ONLINE STUDENT COUNCIL ELECTION SYSTEM

Enes Demirel, Gökay Gülsoy, Berkan Gönülsever, Merve Nur Ozan, Halil İbrahim Buğday, Osman Altunay, Fatih Yelboğa

Software Design Descriptions

Table of Contents

1. Introduction

We want to develop a system that will automate the IZTECH Student Council elections. Therefore, we want this system to do all the operations done in a physical election. In addition to these, we want this system to be used for processes such as receiving candidacy applications before the election, announcements, and qualification control of representatives after the election.

1.1 Purpose

The purpose of the Software Design Explanation document is to explain the IZTECH Online Election System in detail. In this document, we will discuss the purpose of the system, its features, how it works, used, how it will behave and change under different conditions, its interfaces and by whom this system will be used and will be the beneficiary. The aim of this project is to develop a web application where students can vote, and upload their candidacy application documents, representatives can upload their qualification control documents, so they can do these activities online instead of physically.

1.2 Scope

A web-based system with two types of users will be introduced to use our online selection system. The first user type is students. They vote to elect department representatives. Students selected as department or faculty representatives have different authorities. For example, only department representatives can vote for faculty representative candidates. The second user type is university administration. They control candidacy application for department representative, election result, representatives' qualification control documents.

1.3 Context

The aim of the project is to automate the election process via web application. The application will be created by considering the benefit of students, university administration and stakeholders being in mind. This project will be implemented with JavaScript Frameworks. Database will be used for store students' information.

1.4 Summary

This Software Design Description document describes and provides a general overview on the structure, design, and implementation details for our project "Online Election System for IZTECH."

2. References

Group 8 Problem Analysis Report
Group 8 Software Project Management Plan
Group 8 Software Requirement Specification
CENG316-Lecture03-SDD

3. Glossary

University Administration: University's board of directors from the lowest to the highest. (Rectorate, dean's departments, etc.)

Candidates: Students who are candidates for department representation User: All people who can view the site, regardless of whether they are logged in or not.

Election: Student Council Election System: The implemented version of the online election system. A website.

Database: The environment in which the system will store all the data in it.

İYTE ÖBS: İzmir Yüksek Teknoloji Enstitüsü Öğrenci Bilgi Sistemi (İzmir Institute of Technology Student Information System)

Representatives: It is the student body whose department representation is approved after the election. Therefore, it is clear that the relationship of the representatives with ,the system is through students and candidates.

4. Viewpoints

4.1 Identified stakeholders and design concerns.

Our project stakeholders are Tanıl Levent (Faculty Secretary) and İlgim Efetürk (President of the Student Council). The online election system should be designed to be user-friendly, reliable and also easy to update & maintain.

4.2 Context Viewpoint
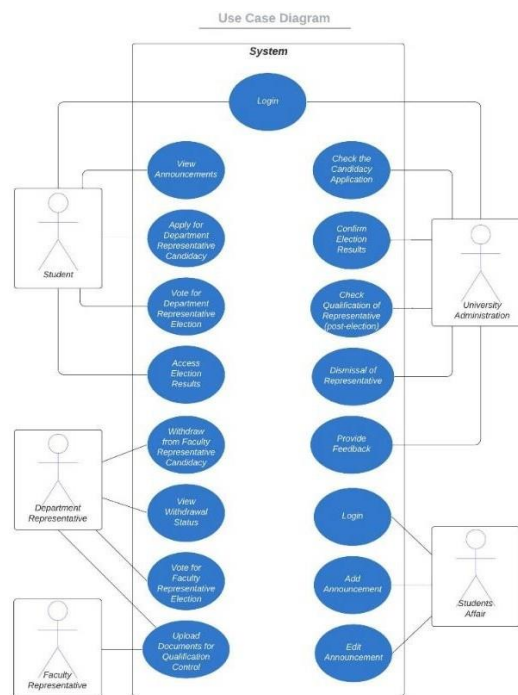
4.3 Informational Viewpoint

Election System: System provides an online voting platform for IZTECH students.

Students: They will assume the role of user, which is the most important part of the system. It is the user community that will interact with the system interface and benefit from the system.

University Administration: It is the university's board of directors from the lowest to the highest. (Rectorate, dean's departments, student affairs department, etc.) It is the important factor for the system, event controls, instructions and procedures are provided from this domain.

Candidates: They are defined as a subset of the student's domain. Candidates for department representatives are indirectly connected to this system on students.

Representatives: It is the student body whose department representation is approved after the election. Therefore, it is clear that the relationship of the representatives with the system is through students and candidates.

The ER diagram is given below.
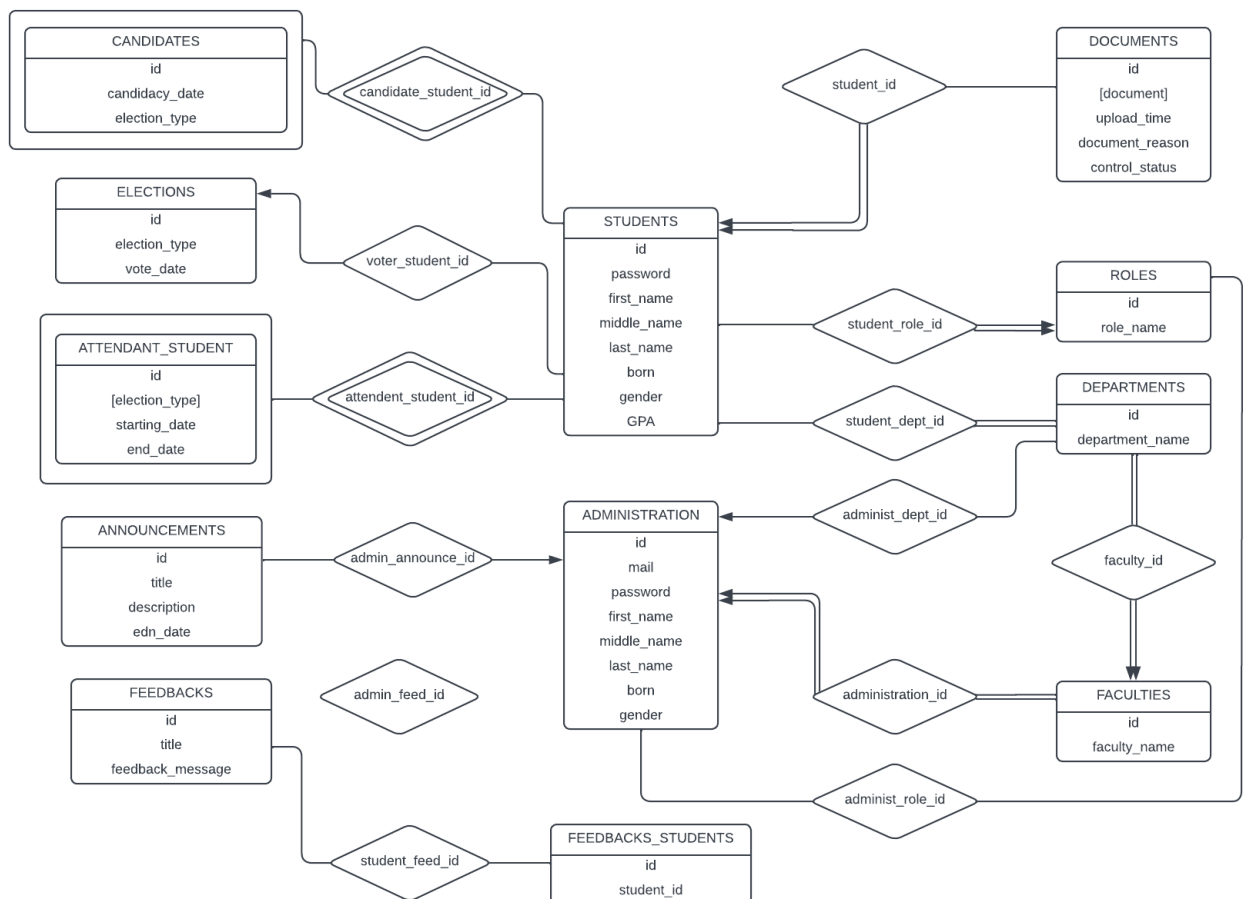
4.4 Interface Viewpoints

Figure-1) Login Portal for Student and University Staff Entry
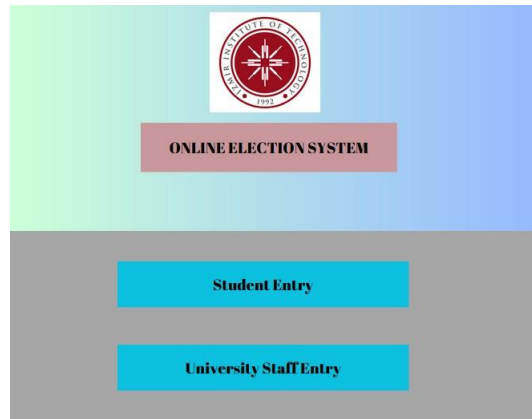
Figure-2) Login Interface for Students
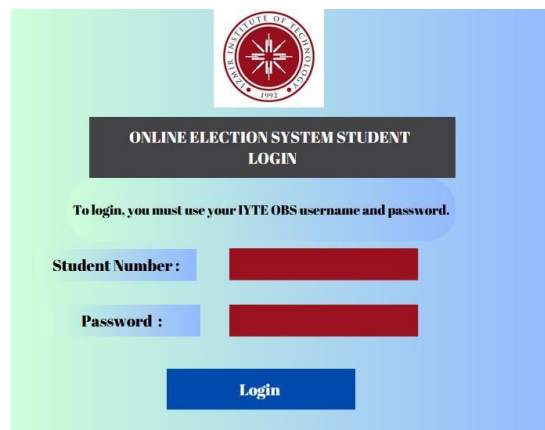
Figure-3) Login Interface for University Staff
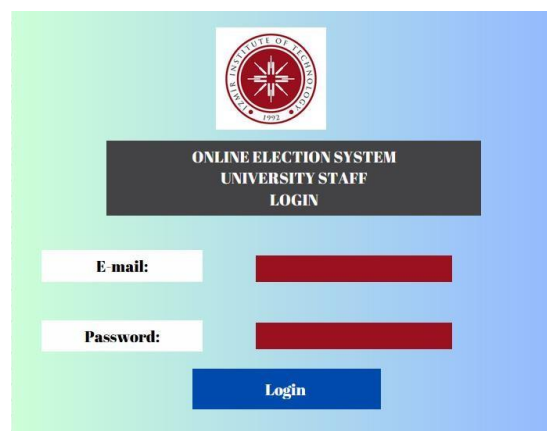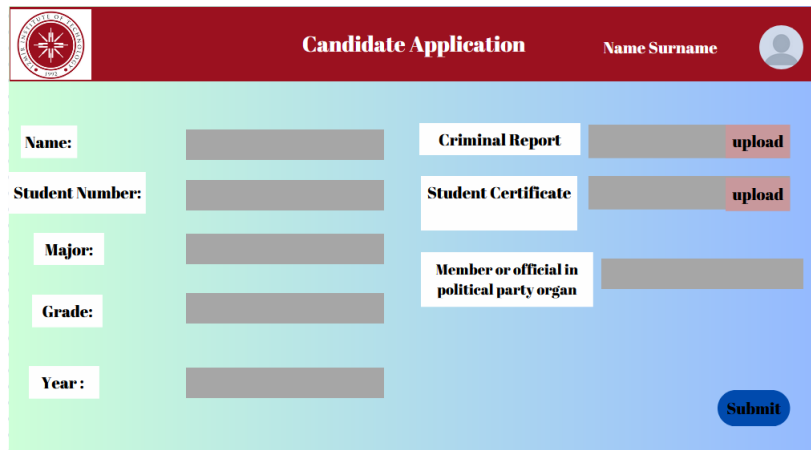
Figure-4) Student's Home Page Interface

Figure-5) Candidate application interface for department representative

Figure-6) Students' Voting Interface for Department Representative
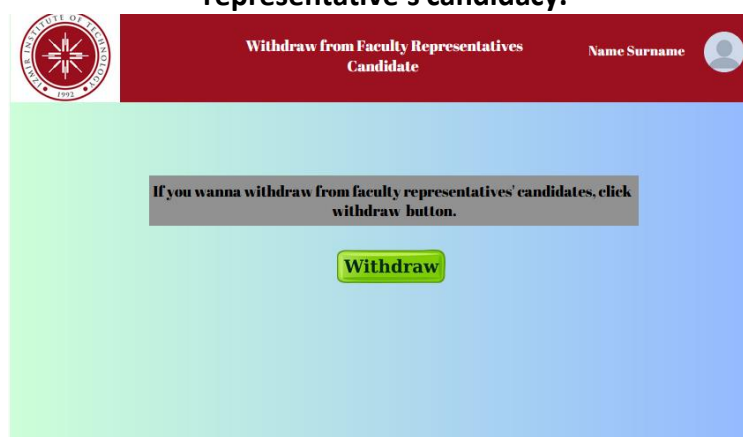
Figure-7) Election Result Interface
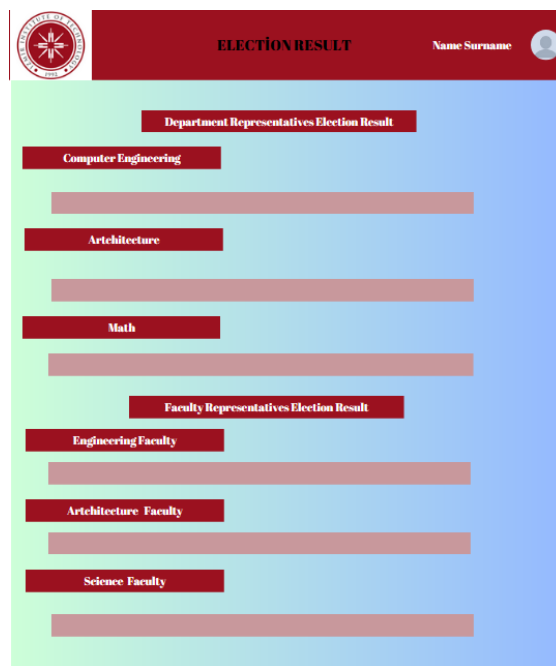
Figure-8) Department Representative Home Page Interface
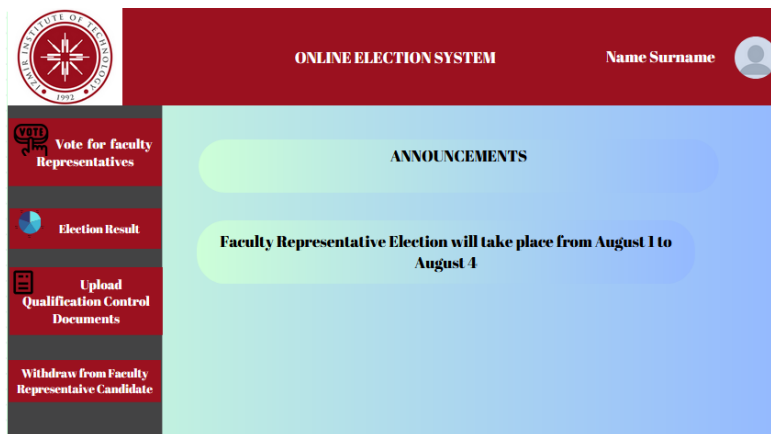
Figure-9) Interface for department representative can withdraw from faculty representative's candidacy.

Figure-10) View withdrawal Status

Withdraw from Faculty Representatives Candidate

Name Surname

Your request to withdraw from the faculty representative candidacy has been forwarded.

Figure-11) Department Representatives' Voting Interface for Faculty Representative

Figure-12) Interface for uploading qualification control document of representatives.

Figure-13) University Staff Home Page Interface

Figure-14) University Staffs Check Students' Candidate Application for Department Representatives Interface

Figure-15) University Staffs Check Qualification Control Documents of Representatives Interface

Figure-16) University Staffs Confirms or not Election Results Interface

Figure-17) University Staff's Editing Announcements Interface

Figure -18) University Staff's Adding Announcements Interface

COMPONENT DIAGRAM

-COMPONENT DIAGRAM-

<<subsystem>> **ONLINE ELECTION SYSTEM FRONT-END**

```
<<component>>        IUser        <<component>>
   User                              Role
```

```
<<component>>  <<realization>>  <<component>>   <<component>>   <<component>>   <<component>>
  Document                         Student        StudentAffair    Department      Administration
```

```
IStudent              IStudentAffair    <<realization>>        IAdministration    <<realization>>
```

```
<<component>>              <<component>>         <<component>>    <<component>>    <<component>>
DepartmentRepresentative   FacultyRepresentative   Announcement      Faculty          Feedback
```

IDepartmentRepresentative   IFacultyRepresentative

IDatabaseController

<<subsystem>> **ONLINE ELECTION SYSTEM BACK-END**

```
<<component>>
DatabaseController
```

IDatabaseController

Component Box

```
<<component>>  <<component>>  <<component>>  <<component>>  <<component>>   <<component>>  <<component>>  <<component>>  <<component>>      <<component>>
DocumentController StudentController DepartmentController FacultyController AdministrationController FeedbackController RoleController ElectionController AnnouncementController AuthController
```

IDocumentController IStudentController IDepatmentController IFacultyController IAdministrationController IFeedbackController IRoleController IElectionController IAnnouncementController IAuthController
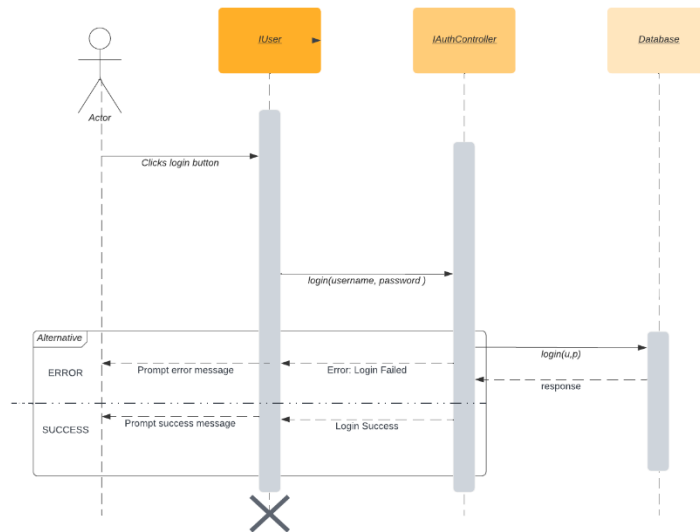
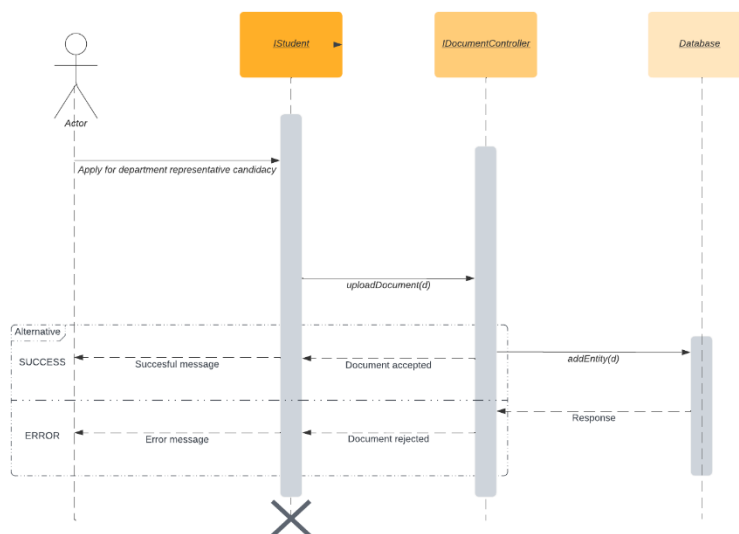# 4.5 Logical Viewpoint

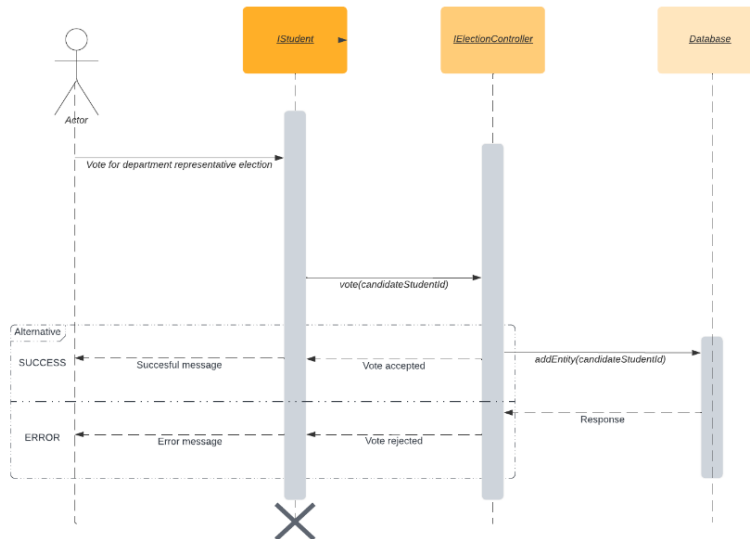**The UML Class Diagram is given below.**
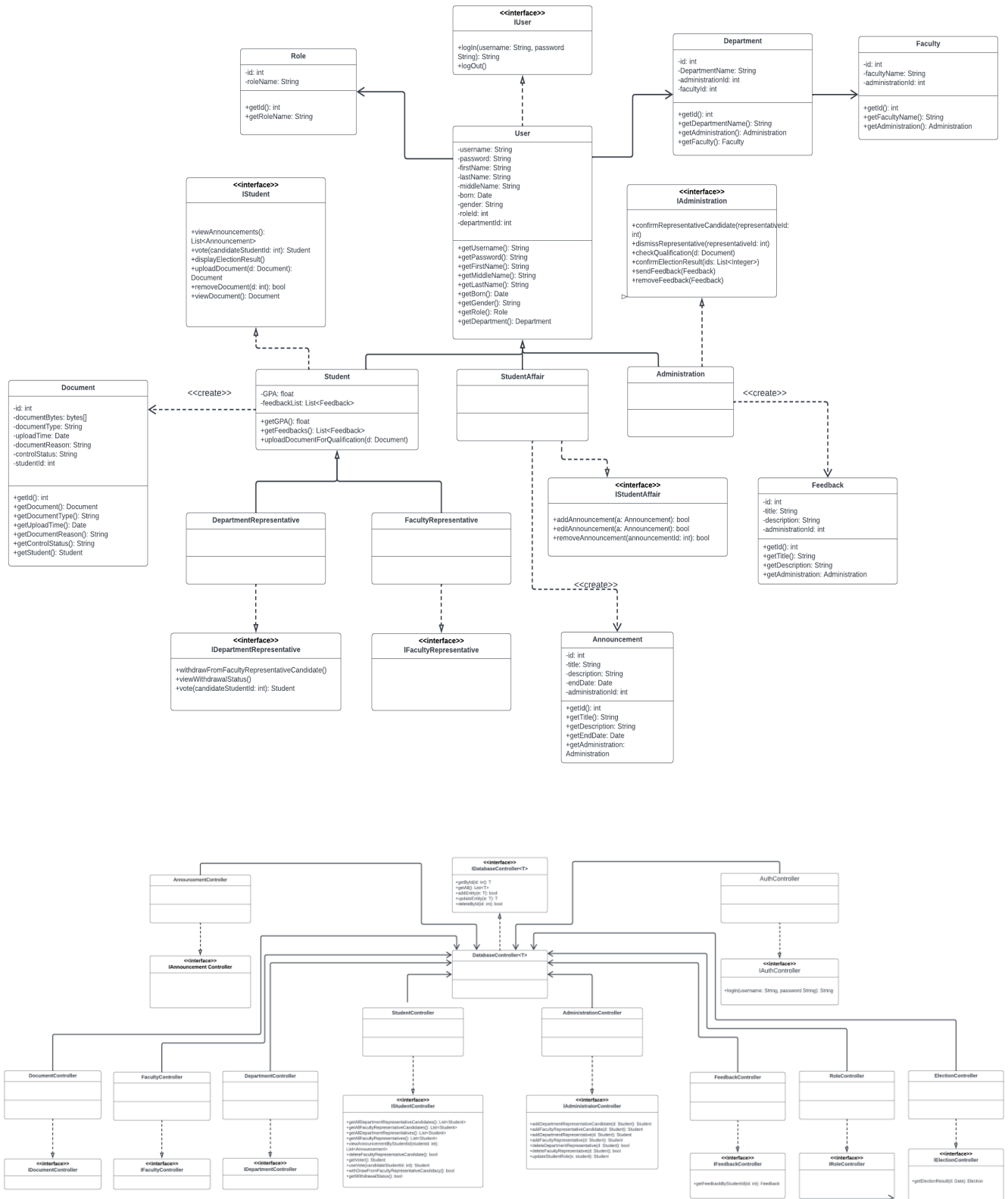
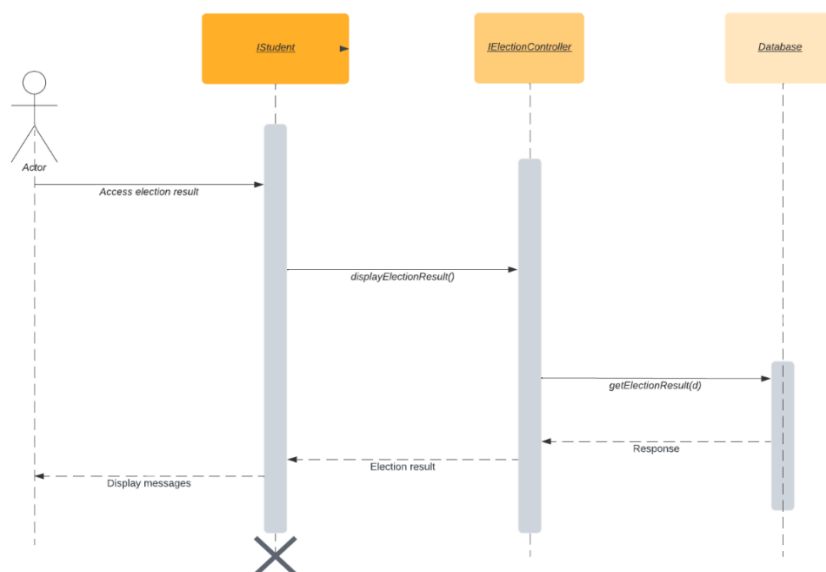# 4.6 Interaction Viewpoint

4.6.1 Login Sequence Diagram
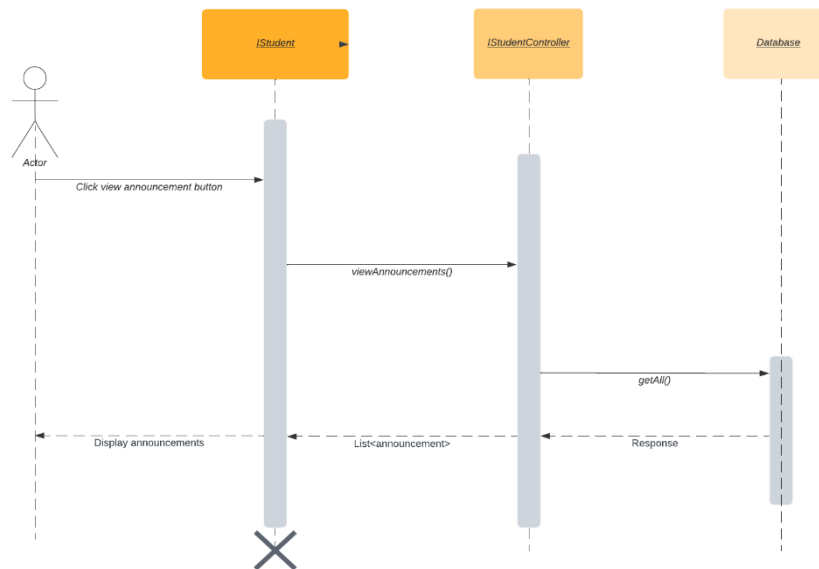

4.6.2 Apply Candidacy Sequence Diagram

4.6.3 Vote for Department Representative Election Sequence Diagram

4.6.4 Access Election Result Sequence Diagram
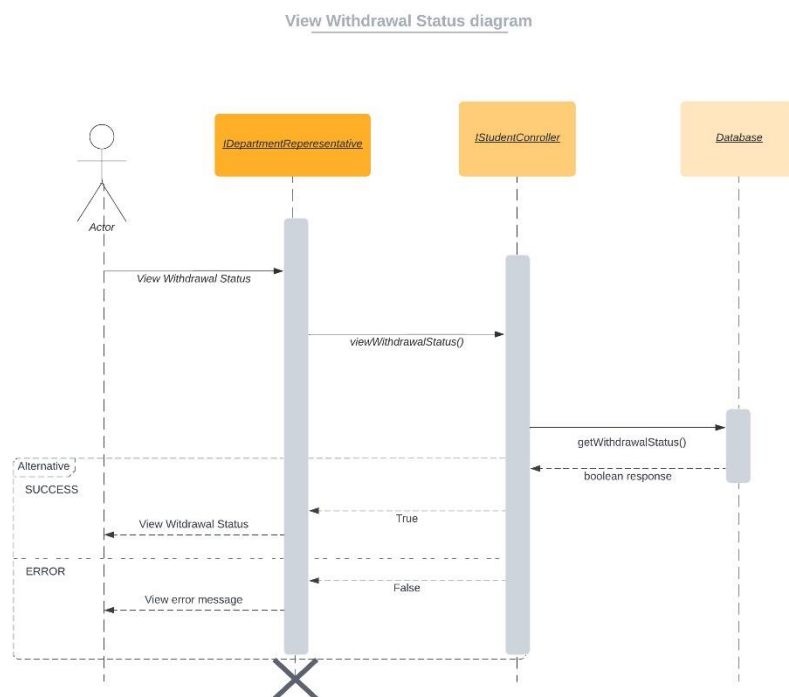
4.6.5 View Announcement Sequence Diagram

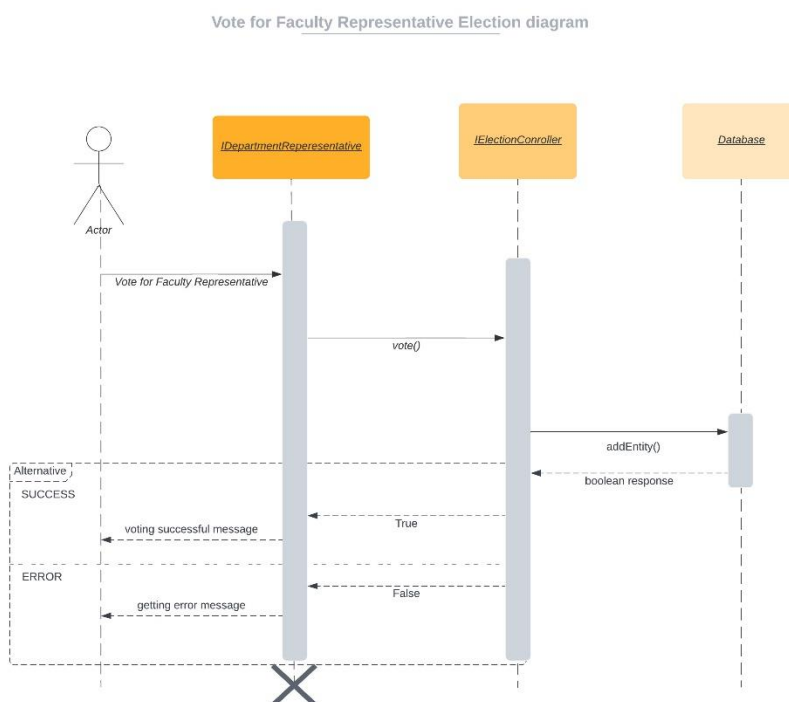## 4.6.6 Department Representatives withdraw from Faculty Representative Candidacy

4.6.7 View Withdrawal Status Sequence Diagram

View Withdrawal Status diagram

4.6.8 Vote for Faculty Representative Election Sequence Diagram
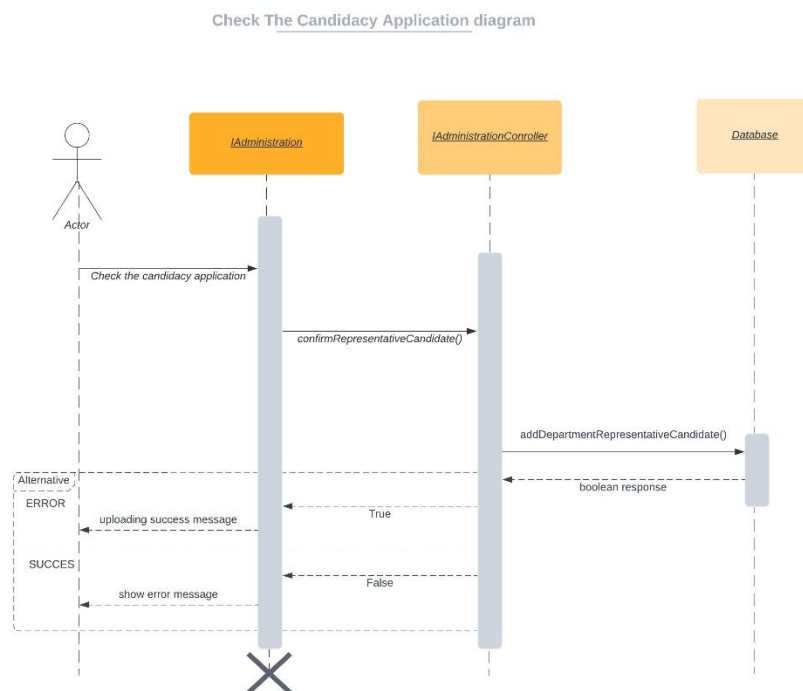
Vote for Faculty Representative Election diagram

4.6.9 Representatives Upload Document for Qualification Control Sequence Diagram

Upload Document For Qualification diagram
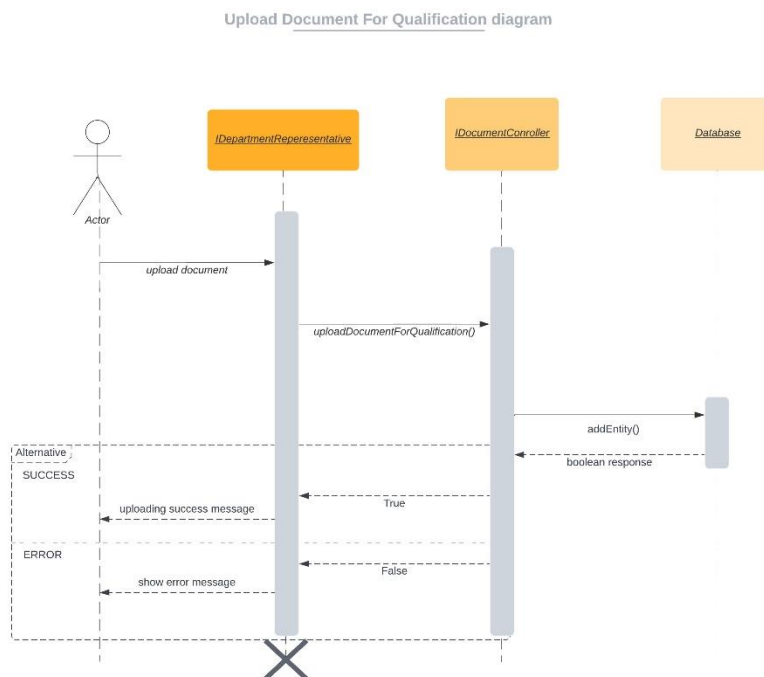
4.6.10 Check Candidacy Application for department representative Sequence Diagram

Check The Candidacy Application diagram

4.6.11 Confirm Election Results Sequence Diagram

Confirm Election Results diagram
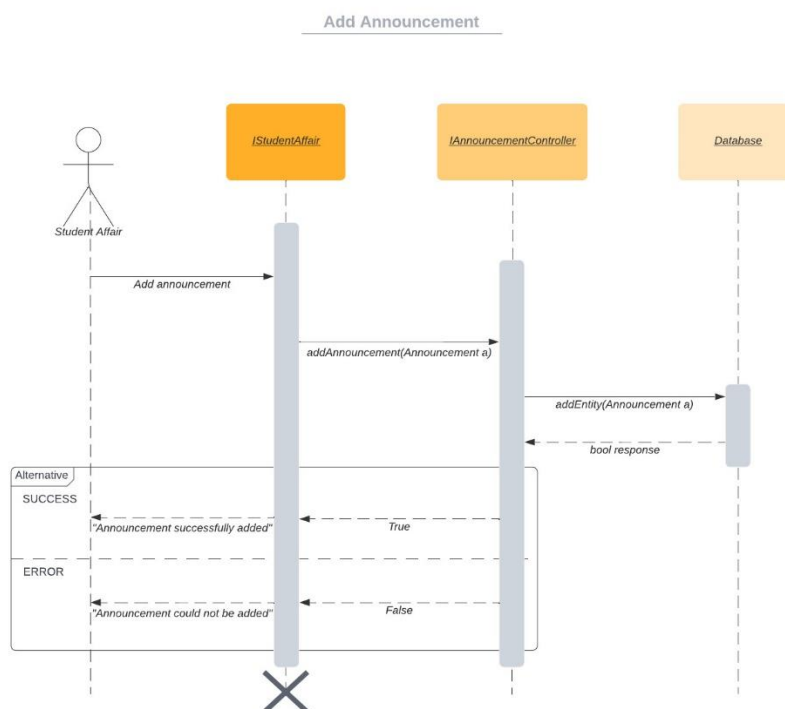
4.6.12 Check Qualification Control Documents of Representatives
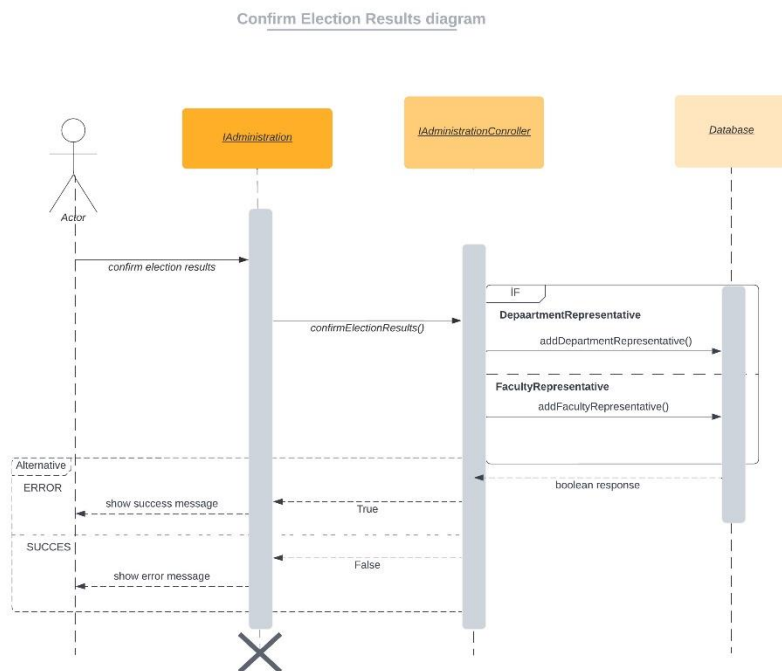
Check Qualification of Representative
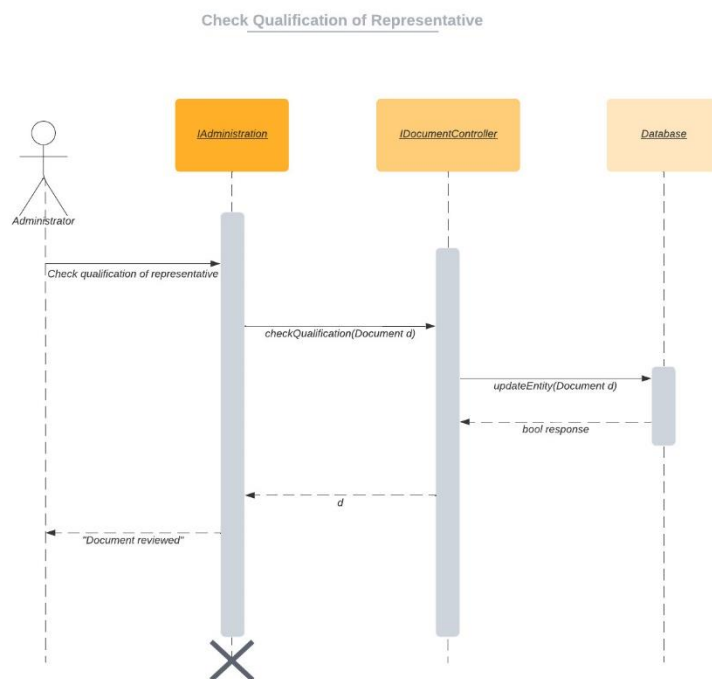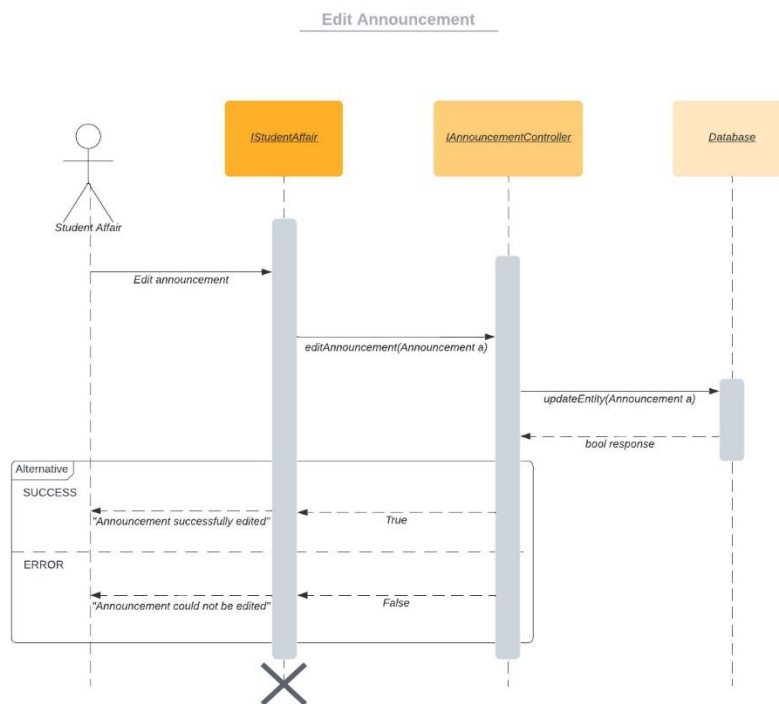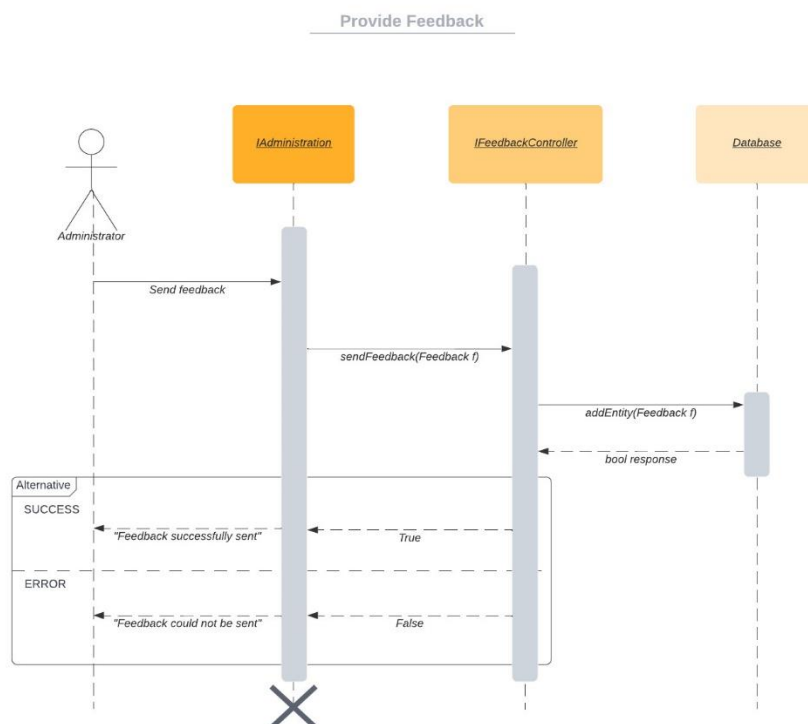
4.6.13 Dismissal of Representative
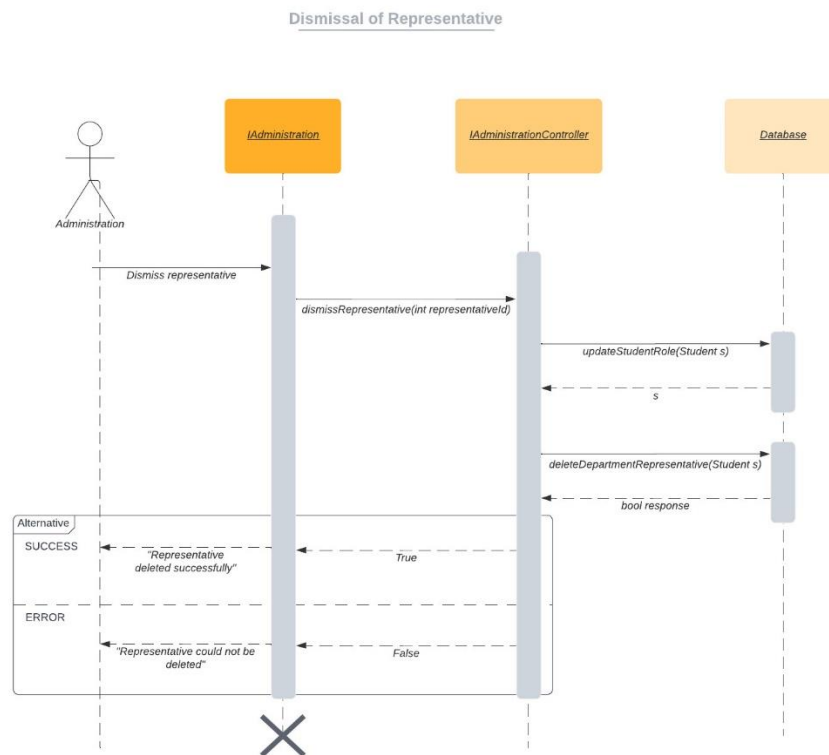
4.6.14 Add Announcement

4.6.15 Edit Announcement

Edit Announcement

IStudentAffair — IAnnouncementController — Database

Student Affair

Edit announcement

editAnnouncement(Announcement a)

updateEntity(Announcement a)

bool response

Alternative

SUCCESS

"Announcement successfully edited"   True

ERROR

"Announcement could not be edited"   False

4.6.16 Provide Feedback Sequence Diagram

Provide Feedback

IAdministration — IFeedbackController — Database

Administrator

Send feedback

sendFeedback(Feedback f)

addEntity(Feedback f)

bool response

Alternative

SUCCESS

"Feedback successfully sent"   True

ERROR

"Feedback could not be sent"   False

4.7 Design Rationale

All steps we've taken towards the development of this program were intended to contribute to the development with our design concerns, reliability, and simplicity, in mind.