

**BILKENT UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING**

CS 319

**ERASMUS PROJECT
Group 2H**

ANALYSIS REPORT ITERATION 2

Emirkan Derköken	22002693
Murat Güney Kemal	22002692
Cengizhan Terzioğlu	22003014
Göktuğ Kuşçu	22002867
Bora Yılmaz	22003359
Onur Asım İlhan	21903375

Instructor: Eray Tüzün

Teaching Assistants: Metehan Saçakçı, Emre Sülün, Muhammad Umair Ahmed,
İdil Hanhan, Mert Kara

Table of Contents

1.	3	
2.	4	
2.1 Functional Requirements		2
2.1.1 Authentication & Authorization		2
2.1.2 User Types		2
2.1.2.1 Admin		2
2.1.2.2 International Office		3
2.1.2.3 Student		3
2.1.2.4 Erasmus Coordinator		4
2.1.2.5 Course Coordinator		5
2.1.2.6 Chair		5
2.1.2.7 Dean/Director		5
2.1.3 Application Forms		6
2.1.4 Features		6
2.1.4.1 To Do List		6
2.1.4.2 Calendar		7
2.1.4.3 Messenger		7
2.1.4.4 Application Assistant		7
2.2 Non-Functional Requirements		8
2.2.1 Usability		8
2.2.2 Reliability		8
2.2.3 Performance		8
2.2.4 Supportability		8
2.2.5 Security		8
2.3 Pseudo Requirements		9
2.4 System Models		10
2.4.1 Use Case Model		10
2.4.2 Dynamic Models		16
2.4.2.1 State Diagrams		16
2.4.2.2 Activity State Diagram		18
2.4.2.3 Sequence Diagram		19
2.4.3 Object and Class Models		20
2.4.4 User Interface		22

1. INTRODUCTION

In this project, Bilkent University's Erasmus web page is going to be redesigned according to the requests of the Erasmus coordinators. This web application aims to make the whole application process easier for both the students and the coordinators. We are also digitalizing the Erasmus application forms in order to reduce the number of papers and working hours wasted.

In this application, students are going to be able to make and follow their applications step by step, see important deadlines and ask their questions to the coordinators through direct messages. Students are also going to be able to choose their Erasmus courses -from an approved list of courses- and they will have the ability to propose new course matchings to the course coordinators through this application. Our application assistant will guide students throughout all these steps and make recommendations & predictions based on past data. Finally, coordinators are going to be able to see applications, approve courses, message students and use their e-signature to approve documents.

2. PROPOSED SYSTEMS

2.1 Functional Requirements

2.1.1 Authentication & Authorization

Authentication and authorization functionalities will be covered by the framework that is going to be used to build this application. Authentication information of every user will be registered by the admin users, whose authentication information are embedded into the application and have access to the admin panel, according to the clients' request and will be stored in the application's designated database. Authorization functionality of this application will be governed by user types, which means that every user will be authorized according to his/her user type and that type's pre-determined authorization.

2.1.2 User Types

User is the common object that will bring in authentication and authorization functions to the project. Users will be able to have a role that will allow them to have different authorizations. Roles include 1) admin, 2) international office, 3) student, 4) erasmus coordinator, 5) course coordinator, 6) administration committee and 7) dean. One user may have many roles. For example, an Erasmus Coordinator may also be the Course Coordinator for some courses in Bilkent; or similarly, an Administration Committee member may be the Course Coordinator.

2.1.2.1 Admin

Admin is the user that has the authorization to modify the authentications and authorizations of the other users.

2.1.2.2 International Office

International Office is the role that can view student applications and post the score tables for each faculty, in order for the placements of the students to be done.

2.1.2.3 Student

Student is the user type that can apply to Erasmus and Bilateral Exchange programs. Students will be able to create an application if they satisfy the minimum requirements (minimum cGPA of 2.5 for Erasmus and 3 for Exchange, completing

two years of undergraduate study at Bilkent) for form creation (exchange type, year and semester to do the exchange), with the ability to modify his/her ongoing applications, such as adding or modifying the university preference list that consists of 5 universities at maximum. Students will also be able to see their list of applications and the status of their applications. An application has several states. The student must first choose their preferred universities. Then they must choose the courses they want to take. After this process, the application will be sent to be approved. The student can see which step of the application process they are on, and whether their application is accepted or not.

Students who were not matched with any university will be placed in the waiting list. Students that were placed into a university can cancel their applications, in which case they will be transferred to the waiting list as well.

After a student is matched with a university, they will create a pre-approval form. This pre-approval form will include a list of courses the student chose among the offerings of that particular university, the university's name, year and semester of the exchange. The pre-approval form will also have a state (academic coordinator inspection, approved and etcetera). In case of potential corner cases (a course not being offered due to prerequisite related issues and etcetera) the students will be able to create more than one pre-approval forms related to their applications. They will upload the pre-approval form document (in pdf format) to the system, later to be signed by the related coordinator.

If a course is not listed among the approved mandatory courses or if the student wants to take an elective course, he or she will be able to create a course proposal request and send it to the related instructor if it is a mandatory course, else to the coordinator.

The student will be able to check the FAQ page related to their department. If students have questions that are not written at the FAQ page, they will be able to send direct messages to the coordinator(s). The message will be listed in the coordinator's to-do list as a new item.

2.1.2.4 Erasmus Coordinator

Erasmus Coordinator is the user role that coordinates the exchange programs, matches students with the universities, nominates the students to the universities before mobility, inspect pre-approval forms and approve/decline elective courses offered by students.

Erasmus Coordinators will be able to see the applications that the students of the same department send. After the students apply to Erasmus or Bilateral Exchange programs -the deadline will be announced by the coordinator- and the international office posts the scores table for the applied students, the coordinators will be able to check the uploaded table. After that, they will match the students with universities by using a priority queue approach.

The coordinators will be able to approve or decline student's applications. If a student's CGPA drops below the required threshold before the mobility, they will get notified and they will have the opportunity to cancel an ongoing application.

The Erasmus coordinators will also be able to modify the FAQ panel of their departments and will have a to-do list. In their to-do lists; coordinators will be able to see the messages sent to them, the pre-approval forms submitted by the students, notifications about changes in student's applications and changes done in the pre-approval forms. They will also be able to reply to the direct messages.

Before the mobility period, they will be able to see the pre-approval forms the students have submitted and will be able to inspect them within the system or by downloading the documents in order to approve or decline a particular form. They will also be able to write comments based on their inspections and be able to change the status of the pre-approval form later on.

Erasmus Coordinator will be able to monitor elective course requests sent by the students and approve/decline those course requests.

They are going to be able to add new universities or modify existing ones with a university request which includes the university's name, the coordinator dealing with the particular university, its quota and some additional information (semester availability, language requirements, etcetera). They will also be able to remove a university from the system.

2.1.2.5 Course Coordinator

Course Coordinator is the user role that can evaluate course proposals of their respective courses. The instructor user will be able to see the previously accepted courses and their corresponding Bilkent course. The course coordinator user will be able to see the proposed courses and the syllabuses of the courses uploaded by the student. After evaluating the proposed course and the uploaded syllabus, the instructor user will be able to decline or accept and add the course to the previously accepted course list. The course coordinator users will also be able to manage their to do lists and get/send notifications.

2.1.2.6 Faculty Administration

Faculty Administration is the user role that can evaluate the final forms. The Faculty Administration will be able to view the applications of the students. It will also be able to accept or reject the final application of a student. Outgoing student transcripts from the partner universities will also be evaluated by the chair user. The chair users will also be able to manage their to do lists and get/send notifications.

2.1.3 Application Forms

Two different forms will exist in the Erasmus application. The first one is the university selection form. In the university selection form the student will select up to 5 partner universities and 1 semester (fall/spring) to apply. Students will finalize their selection forms and they will be matched to a university according to their Erasmus score and their university selections. If a student is matched with a university, they will move on to the second form which is the pre-approval form. In the second form, students will create a list of proposed courses. Students will be able to choose a previously accepted course from their matched university or will be able to propose a new course.

Students will create proposals by uploading the syllabi and the web pages of the courses that they intend to take. Then students will choose the equivalent Bilkent course and submit the proposals. If that proposed course's equivalent Bilkent course is a mandatory course, the system will direct the proposal to the instructor of that course. If the proposed course's equivalent Bilkent course is an elective course, the system will

direct the proposal to the Erasmus coordinator. After the proposed courses are approved, students will then prepare the second form with their approved courses and/or the previously approved courses. The second form will first be evaluated by the coordinator and then by the faculty admission board. If any changes occur, the student can resend the pre-approval form.

2.1.4 Features

To increase the website's functionality, our application will introduce additional features. In the current status of the Erasmus application process, everything revolves around emails. This can get messy and some of the e-mails can go unnoticed or be forgotten by mistake due to the increasing number of incoming emails. A to-do list which precisely shows all of the incomplete tasks will be provided to the exchange coordinators, chair and dean/director in order to prevent any chance of an Erasmus application being overlooked.

Students usually do not check the relevant information that is presented in webpages. Even if they take a look, they might forget some of the information later such as the due dates. A calendar will be provided to all students so that all students will have an easier access to the knowledge that requires careful attention. Other users will also use the calendar to keep track of the time they have left for evaluating the applications and forms.

The exchange coordinators and students agree that communication during the Erasmus program applications can get messy. Coordinators have to answer emails from possibly hundreds of students and during this process some might get lost in the commotion and go unanswered. To reduce this mess, our application will give its users the ability to send direct messages to the relevant people. This will reduce the odds that a student's message goes unanswered, and it will also reduce the delay in communication.

In order to help students to prepare more realistic university preference forms, the application assistant will sort the universities according to the likelihood of the student being able to go to that university based on his/her Erasmus score. Furthermore, the

application assistant will also help students with their course selections considering the courses that they have already taken and the courses offered by the host university.

2.2 Non-Functional Requirements

2.2.1 Usability

- The website should be easily navigatable. Any user who reviewed the mockups of the website should be able to understand the website routing design and navigate through the pages without trouble.
- The calendar and to-do list features should allow users to go about their work without the need of reading any guidelines.
- Students should be able to prepare their course selection forms without leaving the website.

2.2.2 Reliability

- A user's application should not be lost due to any problems.
- The website should have appropriate error messages that show the user what the issue is.

2.2.3 Performance

- The operations a user can perform should be shown clearly. The results of these operations should be shown to the user in less than 3 seconds.
- The load times should be kept below 2 seconds.

2.2.4 Supportability

- The application should support Google Chrome version 107 and above, and Firefox version 102 and above.
- The website design should be consistent across different devices and browsers.

2.2.5 Security

- User information should not be visible to any user that isn't required to see.
- The website should be protected against any data breaches. In order to achieve this, the website should use HTTPS protocol, The data access should be only provided

to the users with permission and the passwords of the users should be stored in an encrypted/hashed form.

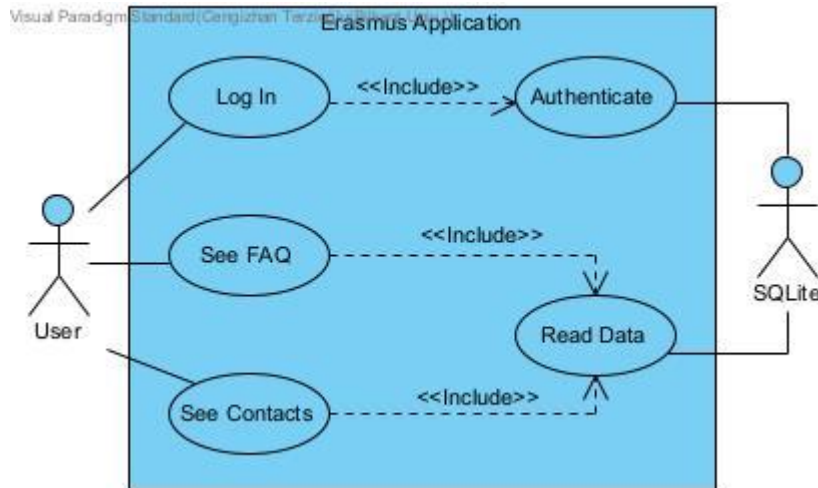
2.3 Pseudo Requirements

- The application must be a web application.
- The language that will be used for the backend must support object-oriented programming.
- Different types of users must exist with different responsibilities.
- The application must reduce paper usage.
- Students must be able to see the status of their application.
- Coordinators must be able to see a to-do list for waiting items.
- The workload between coordinators must be distributed.
- All approved/rejected courses must be logged and that info must be available to students and coordinators.
- Faculty Administration Board decisions must be logged.
- Changes to pre-application and changes/amendments to the course transfer must be supported.
- All forms must be kept in pdf format and query support must exist.

2.4 System Models

The use cases were created from the information from the Can Alkan's presentation and the feedback from the students which applied to Erasmus.

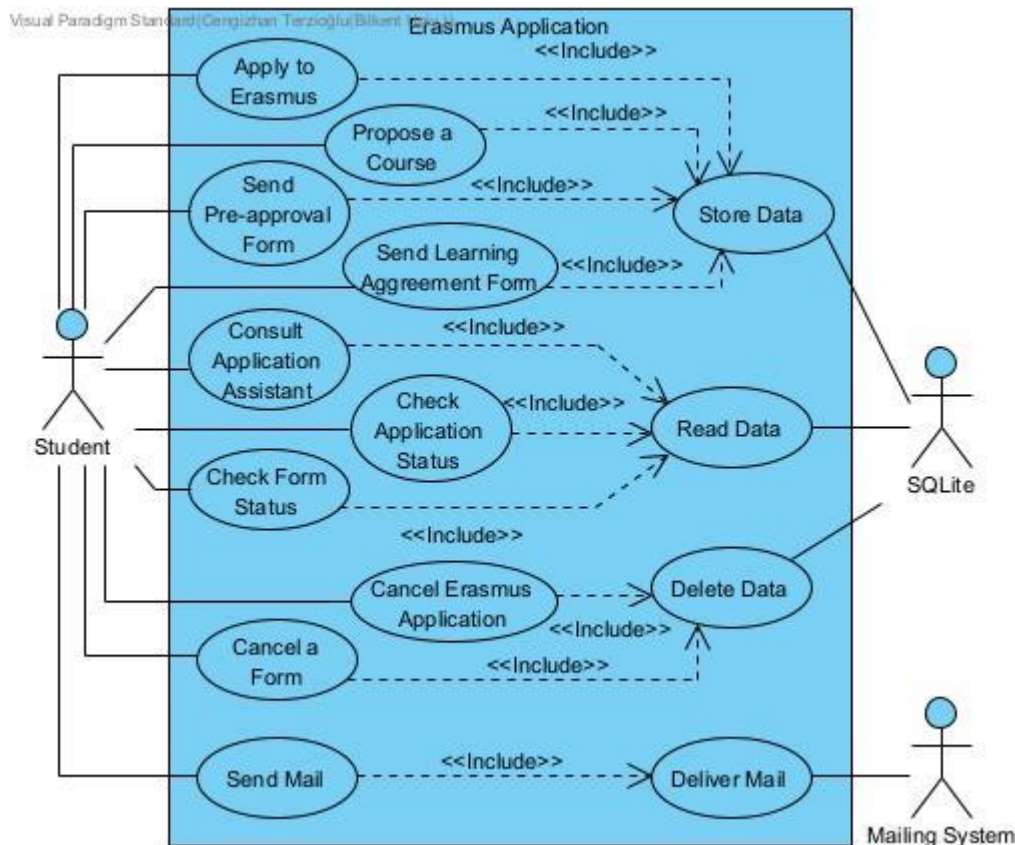
2.4.1 Use Case Model



<i>Use Case:</i>	Log In
<i>Participating Actor:</i>	User
<i>Entry Condition:</i>	User must have an account in the system.
<i>Exit Conditions:</i>	User is logged into their account.
<i>Flow of Events:</i>	1. User enters their Bilkent ID and password. 2. Authentication is made. 3. The user is directed to the related home page after their role is checked.
<i>Quality Requirements:</i>	None

<i>Use Case:</i>	See FAQ
<i>Participating Actor:</i>	User
<i>Entry Condition:</i>	None
<i>Exit Conditions:</i>	The user sees FAQ.
<i>Flow of Events:</i>	1. The user clicks on the “FAQ” button on the navigation bar.. 2. FAQ list is opened.
<i>Quality Requirements:</i>	None

<i>Use Case:</i>	See Contacts
<i>Participating Actor:</i>	User
<i>Entry Condition:</i>	None
<i>Exit Conditions:</i>	The user sees who they can reach about erasmus related issues.
<i>Flow of Events:</i>	1. The user clicks on the “Contacts” button on the navigation bar.. 2. Contacts list is opened.
<i>Quality Requirements:</i>	None



Use Case: Apply to Erasmus
Participating Actor: Student
Entry Condition: The student must not have an already existing erasmus application.
Exit Condition: The user has started the Erasmus application process.
Flow of Events: 1. The student applies to the Erasmus program.
 2. The student opens the list of available universities for their department.
 3. The student selects 5 universities they want in an order.
 4. The student uploads the Erasmus application form.
Quality Requirements: The Erasmus application form must be in pdf format.

Use Case: Propose a course
Participating Actor: Student
Entry Condition: The student must have applied to Erasmus.
Exit Condition: The student proposes a course to be approved.
Flow of Events: 1. The student selects whether the course is mandatory or elective.
 2. If the course is mandatory, the student selects a mandatory course from Bilkent to be matched by the course they are proposing.
 3. The student uploads the syllabus of the course and sends a link to that syllabus.
Quality Requirements: None

<i>Use Case:</i>	Send Pre-approval Form
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must have at most 1 existing pre-approval form.
<i>Exit Condition:</i>	The student uploads their pre-approval form to the system.
<i>Flow of Events:</i>	1. The student opens a list of all the courses that are approved in previous years. 2. The student selects the courses they want to take at the university they are going. 3. The student finalizes and uploads their pre-approval form.
<i>Quality Requirements:</i>	The pre-approval form must be in pdf format.
<i>Use Case:</i>	Send Learning Agreement Form
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must have at most 1 existing learning agreement form.
<i>Exit Condition:</i>	The student uploads their learning agreement form to the system.
<i>Flow of Events:</i>	1. The student fills the learning agreement form. 2. The student uploads their learning agreement form.
<i>Quality Requirements:</i>	The learning agreement form must be in pdf format.
<i>Use Case:</i>	Consult Application Assistant
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must not have an already existing erasmus application.
<i>Exit Condition:</i>	The student has a list of universities they are likely to go according to their Erasmus point.
<i>Flow of Events:</i>	1. The student opens the application assistant. 2. The student sees a list of universities that other students with close Erasmus scores have entered.
<i>Quality Requirements:</i>	None
<i>Use Case:</i>	Check Application Status
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must already have an existing Erasmus application.
<i>Exit Condition:</i>	The user accesses the status of their Erasmus application.
<i>Flow of Events:</i>	1. The student sees the step they are in for Erasmus application.
<i>Quality Requirements:</i>	None
<i>Use Case:</i>	Check Form Status
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must already have an existing form.
<i>Exit Condition:</i>	The student accesses the status of their form.
<i>Flow of Events:</i>	1. The student selects a form they have uploaded. 2. The student sees their form's application status.
<i>Quality Requirements:</i>	None
<i>Use Case:</i>	Cancel Erasmus Application
<i>Participating Actor:</i>	Student
<i>Entry Condition:</i>	The student must already have an existing Erasmus application.

Exit Condition: The student stops their Erasmus application process.

Flow of Events: 1. The student cancels their Erasmus application.

2. All the forms the student sent are removed from the database.

Quality Requirements: None

Use Case: Cancel Form

Participating Actor: Student

Entry Condition: The student must already have an existing form.

Exit Condition: The student deletes their form

Flow of Events: 1. The student selects the form they want to delete.

2. The student deletes the chosen form.

3. The form is removed from the database.

Quality Requirements: None

Use Case: Send Mail

Participating Actor: Student

Entry Condition: None

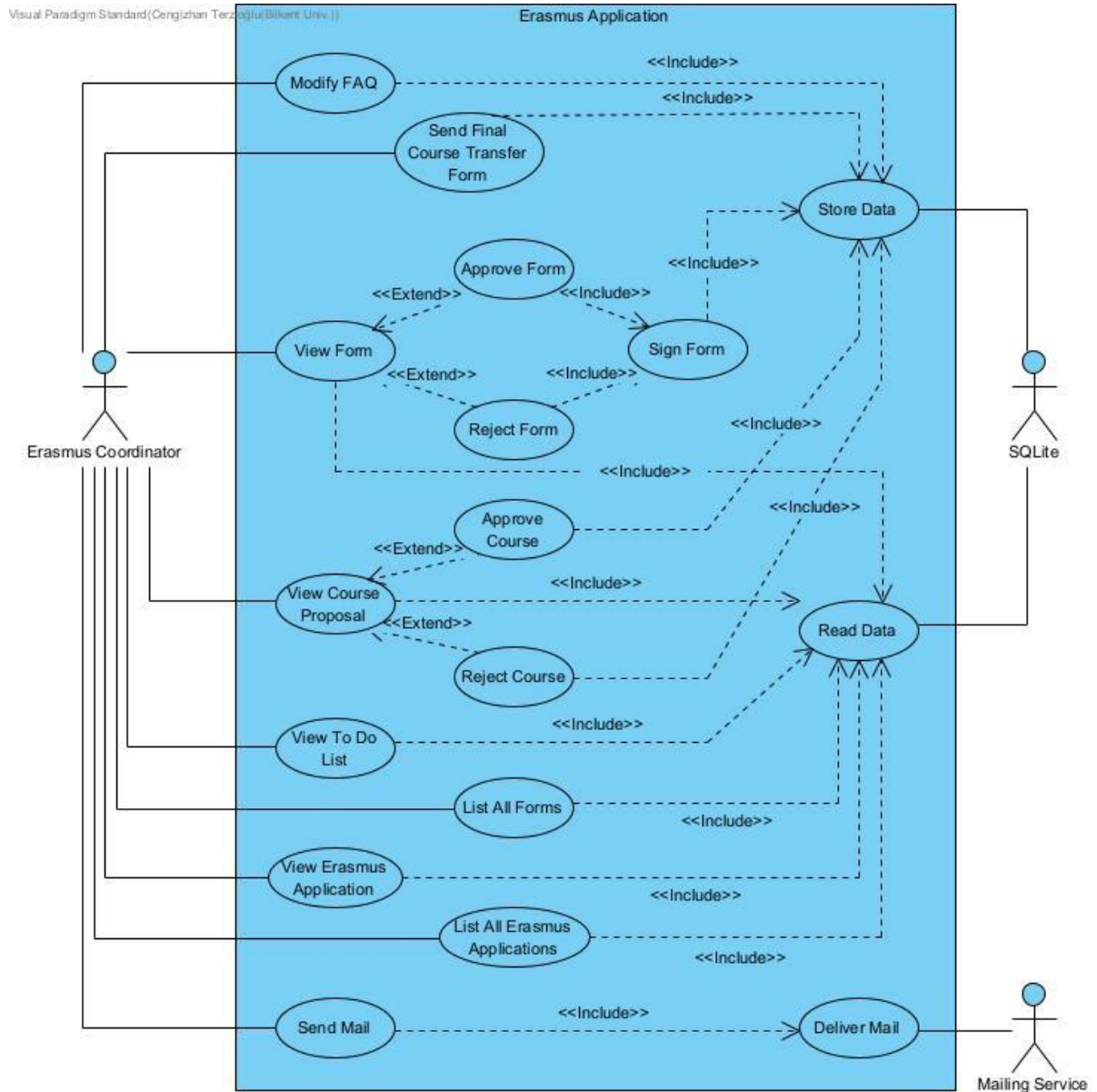
Exit Condition: The student sends a mail to a user who is in their contacts.

Flow of Events: 1. The student selects who they want to send a mail.

2. The student writes their mail.

3. The student sends their mail to the chosen user.

Quality Requirements: None



<i>Use Case:</i>	Modify FAQ
<i>Participating Actor:</i>	Erasmus Coordinator
<i>Entry Condition:</i>	None
<i>Exit Condition:</i>	FAQ is updated.
<i>Flow of Events:</i>	<ol style="list-style-type: none"> 1. Exchange coordinator opens the menu for modifying the FAQ. 2. Exchange coordinator adds new questions and answers to the FAQ. 3. Exchange coordinator updates already existing items.
<i>Quality Requirements:</i>	None

<i>Use Case:</i>	Send Final Course Transfer Form
<i>Participating Actor:</i>	Erasmus Coordinator
<i>Entry Condition:</i>	The official transcript of the student must be sent to the erasmus coordinator from the international office.

Exit Condition: The erasmus coordinator sends the final course transfer form of a student to the faculty administration committee.

Flow of Events: 1. The erasmus coordinator prepares a final course transfer form for a student.

2. The erasmus coordinator uploads the final course transfer form.

Quality Requirements: The final course transfer form must be in pdf format.

Use Case: View Form

Participating Actor: Erasmus Coordinator

Entry Condition: There must be at least one form the erasmus coordinator can see.

Exit Conditions: The erasmus coordinator sees the form.

Flow of Events: 1. The erasmus coordinator selects a form.
2. The erasmus coordinator opens the form they selected.

Quality Requirements: The form must be in pdf format.

Use Case: Approve Form

Participating Actor: Erasmus Coordinator

Entry Condition: The erasmus coordinator has opened a form.

Exit Conditions: The form is approved and its status is updated accordingly.

Flow of Events: 1. The erasmus coordinator clicks the “Approve” button.
2. Status of the form is updated to “Approved”.
3. System requests the signature of the erasmus coordinator.

Quality Requirements: None

Use Case: Reject Form

Participating Actor: Erasmus Coordinator

Entry Condition: The erasmus coordinator has opened a form.

Exit Conditions: The form is declined and its status is updated accordingly.

Flow of Events: 1. The erasmus coordinator clicks the “Reject” button.
2. Status of the form is updated to “Rejected”.
3. System requests the signature of the erasmus coordinator.

Quality Requirements: None

Use Case: Sign Form

Participating Actor: Erasmus Coordinator

Entry Condition: The erasmus coordinator clicks either to the “Approve” or to the “Reject” button.

Exit Conditions: The form is signed. A notification is sent to the student in order to report the status of the form.

Flow of Events: 1. The erasmus coordinator signs the form.
2. A notification is sent to the student in order to report the status of the form.

Quality Requirements: None

Use Case: View Course Proposal

Participating Actor: Erasmus Coordinator
Entry Condition: Proposed course must be elective.
Exit Conditions: The erasmus coordinator sees the elective course proposal.
Flow of Events: 1. The erasmus coordinator selects an elective course.
 2. The erasmus coordinator opens the syllabus of the course they selected.
Quality Requirements: None

Use Case: Approve Course
Participating Actor: Erasmus Coordinator
Entry Condition: The erasmus coordinator has opened an elective course proposal.
Exit Conditions: The course is approved and its status is updated accordingly.
Flow of Events: 1. The erasmus coordinator clicks the “Approve” button.
 2. Status of the elective course is updated to “Approved”.
Quality Requirements: None

Use Case: Reject Course
Participating Actor: Erasmus Coordinator
Entry Condition: The erasmus coordinator has opened an elective course proposal.
Exit Conditions: The course is rejected and its status is updated accordingly.
Flow of Events: 1. The erasmus coordinator clicks the “Reject” button.
 2. Status of the elective course is updated to “Rejected”.
Quality Requirements: None

Use Case: View To Do List
Participating Actor: Erasmus Coordinator
Entry Condition: None
Exit Condition: The erasmus coordinator has the list of their tasks.
Flow of Events: 1. The erasmus coordinator clicks to view their to do list items.
Quality Requirements: None

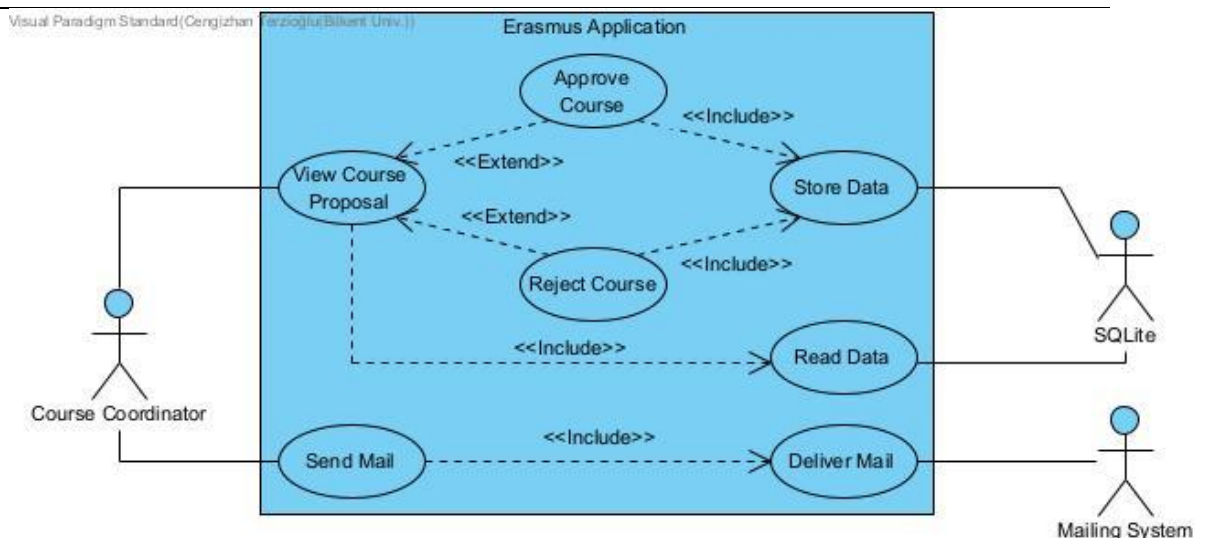
Use Case: List All Forms
Participating Actor: Erasmus Coordinator
Entry Condition: None
Exit Condition: The erasmus coordinator has the list of all forms within specified filtering conditions.
Flow of Events: 1. The erasmus coordinator selects conditions for filtering.
 2. All of the forms that satisfy the conditions are listed.
Quality Requirements: Erasmus coordinator cannot list the forms of other departments.

Use Case: View Erasmus Application
Participating Actor: Erasmus Coordinator

<i>Entry Condition:</i>	There must be at least one student who has applied to the Erasmus program.
<i>Exit Condition:</i>	Erasmus coordinator sees the copy of the application made by the selected student.
<i>Flow of Events:</i>	1. Erasmus coordinator selects the application of a student.
<i>Quality Requirements:</i>	None

<i>Use Case:</i>	List All Erasmus Applications
<i>Participating Actor:</i>	Erasmus Coordinator
<i>Entry Condition:</i>	None
<i>Exit Condition:</i>	The erasmus coordinator has the list of all erasmus applications within specified filtering conditions.
<i>Flow of Events:</i>	1. The erasmus coordinator selects conditions for filtering. 2. All of the erasmus applications that satisfy the conditions are listed.
<i>Quality Requirements:</i>	Erasmus coordinator cannot list the erasmus applications of other departments.

<i>Use Case:</i>	Send Mail
<i>Participating Actor:</i>	Erasmus Coordinator
<i>Entry Condition:</i>	None
<i>Exit Condition:</i>	The erasmus coordinator sends a mail to a user who is in their contacts.
<i>Flow of Events:</i>	1. The erasmus coordinator selects who they want to send a mail. 2. The erasmus coordinator writes their mail. 3. The Erasmus coordinator sends their mail to the chosen user.
<i>Quality Requirements:</i>	None



<i>Use Case:</i>	View Course Proposal
<i>Participating Actor:</i>	Course Coordinator
<i>Entry Condition:</i>	Proposed course must be mandatory.
<i>Exit Conditions:</i>	The course coordinator sees the mandatory course proposal.
<i>Flow of Events:</i>	1. The course coordinator selects a mandatory course.

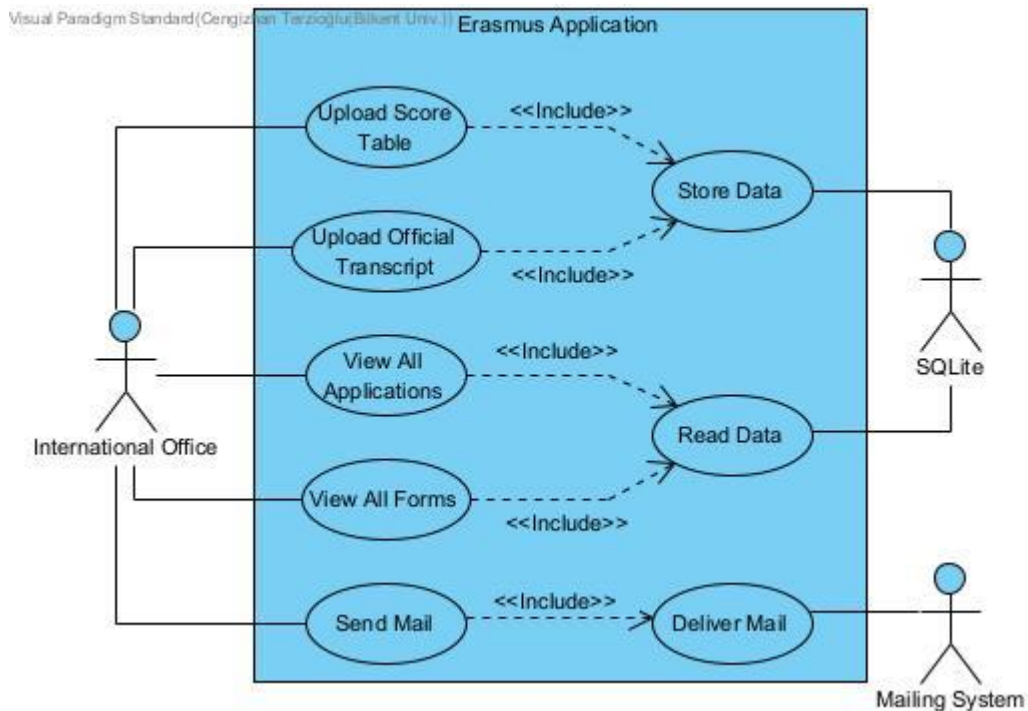
2. The course coordinator opens the syllabus of the course they selected.

Quality Requirements: Course coordinator cannot see the mandatory courses given by other course coordinators.

Use Case: Approve Course
Participating Actor: Course Coordinator
Entry Condition: The course coordinator has opened a mandatory course proposal.
Exit Conditions: The course is approved and its status is updated accordingly.
Flow of Events: 1. The course coordinator clicks the “Approve” button.
2. Status of the mandatory course is updated to “Approved”.
Quality Requirements: None

Use Case: Reject Course
Participating Actor: Course Coordinator
Entry Condition: The course coordinator has opened a mandatory course proposal.
Exit Conditions: The course is rejected and its status is updated accordingly.
Flow of Events: 1. The course coordinator clicks the “Reject” button.
2. Status of the mandatory course is updated to “Rejected”.
Quality Requirements: None

Use Case: Send Mail
Participating Actor: Course Coordinator
Entry Condition: None
Exit Condition: The course coordinator sends a mail to a user who is in their contacts.
Flow of Events: 1. The course coordinator selects who they want to send a mail.
2. The course coordinator writes their mail.
3. The course coordinator sends their mail to the chosen user.
Quality Requirements: None



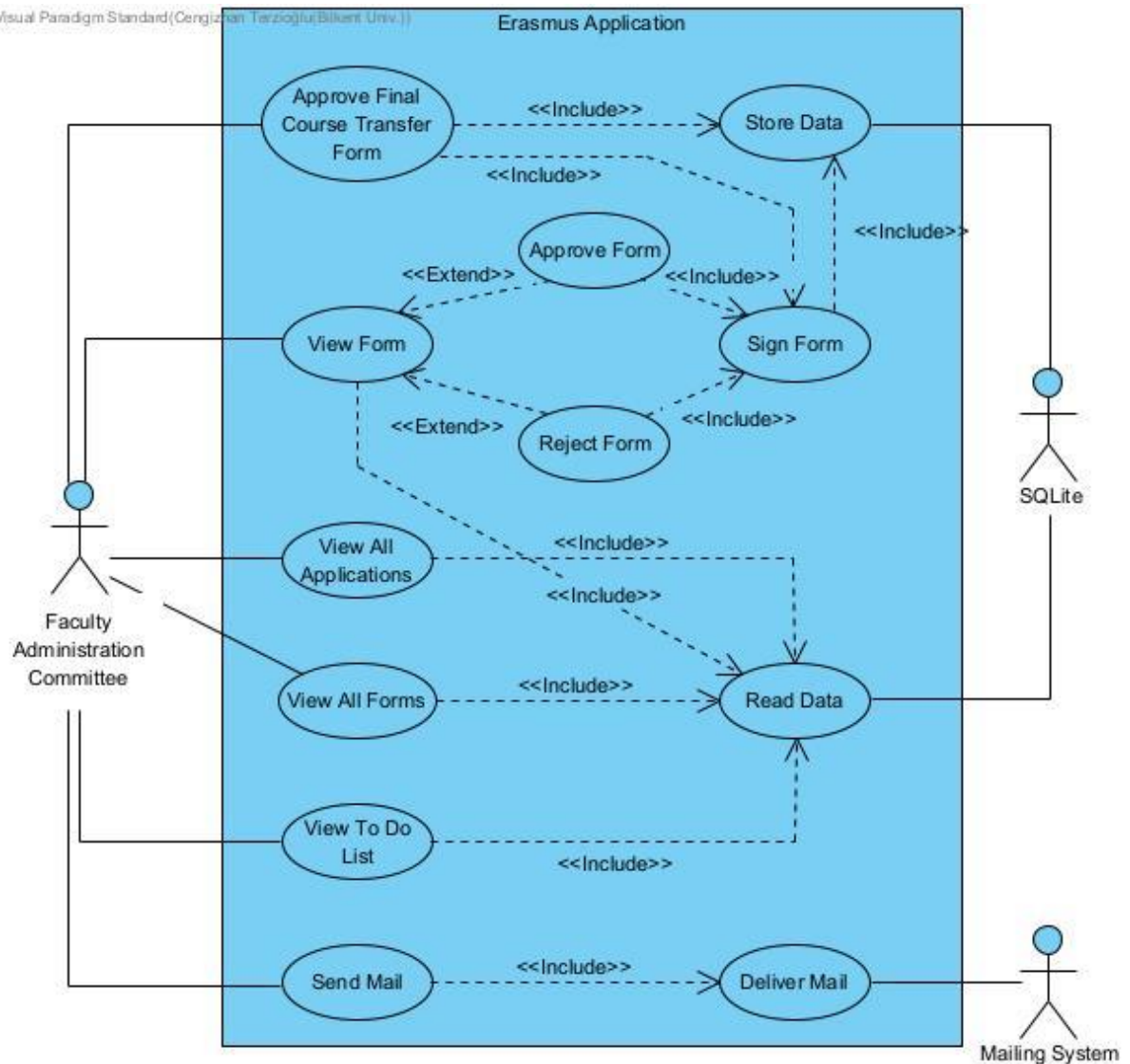
Use Case:	Upload Score Table
Participating Actor:	International Office
Entry Condition:	Erasmus applications are closed.
Exit Condition:	Erasmus score table is uploaded and placement to universities can be done.
Flow of Events:	1. The international office uploads the erasmus score table
Quality Requirements:	Row number of erasmus score table must be equal to the number of students who applied to erasmus.

Use Case:	Upload Official Transcript
Participating Actor:	International Office
Entry Condition:	International office has received the transcript of a student from the host university
Exit Condition:	Official transcript of a student that is sent by the host university is uploaded to the system.
Flow of Events:	1. The international office uploads the transcript of a student that is sent by the host university.
Quality Requirements:	Transcript must be in pdf format.

Use Case:	View All Forms
Participating Actor:	International Office
Entry Condition:	None
Exit Condition:	The international office has the list of all forms.
Flow of Events:	1. The international office selects conditions for filtering. 2. All of the forms that satisfy the conditions are listed.
Quality Requirements:	None

Use Case:	View All Applications
Participating Actor:	International Office
Entry Condition:	None
Exit Condition:	The international office has the list of all applications.
Flow of Events:	<ol style="list-style-type: none"> 1. The international office selects conditions for filtering. 2. All of the applications that satisfy the conditions are listed.
Quality Requirements:	None

Use Case:	Send Mail
Participating Actor:	International Office
Entry Condition:	None
Exit Condition:	The international office sends a mail to a user who is in their contacts.
Flow of Events:	<ol style="list-style-type: none"> 1. The international office selects who they want to send a mail. 2. The international office writes their mail. 3. The international office sends their mail to the chosen user.
Quality Requirements:	None



Use Case: Approve Final Course Transfer Form
Participating Actor: Faculty Administration Committee
Entry Condition: Erasmus coordinator has uploaded the final course transfer form.
Exit Conditions: The final course transfer form is approved and its status is updated accordingly.

Flow of Events:

1. The faculty administration committee opens the final course transfer form.
2. The faculty administration committee clicks the “Approve” button.
3. Status of the form is updated to “Approved”.
4. System requests the signature of the faculty administration committee.

Quality Requirements: The final course transfer form must be in pdf format.

Use Case: View Form
Participating Actor: Faculty Administration Committee
Entry Condition: There must be at least one form the faculty administration committee can see.
Exit Conditions: The faculty administration committee sees the form.
Flow of Events:

1. The faculty administration committee selects a form.

2. The faculty administration committee opens the form they selected.

Quality Requirements: The form must be in pdf format.

<i>Use Case:</i>	Approve Form
<i>Participating Actor:</i>	Faculty Administration Committee
<i>Entry Condition:</i>	The faculty administration committee has opened a form.
<i>Exit Conditions:</i>	The form is approved and its status is updated accordingly.
<i>Flow of Events:</i>	<ol style="list-style-type: none">1. The faculty administration committee clicks the “Approve” button.2. Status of the form is updated to “Approved”.3. System requests the signature of the faculty administration committee.

Quality Requirements: None

<i>Use Case:</i>	Reject Form
<i>Participating Actor:</i>	Faculty Administration Committee
<i>Entry Condition:</i>	The faculty administration committee has opened a form.
<i>Exit Conditions:</i>	The form is declined and its status is updated accordingly.
<i>Flow of Events:</i>	<ol style="list-style-type: none">1. The faculty administration committee clicks the “Reject” button.2. Status of the form is updated to “Rejected”.3. System requests the signature of the faculty administration committee.

Quality Requirements: None

<i>Use Case:</i>	Sign Form
<i>Participating Actor:</i>	Faculty Administration Committee
<i>Entry Condition:</i>	The faculty administration committee clicks either to the “Approve” or to the “Reject” button.
<i>Exit Conditions:</i>	The form is signed. A notification is sent to the student in order to report the status of the form.
<i>Flow of Events:</i>	<ol style="list-style-type: none">1. The faculty administration committee signs the form.2. A notification is sent to the student in order to report the status of the form.

Quality Requirements: None

<i>Use Case:</i>	View All Applications
<i>Participating Actor:</i>	Faculty Administration Committee
<i>Entry Condition:</i>	None
<i>Exit Condition:</i>	The faculty administration committee has the list of all applications.
<i>Flow of Events:</i>	<ol style="list-style-type: none">1. The faculty administration committee selects conditions for filtering.

2. All of the applications that satisfy the conditions are listed.

Quality Requirements: Faculty administration committee cannot list the applications of other faculties.

Use Case: View All Forms
Participating Actor: Faculty Administration Committee
Entry Condition: None
Exit Condition: The faculty administration committee has the list of all forms within specified filtering conditions.
Flow of Events: 1. The faculty administration committee selects conditions for filtering.

2. All of the forms that satisfy the conditions are listed.

Quality Requirements: Faculty administration committee cannot list the forms of other faculties.

Use Case: View To Do List
Participating Actor: Faculty Administration Committee
Entry Condition: None
Exit Condition: The faculty administration committee has the list of their tasks.
Flow of Events: 1. The faculty administration committee clicks to view their to do list items.
Quality Requirements: None

Use Case: Send Mail
Participating Actor: Faculty Administration Committee
Entry Condition: None
Exit Condition: The faculty administration committee sends a mail to a user who is in their contacts.
Flow of Events: 1. The faculty administration committee selects who they want to send a mail.
2. The faculty administration committee writes their mail.

3. The faculty administration committee sends their mail to the chosen user.

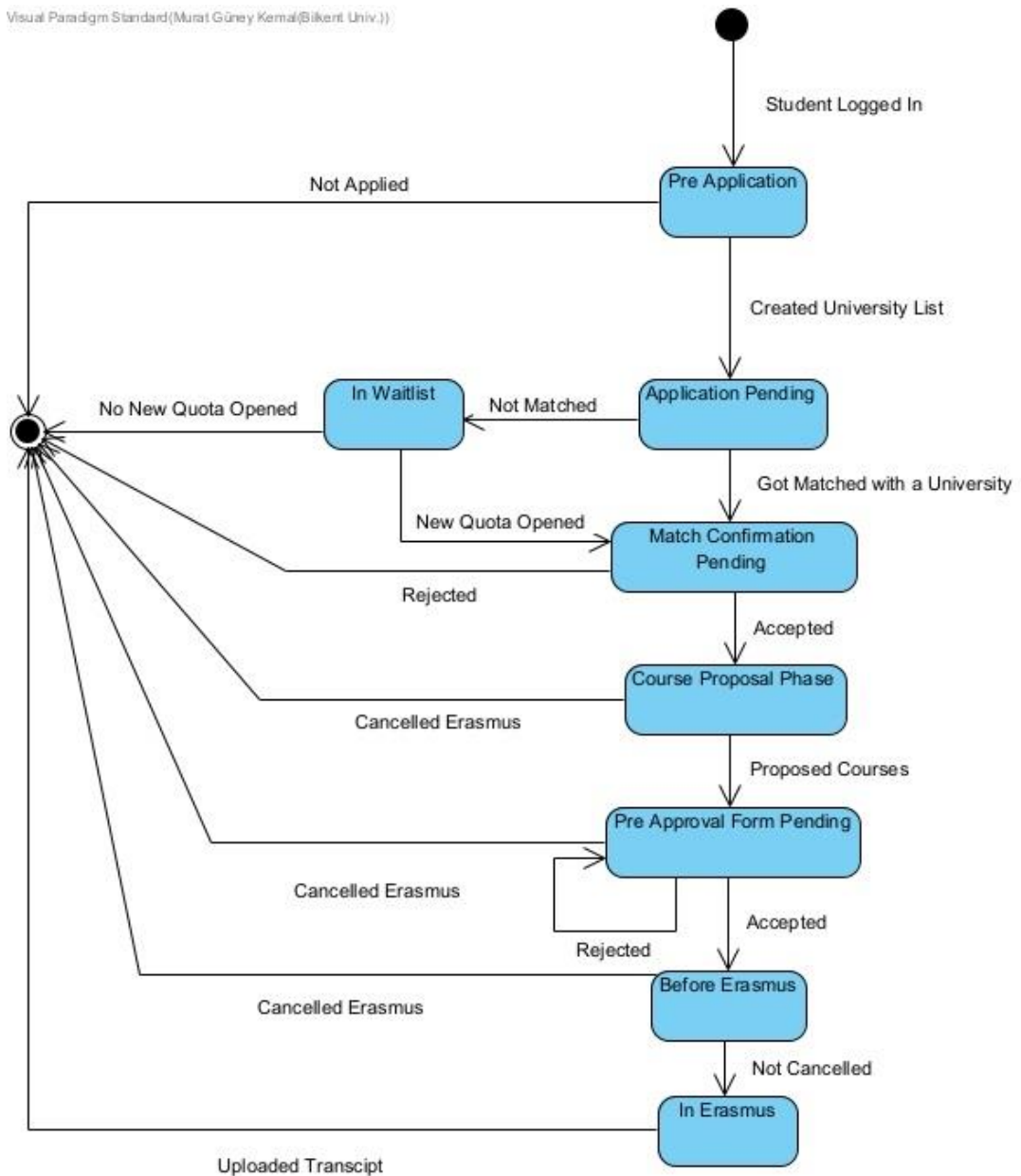
Quality Requirements: None

2.4.2 Dynamic Models

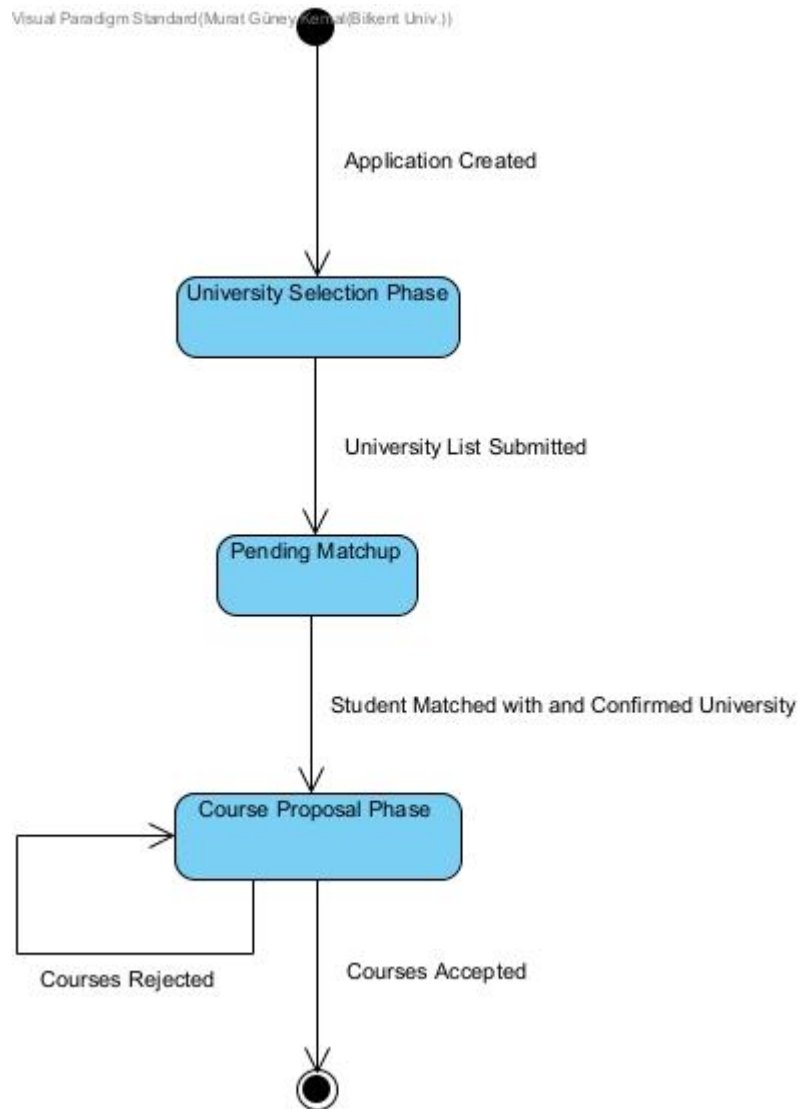
2.4.2.1 State Diagrams

2.4.2.1.1 Student State Diagram

Visual Paradigm Standard (Murat Güneş Kemal (Bilkent Univ.))

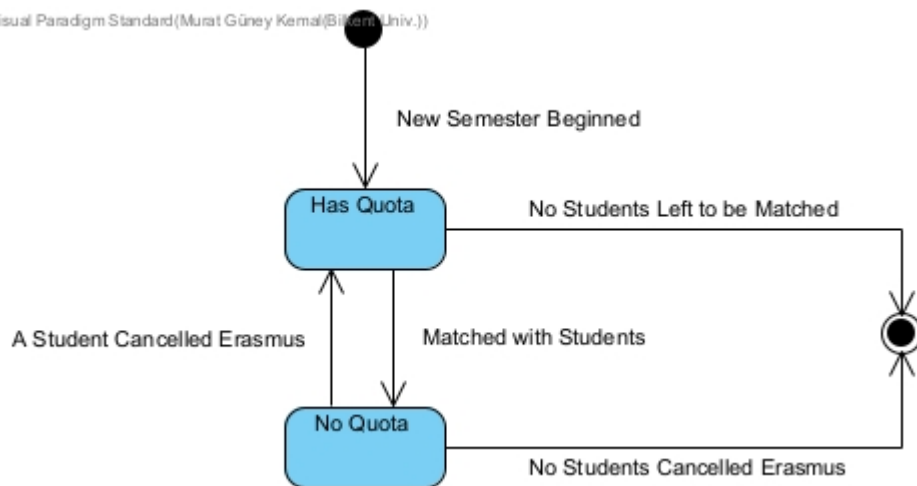


2.4.2.1.2 Application State Diagram



2.4.2.1.3 University State Diagram

Visual Paradigm Standard (Murat Güney Kemal(Bilkent Univ.))



2.4.2.2 Activity Diagram

Visual Paradigm Standard (Murat Güney Kemal (Bilkent Univ.))

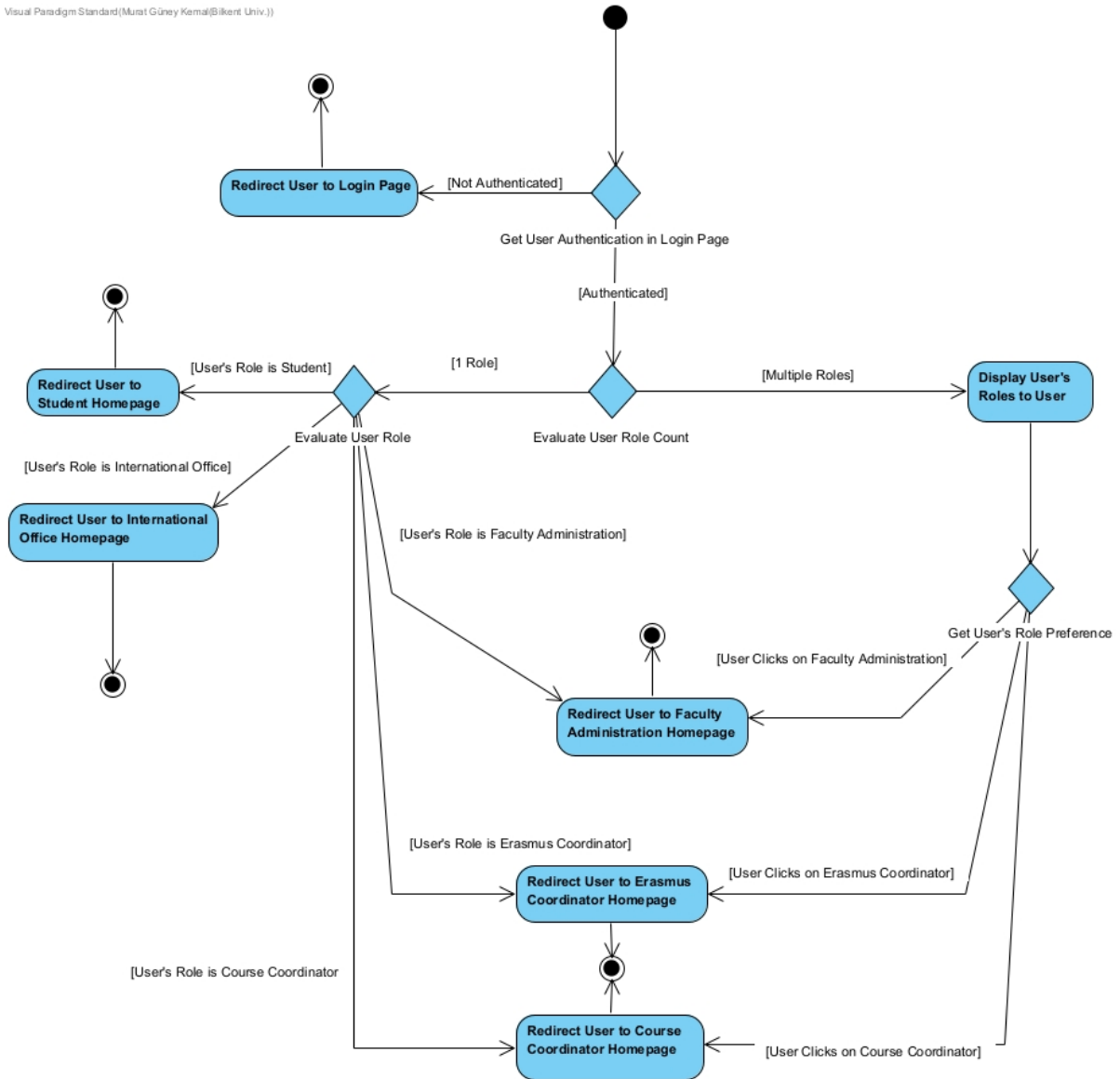


Figure 6: Activity State Diagram

2.4.2.3 Sequence Diagram

Scenario: Start of the preapproval phase to preapproval creation.

The erasmus applications are closed and the preapproval phase begins. International office posts the score table and the erasmus coordinators of each department match the students with the universities automatically using the website. Students can cancel their application or they can create preapproval form.

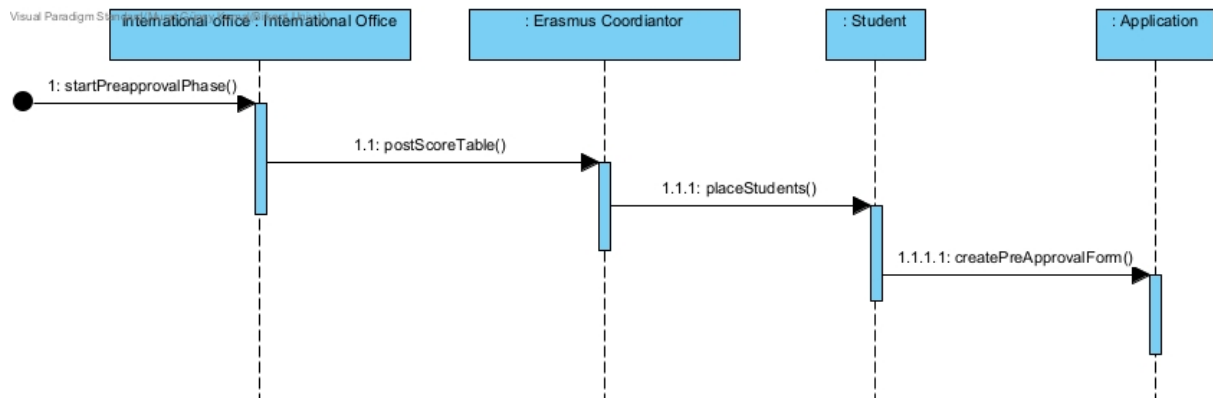


Figure 7: Continue Application Sequence Diagram

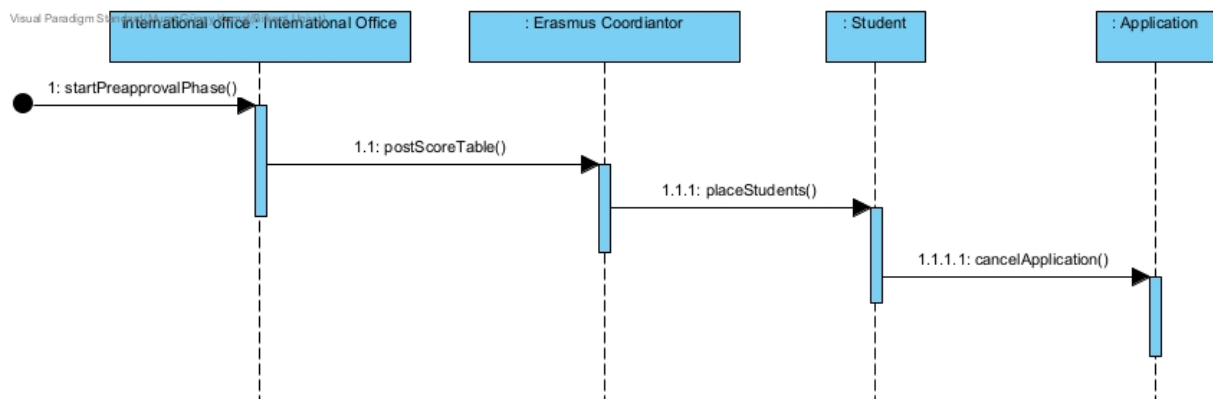
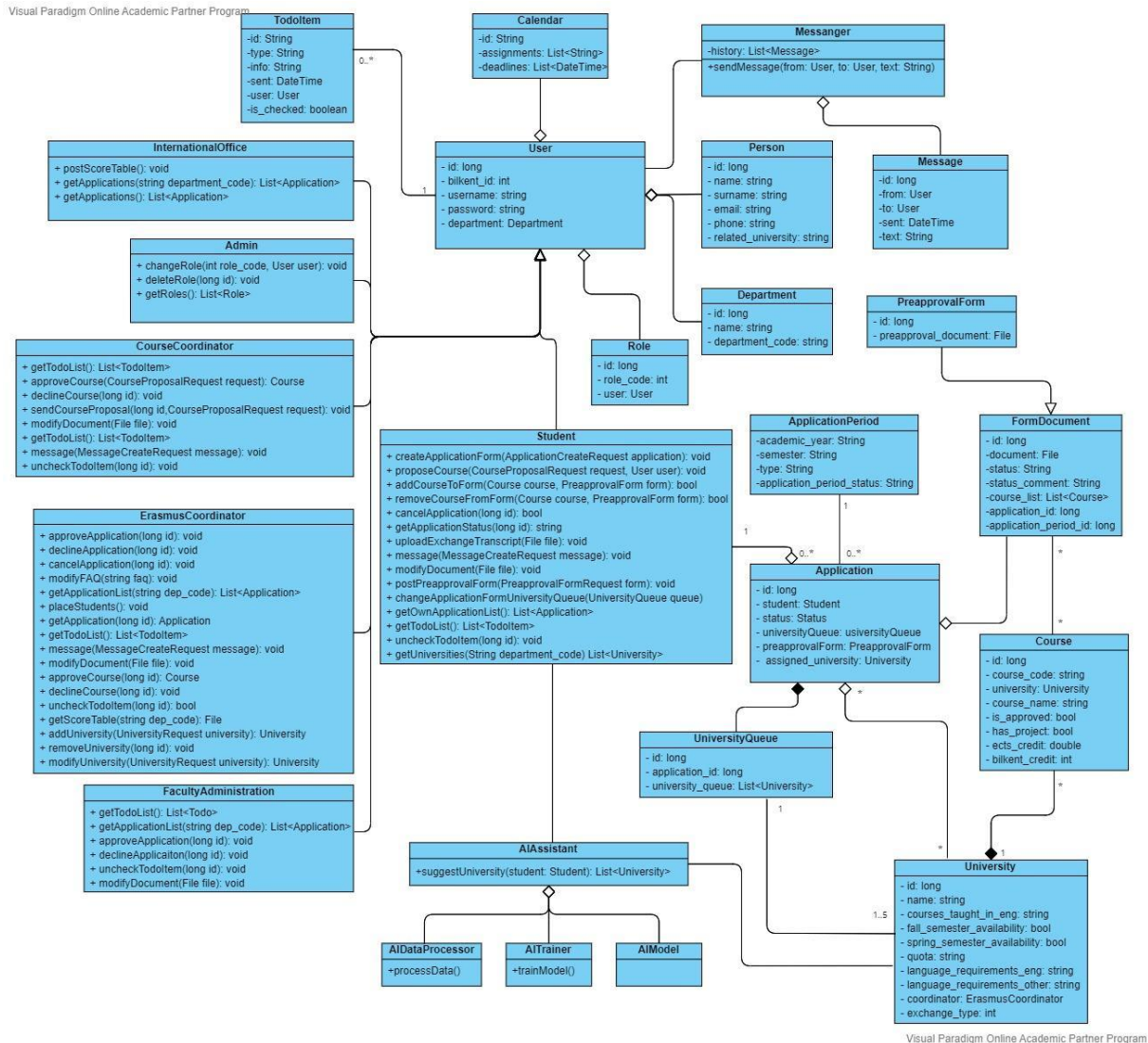


Figure 8: Cancel Application Sequence Diagram

2.4.3 Object and Class Model



The class diagram of the project is present above. The diagram has 25 classes in total, and it shows the relationship between the class objects.

User: This is the main class behind authentication purposes, personal data storage, department and role management. A User may have a Department, Calendar, Person. A User can interact with TodoItems and can use Messenger for messaging purposes.

Role: This is the main class for authorization purposes and is a superclass of Student, ErasmusCoordinator, CourseCoordinator, Chair, Dean, InternationalOffice and Admin. It binds users with certain roles.

Student: This is a subclass of Role and represents the student stakeholder. A Student can have many Applications.

ErasmusCoordinator: This is a subclass of Role and represents the Erasmus/Bilateral Exchange coordinator stakeholder. An ErasmusCoordinator can interact with Application, University, Course and FormDocument.

CourseCoordinator: This is a subclass of Role and represents the course coordinator/instructor stakeholder. A CourseCoordinator can interact with Course.

FacultyAdministration: This is a subclass of Role and represents the administration board stakeholder. A FacultyAdministration can interact with Application and FormDocument.

InternationalOffice: This is a subclass of Role and represents the student international office stakeholder. It can interact with Application.

Admin: This is a subclass of Role. An Admin has the ability to manipulate Role object belonging to other Users.

ToDoItem: This is a class used for the to-do feature in the application. A User can have many ToDoItems.

Calendar: This is a class used for the calendar feature in the application. A User can have one Calendar.

Messenger: This is a class used for the messaging feature in the application. A User can interact with Messenger. A Messenger can have many Messages and depends on User.

Message: This is a class used by Messenger for the messaging feature in the application.

Person: This is a class used by User to represent personal information of the user. A User can have a Person.

Department: This is a class used by User to represent a user's department at Bilkent. A User can have a Department.

Application: This is the main class used for Erasmus/Bilateral Exchange student applications. An Application can exist in an ApplicationPeriod. An Application can have a UniversityQueue, a University and multiple FormDocuments.

ApplicationPeriod: This is the class that holds Applications to act as a filtering mechanism. An ApplicationPeriod can have many Applications.

UniversityQueue: This is the class that holds a list of Universities to act as a student's list of universities to go during the exchange. A UniversityQueue can have up to 5 Universities.

University: This is the class that holds the exchange university information. A University can have multiple Courses.

Course: This is the class that holds course information belonging to an exchange university as well as its equivalent course(s) at Bilkent. A Course can be present on multiple FormDocuments.

FormDocument: This is a superclass that holds a document's information, which is related to the exchange program. A FormDocument can include many Courses.

PreapprovalForm: This is a subclass of FormDocument and specifically holds the necessary file for an application's pre-approval stage.

AIAssistant: This is a class for the assistant feature in the application. AIAssistant can interact with the students to help them select five universities that give them the highest chance of mobility based on their department and Erasmus score. AIAssistant has an AIDataProcessor, an AITrainer and an AIModel.


AIDataProcessor: This is the class for processing the past student mobility placement data so that it can be fed into the machine learning model.


AITrainer: This is the class for training the machine learning model to make accurate University recommendations to Students.

AIModel: This class will contain the chosen scikit-learn classification model that is going to take the processed input data and produce the output of University selection recommendation.


2.4.4 User Interface

Pages Before User Login

[Home](#)[FAQ](#)[Contacts](#)[Login](#)



Bilkent Erasmus



Get Started

Bilkent Computer Engineering Department has been actively involved in exchange programs and has established bilateral agreements with some of the leading universities of the world. Students who are interested in studying abroad are encouraged to apply.

4000+

STUDENT

5+





COUNTRY


20+

PARTNER UNIVERSITY

Partner Institutions

The exchange partners of the Bilkent CS Department include some of the best computer science/engineering departments in the world. All of our exchange partners are among the most established universities of their countries, and most of them have been listed among the World's Top 200 Universities according to the Times Higher Education Supplement. The following is a list of our Erasmus partners:






A Little Bit of AI Touch

We are adding application assistant to help you, do your applications easier and better


[Learn More](#)

Progress Bar


After application you can check every step to not miss any important dead lines!




Application Submitted
(12-04-2022)




Preliminary Stage Passed
(15-04-2022)




Preliminary Stage
(18-05-2022)







Finalization of Chosen Universities
(24-07-2022)



Course Selection
(11-08-2022)



Copyright © All Rights Reserved for Cem Hoca.



[Home](#)[FAQ](#)[Contacts](#)[Login](#)

Frequently Asked Questions



Q.1. What is the student exchange program?

[×](#)

An important component of university education is to develop a better understanding of the world, not just academically but also socially. Student exchange is a means to achieve this goal by enabling students study at an environment different from their home country and institution.

Q.2. What are the financial issues involved for an exchange student?

[+](#)

Q.3. Is housing provided to exchange students at the host universities?

[+](#)

Q.4. Who can apply for the exchange program?

[+](#)

Q.5. Can't I go to exchange on the 4th year?

[+](#)

Q.6. How can I apply for the exchange program?

[+](#)

Q.7. Is there a GPA requirement for exchange applications?

[+](#)[Home](#)[FAQ](#)[Contacts](#)[Login](#)

Contact

Department Coordinators

Can Alkan <http://cs.bilkent.edu.tr/~calkan/>
Ayşegül Dündar <http://www.cs.bilkent.edu.tr/~adundar/>

Administrative Coordinators

Deniz Dicle Arman
Erkin Tarhan terkin@bilkent.edu.tr

Erasmus Students Counselors

Elif Ünsal elif.unsal@bilkent.edu.tr
Buket HATAM buketg@bilkent.edu.tr



Login


BILKENT ID

PASSWORD

[FORGOT YOUR PASSWORD?](#)

[LOGIN](#)

Student Pages



HomeMy ApplicationFAQContactsLogout

Hello Onur!

Apply Erasmus

Press Start Application to apply for erasmus or exchange for next fall or spring semester

[Start Application](#)

Course Selection

Course selection period has not started yet. Students must be matched with a university to advance to this phase.

Application Assistant

Application Assistant advises universities to students based on their erasmus/exchange score. An application must be created in order to access the assistant.

Calendar

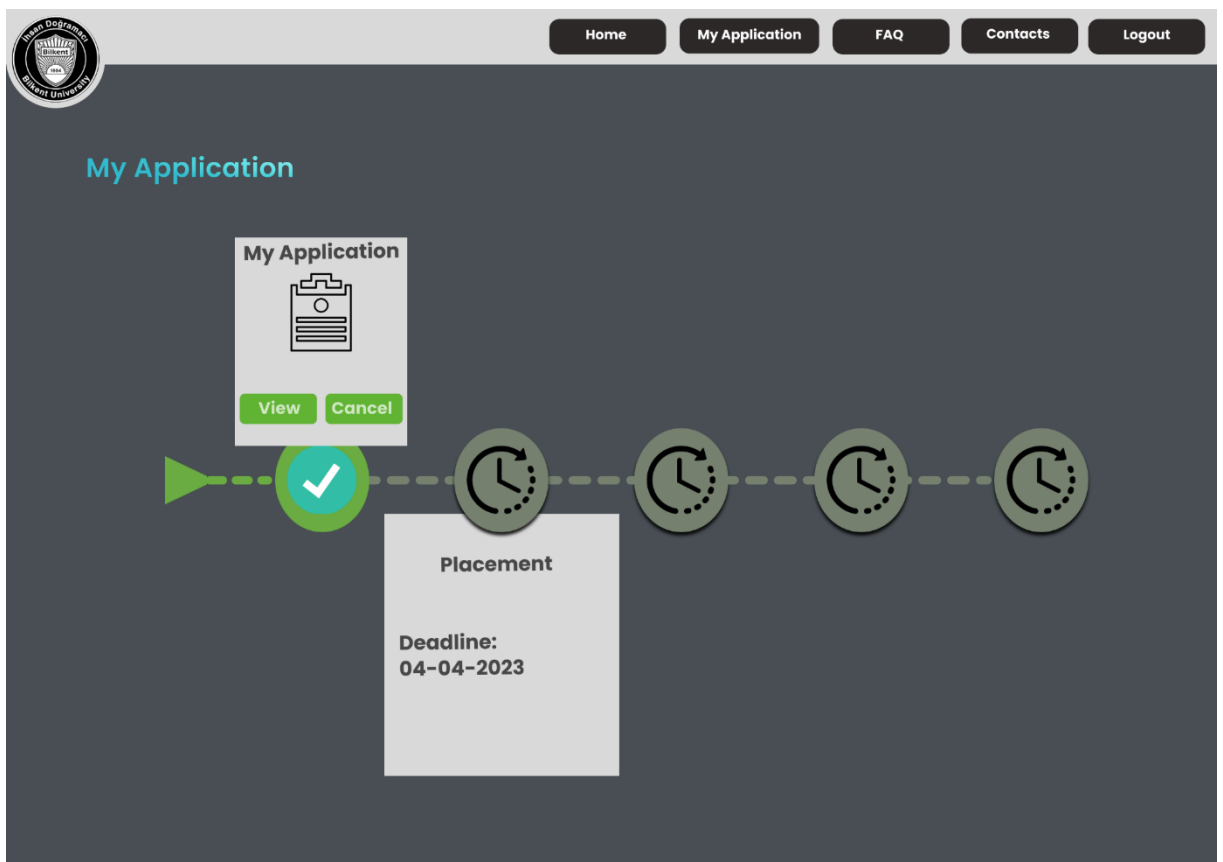
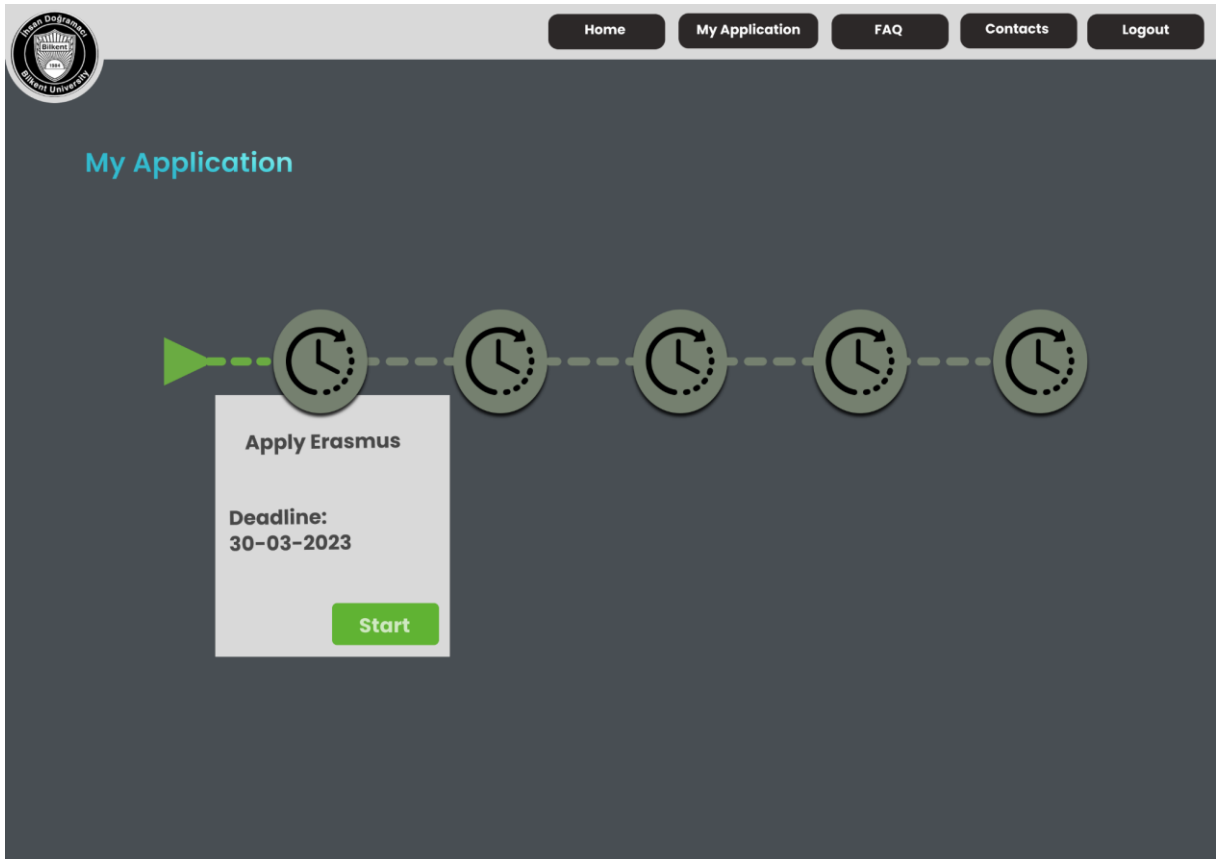
JAN 2022

S	M	T	W	TH	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Closing Deadlines

Application Deadline: 30-03-2023

Review Deadline: 03-04-2023



[Home](#)[My Application](#)[FAQ](#)[Contacts](#)[Logout](#)

My Application

My Application

[View](#)[Cancel](#)


Placement

[View](#)[Cancel](#)

Create
Preapproval Form


Deadline:
Until Student's
Mobility semester

[Choose](#)




[Home](#)
[My Application](#)
[FAQ](#)
[Contacts](#)
[Logout](#)


My Application

My Application



[View](#)
[Cancel](#)

Courses



[View](#)
[Cancel](#)

Courses


[View](#)
[Cancel](#)

Final Courses



[View](#)
[Cancel](#)



Waiting for Approval


Erasmus Coordinator: Approved

Faculty Administration: Pending




[Home](#)
[My Application](#)
[FAQ](#)
[Contacts](#)
[Logout](#)


My Application

My Application



[View](#)
[Cancel](#)

Placement


[View](#)
[Cancel](#)

Courses


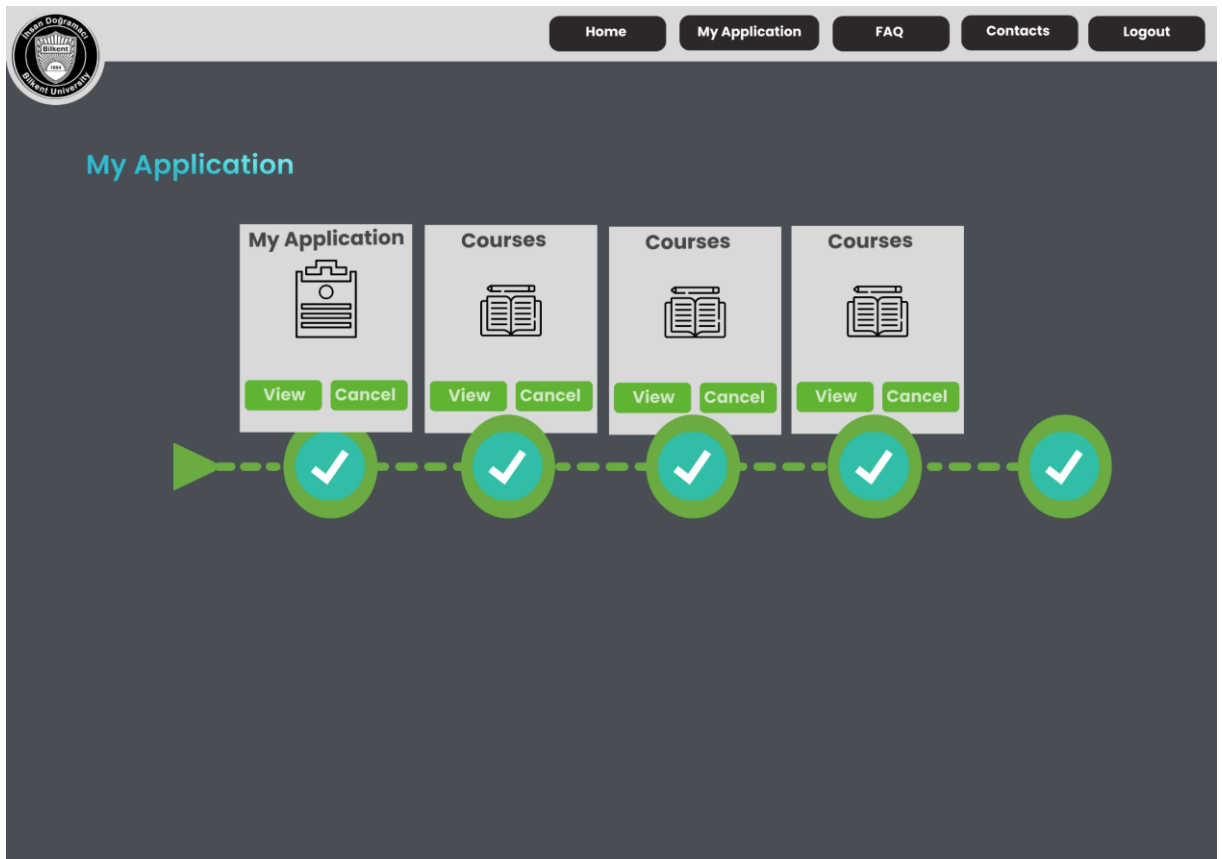
[View](#)
[Cancel](#)




Send Preapproval Form

Deadline:
Until Student's
Mobility semester

[Choose](#)















Erasmus Coordinator Pages



HomeAll ApplicationsUpdate FAQLogout

On Going Applications

Student ID: 21500001	Status: Approved 	 View	 Message
Student ID: 21600001	Status: Denied 	 View	 Message
Student ID: 21600021	Status: Approved 	 View	 Message
Student ID: 2160109	Status: Approved 	 View	 Message

[View All Applications](#)

Calendar

JAN 2022


S	M	T	W	TH	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Closing Deadlines

Review Deadline: 03-04-2023


To Do List

▼ New Tasks
Send Announcement

▼ Today
Review 5 applications
Reply messages 





























▼ Upcoming
Update FAQ
Review 10 elective course proposals


[Add Task](#)



HomeAll ApplicationsUpdate FAQLogout

All Applications

Application ID	Name of Applicant	Dept	Status	Submission Date		
 21504859	Onur Asım İlhan	CS	Approved 	05-01-2023	 View	 Message
 21504009	Murat Güney Kemal	CS	Pending 	05-01-2023	 View	 Message
 21504800	Cengizhan Terzioğlu	CS	Approved 	05-01-2023	 View	 Message
 21500059	Gökтуğ Kuşçu	CS	Declined 	05-01-2023	 View	 Message
 21503219	Emirhan Derköken	CS	Approved 	05-01-2023	 View	 Message
 21502233	Bora Yılmaz	CS	Approved 	05-01-2023	 View	 Message
 21500978	Mehmet Demir	CS	Pending 	05-01-2023	 View	 Message



[Home](#)
[All Applications](#)
[Update FAQ](#)
[Logout](#)

Update FAQ

Frequently Asked Questions

Q.1. What is the student exchange program? An important component of university education is to develop a better understanding of the world, not just academically but also socially. Student exchange is a means to achieve this goal by enabling students study at an environment different from their home country and institution.	×	Edit
Q.2. What are the financial issues involved for an exchange student?	+	Edit
Q.3. Is housing provided to exchange students at the host universities?	+	Edit
Q.4. Who can apply for the exchange program?	+	Edit
Q.5. Can't I go to exchange on the 4th year?	+	Edit

Add Question


What is the student exchange program?

Add Description

An important component of university education is to develop a better understanding of the world, not just academically but also socially. Student exchange is a means to achieve this goal by enabling students study at an environment different from their home country and institution.

[Add](#)

Course Coordinator Pages



[Logout](#)

Last Proposals

Student ID: 21607548	Proposition: CS231 - BILKENT with CS123 - XYZ Univeristy	Status: Approved	View	Message
Student ID: 21605675	Proposition: CS231 - BILKENT with CS123 - XYZ Univeristy	Status: Rejected	View	Message
Student ID: 21504433	Proposition: CS231 - BILKENT with CS123 - XYZ Univeristy	Status: Approved	View	Message

[View All Proposals](#)

Calendar

JAN 2022

S	M	T	W	TH	F
1	2	3	4	5	6
8	9	10	11	12	13
15	16	17	18	19	20
22	23	24	25	26	27

Closing Deadlines

Review Deadline: 01-04-2023

3. Improvement Summary

2.2.1 Usability: Section is updated based on the feedback.

2.2.5 Security: Section is updated based on the feedback.

2.4.1 Use Case Model: Use case diagrams are drawn from scratch. All user types are separated into different diagrams. More use cases are added. Text explanations of most of the use cases are updated. User is added as an actor. Exchange coordinator is changed into erasmus coordinator and dean/director is changed into faculty administration committee.

2.4.2.1.1 Student State Diagram: The diagram is updated based on the feedback.

2.4.2.1.3 University State Diagram: Minor grammar corrections are made.

2.4.2.2 Activity Diagram: Section name is corrected; new diagram is created.

2.4.2.3 Sequence Diagram: 2 new diagrams are created.

2.4.3 Class Diagram: Updated according to feedback, minor typos and ambiguities fixed.

2.4.4 User Interface: Entirety of the user interface mockups are added.