DOKUZ EYLÜL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

<STUDENT REGISTRATION SYSTEM>

CME 3201 Database Management Systems Term Project Report

Final Upload

2020-2021 FALL

2017510067	Şefika Özlem PUL	sefika.pul@ceng.deu.edu.tr
2017510089	Seçil ÖZTÜRK	secil.ozturk@ceng.deu.edu.tr
2016510083	Vedat ÖZCAN	vedat.ozcan@ceng.deu.edu.tr

Table of Content

Phase – I

Abstract

The web application we have designed is briefly a student registration system. The main purpose of the application is to save the student's personal information, the course information that students is enrolled in and the instructor, grade and absentee information of these courses in the database and present them to the users when necessary.

Due to a global epidemic, it is safer for our health to register remotely via the internet rather than face to face. Thanks to the application, we provide these opportunities to users. This application is designed for convenience for students, and for teachers. In this way, users will be able to easily access the information they want.

1. Introduction

The Student Registration System mainly exists to present information about the student. The system records the information about the student and presents the information to the users. The system also keeps information about courses and lectures. There are students and lecturers' entries in the system. In addition all this, the system has an admin and the basic function of the system executed by this admin.

Students can register for the course they want by entering their information, see the grades they have received from the courses and track their absences.

Lecturers recorded in the course can enter the scores and absenteeism information of their students into the system.

Keeping students' information in written files on paper while recording students can cause problems in cases where the information is lost in the future and requires updating later. In practice, data is kept securely in the system and updates are made easier and faster when necessary. In addition, when registering anywhere due to the global epidemic that has occurred today, it has been easier for everyone to register remotely on the internet instead of face-to-face.

2. Purpose and Benefits

The purpose of the application is to enable users, i.e. students, to easily access their information. For example, exam grade information and absenteeism. It is also prepared for teachers to enter and update student grades and absenteeism information. Student information can be easily kept and updated. Students can choose the course they want.

The users are offered the opportunity to access their information safely and accurately whenever they want. It will also be useful for students and lecturers to easily use from their homes due to the global epidemic. For these reasons, it is an application that will benefit users greatly.

3. Mode, Medium, and Environment

This application will be used on the “Web environment”. Mode number of the application in our project will be three. One of the modes in the application is the student mode. The student can register for the course and the lecturers in the course by entering the required information, and they can see the absenteeism and the scores they have received from the courses. Another mode is the teacher mode, where the teacher can access the information of the students recorded in the lecture, enter the exam scores and share the lecture grade related to the lesson. Another mode of ours is admin mode. Admin confirms student enrollment, can make lecturer registration, and update or delete when necessary. In short, it provides management of the system.

There are three entries on the homepage of the application, namely student, lecturer and administrator. If she/he is not registered in the system, he/she enters the information requested from the membership section, selects the password and the type of student or lecturer, and completes the registration to the system. If the student mode is selected, the student number is given by the system. The student enters the student entrance with the student number and the password that he / she has determined. You can register for the course he wants to register, can see the exam score, the absenteeism information and the lecture notes sent by the lecturer. The lecturer enters the entrance by typing the name and password of the lecturer. The administrator can log in by typing his name, password and email. The administrator can view the information of everyone registered in the system and make updates when necessary.

The application will be designed to work on the web.

4. Functionality

Depending on the basic interface and some components of the project, the project team determined the functions to be used in the project's functionality. We can divide these functions into three groups according to the type of user who performed them. All functions are listed below and will be displayed graphically by user type in the following part of the report.

4.1 Functional Requirements

Basic System Functions

- Sign in
- Sign out
- Registration
- Search student
- List students
- Add student
- Edit student’s details
- Delete student
- Search course/section
- List courses/sections
- Take course/section
- Add lecturer
- List lecturers
- List lecturer’s sections
- List lecturer’s students

4.2 Non-Functional Requirements

In the Student Registration System, the student uses the username and password information while logging in and there is no restriction on the number of students registered in the system. In the system, a student cannot change another student's information or delete another student. Besides, the student cannot delete or update courses and sections. On the other hand, admin and lecturers log in to the system using their email and passwords. The instructor can only change the student's absenteeism and grade information.

5. High Level Organization

5.1 Components

- Student component

This is the most important component of the Student Registration System. This component can have various sub-tasks. It plays a big role in the interaction of other components with each other.

- Lecturer component

It is related to Section and student tables and it is obliged to update some features of students.

- Department component

Most components contain the property of this component.

- Course/Section component

This component shows the activities of students in the school system. Besides, it indirectly indicates the relationship between students and instructors.

- Administration

Administration module controls the system in general. It regulates the operation of the system as a whole. It regulates the relationship of the other components.

- Authentication and user management component

This is the major sub system that is responsible for the security of the Student Registration System. It authenticates users and also handles the user management activities such as creating new user accounts, removing accounts from the system etc.

- Public component

This is a relatively small subsystem compared to the other components of the Student Registration System. This is the component which is responsible for the guest viewer (public) to view student results.

5.2 Interfaces

- Student component
    - ***studentProfile:*** This interface is used for the student to view and edit some personal information.
    - ***editProfile:*** The student can change their features such as email, phone and password by entering their profile page.
    - **takeSection:** This interface will provide the available section list for the semester. This allows the student to choose the subjects. This interface is the bridge between the student component and the lecturer component.

- Lecturer component
    - ***lecturerPanel:*** This interface is used to view the sections of the lecturer and to manage some of the information of the students registered in the lecturer course.
    - ***listOwnSection:*** The lecturer can view the information of her/his own sections and section details if desired.
    - ***viewSectionDetails:*** The lecturer updates the absenteeism and grade information of the students registered in her/his own sections**.**
    - ***updateSectionDetails:*** The lecturer displays the information of students enrolled in their course. It updates the absenteeism and grade information when necessary.
    - ***listOwnStudents:*** The lecturer lists all students enrolled in own course.

- Course/Section component
    - ***listCourses/listSections:*** This interface lists all courses and all sections linked to these courses.

- Administration
  - ***adminPanel:*** This interface is used for the system admin to view and manage student information. It can also add a new lecturer.
  - ***listStudents:*** The information of the students registered in the system is displayed, the information update, search student by name and the student can be deleted if desired.
  - ***searchStudentByName:*** If too many students are listed, Admin can search by student's name.
  - ***listCourses:*** General information about the sections in the system is displayed and detailed information of sections can be accessed if desired.
  - ***viewCourseDetails:*** If the course has multiple sections, the Admin displays the information about those sections.
  - ***listLecturer:*** The information of the lecturers in the system is displayed.
  - ***addNewLecturer:*** The Admin adds a new lecturer to the system.

- Authentication and user management component
  - ***studentLogin:*** The part where the student logs into the system.
  - ***studentRegister:*** The part where the student registers to the system with his/her own personal information.
  - ***adminLogin/lecturerLogin:*** The part where the admin and lecturers logs into the system.

6. Functionality vs Polish

The project crew determined the duration of the application as 12 weeks. This period may be shorter or longer. Problems in practice may prolong this period. However, the project team are planning to complete the application within the estimated time. Two weeks application design, 8 weeks application development, 2 weeks testing. Finally, there may be additions and deletions to the application. This may cause the application time to change.

7. Milestones and Timeline

Group members will take an active role in all stages of the project as decided on the subject of the project.

Member Weeks	Şefika Özlem Pul	Seçil Öztürk	Vedat Özcan
Week 1	Finding a project and planning	Finding a project and planning	Finding a project and planning
Week 2	Reporting	Reporting	Reporting
Week 3	Create a diagram	Create a diagram	Create a diagram
Week 4	Code design	Code design	Code design
Week 5	Creating variables and functions	Creating variables and functions	Creating variables and functions
Week 6	Creating classes	Creating classes	Creating classes
Week 7	Coding and reporting	Coding and reporting	Coding and reporting
Week 8	Error correction and coding	Error correction and coding	Error correction and coding
Week 9	Coding and reporting	Coding and reporting	Coding and reporting
Week 10	Debugging and coding	Debugging and coding	Debugging and coding
Week 11	Debugging	Debugging	Debugging
Week 12	Reporting	Reporting	Reporting

Figure 1: Timeline of the project

Phase – II

8. Requirement Specification

Software Requirements

HTML5, CSS3 and JavaScript will be used in the front end of this Project. In addition, some CSS libraries such as Bootstrap will be used.

Php and MySQL will be used on the back end of the project. XAMPP software offers these technologies as a whole. XAMPP is simply web server software. Generally, the part where it is used is to simply create a web server on your computer, handle the Apache, PHP and MySQL installations at once and manage them through a single program.
Besides, CodeIgniter will be used in the project. CodeIgniter is a free, open-source, easy-to-use, object-oriented PHP web application framework, providing a ready-to-use library to use with your own PHP applications. For example, there is a Database API to make it easier and more convenient to execute SQL queries, such as SELECT, UPDATE, DELETE, INSERT, etc., without having to create a lot of repetitive code yourself. This is how an application framework is useful in application development.

Teammates can use different development environments, but generally Visual Studio Code and Sublime Text will be used.

9. Design Specifications

9.1 Architectural Styles

The Student Registration System will be developed under two main architectural styles. Development of the project will be done in MVC architectural style and also Three tier Client/Server Architecture

Three Tire Architecture

Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.

The chief benefit of three-tier architecture is that because each tier runs on its own infrastructure, each tier can be developed simultaneously by a separate development team and can be updated or scaled as needed without impacting the other tiers.

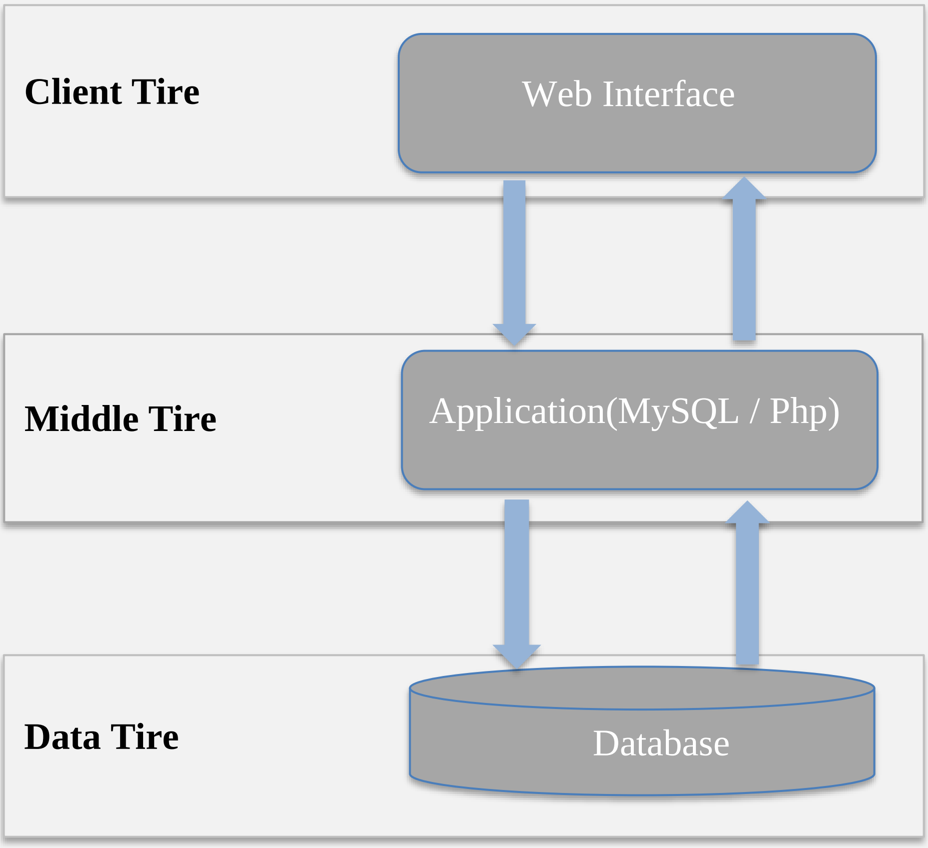


Figure 2: Three Tire Architecture

MVC (Model-View-Controller)

Model–view–controller (usually known as MVC) is a software design pattern commonly used for developing User interface that divides the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

Model: The central component of the pattern. It is the application's dynamic data structure, independent of the user interface It directly manages the data, logic and rules of the application.

View: Any representation of information such as a chart, diagram or table. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

Controller: Accepts input and converts it to commands for the model or view

In addition to dividing the application into these components, the model–view–controller design defines the interactions between them.

- The model is responsible for managing the data of the application. It receives user input from the controller.
- The view means presentation of the model in a particular format.
- The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

The project crew will use this MVC Style for the Student Registration System because, there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown. In some software systems the code between the process logic and interface are mixed. This will reduce the modularity of application and make the system more difficult to maintain. To avoid this problem we have decided to use MVC architectural style to separate the application logic with the interface. The main advantage of this is style allows the data to change independently of its representation and vice versa. Support presentation of the same data in different ways with changes made in one representation shown all of them.

9.2 Database Design and Diagrams

9.2.1 Database Design

ER Diagram

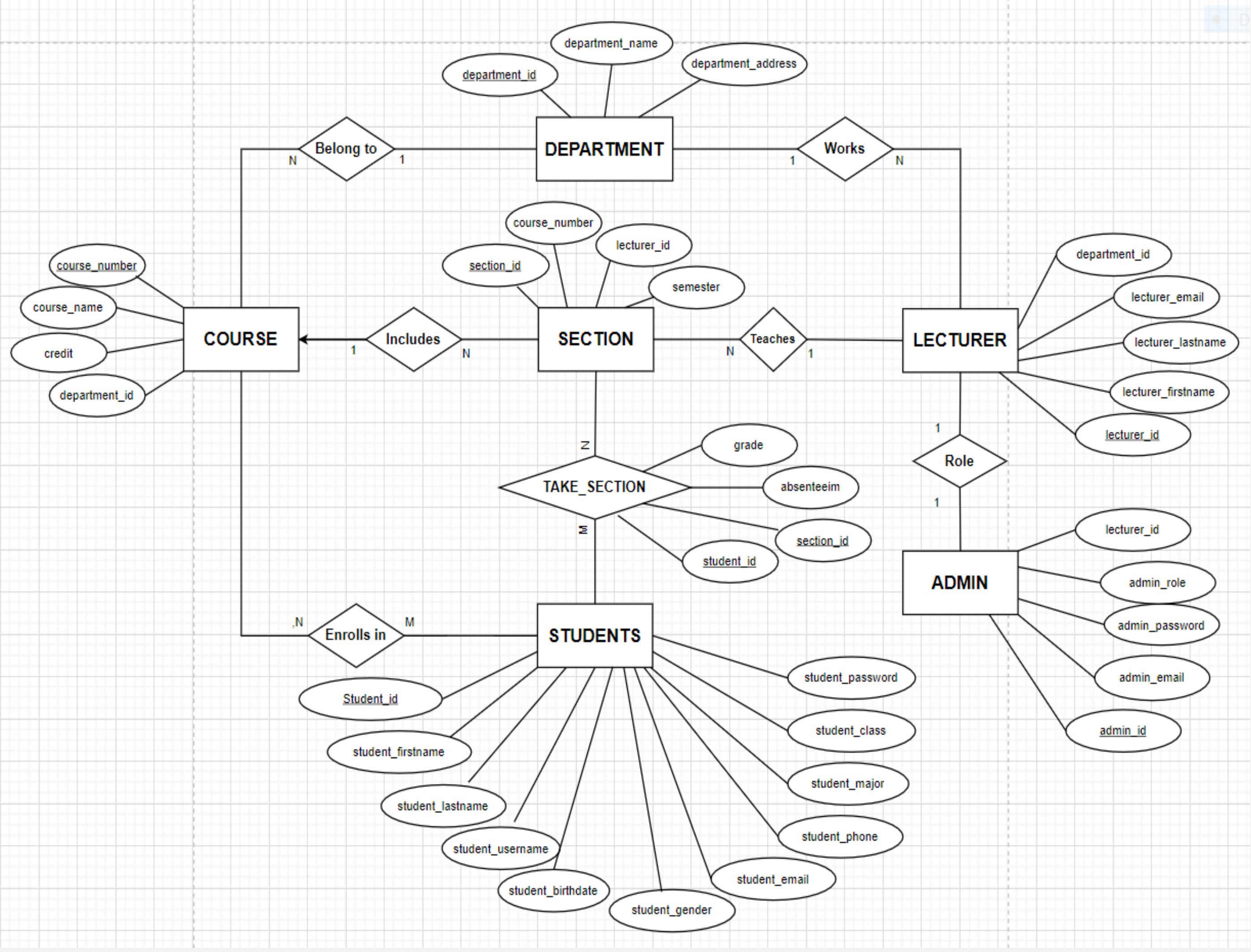


Figure 3: Entity Relationship Diagram

This diagram shows the objects in a student registration system, their properties and their relationships with each other. In this system, there are various departments and multiple courses belonging to each department. There are multiple sections for each course and multiple students enrolled in each course. More than one instructor can work in each department and each instructor can teach more than one section. In addition, each student can take more than one section and more than one student can enroll in each module. Admin and lecturer relationship is based on role feature. If the related role value is 0, the Admin logs into the system. If it is 1, the Lecturer logs into the system.

Relational Diagram

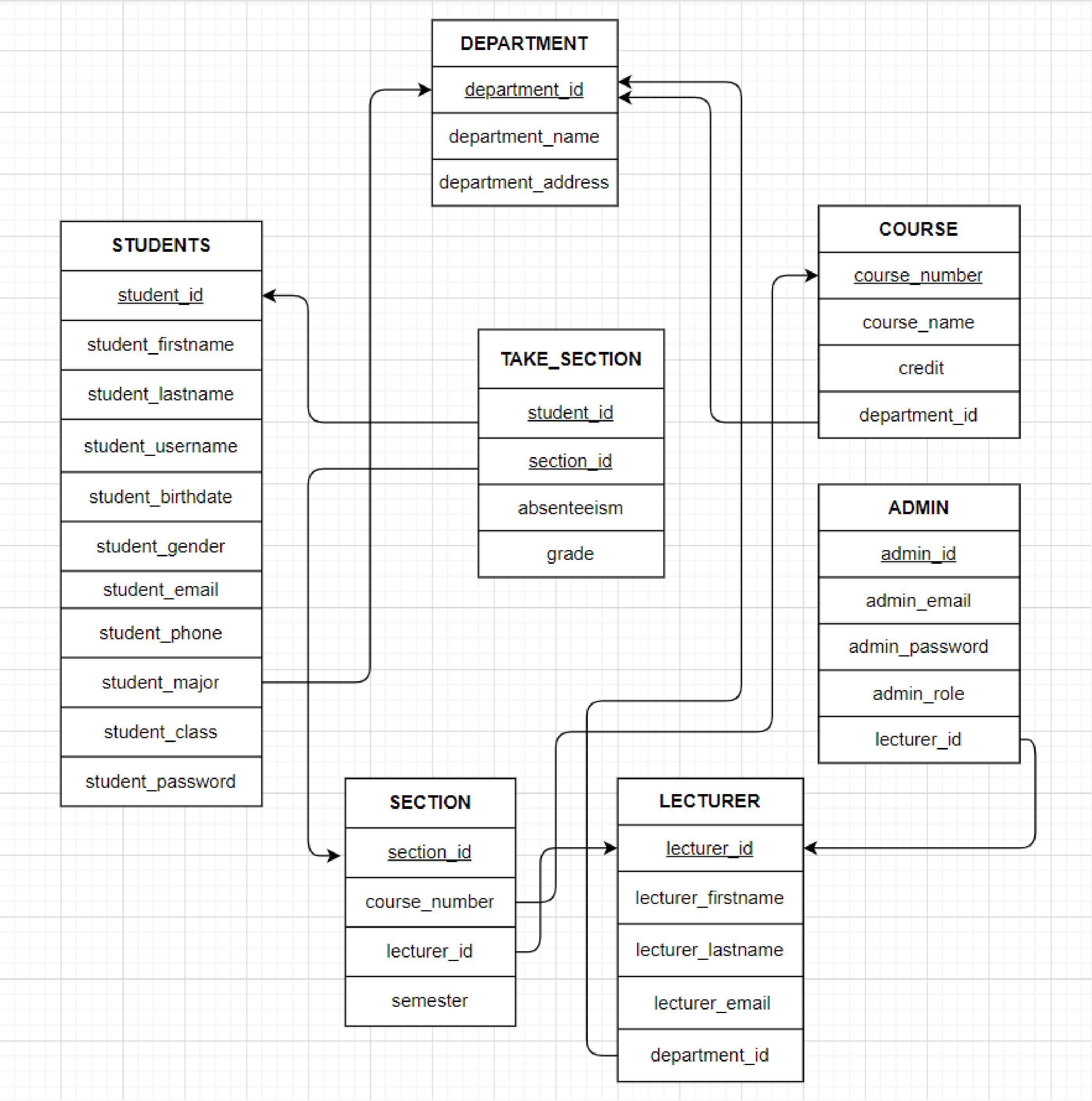


Figure 4: Diagram showing the relationship between entities

9.2.2 Use Case Diagram

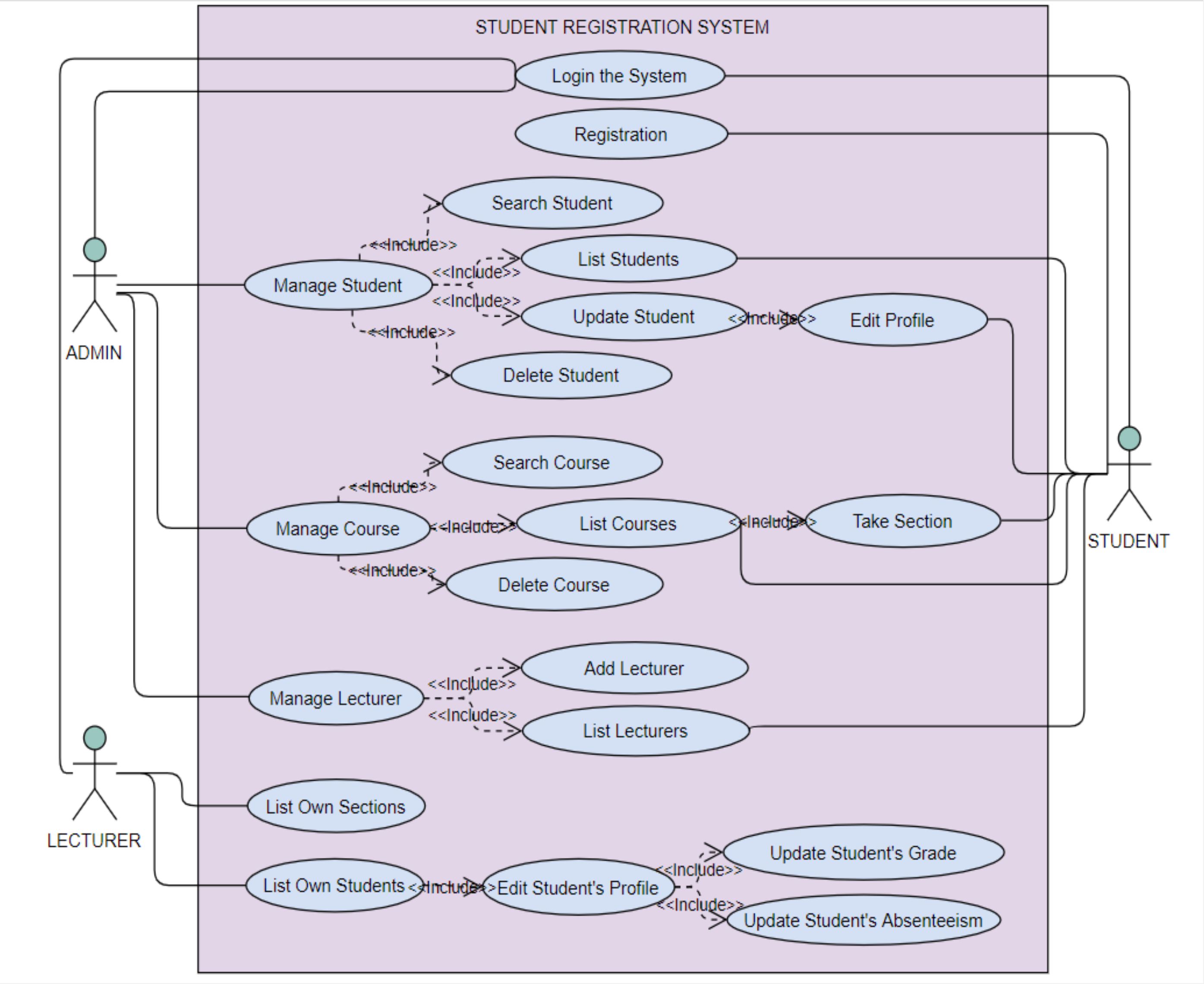


Figure 5: Use Case Diagram

9.2.3 Hierarchy Diagram

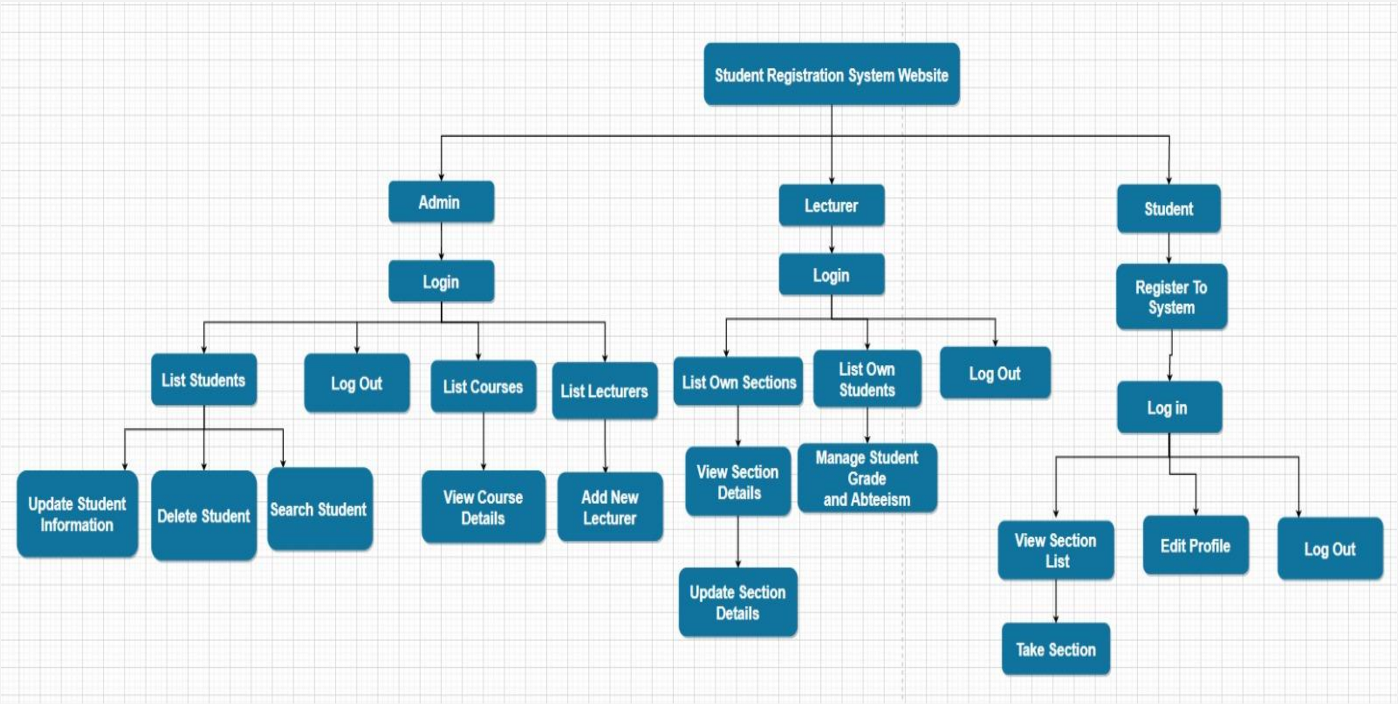


Figure 6: Hierarchy Diagram

10. Interfaces

The Student Registration System consists of many pages. These pages describe the components of the system and provide information about them. The project team successfully coded all the design pages decided and drawn before the implementation phase.

10.1 Multiple Login Page

This page is the first page of the system and offers access according to different user types.

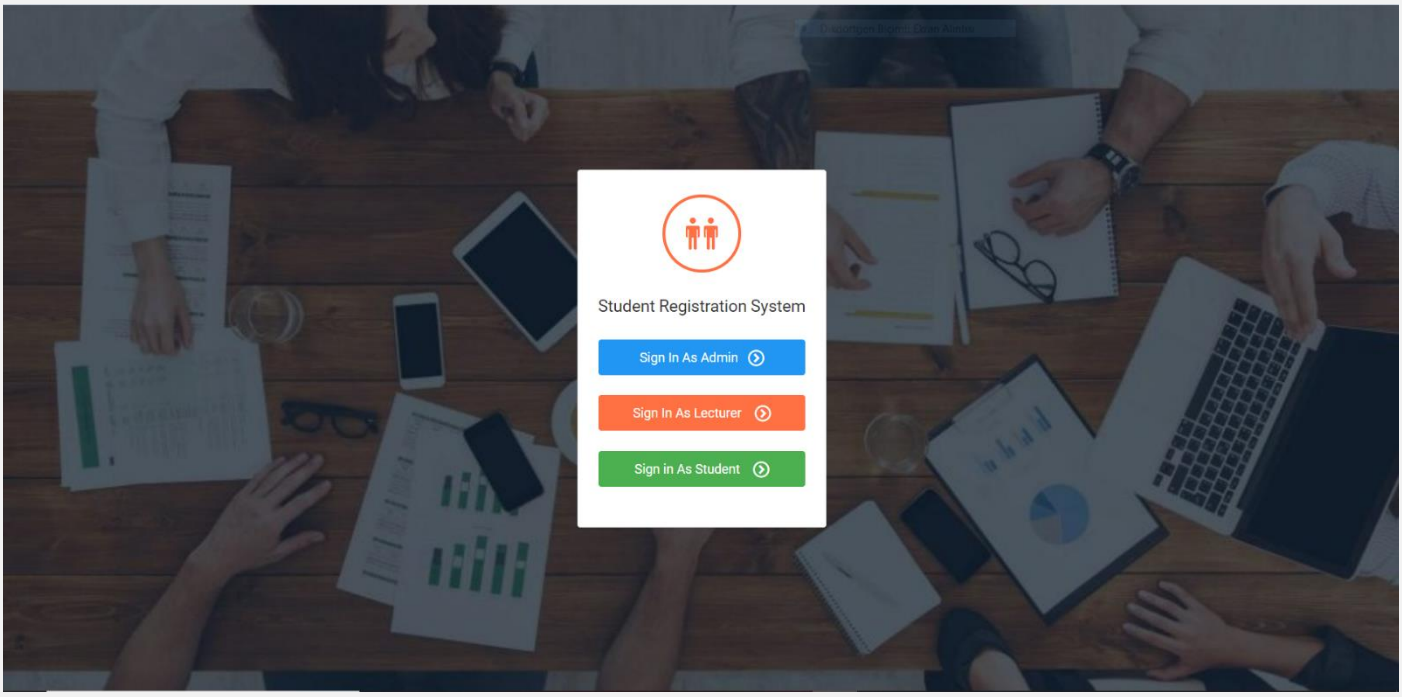


Figure 7: Multiple login page

10.2 Login/Register Page

This page is the page where users register and log in to the system. Students first enter the registration page and register to the system, then enter the login page and log into the system with their username and password. Admin and Lecturer log into the system according to their role.

If the defined role is 0, the Admin logs in. If the role is 1 the Lecturer logs in. The Admin and lecturers logs with their e-mail and password.

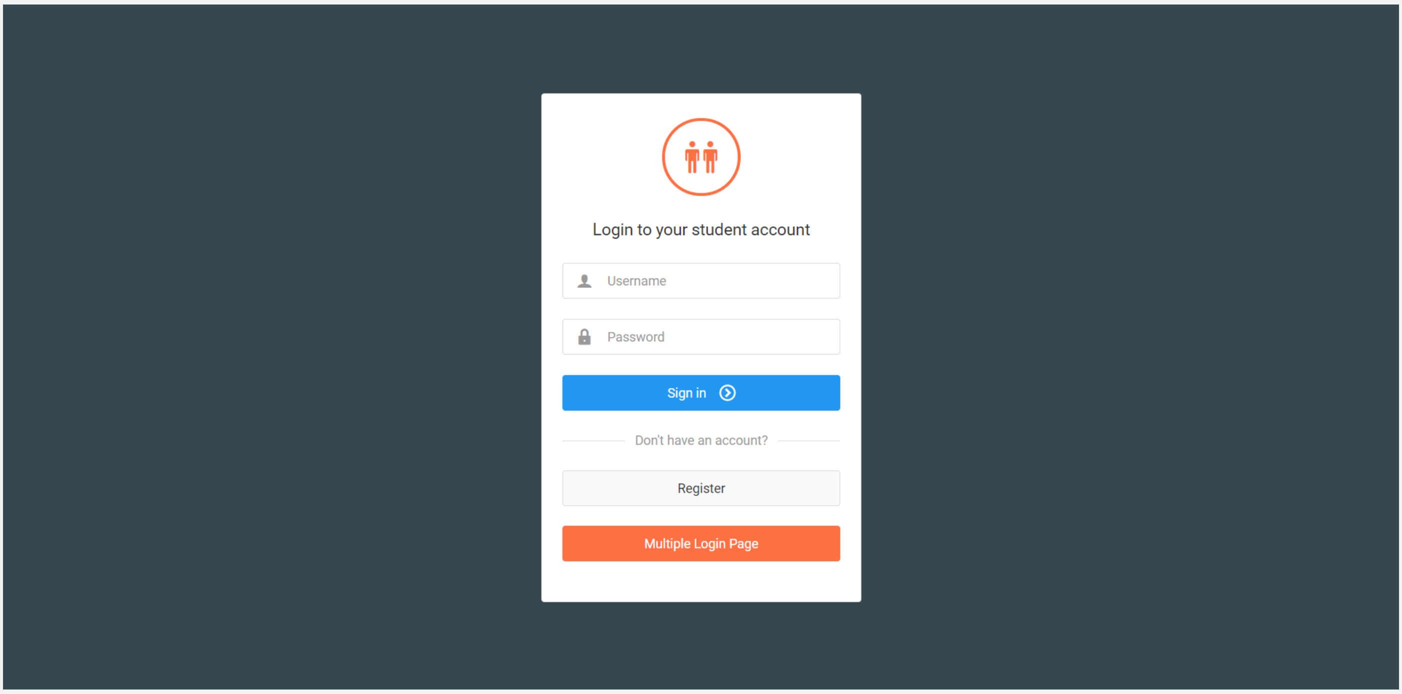
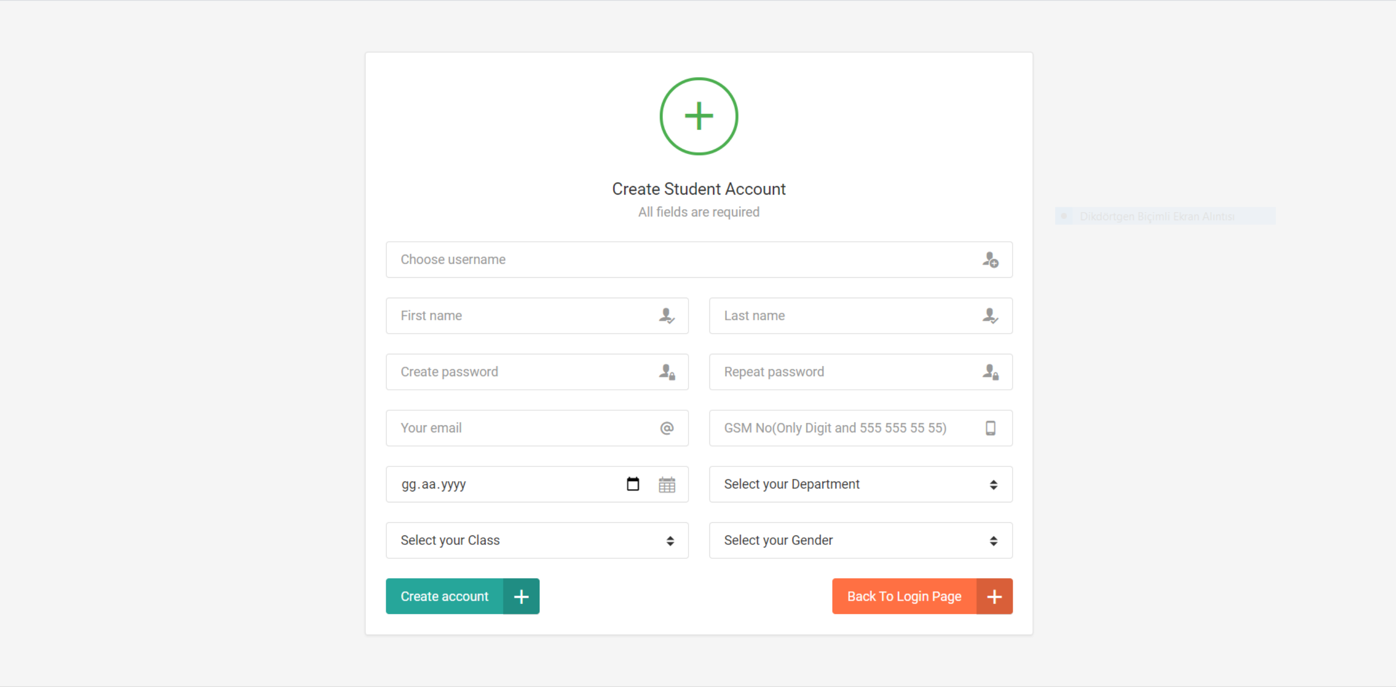


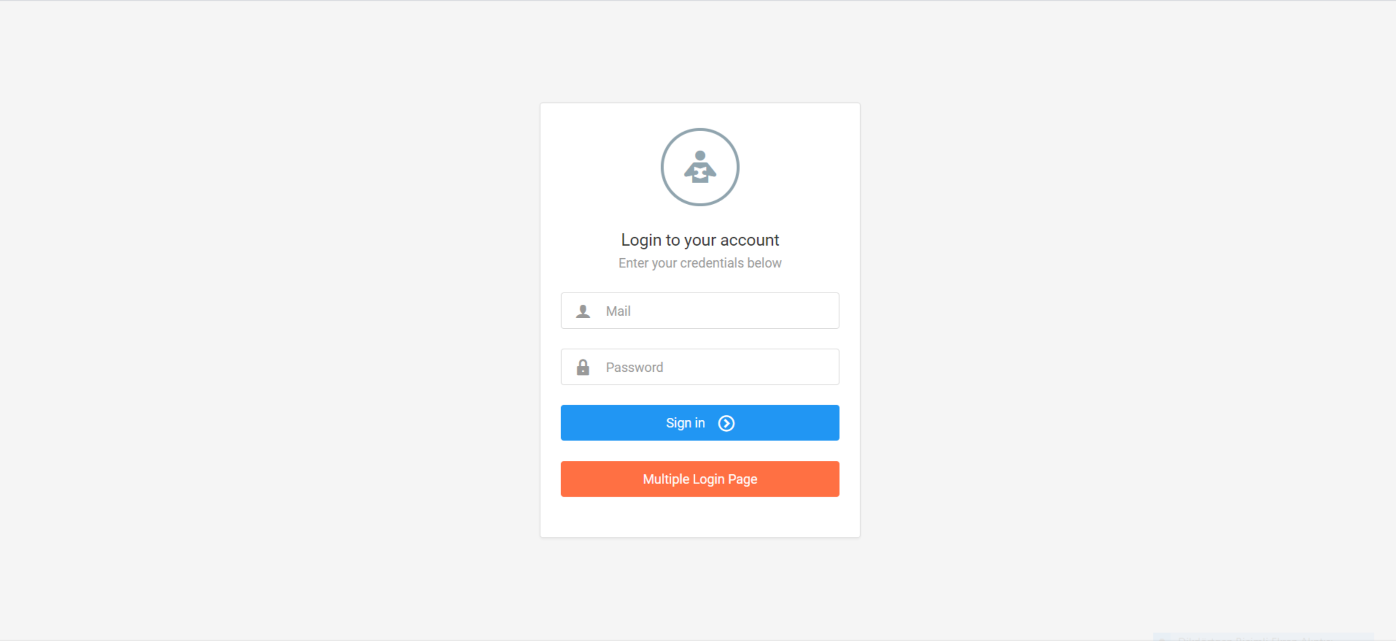
Figure 8: Student login page



Create Student Account

All fields are required

Figure 9: Student register page



Login to your account

Enter your credentials below

Figure 10: Admin/Lecturer login page

10.3 Home Page

This page contains all the components in the system. When students log into the system, they view this page and starts seeing some pictures about the Student Registration System from the sliding panel. Students can get information about the system and communicate from the navigation bar of this page. Besides, they can view courses, lecturers and other students. The student who wants to exit the system returns to the multiple login page by pressing the exit button on the bar.

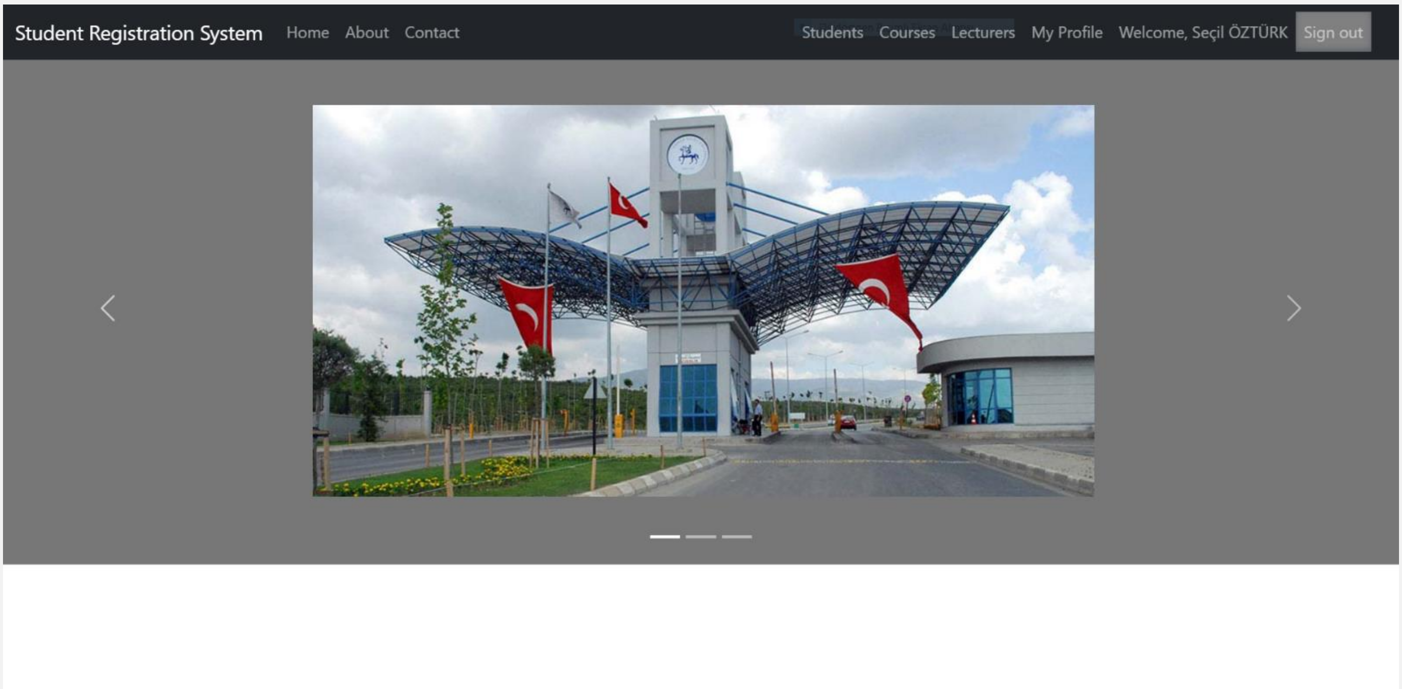
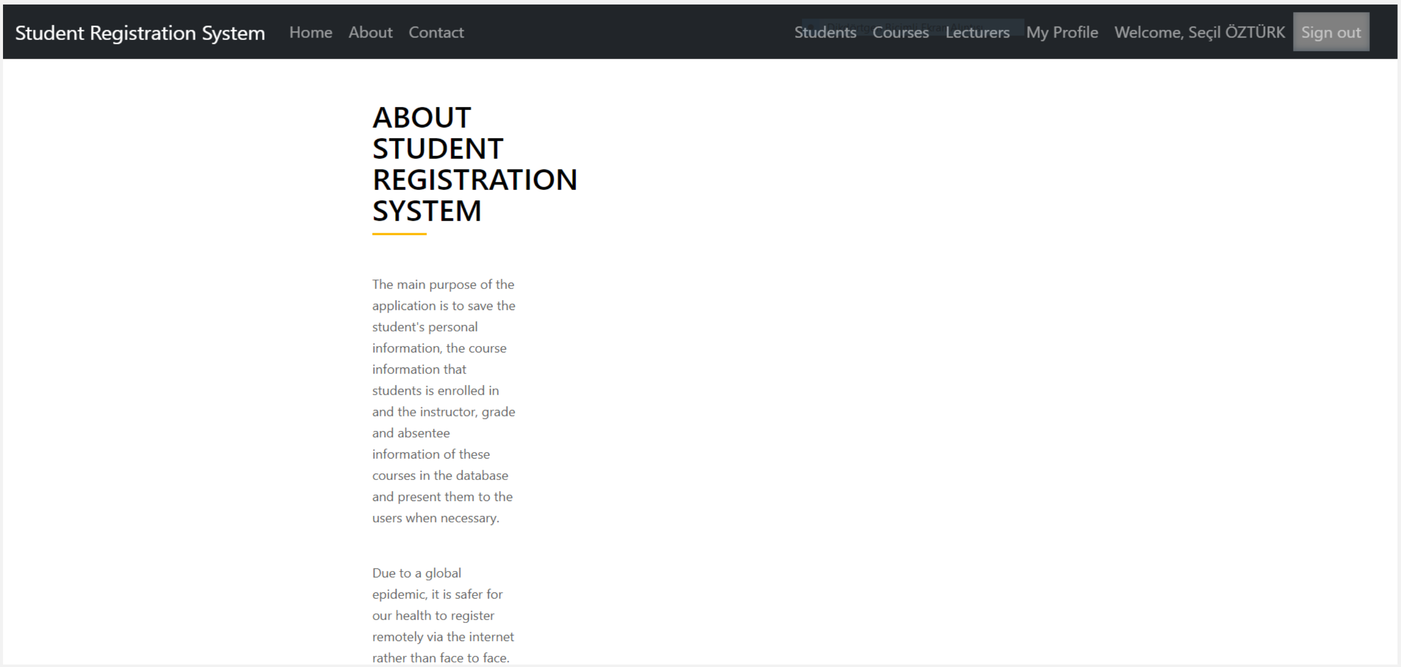


Figure 11: Home page

10.3.1 About Page

This page gives general information about the system.



ABOUT STUDENT REGISTRATION SYSTEM

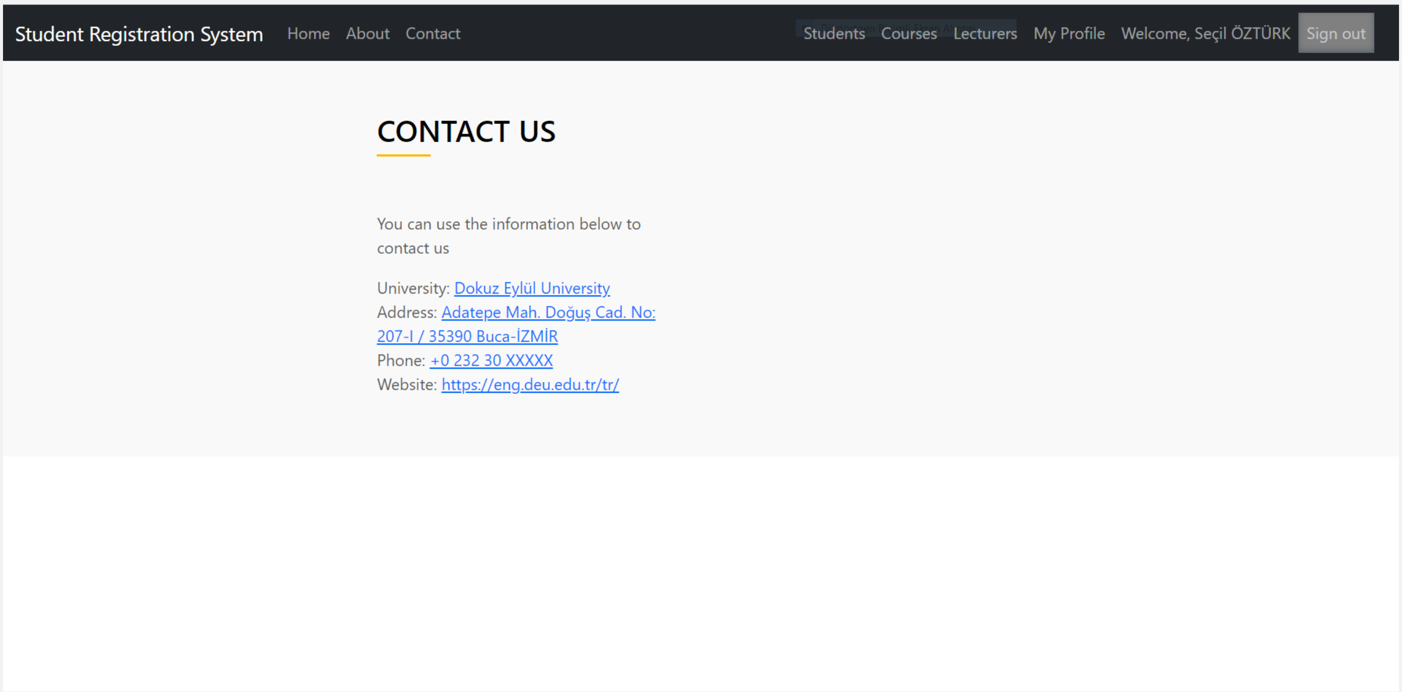
The main purpose of the application is to save the student's personal information, the course information that students is enrolled in and the instructor, grade and absentee information of these courses in the database and present them to the users when necessary.

Due to a global epidemic, it is safer for our health to register remotely via the internet rather than face to face.

Figure 12: About page

10.3.2 Contact Page

It contains some contact information.



CONTACT US

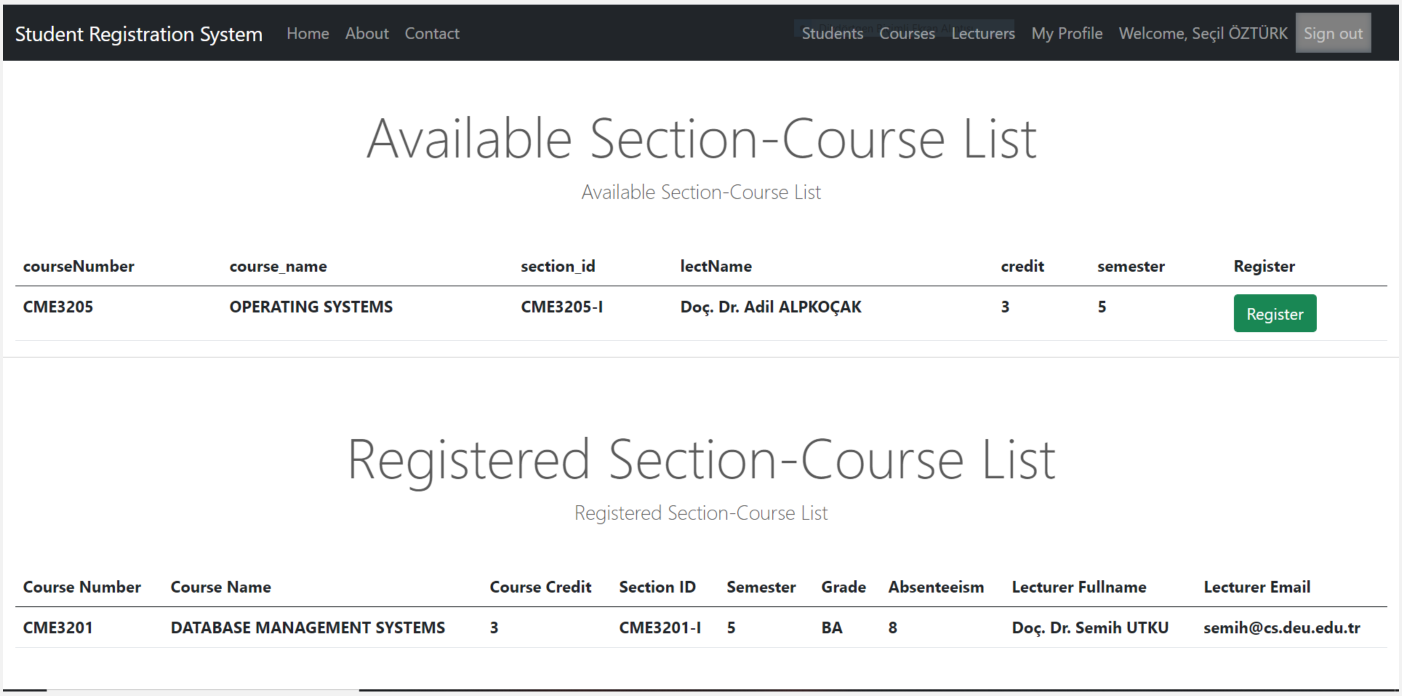
You can use the information below to contact us

University: Dokuz Eylül University
Address: Adatepe Mah. Doğuş Cad. No: 207-I / 35390 Buca-İZMİR
Phone: +0 232 30 XXXXX
Website: https://eng.deu.edu.tr/tr/

Figure 13: Contact page

10.3.4 Courses Page

Courses and their features are shown to the user on this page. Various functions such as enrolling a course, and seeing the sections related to the course are defined on this page through the buttons.



Available Section-Course List

Available Section-Course List

courseNumber	course_name	section_id	lectName	credit	semester	Register
CME3205	OPERATING SYSTEMS	CME3205-I	Doç. Dr. Adil ALPKOÇAK	3	5	Register

Registered Section-Course List

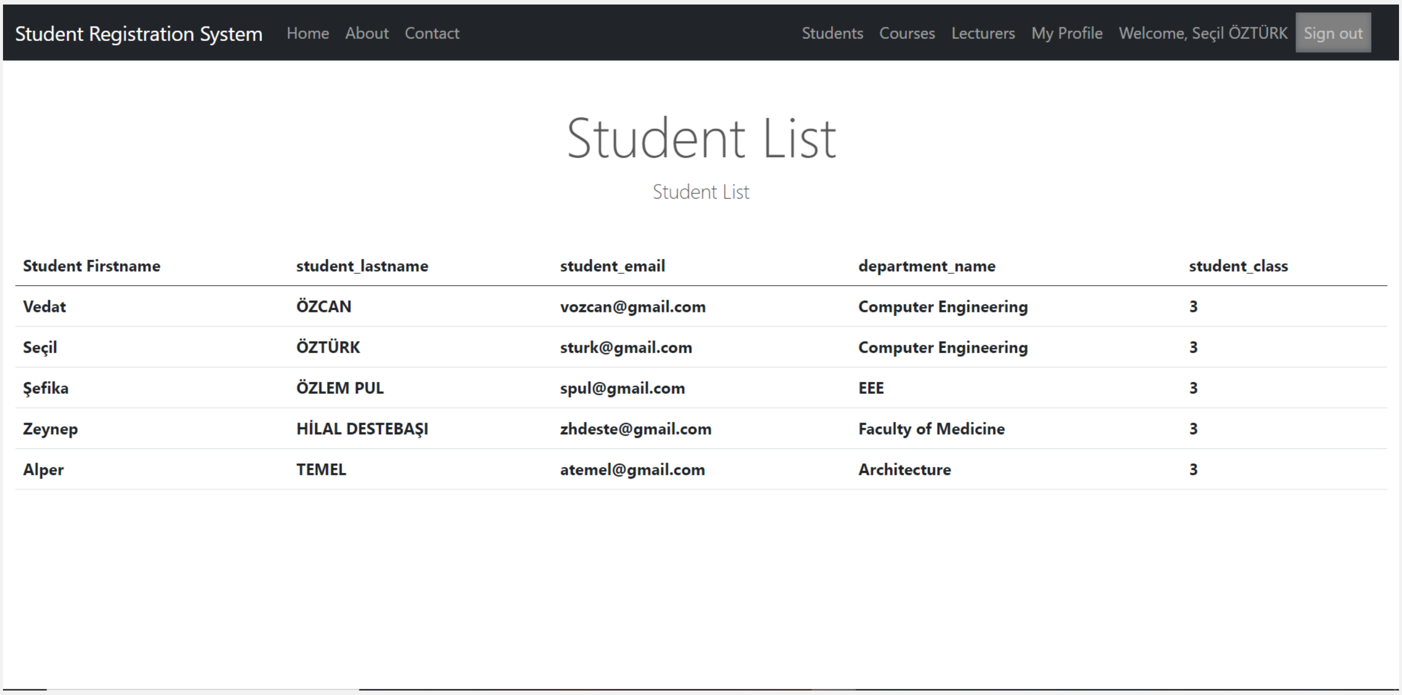
Registered Section-Course List

Course Number	Course Name	Course Credit	Section ID	Semester	Grade	Absenteeism	Lecturer Fullname	Lecturer Email
CME3201	DATABASE MANAGEMENT SYSTEMS	3	CME3201-I	5	BA	8	Doç. Dr. Semih UTKU	semih@cs.deu.edu.tr

Figure 14: Courses/Sections list

10.3.5 Students Page

On this page all student registered in the system are presented to the user as a list with some information.



Student List

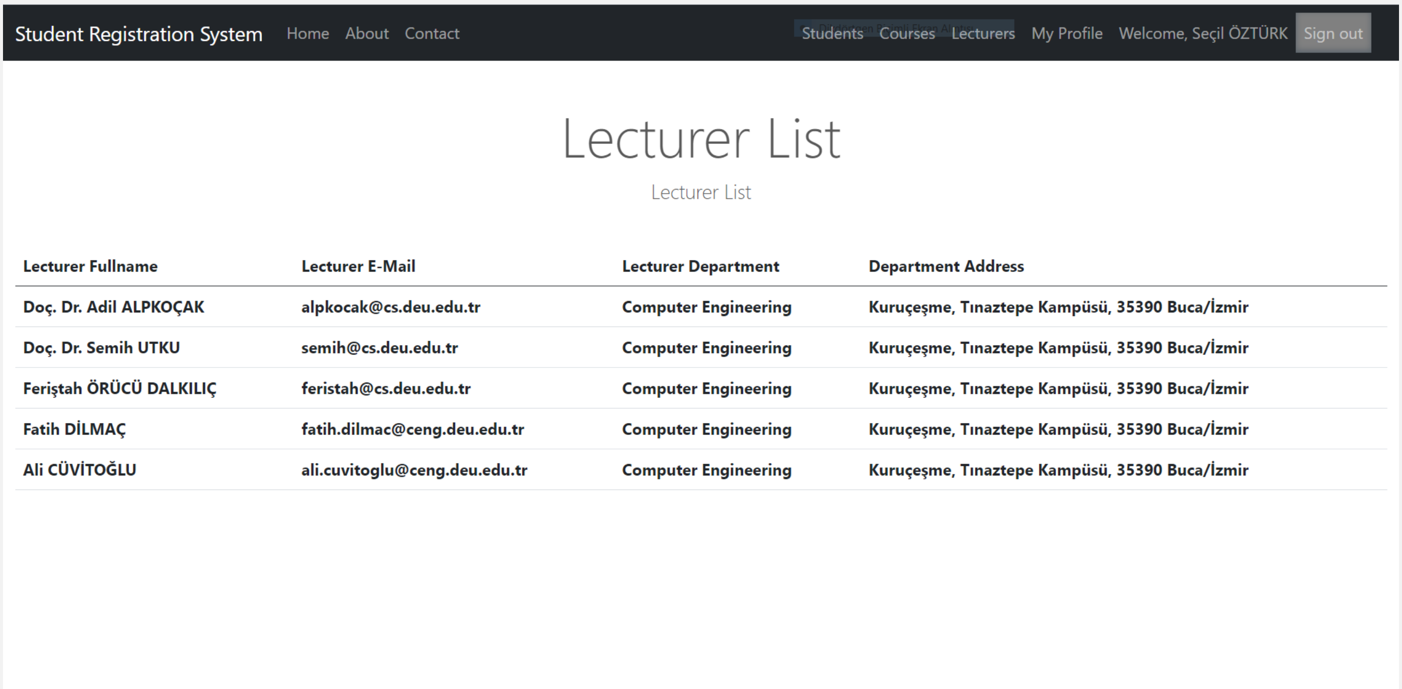
Student List

Student Firstname	student_lastname	student_email	department_name	student_class
Vedat	ÖZCAN	vozcan@gmail.com	Computer Engineering	3
Seçil	ÖZTÜRK	sturk@gmail.com	Computer Engineering	3
Şefika	ÖZLEM PUL	spul@gmail.com	EEE	3
Zeynep	HİLAL DESTEBAŞI	zhdeste@gmail.com	Faculty of Medicine	3
Alper	TEMEL	atemel@gmail.com	Architecture	3

Figure 15: Students list

10.3.6 Lecturers Page

It is the page where the lecturers are listed.



Lecturer List

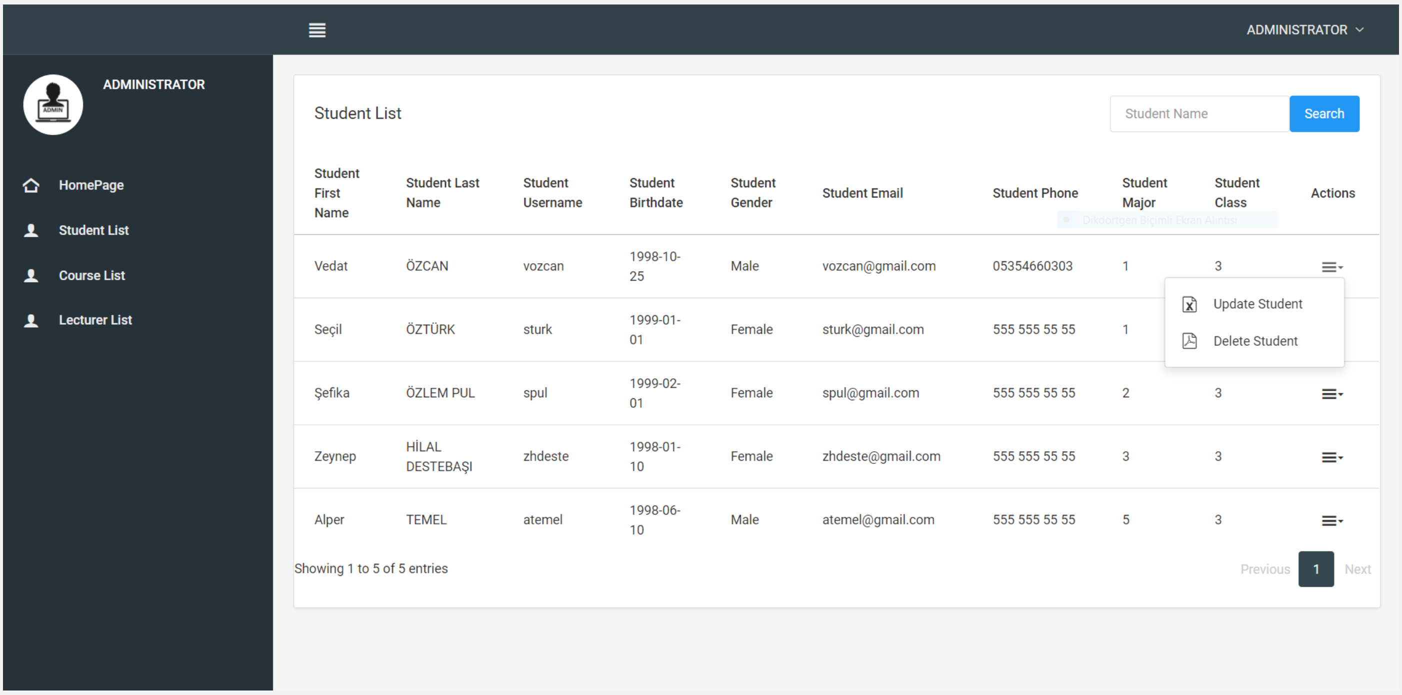
Lecturer List

Lecturer Fullname	Lecturer E-Mail	Lecturer Department	Department Address
Doç. Dr. Adil ALPKOÇAK	alpkocak@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Doç. Dr. Semih UTKU	semih@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Feriştah ÖRÜCÜ DALKILIÇ	feristah@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Fatih DİLMAÇ	fatih.dilmac@ceng.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Ali CÜVİTOĞLU	ali.cuvitoglu@ceng.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir

Figure 16: Lecturers list

10.4 Admin Panel

As mentioned before, it is the part where some basic functions of the system are performed. The action that can be taken by Admin are listed on the panel.

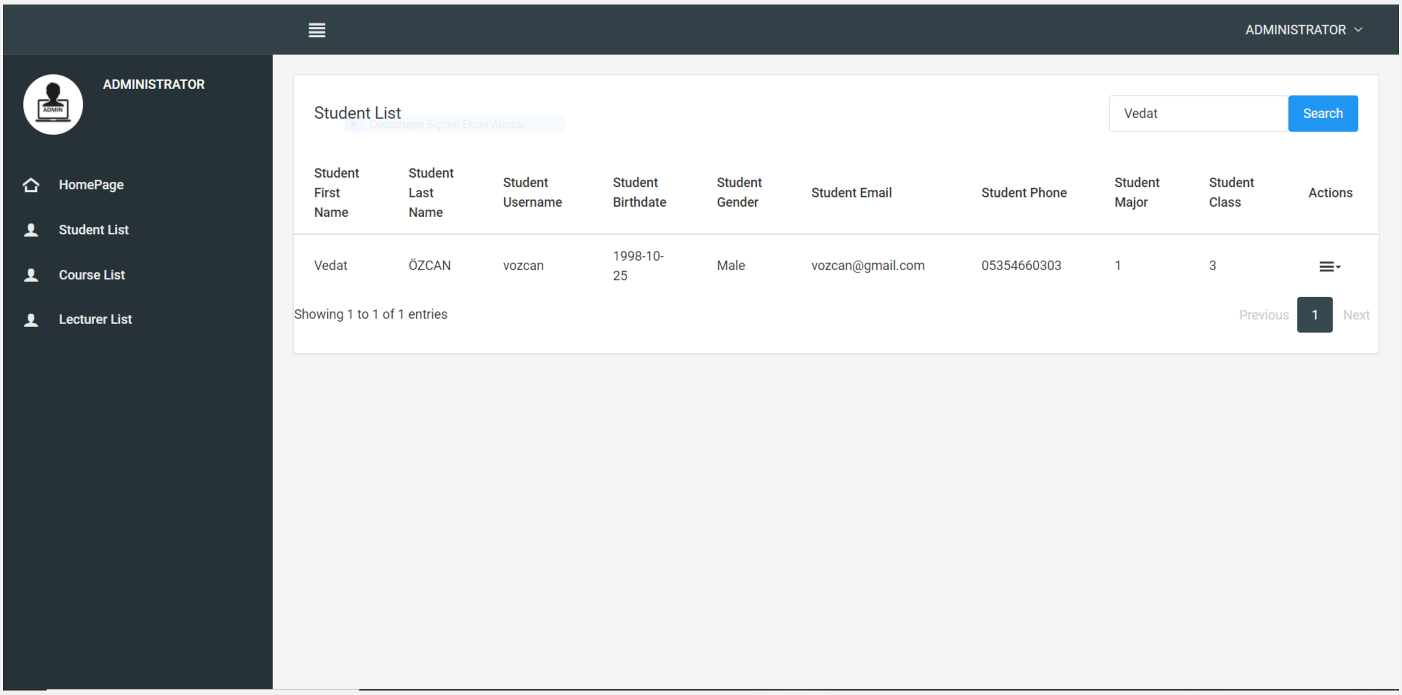


Student List

Student First Name	Student Last Name	Student Username	Student Birthdate	Student Gender	Student Email	Student Phone	Student Major	Student Class	Actions
Vedat	ÖZCAN	vozcan	1998-10-25	Male	vozcan@gmail.com	05354660303	1	3	≡
Seçil	ÖZTÜRK	sturk	1999-01-01	Female	sturk@gmail.com	555 555 55 55	1		
Şefika	ÖZLEM PUL	spul	1999-02-01	Female	spul@gmail.com	555 555 55 55	2	3	≡
Zeynep	HİLAL DESTEBAŞI	zhdeste	1998-01-10	Female	zhdeste@gmail.com	555 555 55 55	3	3	≡
Alper	TEMEL	atemel	1998-06-10	Male	atemel@gmail.com	555 555 55 55	5	3	≡

Showing 1 to 5 of 5 entries

Figure 17: Admin Panel - Students list

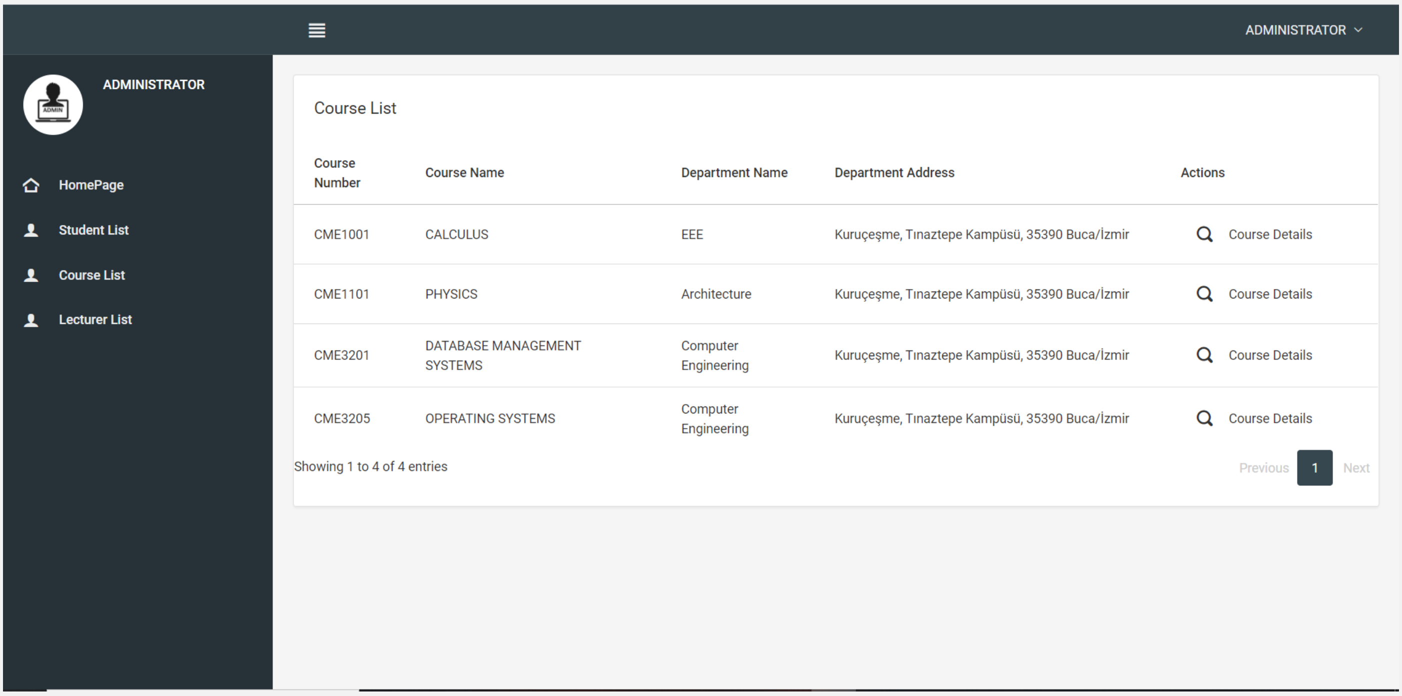


Student List

Student First Name	Student Last Name	Student Username	Student Birthdate	Student Gender	Student Email	Student Phone	Student Major	Student Class	Actions
Vedat	ÖZCAN	vozcan	1998-10-25	Male	vozcan@gmail.com	05354660303	1	3	≡

Showing 1 to 1 of 1 entries

Figure 18: Admin Panel - Search student by name

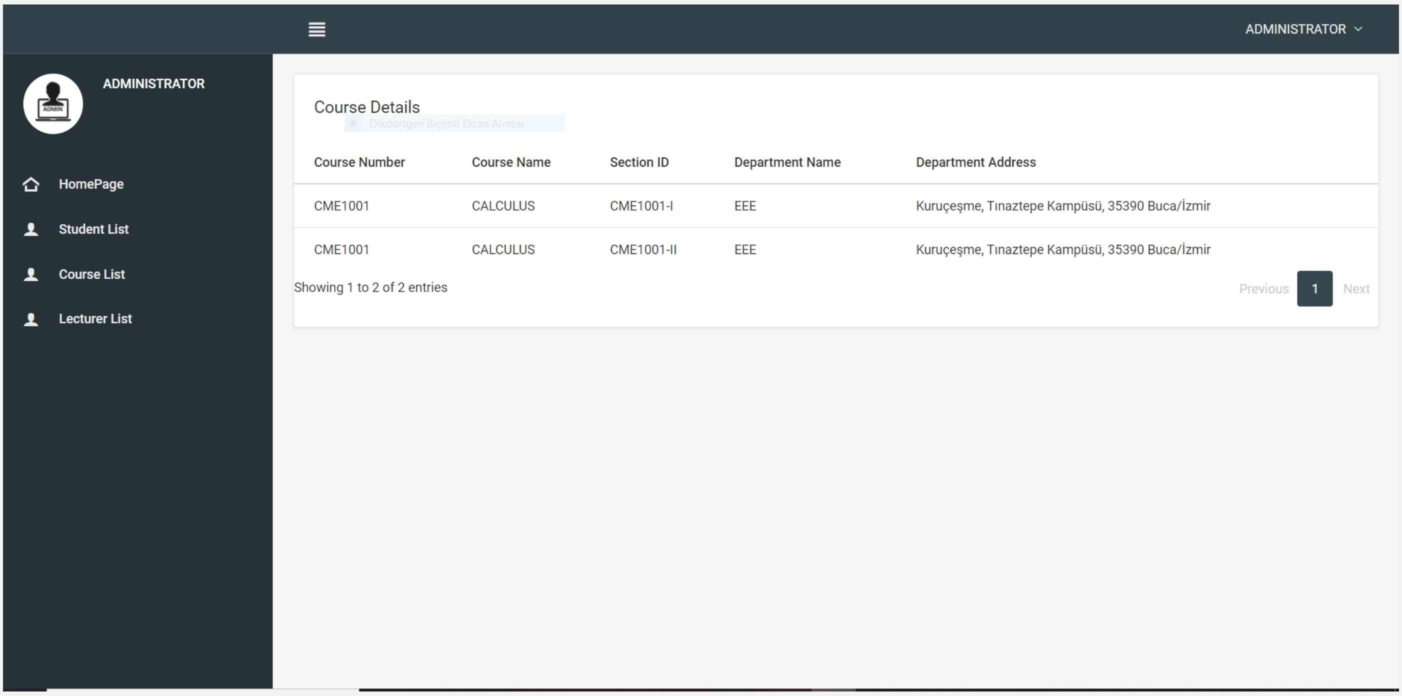


Course List

Course Number	Course Name	Department Name	Department Address	Actions
CME1001	CALCULUS	EEE	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir	Course Details
CME1101	PHYSICS	Architecture	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir	Course Details
CME3201	DATABASE MANAGEMENT SYSTEMS	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir	Course Details
CME3205	OPERATING SYSTEMS	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir	Course Details

Showing 1 to 4 of 4 entries

Figure 19: Admin Panel - Courses list

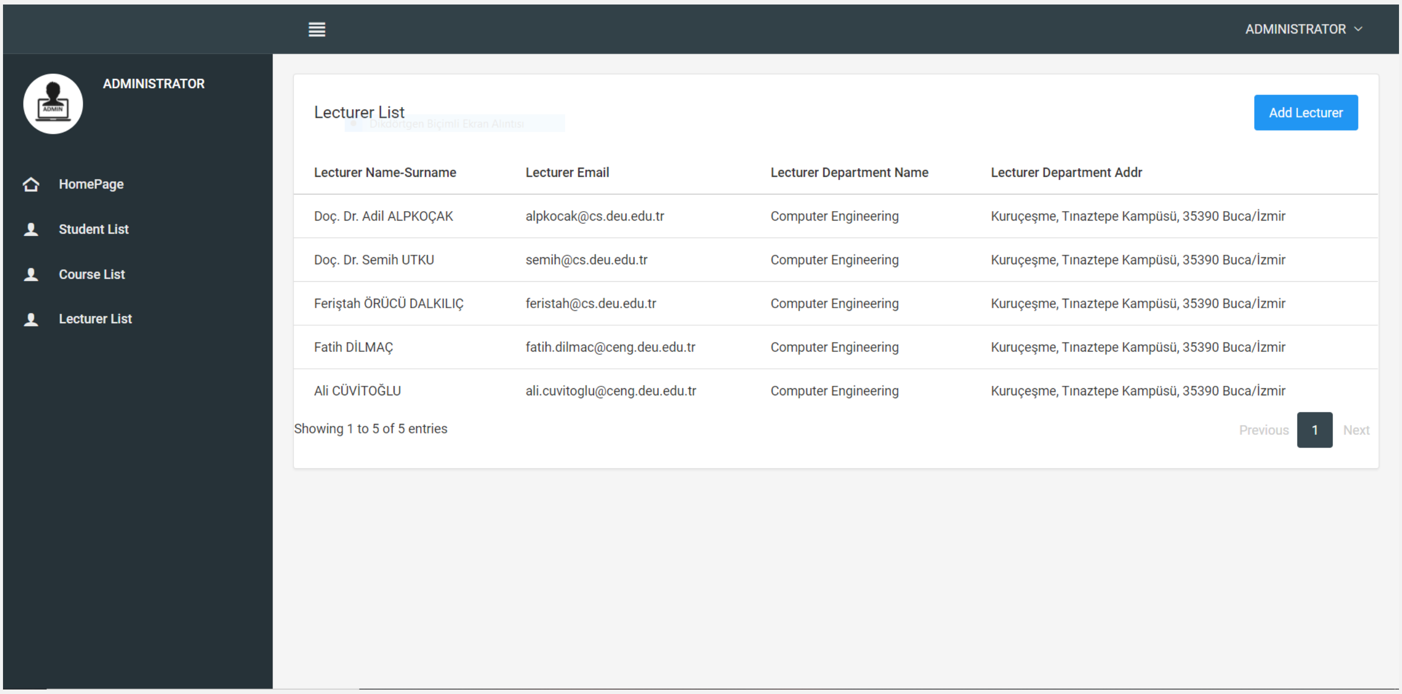


Course Details

Course Number	Course Name	Section ID	Department Name	Department Address
CME1001	CALCULUS	CME1001-I	EEE	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
CME1001	CALCULUS	CME1001-II	EEE	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir

Showing 1 to 2 of 2 entries

Figure 20: Admin Panel - Course detail (Sections list)

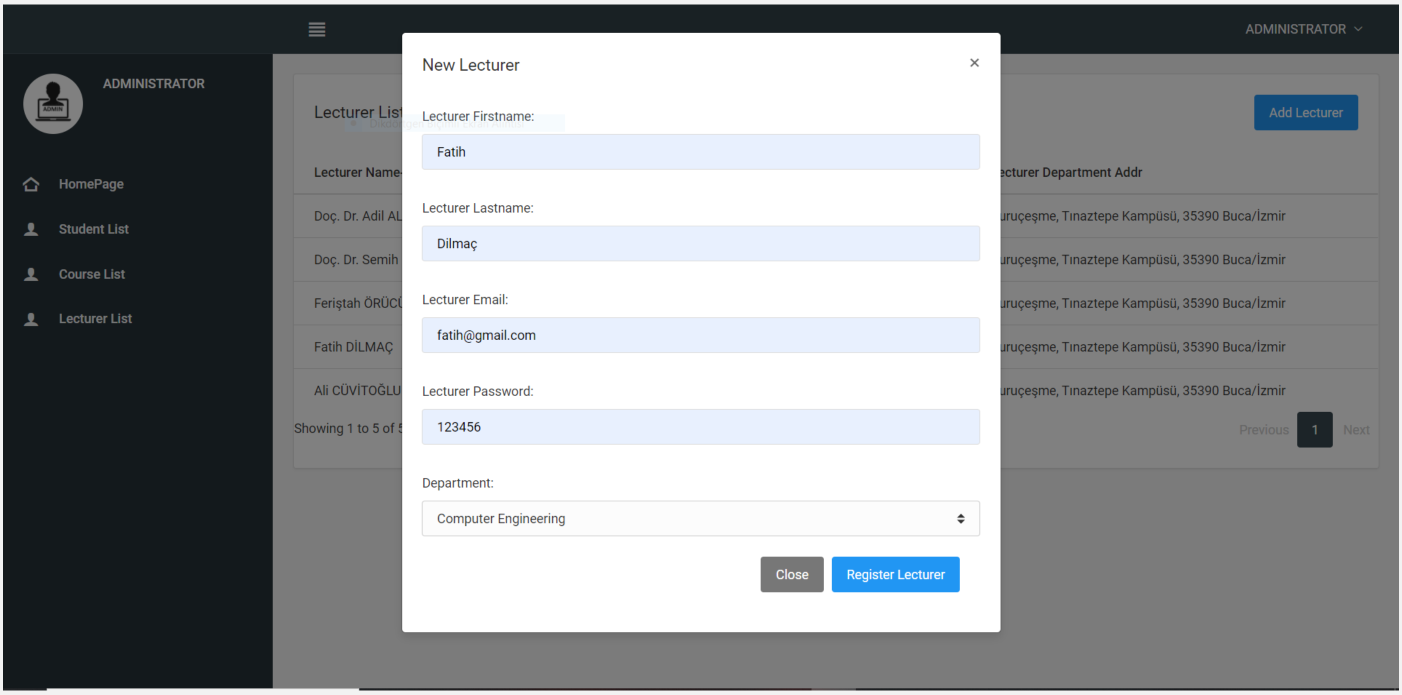


Lecturer List

Lecturer Name-Surname	Lecturer Email	Lecturer Department Name	Lecturer Department Addr
Doç. Dr. Adil ALPKOÇAK	alpkocak@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Doç. Dr. Semih UTKU	semih@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Feriştah ÖRÜCÜ DALKILIÇ	feristah@cs.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Fatih DİLMAÇ	fatih.dilmac@ceng.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir
Ali CÜVİTOĞLU	ali.cuvitoglu@ceng.deu.edu.tr	Computer Engineering	Kuruçeşme, Tınaztepe Kampüsü, 35390 Buca/İzmir

Showing 1 to 5 of 5 entries

Figure 21: Admin Panel - Lecturers list



New Lecturer

Lecturer Firstname: Fatih

Lecturer Lastname: Dilmaç

Lecturer Email: fatih@gmail.com

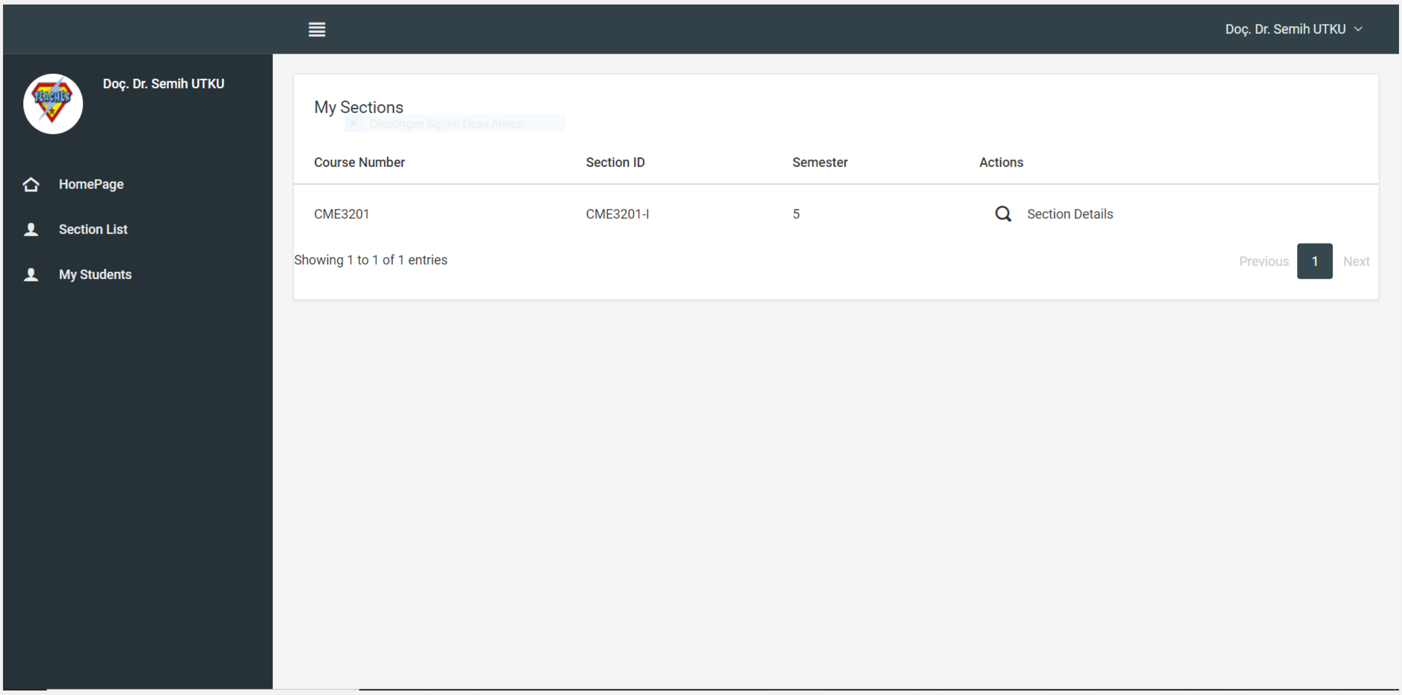
Lecturer Password: 123456

Department: Computer Engineering

Figure 22: Admin Panel - Add a new lecturer

10.6 Lecturer Panel

As in the Admin panel, the action that can be taken by the Lecturer entering the system are listed on the panel.

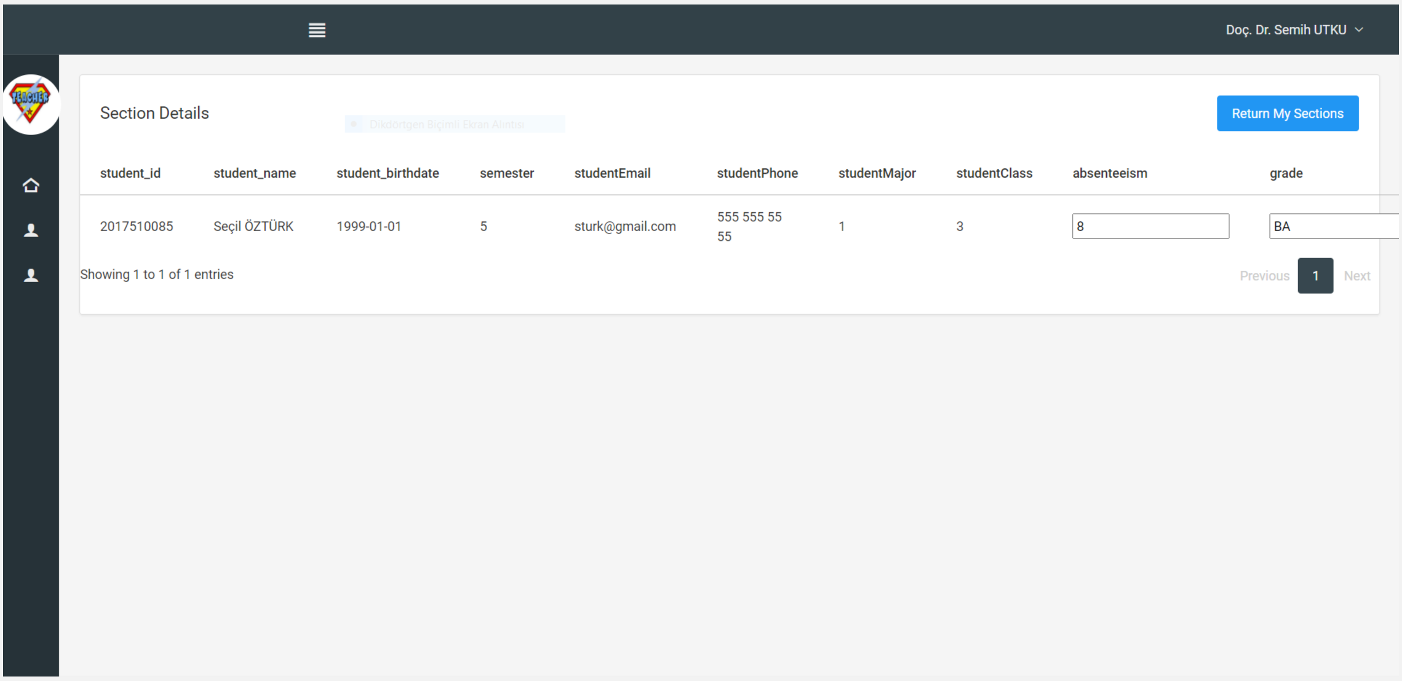


My Sections

Course Number	Section ID	Semester	Actions
CME3201	CME3201-I	5	Section Details

Showing 1 to 1 of 1 entries

Figure 23: Lecturer Panel – Lecturer own sections list

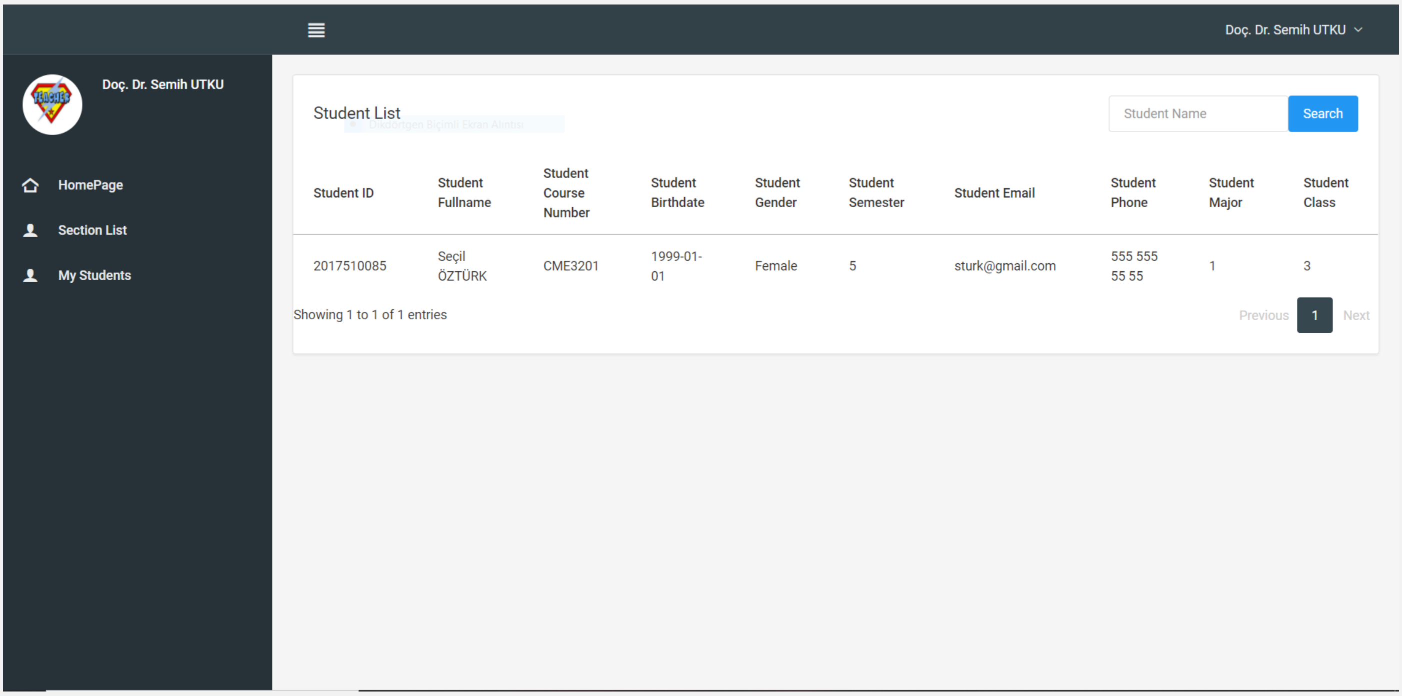


Section Details

student_id	student_name	student_birthdate	semester	studentEmail	studentPhone	studentMajor	studentClass	absenteeism	grade
2017510085	Seçil ÖZTÜRK	1999-01-01	5	sturk@gmail.com	555 555 55 55	1	3	8	BA

Showing 1 to 1 of 1 entries

Figure 24: Lecturer Panel – Lecturer own section details / Update student's information



Student List

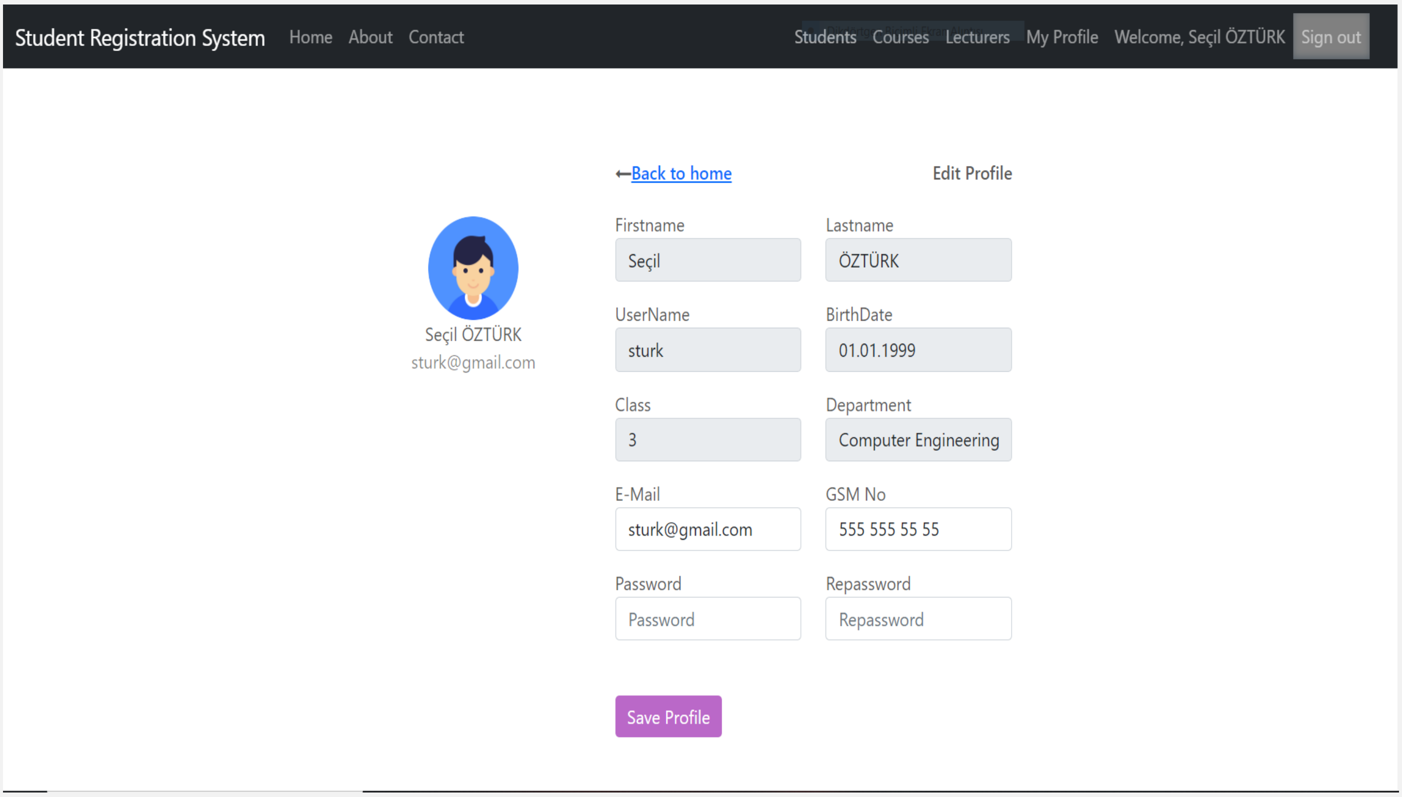
Student ID	Student Fullname	Student Course Number	Student Birthdate	Student Gender	Student Semester	Student Email	Student Phone	Student Major	Student Class
2017510085	Seçil ÖZTÜRK	CME3201	1999-01-01	Female	5	sturk@gmail.com	555 555 55 55	1	3

Showing 1 to 1 of 1 entries

Figure 25: Lecturer Panel – Lecturer own students list

10.5 Student Profile

The page containing the personal information of the student. The student can update their information on this page.



Edit Profile

Firstname: Seçil
Lastname: ÖZTÜRK
UserName: sturk
BirthDate: 01.01.1999
Class: 3
Department: Computer Engineering
E-Mail: sturk@gmail.com
GSM No: 555 555 55 55
Password: Password
Repassword: Repassword

Figure 26: Student profile page

11. Conclusion

When we started the project, we knew that unlike other periods, we would have to work remotely with our friends. This had both negative sides and positive sides. For example, we learned how to run the project remotely in a synchronized fashion. Besides, we learned about new technologies and the usage areas of these technologies. For example, we learned about the CodeIgniter framework and how easy the MVC architecture used by this framework can provide us when writing code.

12. References

[1] https://themeforest.net/item/limitless-responsive-web-application-kit/13080328
(For Admin Panel template)

[2] https://bootsnipp.com/snippets/Ga5nR
(For About Page structure)

[3] https://getbootstrap.com/docs/5.0/examples/
(For Home Page template)

[4] https://www.w3schools.com/
(For HTML, CSS, JavaScript and Bootstrap)

[5] https://online.deu.edu.tr/
(CME3201 courses for MySQL database structure and Php)

[6] https://www.udemy.com/
(Various courses to understand better CodeIgniter and MVC structure)