

ONLINE STUDENT COUNCIL ELECTION SYSTEM

Group 8

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

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Date: 6/12/2022

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, and Abbreviations
- 1.4 References
- 1.5 Overview

2. The Overall Description

- 2.1 Product Perspective
 - 2.1.1 System Interfaces*
 - 2.1.2 User Interfaces*
 - 2.1.3 Hardware Interfaces*
 - 2.1.4 Software Interfaces*
 - 2.1.5 Communication Interfaces*
 - 2.1.6 Operations*
 - 2.1.7 Site Adaption Requirement*
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions and Dependencies
- 2.6 Apportioning of Requirements

3. Specific Requirements

- 3.1 Use Cases
 - 3.1.1** Use Cases for Student
 - 3.1.1.1** Use case 1 for Student
 - 3.1.1.1.1** Use case template
 - 3.1.1.1.2** User Interface
 - 3.1.1.1.3** Performance /Software quality requirements

3.1.1.2 Use case 2 for Student

3.1.1.2.1 Use case template

3.1.1.2.2 User Interface

3.1.1.2.3 Performance /Software quality requirements

3.1.1.3 Use case 3 for Student

3.1.1.3.1 Use case template

3.1.1.3.2 User Interface

3.1.1.3.3 Performance /Software quality requirements

3.1.1.4 Use case 4 for Student

3.1.1.4.1 Use case template

3.1.1.4.2 User Interface

3.1.1.4.3 Performance /Software quality requirements

3.1.1.5 Use case 5 for Student

3.1.1.5.1 Use case template

3.1.1.5.2 User Interface

3.1.1.5.3 Performance /Software quality requirements

3.1.2 Use Cases for University Administration

3.1.2.1 Use case 1 for University Administration

3.1.2.1.1 Use case template

3.1.2.1.2 User Interface

3.1.2.1.3 Performance /Software quality requirements

3.1.2.2 Use case 2 for University Administration

3.1.2.2.1 Use case template

3.1.2.2.2 User Interface

3.1.2.2.3 Performance /Software quality requirements

3.1.2.3 Use case 3 for University Administration

3.1.2.3.1 Use case template

3.1.2.3.2 User Interface

3.1.2.3.3 Performance /Software quality requirements

3.1.2.4 Use case 4 for University Administration

3.1.2.4.1 Use case template

3.1.2.4.2 User Interface

3.1.2.4.3 Performance /Software quality requirements

3.1.2.5 Use case 5 for University Administration

3.1.2.5.1 Use case template

3.1.2.5.2 User Interface

3.1.2.5.3 Performance /Software quality requirements

3.1.2.6 Use case 6 for University Administration

3.1.2.6.1 Use case template

3.1.2.6.2 User Interface

3.1.2.6.3 Performance /Software quality requirements

3.1.2.7 Use case 7 for University Administration

3.1.2.7.1 Use case template

3.1.2.7.2 User Interface

3.1.2.7.3 Performance /Software quality requirements

3.1.2.8 Use case 8 for University Administration

3.1.2.8.1 Use case template

3.1.2.8.2 User Interface

3.1.2.8.3 Performance /Software quality requirements

3.1.3 Use Cases for Department Representative

3.1.3.1 Use case 1 for Department Representative

3.1.3.1.1 User Interface

3.1.3.1.2 Performance /Software quality requirements

3.1.3.2 Use case 2 for Department Representative

3.1.3.2.1 User Interface

3.1.3.2.2 Performance /Software quality requirements

3.1.3.3 Use case 3 for Department Representative

3.1.3.3.1 User Interface

3.1.3.3.2 Performance /Software quality requirements

3.1.3.4 Use case 4 for Department Representative

3.1.3.4.1 User Interface

3.1.3.4.2 Performance /Software quality requirements

3.1.4 Use Cases for Faculty Representative

3.1.4.1 Use case 1 for Faculty Representative

3.1.4.1.1 User Interface

3.1.4.1.2 Performance /Software quality requirements

3.1.5 Use Cases for Students Affair

3.1.5.1 Use case 1 for Students Affair

3.1.5.1.1 Use case template

3.1.5.1.2 User Interface

3.1.5.1.3 Performance /Software quality requirements

3.1.5.2 Use case 2 for Students Affair

3.1.5.2.1 Use case template

3.1.5.2.2 User Interface

3.1.5.2.3 Performance /Software quality requirements

3.1.5.3 Use case 2 for Students Affair

3.1.5.3.1 Use case template

3.1.5.3.2 User Interface

3.1.5.3.3 Performance /Software quality requirements

3.1.6 Use Cases for All Users

3.1.6.1 Use case 1 for All Users

3.1.6.1.1 Use case template

3.1.6.1.2 User Interface

3.1.6.1.3 Performance /Software quality requirements

3.2 External Interfaces

3.3 Generic Performance/Software Requirements

3.4 Entity-Relationship Diagram

3.5 Design Constraints

3.6 Other Requirements

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

Purpose of this software specification requirements document is to provide a complete description of the features that are planned to be in the system. Additionally, it describes how the system operates and how users interact with the system. It also specifies the external systems that are website depends for some functionalities **such as logging in users**.

Intended audience for this software specification requirements document is the development team of the Online Student Council Election System.

1.2 Scope

- A website that allows users to login with authentication, performs different actions for

each user type, allows students to vote and to submit documents, allows the university administration to check documents, confirm the results of the election.

- All users will log-in to the system with their ÖBS information.
- Students will be able to upload the required documents to the system for candidacy application.
- University administration will be able to view the documents uploaded by the candidates to the system and evaluate their applications.
- Students will be able to see and vote only the candidates of their departments in the system.
- Department representatives will be able to see and vote only the candidates of their faculties in the system.
- In order to prevent the same user from voting more than once, the system will close the voting screen to the user's access after the user has voted.
- The system will not deal with the board elections.
- The data of the users will be taken from ÖBS.
- The main goal of the system is to save time and resources for both students and the university by speeding up and making the election process easier. For example, in physical elections, vote counting is carried out by university staff, which takes time and uses many resources of the university. However, in the online election process, the vote count can be viewed instantly at the end of the election.

1.3 Definitions, Acronyms and Abbreviations

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)

SRS: System Requirements Specification

1.4 References

IZTECH Student Council Guideline

Lecture03-IEEE SRS_V1.pdf

CENG 323 – SRS Draft Template.pdf

1.5 Overview

The rest of the document contains overall description part and specific requirement's part.

Overall description part consist of functions of the product, constrains of the system, user characteristics, assumptions, and dependencies. It also contains use case diagram.

Specific requirements part consists of use cases, external interfaces, generic performance/software requirements, logical database requirements and design constraints and are explained.

2. Overall Description

This section gives background information about specific requirements of the system to be developed.

2.1 Product Perspective

All student data to be used in the system will be obtained from İYTE ÖBS. Except for the acquisition of student data, the system will be a completely independent and self-contained.



2.1.1 System Interfaces

System will not deal with storing login information of users. It will depend on İYTE ÖBS interface.

2.1.2 User Interfaces

This system is developed for students and university staff who is responsible for election process. Product will be available on a website and all users will access the system through the web pages according to the functionality. For example, there will be a voting page where users can vote.

2.1.3 Hardware Interfaces

Since system will be implemented as a web application, it can run on any hardware that can run web browsers.

2.1.4 Software Interfaces

Database management system is required for our system because all data about system except login information of users, for example the documents that uploaded by students for candidacy application, must be stored in database.

2.1.5 Communication Interfaces

In communication between student who applies for candidacy and university administration, there will be an interface where university administration will inform the student about their application result.

2.1.6 Operations

This part is explained in the [2.1.2 User Interfaces](#) section.

2.1.7 Site Adaptation Requirements

System does not need any adaptation to a particular platform.

2.2 Product Functions

Log In: Active students and university staff who are authorized to use the system can log into the system by filling out the fields with the information they use in the İYTE ÖBS.

Log Out: Users can log out from the system by clicking the log out button.

View Announcements: After logging in, students can see the announcements by selecting the announcements from menu.

Apply for Department Representative Candidacy: Students who want to apply for candidacy can make the necessary application for it.

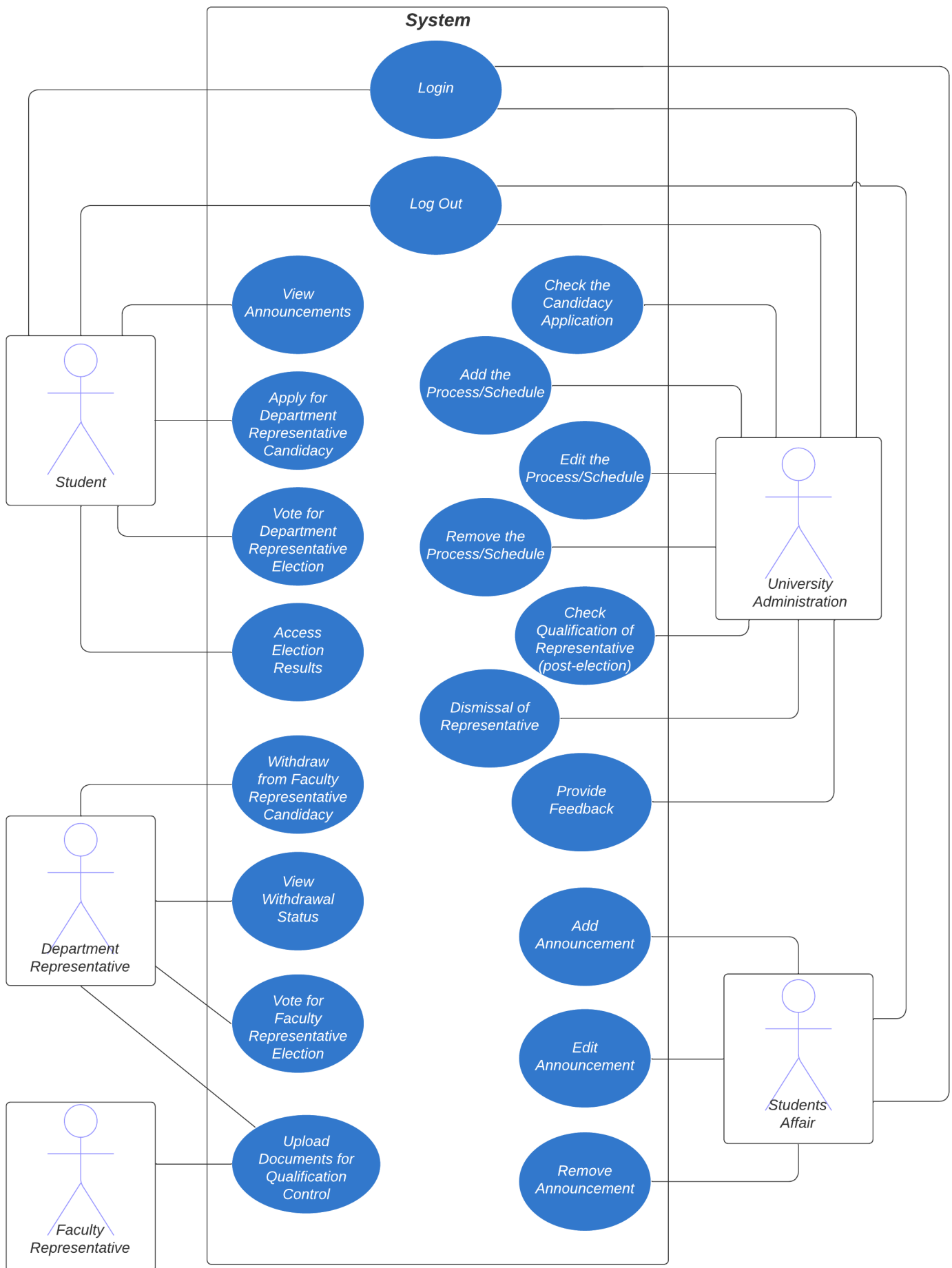
View Application: The relevant university administration can see the documents sent by the students who want to be candidates.

Evaluate Application: The relevant university administration can evaluate application sent by students who want to be candidates.

Vote: All students can vote for the election of department representatives, and all department representatives can vote for the election of faculty representatives. In addition, the system will prevent voting more than once by closing the voting screen for the user after he/she has voted.

Withdraw from Candidacy: Department representatives may want to withdraw from the faculty representative candidacy. In this case, they can do this from the "Withdraw from Faculty Representative Candidacy" page that only department representatives can see.

Use Case Diagram



2.3 User Characteristics

- All users must be competent to use a simple website and the devices/tools such as mobile phones to view this website.
- Students need to know how an election takes place and be able to vote.

2.4 Constraints

Online election system has interfaces from other applications, reliability requirements, safety, and security considerations. However, these constraints are mostly controllable.

- The system cannot operate without internet connection.
- If the İYTE ÖBS system becomes unavailable for any reason, users cannot use the system, since users' login to the system will be fully integrated with İYTE ÖBS.
- System requires remote server which enables the data storage. Because of this, when the server crashes for some reason the system will not be able to function until the server becomes available.
- Since the user information is stored in a database and this database may be vulnerable to attacks, user information may not be private.

2.5 Assumptions and Dependencies

User interfaces and functionalities of the system may change during the development process of the project. In addition, new functionalities can be added that can change the requirements of the system.

2.6 Apportioning of Requirements

The system will have only one version. All requirements will be done once in the first and final version of the system.

3. Specific Requirements

3.1 Use Cases

3.1.1 Use Cases for Student

3.1.1.1 Use case 1 for Student

3.1.1.1.1 Use case template

Use Case ID:	1000		
Use Case Name:	Login the system		
Created By:	Berkan Gönülsever	Last Updated By:	Berkan Gönülsever

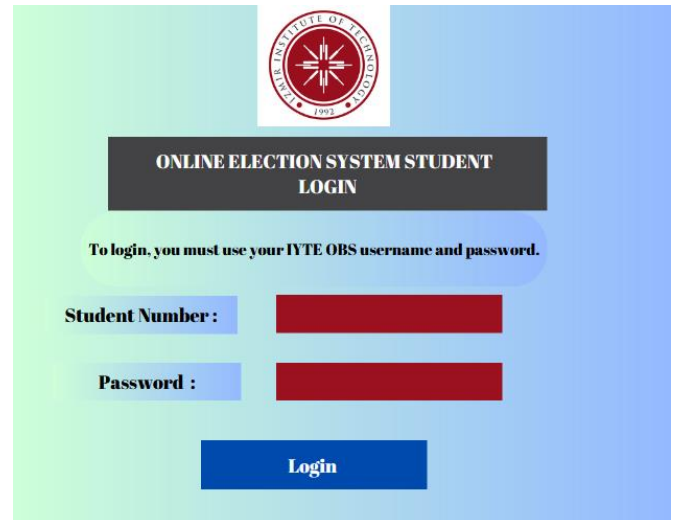
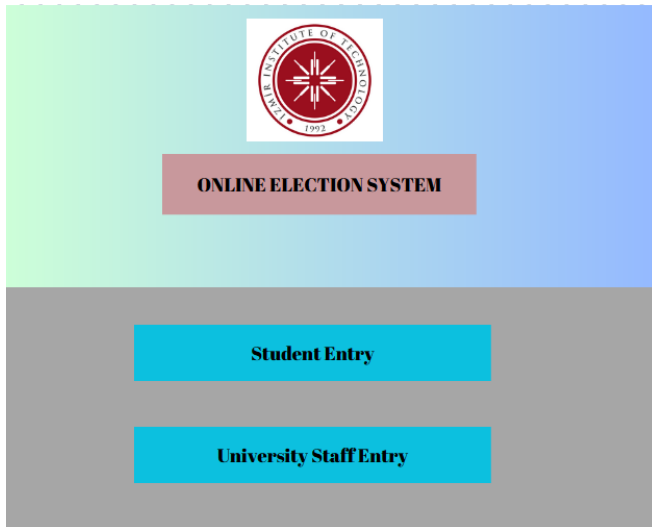
Date Created:	19.11.2022	Date Last Updated:	06.12.2022
---------------	------------	--------------------	------------

Actor:	Student, Secondary Actor(ÖBS)
Description:	Authenticated students can login their account.
Preconditions:	None
Postconditions:	POST-1:Students can access the system. POST-2:The student is directed to the home page.
Priority:	Medium
Frequency of Use:	Whenever students want to use the system.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The student goes into login section. 2. The student enters their student number and password. 3. The system gets related information from ÖBS. 3. The system matches the information sent by the student with the information coming from ÖBS. 4. The system accepts the student. 5. The student accesses the home page.
Alternative Courses:	1000.AC.1) Student number or the password is not correct. <ul style="list-style-type: none"> • 1000.AC.1.1) The system notifies that information is not correct.
Exceptions:	1000.EX.1) The system was unable to get related information from ÖBS. <ul style="list-style-type: none"> • 1000.EX.1.1) The system informs the student about the error.
Includes:	None
Special Requirements:	None
Assumptions:	1) The student is connected to the Internet.
Notes and Issues:	None

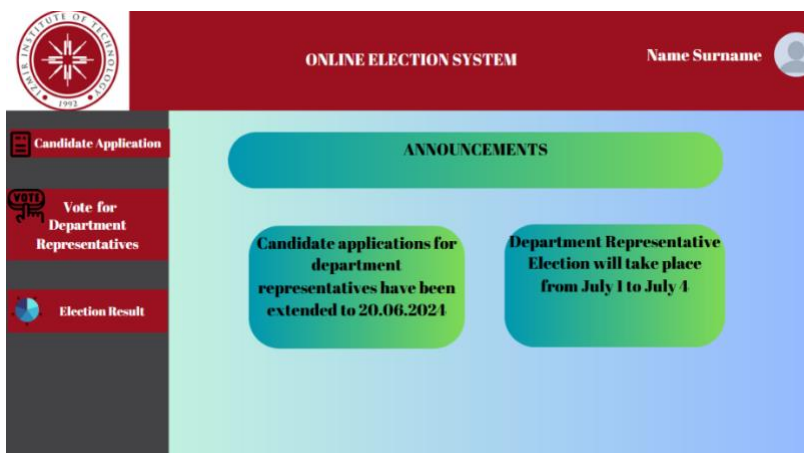
3.1.1.1.2 User Interface

- **Login Interface**

Student who wants to enter the system must first prove that he is a user, therefore the student encounters a screen asking for the student number and ÖBS password. The student can log in to the system by pressing the login button after entering her information.



Homepage for student



3.1.1.1.3 Performance/software quality requirements

- PER-1. 90% of matching must be completed less than 15 seconds.
- ROB-1. The system should be able to recover itself within 20 minutes in 90% of the times when the system cannot receive information from the ÖBS.
- INT-1. The student must enter the account that matches the information entered.
- INT-2. The system should get the information from the ÖBS without any errors.
- SEC-1. Both the information sent by the student and the information coming from ÖBS are encrypted.

3.1.1.2 Use case 2 for Student

3.1.1.2.1 Use case template

Use Case ID:	1001		
Use Case Name:	Apply for candidacy		
Created By:	Berkan Gönülsever	Last Updated By:	Berkan Gönülsever
Date Created:	19.11.2022	Date Last Updated:	06.12.2022

Actor:	Student
Description:	Every student has the right to apply for candidacy, but they must meet certain qualification requirements to become a candidacy.
Preconditions:	PRE-1: The student logged into the system. PRE-2: The student must apply for candidacy on the scheduled application date.

Postconditions:	POST-1:It will be verified whether the students fulfill the requirements for candidacy.
Priority:	High
Frequency of Use:	Once every 2 years within a specified period.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The student selects the candidacy process section from the menu. 2. The student enters the necessary information into the system for student candidacy procedures. 3. The system evaluates the candidacy request by using the information from the database and the information obtained from the student. 4. Student's candidacy application is accepted. 5. Student's candidacy application is recorded by the system. 6. The student is notified of the candidacy result.
Alternative Courses:	<p>1001.AC.1) Student's candidacy application is rejected. 1001.AC.1.1) The reason for rejection is notified to the student.</p> <p>1001.AC.2) The student didn't enter necessary information. 1001.AC.2.1) The system notifies the student that there is missing information.</p> <p>1001.AC.3) Uploaded documents are in the wrong format 1001.AC.3.1) The system notifies the student that documents are in the wrong format.</p> <p>1001.AC.4) The student selects the candidacy process before evaluation is completed. 1001.AC.3.1) The system notifies the student that application is under evaluation.</p>
Exceptions:	<p>1001.EX.1) The system failed to save the student's candidacy. 1001.EX.1.1) The system notifies the student about error. 1001.EX.1.2) If the error is fixed, The system requests a new application from the student.</p>
Includes:	<ul style="list-style-type: none"> • 2001.Check the Candidacy Application • 2005.Provide Feedback
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1) The student is connected to the Internet. 2) The owner of the account is logged into the account.
Notes and Issues:	None

3.1.1.2.2 User Interface

- **Candidacy Process Interface**

Students can see what information the system wants from them as a form and after filling the gaps in the form, they can submit the application by clicking submit button.

The screenshot shows a web interface for a 'Candidate Application'. At the top, there is a red header bar containing the university logo on the left and the title 'Candidate Application' in the center. To the right of the title, there is a label 'Name Surname'. Below the header, the form is divided into two main sections. The left section, with a light green background, contains five input fields: 'Name:', 'Student Number:', 'Major:', 'Grade:', and 'Year:'. The right section, with a light blue background, contains three upload buttons: 'Criminal Report', 'Student Certificate', and 'Member or official in political party organ'. Each upload button has a corresponding input field and an 'upload' button. At the bottom right of the form, there is a blue 'Submit' button.

3.1.1.2.3 Performance/software quality requirements

- PER-1. 90% of candidacy application must be received by the system less than 30 seconds.
- SCA-1. The system must be able to process at least 1000 candidacy application.

3.1.1.3 Use case 3 for Student

3.1.1.3.1 Use case template

Use Case ID:	1002		
Use Case Name:	To vote for department representative election		
Created By:	Berkan Gönülsever	Last Updated By:	Berkan Gönülsever
Date Created:	19.11.2022	Date Last Updated:	06.12.2022

Actor:	Student
Description:	Students vote through the system to determine their representatives.
Preconditions:	PRE-1: The student logged into the system. PRE-2: The student must vote on the scheduled voting date.

Postconditions:	POST-1: The students' votes will be recorded by the system. POST-2: Votes which candidates received are updated.
Priority:	High

Frequency of Use:	It is repeated between once or three times in 2 years, depending on the absolute majority.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The student selects the voting for department representative section from the menu. 2. The system checks if the student has voted before. 3. The system controls student's department. 4. The system directs the student to the voting menu. 5. The student votes for the candidate they want. 6. The system records the student's vote. 7. The system updates the votes which candidates received. 8. The system informs the student that the vote has been processed.
Alternative Courses:	<p>1002.AC.1) Student has voted before.</p> <ul style="list-style-type: none"> • 1002.AC.1.1) The system displays election form. <p>1002.AC.2) No candidate is available.</p> <ul style="list-style-type: none"> • 1002.EX.2.1) The system notifies the student that there is no candidate.
Exceptions:	<p>1002.EX.1) The system failed to record the student's vote.</p> <ul style="list-style-type: none"> • 1002.EX.1.1) The system informs the student about the error. • 1002.EX.1.2) If the error is fixed, The system prompts the student to repeat the vote.
Includes:	None
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1) The student is connected to the Internet. 2) The owner of the account is logged into the account.
Notes and Issues:	None

3.1.1.3.2 User Interface

• Voting Interface

The student encounters a page containing the candidates of the department he/she is in.

After student selected his/her candidacy, she can complete the voting process by clicking the vote button.

• Displaying Interface

If the student has completed the voting process before, they will be transferred here after clicking the voting for department representative button. Here the student can view the election form and her vote.



Vote for Department Representatives

Candidates	Candidates's info	Vote
		VOTE
		VOTE
		VOTE
		SUBMIT

3.1.1.3.3 Performance/software quality requirements

- PER-1. 95% of the student's vote must be recorded less than 10 seconds.
- INT-1. All votes must be recorded and updated without any error.
- ROB-1. The system should be able to recover itself within 5 minutes in 95% of the times when the system cannot record the student's vote.
- SEC-1. After a student votes, no one can see the student's vote.

3.1.1.4 Use case 4 for Student

3.1.1.4.1 Use case template

Use Case ID:	1003		
Use Case Name:	Access election results		
Created By:	Berkan Gönülsever	Last Updated By:	Berkan Gönülsever
Date Created:	19.11.2022	Date Last Updated:	06.12.2022

Actor:	Student
Description:	System provides a reliable way for student to learn about the all election results.
Preconditions:	PRE-1:Student logged into the system.
Postconditions:	POST-1:Student accessed the election results.
Priority:	Medium
Frequency of Use:	Whenever students want to learn election results.
Normal Course of Events:	1. The student selects the election results section from the menu. 2. The system directs the student election results menu where the student can select an election result they want to learn. 3. The student selects an election result they want to learn. 4. The system checks if the election is concluded. 5. The system displays the election results.
Alternative Courses:	1003.AC.1) The election is not concluded. <ul style="list-style-type: none">• 1003.AC.1.1) The system informs the student that the election result has not been concluded yet.
Exceptions:	1003.EX.1) The system failed to check if the election is concluded. <ul style="list-style-type: none">• 1003.EX.1.1) The system informs the student about the error.• 1003.EX.1.2) The system shows the unchecked state of the election results until the error is fixed.
Includes:	2002.Confirm Election Results.
Special Requirements:	None
Assumptions:	1) The student is connected to the Internet. 2) The owner of the account is logged into the account.

Notes and Issues:	None
-------------------	------

3.1.1.4.2 User Interface

• Election Results Interface

Student uses this menu to view current election results. The student can see department representatives, faculty representatives and student council president here.

3.1.1.4.3 Performance/software quality requirements

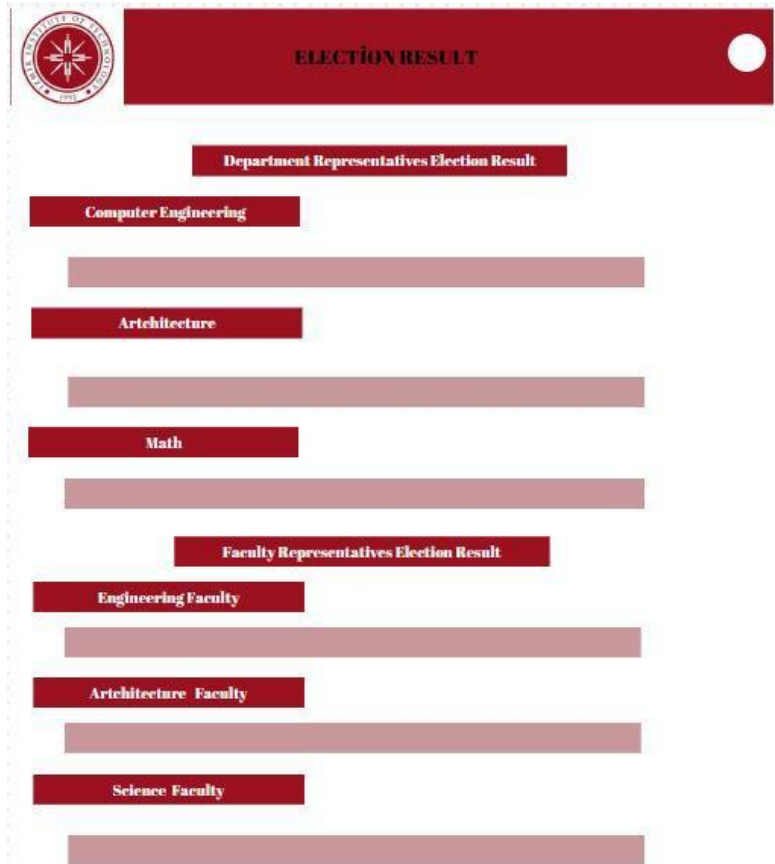
- PER-1. 90% of election result must be accessible less than 15 seconds.
- INT-1. It should be updated within 10 minutes when there is any change regarding the election results.

3.1.1.5 Use case 5 for Student

3.1.1.5.1 Use case template

Use Case ID:	1004		
Use Case Name:	View Announcements		
Created By:	Berkan Gönülsever	Last Updated By:	Berkan Gönülsever
Date Created:	26.11.2022	Date Last Updated:	06.12.2022

Actor:	Student
Description:	It is the panel where students can see the announcements from the student affairs.
Preconditions:	PRE-1:Student logged into the system.
Postconditions:	POST-1:Student viewed the announcements.
Priority:	Medium
Frequency of Use:	Whenever student wants to view their announcements.
Normal Course of Events:	<ol style="list-style-type: none"> 1. Student selects the announcements section from the menu. 2. The system checks if there is a new announcement or an existing announcement is edited. 3. There are no change in announcements. 4. The student accesses the announcements.



	5. The student views available announcements.
Alternative Courses:	<p>1004.AC.1) There is a new announcement in the announcements section.</p> <ul style="list-style-type: none"> • 1004.AC.1.1) The student accesses the announcements. • 1004.AC.1.2) The student encounters a notification which tells there is a new announcement. • 1004.AC.1.3) The student views the content of the new announcement. <p>1004.AC.2) An existing announcement was edited.</p> <ul style="list-style-type: none"> • 1004.AC.2.1) The student accesses the announcements. • 1004.AC.2.1) The student encounters a notification which tells an existing announcement has been changed. • 1004.AC.2.1) The student views the content of the modified announcement.
Exceptions:	<p>1004.EX.1) The system failed to check if there is a change in announcement.</p> <ul style="list-style-type: none"> • 1004.EX.1.1) The system informs the student about the error. • 1004.EX.1.2) The system shows the unchecked state of the announcements until the error is fixed.
Includes:	<ul style="list-style-type: none"> • 5000.Add Announcement • 5001.Edit Announcement
Special Requirements:	None
Assumptions:	<p>1) The student is connected to the Internet.</p> <p>2) The owner of the account is logged into the account.</p>
Notes and Issues:	None

3.1.1.5.2 User Interface

- **Announcements Interface**

When there is an issue that students need to know, the system sends an announcement to the students from this section. Each student has a limited amount of space to receive announcements. Students can see old and new announcements here.

3.1.1.5.3 Performance/software quality requirements

- INT-1. It should be updated within 10 minutes when there is a new announcement.
- SCA-1. It should store at least 100 announcements.

3.1.2 Use case for University Administration

3.1.2.1 Use case 1 for University Administration

3.1.2.1.1 Use case template

Use Case ID:	2000		
Use Case Name:	Login the system		
Created By:	Osman ALTUNAY Fatih YELBOĞA	Last Updated By:	Osman ALTUNAY Fatih YELBOĞA
Date Created:	19.11.2022	Date Last Updated:	06.12.2022

Actor:	University Administration
Description:	University Administration can log in to the system.
Preconditions:	None
Postconditions:	POST-1:University Administration can access the system.
Priority:	Medium
Frequency of Use:	depending on the number of tasks of University Administration
Normal Course of Events:	1. University Administration go into login section. 2. University Administration enter their username and password. 3. System checks information University Administration sends. 4. System accepts the user. 5. University Administration accesses the home page.
Alternative Courses:	1-) Username and password are not correct 1.1-)The system notifies that information is not correct.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The system has at least one administrator.
Notes and Issues:	None

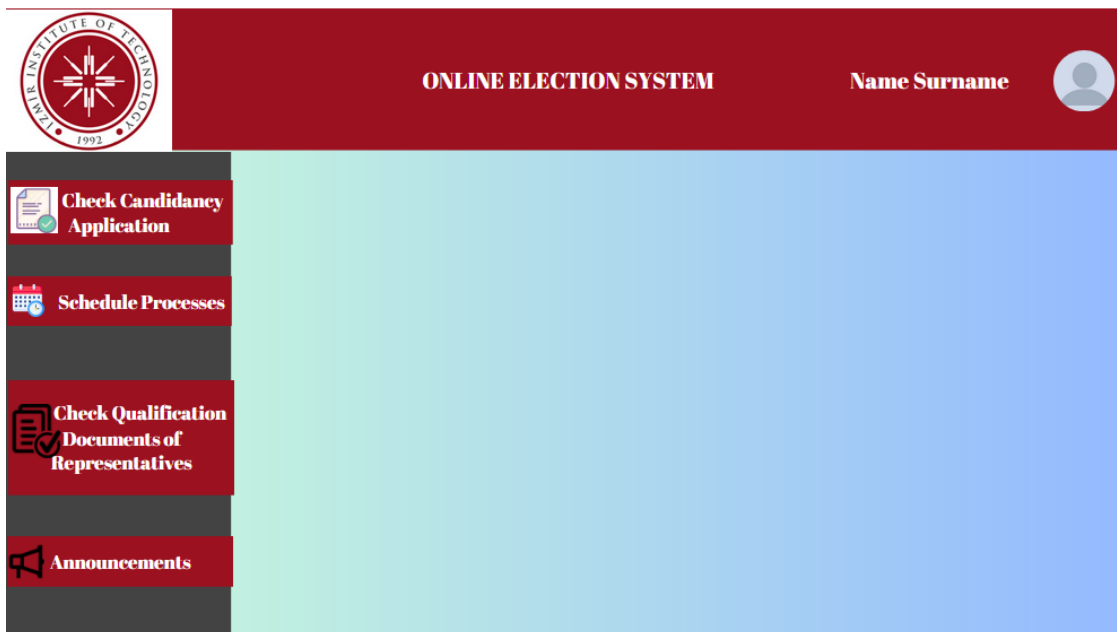
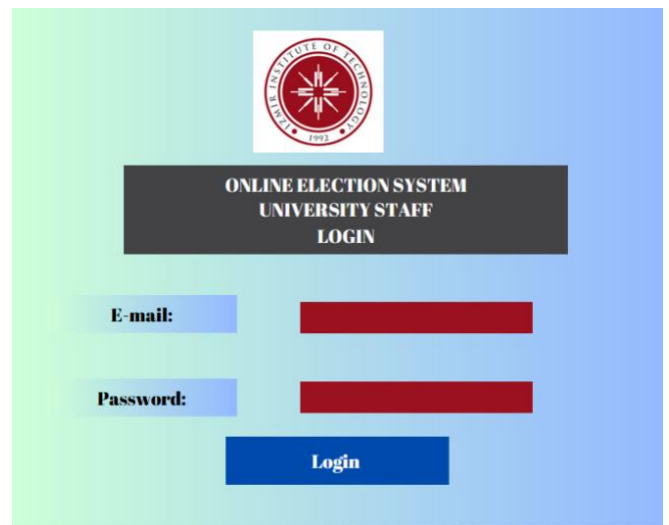
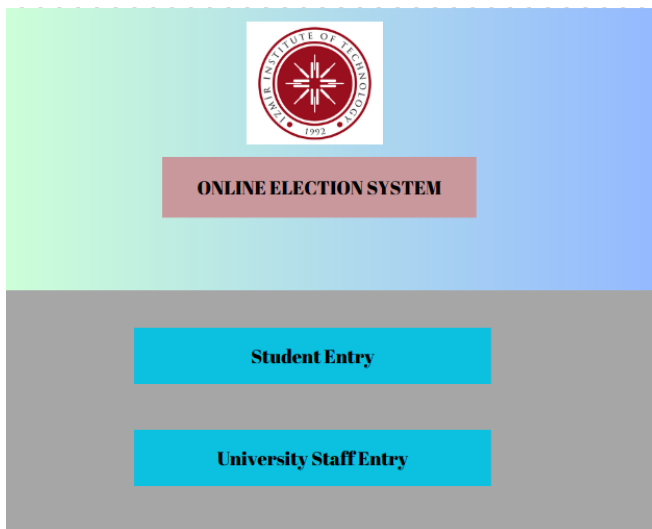
3.1.2.1.2 User Interface

- Admin login interface

Users who are administrators in the system log into the system with this interface. They can log in to the system by correctly filling in the username and password sections on the interface screen.

3.1.2.1.3 Performance/software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.



3.1.2.2 Use case 2 for University Administration

3.1.2.2.1 Use case template

Use Case ID:	2001		
Use Case Name:	Check the candidacy application		
Created By:	Osman ALTUNAY Fatih YELBOĞA	Last Updated By:	Osman ALTUNAY Fatih YELBOĞA

Date Created:	21.11.2022	Date Last Updated:	06.12.2022
---------------	------------	--------------------	------------

Actor:	University Administration Candidate
Description:	Some conditions are requested from the candidates in order to make an election application and they upload them in the system while applying. University administration checks the eligibility of these applications.
Preconditions:	Each candidate uploads an application form to the system.
Postconditions:	University administration approves or rejects applications.
Priority:	High
Frequency of Use:	Depends on the number of candidates.
Normal Course of Events:	1-The student who wants to be a candidate enters the system. 2- The candidate fills the application form and uploads the required documents completely 3-University administration reviews the candidate application. 4-University administration approves the candidate application in the system. 5-The candidate is registered in the system.
Alternative Courses:	1.1 Candidate application is not approved 1. University administration provides feedback to the candidate 2. Candidate resubmits the application form.
Exceptions:	1.1-Failure to receive the necessary information for competency control from ÖBS. - University Administration receives necessary information from the ÖBS system
Includes:	None
Special Requirements:	None
Assumptions:	Students can obtain documents via e-Devlet.
Notes and Issues:	None

3.1.2.2.2 User Interface

- **Review Interface**
- The interface where the university administration reviews the documents
- Displays the candidate information and documents.
- Provides a reference to the Feedback/Mail Interface to prepare feedback

3.1.2.2.3 Performance/software



Candidate Applications

Name	Student's info	Confirmation
Berkan Gonulsever	berkan_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Enes Demirel	enes_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Fatih Yelboğa	fatih_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Gökay Gökay	gokay_info	<input checked="" type="checkbox"/> <input type="checkbox"/>

quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

3.1.2.3 Use case 3 for University Administration

3.1.2.3.1 Use case template

Use Case ID:	2002		
Use Case Name:	Add the process/schedule		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.05.2023

Actor:	University Administration
Description:	The election system process consists of 2 parts, the pre-preparation process where the documents are loaded, and the selection process. The admin adds the schedule/process to prepare the election system.
Preconditions:	None
Postconditions:	It is the initiation of events related to the relevant process in the student panels according to the actions of the admin on the processes.
Priority:	High
Frequency of Use:	Depends on frequency of the election and election preparation.
Normal Course of Events:	1- The admin user logs into the system. 2- The admin goes to the schedule menu. 3- The admin clicks the add process/schedule button. 4- The admin enters the process/schedule type, start and end date inputs. 5- The admin sends the process/schedule to the system by clicking the add button.
Alternative Courses:	If an inappropriate process is added to the program, a warning message will be returned to the system admin.

Exceptions:	None
Includes:	None
Special Requirements:	None

Assumptions:	- One of the most important components of the system is to add the process correctly so suppose that the admin needs to manage this process well.
Notes and Issues:	None

User Interface

• Process/Schedule Interface

- The interface in which university administration manages the schedule/process.
- The university administration prepares the election process by managing the schedule.

3.1.2.3.2 Performance/software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

3.1.2.4 Use case 4 for University Administration

3.1.2.4.1 Use case template

Use Case ID:	2003		
Use Case Name:	Edit the process/schedule		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.05.2023

Actor:	University Administration
Description:	The election system process consists of 2 parts, the pre-preparation process where the documents are loaded, and the selection process. The admin updates the process/schedule to make the changes.
Preconditions:	None
Postconditions:	It is the initiation of events related to the relevant process in the student panels according to the actions of the admin on the processes.
Priority:	High
Frequency of Use:	Depends on frequency of the election and election preparation.

Normal Course of Events:	1- The admin user logs into the system. 2- The admin goes to the schedule menu. 3- The admin clicks the edit process/schedule button. 4- The admin enters the process/schedule type, start and end date inputs. 5- The admin sends the process/schedule to the system by clicking the edit button.
Alternative Courses:	If an inappropriate process is updated to the program, a warning message will be returned to the system admin.

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	- One of the most important components of the system is to edit the process correctly so suppose that the admin needs to manage this process well.
Notes and Issues:	None

User Interface

• Process/Schedule Interface

- The interface in which university administration manages the schedule/process.
- *The university administration prepares the election process by managing the schedule.*

3.1.2.4.2 Performance/software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

3.1.2.5 Use case 5 for University Administration

3.1.2.5.1 Use case template

Use Case ID:	2004		
Use Case Name:	Remove the process/schedule		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.05.2023

Actor:	University Administration
Description:	The election system process consists of 2 parts, the pre-preparation process where the documents are loaded, and the selection process. The admin removes the related wrong process/schedule to delete from the system.
Preconditions:	None
Postconditions:	It is the initiation of events related to the relevant process in the student panels according to the actions of the admin on the processes.
Priority:	High
Frequency of Use:	Depends on frequency of the election and election preparation.
Normal Course of Events:	1- The admin user logs into the system. 2- The admin goes to the schedule menu. 3- The admin clicks the remove process/schedule button of the related process/schedule. 4- The process/schedule is removed by the admin from the system.
Alternative Courses:	If an inappropriate process is deleted to the program, a warning message will be returned to the system admin.

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	- One of the most important components of the system is to remove the process correctly so suppose that the admin needs to manage this process well.
Notes and Issues:	None

User
Interface

• Process/Schedule Interface

- The interface in which university administration manages the schedule/process.
- The university administration prepares the election process by managing the schedule.

3.1.2.5.2 Performance/software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

3.1.2.6 Use case 6 for University Administration

3.1.2.6.1 Use case template

Use Case ID:	2005		
Use Case Name:	Check qualification of representative post-election		
Created By:	Osman ALTUNAY Fatih YELBOĞA	Last Updated By:	Osman ALTUNAY Fatih YELBOĞA

Date Created:	21.11.2022	Date Last Updated:	06.12.2022
---------------	------------	--------------------	------------

Actor:	University Administration Representative
Description:	University administration checks that all elected representatives have the qualifications of representatives during their term of office. Representatives upload the necessary documents to the system at regular intervals.
Preconditions:	Representatives upload documents to the system
Postconditions:	Representative may continue or be dismissed
Priority:	Medium
Frequency of Use:	2 times a year for each representative
Normal Course of Events:	<ol style="list-style-type: none"> 1- Representatives enter the system. 2- Representatives upload the requested documents to the system. 3- University administration reviews the documents in the system 4- University administration completes document checks. 5- The representative continues his/ her duties until the next qualification check.

Alternative Courses:	1.1 Representative uploads missing documents 1- The system does not accept the form and the form cannot be submitted. 2- Representative submits the form again
Exceptions:	1.2- Failure to receive the necessary information for competency control from ÖBS. - University Administration receives necessary information from the ÖBS system
Includes:	This use case includes ‘University Administration use case 2’ use case for the Review Interface
Special Requirements:	None
Assumptions:	Representative can obtain documents via e-Devlet.
Notes and Issues:	None

3.1.2.6.2 User Interface

- **Review Interface**

- The interface where the university administration reviews the documents
- Displays the candidate information and documents.
- Provides a reference to the Feedback/Mail Interface to prepare feedback

3.1.2.6.3 Performance/software quality requirements

The system should support up to 200 process simultaneous.

The system should handle up to 100 users without break or inconsistency.



CHECK QUALIFICATION CONTROL DOCUMENTS

Name	Representatives' info	Confirmation
Berkan Gonulsever	berkan_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Enes Demirel	enes_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Fatih Yelboğa	fatih_info	<input checked="" type="checkbox"/> <input type="checkbox"/>
Gökay Gülsoy	gokay_info	<input checked="" type="checkbox"/> <input type="checkbox"/>

3.1.2.7 Use case 7 for University Administration

3.1.2.7.1 Use case template

Use Case ID:	2006		
Use Case Name:	Dismissal of the representative		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.12.2022

Actor:	University Administration
Description:	Representatives are requested to upload current documents to the system within the specified time interval and in case of violation, the representative is dismissed by the university administration.
Preconditions:	The representative has violated any of the rules for being a representative candidate.
Postconditions:	The representative is dismissed by the university administration.
Priority:	High
Frequency of Use:	Once every 4 months

Normal Course of Events:	1- The decision is approved for the representative to be dismissed. 2- University administration displays the role interface with all students and their roles. 3- The representative is found on the student/role interface where the students are located.
	4- University administration clicks the dismiss button 5- If the 'YES' button is clicked on the confirmation screen, the duty of the relevant representative is dismissed.
Alternative Courses:	The representative appeal. - The representative who has been determined to be dismissed can apply to the relevant university administration to appeal this decision.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	- The representative does not protest during the objection period, their objections will not be accepted even if he/she is right.

3.1.2.7.2 User Interface

Student/Role Interface

The interface where the university administration displays the students and their roles.
Provide all students and their roles in the university by integrating with 'OBS'
Has a simple interface and button that removes the relevant student's role.

3.1.2.7.3 Performance/software quality requirements

- The system should support up to 200 process simultaneous
- The system should handle up to 100 users without break or inconsistency

3.1.2.8 Use case 8 for University Administration

3.1.2.8.1 Use case template

Use Case ID:	2007		
Use Case Name:	Provide feedback		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.12.2022

Actor:	University Administration
--------	---------------------------

Description:	University administration provides feedbacks to the candidates based on documents that are incorrect, incomplete or do not meet the requirements in the process of the candidacy for department representative.
Preconditions:	Candidates must send the incorrect documents to be existed the action.
Postconditions:	Candidates who have not suitable documents are informed via mail by the university administration.
Priority:	Medium

Frequency of Use:	Per process of the candidacy for department representative
Normal Course of Events:	<ol style="list-style-type: none"> 1. The candidates and their documents are displayed in the review interface 2. University administration checks the documents 3. Candidates who have incorrect documents are detected. 4. Candidates mails and notification text are prepared for feedback. 5. The feedbacks are sent to the relevant candidates to inform by university administration.
Alternative Courses:	None
Exceptions:	None
Includes:	Use case 'Check qualification of representative'
Special Requirements:	None
Assumptions:	- Candidates have the active mail provided by the university.
Notes and Issues:	None

3.1.2.8.2 User Interface

Review Interface

The interface where the university administration reviews the documents

Displays the candidate information and documents.

Provides a reference to the Feedback/Mail Interface to prepare feedback.

Feedback/Mail Interface

The interface where the university administration provides feedbacks to the relevant candidates

Includes the candidate mail and notification text.

3.1.2.8.3 Performance/software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

3.1.3 Use Case for Department Representative

3.1.3.7 Use Case 1 for Department Representative

Use Case ID:	3000		
Use Case Name:	Withdraw from faculty representative candidacy.		
Created By:	Merve Nur Ozan	Last Updated By:	Merve Nur Ozan
Date Created:	21.11.2022	Date Last Updated:	

Actor:	Department Representative
Description:	Department representatives who want to withdraw from faculty representative candidates notifies the system.

Preconditions:	Department representatives' selection is over and were determined.
Postconditions:	Faculty representatives' candidates are finalized.
Priority:	Medium
Frequency of Use:	Department representatives can withdraw whenever they want during the withdraw process for faculty representatives.
Normal Course of Events:	<ol style="list-style-type: none"> 1. Department representative login the system. 2. The system redirects department representatives to homepage. 3. Department representatives select withdraw section. 4. The system redirects the department representatives to withdraw section. 5. Department representatives notify the system that he/she wants to withdraw from faculty representatives' candidacy with choosing that withdraw from faculty representatives.
Alternative Courses:	<p>A.C. 5.1 Department representatives have notified the system that they want to withdraw from the candidacy of faculty representatives before.</p> <ol style="list-style-type: none"> 3- The system informs the department representative that their request has been sent to university administration before.
Exceptions:	<p>E. 5.1 The system failed to send the department representatives' request.</p> <ul style="list-style-type: none"> • The system informs the department representatives about the error. • If the error is fixed, system prompts the department representatives to repeat their processes that he/she did.
Includes:	None
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1. Department representatives connected to internet. 2. Department representative login the system successfully 3. System knows which students are department representatives.
Notes and Issues:	None

3.1.3.1 User Interface

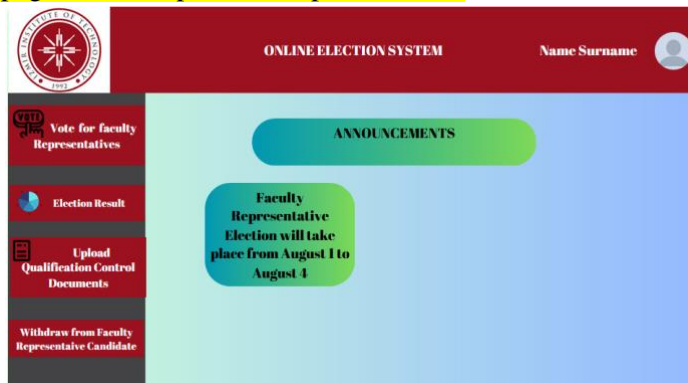
Login Interface

The system is integrated with OBS. Department representatives log in with his/her student number and password the same as OBS login information.

Homepage Interface

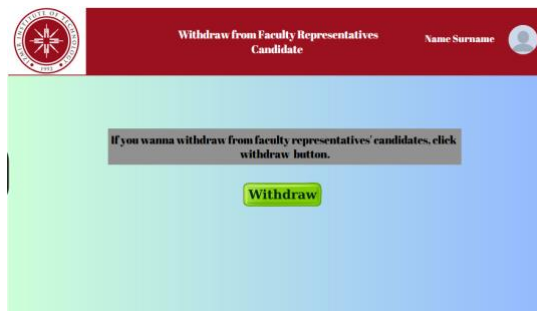
On the main page of the website, there is a menu for tabs such as the home page, faculty representative election, withdraw from faculty representative candidate, qualification control, etc.

Homepage for department representatives



Notifies the system

There is a page that department representatives can notify the system that he/she wants to withdraw from faculty representatives' candidates.



3.1.3.2 Performance /Software quality requirements

The system should support taking up to at least 14 different department representatives can receive requests at the same time, and system should perform this operation without any inconsistency.

The system should support taking up to all different department representative petitions at the same time, and system should perform this operation without any inconsistency.

3.1.3.8 Use Case 2 for Department Representative

Use Case ID:	3001		
Use Case Name:	View withdrawal status		
Created By:	Merve Nur Ozan	Last Updated By:	
Date Created:	21/11/2022	Date Last Updated:	

Actor:	Department Representative
Description:	Department representatives can see their withdrawal status
Preconditions:	Department representatives withdrawn from faculty

	representative candidacy.
Postconditions:	Department representatives can look at their withdrawal status.
Priority:	Medium
Frequency of Use:	Department representatives can see whenever they want.
Normal Course of Events:	<ol style="list-style-type: none"> 1. Department representative login the system. 2. The system redirects department representatives to homepage. 3. Department representatives select withdraw section. 4. The system redirects the department representatives to withdrawsection. 5. If the department representative wants to see their withdrawal status after the withdrawal process, system displays a notification whether their withdraw request has been approved by the university administration or not.
Alternative Courses:	<p>5.1 If the department representative wants to see their withdrawal status during the withdrawal process is continuing.</p> <ul style="list-style-type: none"> • The system informs the department representative that their request has been sent to university administration before.
Exceptions:	<p>The system failed to view the department representatives' withdrawal status.</p> <ul style="list-style-type: none"> • The system informs the department representatives about the error.
Includes:	'Withdraw from faculty representative candidacy.

Special Requirements:	
Assumptions:	1. Department representatives connected to internet 2.Department representative login the system successfully 3.System knows which students are department representatives.
Notes and Issues:	None

3.1.3.8.1 User Interface

Login Interface

The system is integrated with OBS. Department representatives log in with his/her student number and password the same as OBS login information.

Homepage Interface

On the main page of the website, there is a menu for tabs such as the home page, faculty representative election, withdraw from faculty representative candidate, qualification control, etc.

Notifies the system

There is a page that department representatives can see withdrawal status.

3.1.3.8.2 Performance/software quality requirements

The system should support taking up to 14 different department representatives can see their withdrawal status at the same time, and system should perform this operation without any inconsistency.

3.1.3.9 Use Case 3 for Department Representative

Use Case ID:	3002		
Use Case Name:	Vote for faculty representative election		
Created By:	Merve Nur Ozan	Last Updated By:	
Date Created:	21/11/2022	Date Last Updated:	

Actor:	Department representative
Description:	Faculty representatives are selected from among department representatives.
Preconditions:	1. Department representatives were determined for each department. 2. Faculty representatives' candidates are finalized.

Postconditions:	Faculty representatives will be determined for each faculty.
Priority:	Medium
Frequency of Use:	All department representatives at IZTECH vote whenever they want during the selection process for faculty representatives.
Normal Course of Events:	1. Department representative login the system. 2. The system redirects department representatives to homepage. 2. The department representatives select faculty representativeselection process. 3.The system redirects the department representatives to the facultyrepresentative selection process. 4. The system checks if the department representative has votedbefore or not. (4.1) 5. The system shows department representatives to only his/herfaculty representative candidates. 6. Department representatives vote one of them. (6.1)
Alternative Courses:	4.1. Department representatives has voted before. The system notifies the department representative the votingprocesses have been already done
Exceptions:	6.1 The system failed to record the department representatives' votes. <ul style="list-style-type: none"> The system informs the department representatives about the error. If the error is fixed, system prompts the department representatives to repeat their processes that he/she

	voted.
Includes:	None
Special Requirements:	None
Assumptions:	1. Department representatives connected to internet. 2. Department representative login the system successfully. 3. System knows which students are department representatives. 4. There are enough faculty representative candidates for each faculty.
Notes and Issues:	None

3.1.3.9.1 User Interface

Login Interface

The system will be integrated with OBS. Department representatives log in with his/her student number and password the same as OBS login information.

Homepage Interface

On the main page of the website, there is a menu for tabs such as the home page, faculty representative election, withdraw from faculty representative candidate, qualification control, etc.

Voting Interface

It shows faculty representatives candidates that department representatives vote one of them.



Vote for Faculty Representatives

Candidates	Candidates's info	Vote
		VOTE
		VOTE
		VOTE

SUBMIT

3.1.3.9.2 Performance/Software quality requirements

- The system should support receiving at least 14 of the department representatives votes for faculty representatives candidates votes at the sametime.

3.1.3.10 Use case 4 for Department Representative

Use Case ID:	3003		
Use Case Name:	Upload documents for qualification control		
Created By:	Merve Nur Ozan	Last Updated By:	
Date Created:	21/11/2022	Date Last Updated:	

Actor:	Department Representative
Description:	Department representatives upload their documents to continuethair duty
Preconditions:	Department representatives were determined.

Postconditions:	It will be decided whether the department representative will continue to duty or not.
Priority:	Medium
Frequency of Use:	All department representatives at IZTECH upload their documents whenever they want during the qualification control process for department representatives.
Normal Course of Events:	<ol style="list-style-type: none"> 1. Department representative login the system. 2. The system redirects department representatives to homepage. 3. Department representatives select qualification control section. 4. The system redirects the department representatives to qualification control section. 5. Department representatives upload the newly dated documents to the system. (5.1)
Alternative Courses:	<p>5.1 Department representatives have uploaded documents before.</p> <p>The system informs the department representative that their documents has been uploaded before.</p>
Exceptions:	<p>5.1 There may be a system error while uploading their documents.</p> <ul style="list-style-type: none"> • The system informs the department representatives about the error. • If the error is fixed, system prompts the department representatives to repeat their processes that he/she did. <p>5.2 The department representative may attempt to upload a file larger than 20mb.</p> <p>The system gives an error that you cannot upload files of 20 mb and larger.</p>
Includes:	None
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1. Department representatives connected to internet. 2. Department representative login the system successfully. 3. System knows which students are department representatives.
Notes and Issues:	None

3.1.3.10.1 User Interface

Login Interface

The system will be integrated with OBS. Department representatives log in with his/her student number and password the same as OBS login information.

Homepage Interface

On the main page of the website, there is a menu for tabs such as the home page, faculty representative election, withdraw from faculty representative candidate, qualification control, etc.

UPLOAD QUALIFICATION CONTROL DOCUMENTS MERVE NUR OZAN

Name:

Student Number:

Major:

Grade:

Year:

Criminal Report

Student Certificate

Member or official in political party organ

Qualification Control Interface

Department representatives can see what information the system wants from them and, then they upload necessary document, they can send the documents by clicking sending button.

3.1.3.10.2 Performance /Software quality requirements

- The system should support sending at least 14 of department representatives' qualification control documents to university administration simultaneously.

3.1.4 Use Case for Faculty Representative

3.1.4.7 Use Case 1 for Faculty Representative

Use Case ID:	4000		
Use Case Name:	Upload documents for qualification control		
Created By:	Merve Nur Ozan	Last Updated By:	
Date Created:	21/11/2022	Date Last Updated:	

Actor:	Faculty Representative
Description:	Faculty representatives upload their documents to continue their duty
Preconditions:	Faculty representatives were determined.
Postconditions:	It will be decided whether the faculty representative will continue their duty or not.
Priority:	Low

Frequency of Use:	All faculty representatives at IZTECH upload their documents whenever they want during the qualification control process for faculty representatives.
Normal Course of Events:	<ol style="list-style-type: none"> 1. Faculty representative login the system. 2. The system redirects faculty representatives to homepage. 3. Faculty representatives select qualification control section. 4. The system redirects the faculty representatives to qualification control section. 5. Faculty representatives upload the newly dated documents to the system. (5.1)
Alternative Courses:	5.1 Faculty representatives have uploaded documents before. The system informs the department representative that their documents has been uploaded before.
Exceptions:	<p>5.1 There may be a system error while uploading their documents.</p> <ul style="list-style-type: none"> • The system informs the faculty representatives about the error. • If the error is fixed, system prompts the faculty representatives to repeat their processes that he/she did. <p>5.2 The faculty representative may attempt to upload a file larger than 20mb. The system gives an error that you cannot upload files of 20 mb and larger.</p>
Includes:	None
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1. Faculty representatives connected to internet. 2. Faculty representative login the system successfully. 3. System knows which department representatives are faculty representatives.
Notes and Issues:	None

3.1.4.7.1 User Interface

Login Interface


The system will be integrated with OBS. Department representatives log in with his/her student number and password the same as OBS login information.

Homepage Interface


On the main page of the website, there is a menu for tabs such as the home page, faculty representative election, withdraw from faculty representative candidate, qualification control, etc.

Qualification Control Interface

Faculty representatives can see what information the system wants from them and, then they upload necessary document, they can send the documents by clicking sending button.



UPLOAD QUALIFICATION CONTROL DOCUMENTS

MERVE NUR OZAN


Name:

Student Number:

Major:

Grade:

Year :

Criminal Report upload

Student Certificate upload

Member or official in political party organ

Submit

3.1.4.7.2 Performance /Software quality requirements

- The system should support sending at least 3 of faculty representatives' qualification control documents to university administration simultaneously.

3.1.5 Use Cases for Student Affairs

3.1.5.1 Use case 1 for Student Affairs

3.1.5.1.1 Use case template

Use Case ID:	5000		
Use Case Name:	Add Announcement		
Created By:	Enes Demirel	Last Updated By:	Enes Demirel
Date Created:	27.11.2022	Date Last Updated:	27.11.2022

Actor:	Student Affairs
Description:	Adding new announcements to the announcements section where students can get information about the election process.
Preconditions:	PRE-1: User logged into the system. PRE-2: User must be an employee of student affairs.
Postconditions:	POST-1: New announcements have been added to the announcements section and new announcements are displayed first.
Priority:	Medium
Frequency of Use:	When all students must be informed during the election process.

Normal Course of Events:	<ol style="list-style-type: none"> 1. User clicks on the "Add new announcement" section that can only be visible to student affairs. 2. The user fills out the form where he/she can add a picture and a written explanation for the announcement to be added. 3. User clicks the "Submit" button. 4. New announcement publishes.
Alternative Courses:	5000.AC.1) User tries to submit blank form. System issues a warning message to the user that he/she cannot submit a blank form.
Exceptions:	5000.EX.1) The announcement could not be published due to some problem in the system. The announcement is sent to the e-mail addresses of the students registered in the system.
Includes:	None
Special Requirements:	None
Assumptions:	<ol style="list-style-type: none"> 1) User is connected to the Internet. 2) The owner of the account is logged into the account.
Notes and Issues:	None

3.1.5.1.2 User Interface

• Add New Announcement Interface

A web page that can only be seen by student affairs department employees who have the authority to add new announcements. It consists of a form where they can add pictures and written text, and a "Submit" button.

3.1.5.1.3 Performance/software quality requirements

The system should only allow one user to add a new announcement at a time.

3.1.5.2 Use case 2 for Student Affairs

3.1.5.2.1 Use case template

Use Case ID:	5001
Use Case	Edit Announcement

Name:			
Created By:	Enes Demirel	Last Updated By:	Enes Demirel
Date Created:	27.11.2022	Date Last Updated:	27.11.2022

Actor:	Student Affairs
Description:	Editing the announcements that need to be changed among the published announcements.
Preconditions:	PRE-1: User logged into the system. PRE-2: User must be an employee of student affairs.
Postconditions:	POST-1: The latest version of the edited announcement is published.
Priority:	Medium
Frequency of Use:	When it is necessary to change announcements containing incorrect information or typographical errors.
Normal Course of Events:	1. User clicks on the "Published announcements" section that can only be visible to student affairs. 2. The user clicks on the announcement he/she wants to edit. 3. The picture and written text of the announcement are presented to the user in an editable form. 4. User clicks the "Submit" button after process is finished. 5. Latest version of the announcement publishes.
Alternative Courses:	5001.AC.1) User tries to submit blank form. System issues a warning message to the user that he/she cannot submit a blank form.
Exceptions:	EX.1) The announcement could not be published due to some problem in the system. The updated version of the announcement is sent to the e-mail addresses of the students registered in the system.
Includes:	None
Special Requirements:	None
Assumptions:	1) User is connected to the Internet. 2) The owner of the account is logged into the account.
Notes and Issues:	None

3.1.5.1.2 User Interface

- **Edit Announcement Interface**

A web page that can only be seen by student affairs department employees who have the authority to edit published announcements. It consists of a form where they can edit pictures and written text of the published announcement, and a "Submit" button.

3.1.5.1.3

Performance/software quality requirements

- The system should only allow one user to edit one announcement at a time.

3.1.5.3 Use case 3 for Student Affairs

3.1.5.3.1 Use case template

Use Case ID:	5002		
Use Case Name:	Remove Announcement		
Created By:	Enes Demirel	Last Updated By:	Enes Demirel
Date Created:	07.05.2023	Date Last Updated:	07.05.2023

Actor:	Student Affairs
Description:	Removing announcement from the announcements section where students can get information about the election process.
Preconditions:	PRE-1: User logged into the system. PRE-2: User must be an employee of student affairs.
Postconditions:	POST-1: The announcement that determined by the student affair is removed from the announcements section.
Priority:	Medium
Frequency of Use:	When the announcement is not needed anymore.

Normal Course of Events:	. User clicks the announcement that he/she wants to remove. . User clicks on the "Remove announcement" section that can only be visible to student affairs. . User clicks "Yes" in the confirmation box that appears after selecting delete and asking "Are you sure?". . Announcement is removed.
Alternative Courses:	5002.AC.1) User selects "No" in the confirmation box after clicking "Remove announcement." No change takes place.
Exceptions:	5000.EX.1) The announcement could not be removed due to some problem in the system. System issues a warning message to the user saying "Operation could not be performed" asks him/her to do it later.
Includes:	None
Special Requirements:	None
Assumptions:	3) User is connected to the Internet. 4) The owner of the account is logged into the account.
Notes and Issues:	None

3.1.5.3.2 User Interface

Delete Announcement Interface

A web page that can only be seen by student affairs department employees who have the authority to delete published announcements.

3.1.5.3.3 Performance/software quality requirements

- The system should only allow one user to delete one announcement at a time.

3.1.6 Use Cases for All Users

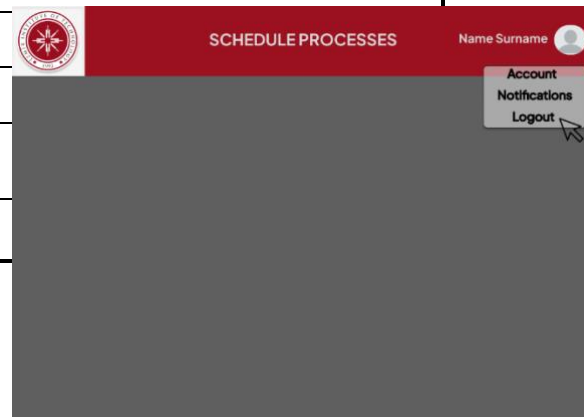
3.1.6.1 Use case 1 for All Users

3.1.6.1.1 Use case template

Use Case ID:	6000		
Use Case Name:	Log out from the system		
Created By:	Fatih YELBOGA Osman ALTUNAY	Last Updated By:	Fatih YELBOGA Osman ALTUNAY
Date Created:	18.11.2022	Date Last Updated:	06.05.2023

Actor:	Students, University Administrations (all system users)
Description:	Authenticated users can log out from the system.
Preconditions:	None
Postconditions:	Redirect to log in page.
Priority:	High
Frequency of Use:	Whenever the user wants to log out from the system.
Normal Course of Events:	1- The user clicks on the profile. 2- The user logs out from the system by clicking the log out button on the profile.
Alternative Courses:	None

Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None



3.1.6.1.2 User Interface

- **Log out Interface**

- The interface in which users log out from the system.

3.1.6.1.3 Performance /Software quality requirements

- The system should support up to 200 process simultaneous.
- The system should handle up to 100 users without break or inconsistency.

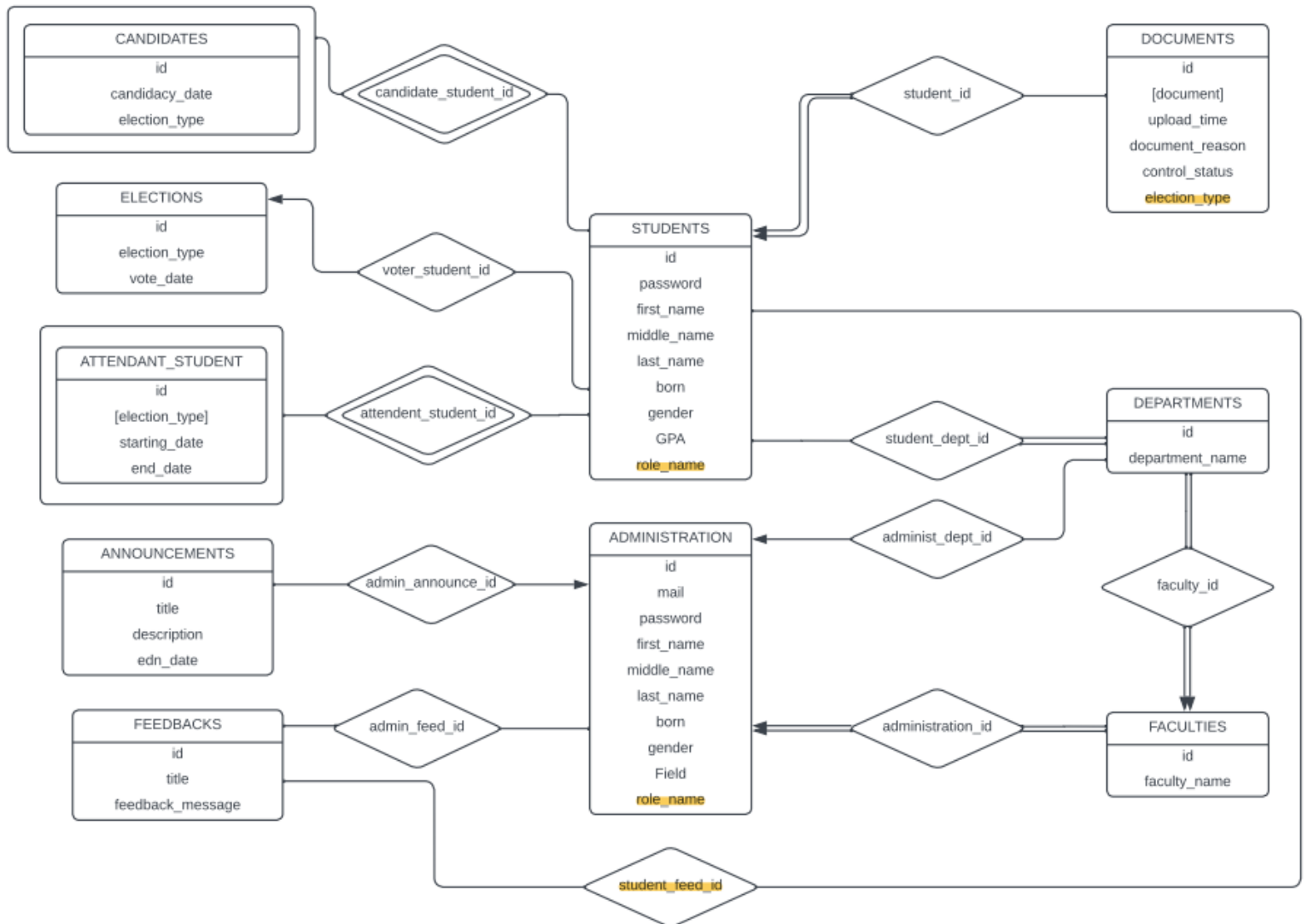
3.2 External interfaces (other than user interfaces)

There are no external interfaces other than use case interfaces

3.3 Generic performance/software requirements

- The database and the software project should be designed and integrated in the most optimum strategy to give the best result. For example, the project must minimize dependencies and reduce the data traffic.
- The precautions should be taken by using triggers/assumptions/constraints to avoid the errors/bugs that may occur in the database.
- The database must be large enough.
- Transition time between any two interfaces should be short.
- In order for the related interfaces to be understandable and aesthetic by the users, the software project should be simple and plain at the part of front-end.

3.4 Entity-Relationship Diagram



3.5 Design Constraints

Designing the web application is rarely an unconstrained process. This section is based on assumptions about future implementations.

- Programming language to be used - It is not known yet, but we are planning to build a web application using Java Spring and Oracle.
- Execution Environment or Operating System - No final decision could be reached with Business Partners.
- Performance expectations
- User Interface needs
- Privacy: after casting a vote, no one should be able to see the voter to this vote.
- Democracy: all eligible voters must be able to vote, one person - one vote and no one can vote more than once or vote for others.
- Security: throughout the voting process, a vote cannot be tampered with.

3.6 Other Requirements

1. For the selection system, the database of the university is needed. (Includes student information)
2. Internet is required for users to use the system.
3. The documents of the students who want to be candidates according to the election directive are required.
4. For users to login the system, university e-mails must be active. (Active student)