# 4.ICES4HU
# Software Design Description

## 1.    Revision History

| Version | Date | Author | Change Description |
|---|---|---|---|
| 1 | 26.05.2023 | Zübeyde Civelek | Introduction and, Design Constraints and Decisions |
| 1 | 26.05.2023 | Ayşe İrem Yalçın | User Interface Design |
| 1 | 26.05.2023 | Ayça Akyol | Document Overview |
| 1 | 26.05.2023 | Şura Nur Ertürkmen | Update ER diagram and Class Diagram |
| 1 | 26.05.2023 | Selahattin Can Ölçer | Final Documentation and Overview |

## 2.  INTRODUCTION

### 2.1   Purpose and Scope

The purpose of this document is to provide a comprehensive overview of the ICES4HU project's design, ensuring its alignment with the functional and non-functional requirements outlined in the Software Requirements Specification (SRS) document. The Software Design Document (SDD) serves as a detailed representation of the project, offering extensive information to stakeholders involved in the design process.

The SDD aims to address design concerns by presenting architectural diagrams, design illustrations, and various tools that effectively portray the structure of the software. Additionally, it encompasses references to comprehensive specifications of smaller components within the software design. By doing so, the SDD facilitates a clear understanding of the project's design and fosters effective communication between the design team and stakeholders.

### 2.2   Document Overview

This document consists of 6 sections, which are outlined below:

1.   Revision History: This section is used to keep track of changes made to the document over time.
2.   Introduction: The introduction section is further divided into five subsections, each serving a specific purpose:

> a. Purpose and Scope: This subsection describes the overall purpose and scope of the document, providing an overview of its objectives and intended audience.

> b. Document Overview: The document overview subsection summarizes the sections included in this document, providing a high-level summary of its contents.

> c. System Overview: In this subsection, a general description of the complete system is provided using diagrams or other visual representations to give readers a comprehensive understanding of its structure.

> d. Definitions, Acronyms, and Abbreviations: This subsection includes a list of all non-standard terms, acronyms, and abbreviations used throughout the document. It ensures that readers have a clear understanding of the specific terminology employed.

e. References: This subsection lists any external references or sources that were used to inform the design decisions and development process.

3. Design Constraints and Decisions: This section explores the general constraints imposed by the design process and discusses their impact on the system architecture. It also addresses any constraints imposed by the hardware and software environment and outlines the design decisions made to overcome these limitations.

4. Design Details: The design details section expands on the design model defined during the architectural design phase. It consists of four subsections:

    a. Software Components: This subsection includes a class diagram depicting the software components and their relationships within the system.

    b. Software Behavior: The software behavior section provides detailed information on the system's functionality.

    c. Data Model (E-R Diagram): In this subsection, an Entity-Relationship (E-R) diagram is presented, illustrating the components of the data model and their relationships.

    d. User Interface Design: The User Interface Design section showcases the interfaces of the software. It presents an updated and more detailed version of the interfaces described in the SRS file.

5. Requirements Traceability: This section establishes the relationship between the software requirements and the software design. It may include a traceability matrix that maps classes or components to specific requirements, demonstrating how the design addresses and fulfills each requirement.

6. Annexes: The Annexes section contains additional information and details that require further explanation.

## 2.3   System Overview

ICES4HU (Instructor and Course Evaluation System for Hacettepe University) is an online platform specifically designed for Hacettepe University to facilitate the evaluation of instructors and courses. It provides an easy-to-use interface. 6 key functionalities of the system are provided in a diagram.
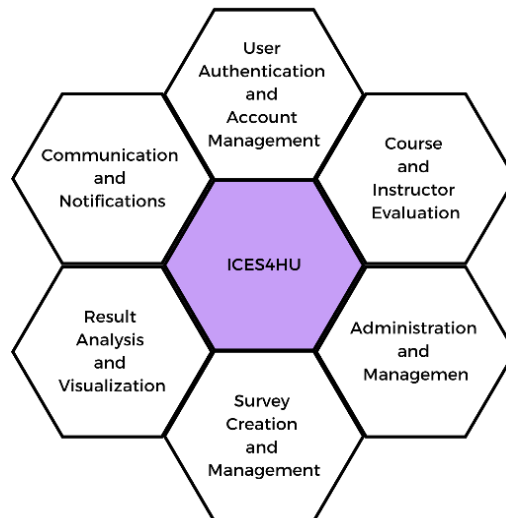
The system aims to improve the quality of education by enabling students to provide feedback through electronic surveys.The system consists of four main modules: Administration, Student, Instructor, and Department Manager. Each module provides distinct functionalities tailored to the needs of the respective user roles.

ICES4HU is designed with a flexible, maintainable, attractive, and extensible design. The system follows the GitHub flow for collaborative development and deployment. It is built on the Model-View-Controller (MVC) pattern, which divides the system into three logical components: Model, View, and Controller. The Model component handles data management and associated operations, the View component manages data presentation to users, and the Controller component manages user interactions and communication between the View and Model components.

With its comprehensive functionalities and user-friendly interface, ICES4HU provides a reliable and efficient platform for instructor and course evaluation at Hacettepe University, empowering stakeholders to enhance the quality of education and foster continuous improvement.
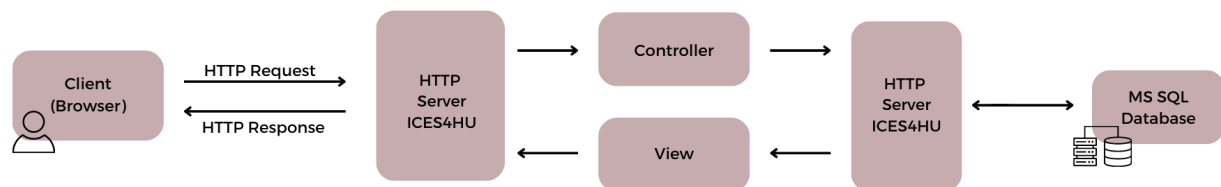


The project's database is chosen to be MS SQL. The MVC architecture is used to return the response when a client submits a request to the system.

### 2.4 Definitions, Acronyms, and Abbreviations

- SRS – System Requirements Specification

- MVC - Model-View-Controller

- GUI - Graphical User Interface

## 3. Design Constraints and Decisions

The project is built on the principles of the Model-View-Controller (MVC) architectural design. It comprises three main components: the model, view, and controller. The system requires users to be logged in and registered in order to access its functionalities. For logged-out users, only the login page is visible. The user registration process is subject to approval by administrators, ensuring flexibility within the constraints of web applications.

The back-end of the system is developed using C#-programmed ASP.NET, while the preferred database is MS SQL. On the frontend, Javascript, React, and CSS are utilized. ASP.NET facilitates the implementation of the MVC architecture, allowing for effective separation of concerns.

Software development and version control are managed through GitHub. Issues and bugs are tracked within the Projects section of the demo-final repository, which contains all the project's code.

Administrators hold the responsibility of managing user accounts and handling associated tasks. To ensure maximum availability, the system is designed to be accessible 24/7, with a target uptime of at least 99%.

Confidential communication between students and administrators is safeguarded by employing encryption measures to maintain the privacy and security of sensitive information.

The system provides clear and informative error messages to users, guiding them on how to handle encountered errors. The graphical user interface (GUI) is designed to be intuitive and straightforward, enhancing user experience and ease of navigation.
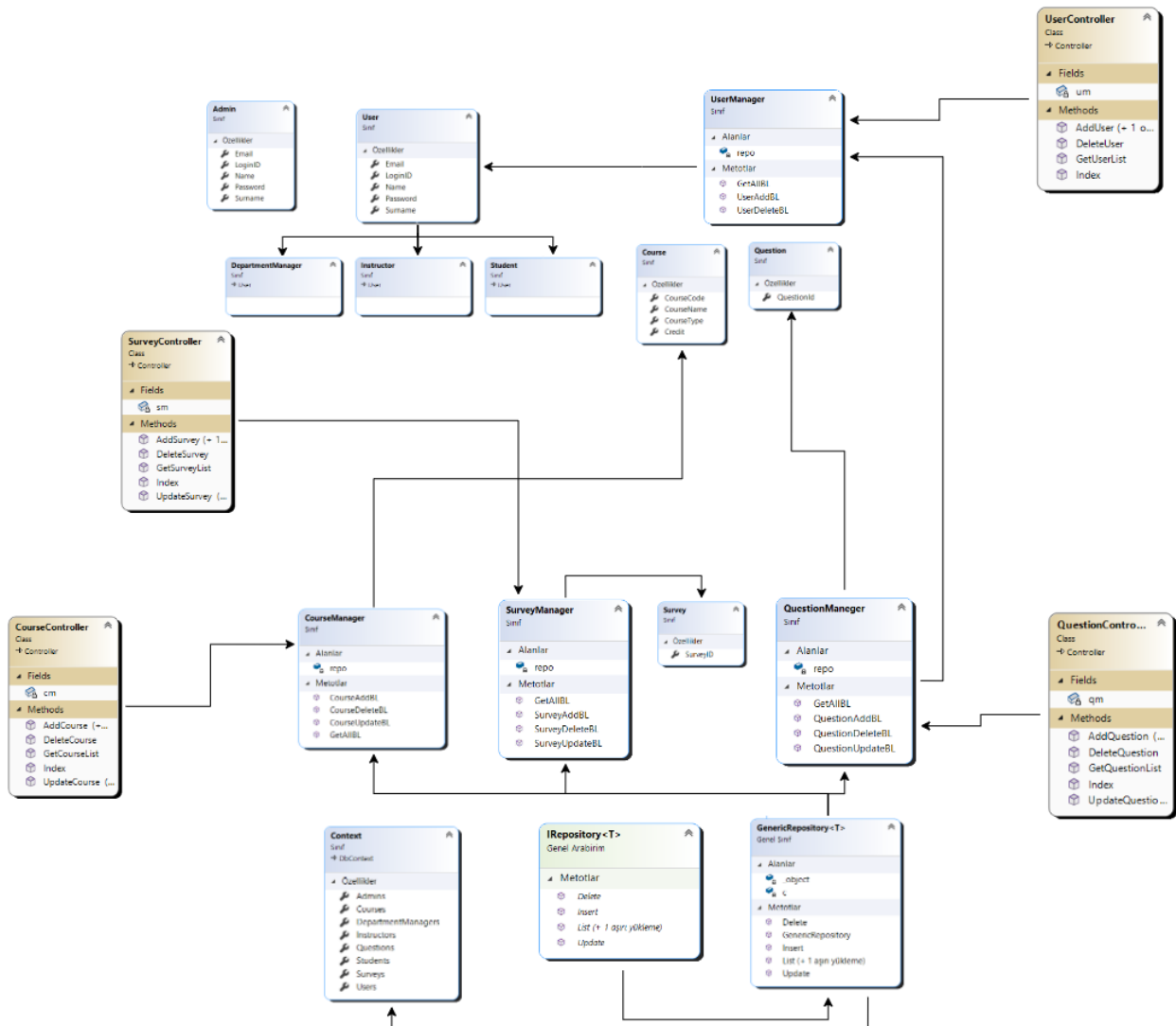
Compatibility is prioritized, and the system is designed to function seamlessly across different operating systems and browsers, ensuring a consistent user experience regardless of the platform used.
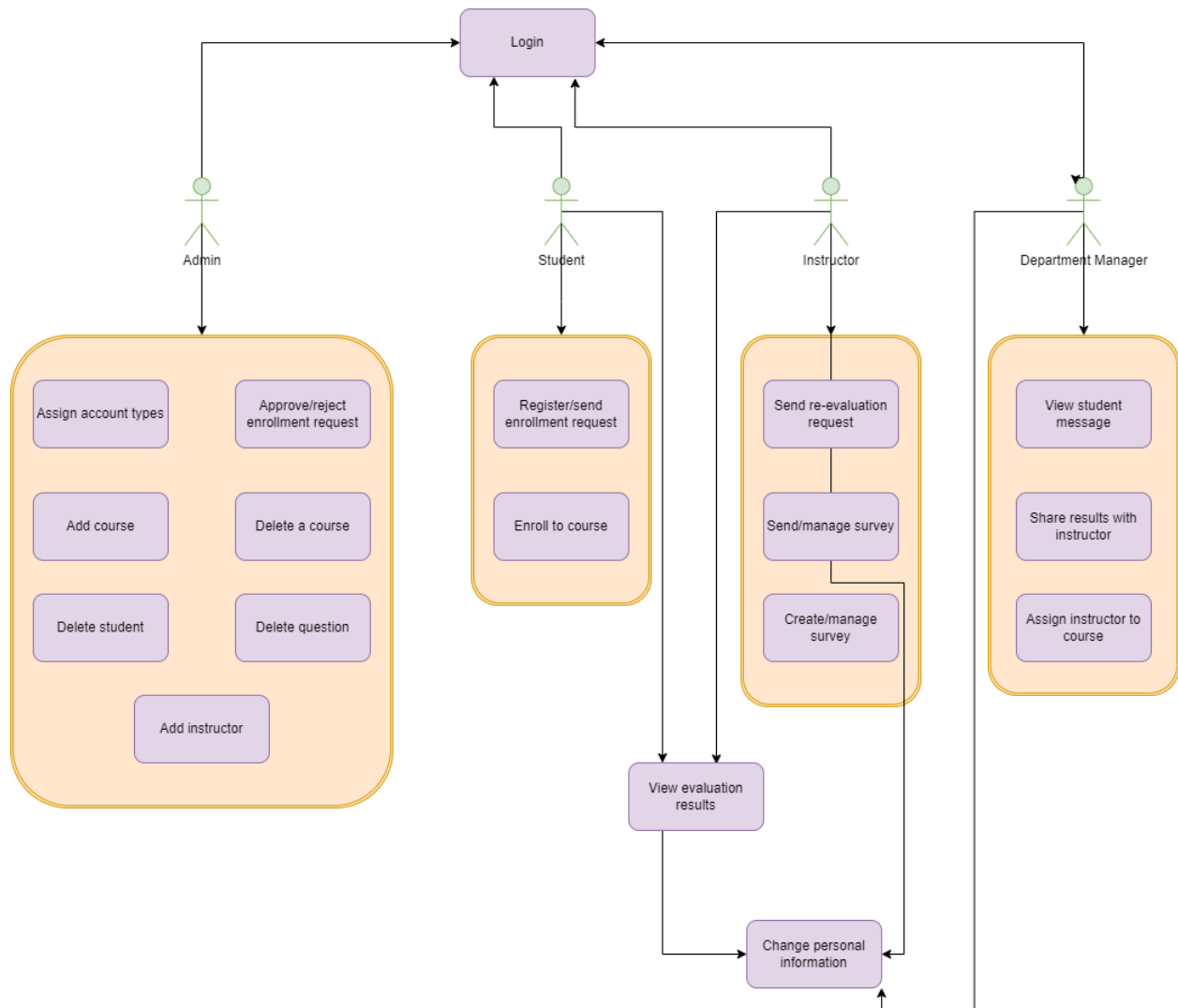
## 4.   Design Details

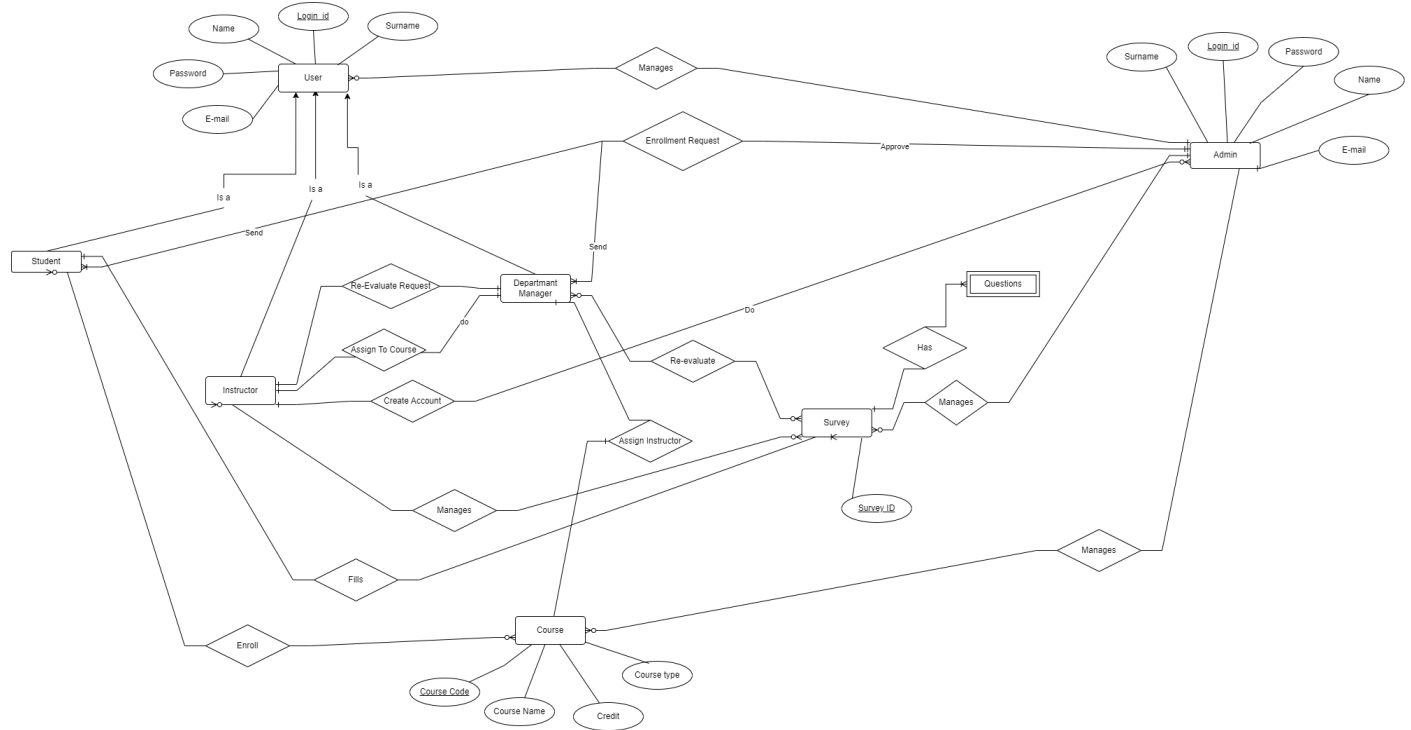### 4.1    Software Components

### Class Diagram

## 4.2    Software Behavior

## Use Case Diagram

### 4.3    Data Model (E-R Diagram)

#### E-R Diagram

## 4.4 User Interface Design

### 4.4.1 Change Personal Information Page

### 4.4.2 Answer Survey

### 4.4.3 Student Messages Admin



### 4.4.4 Admin Menu

### 4.4.5 Admin Manage Course

## Manage Course

| | Course Name | Course Code | Credit | Course Type |
|---|---|---|---|---|
| ☐ | Software Engineering | BBM382 | 4 | Mandatory |
| ☐ | Data Structures | BBM201 | 5 | Mandatory |
| ☑ | Algorithims | BBM202 | 5 | Mandatory |
| ☐ | Computer Vision | BBM416 | 6 | Elective |
| ☐ | Machine Learning | BBM406 | 6 | Elective |

Delete Selected Courses    Add New Course

## 4.4.6 Add Instructor / Accept Pending Enrollment Requests

## 5. Tracebility Table

| Use Case ID | UC-1 | UC-2 | UC-3 | UC-4 | UC-5 | UC-6 | UC-7 | UC-8 | UC-9 | UC-10 | UC-11 | UC-12 | UC-13 | UC-14 | UC-15 | UC-16 | UC-17 | UC-18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Classes** | | | | | | | | | | | | | | | | | | |
| Survey | | | | | | | | | | | | ☑ | | ☑ | | | | ☑ |
| SurveyController | | | | | | | | | | | | ☑ | | ☑ | | | | ☑ |
| User | ☑ | | | | ☑ | | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | | | |
| UserController | ☑ | | | | ☑ | | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | | | |
| Admin | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ | | | | | | | | | | |
| Course | | | ☑ | ☑ | | | | | | | | | | | | ☑ | | |
| CourseController | | | ☑ | ☑ | | | | | | | | | | | | ☑ | | |
| Question | | | | | ☑ | | | | | | | | | | | | | |
| QuestionController | | | | | | | | | | | | | | | | | | |
| EnrollmentRequest | | | | | | ☑ | | | ☑ | | | | | | | | | |
| EnrollCourse | | | | | | | | | | ☑ | | | | | | | | |
| Response | | | | | | | | | | | | | | | | | | |
| ResponseController | | | | | | | | | | | | | | | | | | |
| Re-Evaluation | | | | | | | | | | | | | | | ☑ | | ☑ | |
| Re-EvaluationController | | | | | | | | | | | | | | | ☑ | | ☑ | |
| Account | ☑ | ☑ | | | | | | | | | ☑ | | | | | | | |
| AccountController | ☑ | ☑ | | | | | | | | | ☑ | | | | | | | |
| Evaluation | | | | | | | | | | | | | ☑ | | ☑ | | | |
| EvaluationController | | | | | | | | | | | | | ☑ | | ☑ | | | |
| Intructor | | | | | | | ☑ | ☑ | | | | ☑ | ☑ | ☑ | ☑ | ☑ | | ☑ |
| Student | | | | | | | ☑ | | ☑ | | | | ☑ | | | | | |
| DepartmentManager | | | | | | | ☑ | | | | | | | | | ☑ | ☑ | ☑ |

## 6 Sequence and State Chart Diagrams

## -> AppendixSequenceandStateChartDiagrams