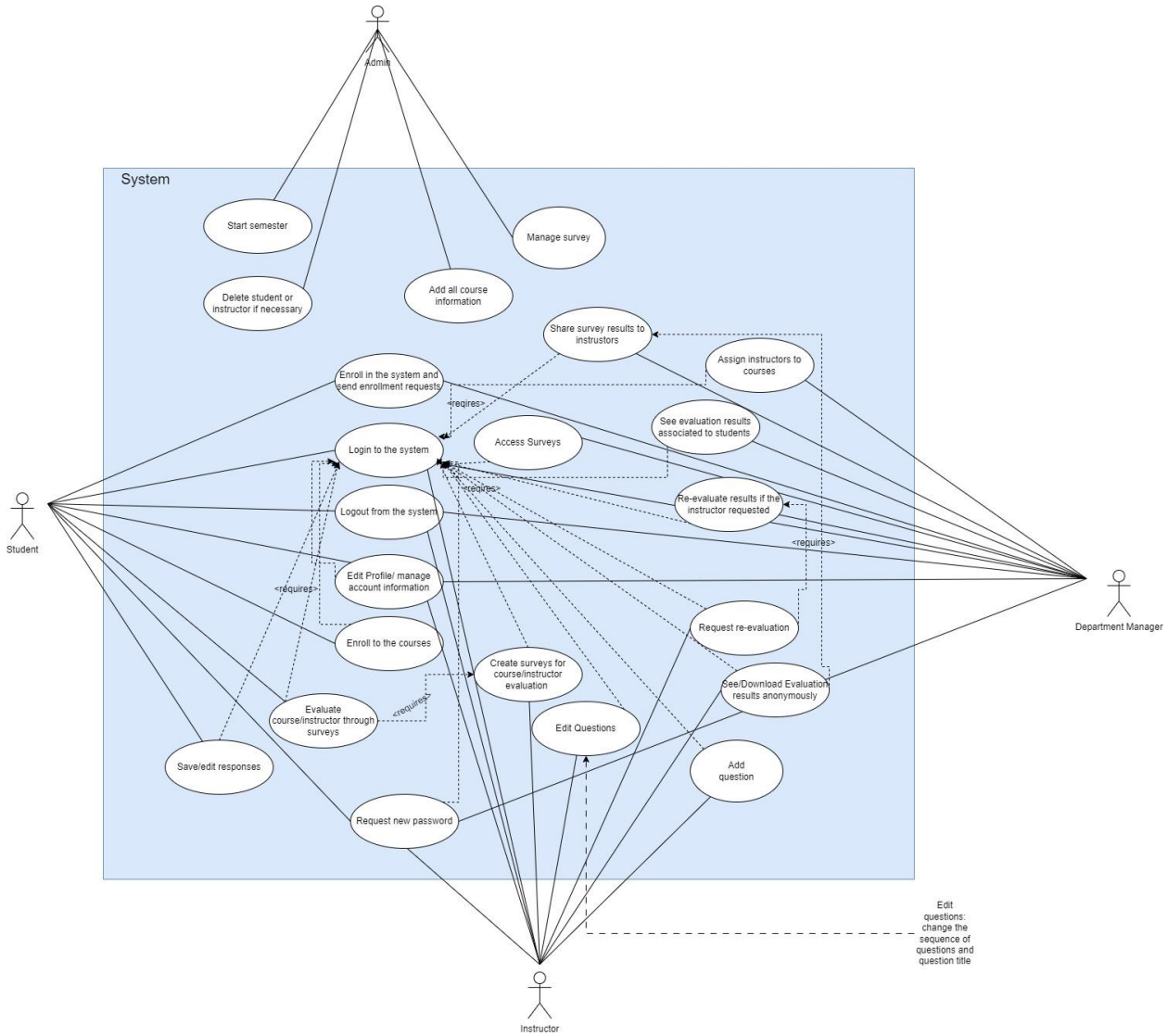
ICES4HU
System-Wide Requirements Specification

1. Introduction

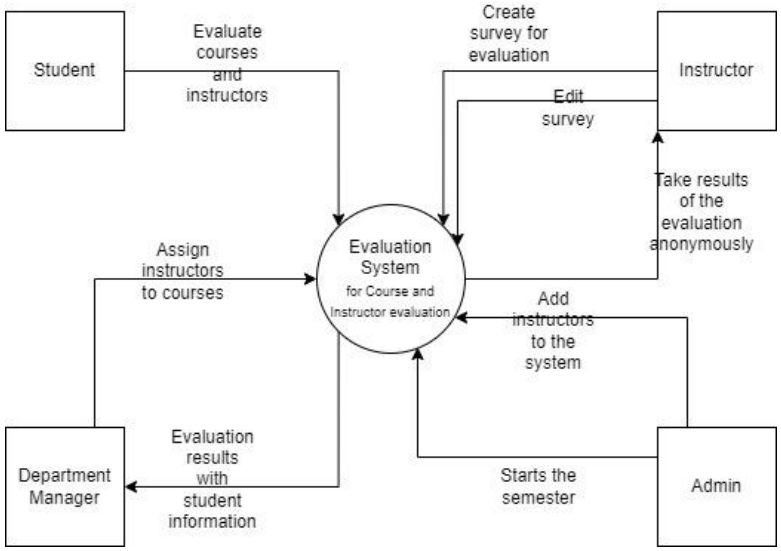
A web based system called “Instructor and Course Evaluation System for Hacettepe University” (ICES4HU) was created to make the process of evaluating courses and instructors in universities more efficient. Students can use the system’s user-friendly interface to evaluate their courses and instructors, and department managers can view and analyze the forms. To guarantee that evaluations are done on time, ICES4HU also has features like customisable evaluation forms and automated reminders. ICES4HU aims to improve the quality of education at universities by providing relevant feedback on the effectiveness of courses and instructors.

2. System-Wide Functional Requirements

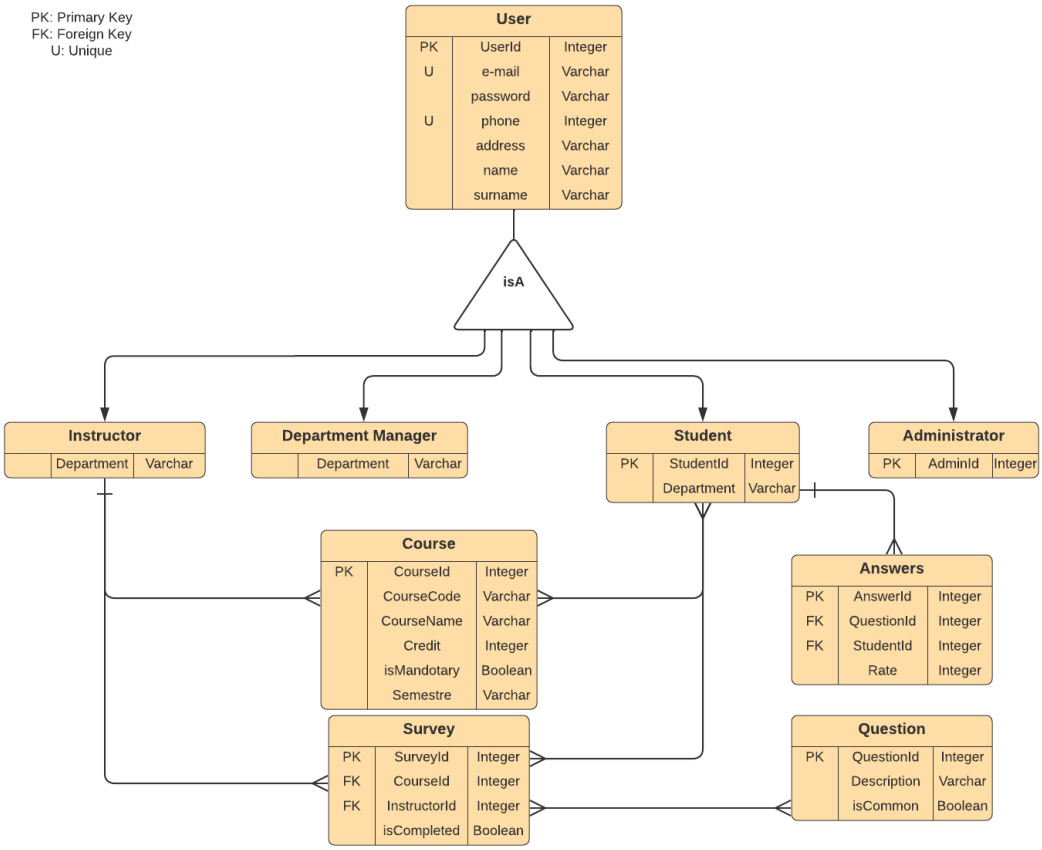
The system’s use case diagram:



The system's context diagram:



The system's ER diagram:



The system's use cases:

Use Case	Use Case Number	Use Case Code
Enroll to the System	1	ICES4HU-UC-001
Login	2	ICES4HU-UC-002
Logout	3	ICES4HU-UC-003
Edit Profile/ Manage Account	4	ICES4HU-UC-004
Request new Password	5	ICES4HU-UC-005
Evaluate course/instructors	6	ICES4HU-UC-006
Save/Edit Response	7 (7.1 and 7.2)	ICES4HU-UC-007-1 ICES4HU-UC-007-2
Enroll to courses	8	ICES4HU-UC-008
Create Survey	9	ICES4HU-UC-009
Add question	10	ICES4HU-UC-010
Edit question	11	ICES4HU-UC-011
See/Download evaluation results anonymously	12 (12.1 and 12.2)	ICES4HU-UC-012-1 ICES4HU-UC-012-2
Request re-evaluation	13	ICES4HU-UC-013
Access Surveys	14	ICES4HU-UC-014
See evaluation results(w. names)	15	ICES4HU-UC-015
Assign instructor to courses	16	ICES4HU-UC-016
Re-evaluate results	17	ICES4HU-UC-017
Share survey results w. instructors	18	ICES4HU-UC-018
Start semester	19	ICES4HU-UC-019
Add all course information	20	ICES4HU-UC-020
Delete the student/instructor	21	ICES4HU-UC-021
Manage surveys	22	ICES4HU-UC-022

Students should be able to choose their instructor or course to give feedback on course content or instructor.
It should be possible for instructors to create and manage course evaluations.
Instructors should be able to view evaluations of their courses anonymously, as well as monitor their own

performance metrics and receive feedback or evaluation analysis from administrators.

Administrators should be able to manage user accounts, course offering timings, and evaluation schedules, as well as generate reports and analytics about evaluations.

The department manager should have access to the profiles of the students who submitted evaluation forms also, if requested by instructors, they can reevaluate the forms.

System administrators should be able to set and manage user roles and permissions for various system components using role-based access control, which should be provided by the system.

Based on their roles and permissions in the system, users should only be allowed to view and access evaluations for courses or instructors with whom they are associated.

3. System Qualities

To provide a user-friendly, stable, effective, and maintainable system, the course and instructor evaluation system should be developed for usability, reliability, performance, and supportability.

3.1 Usability

- The system must have a user-friendly interface and clear instructions to make it simple to navigate and complete tasks quickly.
- The system must inform users in a clear and simple way about the results of their actions and other potential problems could have occurred.
- The system should offer a search bar that allows users to quickly find courses and instructors based on keywords, categories, or other relevant criteria.
- A consistent user experience should be offered by the system on a variety of platforms and devices, including desktop and laptop computers, tablets, and smartphones.
- The system should include simple directions on how to complete evaluations, including how to score the different survey questions about the course or instructor.
- Users should be able to preserve their progress in the system and come back to evaluations at a later time if necessary.
- Users should be able to quickly access help or support from the system if they run into any issues while completing the evaluation.

3.2 Reliability

- The system should be stable and perform consistently, with minimal downtime or system failures.
- In the event of a system failure, the system should have backup and recovery procedures to guarantee that user data is not lost.
- The system must be built to handle unexpected input or mistakes without collapsing or corrupting data.
- To make sure the system is reliable and performs as expected, it should pass tests before being released.
- To protect user data and prevent unauthorized access, the system should be equipped with the necessary security procedures.
- Any system faults or concerns should have a clear and efficient reporting and repair method.

3.3 Performance

- The system should be designed for speed and efficiency with little lag and short loading time.
- The system should be able to handle a large number of concurrent users and evaluations without significant delays or system overload.
- The system must be able to process evaluation data of different sizes and types, and it must produce reports and analytics with accuracy and speed.
- The system should be built to support large numbers of concurrent users without facing performance loss.
- The system should be able to generate reports and analytics fast, enabling users to examine and understand evaluation data in real-time.

3.4 Supportability

- It should be simple to update and maintain the system, and administrators and users should have access to clear documentation.

- To enable future updates and integration with other systems, the system should be developed with modularity and flexibility in consideration.
- To help users with technical problems and questions, the system should include a quick and effective customer support service.
- To help administrators identify and solve system issues, the system must include debugging tools and clear, brief error messages.

4. System Interfaces

4.1 User Interfaces

4.1.1 Look & Feel

The system should have a modern and intuitive user interface, with a clean and consistent design that is visually appealing and easy to navigate. The color scheme, typography, and graphics should be carefully chosen to enhance the user experience and reflect the brand identity of the university.

The user should feel as though they are in a serious but also welcoming environment.

4.1.2 Layout and Navigation Requirements

The system should have a clean, well-organized interface with elements grouped logically and buttons and menus placed consistently. With obvious labels and descriptions for each feature or part, navigation should be easy to use. Users should be given a breadcrumb trail or other navigational assistance by the system to help them maintain track of their position within it.

4.1.3 Consistency

To make sure users can quickly recognize and understand how to interact with the different system components, the system should maintain a consistent design and user interface across all pages and functions. All buttons, links, and other interface elements should be labeled and placed consistently, and a consistent color palette, layout, and typography should be used throughout the system.

4.1.4 User Personalization & Customization Requirements

Users should be able to customize the system so that it fits their needs by changing things like their profile picture and notification preferences.

User passwords must be less than 18 characters in length and greater than 8 characters in length. A user may enter any password s/he likes if that rule is applicable.

4.2 Interfaces to External Systems or Devices

4.2.1 Software Interfaces

The Maven database will be utilized for this project. This database was chosen because it is simple for the public to access. Java Database Connectivity API is required to connect to this database (JDBC). The selection of this API was determined in part for its effective connection tools and the usage of Java language in the system's development.

4.2.2 Hardware Interfaces

The Department Server must be accessible to the development team.

4.2.3 Communications Interfaces

There will be no communication interface for this project.

5. Business Rules

5.1 Evaluation Rules

5.1.1 Course Evaluation Form Requirement Rule

Each course must have at least one evaluation form associated with it.

5.1.2 Student Evaluation Submission Limit Rule

Students can only submit one evaluation for each course per semester.

5.1.3 Completion Timeframe

Evaluations must be completed in the specific timeframe.

5.1.4 Minimum Number of Questions Rule

Each evaluation must have a minimum set of questions, defined by the university before.

5.2 Student Rules

5.2.1 Reminder Rule

Students must be reminded to give feedback to evaluations to ensure a high response rate.

5.2.2 Evaluation Completion Confirmation Rule

After completing their evaluations, students should receive notification that their evaluations were successful.

5.3 Instructor Rules

5.3.1 Evaluations Access Rule

Instructors cannot view evaluation results until the final grades have been submitted.

5.3.2 Evaluator Anonymity Rule

To guarantee anonymity and prevent discrimination, instructors must be unable to see who submitted the evaluation.

5.4 System Rules

5.4.1 Privacy Rule

All user data must be kept private and secret by the system, in line with the laws and regulations.

5.4.2 System Availability Rule

System must be available 24/7, with minimum downtime and interruption.

5.4.3 Reporting and Analytics Rule

Based on the information gathered through evaluations, the system must deliver reliable and accurate reporting and analytics.

6. System Constraints

This project's user interface (UI) part will be developed using React, and the backend part will be developed using Java Spring-Boot.

To make sure the system is reliable and error-free, it must undergo extensive testing on a regular basis.

The program must work with many operating systems (e.g., Windows, macOS, Linux). and popular web browsers like Chrome, Firefox, Safari, and Edge.

7. System Compliance

7.1 Licensing Requirements

- ICES4HU provides an agreeable, user-friendly licensing agreement for its users. This system is specific for Hacettepe University.
- For students, legal agreement with ICES4HU ends when s/he graduates from Hacettepe University. For instructors, legal agreement with ICES4HU ends when s/he ends his/her agreement with Hacettepe University.

7.2 Legal, Copyright, and Other Notices

ICES4HU system provided by REACTion supplies information only for Hacettepe University. Since this is a survey platform the answers gathered will be kept confidential unless an emergency occurs.

All users will be responsible for their acts in ICES4HU platform. REACTion accepts no responsibility or liability for the use of the software.

REACTion reserves the right to change software without notification to users. However, Hacettepe University management must be discussed in accordance with this change since the system is unique for Hacettepe University.

7.3 Applicable Standards

The system shall adhere to the legal requirements pertaining to copyright and safeguarding of personal data in Turkey, ensuring full compliance with the statutory regulations of each respective jurisdiction.

In the ICES4HU project, the agreed decision is to follow Java coding standards.

8. System Documentation

User Guide: This guide offers step-by-step instructions for using the system, including how to access and navigate various interface elements, how to create and manage evaluations, and how to generate reports or analytics.

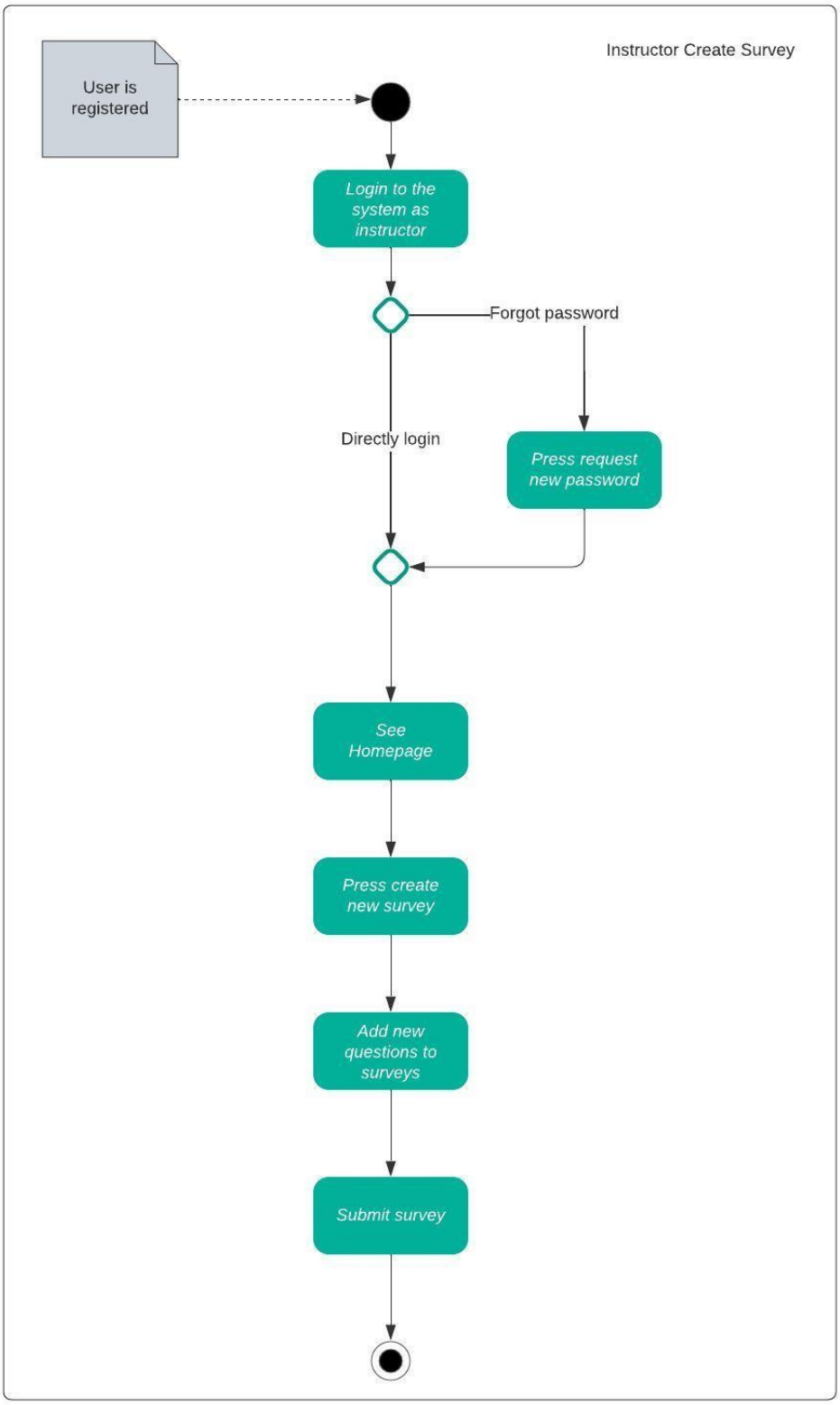
Release Notes: This document includes any known issues or limitations along with the modifications, bug fixes, and upgrades made in each system release or update.ha

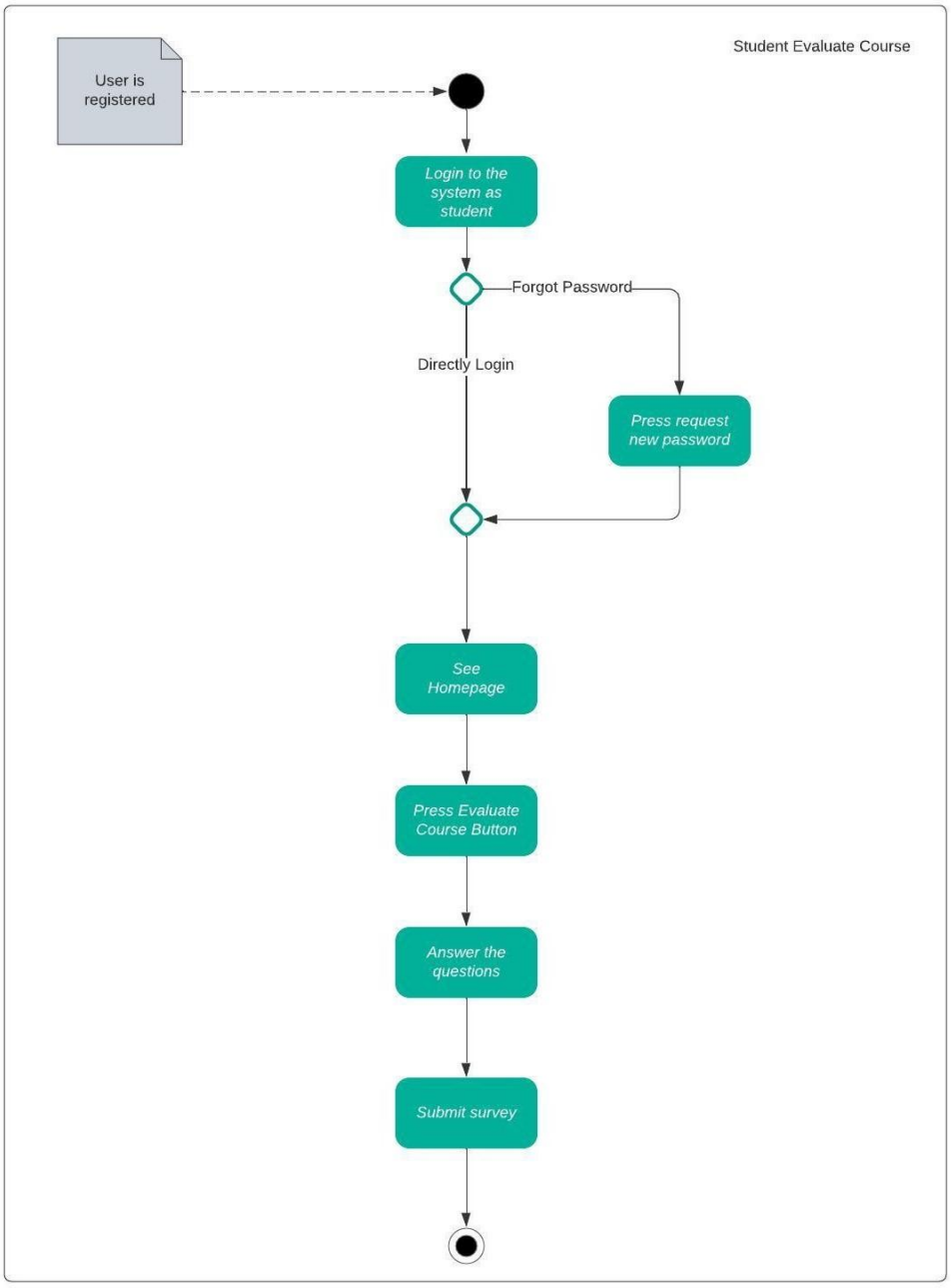
Installation Guide: There won't be an installation guide because the system is a web page.

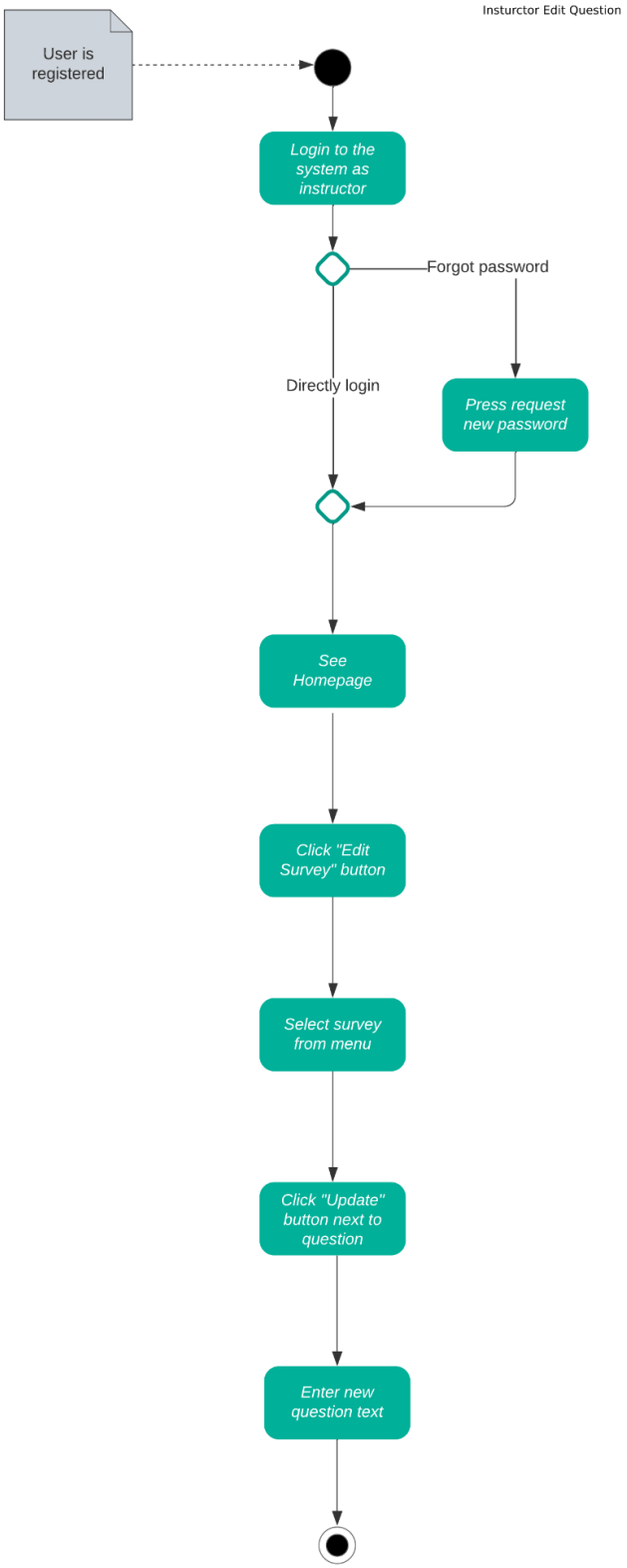
Support Materials: These could be FAQs, how-to manuals, or other tools to assist users or administrators in solving typical problems or questions.

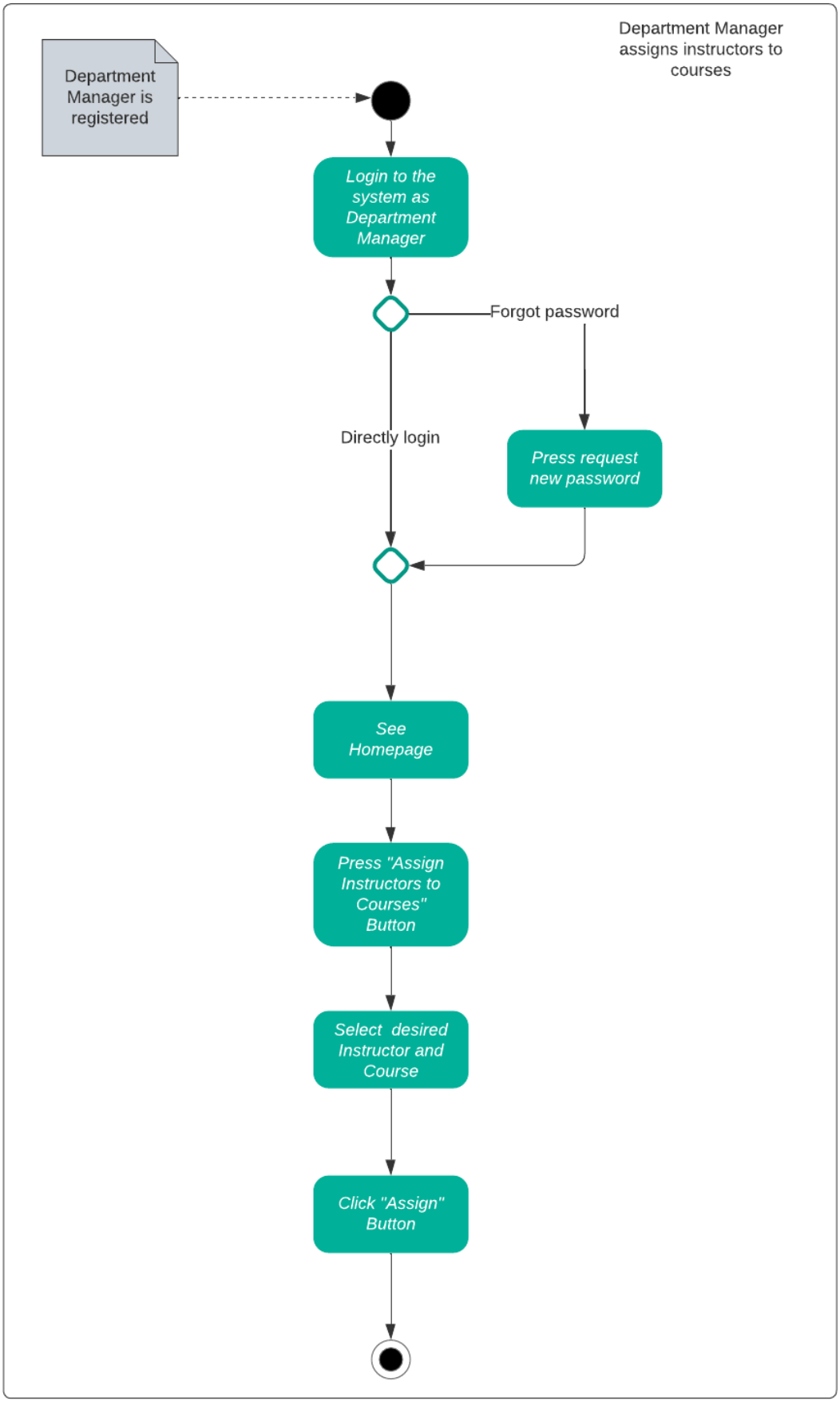
Technical Specification: The architecture, design, and implementation of the system are extensively described in this paper, along with details on the techniques developed, the data model, and the system interfaces.

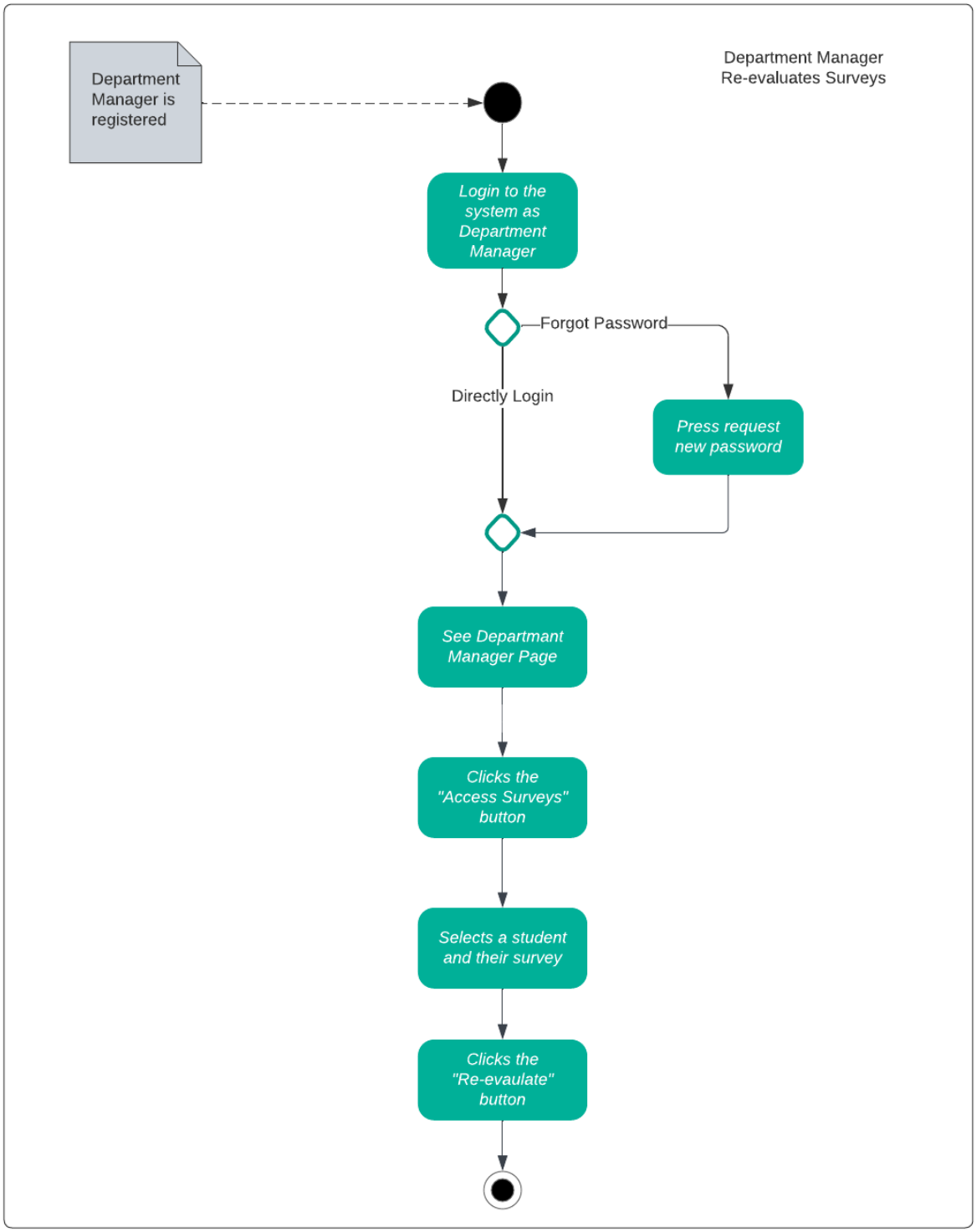
The system's activity diagrams:




Student Evaluate Course
User is registered
Login to the system as student
Forgot Password
Directly Login
Press request new password
See Homepage
Press Evaluate Course Button
Answer the questions
Submit survey


Insturctor Edit Question
User is registered
Login to the system as instructor
Forgot password
Directly login
Press request new password
See Homepage
Click "Edit Survey" button
Select survey from menu
Click "Update" button next to question
Enter new question text


Department Manager assigns instructors to courses
Department Manager is registered
Login to the system as Department Manager
Forgot password
Directly login
Press request new password
See Homepage
Press "Assign Instructors to Courses" Button
Select desired Instructor and Course
Click "Assign" Button


Department Manager is registered
Department Manager Re-evaluates Surveys
Login to the system as Department Manager
Forgot Password
Directly Login
Press request new password
See Departmant Manager Page
Clicks the "Access Surveys" button
Selects a student and their survey
Clicks the "Re-evaulate" button

	Deniz Erkin Kasaplı (Software Project Manager)	Berkay Barulay (Software Analyst)	Ahmet Karaca (Software Architect)	Gizem Aleyna Tuzcu (Software Configuration Manager)	Hasan Malkoç (Software Tester)
Software Requirement Specification	***X***		***X***	***X***	
Graphical User Interface Design	***X***	***X***		***X***	
Graphical User Interface Design Document		***X***			
Use Case Definitions Document	***X***	***X***	***X***	***X***	***X***
Test Case Definitions Document					***X***
Activity Diagram	***X***	***X***	***X***	***X***	***X***
Use Case List	***X***	***X***	***X***	***X***	***X***

Use Case Diagram	***X***				
Context Diagram	***X***				
ER Diagram			***X***		***X***