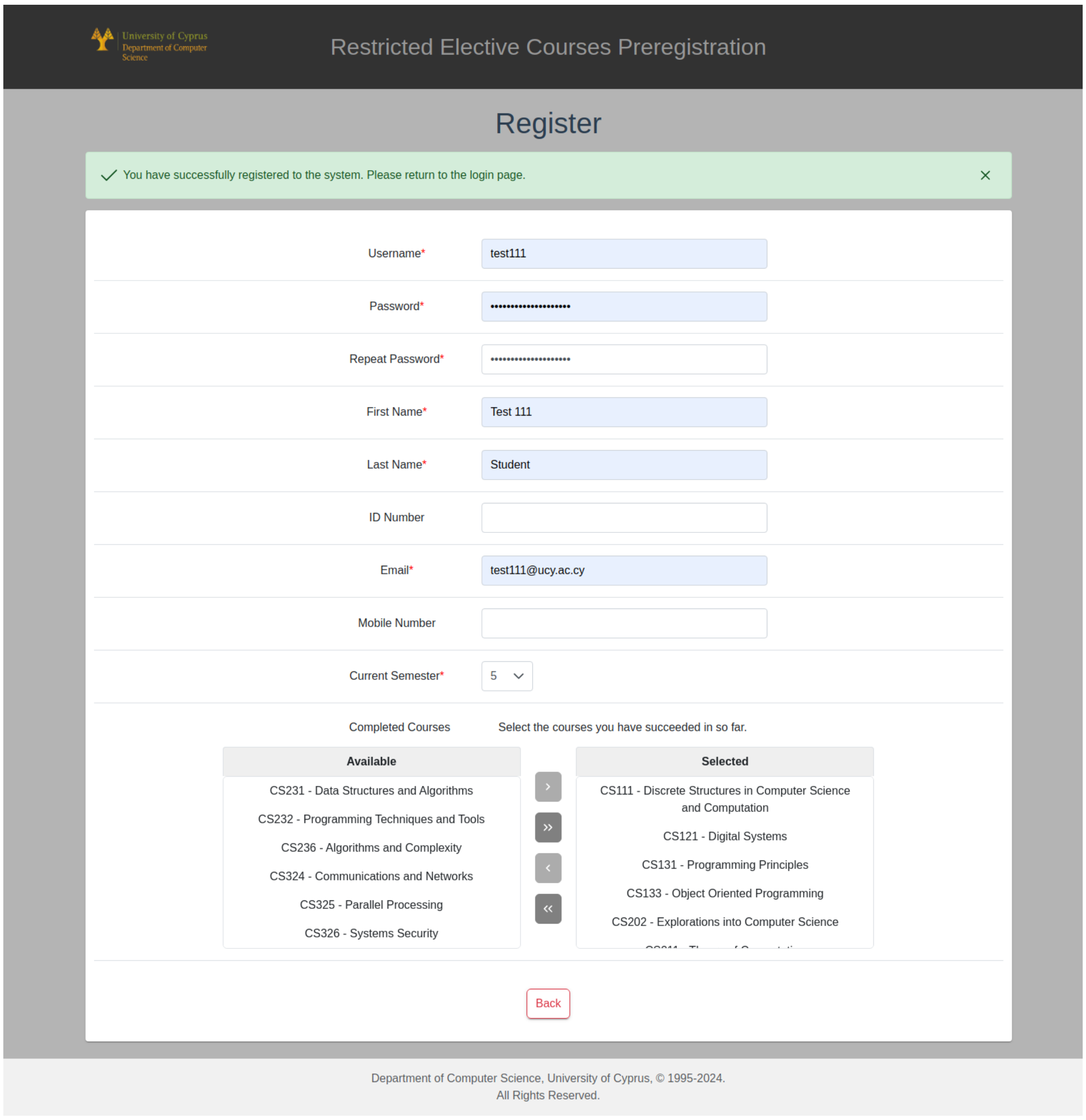

University of Cyprus
Department of Computer Science
Restricted Elective Courses Preregistration
Sign in to Department's Restricted Elective Courses Preregistering
Username
dshimitras
Password
Sign in
Register
Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-5. Student login page

4.3.1 Register

Students will need to first register to the system to create their profile, through the Login page. They will need to provide their username and password, personal details and can also input some of their academic information for a better interaction with the system while submitting their courses. As future implementation, there could be an integration with the Department's or University's internal systems that would retrieve most of this information, to significantly improve the new system's functionality in user credentials authentication and always have the correct updated user personal and academic information. However, for the purpose of this project, the process of user registration within the new system was needed.



Restricted Elective Courses Preregistration

Register

✓ You have successfully registered to the system. Please return to the login page.

Username*	test111
Password*	••••••••••••••••••••
Repeat Password*	••••••••••••••••••••
First Name*	Test 111
Last Name*	Student
ID Number	
Email*	test111@ucy.ac.cy
Mobile Number	
Current Semester*	5

Completed Courses — Select the courses you have succeeded in so far.

Available	Selected
CS231 - Data Structures and Algorithms	CS111 - Discrete Structures in Computer Science and Computation
CS232 - Programming Techniques and Tools	CS121 - Digital Systems
CS236 - Algorithms and Complexity	CS131 - Programming Principles
CS324 - Communications and Networks	CS133 - Object Oriented Programming
CS325 - Parallel Processing	CS202 - Explorations into Computer Science
CS326 - Systems Security	

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-6. Student registration

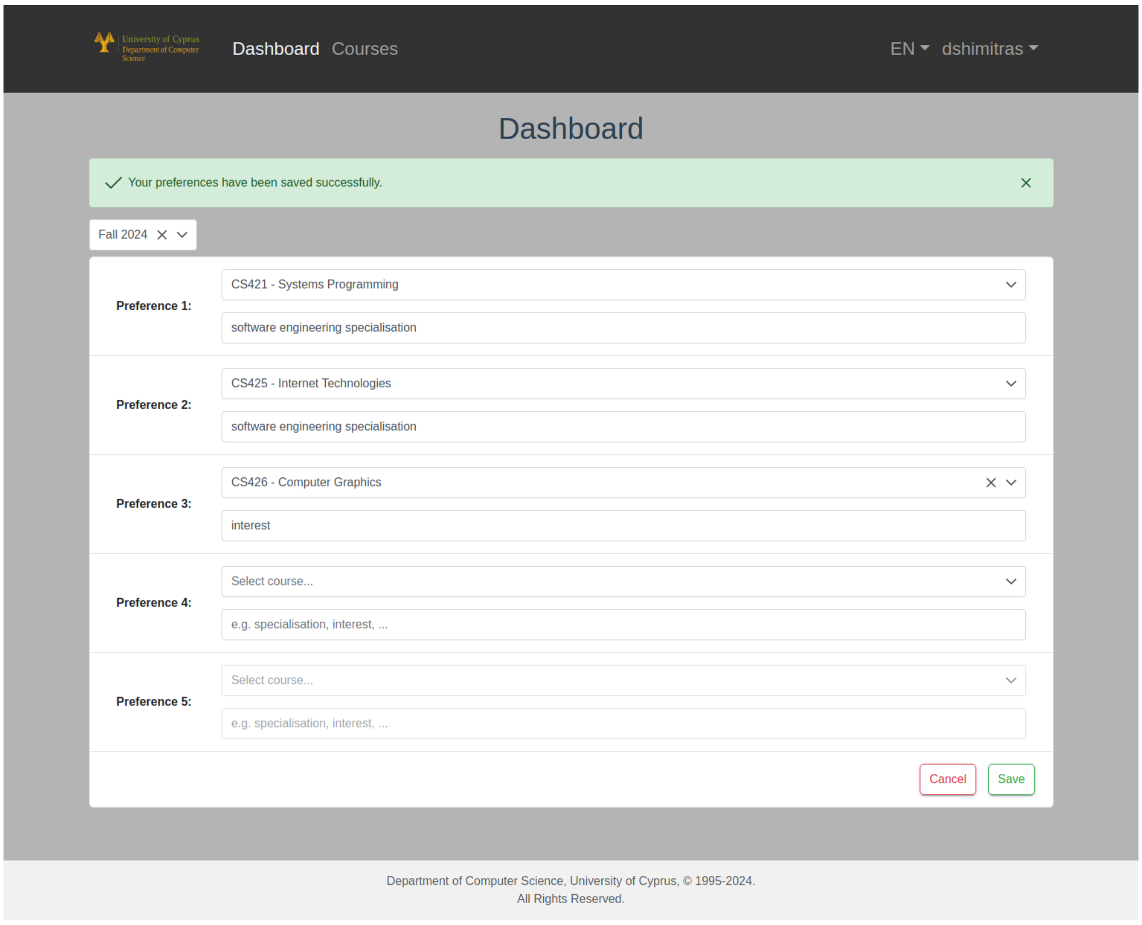
4.3.2 Login

If a student has already registered to the system, they can proceed to log in using the username and password they have used to sign up. At a later phase, the login process could ideally be made via integration with an API of the University's or Department's, which would authenticate the students' central credentials, as it is made with other external systems used, such as Moodle, Blackboard, and others.

4.3.3 Submit Preferences

With a successful login from a student using their username and password, they are redirected to the Dashboard page, which includes the component for submitting their preferences.

Within the preferences components, students can select one of the future semesters from the dropdown list, when their current stored preferences will be retrieved from the database, if already exist, and populate the form. Upon selecting a semester, preferences of restricted elective courses may be chosen in order of their interest in each course. Their first course selection should be the one with the highest priority of wanting to take the course in the semester, and so on. A short descriptive text may be given for each preference selection, as a reason for choosing the corresponding course to attend in the future. Eventually, they can submit their chosen courses for the selected semester to be saved in the database.



Your preferences have been saved successfully.

Fall 2024

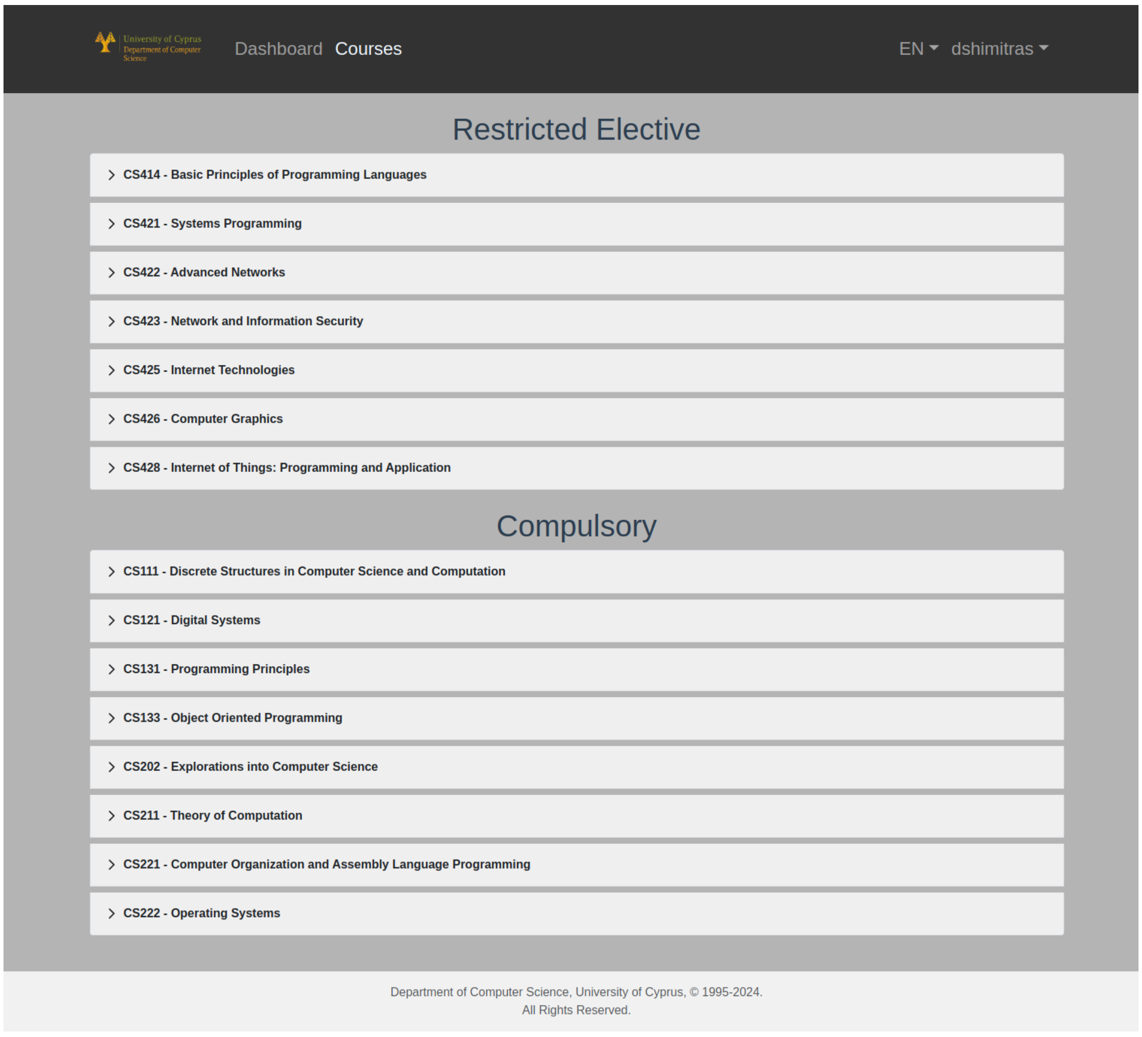
Preference 1:	CS421 - Systems Programming software engineering specialisation
Preference 2:	CS425 - Internet Technologies software engineering specialisation
Preference 3:	CS426 - Computer Graphics interest
Preference 4:	Select course... e.g. specialisation, interest, ...
Preference 5:	Select course... e.g. specialisation, interest, ...

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-7. Selection of course preferences for a semester

4.3.4 View Courses

In a simple page, the information about the courses offered by the Department are presented in a similar way found in the Department's website, where the students can review to decide which ones would prefer to attend in the next semesters, if they wish. Both compulsory and restricted elective courses can exist in the system and be displayed in this page, which are distinguished into different sections, giving emphasis on the restricted ones, being the subject of this project.



Restricted Elective

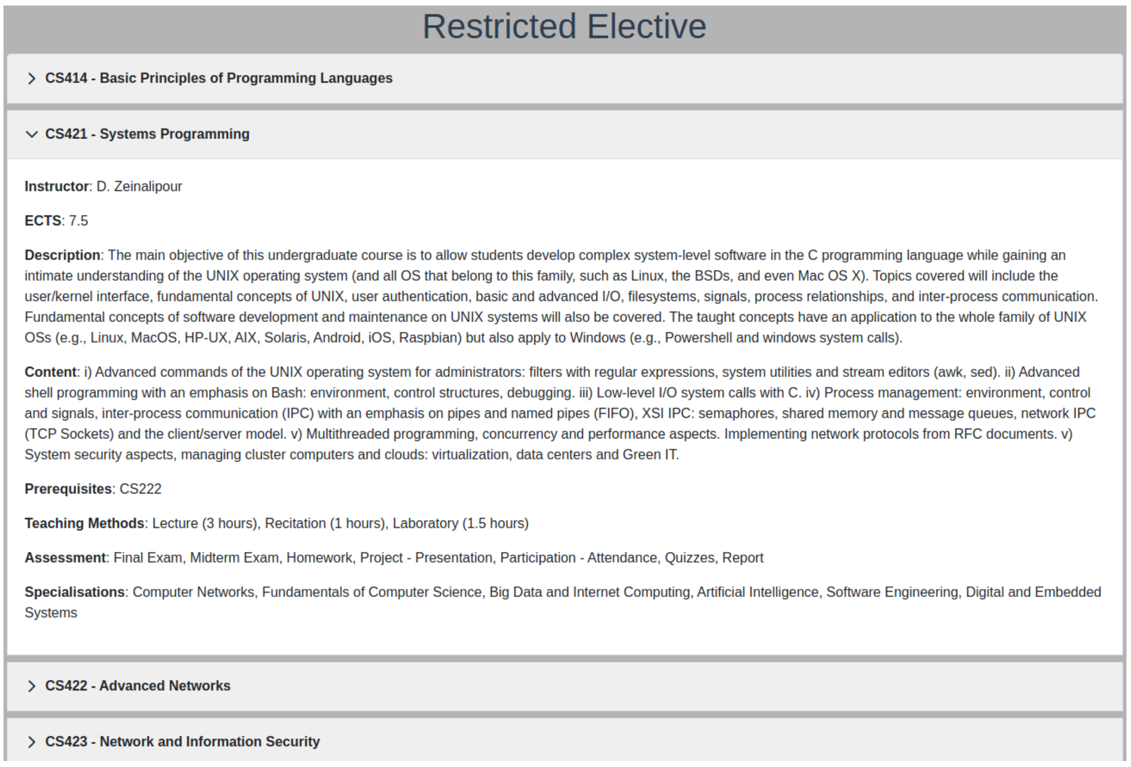
- CS414 - Basic Principles of Programming Languages
- CS421 - Systems Programming
- CS422 - Advanced Networks
- CS423 - Network and Information Security
- CS425 - Internet Technologies
- CS426 - Computer Graphics
- CS428 - Internet of Things: Programming and Application

Compulsory

- CS111 - Discrete Structures in Computer Science and Computation
- CS121 - Digital Systems
- CS131 - Programming Principles
- CS133 - Object Oriented Programming
- CS202 - Explorations into Computer Science
- CS211 - Theory of Computation
- CS221 - Computer Organization and Assembly Language Programming
- CS222 - Operating Systems

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-8. Courses available in the system


Restricted Elective

> CS414 - Basic Principles of Programming Languages

CS421 - Systems Programming

Instructor: D. Zeinalipour

ECTS: 7.5

Description: The main objective of this undergraduate course is to allow students develop complex system-level software in the C programming language while gaining an intimate understanding of the UNIX operating system (and all OS that belong to this family, such as Linux, the BSDs, and even Mac OS X). Topics covered will include the user/kernel interface, fundamental concepts of UNIX, user authentication, basic and advanced I/O, filesystems, signals, process relationships, and inter-process communication. Fundamental concepts of software development and maintenance on UNIX systems will also be covered. The taught concepts have an application to the whole family of UNIX OSs (e.g., Linux, MacOS, HP-UX, AIX, Solaris, Android, iOS, Raspbian) but also apply to Windows (e.g., Powershell and windows system calls).

Content: i) Advanced commands of the UNIX operating system for administrators: filters with regular expressions, system utilities and stream editors (awk, sed). ii) Advanced shell programming with an emphasis on Bash: environment, control structures, debugging. iii) Low-level I/O system calls with C. iv) Process management: environment, control and signals, inter-process communication (IPC) with an emphasis on pipes and named pipes (FIFO), XSI IPC: semaphores, shared memory and message queues, network IPC (TCP Sockets) and the client/server model. v) Multithreaded programming, concurrency and performance aspects. Implementing network protocols from RFC documents. v) System security aspects, managing cluster computers and clouds: virtualization, data centers and Green IT.

Prerequisites: CS222

Teaching Methods: Lecture (3 hours), Recitation (1 hours), Laboratory (1.5 hours)

Assessment: Final Exam, Midterm Exam, Homework, Project - Presentation, Participation - Attendance, Quizzes, Report

Specialisations: Computer Networks, Fundamentals of Computer Science, Big Data and Internet Computing, Artificial Intelligence, Software Engineering, Digital and Embedded Systems

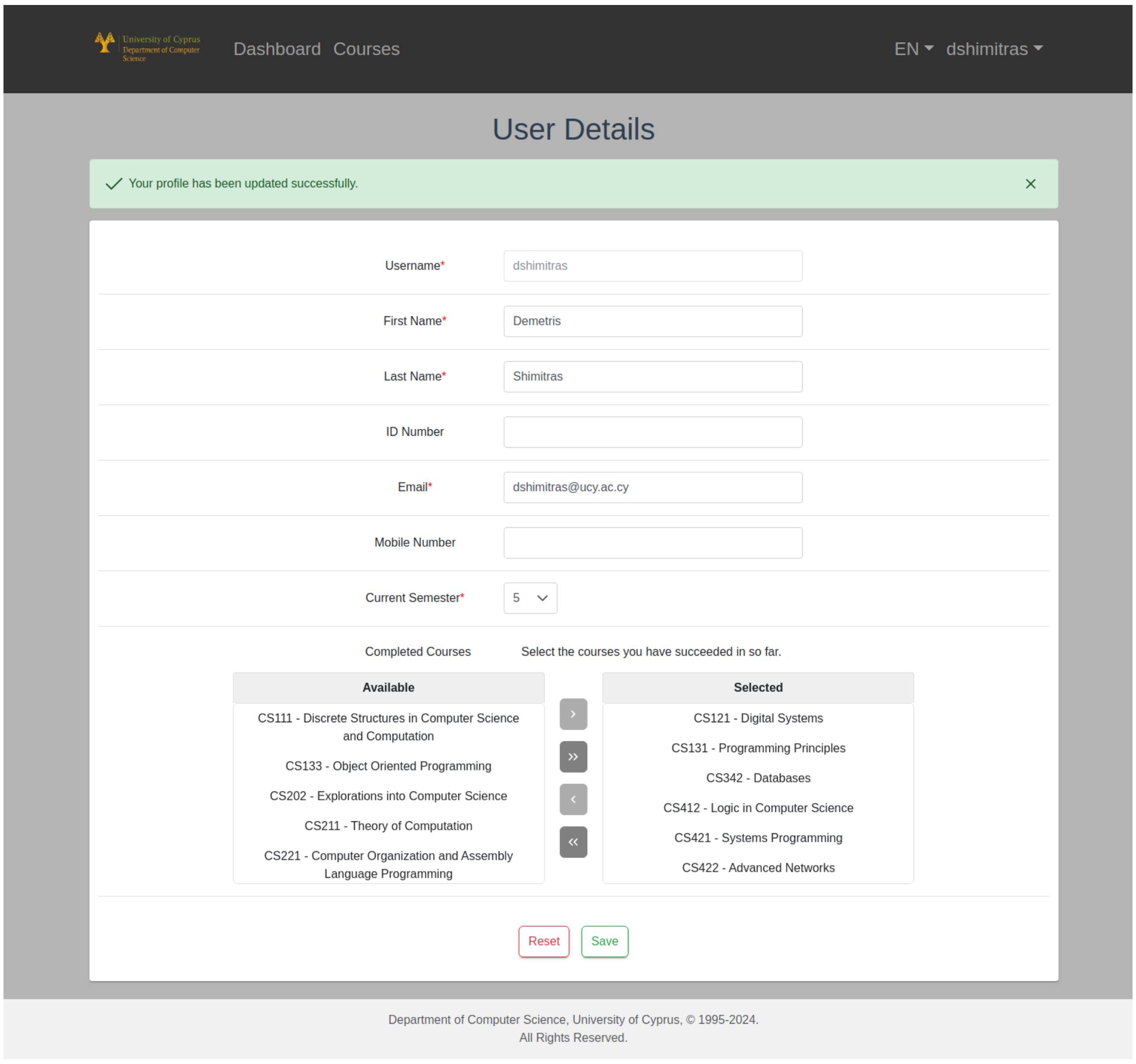
> CS422 - Advanced Networks

> CS423 - Network and Information Security

Figure 4-9. Available courses information

4.3.5 View and Update User Details

Identical to their registration, students will be able to change their personal and academic information in the system at any moment, mainly to change their current semester studying in and any new courses they have succeeded to so far during their program of study.



User Details

Your profile has been updated successfully.

Username*	dshimitras
First Name*	Demetris
Last Name*	Shimitras
ID Number	
Email*	dshimitras@ucy.ac.cy
Mobile Number	
Current Semester*	5

Completed Courses — Select the courses you have succeeded in so far.

Available	Selected
CS111 - Discrete Structures in Computer Science and Computation	CS121 - Digital Systems
CS133 - Object Oriented Programming	CS131 - Programming Principles
CS202 - Explorations into Computer Science	CS342 - Databases
CS211 - Theory of Computation	CS412 - Logic in Computer Science
CS221 - Computer Organization and Assembly Language Programming	CS421 - Systems Programming
	CS422 - Advanced Networks

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-10. Student profile. Includes some personal and academic information

4.3.6 Logout

Students can log out of their user account when they are finished submitting their preferences or performing other actions, redirected to the login page again.

4.4 Faculty Functionality

Faculty users will be logging in to the system through a different subdomain of the web application, with generally distinguished functionality from that of the students. No registration page is available for faculty users. Initially, an administrator faculty user could be manually created in the database to start configuring the system. Similarly with instructor faculty users, no current functionality allows for registering, or for admins to add them to the system (which can be included with future implementation). Ideally, as with students, integrating with the University's or Department's internal systems to retrieve user information could simplify registration and login process, and increase user experience.

The functionalities existing in the system are described in the sections below, specifying any differences that there can be between an administrator user and a non-administrator one.

4.4.1 Login

All faculty users log in their user accounts with the username and password. As mentioned before, an integration with the internal systems of the Department and University would allow accurate and centralised authentication of the users. Since this was not possible in the scope of the development of this project, a registration form for faculty users was omitted to avoid unauthorised access.

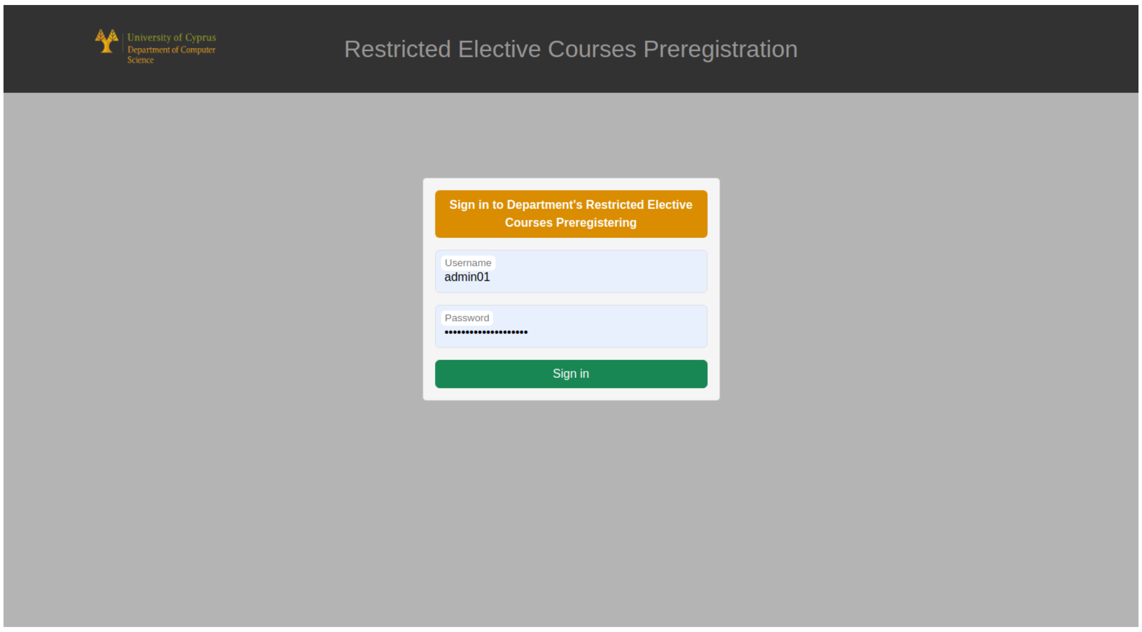
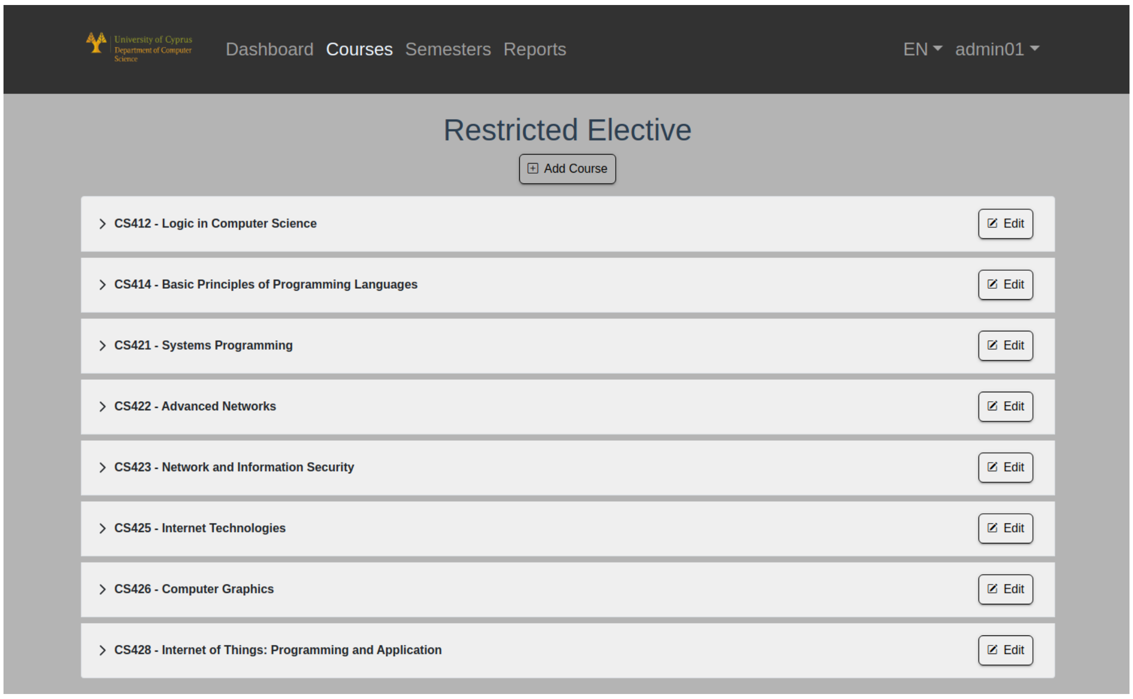

University of Cyprus
Department of Computer Science
Restricted Elective Courses Preregistration
Sign in to Department's Restricted Elective Courses Preregistering
Username
admin01
Password
Sign in

Figure 4-11. Faculty login page

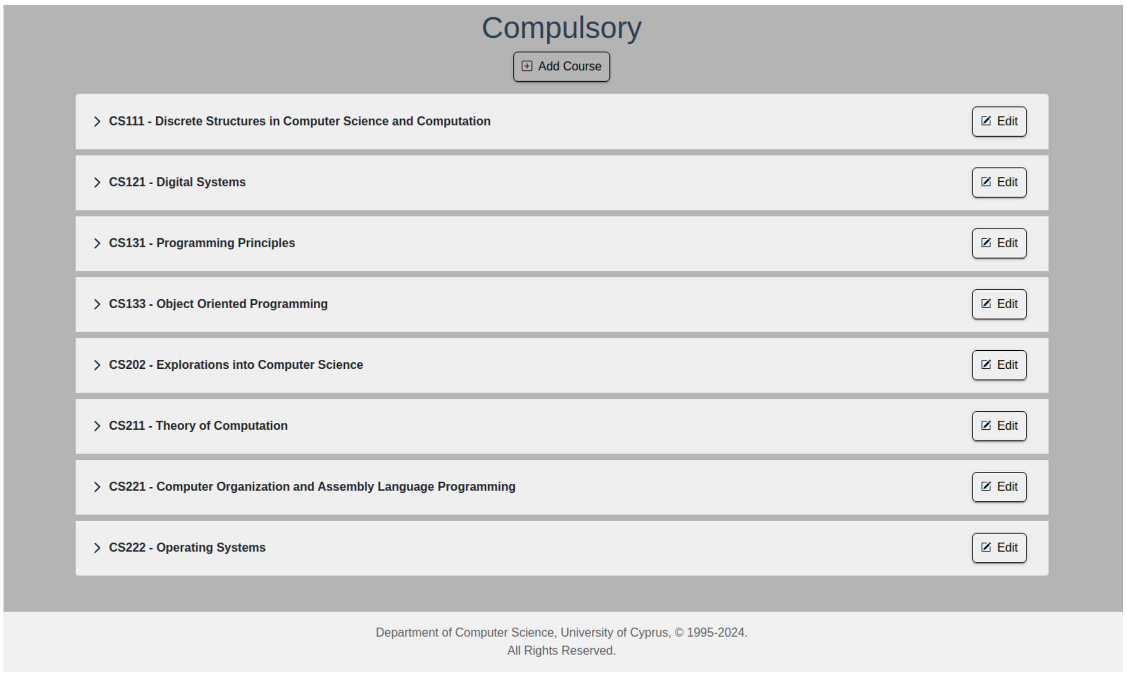
4.4.2 Courses

The Courses page for faculty users is presented in the same way as it is for the students, with the variation that administration users have the capability to add new courses in the system, or update existing ones.



CS412 - Logic in Computer Science
CS414 - Basic Principles of Programming Languages
CS421 - Systems Programming
CS422 - Advanced Networks
CS423 - Network and Information Security
CS425 - Internet Technologies
CS426 - Computer Graphics
CS428 - Internet of Things: Programming and Application

Figure 4-12a. Administrator courses information and edit capability



Compulsory

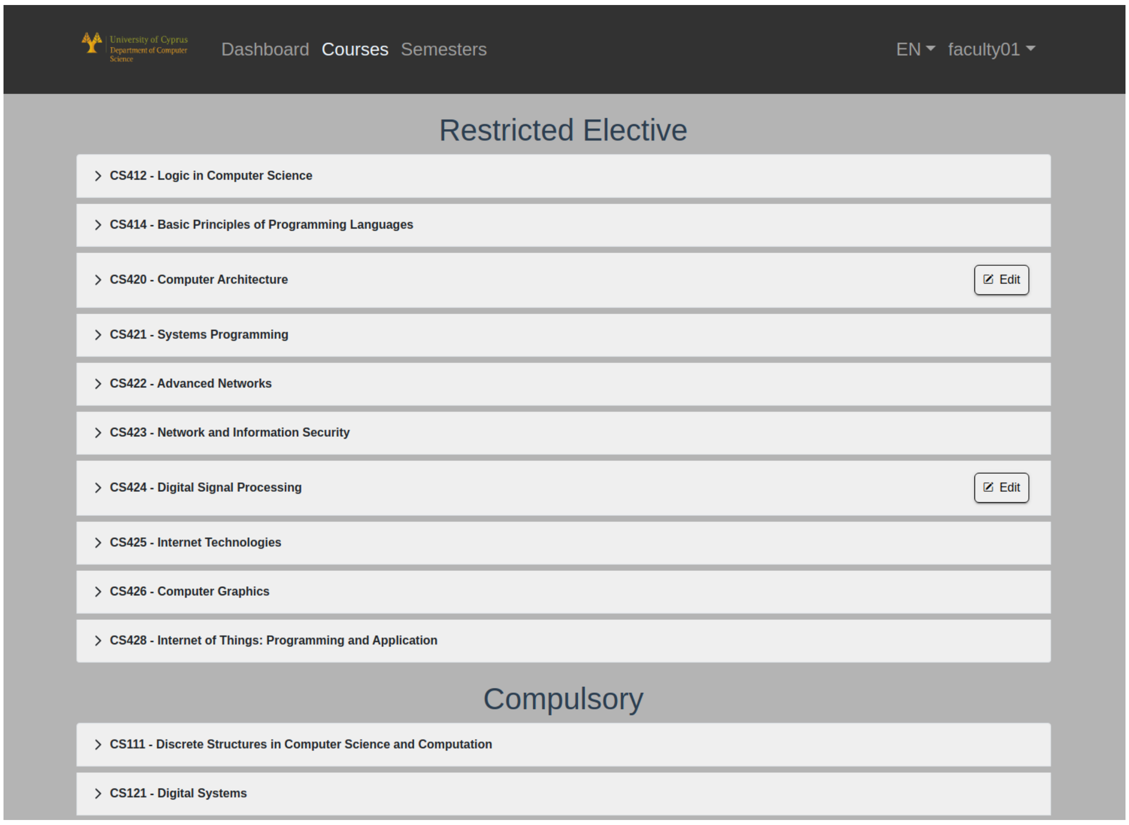
Add Course

- CS111 - Discrete Structures in Computer Science and Computation — Edit
- CS121 - Digital Systems — Edit
- CS131 - Programming Principles — Edit
- CS133 - Object Oriented Programming — Edit
- CS202 - Explorations into Computer Science — Edit
- CS211 - Theory of Computation — Edit
- CS221 - Computer Organization and Assembly Language Programming — Edit
- CS222 - Operating Systems — Edit

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-12b. Administrator courses information with edit capability

Faculty users that do not have administrative privileges will also be able to edit a course in the list, with the condition that the instructor set for the course is the same user.



Restricted Elective

- CS412 - Logic in Computer Science
- CS414 - Basic Principles of Programming Languages
- CS420 - Computer Architecture — Edit
- CS421 - Systems Programming
- CS422 - Advanced Networks
- CS423 - Network and Information Security
- CS424 - Digital Signal Processing — Edit
- CS425 - Internet Technologies
- CS426 - Computer Graphics
- CS428 - Internet of Things: Programming and Application

Compulsory

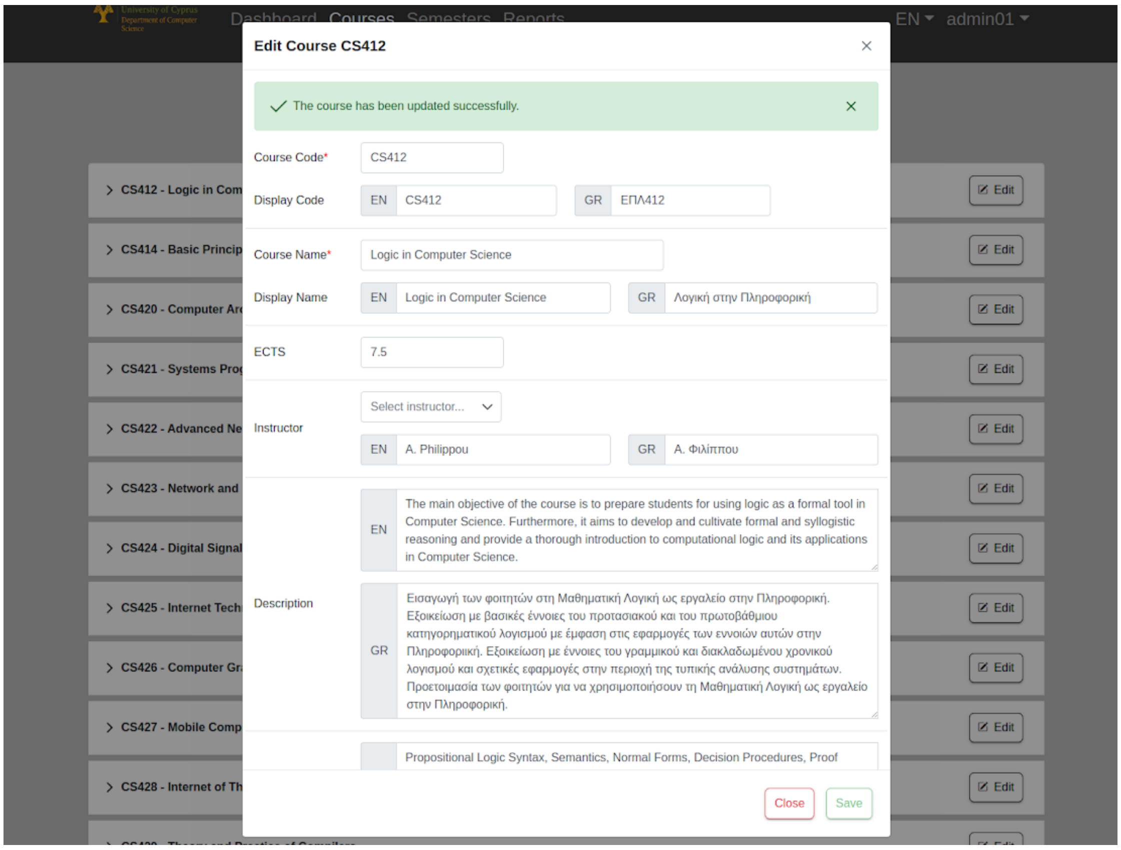
- CS111 - Discrete Structures in Computer Science and Computation
- CS121 - Digital Systems

Figure 4-13. Non-admin courses information with minimal editing capability

Adding a new course opens a popup to fill in information and other functional details about the course to be saved in the database, whereas choosing to edit an existing one, opens the popup with populated fields based on the data already stored in the database for the specified course.

The details that can be given when adding or updating a course include the course's code and name, which are required at the least. Additional data that can be provided are listed below:

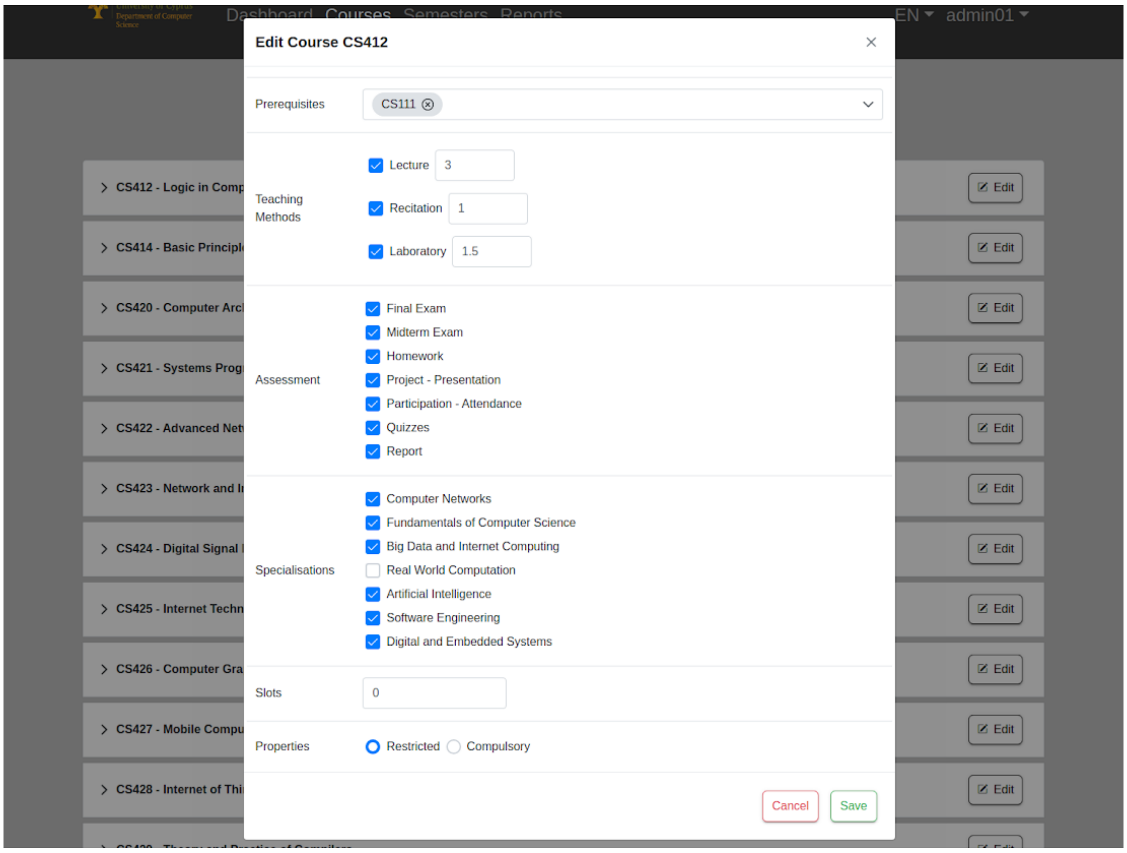
- The number of credits (ECTS) the student earns when succeeding the course
- The instructor that will possibly be teaching the course
- An overview of the course content and objectives
- A multi-selection of other courses which are prerequisite to the current one
- Selection of teaching methods for the course (e.g. Lectures, Labs)
- Selection of specialisations that the course can contribute towards acquiring them. The selection is applicable only for restricted-type courses.
- The type of the course: Compulsory or Restricted Elective.


Edit Course CS412

✓ The course has been updated successfully.

Course Code*	CS412
Display Code	EN CS412 / GR ΕΠΛ412
Course Name*	Logic in Computer Science
Display Name	EN Logic in Computer Science / GR Λογική στην Πληροφορική
ECTS	7.5
Instructor	Select instructor... / EN A. Philippou / GR Α. Φιλίππου
Description	EN The main objective of the course is to prepare students for using logic as a formal tool in Computer Science. Furthermore, it aims to develop and cultivate formal and syllogistic reasoning and provide a thorough introduction to computational logic and its applications in Computer Science.
	GR Εισαγωγή των φοιτητών στη Μαθηματική Λογική ως εργαλείο στην Πληροφορική. Εξοικείωση με βασικές έννοιες του προτασιακού και του πρωτοβάθμιου κατηγορηματικού λογισμού με έμφαση στις εφαρμογές των εννοιών αυτών στην Πληροφοριική. Εξοικείωση με έννοιες του γραμμικού και διακλαδωμένου χρονικού λογισμού και σχετικές εφαρμογές στην περιοχή της τυπικής ανάλυσης συστημάτων. Προετοιμασία των φοιτητών για να χρησιμοποιήσουν τη Μαθηματική Λογική ως εργαλείο στην Πληροφορική.
	Propositional Logic Syntax, Semantics, Normal Forms, Decision Procedures, Proof

Figure 4-14a. Edit course information

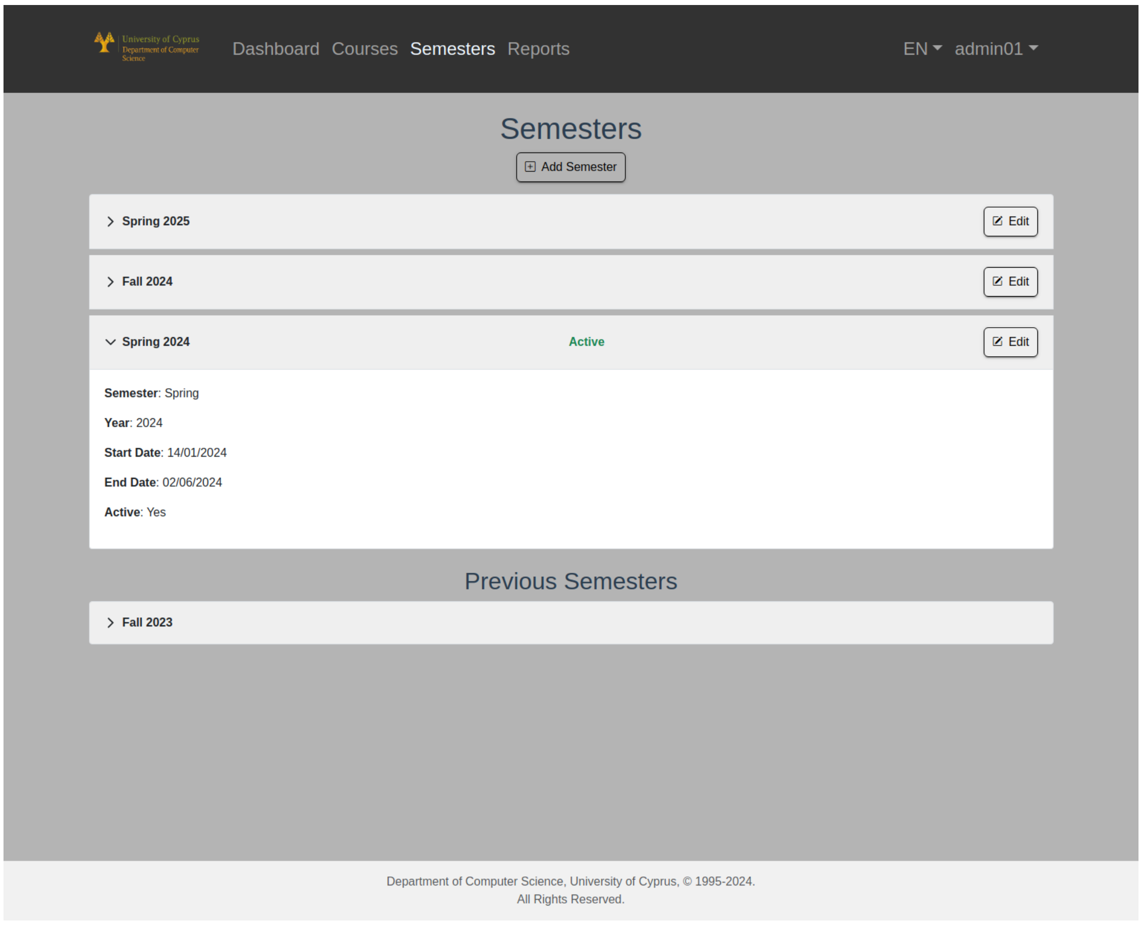

Edit Course CS412

Prerequisites	CS111
Teaching Methods	✓ Lecture 3; ✓ Recitation 1; ✓ Laboratory 1.5
Assessment	✓ Final Exam; ✓ Midterm Exam; ✓ Homework; ✓ Project - Presentation; ✓ Participation - Attendance; ✓ Quizzes; ✓ Report
Specialisations	✓ Computer Networks; ✓ Fundamentals of Computer Science; ✓ Big Data and Internet Computing; ☐ Real World Computation; ✓ Artificial Intelligence; ✓ Software Engineering; ✓ Digital and Embedded Systems
Slots	0
Properties	[✓ Restricted] Compulsory

Figure 4-14b. Edit course information

4.4.3 Semesters

Semesters page and configuration from the administrator users, works in a similar way as with the courses, denoting that the existing semesters are presented as a list, which can be modified by selecting one for edit, through a modal. A semester is defined by its term, 'Spring', 'Fall', or 'Summer', and the year, e.g. 'Spring 2024'. Admins must also define the semesters' start and end date, which are used to show the future semester only to the students when submitting their preferences. There is also an additional flag field that indicates if a semester is active (set manually). Only one semester can be active at a time.



Semesters

Spring 2025 Edit

Fall 2024 Edit

Spring 2024 Active Edit

Semester: Spring

Year: 2024

Start Date: 14/01/2024

End Date: 02/06/2024

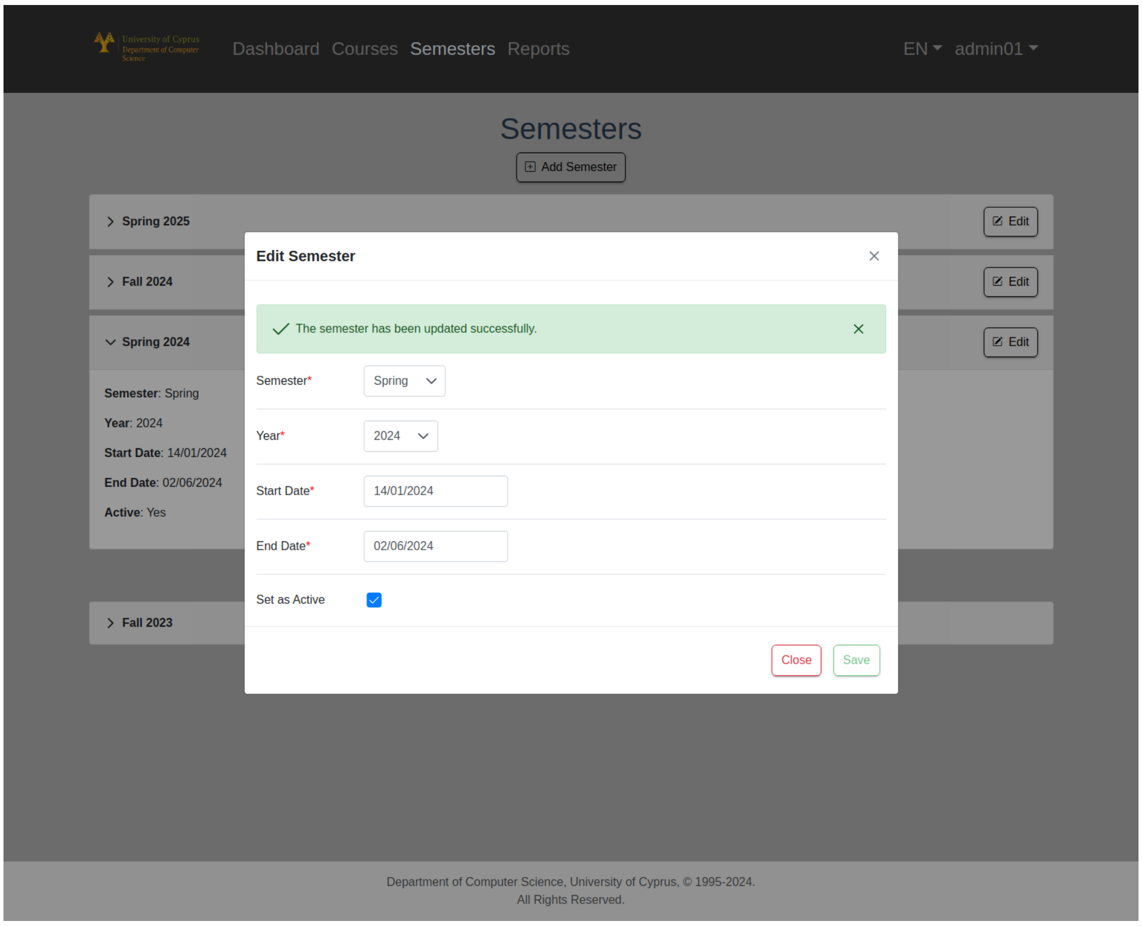
Active: Yes

Previous Semesters

Fall 2023

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-15. Semester information (with administrator edit capability)



Semester: Spring
Year: 2024
Start Date: 14/01/2024
End Date: 02/06/2024
Active: Yes

The semester has been updated successfully.

Semester*	Spring
Year*	2024
Start Date*	14/01/2024
End Date*	02/06/2024
Set as Active	✓

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-16. Edit semester information

Faculty users that are not administrators do not have the capability to add or edit the semester, only to view the page.

4.4.4 Reports

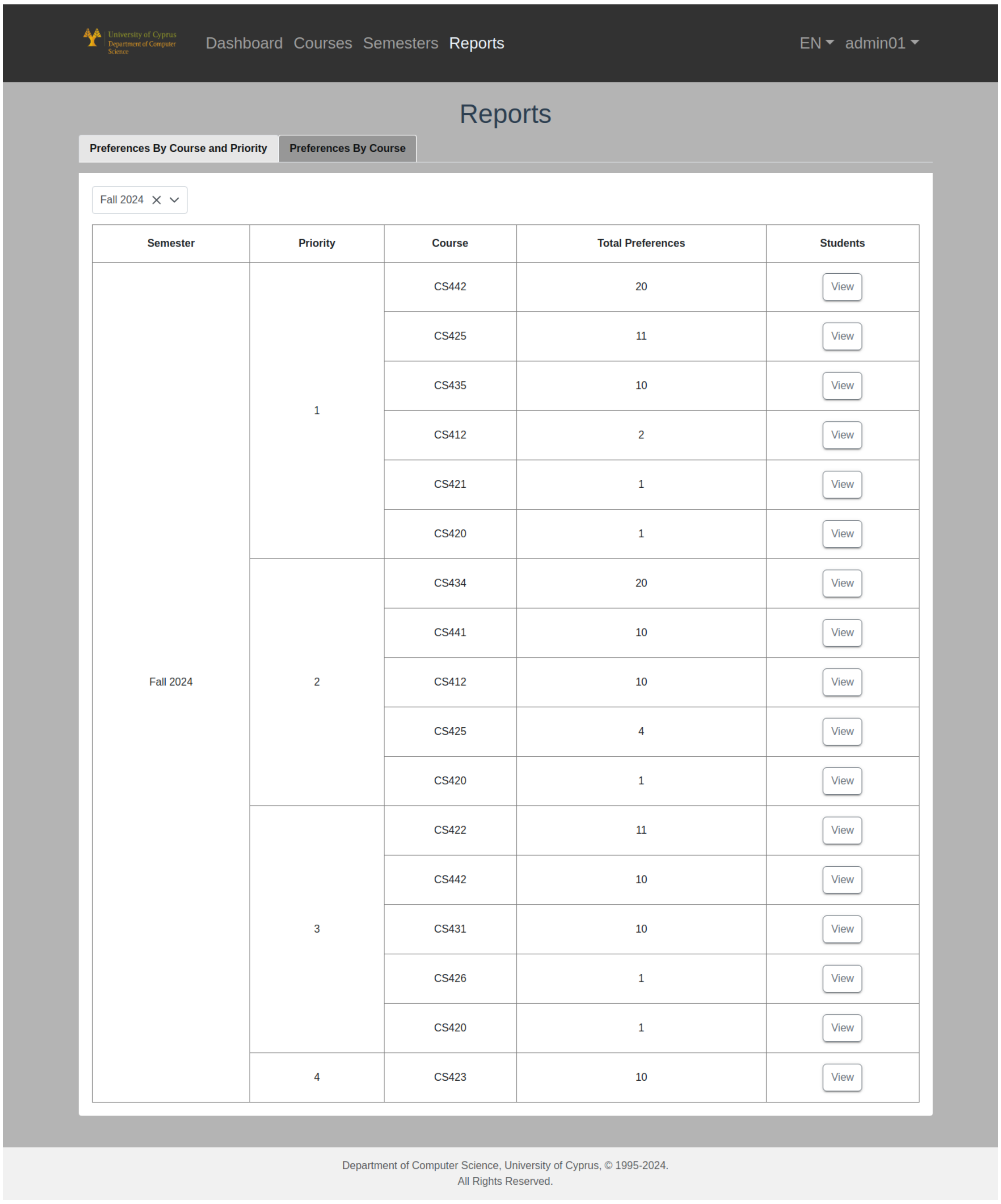
The purpose of this implementation as a solution of the problem is to aid the Computer Science Department to offer the restricted elective courses that would satisfy the most students in each semester, based on the courses they would like to attend. So far, the students can submit their preferences for a semester, by selecting the courses with priority on which one they most want to take to the least wanted.

Therefore, what is needed next, is to take the students' submitted preferences into consideration in order to acquire the restricted courses that are most demanded. For this, the Reports page will accommodate any generated results in regard to helping decide which courses would be more suitable to offer in each of the next semesters, based on the students' input preferences. Different types of results are presented in separate tabs inside the Reports page. Only admin users can view this page.

Currently, two results (reports) are generated in the system for the submitted preferences. The first tab in the Reports page shows the aggregated preferences based on the semester, the course, and the priority that was selected for the course. In the second tab, another aggregation result shows the total number of preferences a course has for a semester. The two aggregations are described more below.

4.4.4.1 Total preferences per course and priority

This tab presents a table with the total number of students that have selected a course and the priority they had chosen it for. This information is given for one semester at a time; an admin user must select a specific semester to get the aggregation results. The report could indicate whether a course has a high demand from the students for the selected semester, and if with a high priority, it could be considered as a probable course offer. Additionally, if there is not much interest for any course as first priority, but many students have preferred a course as their second or third priority, it could also be a possible offer for the semester, since it would still satisfy enough students on some level.



Reports

Preferences By Course and Priority | Preferences By Course

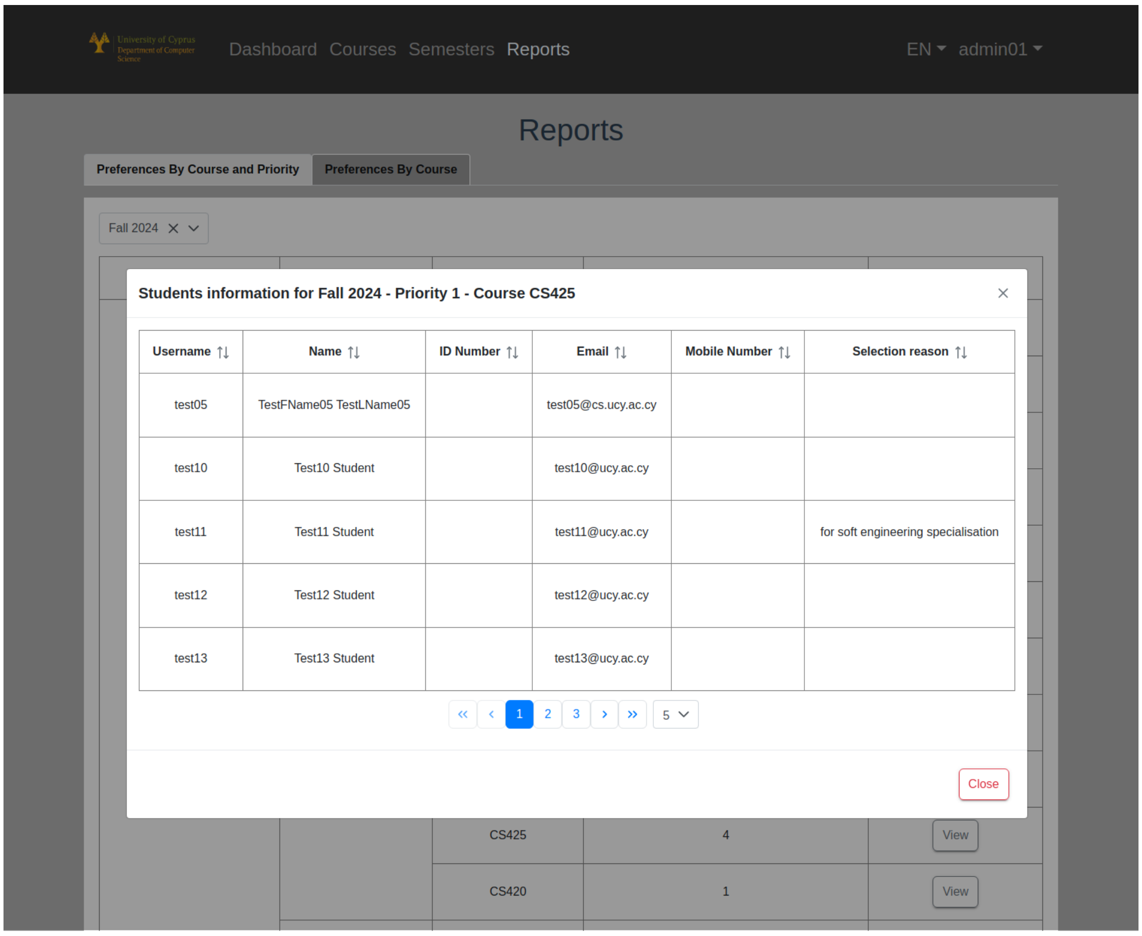
Fall 2024

Semester	Priority	Course	Total Preferences	Students
Fall 2024	1	CS442	20	View
		CS425	11	View
		CS435	10	View
		CS412	2	View
		CS421	1	View
		CS420	1	View
	2	CS434	20	View
		CS441	10	View
		CS412	10	View
		CS425	4	View
		CS420	1	View
	3	CS422	11	View
		CS442	10	View
		CS431	10	View
		CS426	1	View
		CS420	1	View
	4	CS423	10	View

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-17. Total preferences per Semester, Course and Priority

Administrators will have the option to view a list of information about the students who have made a specific preference selection for the semester, course and the priority they had given it. The students' information includes their name, username, id number, email and mobile phone number, as they have submitted them in the system. Moreover, if students have provided a reason of why they had selected the specific course, it will be displayed to the table.



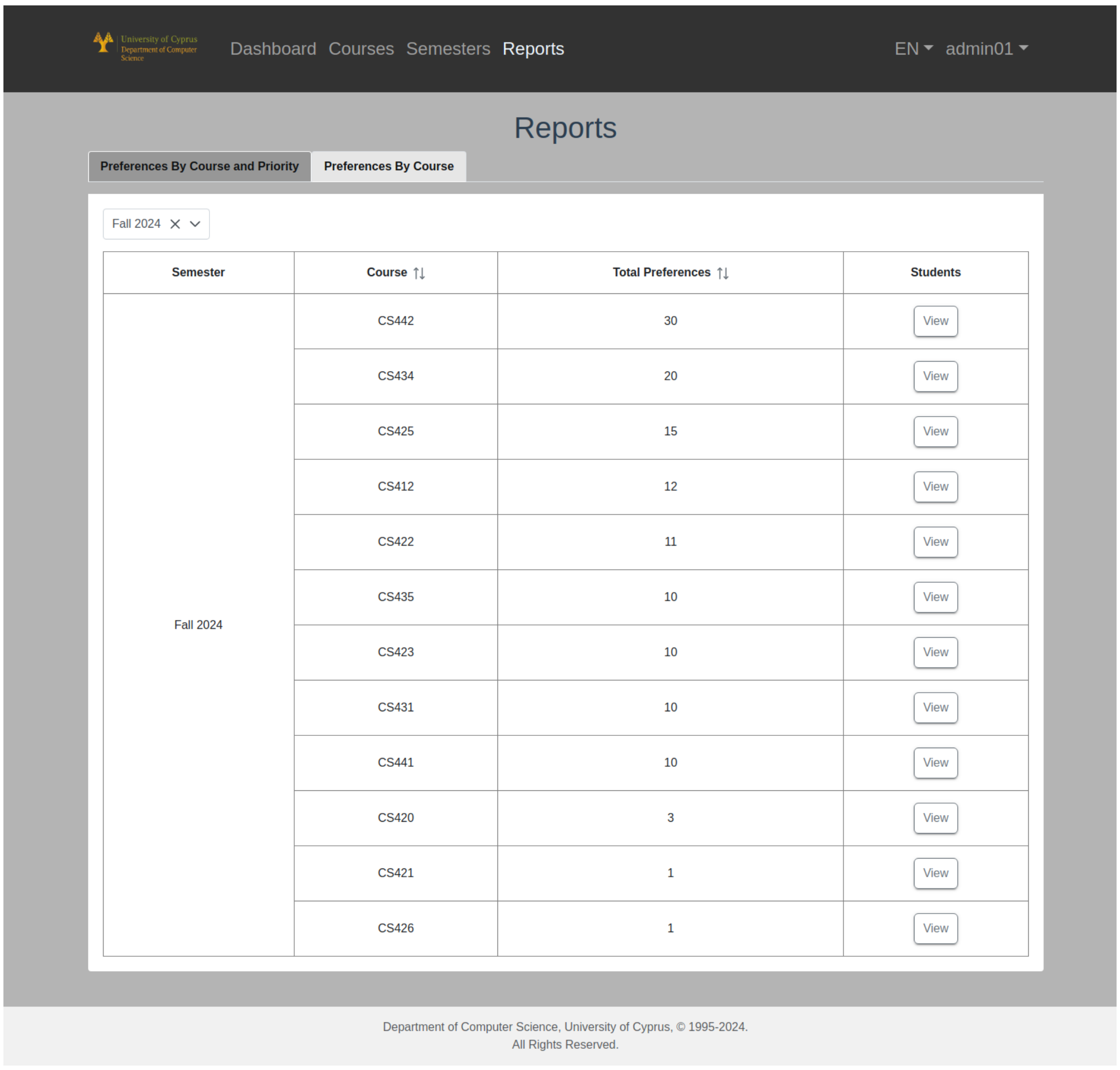
Students information for Fall 2024 - Priority 1 - Course CS425

Username	Name	ID Number	Email	Mobile Number	Selection reason
test05	TestFName05 TestLName05		test05@cs.ucy.ac.cy		
test10	Test10 Student		test10@ucy.ac.cy		
test11	Test11 Student		test11@ucy.ac.cy		for soft engineering specialisation
test12	Test12 Student		test12@ucy.ac.cy		
test13	Test13 Student		test13@ucy.ac.cy		

Figure 4-18. Student information for a Semester-Course-Priority preference

4.4.4.2 Total preferences/students per course

Similar to the first report tab, the second one will show the total selections of each course made by the students for a semester, regardless of the priority they specified. This can immediately indicate which courses are the most demanded.


University of Cyprus Department of Computer Science
Dashboard
Courses
Semesters
Reports
EN
admin01

Reports

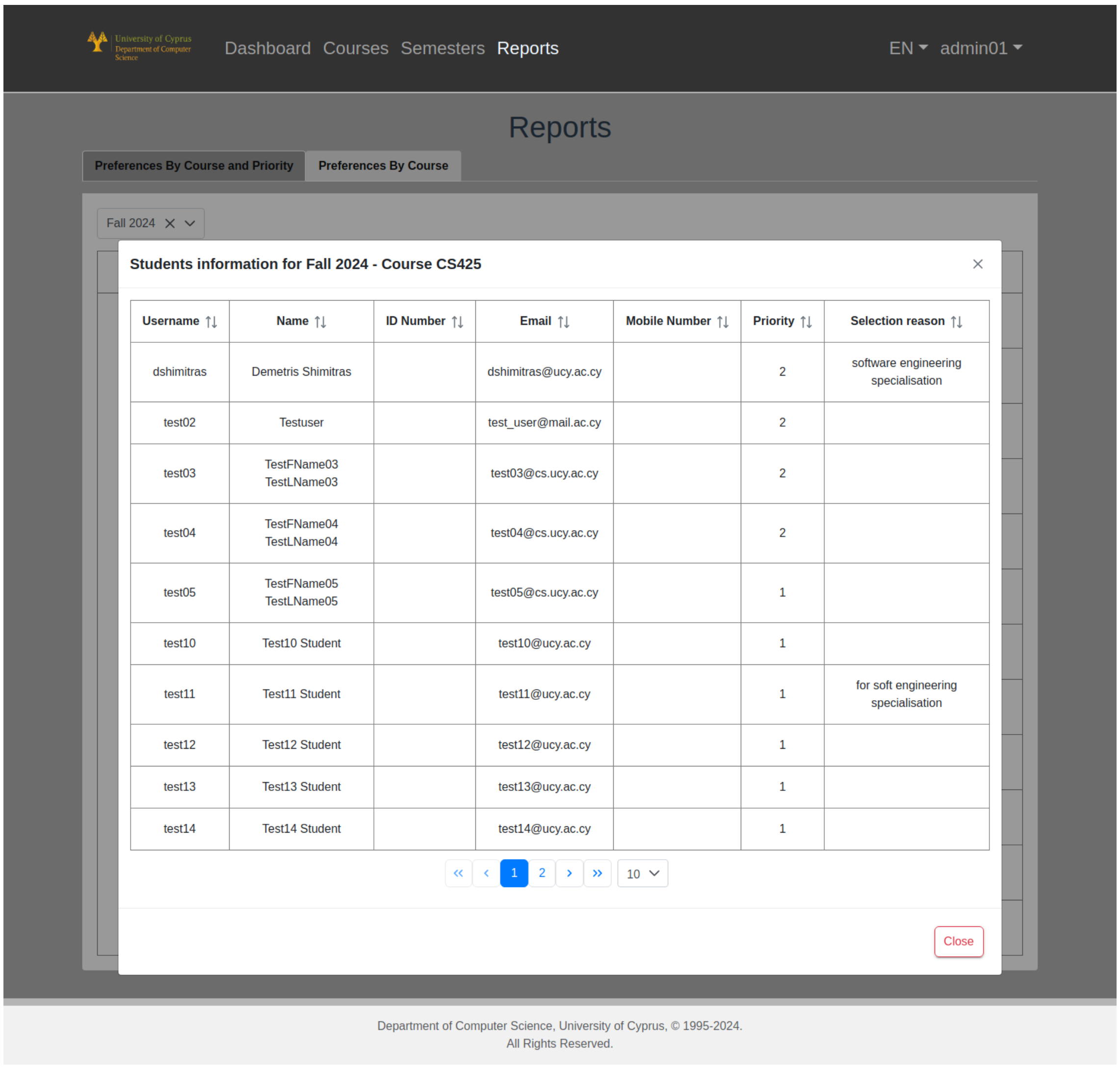
Preferences By Course and Priority
Preferences By Course
Fall 2024

Semester	Course	Total Preferences	Students
Fall 2024	CS442	30	View
	CS434	20	View
	CS425	15	View
	CS412	12	View
	CS422	11	View
	CS435	10	View
	CS423	10	View
	CS431	10	View
	CS441	10	View
	CS420	3	View
	CS421	1	View
	CS426	1	View

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-19. Total preferences per Semester and Course

Again, admin users can view the students' information that have made each selection. The priority each student has set their preference to is added to this data.


University of Cyprus Department of Computer Science
Dashboard
Courses
Semesters
Reports
EN
admin01
Reports
Preferences By Course and Priority
Preferences By Course
Fall 2024
Close

Students information for Fall 2024 - Course CS425

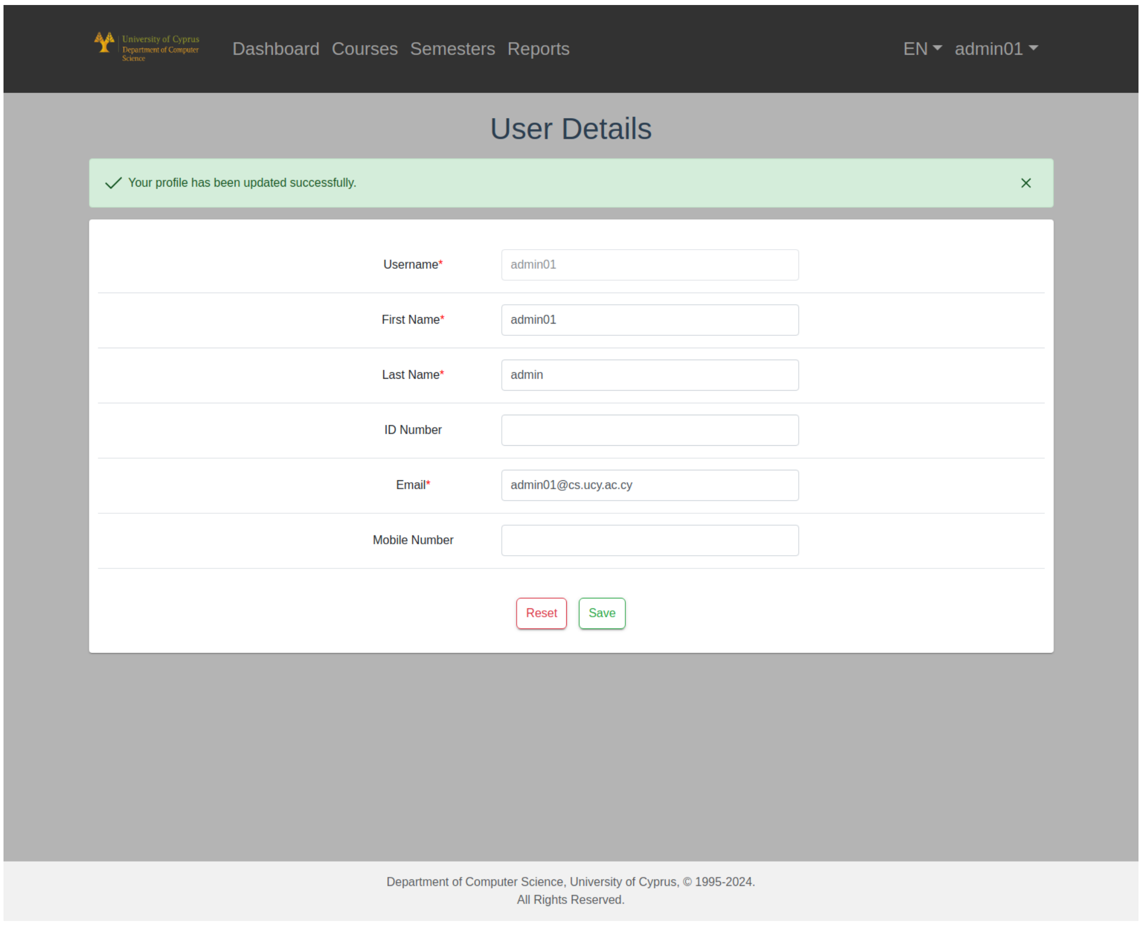
Username	Name	ID Number	Email	Mobile Number	Priority	Selection reason
dshimitras	Demetris Shimitras		dshimitras@ucy.ac.cy		2	software engineering specialisation
test02	Testuser		test_user@mail.ac.cy		2	
test03	TestFName03 TestLName03		test03@cs.ucy.ac.cy		2	
test04	TestFName04 TestLName04		test04@cs.ucy.ac.cy		2	
test05	TestFName05 TestLName05		test05@cs.ucy.ac.cy		1	
test10	Test10 Student		test10@ucy.ac.cy		1	
test11	Test11 Student		test11@ucy.ac.cy		1	for soft engineering specialisation
test12	Test12 Student		test12@ucy.ac.cy		1	
test13	Test13 Student		test13@ucy.ac.cy		1	
test14	Test14 Student		test14@ucy.ac.cy		1	

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-20. Student information for a Semester-Course preference

4.4.5 View and Update User Details

Faculty, much like students, will be able to change their personal information in the system through the User Details page. There are no academic details associated with faculty users, therefore only the name and contact information are available for change. As with other user data previously mentioned, an API integration can simplify this process, where all data will always be retrieved from the core systems of the Department and the University.



User Details

Your profile has been updated successfully.

Username*	admin01
First Name*	admin01
Last Name*	admin
ID Number	
Email*	admin01@cs.ucy.ac.cy
Mobile Number	

Department of Computer Science, University of Cyprus, © 1995-2024.
All Rights Reserved.

Figure 4-21. Faculty profile

4.4.6 Logout

Faculty users can log out of their account as well, redirected to their login page again.

Chapter 5

5. Future work

5.1 Integrations

A main area that can be considered as future work, which has been mentioned before in previous chapters, is related to integrations. Thinking of security, what comes to mind is the login and authentication process of this website.

5.1.1 Login authorisation

One of the major improvements related to security would have to be the connection of the login authentication using an API of the Computer Science Department, or the University of Cyprus. Such an integration would mean that the student account connected to the email or username is a verified account of an existing and active student of the University and the Department, leading to a more secure login on the website. The same would apply for the faculty users. Furthermore, this integration would mean that there will be no need for registration of any user in the system.

5.1.2 User data retrieval

Another benefit of integrating the solution with the University's or the Department's internal systems is to acquire other user data. The most significant information that can be used would be the students' academic details as well as some personal and contact data.

Some academic information that could help to improve the functionality of the system would be the sequence number of their active semester and the courses they have completed so far in their study. The first can be used to decide if a student will be allowed to submit preferences, for example first year students wouldn't need to, since prerequisites of many restricted elective courses are still not yet completed at that point. The latter is important to know which of the courses a student has succeeded in, for validating prerequisites when selecting one as a preference, in regard to whether it can be taken in the next semesters.

Personal information, such as a user's name and ID number can be retrieved from the core systems instead of given as input by the user themselves. Then, the information will always be accurate and up to date, as well as used to identify and validate that the right person is performing an action, like a validated student submitting a preference. Contact details of the students, but also for faculty users, including the email and mobile number, may be used to reach a student, for example to notify for a low-seat, or an admin user to send them a summary report of the students' selected preferences.

Therefore, an integration for user data retrieval could improve the overall system functionality and user experience, and simultaneously producing more accurate results.

5.1.3 Courses data

The existing Computer Science Department database for sure contains a variety of information regarding the provided courses. This is something that can become available through an integration.

As a consequence, the course data would not need to be filled in by Faculty users. The information would instead be retrieved through an integration with the department's database. The maintenance and needed updates related to the information of each course, such as the title, description, instructor, which semesters it would be offered in, could be minimised and reduce the user’s attention.

Additionally, the functionality of manually managing the courses through the new system could remain and be combined with this integration. The course data could be retrieved from the Department or University and stored locally to the system’s database. However, as this hybrid process can have a considerable impact on both sides if done at each point where the courses are needed, it can be instead carried out whenever a user logs in, or even when an administrator logs in specifically. If any information on these data should change, the administrator can still update the courses with the existing functionality.

Therefore, this can ensure that the courses data are not completely outdated, but also more importantly reduces the manual work that the administrators should do in the system.

5.2 Decision making algorithms

Possibly the most important suggestion for future work of this preference registration tool is the addition of an analysis and decision-making algorithm that would give suggestions to the faculty about the restricted elective courses registration of students.

A possible outcome of an analysis algorithm could be the suggestion of capacity and frequency of each restricted elective course, for a greater satisfaction between students. The student preferences can indicate how popular different courses are. Such a realisation can help the faculty decide how many students a course can accommodate at a time, or even how frequently the course could be offered.

The results of the decision-making algorithm could suggest which students get registered to each elective course. The registered student preferences could act as a measure of satisfaction, giving a different weight, a priority based on which the algorithm could give these suggestions. Considering the preferences of all students, a fair suggestion would ensure an average satisfaction rate between all students.

Furthermore, the suggestive algorithm could take into consideration the semester in which each student is. The preferences stated by more senior students that have less semesters left to complete their study, and therefore are going to graduate sooner than other students, could be given a higher priority. This could be applied for more criteria, such as the overall grade of each student, the number of completed ECTS, and so on. The declared course preferences given by students that are considered to have higher priority could be given a higher weight by the decision-making algorithm, having a bigger chance of being proposed in the registration schedule.

Based on some research, the following algorithms appear to possibly be good fits regarding the points mentioned above and are worth investigating for future work:

- Gale-Shapley Matching algorithm[18]
- TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution[19]
- Multiple Factor Weighted sorting[20]

Such an addition would have a high value for the faculty and the decision making of the Computer Science Department related to the restricted elective courses. However, students would benefit the most from this as they would be able to register more fairly to the elective courses that interest them, as well as more of them could achieve goals related to the specialisation they could attain.

5.3 Additional proposed functionality

Additional features could be implemented in the system that would enhance the user experience and improve the overall functionality. Some features that are proposed to be added are presented below.

5.3.1 Specialisation information and analysis

The Computer Science Department has a few specialisations defined based on specific required courses. When students complete a list of courses that satisfy the criteria of a specialisation, they can claim that specialisation to appear on their graduation degree.

With the information of which courses belong to each specialisation, the completed courses of a student, and their submitted preferences, the system can provide a breakdown of the specialisations of the Department.

In this breakdown, students could view all the specialisations that they already qualify for based on their completed courses. For other specialisations the system could inform them on how many more courses are needed to be able to claim them, and which of the courses would do so.

5.3.2 Report document export

A functionality that can be useful for the faculty administration is some kind of reporting tool that presents specific (static) data, or even dynamic data based on parameters that can be filled in by a faculty user, in the form of a report document. These can be in the form of a CSV or Excel file, or a formatted document, like a word processing document or a PDF.

Report documents based on specific queries, of which the following are proposed:

- Preferences per semester
- Preferences per student
- Preferences per course
- Preferences per combination of two or more of the parameters; semester, student, course

Report documents or exports of information available in the web application can be very useful for the faculty, in various ways.

Chapter 6

6. Conclusion

6.1 Summary

In conclusion of this thesis report, the purpose of this project was to provide a solution for an issue that persists during the registration period through the semesters in the Computer Science Department of the University of Cyprus. Students struggle to claim a spot for restricted elective courses they would like to attend in the semesters, due to the low number of seats offered and other students of higher credit, who have registration priority, already claiming those seats. This results in the students taking on classes that they would not be normally interested in, or not being able to achieve goals, such as qualifying for a specialty upon their graduation.

This web application system has been developed with the intention of improving the scheduling of the courses offered and providing a smoother registration process to the students. To do so, the students will be able to log in the new system and put in their preferences; which restricted elective courses they would prefer to be attending in the next semesters. Administration could then review and accumulate these preferences to gain an insight of the most demanded courses by the students. Based on that, it could be decided whether to make changes on the initial schedule, such as adding extra classes for a high-demand course, offering the course in extra semesters, and so forth.

Eventually, more updates and additional functionality would improve the impact on the desired result of this new system. An ideal function could even automate, in some way, the scheduling of the courses, or at least minimise the effort being put into creating it.

Hopefully, all the above will help to mitigate the persisting issue faced with the restricted elective courses.

References

[1] MongoDB. “The Most Popular Database for Modern Apps.” MongoDB, 2019, www.mongodb.com/.

[2] OpenJS Foundation. “Express - Node.js Web Application Framework.” Expressjs.com, 2017, expressjs.com/.

[3] You, Evan. “Vue.js.” Vuejs.org, 2014, vuejs.org/.

[4] Node.js. “Node.js.” Node.js, 2023, nodejs.org/en.

[5] NPM. “Npm | Build Amazing Things.” Npmjs.com, 2019, www.npmjs.com/.

[6] “SPA (Single-Page Application) - MDN Web Docs Glossary: Definitions of Web-Related Terms | MDN.” Developer.mozilla.org, developer.mozilla.org/docs/Glossary/SPA.

[7] Wilson, Doug. “Body-Parser.” Npm, 26 Apr. 2019, www.npmjs.com/package/body-parser.

[8] Mongoose. “Mongoose ODM V5.8.2.” Mongoosejs.com, 2011, mongoosejs.com/.

[9] “Express-Session.” Npm, www.npmjs.com/package/express-session.

[10] Hanson, Jared. “Passport.js.” Passport.js, www.passportjs.org/.

[11] “Pino.” Npm, 13 May 2024, www.npmjs.com/package/pino. Accessed 13 May 2024.

[12] You, Evan, and Eduardo San Martin Morote. “Vue Router.” Router.vuejs.org, router.vuejs.org/.

[13] “What Is Vuex? | Vuex.” Vuex.vuejs.org, vuex.vuejs.org/.

[14] “PrimeVue | Vue UI Component Library.” Www.primefaces.org, primevue.org/.

[15] Bootstrap. “Bootstrap.” Getbootstrap.com, 2022, getbootstrap.com/.

[16] Axios. “Getting Started | Axios Docs.” Axios-Http.com, 2023, axios-http.com/docs/intro.

[17] Kawaguchi, Kazuya. “Vue I18n.” Vue-I18n.intlify.dev, vue-i18n.intlify.dev/.

[18] Gale, D. and Shapley, L.S. (1962). College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1), p.9. doi:https://doi.org/10.2307/2312726.

[19] Soczewica, R. (2021). *What is TOPSIS?* [online] Medium. https://robertsoczewica.medium.com/what-is-topsis-b05c50b3cd05.

[20] Stack Overflow. (n.d.). *How to provide most relevant results with Multiple Factor Weighted Sorting*. https://stackoverflow.com/questions/8760570/how-to-provide-most-relevant-results-with-multiple-factor-weighted-sorting