
**ÇANKAYA UNIVERSITY**
**FACULTY OF ENGINEERING**
**COMPUTER ENGINEERING DEPARTMENT**

# Project Report
**Version 1**

## CENG 407
Innovative System Design and Development I

## *201907*
## *PRIVATE TUTORING APPLICATION*

*Hande AYDIN*
*201511410*
*Eyüp Can TURĞUT*
*201411062*
*Damla YILDIRIM*
*201411068*
*Serhat CENKAYOĞLU*
*201511301*

Advisor: *Roya Choupani*
Co - Advisor: *Faris Serdar Taşel*

# Table of Contents

## Abstract

Teachers who want to give private lessons and students who want to take private lessons have difficulty finding each other. We noticed this problem and decided to make a web project to provide a solution. Students will be able to reach the teachers with the criteria they want after they register to the site and choose courses according to their curriculum. In this project, we found certain search criteria from similar sites we looked at. We have prepared a database plan for our web project. We decided to build the foundation for the project through MVC. In the model section MsSQL; we decided to use JavaScript in view, CSS and C # in control.

**Key words:**

Tutoring, Web application, Curriculum, Search by criteria, Teacher, Student.

## Özet:

Özel ders vermek isteyen öğretmenler ve özel ders almak isteyen öğrenciler birbirlerini bulmak konusunda sıkıntı yaşamaktadır. Biz bu sorunu fark ettik ve çözüm sağlamak için bir web projesi yapmaya karar verdik. Öğrenciler siteye kayıt olduktan sonra istedikleri kriterlerdeki öğretmenlere ulaşıp, öğretmenlerin ders programlarına göre ders seçebileceklerdir. Bu projede baktığımız benzer sitelerden belli arama kriterleri bulduk. Web projemiz için database planı çıkarttık. Proje için esas temeli MVC üzerinden oluşturmaya karar verdik. Model kısmında MsSQL; view kısmında JavaScript, CSS ve kontrol kısmında C# kullanmaya karar verdik.

**Anahtar Kelimeler:**

Özel ders, Web uygulaması, Ders programı, Kriterlere göre arama, Öğretmen, Öğrenci.

# 1. Introduction

## 1.1 Problem Statement

Students who want to take private lessons have problems finding teachers. They find it difficult to find teachers that meet their own criteria. Teachers, on the other hand, give private lessons, posters and so on. tools. Our goal is to create a useful platform to solve this problem. Our overall goal is to provide a convenience in the field of education and to design a website accessible to all audiences.

## 1.2 Background or Related Work

If we do some research on the internet, we may encounter a few more web-based tutoring applications such as Edunoor, Apprentus and Classgap. The common feature of these applications is that they find the teacher that suits you according to the criteria you set. First, the student enters the site and then specifies which course he / she wants to take, the level of the course, the purpose for which he / she took the course, the time period in which he / she can take the course, and information such as name and email. The system brings the appropriate teachers to the student according to the specified information. If there is a teacher who is wanted to get in contact, student signs up or logs in before doing it.

This process is different in our web application. First, the student enters the site and becomes a member. If the student is a member, he / she logs in directly. The system directs the student to the home page. From here, the student can access the syllabus, edit the syllabus and see the teachers with the highest votes. To take a course, the student clicks the teacher / course search button. Then the search page opens. Student fills the desired criteria such as course category, education level, course name, price gap, city, etc. to search for a course or teacher on demand. The system then brings the student the desired courses or teachers according to the criteria.

## 1.3 Solution Statement

The aim of our project is to bring people who want to take private lessons and give private lessons. Enrolled teachers can add courses to their curriculum. After the students register to the site, according to the location, education level, wage range, the teacher will search; teachers who have received votes at the rate they want, can send requests to the appropriate spaces in their curriculum by looking at their comments. Students or teachers may postpone the existing courses in the curriculum and add notes.

# 2. Literature Search

## 2.1 Abstract

Nowadays, people who want to give and receive private lessons cannot reach exactly what they want thanks to some of the sites currently in effect. Together with our project, we aim to develop and complete the shortcomings. The main purpose of this project is to facilitate accessibility to private lessons. In this project, we want to save time and ensure that people reach what they want through a quality and secure platform. With this project, people who want to teach and who want to take lessons can easily communicate with each other.

## 2.2 Literature Review

Several researches were carried out on the problem of taking and giving private lessons. In classgap et al.[1], training assignments for each person and each instructor is administered according to the review. The authors focused on developing a tutoring system with a web-based application. It is indicated that one should choose the appropriate teacher. In order to avoid any security problems during the selection, it is said that everything is paid attention and the trust of the person is gained. The adequacy of the number of trainers is mentioned. Transportation is also brought to the forefront, saving both time and money. All trainers have their own categories and capacity. The system can choose its own programmed time, it is also offered specific instructors. This software uses real data and the appropriate algorithm. According to some features, instructors are ranked. Using this algorithm, the system maps the appropriate instructor. This algorithm stops when mapped.

## 2.3 Private Tutoring

### 2.3.1 What is Private Tutoring

Private tutoring is usually kind of teaching given one to one and paid. It is one of the most effective methods used by everyone. It can also involve small groups of pupils.

If the student is in need of private lessons, his/her family will follow the student's situation and determines if there is a problem in the lessons according to the needs. Or the person can be an adult. According to his/her own situation, s/he sees the deficiencies and determines which information is missing.

One of the most important steps in the private lesson is identifying the need for information. In order to complete this need, a specialist instructor is needed. The trainer is selected for each branch.

In order for the tutoring to be efficient, both the instructor must be an expert or mastered in that subject. The selection of the instructor is determined according to the criteria such as the instructor's educational status, experience and ability to transfer information.

### 2.3.2 Who is Tutoring?

Most of private tuitions are provided by self-employed private tutors. Many of these tutors work independently, meaning that they enter into verbal or written contracts directly with parents, carers or adult students. Independent private tutors will often advertise their services. Other tutors are provided through private tuition agencies. Tuition agencies charge parents, carers and adult students a fee for finding them a tutor. This can be an introduction fee, commission for the duration of the tutoring contract, or both. A third provider of private tuition are private tuition centres. These centres are sometimes run on a franchise model. Well-known examples of private tuition centres in the UK are Kumon, Explore Learning and Magikats. Tuition in these centres is normally provided to small groups of learners. Schools in the state education system also use the services of private tutors. These tutors are usually provided by private tuition agencies or educational charities such as Action Tutoring.Summary.

### 2.3.3 Where to Tutor?

Private tutoring is usually given at the student’s home, the tutor’s home, a private tuition centre or on study centers. Lessons are either face to face (sometimes called 'in person' tuition) or online. Online tuition is an increasingly popular option for tutoring children and adult students.

## 2.4 Artificial Intelligence

To recommend suitable lessons for users, we may use artifical intelligence.

Artificial intelligence is the ability of a computer or a computer-controlled robot to perform various activities in the same way as intelligent creatures. The acronym of artificial intelligence is AI, often used in computer science. Artificial intelligence studies generally focus on analyzing human thinking methods and developing similar artificial instructions.

## 2.5 File Sharing

This In our application, when tutor needs to share course notes, we need data sharing.

File sharing is an application for online distribution or presentation to digital media such as computer programs, multimedia (audio, picture and video), documents or e-books. File sharing can be accomplished in a variety of ways. Storage, data transmission and distribution is achieved through manual sharing using removable media, central servers in computer networks, World Wide Web-based Hyperlink documents and distributed peer-to-peer networking.

## 2.6 Chat

For one to one lessons, tutors need to communicate with each other to deal and find a middle way. Chatting is fast way to meet that need.

Chat or online chat is a system of communication with the internet path seeker. Chat types can be found on the forums, on websites such as Facebook and other social networks. There are lots of chat types. Most commonly types are Instant Messaging, ICQ and IRC.

### 2.6.1 Instant Messaging

Instant messaging is a real-time conversation with people that you have added to your list as a member through a computer program. Depending on the program feature, video and voice calls may be possible. The most popular instant messaging programs are Pidgin, Kopete, ICQ, Yahoo Messenger, MSN Messenger and Google Talk. In fact, the history of instant messaging goes back to the Internet. Instant messaging was first seen in operating systems such as CTSS and Multics. Nowadays, some instant messaging software has started to offer video conferencing and Voice Over IP.

### 2.6.2 ICQ

ICQ is a very popular communication program. You can chat with people you have received in your ICQ list, make one-to-one or multiple chats with the chat option and transfer files. You can even send sms. You can also look at your emails and make friends around the world by making calls. ICQ mean is pronunciation of "I seek you." sentence.

### 2.6.3 IRC

IRC (Internet Relay Chat) is one of the most commonly used protocols in the internet world. The protocol is first introduced in 1988 by Finnish student Jarkko Oikarinen, has become one of the most popular chat networks, especially since the early years of the Internet. Even now, IRC service is provided through many servers in our country and in the world, allowing people of the same opinion to chat and share information with each other.

## 2.7 Summary

Our project is basically a private tutoring program (based on web application) that makes life easier for persons that wants to tutor. Our application includes many topics such as artificial intelligence, online chatting, file sharing, videoconferencing, etc. Except these, we will consider real tutors' comments to understand what kind of features they would like, what are the difficulties they had experienced for now and how can we solve that.

# 3. Software Requirements Specification

## 3.1 Introduction

### 3.1.1 Purpose

This document is a software requirement specification for our private tutoring application project which is based on web. Main purpose of this document is to give detailed information about the functionalities, constraints and software requirements of the project. The purpose of this document is describing our private tutoring application. The main purpose of the project is to develop a high-quality application with education theme, which available on all platforms that supports web. There are two types of users, student and teacher. The first type of user (student) can register and take classes that have been on database. The second type of user (teacher) can create classes. Also, every user can register as both kind of user types.

### 3.1.2 Vision Statement

Our aim in this project is to create a platform that will bring students and tutors together. Students who want to take private lessons and teachers who want to give private lessons can register to the system.

Students can choose courses from the appropriate parts of the instructor's course schedules and send course requests.

Students will be able to pre-select price and location information from the search criteria section. As the system matches students 'and teachers' basic criteria, these issues will easily be overcome. Feature List: The main feature of the project is that it brings together teachers who want to give private lessons and students who want to take private lessons. Algorithms:

1. Adding courses.
2. Search for private courses according to criteria.
3. Creating a course schedule.

- Registered teachers can create new courses.
- Students can choose the appropriate courses according to their schedule.
- Teachers may postpone their private lessons.
- Upcoming tutoring and canceled tutoring will be notified to the user screen.
- Teachers and students can see each other's profile information and communicate.

### 3.1.3 Glossary

| Term | Definition |
|---|---|
| User | Person that registers and uses the web application. |
| Registeration | The process to start a membership in web application. |
| Login | Entering the system via user name and password. |
| Schedule | Table of courses that user have taken. |
| Chat | A system of communication with the internet path seeker. |
| Contact | Communicating with student or teacher in order to give or receive specific information about course. |
| Profile Page | User's own personel page that has specific information about who s/he is. |
| Notification | The action of notifying when there is a progress about accepted or refused request, feedback, message etc. |
| Comment | Commentary of students that took courses about related teacher. |
| CV | It is a document in which it describes person's work experiences, responsibilities, achievements, competencies and skills. |

*Table 1: Glossary of SRS*

### 3.1.4 Overview of Document

This subsection should

1. Describe what the rest of the SRS contains;
2. Explain how the SRS is organized.

## 3.2 Overall Description

### 3.2.1 Product Perspective

Private Tutoring Application is a platform that aims to unite people who want to take and give private lessons. Users who want to take courses can access the instructors by selecting the features they want, and users who want to teach can easily communicate to provide services.

### 3.2.2 Development Methodology

We will benefit from the agile software development process in our project. To give some information about the agile software process; Agile methodology is a special approach to project management used in software development. This method helps teams respond to the foresight of the software development process. With incremental, iterative business sequences known as Sprint. Sprint is the time allocated for a phase at the end of a project. When the sprints expire, the project is deemed completed. There may be disagreement among Team Members about whether developing is satisfactory; however, it will not be done in a project before the project is completed. The remaining phases of the project continue to be developed inside in their own time. Benefits of using the agile method: The agile method derives from the experience of leading software experts in real life projects. This has removed the challenges of traditional development and links at the borders. Then the agile method was considered a better solution than the project developer in the industry. Immediately its immediate software developer used a form of agile method. This method provides a light framework for auxiliary teams. Helps them work and focus on fast delivery. This focus provides assistance to skilled organizations in reducing overall risks associated with software development.

## 3.3 Requirement Specification

### 3.3.1 External Interface Requirements

#### *3.3.1.1 User Interfaces*

This app has been developed for anyone who wants to use it. Users can access the site via the web. The home page consists of login and register section and site information. After login, profile pages are accessed. The profile page has been created specifically for users taking courses or giving lectures. This page consists of searching, updating profile, notification and calendar.

#### *3.3.1.2 Hardware Interfaces*

There are no external hardware interface requirements.

#### *3.3.1.3 Software Interfaces*

Our application is a web-based software, so the application requires an internet browser.

#### *3.3.1.4 Communication Interfaces*

There are no external communication interface requirements.

### 3.3.2 Functional Requirements

#### *3.3.2.1 Product Functions*

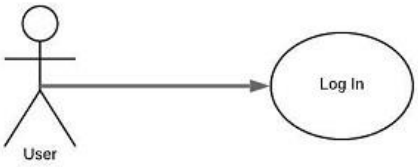



**Figure 1 - Log In:** User fills necessary information which are user name and password and enters the system.

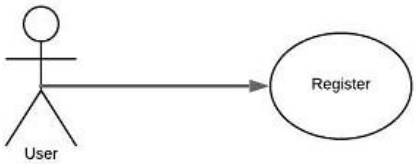



**Figure 2 - Register:** User becomes a member to log into the system.

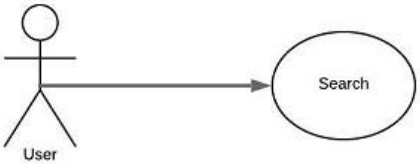



**Figure 3 - Search:** User searches the course or teacher name to find avaible courses easily.

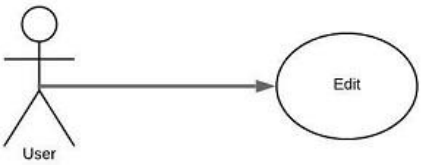


**Figure 4 - Edit:** User can manage the schedule. If there is a necessary situation, classes can be edited as cancelled, postponed, changed, etc.

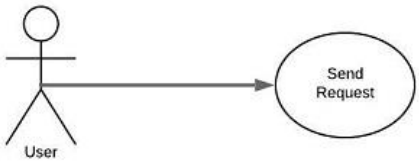



**Figure 5 - Send Request:** To take avaible course, student sends a request to the teacher. If teacher accepts, student takes the course. Else, student can't take that course.

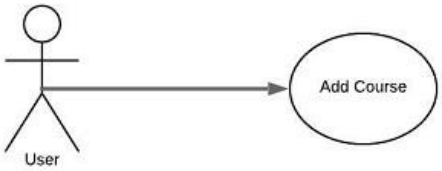



**Figure 6 - Add Course:** If student can't find the course that s/he want to learn, a new course could be added.

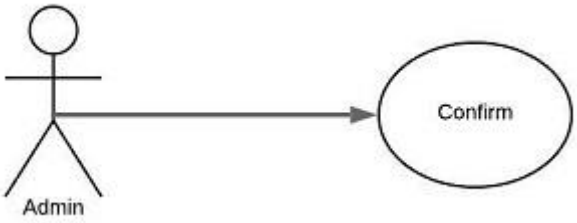



**Figure 7 - Confirm:** If there is a request about class, teacher confirms it to give a lecture.

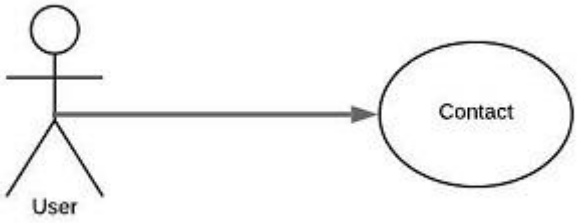



**Figure 8 - Contact:** To meet and deal, teacher and student contact via chatting.

### *3.3.2.2 Use Case Diagram*

#### 3.2.2.2.1 Use Cases

- Login
- Register
- Search
- Edit
- Add lesson
- Contact
- Send Request
- Confirm

#### 3.2.2.2.2 Diagram

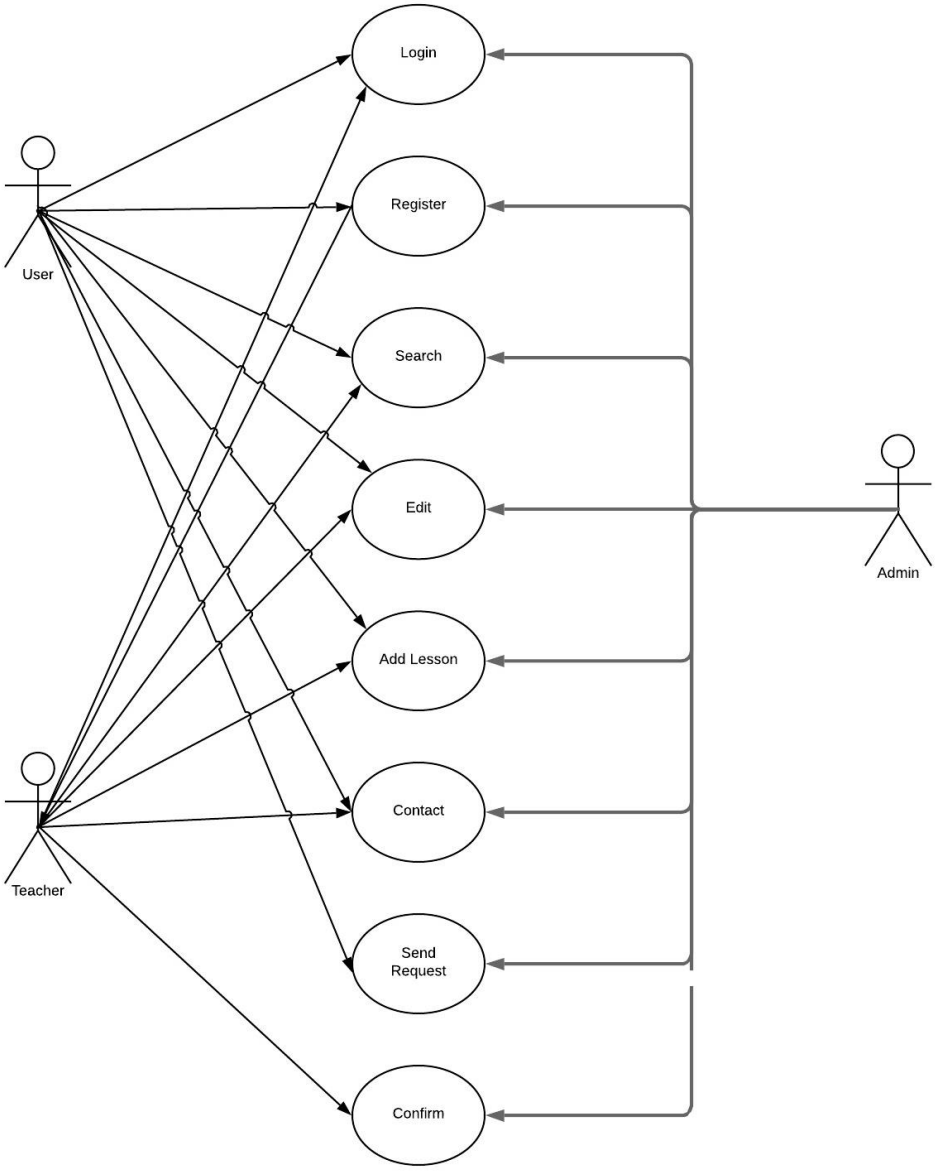



*Figure 9 – Use Case Diagram*

#### 3.2.2.2.3 Brief Description

The use case diagram diagram is shown above. The user can log in when he opens the website. If you do not have an account you can register using the register button. After login, you can search for a course or instructor in the search section. The teacher or student can edit his profile or syllabus using the edit button. The teacher sends the request that he wants to add this course to his / her admittance by entering the information of the course he wants to add with the add lesson button. If admin approves, that course opens. Persons who wish to take or

give the course can communicate with each other in the contact section. The student can send a request to the teacher about the course he / she wants to take. If the teacher approves the course, the person can take the course.

#### 3.2.2.2.4 Initial Step-By-Step Description

1. If the user has an account, he can login to the website by clicking the login button after entering their information. The website will search the database for the password and username matching and confirm whether the user has typed the password correctly. If the password is entered incorrectly, there will be no user password match in the database. Therefore, the user will not be able to login. A notification is sent to the user.
2. If the user does not have an account before, he can create an account by using register button. On the opened page the user has to fill in all the information.
3. The user logging in to the site makes a search by entering the required information in the search button section of the course or a specific teacher he / she wants to take.
4. Users can update their personal information or delete, add, etc. in the syllabus. uses the edit button to perform actions.
5. The database contains information indicating that the user is a teacher or a student. When the teacher ID is determined from the database, the homepage that comes after the login page has the add button for the teachers only on the profile page. By clicking the Add button, the user fills in the requested information on the opened page. As soon as the submit button is clicked, admine goes to the request that he / she wants to open the course. If admin approves the course, it opens.
6. Users communicate with each other to exchange information and communicate.
7. Users send a request for taking courses to the instructor at appropriate times according to the courses given by the instructor in the profile of the instructor. It also assigns requests by accessing the desired courses or teachers by using the search button.
8. Admin or instructor approves or rejects incoming requests.

### *3.2.2.3 Activity Diagram*

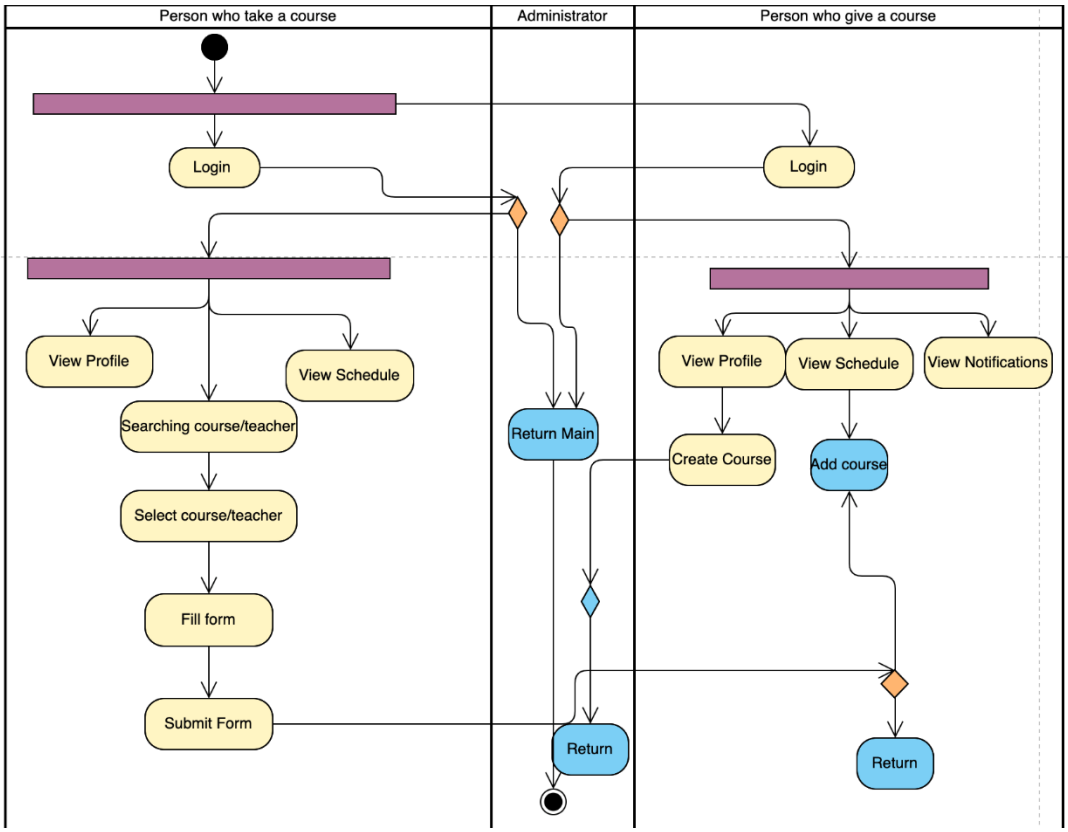



*Figure 10 – Activity Diagram*

# 4. Software Design Description

## 4.1 Introduction

### 4.1.1 Purpose

The aim of our project is to create a platform to meet teachers who provide private lessons and students who want to take private lessons. Our target audience is students who want to take private lessons and teachers who want to give private lessons. the target audience will be able to use this platform to provide private lesson interaction. In this platform we will be able to see which students want to take which courses and which teachers are teaching which courses. Students will also be able to see in which districts the teachers who are giving the courses they want can give lectures so that they can easily reach and meet. The main purpose of this project is to meet people who want to take and give private lessons safely and quickly. This software document includes class diagram, activity diagram, system architecture and motivation.

#### 4.1.2 Scope

It is a web project that includes people who want to take private lessons and who want to take private lessons. We realize that time is one of the most important things for users. Users hope to reach the page quickly, users who can not get what they want in terms of speed will not want to visit the site again. At the same time, if more than one user wanted to connect to the site at the same time, our site would lose serious speed. But with ADO.Net we overcome these problems. With ADO.Net, we are able to quickly make connections with the database and withdraw data. This project table, user, courses, calendar and so on. will use a database that contains all the necessary information. Details of database tables are described in this document.

We use MVC, an architectural software to determine how to allocate responsibilities on the code for our web project. In the classical understanding, all the information drawn from the database, the piece of code that holds this information and the pieces of code that make up the texts on the screen are together. In addition, when we want to change an information on the screen, the tasks performed by the save button are always in a combination of the codes that take the text from the screen and convert it to a code and save it in the database. We solve this with MVC. There are admin and user in our project. Users are divided into two groups as taking courses and giving courses. If the user is not yet registered in the system, he cannot fully access the site. After the user becomes a member of the system, the user enters the system with the member name and password of his choice. The administrator makes the necessary approvals from the back and presents the updated data to the users. The registered user can select data from the system according to the updated data and create data. When the user exits the system can exit.

### 4.1.3 Glossary

Example glossary for SDD.

**Table 2 Glossary of SDD**

| **Term** | **Definition** |
|---|---|
| User | Person that utilizes and uses the system such as web application, computer, operating system etc. |
| Web Browser | A browser is a generic name for software that allows users to acquire and view information on the web. |
| Admin | Person who owns the most authority on the systems like computer, network, etc. |
| GUI (Graphical User Interface) | A user interface that includes graphical elements such as icons, buttons, etc. to interact with the system. |
| MVC | It is an architectural pattern that is used in software engineering. |
| SDD (Software Design Description) | Description that is written for the software product. |

| | |
|---|---|
| UML Diagram | It is a collection of methods that determine and explain how software system is modeled. |
| Web Site | A place on the world wide web. |

#### 4.1.4 Overview of document

This chapter provides information about the general items of the documentation. In section 1, we explained the general purpose of the project and its target audience. we gave information about the general tools used in the project. In the second section, we explained the scope of the project with whom and how it interacted and the interaction of the tools with the user during this interaction process. In the 3rd section, the terms we use in the documentation have a dictionary meaning. In section 4, we talked about the general factors that led us to make this project. We also explained our main sources of motivation. In the 5th section, we talked about the general architectural structure of the project. These are divided into three parts as database, interface and cod part separately. 7th, 8th and 9th sections we talked about class, activity and use case diagrams. In the 10th section, we explained the interface design by supporting it with visuals.

#### 4.1.5 Motivation

We are a group of senior students in computer engineering department. Our main motivation in choosing this project is our search for solutions to some problems in our student life. We needed such a platform when we wanted to take private lessons from the courses we failed. We encountered brochures distributed by teachers who wanted to give private lessons in our school. We witnessed that teachers who want to give private lessons have problems finding students. We took such a step to solve this problem. When we talked to our group friends, we decided that this was a big problem and we had to develop a project that solves this problem. We learned how to develop this system in web development.

## 4.2 Platforms to Be Used

#### 4.2.1 MSSQL

MSSQL is a suite of database software published by Microsoft and used extensively within our enterprise.

#### 4.2.2. ASP.NET

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages developed by Microsoft to allow programmers to build dynamic web sites, applications and services.

#### 4.2.3. C#

C# is a general object-oriented programming (OOP) language for networking and Web development. C# is specified as a common language infrastructure (CLI) language.

#### 4.2.4. HTML - CSS - JS

HTML is the standard markup language for Web pages. Cascading Style Sheets, or CSS, allow you to specify things like the font you want on your page, the size of your text, whether the page is to have 2 columns, whether your text is to be in bold or italics, and so on. In other words, it is the part that lets you control the appearance of your web page. Javascript (JS) is a scripting languages, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code.

## 4.3 System Architecture

Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph. Use this style for the paragraph.

### 4.3.1 Architectural Design

#### *4.3.1.1 All User*

The user must log in to access the personal login page. The user can login to the system with their chosen username and password. After login, you will be redirected to the homepage. On the Homepage the user can view his own calendar. At the top of the main page, there is a search button, course schedule editing button, profile and notification buttons. It can also provide information from the top-rated and interested parts on the left side. When the user enters the search section, the page is divided into two sections. Filter options are available for searching on the left side. On the right side there are photographs, names, surnames and brief summaries of various teachers. Clicking on the short teacher information will lead you to a new page that gives detailed information of the teacher. This page includes the teacher's curriculum, the teacher's CV, the teacher's rating, the photo, and the comments about the

teacher. In addition, the user who wants to take courses in the same page can see which courses the teacher gives. By selecting the empty parts above the syllabus, he / she can send a request to the teacher that s / he wants to take the syllabus. The teacher can see these requests on the notifications page by clicking the notifications button on the main page. From the notifications page, you can see the information of the people who send requests and communicate with the same people. It can also approve or reject incoming requests.

Operations: Introduction (), View Timetable (), View Personal Information (), Set Schedule () These four operations are common to user types. Name: Introduction () Prerequisite: All information about users comes from the database. Post condition: The user can view the course schedule and personal information. Can set personal information and syllabus. Exception Path: If the information, user name and password or click on the log out button. Flow of events:

1. User opens Private Tutoring Application website.
2. User clicks the input button.
3. The system logs on.

Name: View Personal Information () Prerequisite: Introduction to the system. Final condition: The user can see personal information. Exception Path: None Events Flow: When the user logs in to the system, they can see personal information on my profile page. Name: View Course Calendar () Prerequisite: Login to the system. After the condition: The user can see the syllabus. Exception way: None

Events Flow: When the user enters the system, he can see the syllabus on the main page.

Name: Set Schedule () Prerequisite: Access the Edit page Post-Condition: The user can update the syllabus. Exception Path: None

Events Flow:

1. The user clicks the edit schedule button on the main page.
2. Makes necessary updates on the opened page.
3. Save by clicking the Save button.

#### *4.3.1.2 Teacher*

Name: Request to Open a Course ()

Pre-condition: Login to the system and enter all the necessary information about the course. Post-requisite: The teacher makes a request for a course. Exception Path: If the teacher has already been required for this lesson, or clicks to the cancel button. Flow of events:

1. Teacher clicks the profile button on the home page.
2. After accessing the profile page, click the create course button.
3. Complete the necessary information for the course and click the save button.
4. The system goes to the request to open the course.

#### *4.3.1.3 Student*

Name: Request to Take Courses ()

Pre-condition: Login to the system and make the necessary choices related to the course.

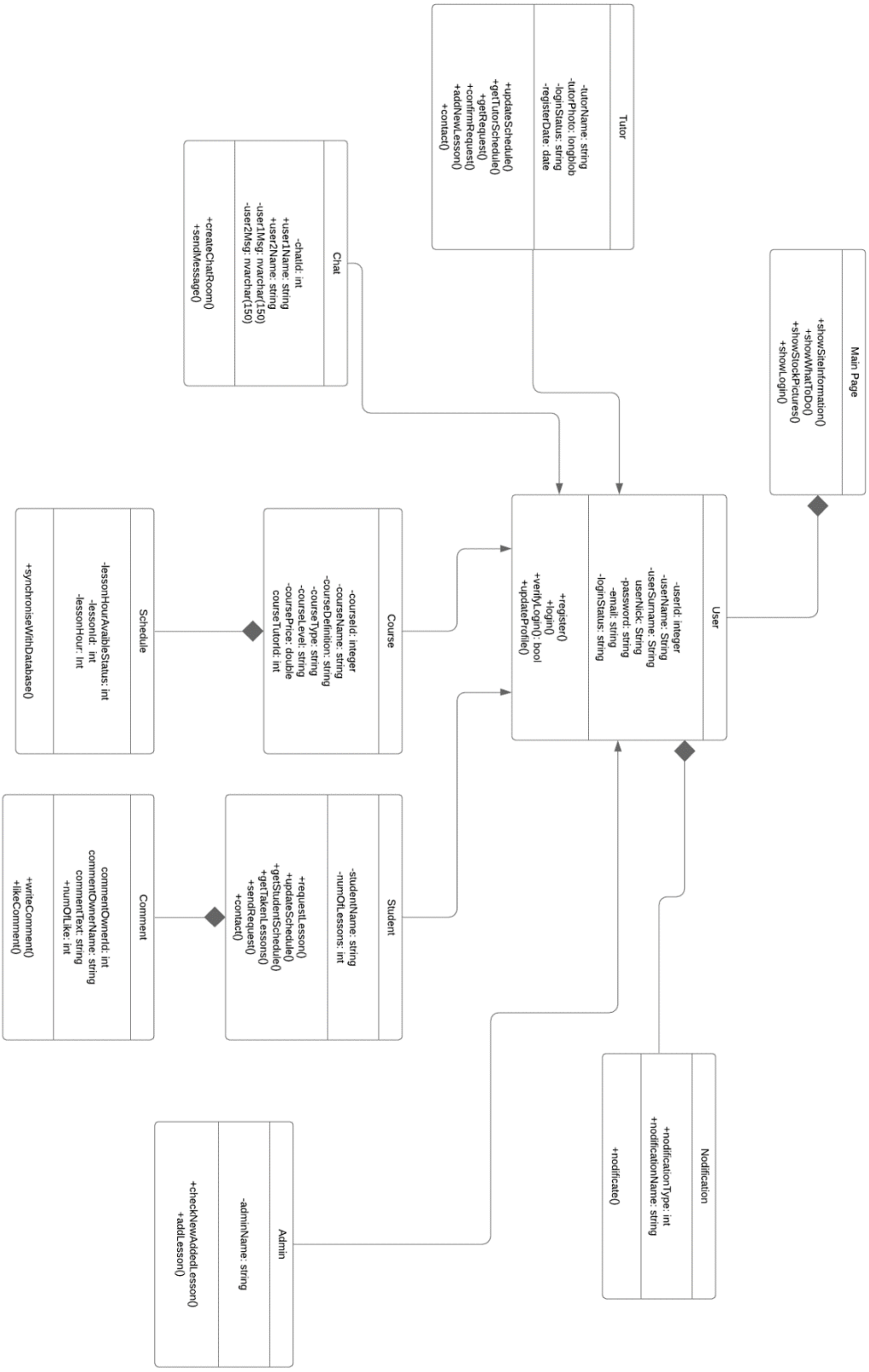
Post-condition: The student requests for a course.

Exception Path: If the student has already been required for this lesson, or clicks to the cancel button.

Events Flow:

1. The student searches for the desired course or teacher in the search button on the homepage.
2. In the page that opens, it filters the courses or clicks on the teacher profile.
3. According to the profile page of the teacher, according to the courses given in the course schedule of the teacher, he / she sends a request according to his wish.
4. According to the request, the course can take or not.

### 4.3.2 UML Class Diagram



## 4.4 Database Design

### 4.4.1 Tables

#### *4.4.1.1 User Table*

| Attribute Name | Attribute Type |
|---|---|
| userId **(PK)** | int |
| userType | int |
| userName | nvarchar(30) |
| userSurname | nvarchar(30) |
| userAge | tinyInt |
| userNick | nvarchar(30) |
| userPassword | string |
| userPhone | int |
| educationStatus | string |
| email | string |
| loginStatus | bool |
| ipAdress | string |

#### *4.4.1.2 Student Table*

| Attribute Name | Attribute Type |
|---|---|
| studentId **(PK)** | int |
| studentName | nvarchar(30) |
| studentSurname | nvarchar(30) |
| studentAge | tinyInt |
| studentNick | nvarchar(30) |
| studentPassword | string |
| studentPhone | int |
| educationStatus | string |
| email | string |
| registerDate | date |
| studentScheduleId **(FK)** | int |

#### *4.4.1.3 Tutor Table*

| Attribute Name | Attribute Type |
|---|---|
| tutorId **(PK)** | int |
| tutorName | nvarchar(30) |
| tutorSurname | nvarchar(30) |
| tutorAge | tinyInt |
| tutorNick | nvarchar(30) |
| tutorPassword | string |
| tutorPhone | int |
| graduaitonStatus | string |
| email | string |
| CV | string |
| registerDate | Date |
| tutorScheduleId **(FK)** | int |

#### *4.4.1.4 Admin Table*

| Attribute Name | Attribute Type |
|---|---|
| adminId **(PK)** | int |
| adminName | nvarchar(30) |
| adminSurname | nvarchar(30) |
| adminAge | tinyInt |
| adminNick | nvarchar(30) |
| adminPassword | string |
| userPhone | int |
| email | string |

#### *4.4.1.5 Lesson Table*

| Attribute Name | Attribute Type |
|---|---|
| lessonId **(PK)** | int |
| tutorId **(FK)** | int |
| lessonName | string |
| lessonCategory | string |
| lessonLevel | string |
| lessonDefinition | string |
| priceMin | double |
| priceMax | double |
| adress | string |

#### *4.4.1.6 Schedule Table*

| Attribute Name | Attribute Type |
|---|---|
| scheduleID **(PK)** | int |
| tutorID **(FK)** | int |
| studentID **(FK)** | int |
| lessonID **(FK)** | int |
| hour | int |
| day | tinyInt |
| week | tinyInt |
| year | int |
| note | nvarchar(255) |
| status | tinyInt |

#### *4.4.1.7 Comment Table*

| Attribute Name | Attribute Type |
|---|---|
| commentID **(PK)** | int |
| tutorID **(FK)** | int |
| studentID **(FK)** | int |
| rate | tinyInt |
| comment | string |
| isHide | bool |

#### *4.4.1.8 Nodification Table*

| Attribute Name | Attribute Type |
|---|---|
| nodifID **(PK)** | int |
| tutorID **(FK)** | int |
| studentID **(FK)** | int |
| lessonID **(FK)** | int |
| type | tinyInt |

### 4.4.2 Database Diagram

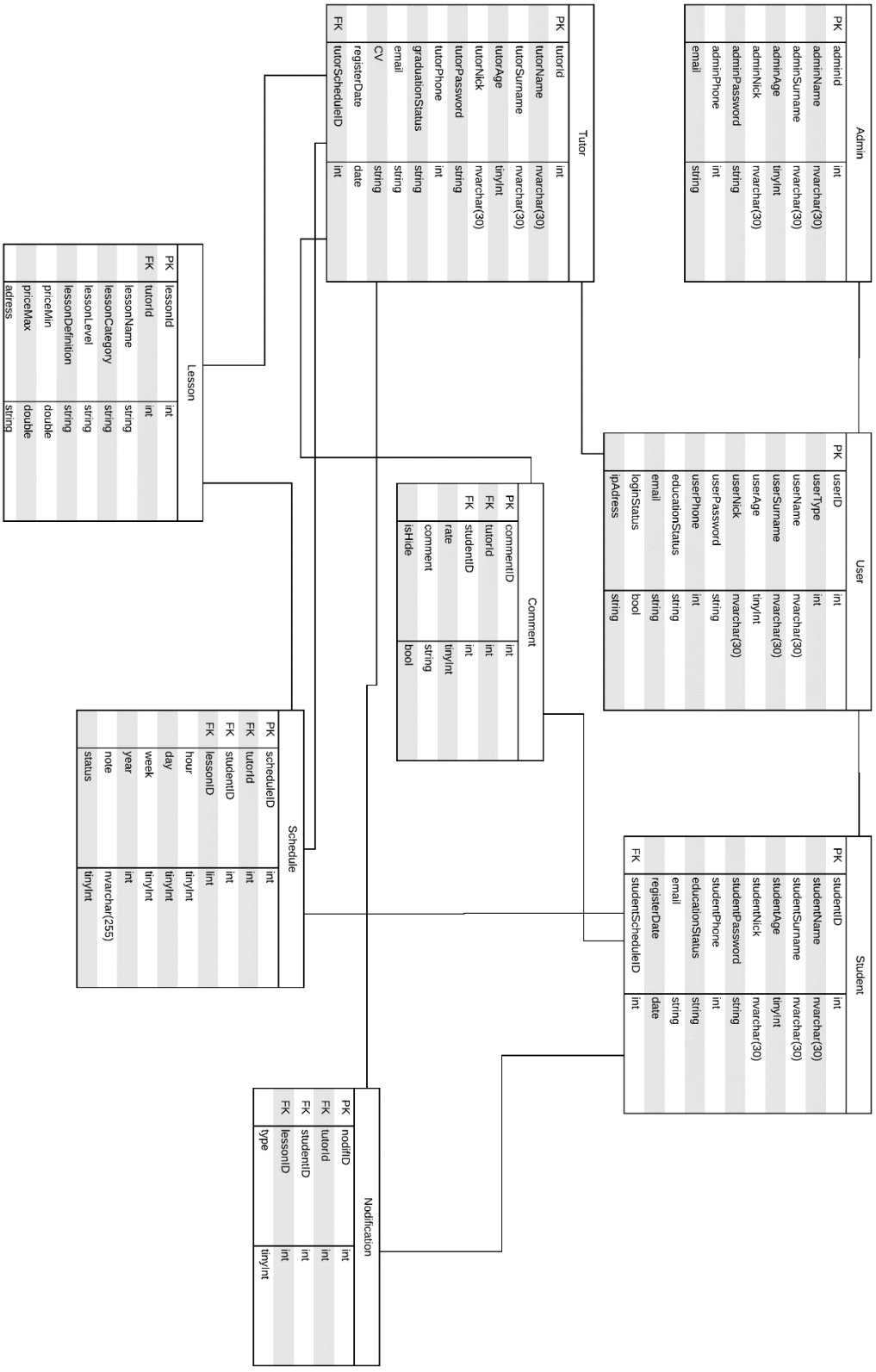



**Figure 11 Database Diagram**

## 4.5 Interface design

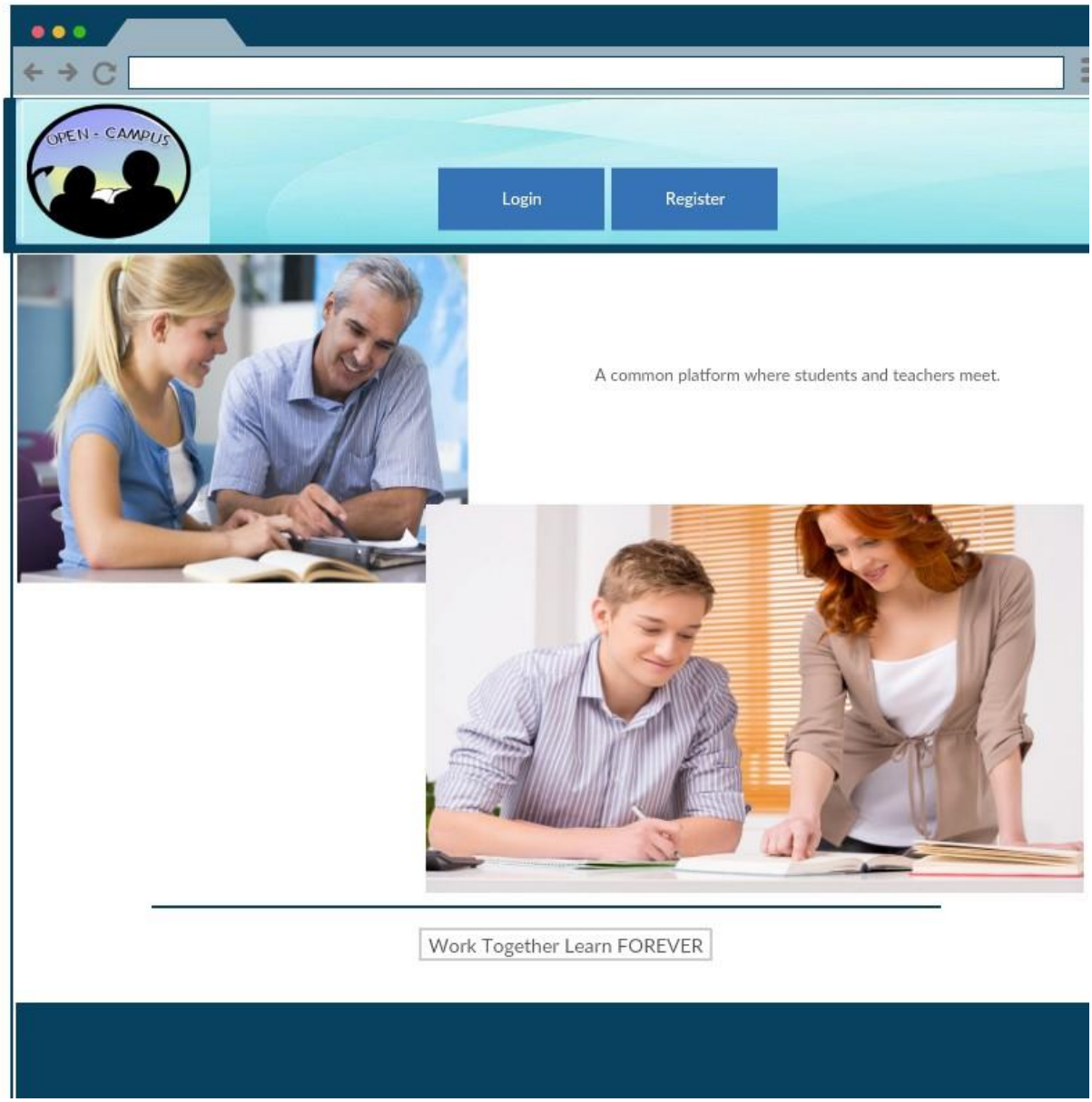



**Figure 11 Login/Register Page**

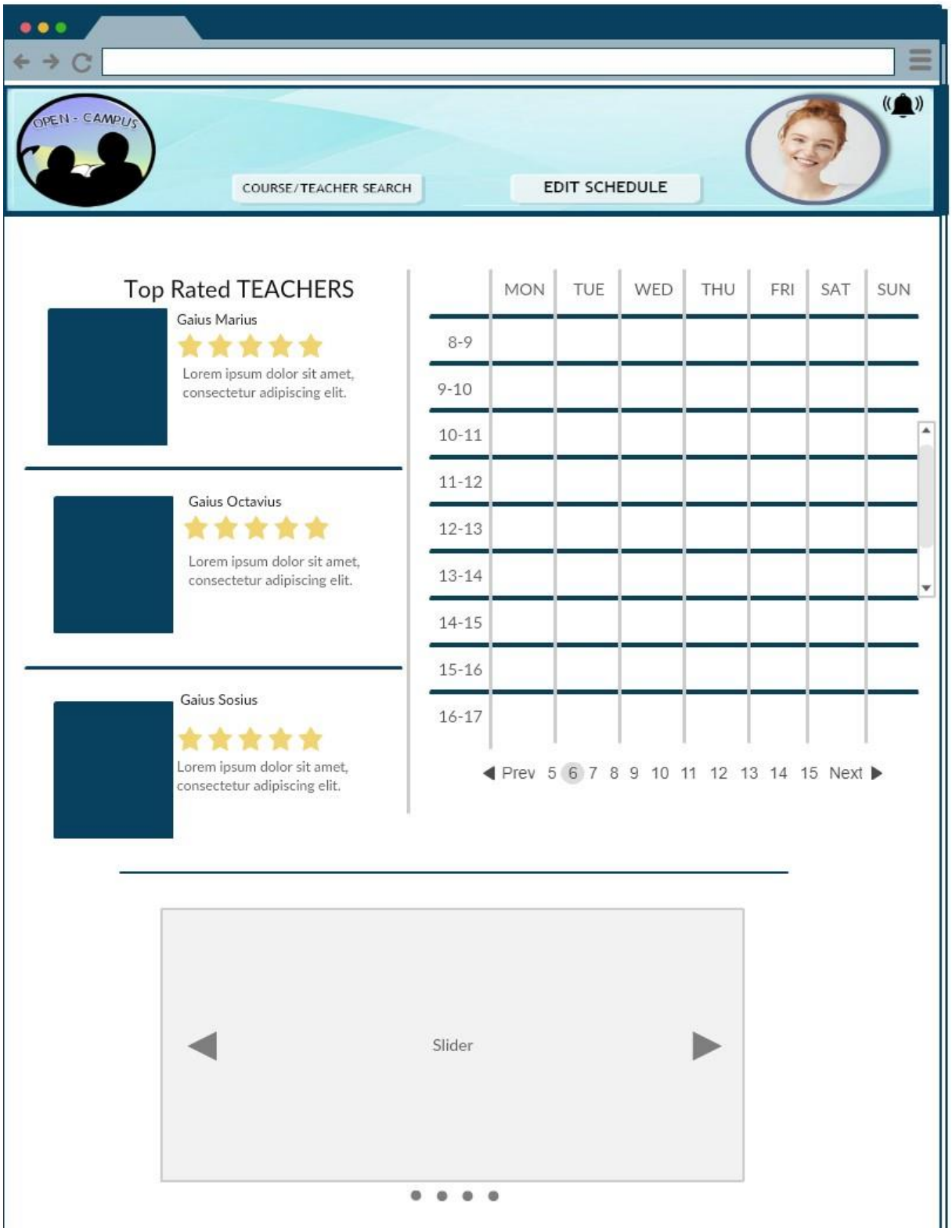



**Figure 13 Main Page After Login**

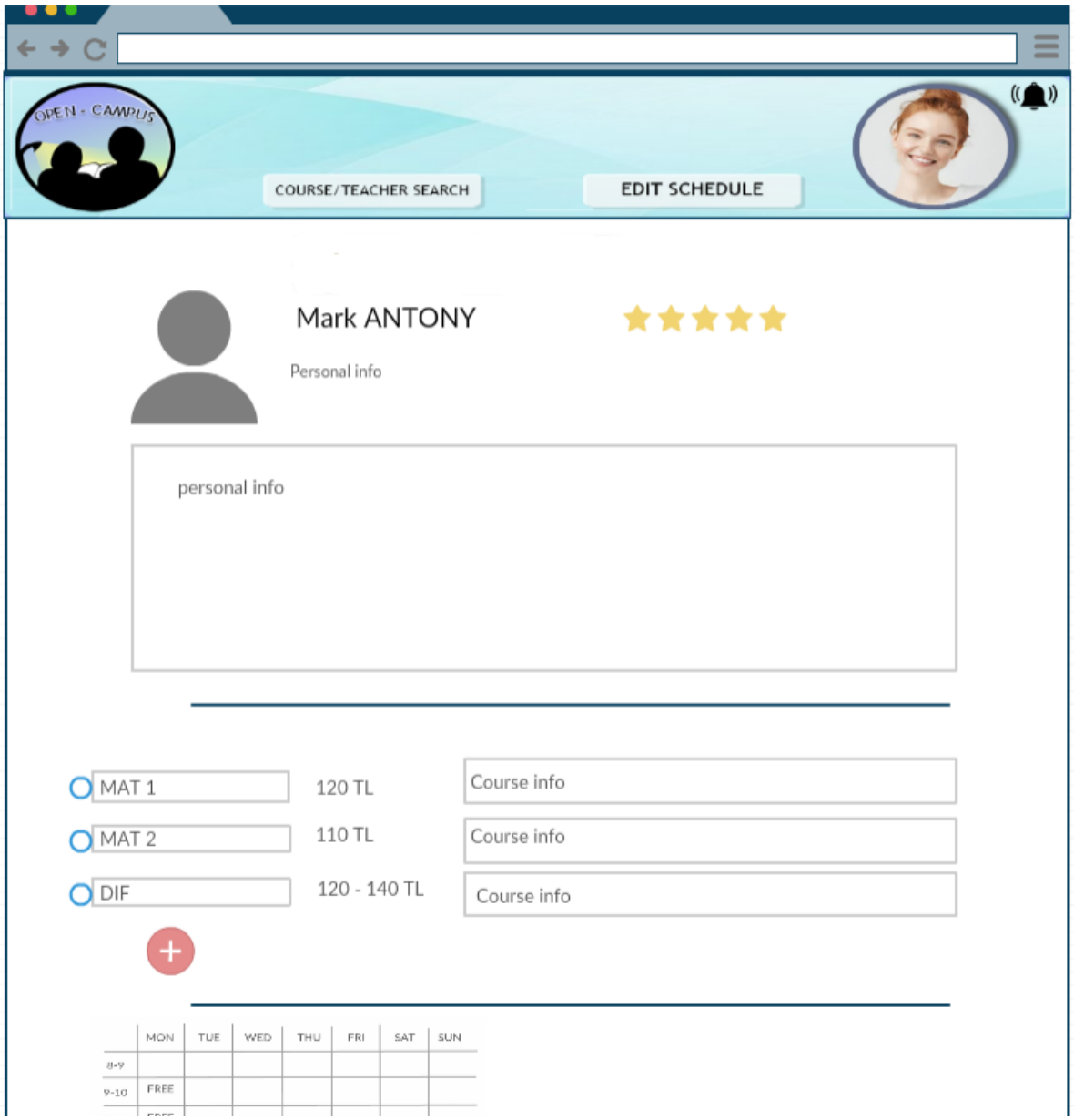



**Figure 13 Teacher Profile Page**

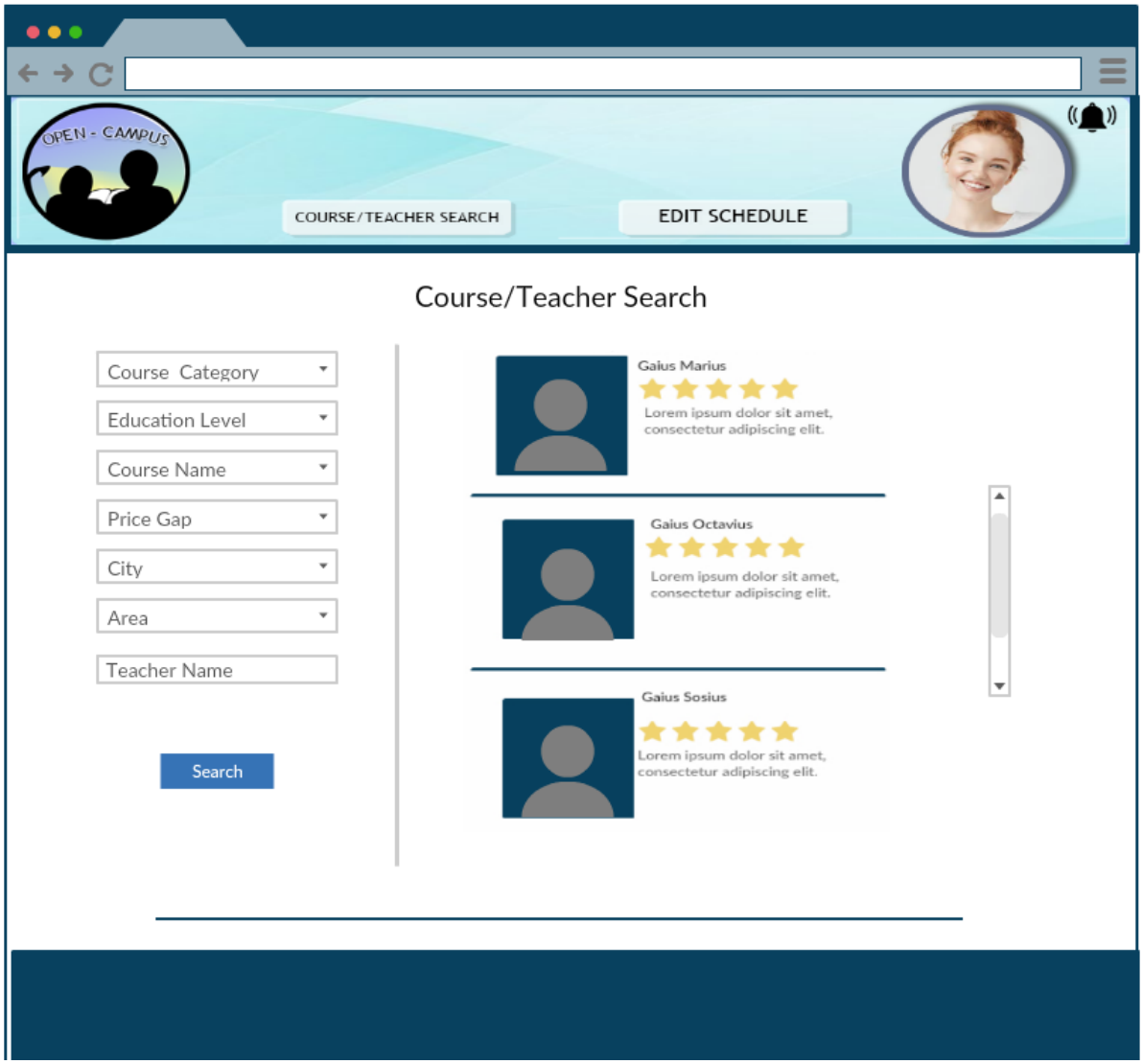



**Figure 14 Course/Teacher Search Page**

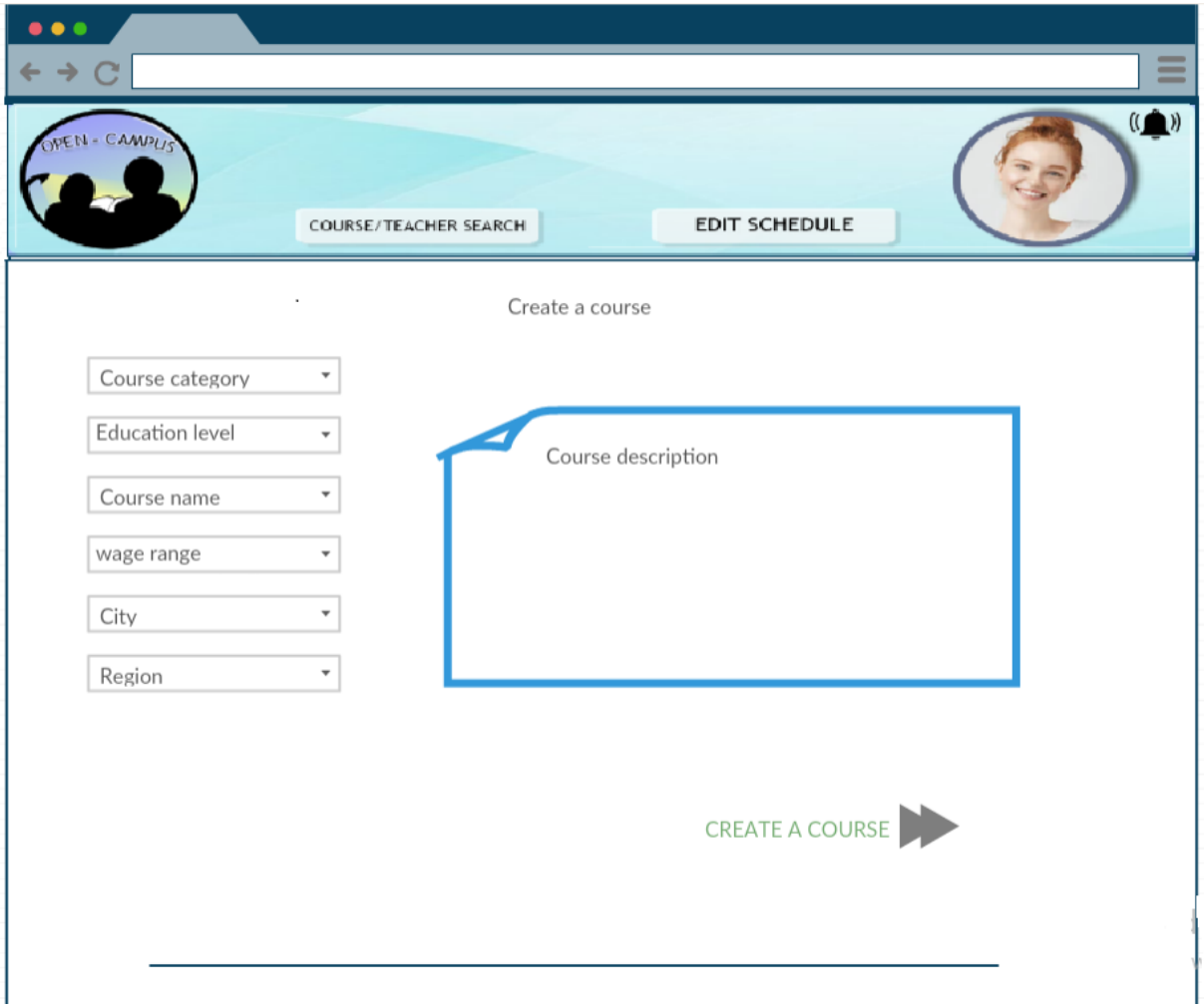



**Figure 15 Create Course Page**

# References


[1] Classgap, https://www.classgap.com/en

[2] Job Profiles - Private Tutor (Prospects), https://www.prospects.ac.uk/job-profiles/private-tutor

[3] Private Tutoring (Success Makers), https://www.successmakers.org.uk/

[4] Videoconference (SearchUnifiedCommunications), https://searchunifiedcommunications.techtarget.com/definition/video-conference

[5] Artificial Intelligence (Stanford University), http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html

[6] What is AI? (Oracle), https://www.oracle.com/tr/artificial-intelligence/what-is-artificial-intelligence.html

[7] File Sharing (QnapWorks), http://www.qnapworks.com/Features-File-Sharing.asp

[8] File Sharing on Computer Networks (Lifewire), https://www.lifewire.com/file-sharing-on-computer-networks-817371

[9] Online Chat (Wikipedia), https://en.wikipedia.org/wiki/Online_chat

[10] How Chat Rooms Work (Howstaffworks), https://computer.howstuffworks.com/internet/social-networking/information/chat-room.htm

[11] Instant Messaging (Wikipedia), https://en.wikipedia.org/wiki/Instant_messaging

[12] How Instant Messaging Works (Howstaffworks), https://computer.howstuffworks.com/e-mail-messaging/instant-messaging.htm

[13] ICQ, https://icq.com/

[14] IRC: Internet Relay Chat (Google), https://developers.google.com/open-source/gsoc/resources/irc

[15] Internet Relay Chat Protocol (IETF | Internet Engineering Task Force), https://tools.ietf.org/html/rfc1459

[16] MSSQL (Atlantic), https://www.atlantic.net/what-is-mssql/

[18] C# (Technopedia), https://www.techopedia.com/definition/26272/c-sharp

[19] HTML (W3Schools), https://www.w3schools.com/html/default.asp

[20] HTML Tutorial (The Site Wizard), https://www.thesitewizard.com/html-tutorial/what-is-html.shtml

[21] Javascript (Technopedia), https://www.techopedia.com/definition/3929/javascript-js