

# Bilkent University

CS-319

Object-Oriented Software Engineering



Fall 2022/2023

## Requirement Analysis Report

Group "erasmus muarrem" - 2D

Instructor: Eray Tüzün

TAs: İdil Hanhan, Muhammad Umair Ahmed,

Mert Kara, Emre Sülün

10/12/2022

<u>Member Name</u>	<u>Student ID</u>	<u>Section</u>
Korhan Kemal Kaya	21903357	02
Halil Alperen Gözetin	21902464	02
Hazal Buluş	21903435	02
Kürşad Güzelkaya	21902103	02
Süleyman Gökhan Tekin	21902512	02

<b>1. Introduction</b>	<b>3</b>
<b>2. Proposed System</b>	<b>3</b>
<b>2.1 Functional Requirements</b>	<b>5</b>
2.1.1 Actor Types	5
2.1.1.1 Outgoing Student	5
2.1.1.2 Department Coordinator	5
2.1.1.3 Course Coordinator	5
2.1.1.4 Administrative Staff	5
2.1.1.5 Incoming Student	6
2.1.2 Login	6
2.1.3 View Profile	6
2.1.4 View Application	6
2.1.5 Add Document	6
2.1.6 View Student Placement List	6
2.1.7 View Waiting Bin	7
2.1.8 Respond to Pre-Approval Form	7
2.1.9 Create Pre-Approval Form	7
2.1.10 Create Course Request	7
2.1.11 Respond to Course Request	7
2.1.12 Add an Announcement	7
2.1.13 Edit an Announcement	8
2.1.14 View To-Do List	8
2.1.15 Send Replacement Offer	8
2.1.16 Add a University	8
2.1.17 Edit University Details	8
2.1.18 Respond to Replacement Offer	8
2.1.19 Request Document	9
2.1.20 Add Language Proficiency	9
2.1.21 Cancel Application	9
2.1.22 Cancel Placement	9
2.1.23 View Announcements	9
<b>2.2 Non-Functional Requirements</b>	<b>9</b>
2.2.1 Usability	9
2.2.2 Reliability	10
2.2.3 Response Time	10
2.2.4 Supportability	11
2.3.2 Object Models and Controller Classes	25
2.3.3 Dynamic Models	28
2.3.3.1 Activity Diagrams	28
2.3.3.1.1 Student Placement Activity	28

2.3.3.1.2 Course Request Activity	29
2.3.3.1.3 Pre-Approval Form Creation Activity	30
2.3.3.2 Sequence Diagram	30
2.3.3.3 State Diagrams	31
2.3.3.3.1 Administrative Staff's State Diagram	31
2.3.3.3.2 Outgoing Students' Placement Process State Diagram	32
2.3.3.3.3 Outgoing Students' Pre-approval Process State Diagram	33
2.3.4 User Interface	35
<b>3. Glossary and References</b>	<b>51</b>
<b>4. Appendix</b>	<b>52</b>

## **1. Introduction**

Every year, many Bilkent students apply to Erasmus and Exchange programs, and this application and nomination period for students covers around a period of 3-4 months. [1] Although this process requires intensive mailing and paperwork, there is no real action taken to resolve this problem by digitalizing the steps. An application called “App Erasmus Bilkent” is created in this direction; nevertheless, it’s insufficient in providing support for the whole process [2].

In accordance with this, we sought the opportunity to develop a more comprehensive application by addressing the different needs of this problem and decided to build a web application that will make the process more manageable for different participating actors, such as students, department, and course coordinators, and the administrative staff. Through our application, the outgoing students will be able to conduct post-application steps such as pre-approval forms, course and document requests, application tracking, and other actions. On the other hand, department coordinators will easily manage the placement and Pre-Approval Form processes, administrate the language requirements of students and inform them through announcements. Another essential part of the process is the administrative staff, who will be able to manage the contracted partner universities and their details and respond to document-related demands of students. Finally, we also aimed to provide support to incoming students in our application so that they can propose a list of courses they wish to take and track their status.

## **2. Proposed System**

In the application that we propose, we plan to provide a simple and revealing user interface with many different functionalities to provide full support to both students and administrative people by addressing the needs that we discovered in our interview with the administrative Exchange coordinator Yelda Ateş (See Appendix-A). We will provide different types of accounts (Outgoing Student, Department Coordinator, etc.) with different permissions and functionality to make sure the entire process works in a coordinated manner.

After logging into their accounts, the Outgoing Students will interact with the app by switching between their application page, profile page, and other pages linked with interactions. On the application page, they can keep track of all the files uploaded, either by themselves or by an administrative person, to the system related to their application process. Also, on this page, they will find various details like the type of application, acceptance

situation, current status, next action to take, submitted and evaluated pre-approval forms, etc., that will help them to easily manage their application process. Besides that, the main page (profile page) will display the announcements made by the department coordinators and also the document requests. The student will be able to create or delete a request or view the incoming administrative requests.

Department Coordinators, on the other hand, will be able to view the placement list and the waiting bin. Besides viewing, they will also be able to manage the placement process and handle situations when changes in the applications occur (for ex., cancellation). Another page that they will interact with is the announcements page, where they will edit, delete, or create announcements. The Pre-Approval forms will also be managed by the Department Coordinators on a separate page, where they will accept or decline the requests of students and attach feedback to them. Besides that, the elective course requests will require approval from Department Coordinators, and functionality will be provided for that purpose. Another thing that Department Coordinators will have control over will be the language details of the students, and they will be able to add specific language information (the language exam results) to any student.

The authority to add, delete, or edit partner universities will be given to the Administrative Staff by providing a separate page to display and interact with a list of universities. Even though they won't intervene with the placement process, the Administrative Staff will be able to view the placement list and also the waiting bin, and they will be able to interact with any student by simply searching for them in the list. The Administrative Staff will be provided a "Document Requests" page to view and respond to document (Acceptance Letter, Learning Agreement, etc.) requests of the students by uploading the necessary document.

The Course Coordinator in our app will be responsible for one or more courses. They will view and respond to mandatory course requests (which contain a description and the syllabus of the corresponding course) of students by attaching feedback. Also, Course Coordinators will send a request to the students to upload a final project report after mobility if the course requires a final project. Similar to the Administrative Staff and Department Coordinator, they will also be able to view the list of all applicants.

Our other aim is to add chat functionality for all the actors mentioned to make our application more convenient for users and integrated. This will be done by designing a simple user interface in which the users will be able to continue their interactions with our app while using the chat. Besides that, in order for the Department Coordinators, Administrative Staff,

and Course Coordinators to maintain the process and manage their time, we planned to add a “To-Do List” page for viewing the necessary actions such as responding to forms, uploading documents, making announcements, etc.

Therefore, we plan to deliver all the functionality mentioned to the user in a way that the user will be able to take the necessary action in seconds and not need an outer source to get the job done other than our application.

## **2.1 Functional Requirements**

### **2.1.1 Actor Types**

#### ***2.1.1.1 Outgoing Student***

The actor “Outgoing Student” represents students who are currently studying at Bilkent University and have applied to Erasmus or Exchange programs. They are responsible for creating Pre-Approval forms and course requests and uploading final reports. Outgoing students are able to request specific documents from Administrative Staff.

#### ***2.1.1.2 Department Coordinator***

The actor “Department Coordinator” represents the coordinators of the departments at Bilkent University. They are responsible for sending replacement offers to the students (in case of cancellations) who are in the Waiting Bin after the initial placements are done by the placement algorithm of the system, responding to elective course requests, and responding to Pre-Approval forms. The Department Coordinator can also view the student placement list and add a language proficiency to the student, and add or edit an announcement. They have their own “To-Do List” page.

#### ***2.1.1.3 Course Coordinator***

The actor “Course Coordinator” is responsible for responding to course requests coming from Outgoing Students, and they can view the list of applications and each student’s profile. They will also be able to request final reports from students after the mobility, and they also have their own “To-Do List” page to manage their work more efficiently.

#### ***2.1.1.4 Administrative Staff***

The actor “Administrative Staff” represents other responsible actors in Erasmus and Exchange program processes other than Course Coordinators and Department Coordinators. They can submit required documents according to requests coming from students and view the list of applications and individual student profiles. The responsibility of adding or editing university details will also be given to them. They have their own “To-Do List” page to manage their work more efficiently.

#### ***2.1.1.5 Incoming Student***

The actor “Incoming Student” represents students coming from other universities to Bilkent with Erasmus or Exchange programs. They can create a course proposal for their planned courses and edit and submit this proposal.

### **2.1.2 Login**

The initial screen of the app is the login screen. In the login screen, users, which can have any role in our application, can successfully log in to the app if they input the correct credentials.

### **2.1.3 View Profile**

Outgoing Students and Incoming Students can view their individual profiles to track information related to them and also the updates in the application process. Other actors can view the profile of each student by accessing them through the Waiting Bin or Placement List.

### **2.1.4 View Application**

Outgoing Students can view their application process status. Other actors can view the application processes of each student again by accessing them through the Waiting Bin or Placement List.

### **2.1.5 Add Document**

Outgoing Students and Administrative Staff are able to upload the required documents to the system. This functionality will be used in different processes like Pre-Approval Forms, Course Requests, etc.

### **2.1.6 View Student Placement List**

Administrative Staff, Department Coordinators, and Course Coordinators are able to see the students according to their placements and interact with the list elements. The users with these roles can also view the students' profiles through this placement list.

### **2.1.7 View Waiting Bin**

Administrative Staff, Department Coordinators, and Course Coordinators are able to see the students in the waiting bin who have made their application for Erasmus/Exchange but haven't been placed in a university yet.

### **2.1.8 Respond to Pre-Approval Form**

Department Coordinators are able to respond to Pre-Approval forms created by the outgoing students by accepting or rejecting them. In case of rejection, feedback should be provided by the Department Coordinator to the Outgoing Student.

### **2.1.9 Create Pre-Approval Form**

Outgoing Students are able to create Pre-Approval forms in order to register for new courses in their mobility period of Erasmus/Exchange programs. They can submit and send their forms to the Department Coordinator simply by clicking a button.

### **2.1.10 Create Course Request**

Outgoing Students are able to create a new course request if they have not requested that course before. Depending on the type of course they requested, these requests are sent to



the course coordinators (if the course is a mandatory course) or the department coordinators (if the course is an elective course).

#### **2.1.11 Respond to Course Request**

Course Coordinators are able to respond to the mandatory course requests which are requested by Outgoing Students. On the other hand, if the requested course is an elective course, Department Coordinators are able to respond to these requests, which are requested by outgoing students.

#### **2.1.12 Add an Announcement**

Department Coordinators are able to add new announcements to inform Outgoing Students about the changes/updates in the process simply by entering a title and description for the announcement.

#### **2.1.13 Edit an Announcement**

Department Coordinators are able to edit announcements they have added to the system before simply by changing the details like title and description or are able to delete them completely.

#### **2.1.14 View To-Do List**

Administrative Staff, Department Coordinators, and Course Coordinators are able to see view their next needed actions in a “To-Do” list. They can also mark the items in the list as complete after they have done it.

#### **2.1.15 Send Replacement Offer**

If the Waiting Bin is not empty and the student to which the request is going to be sent is not already admitted to a university, the Department Coordinators can send a replacement offer.

#### **2.1.16 Add a University**

The Administrative Staff can add a new university for the future Erasmus/Exchange periods if the university planned to be added is not already existing in the system by filling in details like the quota, accepted courses, etc.

#### **2.1.17 Edit University Details**

The Administrative Staff can edit an existing university in the Erasmus/Exchange application system by changing the details like the quota, accepted courses, etc.

#### **2.1.18 Respond to Replacement Offer**

Outgoing Students who are in the Waiting Bin are able to respond to the replacement offers if they have any replacement offers from the department coordinator simply by clicking on the accepting button.

#### **2.1.19 Request Document**

Outgoing Students are able to request documents they need, such as Acceptance Letter and Learning Agreement, from the Administrative Staff and wait for the response.

#### **2.1.20 Add Language Proficiency**

Department Coordinators are able to add the language proficiency (i.e., German) for the Outgoing Students whose language proficiency is proven with exams that are already held. With this functionality, Outgoing Students with adequate language levels satisfy the language requirements of universities.

#### **2.1.21 Cancel Application**

If the cancellation deadline for the applications is not passed, Outgoing Students are able to cancel their applications to Erasmus/Exchange and terminate the whole process. If it exists, their placement at a university is canceled as well.

### **2.1.22 Cancel Placement**

In case an Outgoing Student is not satisfied with the university they have been placed by the system, they are able to cancel this placement, and if accepted, they will be placed in the Waiting Bin.

### **2.1.23 View Announcements**

Outgoing Students and Department Coordinators are able to view and interact with the announcements, which are initially added by Department Coordinators, each element containing information about the updates to the Erasmus/Exchange process.

### **2.1.24 Create Course Proposal**

Incoming students are able to create course proposals, to take a course during their Erasmus process in the host university which will then be reviewed by the Administrative Staff. They will also be able to make changes to the proposal as well track its status.

## **2.2 Non-Functional Requirements**

### **2.2.1 Usability**

Since the purpose of creating this system is to facilitate the process of the students who will go to Erasmus and Exchange programs, and the course coordinators, department coordinators, and staff responsible for Erasmus Exchange programs, a useful and practical system is required. Users of the system should benefit more from the ongoing system. Since the purpose of the system is to minimize the use of paper transfer and online mailing, it will be designed so that all users can upload certain required documents. Regardless of the screen size, the font size and the size of the buttons will be easily visible on each screen. The font on each screen will be 14 pt, the dimensions of the buttons will never be less than 7% of the screen, and information tables will not be more than 5 columns so that the texts and any other properties will be easily readable by the user. Additionally, the user will always be steered toward the appropriate input type whenever input is necessary; for example, if a number type is required, the user will not be allowed to write a text. Also, the system will be as simple and straightforward as possible by implementing the UI according to user experience. For instance, we will use sidemenu drawer, which is common in many applications and familiar to many

users, to provide navigation between main pages. To find a student profile, users will be able to search for a specific student (such as in the placement list that consists of too many students), which will enable them to find the specific student they are looking for. This will allow users to access the information they want easier and faster, which will increase usability in general.

### **2.2.2 Reliability**

Information and documents of the users will be stored privately. The passwords of the users will be stored as encrypted. Users will use the activation code that is sent to their email by the application to change their passwords in case they forget. To protect the application from unexpected crashes, possible errors and exceptions will be handled.

### **2.2.3 Response Time**

Time is a significant concern for both coordinators and students. Hence the application should fasten the application process as much as possible. Thanks to the application, most of the time wasted in the mailing process will be saved. Also, the application will reduce the paperwork, which also will save time by cutting out the time required to transfer papers between members.

The user interface of the application will be designed in a way that users can manage their work both easily and fastly without wasting time figuring out how the application works as we explained in usability part. Users will be able to perform an action in less than five clicks. Also, performance tests will be performed after the implementation to ensure the response time of the application is not problematic. The application's performance score in the lighthouse's performance test will be at least 85. In addition, in the implementation, lazy loading and coded splitting will be used to save time wasted by loading unnecessary data. Lastly, the database connection will be made in a way that users can reach their data with simple queries and simple responses without wasting time waiting for large chunks of data to come from the database.

### **2.2.4 Supportability**

The code structure of the application should be divided into components in order to be compatible with possible changes and innovations that can be added to the application in the

future, and the code structure should be suitable for division; thus, making changes in small code blocks will cause a few errors. Likewise, in order to facilitate any possible change and increase readability, writing code blocks in an orderly manner and explanatory comments to be added at necessary points will facilitate the understanding and reusability of components. Thanks to the reusability of the components, functionalities that will upgrade the application will be added in a faster and easier way since if we need to add a small component on four different pages, instead of copying and pasting the code segment of the component, again and again, we will call the class of the component with required properties which provide making a difference in components. Also, design patterns that we will use like MVC, Facade and Singleton increase the reusability of the components.

## **2.3 System Models [3]**

### **2.3.1 Use-Case Model**

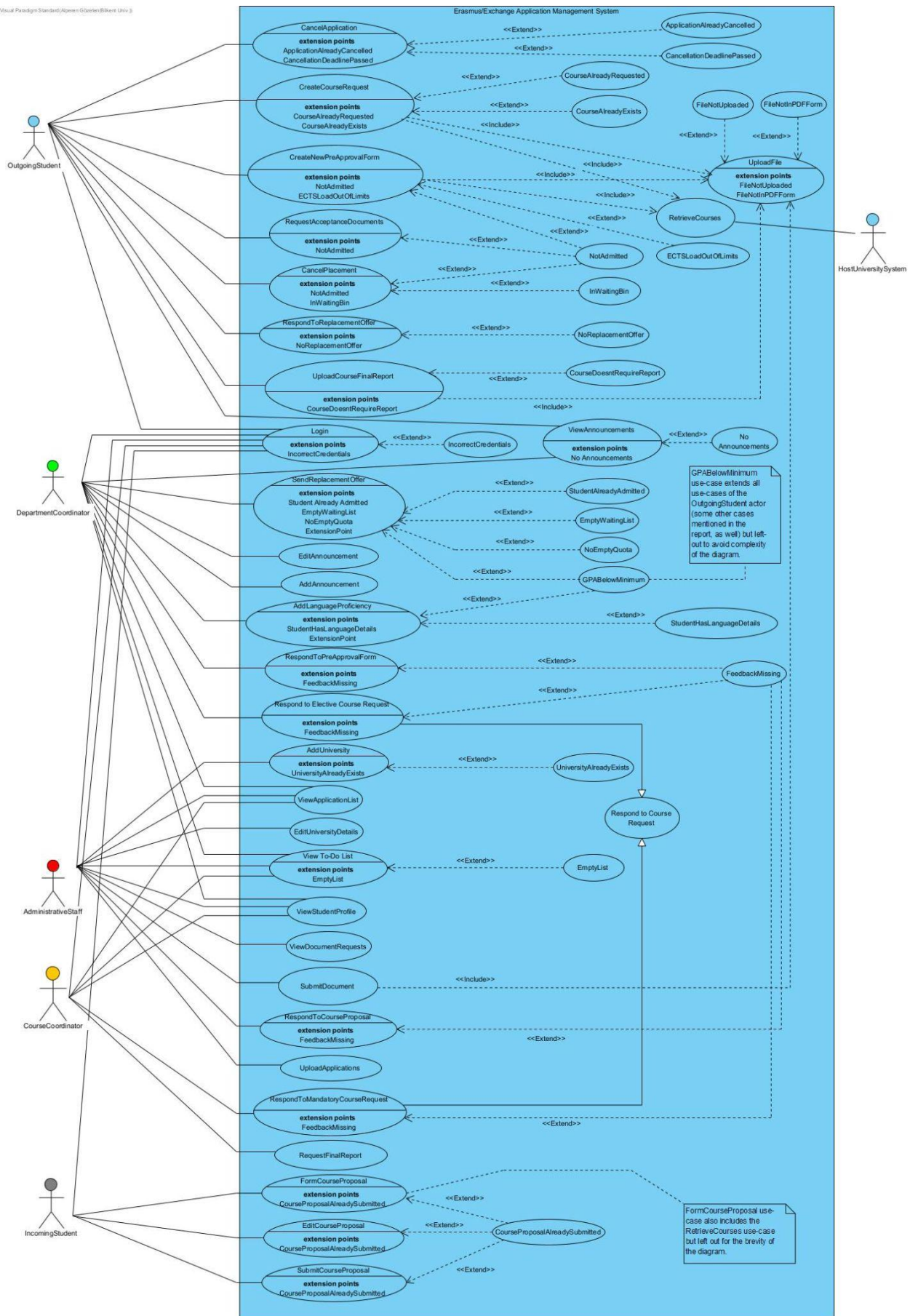


Fig. 1: General Use-Case Diagram (Click [here](#) for larger size image)

Every use case described below has the following two common requirements:

- The server must be in a running state.
- The correspondent actor must be connected to the internet.

Besides that, every use-case that is connected with the OutgoingStudent actor (together with the SendReplacementOffer, AddLanguageProficiency and RespondToPreApprovalRequest, and SubmitDocument use-cases that are directly related with the application status) has the requirement that the current GPA of the student is above 2.5.

---

**Name:** Login

**Participating Actor:** Outgoing Student, Department Coordinator, Administrative Staff, Course Coordinator, Incoming Student

**Entry Condition:**

- Opening the application page and having a valid account.

**Exit Condition:**

- The actor successfully logs in.

**Flow of Events:**

1. The actor enters a username
2. The actor enters a password
3. The information inputted by the actor will be checked, and the actor will be logged in to the application.

**Special/Quality Requirements:**

- Once the information is inputted by the actor, they should be logged in to the application in 1 second.

---

**Name:** Cancel Application

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor should have an existing application
- The deadline for the application cancellation shouldn't have already passed.

**Exit Condition:**

- The actor has had their application canceled and is no more considered for any university.

- If exists, the placement of the student is canceled as well.

**Flow of Events:**

1. The actor picks the accepting option of the clarification question.

**Special/Quality Requirements:**

- The application must be canceled within 1 second after the actor picks to cancel the application process.
- 

**Name:** Create Course Request

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The course requested by the actor shouldn't already exist
- The course shouldn't have been previously requested by the same actor

**Exit Condition:**

- The actor has a waiting course request, and the Course Coordinator or Department Coordinator (depending on if the course is mandatory or elective) has a waiting request approval.

**Flow of Events:**

1. The actor needs to choose between the elective or mandatory type of course.
2. The actor fills in the necessary information related to the course
3. The actor uploads the syllabus for the course
4. The actor clicks to submit the request.

**Special/Quality Requirements:**

- The corresponding document must be uploaded to the system 1 second after the actor picks to create the course request, and the request is sent to either the course coordinator or department coordinator.
- 

**Name:** Create New Pre-Approval Form

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor must be already admitted to a university
- The total ECTS of the courses in the form must be between 30 and 40 (the second number isn't strict here but it's our assumption for the design)



**Exit Condition:**

- The Pre-Approval Form is uploaded to the system, and the approval request is sent to the corresponding Department Coordinator

**Flow of Events:**

1. The actor chooses (merges) the courses such that the total ECTS course load is between the limits (minimum of 30 and maximum of 40 as our assumption for the design).
2. The actor uploads the Pre-Approval form in PDF format.
3. The actor picks the submitting option.

**Special/Quality Requirements:**

- The corresponding document must be uploaded to the system 1 second after the actor picks to create the form, and the request is sent to the Department Coordinator.
- 

**Name:** Request Acceptance Documents

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor must be already admitted to a university

**Exit Condition:**

- The actor has a waiting document request, and the Administrative Staff has a waiting request for approval.

**Flow of Events:**

1. The actor picks the type of document they request.
2. The actor picks the requesting option.

**Special/Quality Requirements:**

- The corresponding document must be uploaded to the system 1 second after the actor picks to create the form, and the request is sent to the Administrative Staff.
- 

**Name:** Cancel Placement

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor who attempts to cancel the placement must be currently admitted to a university

- The actor shouldn't be in the waiting bin, which means should have already canceled the placement.

**Exit Condition:**

- The actor has their current placement canceled and put in the Waiting Bin if the actor wants to.

**Flow of Events:**

1. The actor chooses between being put in the Waiting Bin for further placements or not.
2. The actor cancels their placement by picking the cancellation option.

**Special/Quality Requirements:**

- The placement must be canceled within 1 second after the actor picks to cancel the placement.

---

**Name:** Respond to Replacement Offer

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor has a waiting replacement offer from the Department Coordinator.

**Exit Condition:**

- The actor is placed in the corresponding university if the actor accepts the offer; no change if the actor declines.

**Flow of Events:**

1. The actor chooses to accept the replacement offer.

**Special/Quality Requirements:**

- The actor should be placed in the corresponding university 1 second after they choose to accept.

---

**Name:** Upload Course Final Report

**Participating Actor:** Outgoing Student

**Entry Condition:**

- The actor is enrolled in a course in mobility that requires a final report.

**Exit Condition:**

- The actor has a final report uploaded to the system waiting for approval from the course coordinator, and the course coordinator has a report waiting for evaluation.

**Flow of Events:**

1. The actor enters the course name
2. The actor uploads the corresponding final report
3. The actor clicks to submit the final report.

**Special/Quality Requirements:**

- The corresponding document must be uploaded to the system 1 second after the actor picks to submit the final report, and the information is sent to the Administrative Staff.
- 

**Name:** View Announcements

**Participating Actor:** Outgoing Student, Department Coordinator

**Entry Condition:**

- There's at least one announcement to be displayed.

**Exit Condition:**

- The list of announcements is displayed to the actor.

**Flow of Events:**

1. The actor chooses to view the list of announcements.
2. The actor interacts with the list elements, which contain information related to the applications of students and also links to their profiles.

**Special/Quality Requirements:**

- None
- 

**Name:** Send Replacement Offer

**Participating Actor:** Department Coordinator

**Entry Condition:**

- There must be at least 1 open quota to be able to make an offer.
- There must be at least one student in the Waiting Bin (not admitted).
- The student who the offer is going to be sent should not be admitted before.

**Exit Condition:**

- The offer is sent to the student, waiting for their approval.

**Flow of Events:**

1. The actor chooses to send a replacement offer when interacting with the student.

**Special/Quality Requirements:**

- The offer should be sent to the student 1 second after the actor chooses to send a replacement offer.

---

**Name:** Add an Announcement

**Participating Actor:** Department Coordinator

**Entry Condition:**

- The actor chooses to add a new announcement.

**Exit Condition:**

- A new announcement has been added and can now be displayed by Outgoing Students and Department Coordinators.

**Flow of Events:**

1. The actor enters information related to an announcement, such as the title and description.
2. The actor chooses to add the announcement.

**Special/Quality Requirements:**

- The announcement should be added to the system 1 second after the actor chooses to submit and should be ready to be displayed.

---

**Name:** Edit an Announcement

**Participating Actor:** Department Coordinator

**Entry Condition:** The actor chooses to edit the announcement

**Exit Condition:** The announcement is updated according to the provided info and can be displayed by the users of the application.

**Flow of Events:**

1. The actor changes/fills in information related to an announcement, such as the title and description, or deletes the announcement.
2. The actor chooses to submit the changes.

**Special/Quality Requirements:**

- The changes to the announcement should be added to the system 1 second after the actor chooses to submit and should be ready to be displayed.

---

**Name:** Add Language Proficiency

**Participating Actor:** Department Coordinator

**Entry Condition:**

- The language detail (i.e., German) which is to be added to the student shouldn't already exist for the student.

**Exit Condition:**

- The language detail is added to the student, and if their level is sufficient, they now satisfy the language requirements of certain universities.

**Flow of Events:**

1. The actor picks a language and enters the proficiency level of the student according to the exam results.
2. The actor chooses to submit the information.

**Special/Quality Requirements:**

- The information should be added to the student 1 second within the actor who chooses to submit language details.

---

**Name:** View Application List

**Participating Actor:** Department Coordinator, Administrative Staff, Course Coordinator

**Entry Condition:**

- The actor chooses to view the list of applicants.

**Exit Condition:**

- The list of all applicants is shown to the actor.

**Flow of Events:**

1. The list of applications is displayed for the actor to interact.

**Special/Quality Requirements:**

- None

---

**Name:** Respond to Pre-Approval Form

**Participating Actor:** Department Coordinator

**Entry Condition:**

- A feedback description should be given by the Department Coordinator in case of rejection of the form, with no such condition for acceptance.

**Exit Condition:**

- The student who submitted has an evaluation for their form, and if approved, they can register for this list of courses in mobility.

**Flow of Events:**

1. The actor views the request and downloads the form for inspection.
2. The actor enters feedback in case of impropriety in the form.
3. The actor chooses to approve or reject the Pre-Approval Form.

**Special/Quality Requirements:**

- The response should be uploaded to the system and sent to the student within 1 second after the actor chooses to submit the response.

---

**Name:** Respond to Elective Course Request

**Participating Actor:** Department Coordinator

**Entry Condition:**

- A feedback description should be given by the Department Coordinator in case of rejection of the course request, with no such condition for acceptance.

**Exit Condition:**

- The student who submitted the course request has an evaluation for their request and can add this course to the Pre-Approval Form if approved.

**Flow of Events:**

1. The actor views the course request and downloads the syllabus for inspection.
2. The actor enters feedback in case of significant divergence between the courses.
3. The actor chooses to approve or reject the course request.

**Special/Quality Requirements:**

- The response to the course request should be uploaded to the system and sent to the student 1 second after the actor chooses to submit the response.

---

**Name:** View Student Profile

**Participating Actor:** Department Coordinator, Administrative Staff, Course Coordinator

**Entry Condition:**

- The actor chooses the view student profile option.

**Exit Condition:**

- Information related to the student is displayed for the actor to interact.

**Flow of Events:**

1. The actor sees different information related to the student

**Special/Quality Requirements:**

- None
- 

**Name:** Add a University

**Participating Actor:** Administrative Staff

**Entry Condition:**

- The university that is trying to be added by the actor shouldn't already exist.

**Exit Condition:**

- A new university has been added to the application system.

**Flow of Events:**

1. The actor fills in necessary information related to the university, such as the type of agreement (Erasmus, Exchange, or both), quotas, semesters, etc.
2. The actor chooses to submit the new university.

**Special/Quality Requirements:**

- The university should be added to the system within 1 second after the actor chooses to submit.
- 

**Name:** Edit University Details

**Participating Actor:** Administrative Staff

**Entry Condition:**

- The actor chooses to edit the details of a university when interacting.

**Exit Condition:**

- The information related to the corresponding university is updated.

**Flow of Events:**

1. The actor edits necessary information related to the university, such as the type of agreement (Erasmus, Exchange, or both), quotas, semesters, etc.
2. The actor chooses to submit the changes.

**Special/Quality Requirements:**

- The changes to the university should be added to the system within 1 second after the actor chooses to submit.

---

**Name:** View Document Requests

**Participating Actor:** Administrative Staff

**Entry Condition:**

- The actor chooses to view the document requests by the students.

**Exit Condition:**

- The document requests are displayed for the actor to interact

**Flow of Events:**

1. The list of applications is displayed for the actor to interact and respond.

**Special/Quality Requirements:**

- None

---

**Name:** Submit Document

**Participating Actor:** Administrative Staff

**Entry Condition:**

- The actor chooses to respond to a waiting document request by the student.

**Exit Condition:**

- The student receives a document and can view it on their page.

**Flow of Events:**

1. The actor chooses the request to respond.
2. The actor uploads the corresponding document and chooses to submit it.

**Special/Quality Requirements:**

- The document added by the actor should be uploaded to the system and sent to the student 1 second after the actor chooses to submit the document.

---

**Name:** Respond to Course Proposal

**Participating Actor:** Administrative Staff

**Entry Condition:**



- The actor chooses to respond to a course proposal submitted by an incoming student and provides feedback in case of rejection, with no such condition for approval.

**Exit Condition:**

- The incoming student who submitted the course proposal has an evaluation for it and can take this list of courses if approved.

**Flow of Events:**

1. The actor inspects the proposal submitted by the incoming student.
2. The actor enters feedback in case of impropriety in the proposal.
3. The actor chooses to approve or reject the course proposal

**Special/Quality Requirements:**

- The response to the course proposal should be uploaded to the system and sent to the incoming student within 1 second after the actor chooses to submit the response.

---

**Name:** Upload Applications

**Participating Actor:** Administrative Staff

**Entry Condition:**

- The actor chooses to upload the applications for starting the Erasmus/Exchange process.

**Exit Condition:**

- The Erasmus/Exchange process has started, and the initial placements of the students have been decided by the internal placement algorithm of the system.

**Flow of Events:**

1. The actor chooses to upload the applications.
2. The actor submits the uploaded document.

**Special/Quality Requirements:**

The initial placements of the students must be determined within 15 seconds after the actor chooses to submit the application document.

---

**Name:** Respond to Mandatory Course Request

**Participating Actor:** Course Coordinator

**Entry Condition:**

- A feedback description should be given by the Course Coordinator in case of rejection of the course request, with no such condition for acceptance.

**Exit Condition:**

- The student who submitted the course request has an evaluation for their request and can add this course to the Pre-Approval Form if approved.

**Flow of Events:**

1. The actor views the course request and downloads the syllabus for inspection.
2. The actor enters feedback in case of significant divergence between the courses.
3. The actor chooses to approve or reject the course request.

**Special/Quality Requirements:**

- The response to the course request should be uploaded to the system and sent to the student 1 second after the actor chooses to submit the response.

---

**Name:** Form Course Proposal

**Participating Actor:** Incoming Student

**Entry Condition:**

- The actor shouldn't have already submitted their course proposal.

**Exit Condition:**

- The actor has their form prepared with the information provided.

**Flow of Events:**

1. The actor fills out the form by providing related information.
2. The actor chooses to save the changes.

**Special/Quality Requirements:**

- The course proposal created by the student should be saved and ready to view in 1 second after the actor chooses to submit it.

---

**Name:** Edit Course Proposal

**Participating Actor:** Incoming Student

**Entry Condition:**

- The actor should have a course proposal to edit, and it shouldn't have already been submitted.

**Exit Condition:**

- The previous proposal belonging to the actor is modified according to the information provided and waiting for submission.

**Flow of Events:**

1. The actor inspects the current form and makes changes to details.
2. The actor chooses to save the changes.

**Special/Quality Requirements:**

- The changes made to the course proposal by the student should be saved and ready to view in 1 second after the actor chooses to submit it.

---

**Name:** Submit Course Proposal

**Participating Actor:** Incoming Student

**Entry Condition:**

- The actor's GPA stays above the required minimum 2.5 before the mobility.
- The actor should have a course proposal to submit, and it shouldn't have already been submitted.

**Exit Condition:**

- The actor has their form submitted, and the information is sent to the administrative staff.

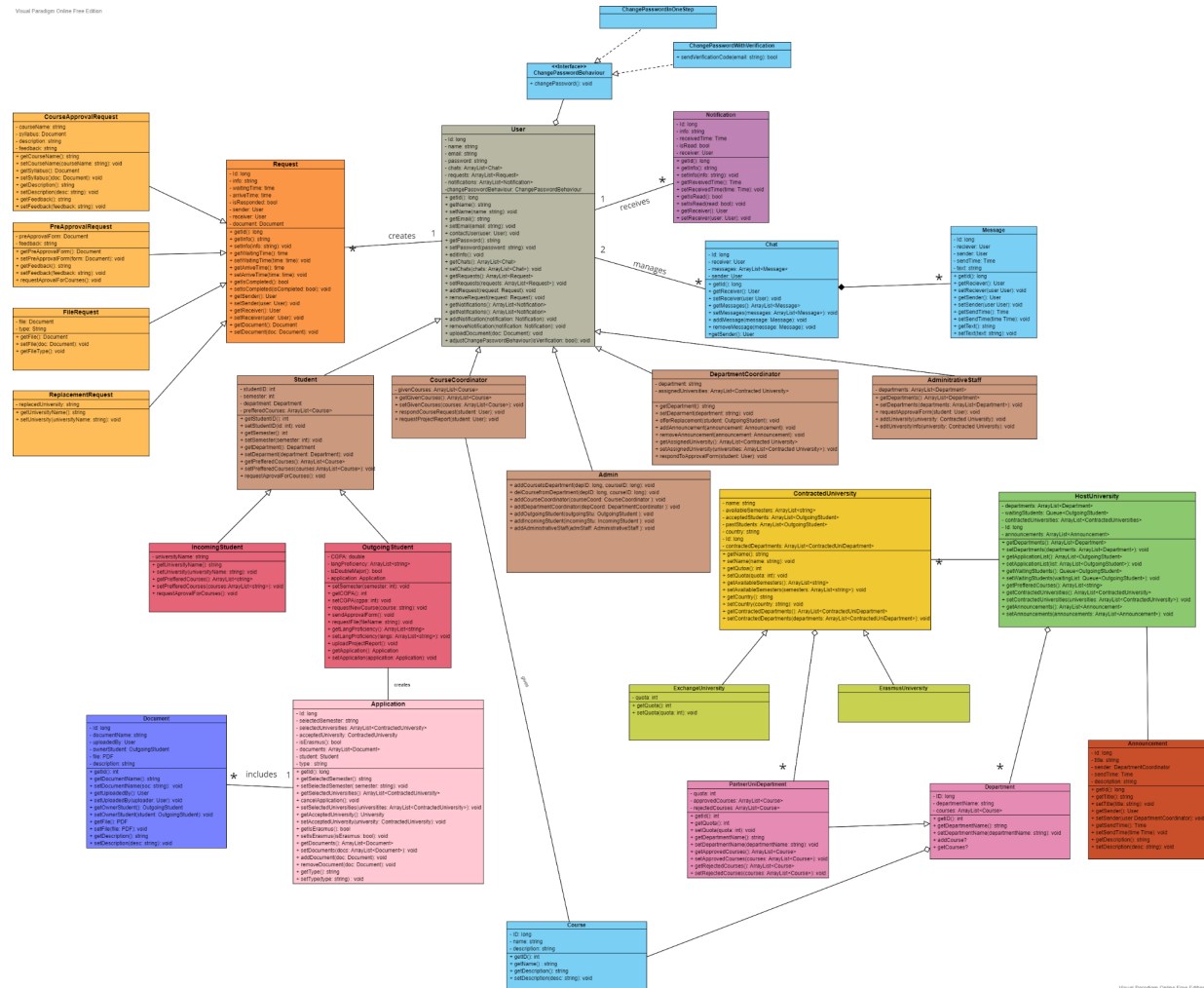
**Flow of Events:**

1. The actor inspects the prepared form before submitting it.
2. The actor chooses to submit the form.

**Special/Quality Requirements:**

- The proposal submitted by the actor should be saved and ready to view in 1 second after the actor chooses to submit it.

### 2.3.2 Object Models and Controller Classes



**Fig. 2:** Class Diagram (Click [here](#) for larger size image)

### Explanation

The whole application will be handled by two main classes: User and HostUniversity. The User class represents common features and operations of all actors, and it contains five classes for different actors. The User class contains the authorization information of the related user. User also has two different types of changing password so it has an instance of ChangePasswordBehaviour interface which has two types: ChangePasswordInOneMove and ChangePasswordWithVerification. Both of these classes have changePassword function in common but they implement differently. ChangePasswordInOneMove gets new password and changes the password when user is logged in. ChangePasswordWithVerification sends a verification email and then changes the password when user is not logged in. Users can have more than one chat to communicate with other users. Hence, the User class has a one-to-many association with the Chat class. The Chat class is used to keep messaging info and operations.

Also, there is the Message class which is a part-of Chat class and is used to keep details of the messages. Users can have notifications about their application processes. Therefore, the User class has a one-to-many association with the Notification class. The Notification class is used to keep notification details. Some of the user types can send or receive different requests about documents like Pre-Approval Form, Acceptance Letter, and Learning Agreement. Hence, the User class has a one-to-many association with the Request class. The Request class is used to represent a request and its details. The Request class contains four classes for different types of requests: CourseApprovalRequest, PreApprovalRequest, FileRequest, and ReplacementRequest. CourseApprovalRequest contains the details and syllabus of the course proposed for approval. Also, CourseCoordinator can send feedback with the response to this request. PreApprovalRequest contains the selected courses and related Document of the request. Also, DepartmentCoordinator can send feedback with the response of this request. FileRequest contains the file type (Acceptance Letter, Learning Agreement or Final Report) and related file of the request. The User class consists of five classes: Student, CourseCoordinator, DepartmentCoordinator, AdministrativeStaff, and Admin.

The Student class represents the information of the student, and it consists of two classes: IncomingStudent and OutgoingStudent. The IncomingStudent is a basic user that comes from contracted universities, and its class represents specific information about the incoming students and their course request operations. The OutgoingStudent is also another type of basic user who is applied to go to the contracted universities, and its class represents detailed information about the outgoing student and their application-related operations. Since each outgoing student has just one application, the OutgoingStudent class has a one-to-one association with the Application class. The Application class is used to represent details of the application and related documents. An application can have more than one document, so the Application class has a one-to-many association with the Document class. The Document class is used to represent details of the document and related files.

CourseCoordinator is another type of user who is responsible for the course-related staff. The CourseCoordinator class is used to represent the course details of the coordinator and have some operation to handle course requests of the students. DepartmentCoordinator is responsible for the application process. The Department Coordinator class has some information about the coordinator and the related operations of the coordinator. The Department Coordinator also can make some announcements. AdministrativeStaff is a user type that provides some documents to the students and handles the contracted university information. The Administrative class has department information on the staff and university-related operations.

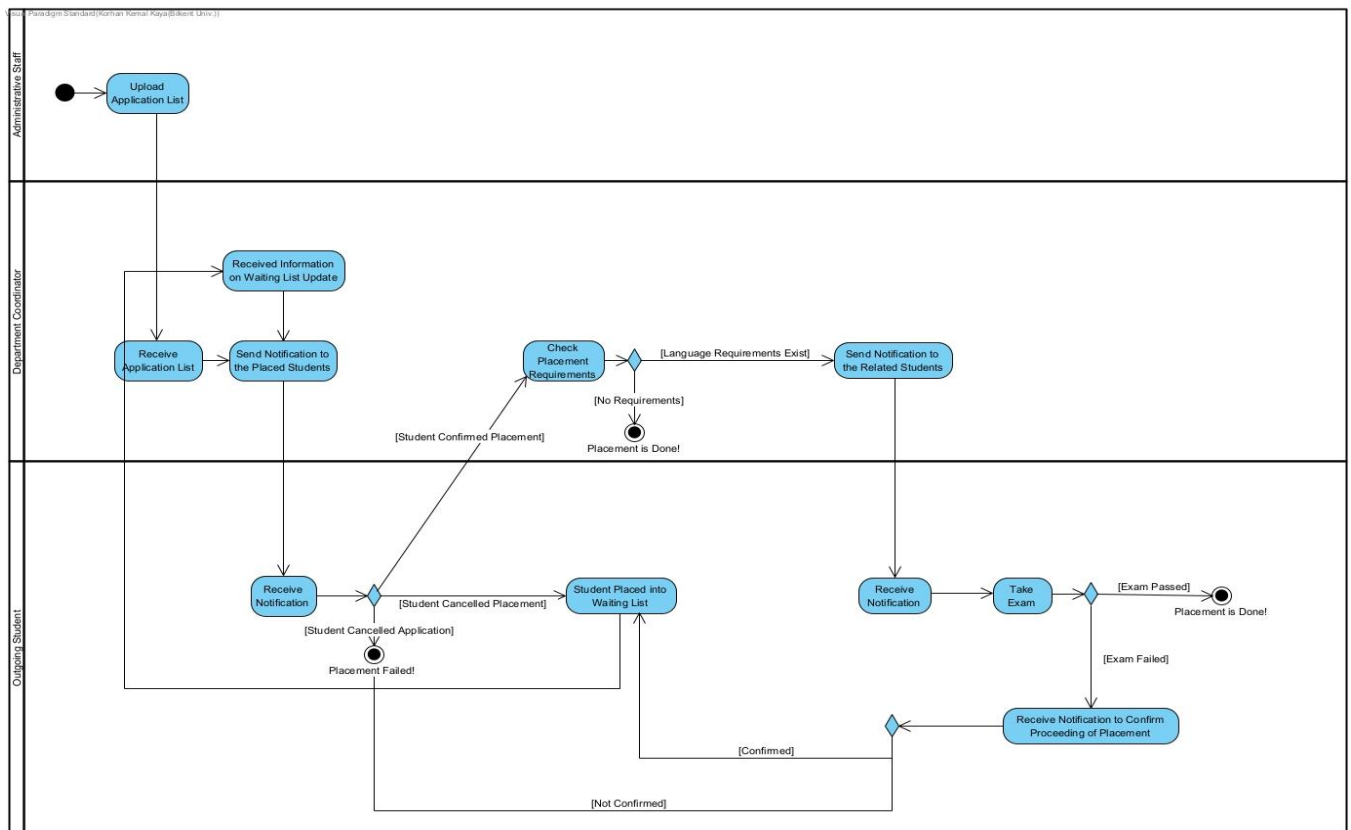
The HostUniversity represents the university that uses this application for its Erasmus and Exchange process. The HostUniversity class has some information about courses and students that applied to Erasmus and Exchange. The HostUniversity class has one-to-many association with Department class and Department class has one-to-many association with Course class for course related information. The HostUniversity class also represents the announcements made by the DepartmentCoordinator. Hence, The HostUniversity class has a

one-to-many association with the Announcement class. The Announcement class is used to represent the details of the announcements. The HostUniversity has more than one contracted university to change students. Therefore the HostUniversity class has a one-to-many association with the ContractedUniversity class. The ContractedUniversity class represents details of the university, and it consists of two classes: ExchangeUniversity and ErasmusUniversity. The ExchangeUniversity class contains quota and department name information. The ErasmusUniversity class contain quota information. ContractedUniversity has a one-to-many association with the ContractedUniDepartment class which is a child class of the Department class. The ContractedUniDepartment class is used to represent the quota and name information of the departments

## 2.3.3 Dynamic Models

### 2.3.3.1 Activity Diagrams

#### 2.3.3.1.1 Student Placement Activity

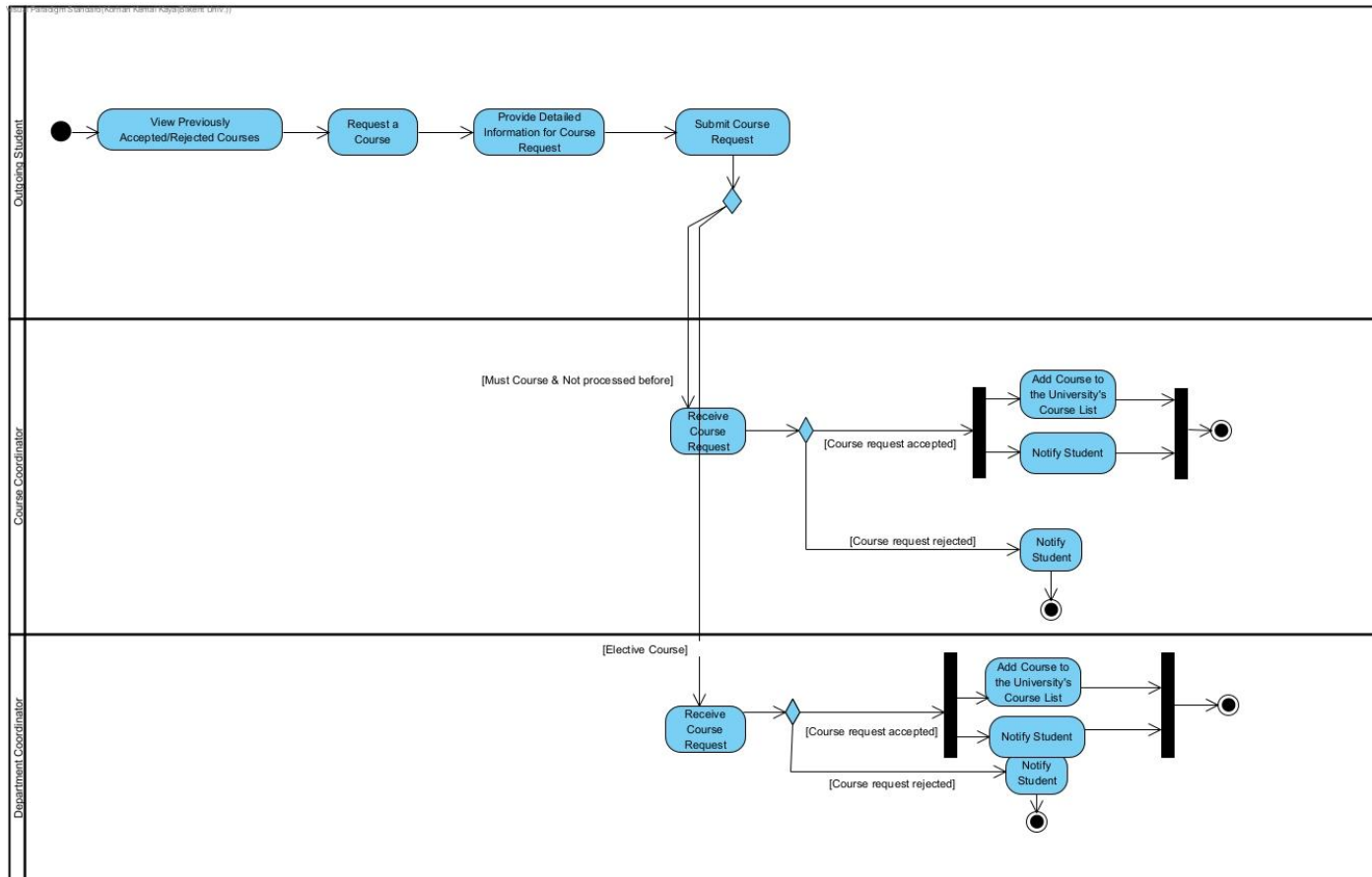


**Fig. 3: Student Placement Process Activity Diagram**

After the mobility programs' applications were received by the Administrative Staff, they uploaded the list to the system. After the Department Coordinator gets the application list, the system does the placements. Observing the placed students, the Department Coordinator can notify them about their status. Students that receive the notification may or may not continue the process. If a student does not confirm the placement, based on their wish, either they will be placed into a waiting bin, or their placement will fail. If a student confirms their placement, then language requirements for the school are checked. According to the result, the

student may get a notification for sufficiency in the language exam, including information about the exam time, place, etc. If a student cannot pass the exam, they will be asked whether they want to be placed into the waiting bin. If there is any change in the waiting bin, the department coordinator will be notified.

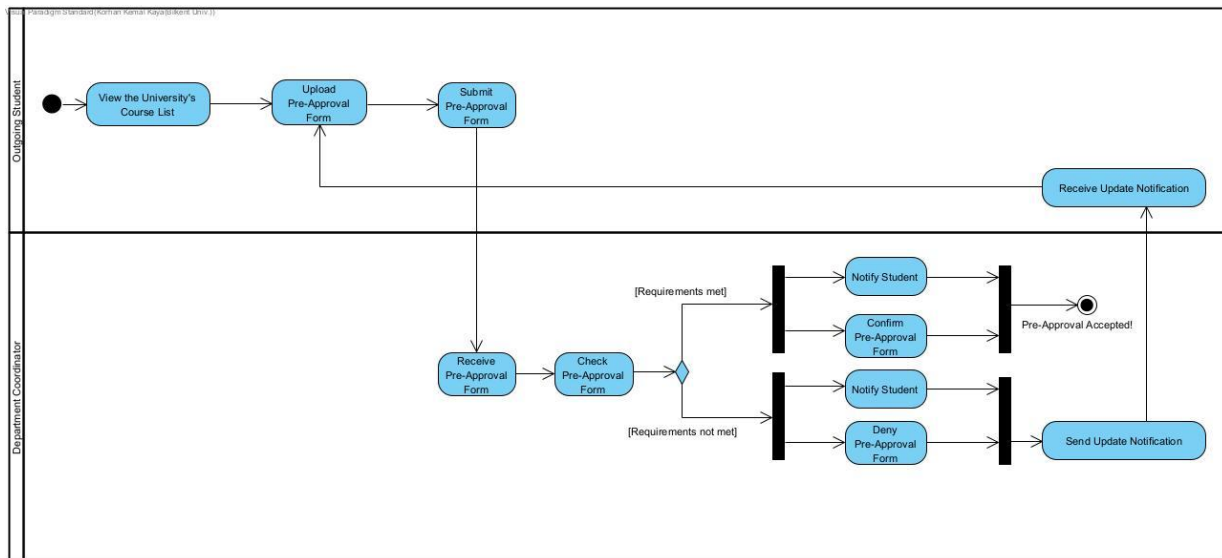
### 2.3.3.1.2 Course Request Activity



**Fig. 4:** Course Request Activity Diagram

Before requesting a course, a student should see the previously accepted courses of the partner university. This will make a student avoid creating a course request for a course already accepted. After creating a course request, a student should specify its ECTS credits, syllabus, workload (whether a project exists or how many midterms it has), and the Bilkent course to be exempted. When the specification is done, the student can submit the course request. If the Bilkent course to be exempted is a must and not accepted before for the relative course in the partner university, the course coordinator receives the request. If it is an elective course, the department coordinator gets the request. After examining the courses, they can either accept the request and add the course to the partner university's course list or reject it. In both cases, the student will get notified.

### 2.3.3.1.3 Pre-Approval Form Creation Activity

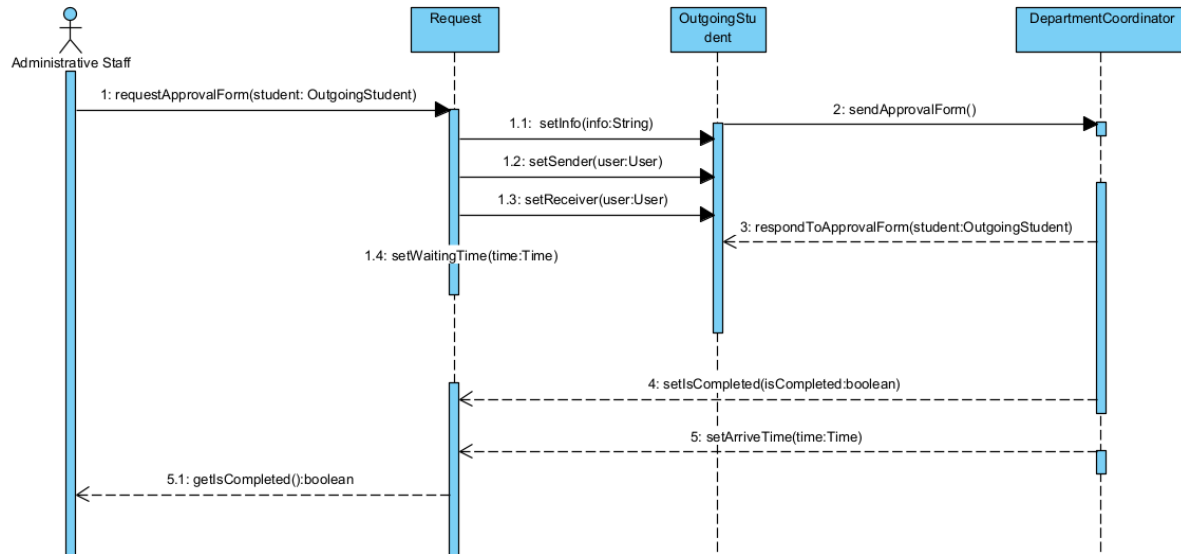


**Fig. 5:** Pre-Approval From Creation Process Activity Diagram

Same as the course request activity, before creating a pre-approval form, the student should see the university's course list to avoid filling out pre-approval form with a course that is not accepted before. After that, the student can upload the pre-approval form and submit it. Receiving it, the department coordinator can check if the requirements are met. If they are not met, the student will get notified to update it and upload it again to the system to process continue.

### 2.3.3.2 Sequence Diagram



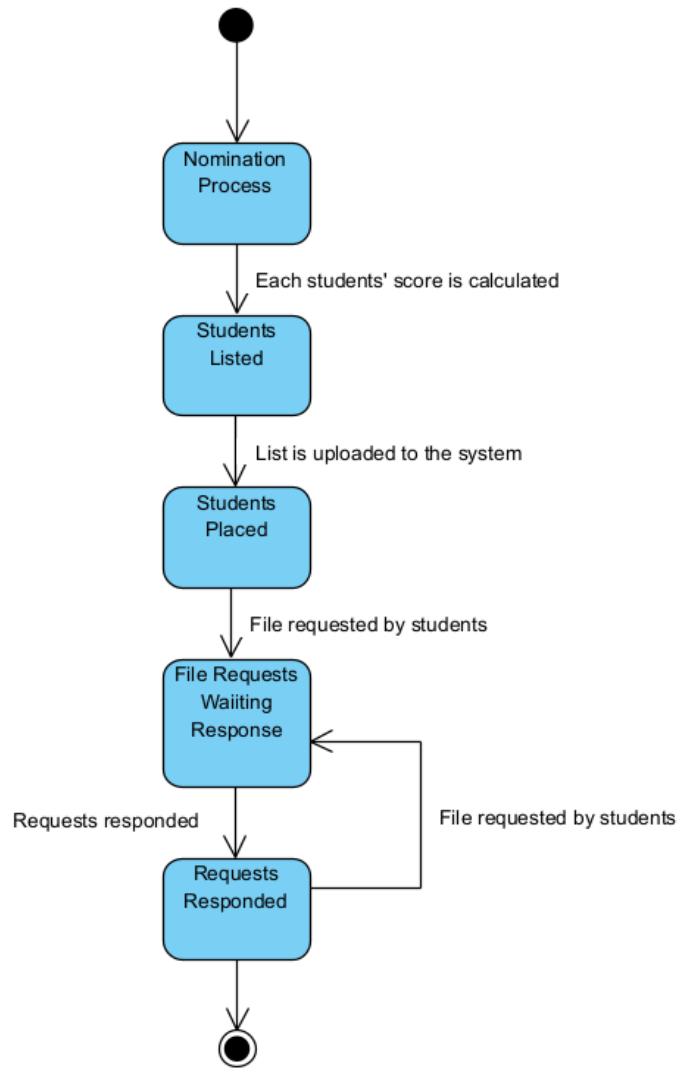


**Fig. 6: Approval Form Submission Sequence Diagram**

Order of approval form submission process is shown in the above sequence diagram. First, the Administrative Staff actor creates a Request object for approval form by calling requestApprovalForm() function, after that, in the Request object, info, waiting time, sender and receiver information of the request is set with setters. After a Request is created with these attributes, the Request object goes to the receiver, which is OutgoingStudent object. OutgoingStudent sends the approval form to the DepartmentCoordinator object with sendapprovalForm() function. DepartmentCoordinator responds the approval form and the response goes back to OutgoingStudent. After the response is given, arrival time and isCompleted setter functions are called and go back to the Request object. Request object returns a boolean to the Administrative Staff user after the process is completed.

### 2.3.3.3 State Diagrams

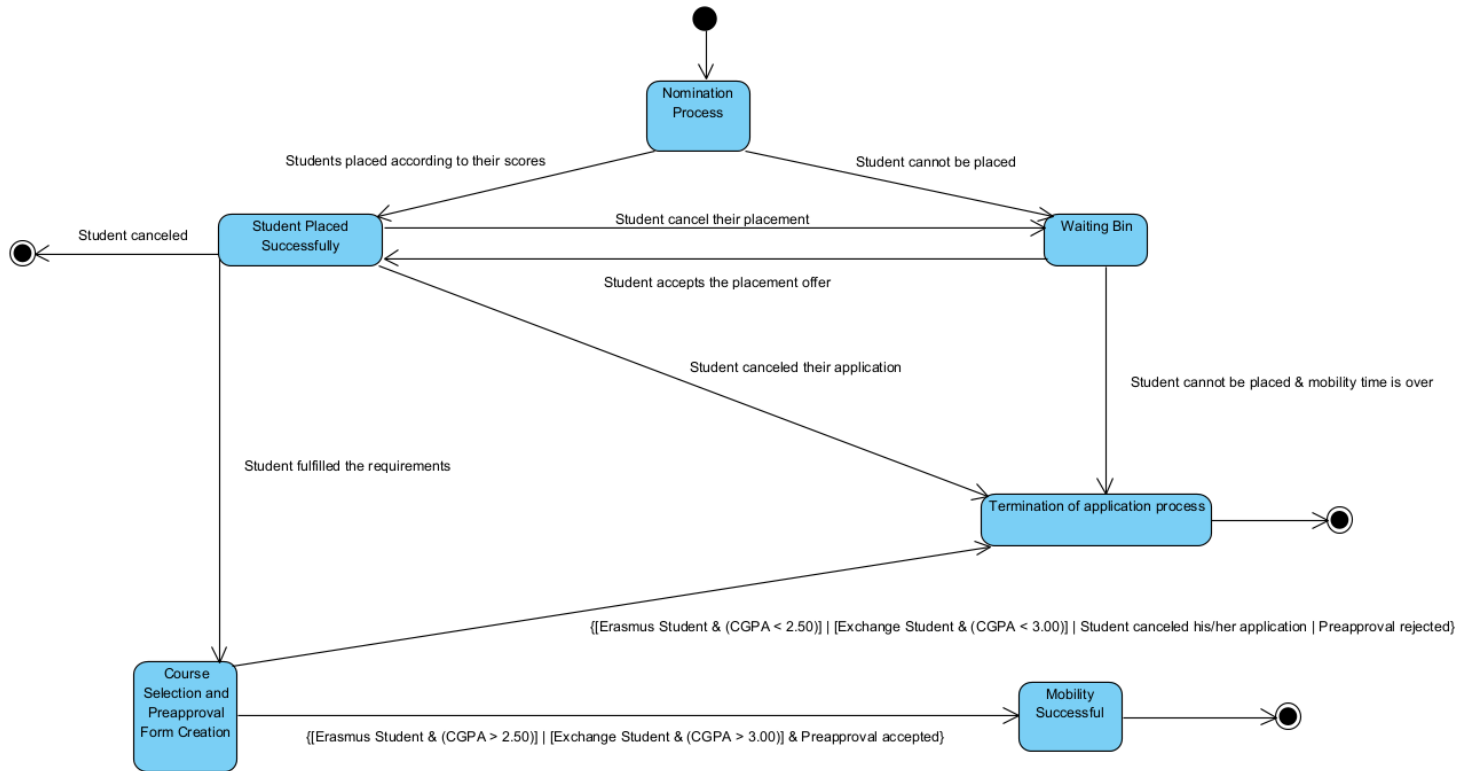
#### 2.3.3.3.1 Administrative Staff's State Diagram



**Fig. 7:** *Administrative Staff's State Diagram*

This diagram illustrates the states of the Administrative Staff role in our app. At first, after the nomination process is completed, the students, who want to go to the Erasmus/Exchange program, are listed and sorted according to their placement scores calculated. Once the students are listed, administrative staff upload this list to the system, and the state changes. In the next state, students who have applied to the Erasmus/Exchange program are placed at their universities. Once the students are placed in their universities during their application process, any student can request files from administrative staff, and this changes the state again. In this state, administrative staff responds to these requests and each time a new file request comes from students, this state occurs again. Once the application process ends, the state moves to the final state.

### 2.3.3.3.2 Outgoing Students' Placement Process State Diagram

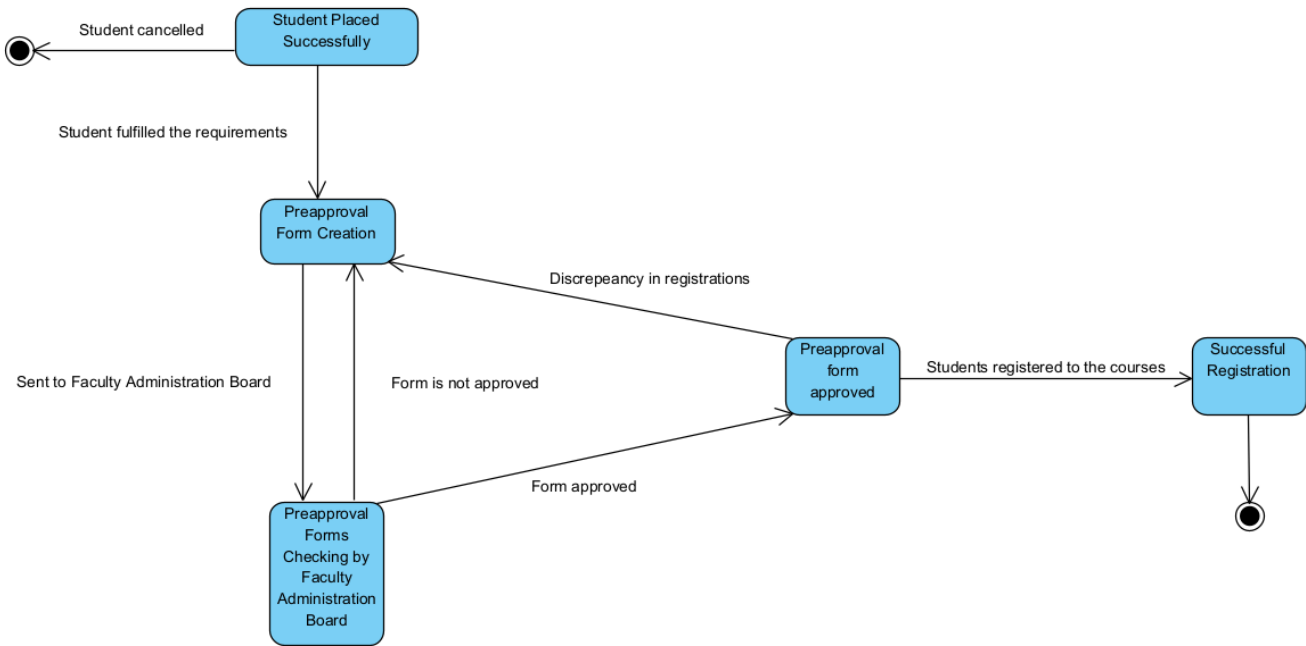


**Fig. 8:** Outgoing Students' Placement Process State Diagram

This diagram illustrates the state of the pre-approval process of an outgoing student. At first, after the nomination process is completed, the students, who want to go to the Erasmus/Exchange program, are listed and sorted according to their scores calculated, and according to these scores, students are placed in the universities. At this point, a student's state can be "Placed successfully" or "Waiting Bin", depending on their score and university choices. If the student's state is "Waiting Bin" and there is a placement offer coming to them, they can move to the "Placed successfully" state if they accept the offer. On the other hand, if they decline the placement offer or there is no placement offer coming to them until the end of the deadline, they move to the "Termination of the process" state. Once the student is placed successfully their university, they can move to the final state by canceling their application. Also, the student can move to the "Waiting Bin" state by canceling their placement to be placed at another university. If they have not canceled his/her application or placement and fulfilled

the requirements, they move to “Course Selections and Preapproval Form Creation” state and unless they cancels their application or their preapproval rejected or their CGPA is below the limit (2.50 or 3.00 depending on the whether the student is an Erasmus student or Exchange student) and moves to the “Termination of the Process” state, they move to “Mobility successful” state. Once the student is in the “Mobility successful” state or “Termination of the Process” state, their state goes to the final state.

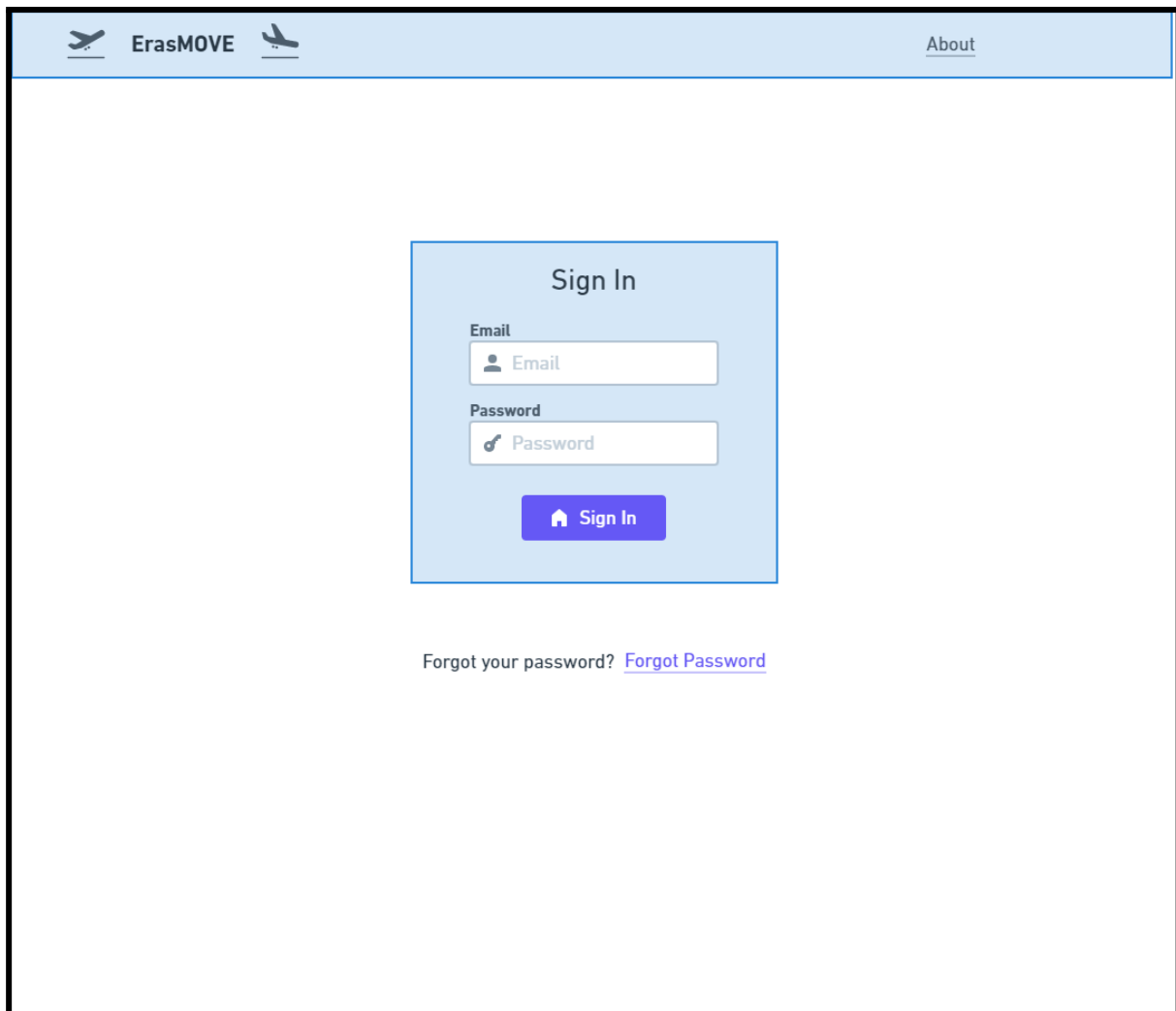
### 2.3.3.3.3 Outgoing Students’ Pre-approval Process State Diagram



**Fig. 9:** Outgoing Students’ Pre-Approval Process State Diagram



This diagram illustrates the state of the placement process of an outgoing student. Once the student is placed successfully at his/her university, s/he can move to the final state by canceling his/her application. If s/he has not canceled his/her application or placement, s/he moves to the “Pre-approval form creation” state once s/he starts selecting courses. In this state, students can move to “Pre-Approval forms checking by the Faculty Administration Board” by sending the pre-approval form to the Faculty Administration Board. In this state, if the pre-approval form is approved, the student moves to the “Pre-Approval form approved” state. On the other hand, if the pre-approval form of the student is not approved, the student moves to the “Pre-Approval form creation” state. Once the student is in the “Pre-approval form approved” state, unless there is a discrepancy in the registrations and the student registers for the courses, the student moves to the “Successful registration” state and moves to the final state from there. However, if there is a discrepancy in registrations, the student moves back to the “Pre-Approval form creation” state.

### 2.3.4 User Interface



The image shows a web application interface for a login page. At the top, there is a light blue header bar. On the left side of the header, there is a logo consisting of a stylized airplane icon followed by the text "ErasMOVE" and another stylized airplane icon. On the right side of the header, there is a link labeled "About". The main content area is white and contains a central light blue box with a blue border. Inside this box, the text "Sign In" is centered at the top. Below this, there are two input fields. The first is labeled "Email" and contains a person icon and the text "Email". The second is labeled "Password" and contains a key icon and the text "Password". Below these fields is a blue button with a white house icon and the text "Sign In". Below the central box, there is a link that says "Forgot your password?" followed by a blue underlined link labeled "Forgot Password".


*Fig. 10: Login Page UI*

 ErasMOVE 


[About](#)

### Forgot Password


Email

 Email


Confirmation code will send to your mail.

 Send


Confirmation Code


 Code

New Password

 Password

New Password Again



 Password

 Change

Go back to login. [Login Page](#)

Fig. 11: Forgot Password Page UI

37

 **ErasMOVE** 

[Profile](#) Application

Outgoing Student

Chat

Sign Out

## Profile


<b>Full Name:</b> John Doe	<b>Department :</b> Computer Science
<b>Email:</b> john.doe@ug.bilkent.edu.tr	<b>CGPA:</b> 4.0
<b>Student ID :</b> 21901881	<b>Semester:</b> 6
<b>Language Proficiency :</b> English C1	

### Announcements

Placement has been made you can check your application!

If you got accepted, please upolad your preapproval forms!

#### Request Sent

Waiting response for preapproval form 

Create New Request

#### Request Received


Preapproval form request 

Fig. 12: Outgoing Student Profile Page UI

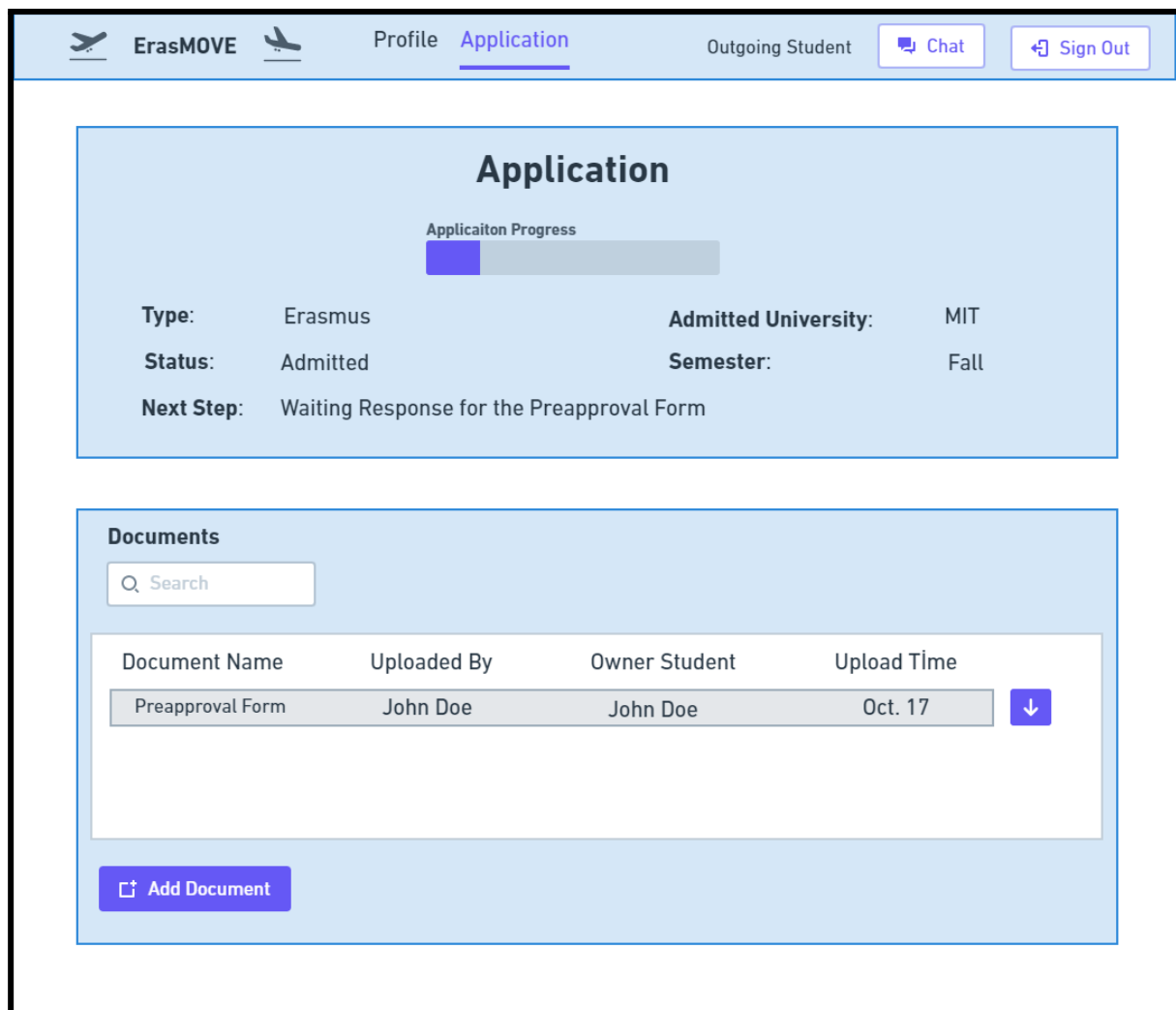


Fig. 13: Outgoing Student Application Page UI



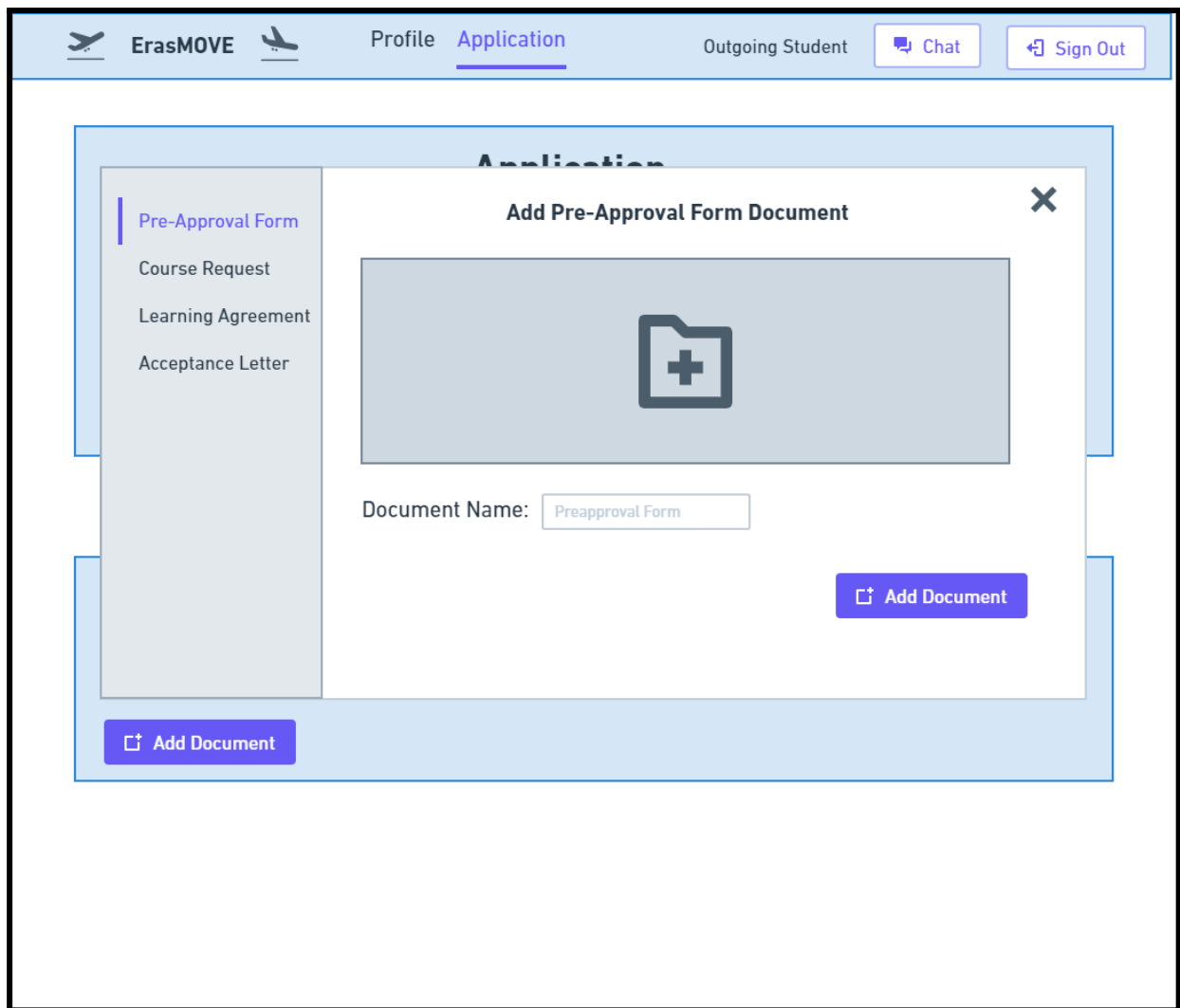




Fig. 14: Add Pre-Approval Document Modal UI

 **ErasMOVE** 

Profile Application

Outgoing Student

Chat

Sign Out

Application

Pre-Approval Form


**Course Request**

Learning Agreement

Acceptance Letter

Add Document

Add Course Request Document



Document Name:

Equal Course:

Description:

Add Document

Fig. 15: Add Course Request Document Modal UI

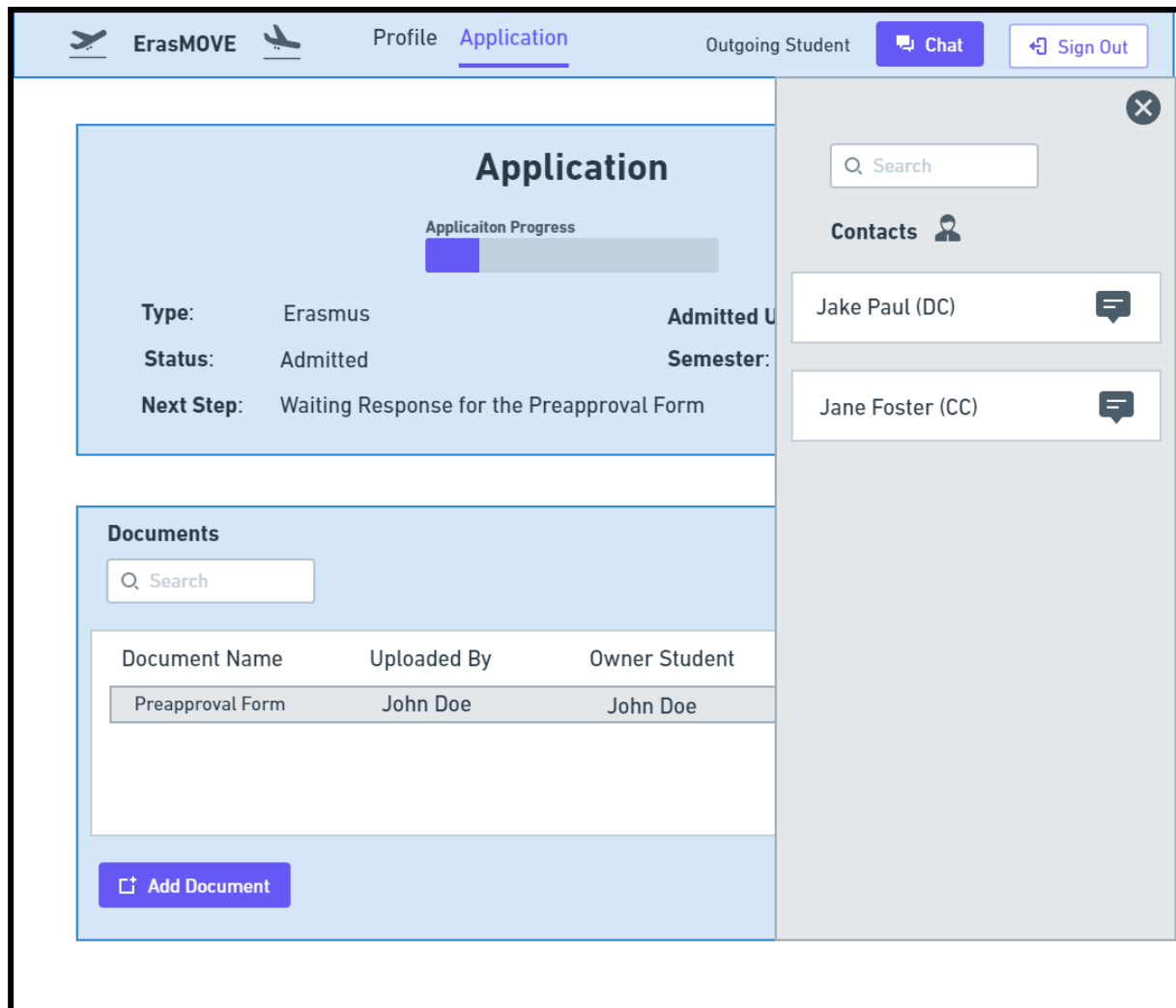


Fig. 16: Chats Modal UI

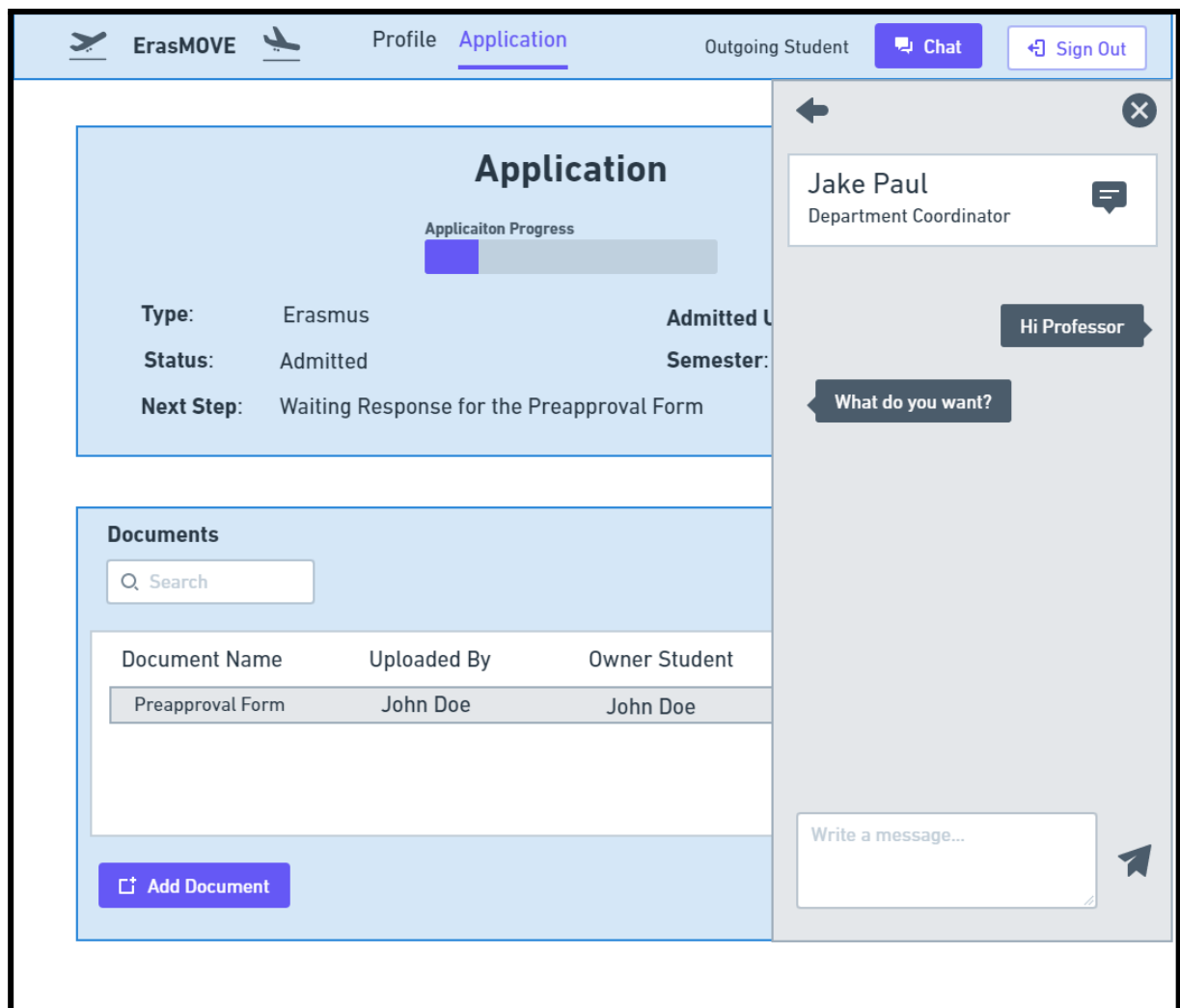




Fig. 17: Chat Modal UI

 **ErasMOVE** 

Department Coordinator [Chat](#) [Sign Out](#)

**Placement List**

**Waiting Bin**

**Announcements**

**To Do List**

### Placement List















	Surname	Name	Placement Score	Placed University	
1	Güzelkaya	Kürşad	99.8	MIT University	
2	Doe	John	98.2	MIT University	
3	Tekin	Gökhan	97.1	Kingston University	
4	Buluş	Hazal	96.0	University of Saarland	
5	Perez	Juan	95.1	University of Saarland	
6	Zhang	San	91	Bamberg University	
7	Güzelkaya	Kürşad	87.5	Bamberg University	
8	Güzelkaya	Kürşad	84.2	Bamberg University	
9	Güzelkaya	Kürşad	83	Bamberg University	
10	Güzelkaya	Kürşad	81.9	Bamberg University	

Fig. 18: Placement List Page UI

 **ErasmusMOVE** 

Department Coordinator

 Chat

 Sign Out

Placement List

Waiting Bin

Announcements

To Do List

Q Search

Total empty quotas: 5











	Surname	Name	Score		
1	Güzelkaya	Kürşad	75	Send Placement Offer	
2	Doe	John	72	Send Placement Offer	
3	Tekin	Gökhan	71	Send Placement Offer	
4	Buluş	Hazal	68	Send Placement Offer	
5	Perez	Juan	64	Send Placement Offer	
6	Zhang	San	60	Send Placement Offer	
7	Güzelkaya	Kürşad	57.5	Send Placement Offer	
8	Güzelkaya	Kürşad	54.2	Send Placement Offer	
9	Güzelkaya	Kürşad	53	Send Placement Offer	
10	Güzelkaya	Kürşad	51.9	Send Placement Offer	

Fig. 19: Waiting Bin Page UI

ErasMOVE

Department Coordinator

Chat

Sign Out

### Profile

**Full Name:** John Doe
**Department :** Computer Science

**Email:** john.doe@ug.bilkent.edu.tr
**CGPA:** 4.0

**Language Proficiency :** English C1
**Student ID :** 21901881

Add Language

**Semester:** 6

### Application

Applicaition Progress

**Type:** Erasmus
**Admitted University:** MIT

**Status:** Admitted
**Semester:** Fall

**Next Step:** Waiting Response for the Preapproval Form

### Documents

File Name	Uploaded By	Owner Student	Upload Time	
Preapproval Form	John Doe	John Doe	Oct. 17	↓

Add Document

Fig. 20: Student Details Model UI

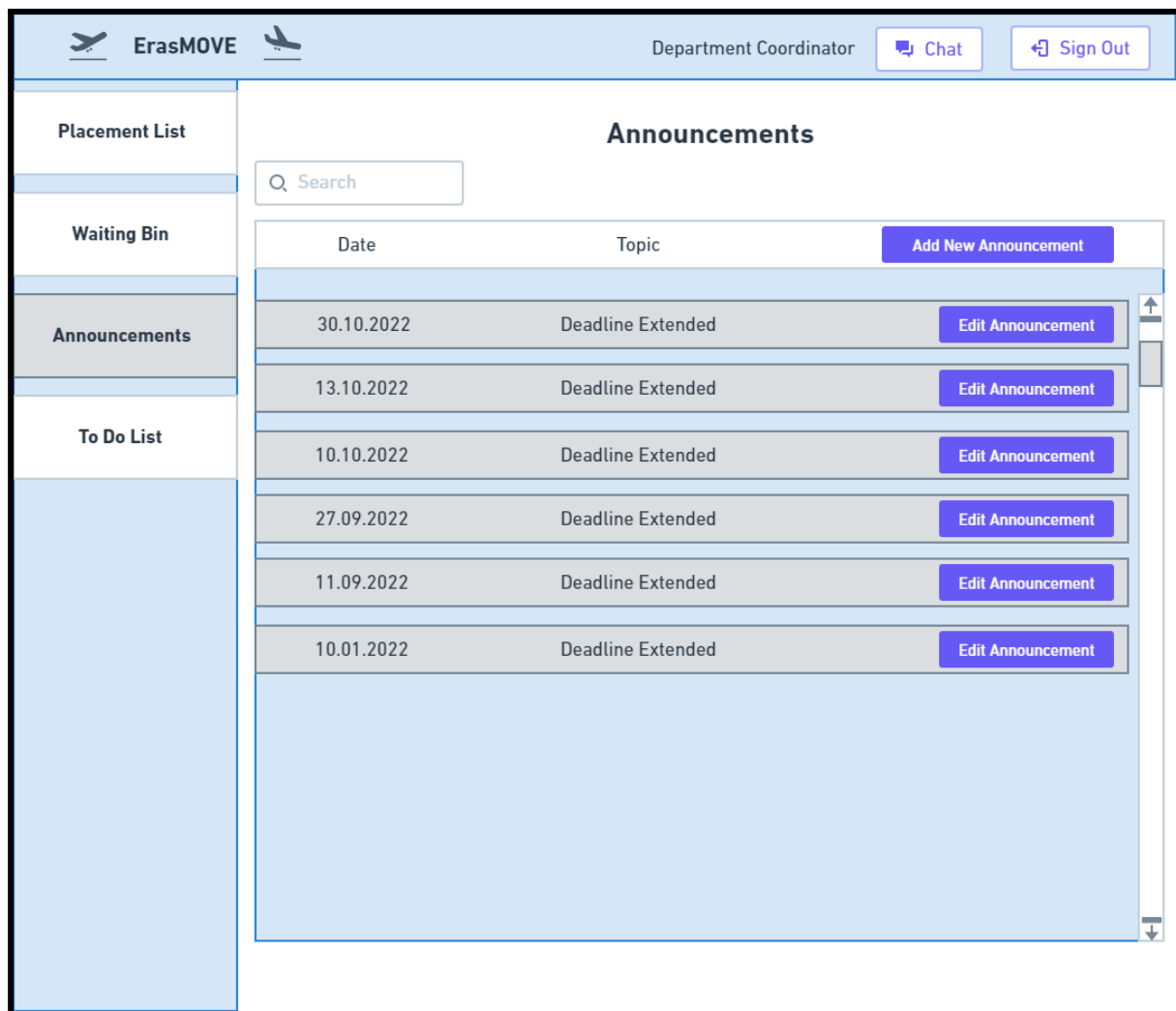




Fig. 21: Announcement Page UI



 **ErasMOVE** 

Course Coordinator

Chat

Sign Out

Course Requests

To Do List

### Course Requests












Surname	Name	Course Request File		
Güzelkaya	Kürşad	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Doe	John	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tekin	Gökhan	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Buluş	Hazal	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Perez	Juan	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Zhang	San	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Güzelkaya	Kürşad	 CourseReq.pdf	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fig. 22: Course Requests Page UI

 ErasmOVE 

Course Coordinator

 Chat

 Sign Out

Placement List

Waiting Bin

Announcements

To Do List

### To Do List

☐ You have 3 Pre-Approval Form requests.

30.10.2022







☒ You completed 21 Learning Agreement requests.

30.10.2022

#### Request Received

Pre-Approval

Learning Agreement

Pre-Approval Form	John Doe	Oct. 17		
Pre-Approval Form	Jake Smith	Oct. 19		
Pre-Approval Form	Kürşad Güzelkaya	Oct. 21		

#### Request Sent


 Create New Request

Fig. 23: To-Do List Page UI

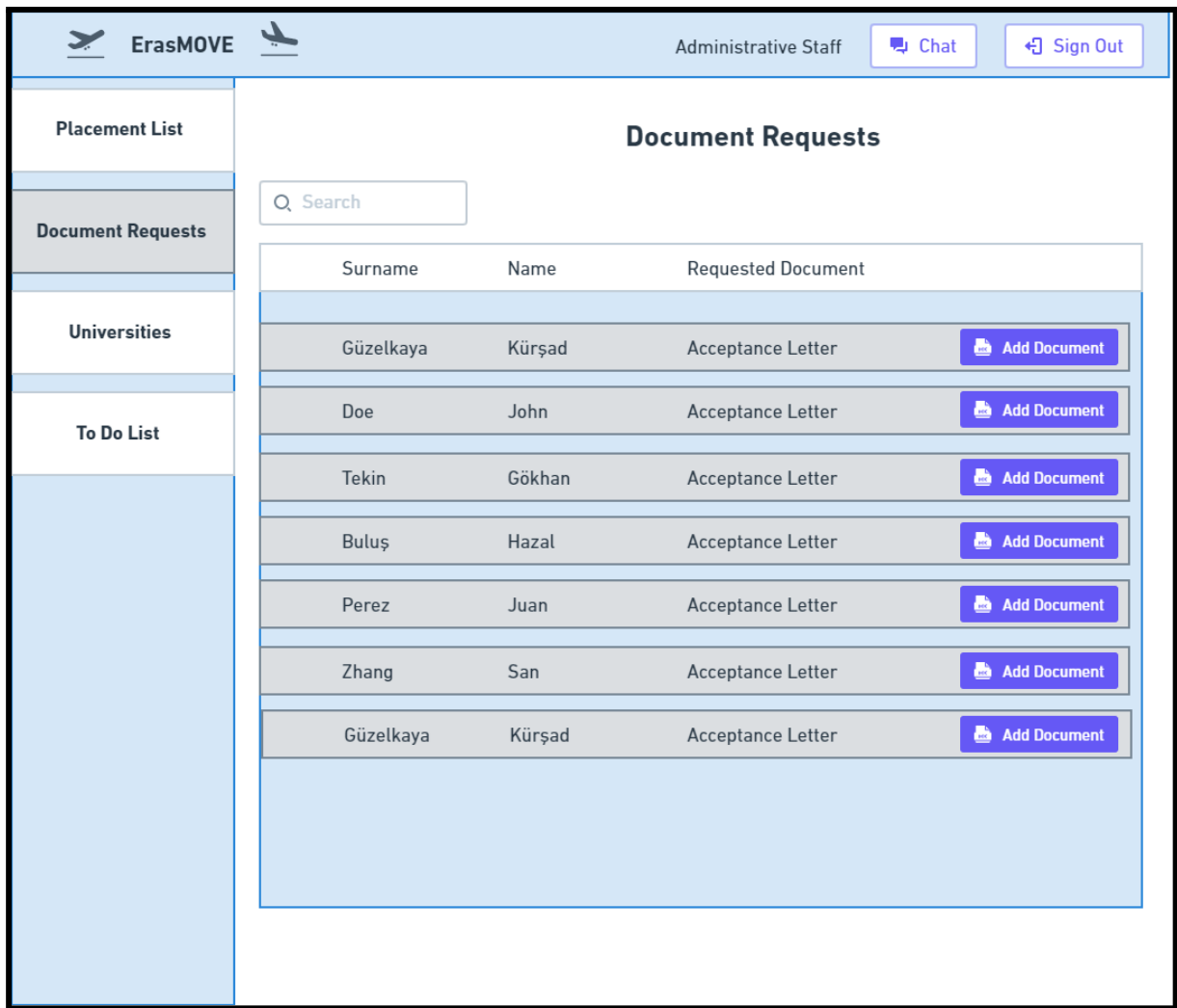




Fig. 24: Document Requests Page UI

 ErasMOVE 

Profile Proposal

Incoming Student

Chat

Sign Out

### Course Proposal

Course 1:

Course Name

Course 2:

Course Name

Course 3:

Course Name

+

 Add More Course

Submit

Fig. 25: Course Proposal Page UI

### 3. Glossary and References

- [1] “Exchange Programs,” [www.cs.bilkent.edu.tr/~exchange/](http://www.cs.bilkent.edu.tr/~exchange/). [Online]. Available: <http://www.cs.bilkent.edu.tr/~exchange/> [Accessed: Nov. 03, 2022].
- [2] “Erasmus+ online application and Management System - Exchange Office - Bilkent University,” *Bilkent University | Erasmus+ Application System*. [Online]. Available: <https://app.erasmus.bilkent.edu.tr/>. [Accessed: Oct.31, 2022].
- [3] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

## 4. Appendix

### Appendix A - Interview with Yelda İrem Ateş

*This is a translated version of our interview with Yelda İrem Ateş, who is the Exchange Coordinator of the Faculty of Engineering. P1, P2, and P3 represent our team members.*

**P1:** Hello, Mrs. Ates. Firstly, thank you for taking your time for us. We will make an Erasmus/Exchange Application Manager program. Since you are part of this process, we wanted to get information about your role and make sure that our implementation would be beneficial for you.

**Ates:** It is great to hear that. An application that can digitalize and autonomize some work of ours will be so appreciated. How can I help you?

**P2:** Well. We believe there are different exchange coordinators for different faculties. Could you please start by expressing the role of an exchange coordinator in the placement process?

**Ates:** Yes, as you assume, there are different coordinators. As an exchange coordinator, my main task is making sure the process goes smoothly for everyone involved and ensuring the flow of necessary files. Most importantly, we nominate students to the universities they will be going to. Also, after nomination, if there is any obstacle or need, we keep track of the communication with the partner university. Other than that, we have some crucial roles in the placement process. For example, after Registrar's Office uploads the applications for mobility programs, the Department Coordinators and we will be responsible for placing students manually. Another one is that we make sure that students get their feedback on pre-approval forms and whether they update it or not. Also, if there is a new agreement with a university or the quota of some university has changed, we are the ones who update the system.

**P3:** Thank you! Could you list some of the "essential functions" that you expect from this application and that would be most beneficial?

**Ates:** During the process, many files can be digitalized. Especially, Pre-Approval forms and learning agreement forms are hand-filled and hand-delivered. This situation makes it difficult to track every file. Also, sometimes some students might have forgotten to bring their forms. This endangers their placement process. Thus, a system that those files can be uploaded and tracked digitally and also can send notifications to students may be very helpful. Also, it will be great to see some processes are autonomized, such as placing the applied students in partner universities. Is there another question?

**P2:** No, actually not. Your answers were very helpful. Again, thank you for your time and help!

**Ates:** It was nice to meet you guys. I am looking forward to see your application.