ICES4HU
Software Design Description

1.   Revision History

Version	Date	Author	Change Description
1	21.05.2023	Hasan Malkoç	Created the document
1	21.05.2023	Gizem Aleyna Tuzcu	Generated the UML diagrams and edited the informations in all sections
1	23.05.2023	Berkay Barulay	Edited the User Interface Design section.

2. INTRODUCTION

2.1   Purpose and Scope

A software design description document is a report that details the overall architecture of a software product. The software development team receives such design documents, which are often created by software designers or project managers and provide an overview of what must be built and how, from these parties. A software design document aids in ensuring that everyone is aware of and understands the software's design specifications. It details what the product is capable of doing and how it can be done.

2.2   Document Overview

The Instructor and Course Evaluation System's design and architecture are described in this Software Design Document. It describes the system's architectural elements, data flow, and relationships between various modules, as well as its functional and non-functional requirements. The document also offers information on security considerations, and integration specifications for the system.

2.3   System Overview

ICES4HU provides a platform for students to evaluate the courses and professors which they have taken over the semester, allowing them to make improvements to the educational institution. The system includes four main modules. These Are the administration module,student module,instructor module,and department manager module. These 4 modules and their interaction with the system look like this:
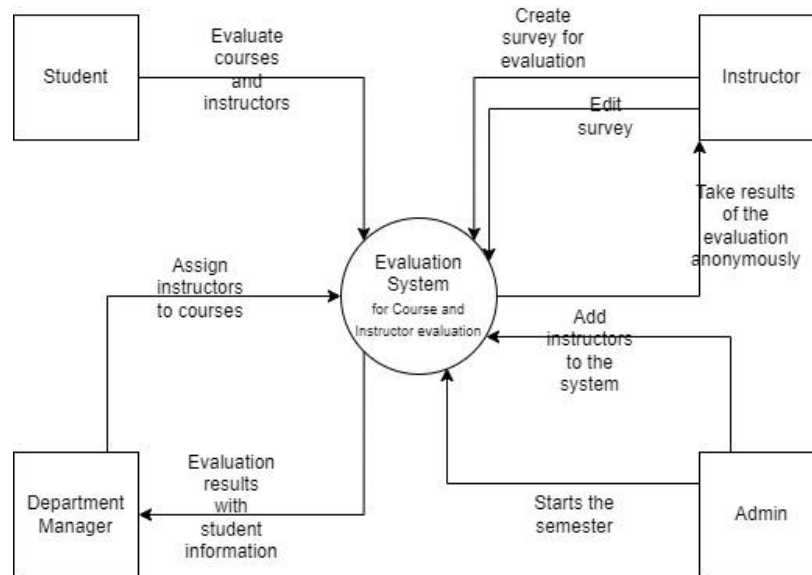
2.4   Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
API	Application Programming Interface
ER	Entity-Relationship
GUI	Graphical User Interface

3.   Design Constraints and Decisions

- The web application for the course and instructor evaluation system was built using:
    - React
    - JavaSpring
    - PostgreSQL

- The application assures:
    - Platform compatibility
    - Security protocols
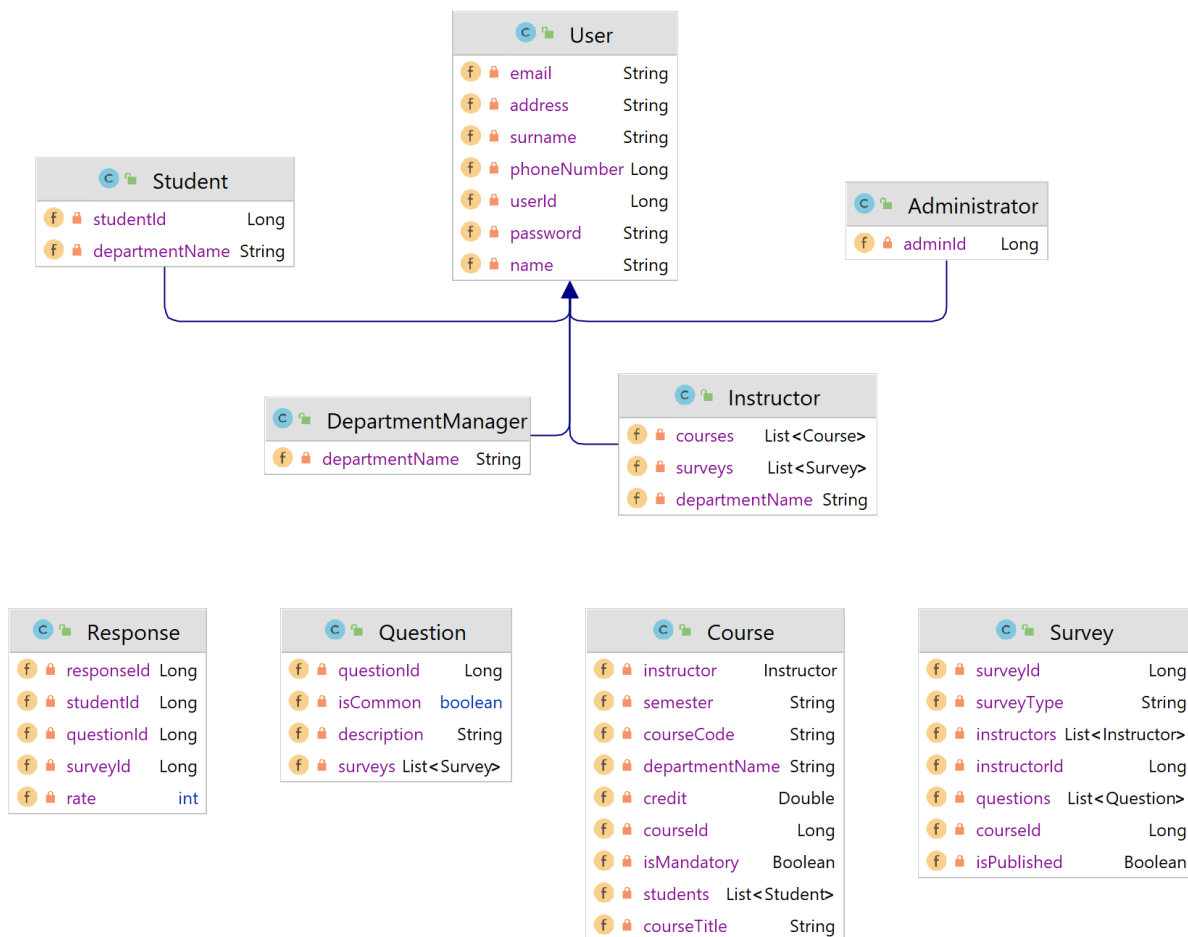   - Performance optimization

- It employs:
  - RESTful API design
  - Component-based frontend architecture
  - Layered backend architecture
  - Stable database schema

- It features:
  - User-friendly GUI
  - Comprehensive testing and quality assurance practices.

4. Design Details

4.1   Software Components

Entities

Special Request Types:

AssignCourseRequest

f	courseCode	String
f	semester	String
f	userId	Long

DeleteCourseRequest

f	semester	String
f	courseCode	String

LoginRequest

f	email	String
f	password	String
f	userType	String

CreateSurveyRequest

f	surveyType	String
f	questionDTOS	List<QuestionDTO>
f	instructorId	Long
f	courseId	Long

Business and Controllers

QuestionService

m	getCommonQuestions()	List<Question>
m	add(Question)	void
m	getQuestionById(Long)	Question

UserService

m	add(User)	User
m	getUserById(Long)	User
m	getAllUsers()	List<User>
m	getInstructorsByDepartmentManager(String)	List<Instructor>
m	login(String, String)	User
m	updateUser(User)	User
m	findByEmail(String)	User
m	delete(Long)	void

SurveyService

m	createSurvey(Survey)	Survey

CourseService

m	save(Course)	Course
m	findCourse(Course)	Course
m	getCourseById(Long)	Course
m	findByCourseCode(String)	Course
m	getCoursesByDepartmentName(String)	List<Course>
m	delete(Long)	void
m	getAllCourses()	List<Course>
m	add(Course)	Course
m	getCoursesByStudent(Student)	List<Course>

QuestionManager

f	questionDao	QuestionDao
m	add(Question)	void
m	getQuestionById(Long)	Question
m	getCommonQuestions()	List<Question>

UserManager

f	userDao	UserDao
m	getInstructorsByDepartmentManager(String)	List<Instructor>
m	updateUser(User)	User
m	add(User)	User
m	login(String, String)	User
m	getUserById(Long)	User
m	findByEmail(String)	User
m	getAllUsers()	List<User>
m	delete(Long)	void

SurveyManager

f	surveyDao	SurveyDao
m	createSurvey(Survey)	Survey

CourseManager

f	courseDao	CourseDao
m	findCourse(Course)	Course
m	getCoursesByDepartmentName(String)	List<Course>
m	getCoursesByStudent(Student)	List<Course>
m	getAllCourses()	List<Course>
m	findByCourseCode(String)	Course
m	save(Course)	Course
m	getCourseById(Long)	Course
m	add(Course)	Course
m	delete(Long)	void

QuestionController

f	questionService	QuestionService
m	initializeCommonQuestions()	void
m	getCommonQuestions()	ResponseEntity<?>

UserController

f	userService	UserService
m	updateUser(User)	ResponseEntity<?>
m	getAllUsers()	ResponseEntity<List<User>>
m	getUserById(Long)	ResponseEntity<User>
m	add(String, String)	ResponseEntity<?>
m	test()	String
m	login(LoginRequest)	ResponseEntity<?>
m	delete(Long)	ResponseEntity<?>
m	getInstructorsByDepartmentManager(Long)	ResponseEntity<?>

SurveyController

f	userService	UserService
f	courseService	CourseService
f	surveyService	SurveyService
f	questionService	QuestionService
m	createSurvey(CreateSurveyRequest)	ResponseEntity<?>

CourseController

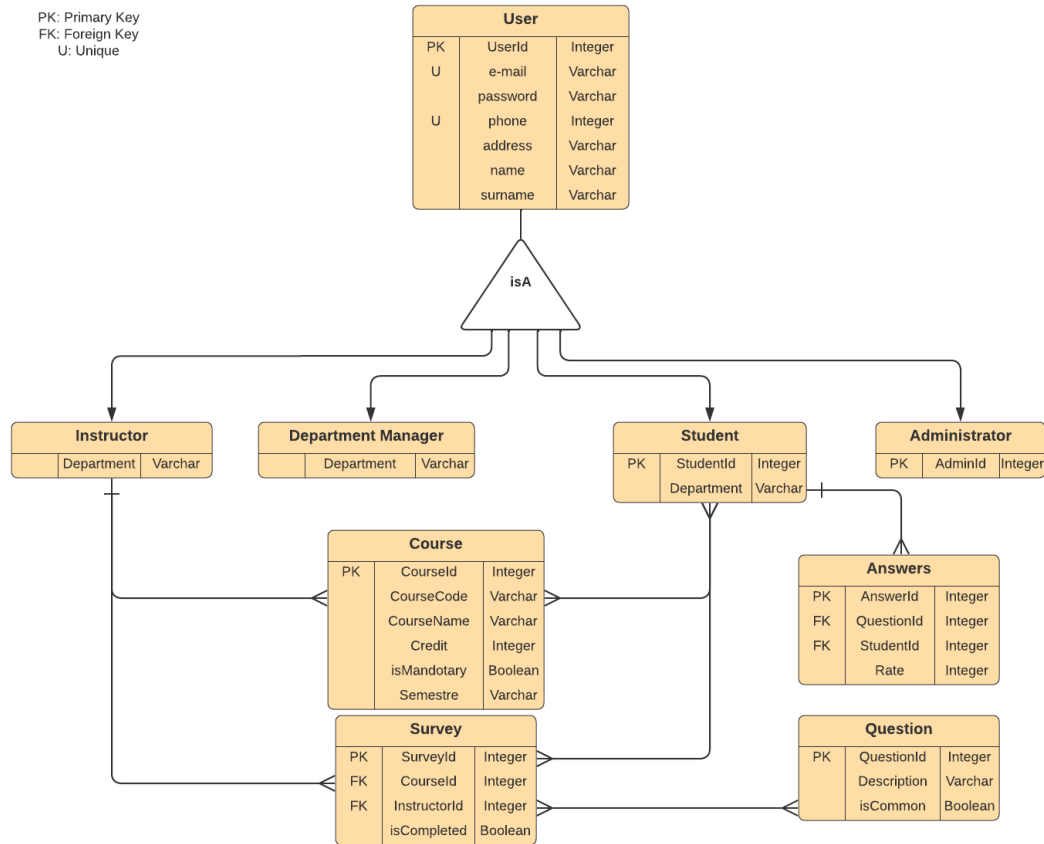|		
f	userService	UserService
f	courseService	CourseService
m	assignInstructorToCourse(AssignCourseRequest)	ResponseEntity<?>
m	deleteCourse(DeleteCourseRequest)	ResponseEntity<?>
m	add(String)	ResponseEntity<?>
m	getCoursesByStudent(Long)	ResponseEntity<?>
m	getAllCourses()	ResponseEntity<List<Course>>
m	assignStudentToCourse(AssignCourseRequest)	ResponseEntity<?>
m	getCoursesByDepartmentManager(Long)	ResponseEntity<?>

4.2   Software Behavior
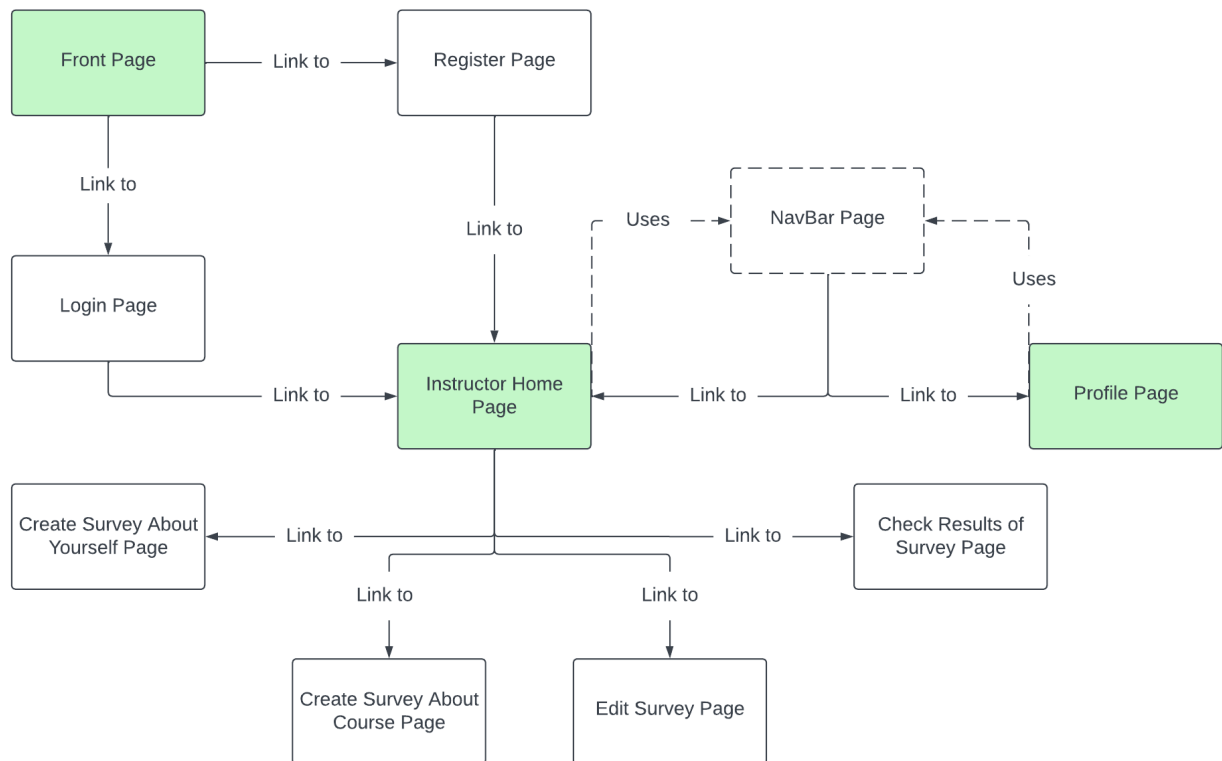
Use Case Diagram

4.3   Data Model (E-R Diagram)

E-R Diagram

4.4 User Interface Design

In our website there are three main pages which are named as front page, home page and profile page. With these three main pages, there are several subpages that can be reached from the main pages and also subpages. Lastly there is a navbar page which provides an interface with several same functions for Home Page and Profile Page.

To inform people about what the website does, when the user clicks the link of the website, a page with the brief information about what the system does with login and register buttons will be seen. After the user logins to the system, the Home Page with the NavBar Page will be shown first.

In the NavBar Page, 2 main functions can be done by the user. Firstly, users can reach two main functions (which is "go to home page" and "go to profile page") and the other one is for logout. NavBar Page is added on all pages below.

In the Profile Page, users can see all their information, edit their profile and request a new password.

In the Home Page, users can access the following pages: create surveys about themselves/courses, edit surveys and check results of surveys.

In the "Create Survey About Yourself" Page, users can add new questions to their survey, edit the existing questions and also delete questions.

In the "Create Survey About Course" Page, users can choose the course that the survey is about (from the dropdown menu), add new questions to their survey, edit the existing questions and also delete questions.

In the "Edit Survey" Page, users can choose the desired survey (from the dropdown menu), edit the existing questions and also delete questions.

In the "Check Results of Survey" Page, users can choose the desired survey (from the dropdown menu), request re-evaluation of the answers and also download evaluation results.
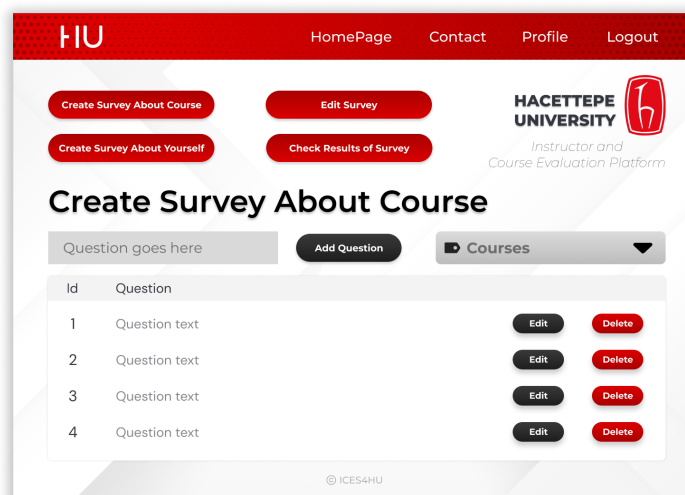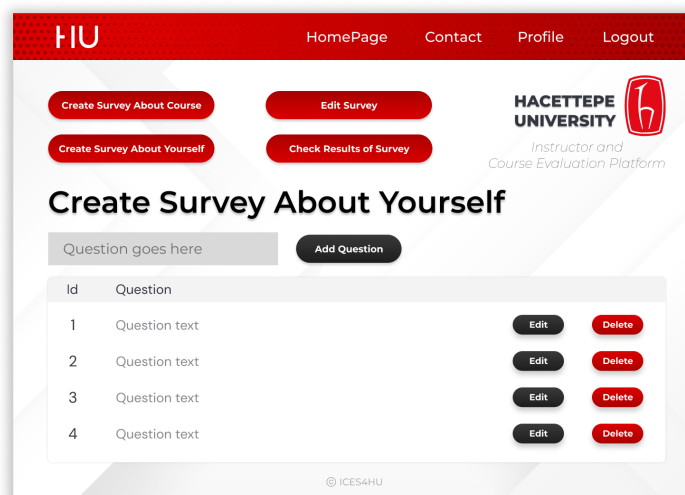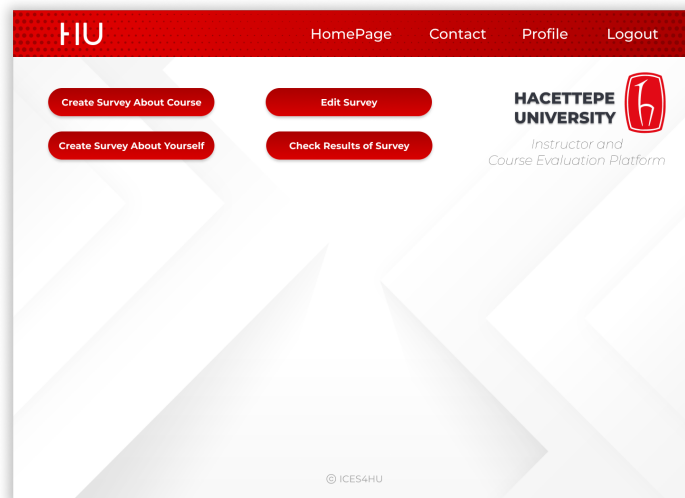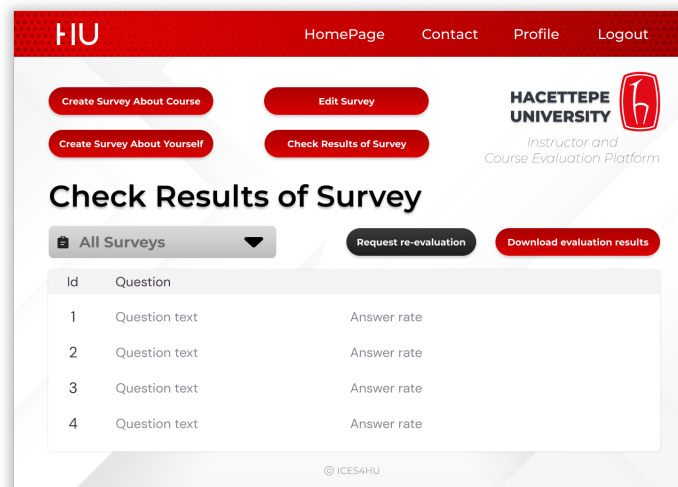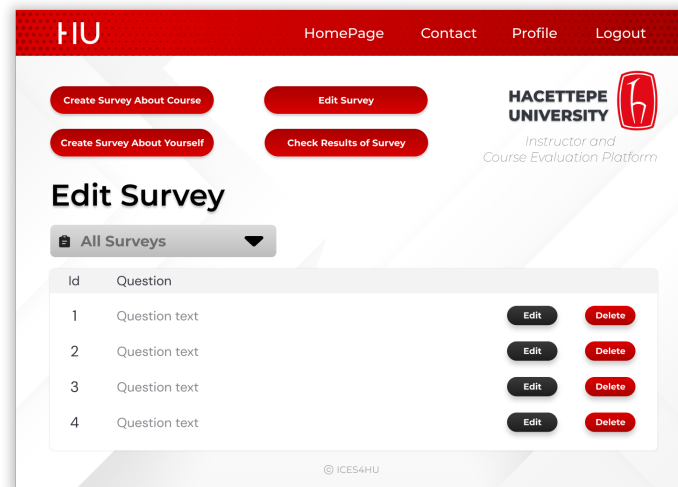
5.   Requirements Traceability

	Login	Logout	Admin adds user	Admin adds course	Delete Course	Assign Instructors to courses	Assign Students to Courses	Get Common Questions	Update User	Create Survey	Complete Survey
User Controller	X	X	X	X	X	X	X		X	X	X
UserService	X	X	X	X	X	X	X		X	X	X
CourseController				X	X	X	X			X	

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
CourseService				X	X	X	X			X	
QuestionController								X			
QuestionService								X		X	X
SurveyController										X	X
SurveyService										X	X
ResponseController											
ResponeService											
Course				X	X	X	X			X	X
User	X		X	X	X	X	X		X		
Admin			X								
Student			X				X				X
Instructor			X			X				X	
DepartmentManager			X								
UserDao	X		X	X	X	X	X		X		
Survey										X	
Question								X		X	
Response											
QuestionDao									X	X	
SurveyDao										X	
QuestionDTO									X		

6.   Annexes

Login Sequence Diagram
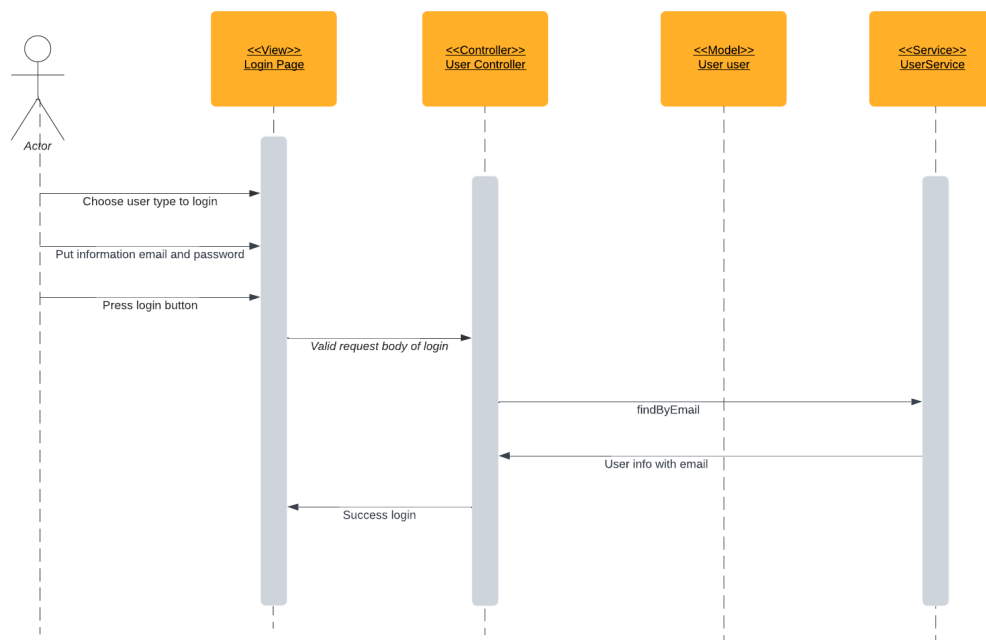
Login Statechart Diagram
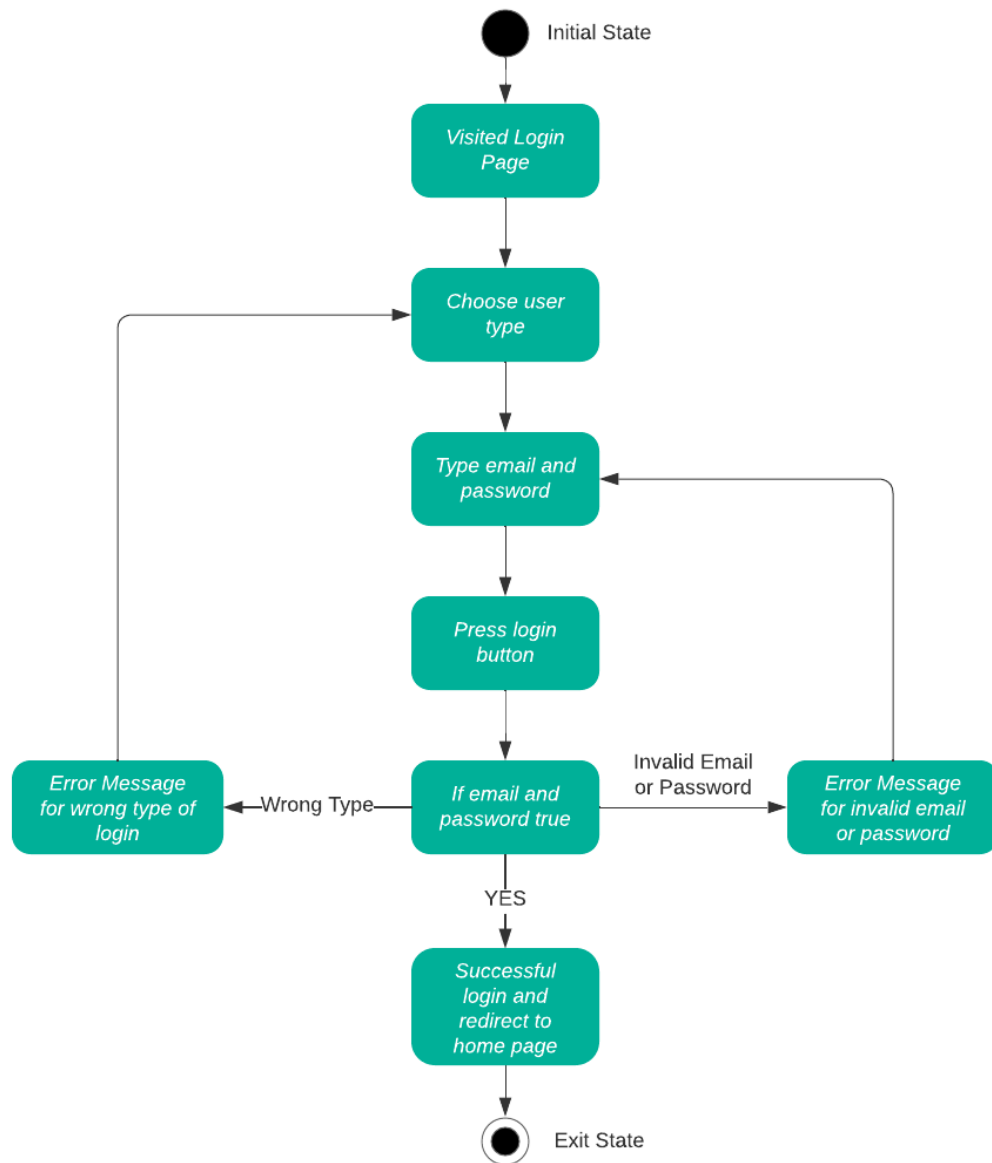
Edit Profile Sequence Diagram
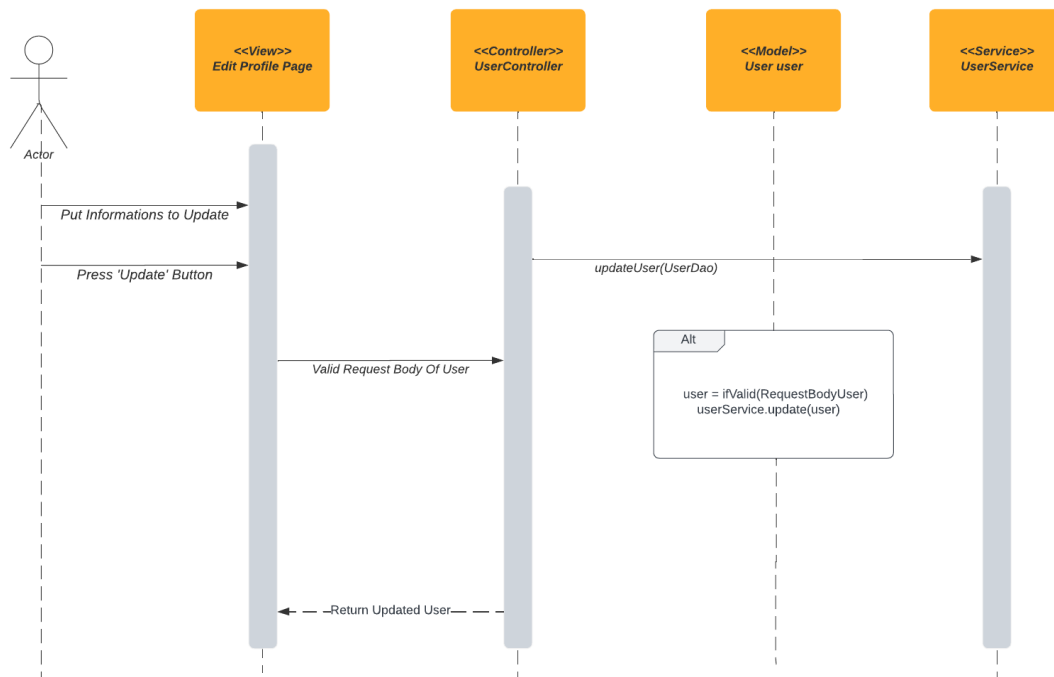
Edit Profile Statechart Diagram
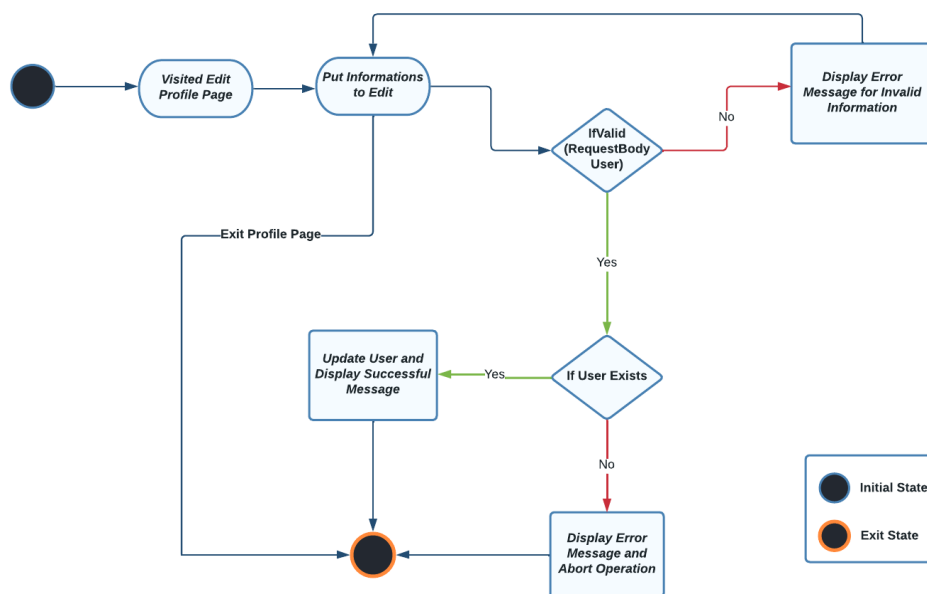
Check Survey Results Sequence Diagram
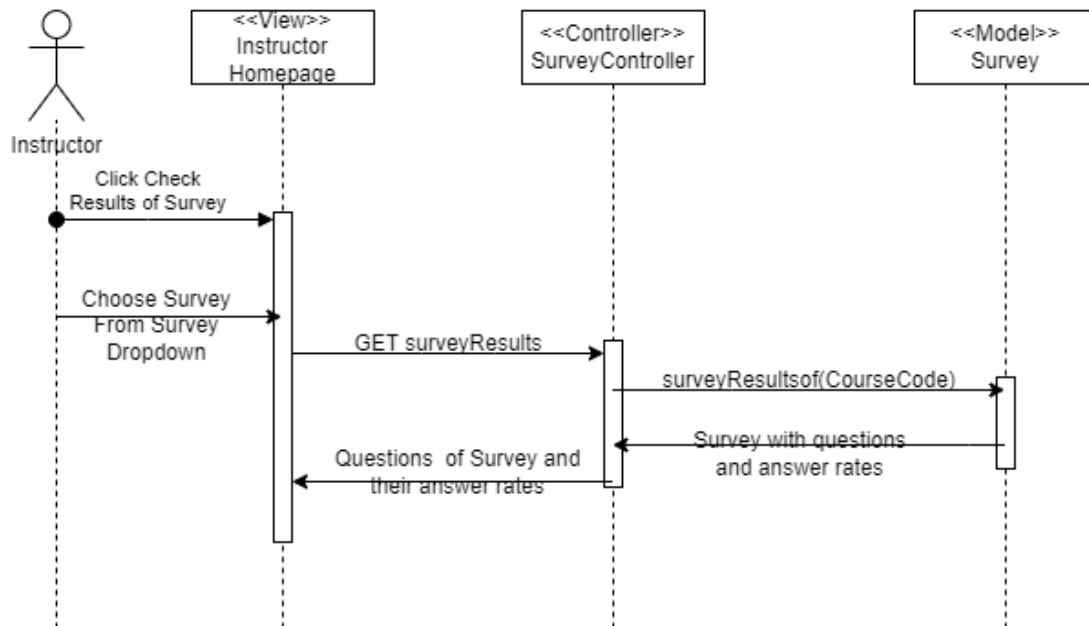
Check Survey Results Statechart Diagram
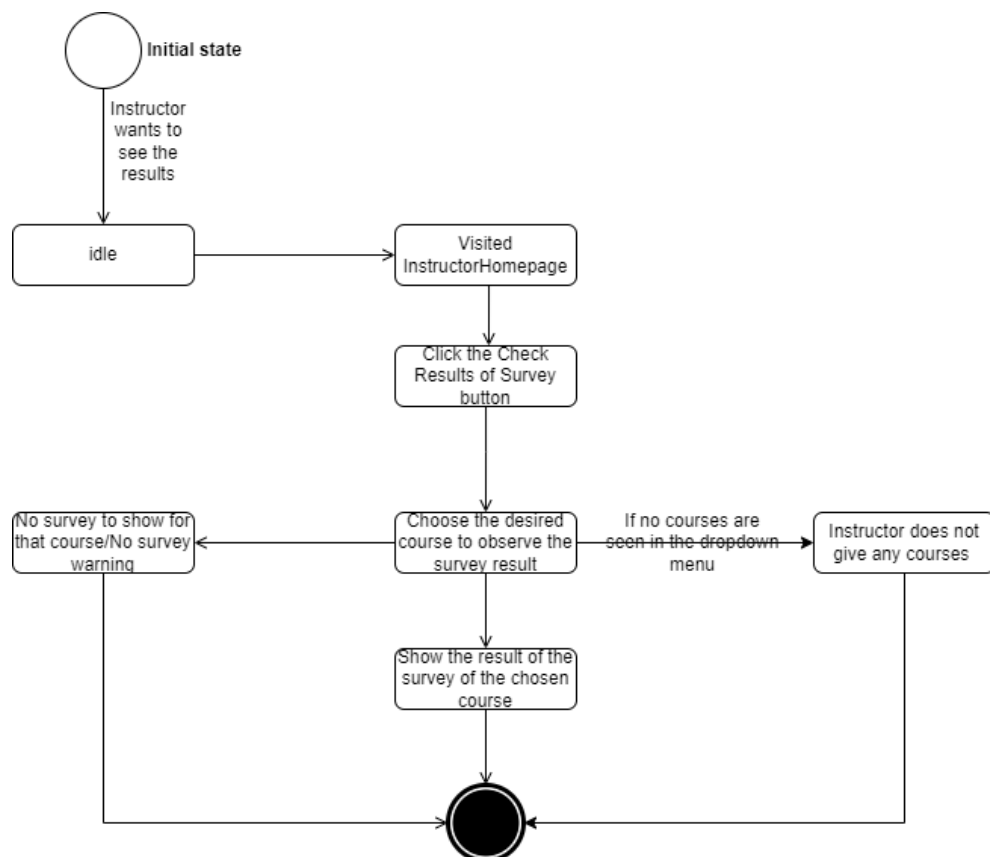
Delete User Sequence Diagram
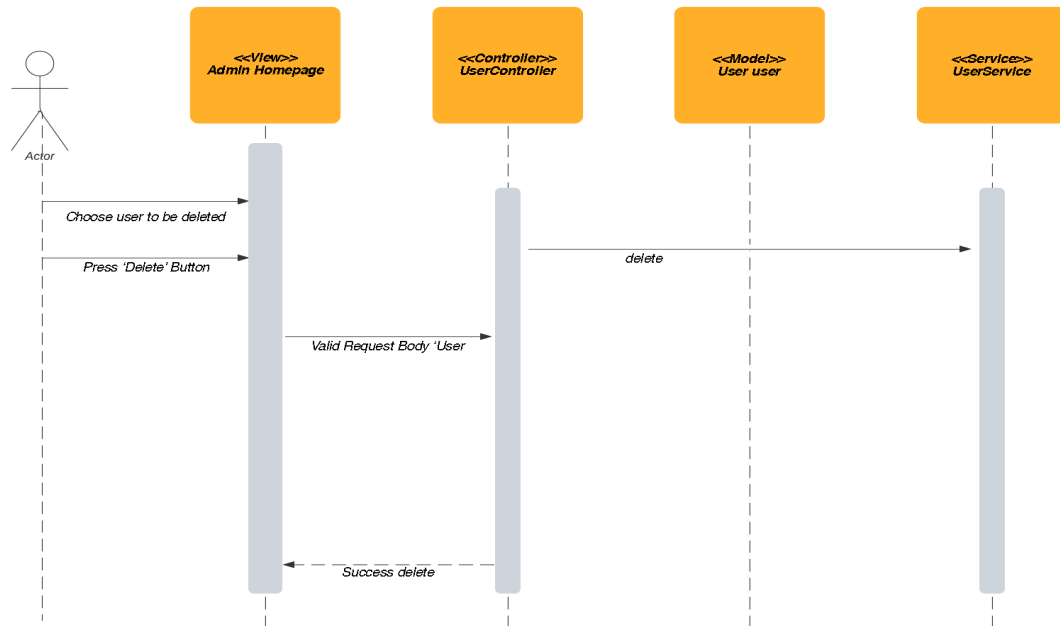
Delete User Statechart Diagram
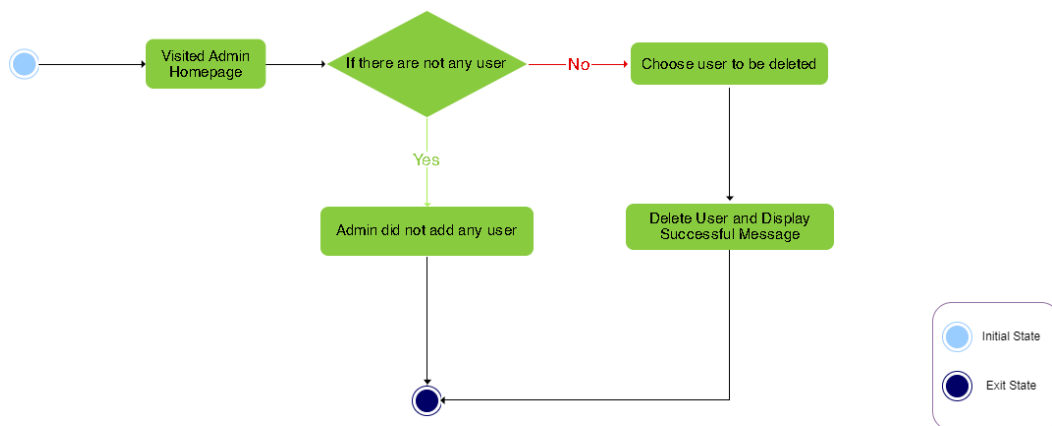
Add Course Sequence Diagram
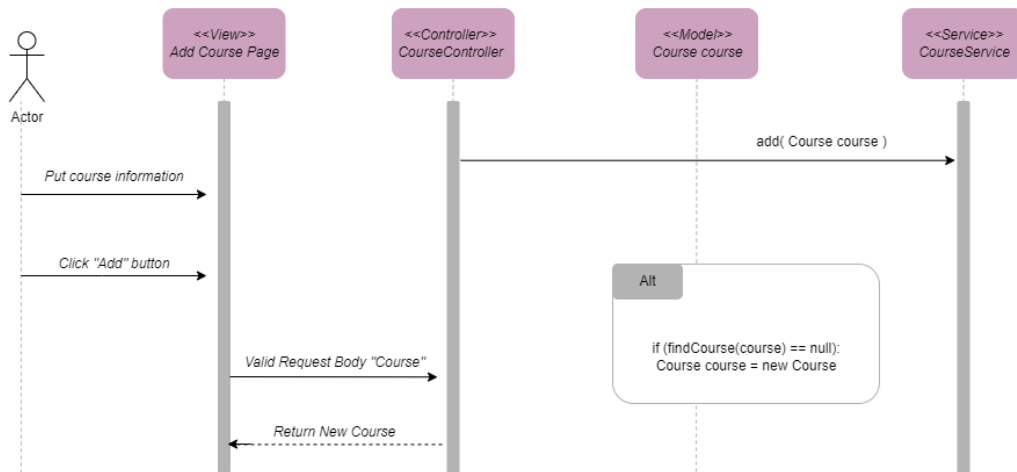
Add Course Statechart Diagram