



Bilkent University

Department of Computer Engineering

CS319 Term Project

Section 2

Group 2G

toera

Requirement Analysis Report

Group Members:

- İlker Özgen - 21902719
- Onurcan Ataç - 22002194
- Kutay Şenyiğit - 21902377
- Utku Kurtulmuş - 21903025
- Mert Ünlü - 22003747
- Ömer Oktay Gültekin - 21901413

Instructor:

Eray Tüzün

TA's:

- Muhammad Umair Ahmed
- Emre Sülün
- İdil Hanhan
- Mert Kara
- Metehan Saçakçı

Table of Contents

1. Introduction	4
2. Proposed System	4
2.1 Functional Requirements	5
2.1.1 Login	5
2.1.2 Dashboard	5
2.1.3 Submit Pre-Approval Form	6
2.1.4 Discard Placement	6
2.1.5 Display Previous Courses	6
2.1.6 Create Pre-Approval Form	6
2.1.7 Add to Pre-Approval Form	6
2.1.8 Remove From Pre-Approval Form	6
2.1.9 Send Pre-Approval Form	6
2.1.10 Request New Course	6
2.1.11 Report Problem	6
2.1.12 Review University	6
2.2 Non-functional Requirements	8
2.2.1 Usability	8
2.2.2 Reliability	9
2.2.3 Manageability	10
2.2.4 Time Cost	10
2.2.5 Sustainability	10
2.2.6 Availability	10
2.3 System Models	11
2.3.1. Use Case Model	11
2.3.2. Object Models and Class Diagram	20
2.3.3. Dynamic Models	23
2.3.3.1 Activity Diagrams	23

2.3.3.1.1 Create User Activity	23
2.3.3.1.2 Submit Pre-approval Form Activity	24
2.3.3.2 Sequence Diagram	25
2.3.3.3 University State Diagram	26
2.3.3.4 WishedCourse and Course State Diagram	27
2.3.3.5 Application State Diagram	28
2.3.4. User Interface	29
3. References	63

1. Introduction

Erasmus procedure in Bilkent has always been considerably disorganized and not connected due to the lack of a comprehensive system. Emailing is the main communication method during most of the Erasmus procedure, which leads to difficulties in keeping track of and accessing relevant information.

Therefore, a web-based Erasmus Application, *toera*, that enables all users (Students, Coordinators, etc.) to interact with each other and push the procedure forward easily without errors was needed. With this Erasmus manager, the aim is to minimize the use of emailing and other inefficient methods of communication and enable all users to be on the same page at all times. Viewing application and approval status, uploading forms, asking relevant questions, and viewing course information are all part of the system. Due to the nature of the Erasmus procedure, different kinds of users are involved in the system. The abilities, allowed features, and views depend on the user type, meaning that while a student can not approve a given form, the corresponding coordinator can. Users can also generate and send all the needed forms for every stage of the procedure. Notifications, task lists for students and coordinators, and direct messaging features are all components of the system to prevent any possible delay and difficulty.

2. Proposed System

Erasmus Application Manager aims to ease the Erasmus application process at Bilkent University by providing a user-friendly user interface and various features. The Erasmus coordinators, course instructors, department secretaries, international student office, and faculty administration committee members will be given accounts according to their roles. The system admins will create these accounts. After the first applications are made, the students eligible to attend the Erasmus program will also be provided with accounts. The system will automatically create the account using the ranking data provided by the international student office.

The dashboard page will be opened when a user signs in to the application. The user will be able to navigate to the other pages from this page. Students will be able to see their application status, to-do list, messages, and notifications in the dashboard. The other types of users can view their to-do lists, messages, and notifications in the dashboard. The users will also be able to view each others' profile pages.

Inside the application page, the students will be able to see their application status, cancel their applications, or discard their placements. Using the pre-approval form page, the students will be able to add new courses, convert the form to PDF, and send their pre-approval form. The Erasmus coordinators will be able to convert the form to PDF, edit the form, give feedback, and approve the pre-approval form on this page. Faculty administration members can also convert the form to PDF, edit the form, give feedback, and approve or reject the pre-approval form inside the page.

Using the add course page, students can propose a new course they would like to take. They can see the previously accepted courses and nominate a course. The Erasmus and course coordinators can accept or reject the exemption request.

On the learning agreement form page, the students will be able to view and edit the information and convert the form to PDF. Similarly, on the course transform page, they will be able to view and edit the document and convert it to PDF.

Users can view the relevant information and upload their profile pictures on their own profile pages. Users can view the relevant information on others' profile pages and message them using the button on the page. On a university and course profile page, the relevant university information will be shown along with the comments made on that university and the university's ranking in the app's review system.

On the settings page, the users will be able to change their passwords and E-Mail addresses that are registered to the system. They will also be able to disable notifications and prefer to receive E-Mails for notifications.

On the chat page, the users can send and receive messages to each other and search for other users in chat.

2.1 Functional Requirements

2.1.1 Login

The login screen allows students to enter the application with their university mail addresses. The initial account information will be sent to the student's email with a random password. After logging in the first time into the application with the given random passwords via email, students will be able to change new passwords.

2.1.2 Dashboard

2.1.2.1 Application Status

Students will be able to see the current status of their application. They can also review the stages they have to complete in the future or the stages they have passed.

2.1.2.2 TODO

Course coordinator, Erasmus coordinator, international student office, department secretary, students, and faculty administration committee members will be told what task and when they have to do a task. If the due date of the assignment is closed, they will be notified via mail or in-app notification.

2.1.2.3 Last Messages

The users will be able to see the most recent unread messages on the dashboard.

2.1.2.4 Last Notifications

The users will be able to see the most recent notifications on the dashboard.

2.1.3 Submit Pre-Approval Form

Students will be able to enter required information in order to create and submit their pre-approval form in the application.

2.1.4 Discard Placement

It allows students to discard their placement after they have been placed in a university.

2.1.5 Display Previous Courses

Students will be able to see the previously approved courses on the platform.

2.1.6 Create Pre-Approval Form

Students can create a wishlist to decide on their courses during the Erasmus period. Wishlists were created before submitting the pre-approval form to get feedback from the Erasmus coordinator.

2.1.7 Add to Pre-Approval Form

Students can add the courses they want to take to the wishlist.

2.1.8 Remove From Pre-Approval Form

Students can remove a course from the wishlists.

2.1.9 Send Pre-Approval Form

Students can send their wishlists to the Erasmus coordinator to get feedback.

2.1.10 Request New Course

Students can request a new course to be opened in the Erasmus period.

2.1.11 Report Problem

Students can report the problems they have faced while using the application.

2.1.12 Review University

Students can see the reviews of universities by former Erasmus students.

2.1.13 Approve Nominated Courses

Course coordinators can approve the nominated course if the course matches the expected requirements.

2.1.14 Reject Nominated Courses

Course coordinators can reject the nominated course if the course does not satisfy the expected standards.

2.1.15 Approve Course Transfer Form

Faculty administration committee members can review the final results of courses that students took during the erasmus period. If the results satisfy the expected requirements, the committee member can approve the course transfer form.

2.1.16 Reject Course Transfer Form

Faculty administration committee members can review the final results of courses that students took during the Erasmus period. If the results do not satisfy the expected requirements, the committee member can reject the course transfer form.

2.1.17 Active Application Period

Erasmus coordinators can start the application period.

2.1.18 Upload Erasmus Placement Result

Erasmus coordinators can upload the Erasmus placements results to the system.

2.1.19 Notify Placement

Erasmus coordinators can notify the students regarding the replacement results after they have uploaded the Erasmus placement results to the system.

2.1.20 Accept Pre-Approval Form

Faculty administration committee members accept the form if it satisfies the expected requirements.

2.1.21 Reject Pre-Approval Form

Faculty administration committee members can review the pre-approval form and decide to reject the form if it does not satisfy the criteria.

2.1.22 Upload Signature

Erasmus coordinators can upload their signatures to the system to use in future approval processes.

2.1.23 Display Logs

Notable activities done by actors will be kept and displayed for the Erasmus Coordinators.

2.1.24 Add University

System admins can add a new university to the system if there is a new Erasmus agreement with the university.

2.1.25 Remove University

System admins can remove a university from the system if the Erasmus agreement with the university is outdated.

2.1.26 Edit University Information

System admins can update the university information.

2.1.27 Submit Learning Agreement

Erasmus candidates can upload their learning agreement document so that the coordinators can view.

2.1.28 Nominate Students

Erasmus coordinators can nominate the students according to their placements.

2.1.29 Cancel Application

Erasmus candidates, coordinators and international student office can cancel an application.

2.1.30 Upload Course Transfer Form

Course coordinators can upload their course transfer form to the system.

2.1.31 Upload Signature

Students and Erasmus coordinators can upload their signature in order to be used in documents.

2.2 Non-functional Requirements

2.2.1 Usability

The application is much more usable than the current system in terms of the following criteria:

2.2.1.1 Erasmus Information Accessibility

In the current system, Erasmus candidates need to search all Erasmus-related pages of Bilkent University to educate them on the Erasmus process. Even though some information, such as previously accepted courses, is not easy to find on the main pages of Bilkent University, one needs more likely to communicate with the Erasmus coordinator to get these documents. However, all Erasmus info will be included in the application so that candidates effortlessly find what they need

on the application.

2.2.1.2 Host Institution Course Selection

In the current system, candidates are given previously accepted courses with only the basic course info and exempt Bilkent course info for that course. Candidates are supposed to parse roughly 3500 row two spreadsheets, which also have some missing info, and create a wish list. However, the application will allow users to specify the Bilkent courses the candidates wish to be exempted from and show them the list of courses they can take from the host institution for the specified course, while the detailed course info is one click away from the candidate. Additionally, candidates can nominate new courses from the app with one click; the application will do the rest.

2.2.1.3 PaperWork

In the current system, candidates are supposed to fill the forms in paper format and give a copy of the forms to all that are involved in the process of that specific form. However, the application lets the users fill all the necessary parts for all forms that are needed throughout the Erasmus Process (Pre-Approval, Course Transfer, Learning Agreement). With the filled digital forms, the application generates the forms in the desired format with the given information by users. Authorized users can see the generated forms in PDF version. Moreover, the application notifies the related users when there is a change in the application status.

2.2.1.4 Communication

In the current system, communication is done through emails, making it very likely to miss some emails for the Erasmus coordinators because of the massive amount of mail load. The proposed system allows all users to communicate through the app via chats so that The Faculty Committee Member can give the feedback for the form directly to the candidate or the Erasmus coordinator can propose changes in the pre-application form to the candidate easily.

2.2.1.5 Task Management

The current system is especially challenging for the Erasmus coordinators to keep track of the tasks that need to be done for all candidates. The proposed system has features that enable all users to see the to-do list and notify all users of the deadline for the closest tasks. Even for authorized users, clicking on the candidate's application in the to-do list opens that applications page; therefore, authorized users can use the proposed system by just using the to-do list they have.

2.2.2 Reliability

- Necessary user information and credentials will be stored privately by encrypting algorithms. Course and University reviews section will be written anonymously.
- The proposed system will enhance the validity of the forms in terms of not containing any invalid courses by only allowing the addition of courses which have already been checked by responsible coordinators.
- The system shifts the email-based communication between coordinators and students to a more compact system that allows in application communicating and notifying related users so that

the chance of missing/not seeing a mail is reduced to minimum (There won't be missing an Erasmus related email between all other emails in this system.). By reducing the risk of missing mail, the procedure continues with less problems, with more reliability.

- The proposed system allows admin users to use all the functionality the proposed system gives so that the system is reliable in terms of human errors and lack of talent to use the system for some specific user. The user with some problem can always ask for help from the admin user.
- The proposed system will make the waiting bin and the accepted list automatically so that the system minimizes human error.

2.2.3 Manageability

The proposed system will regulate the Erasmus Applications of the entire Bilkent University. The system will manage the applications by the divide and conquer principle:

- Bilkent and the host institutions will be represented as their departments.
- The applications will be shared between the Erasmus coordinators by their host institutions.
- All host courses will be linked to the courses to be exempted from Bilkent University.

Also, the Erasmus coordinators will be able to add or remove the University and change the host department's quota and the semester availability of the University. Courses will also be editable by the Erasmus coordinators.

2.2.4 Time Cost

The current system is mainly on excel spreadsheets which should be migrated into the app, and the missing information should be corrected. However, the preparation time will be compensated once the proposed system starts to operate, since the system takes the excel sheet and generates JSON data from it which can be easily done in milliseconds. Naturally, the system can integrate the excel data to the app faster than the current system which includes manual work of the Erasmus Coordinator to determine which student should go to which University based on their stats.

2.2.5 Sustainability

The project's open-source code, diagrams, and reports will make the project sustainable by any computer engineer authorized as an admin.

2.2.6 Availability

The proposed system will be available for all actors participating in the Erasmus application process.

2.3 System Models

2.3.1. Use Case Model

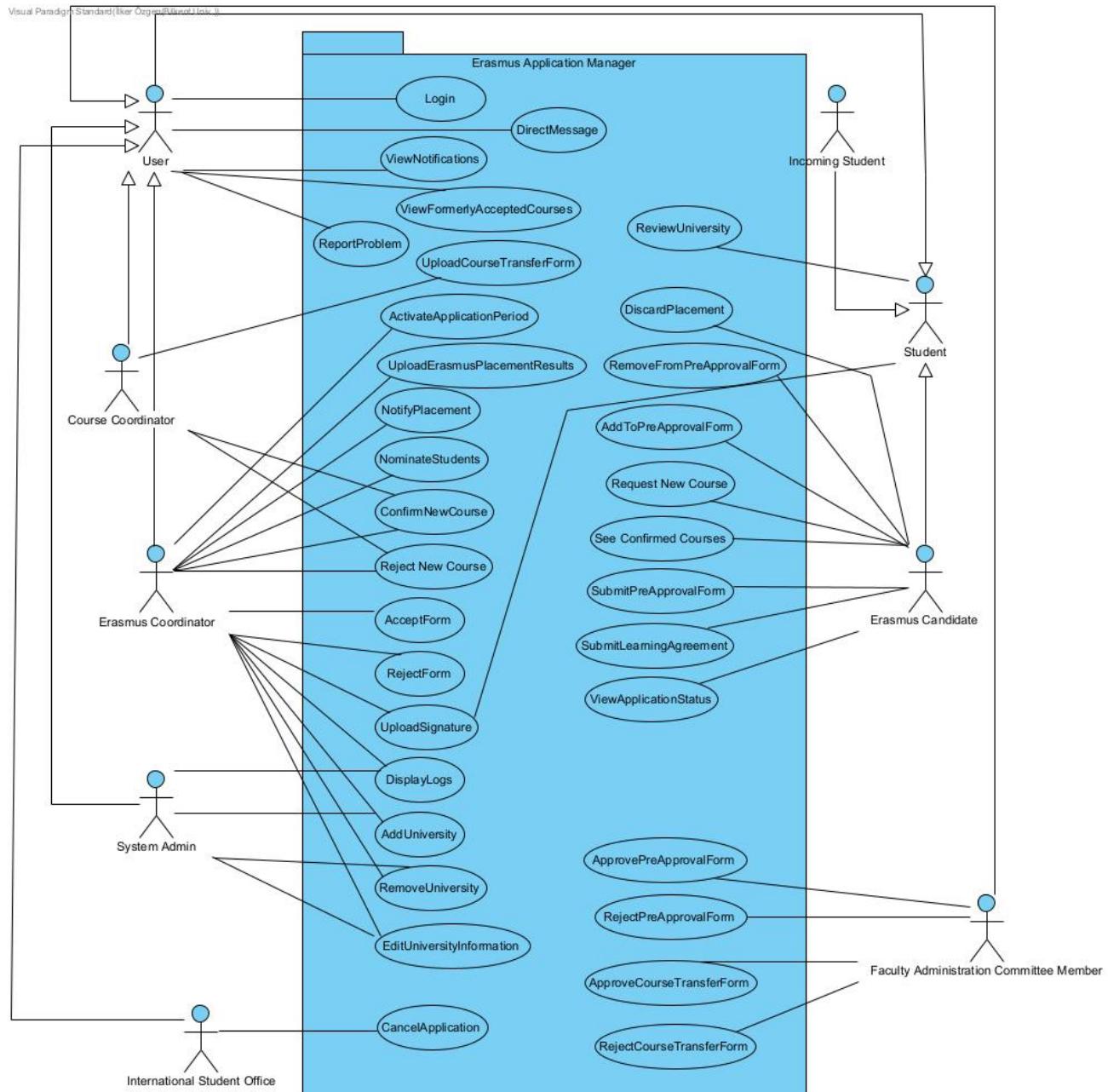


Fig. 1: General Use Case Diagram

Login

- **Use Case:** Login
- **Participating Actor:** User
- **Entry Conditions:** User is not logged in
- **Exit Conditions:** User clicks logout
- **The Flow of Events:**
- 1. User enters the username and password, and clicks login

ActivateApplicationPeriod

- **Use Case:** ActivateApplicationPeriod
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** none
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
- 1. Actor clicks activate button.

UploadErasmusApplicationResults

- **Use Case:** UploadErasmusApplicationResults
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** Application results are ready
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
- 1. Actor uploads the results
- 2. Actor clicks submit

NotifyPlacement

- **Use Case:** NotifyPlacement
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** Placements are ready
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
- 1. Actors clicks on notify button
- 2. System sends notification to the students

NominateStudents

- **Use Case:** NominateStudents
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** none
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
- 1. Actors selects the students and clicks on nominate

ConfirmNewCourse

- **Use Case:** ConfirmNewCourse
- **Participating Actor:** Course Coordinator
- **Entry Conditions:** a new course request has been made
- **Exit Conditions:** course is confirmed or rejected
- **The Flow of Events:**
 1. Actor views the course
 2. Clicks on confirm

RejectNewCourse

- **Use Case:** RejectNewCourse
- **Participating Actor:** Course Coordinator
- **Entry Conditions:** a new course request has been made
- **Exit Conditions:** course is confirmed or rejected
- **The Flow of Events:**
 1. Actor views the course
 2. Clicks on reject

AcceptForm

- **Use Case:** AcceptForm
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** form is submitted
- **Exit Conditions:** form is accepted or rejected
- **The Flow of Events:**
 1. Actor views the form
 2. Clicks on accept

RejectForm

- **Use Case:** RejectForm
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** form is submitted
- **Exit Conditions:** form is accepted or rejected
- **The Flow of Events:**
 1. Actor views the form
 2. Clicks on reject

UploadSignature

- **Use Case:** UploadSignature
- **Participating Actor:** Erasmus Coordinator
- **Entry Conditions:** none
- **Exit Conditions:** signature is uploaded
- **The Flow of Events:**

1. Actor uploads their digital signature

DisplayLogs

- **Use Case:** DisplayLogs
 - **Participating Actor:** Erasmus Coordinator, System Admin
 - **Entry Conditions:** logs are entered
 - **Exit Conditions:** operation is completed
 - **The Flow of Events:**
1. Actor views the logs

AddUniversity

- **Use Case:** AddUniversity
 - **Participating Actor:** Erasmus Coordinator, System Admin
 - **Entry Conditions:** university is not added before
 - **Exit Conditions:** operation is completed
 - **The Flow of Events:**
1. Actor enters the university information
 2. Clicks on add

RemoveUniversity

- **Use Case:** RemoveUniversity
 - **Participating Actor:** Erasmus Coordinator, System Admin
 - **Entry Conditions:** university is added and not removed yet
 - **Exit Conditions:** operation is completed
 - **The Flow of Events:**
1. Actor finds the university to be removed
 2. Clicks on remove

EditUniversityInformation

- **Use Case:** EditUniversityInformation
 - **Participating Actor:** Erasmus Coordinator, System Admin
 - **Entry Conditions:** university is added and not removed
 - **Exit Conditions:** operation is completed
 - **The Flow of Events:**
1. Actor enters the new information
 2. Clicks on apply changes

DirectMessage

- **Use Case:** DirectMessage
- **Participating Actor:** Erasmus Coordinator, System Admin, Student, Course Coordinator
- **Entry Conditions:** none

- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actors types a message to the desired person
 2. Clicks on send

ViewNotifications

- **Use Case:** ViewNotifications
- **Participating Actor:** Erasmus Coordinator, System Admin, Student, Course Coordinator
- **Entry Conditions:** notifications are enabled
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actor views the notifications

ViewCalendar

- **Use Case:** ViewCalendar
- **Participating Actor:** Erasmus Coordinator, System Admin, Student, Course Coordinator
- **Entry Conditions:** none
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actor views the calendar with deadlines

CancelApplication

- **Use Case:** CancelApplication
- **Participating Actor:** International Student Office
- **Entry Conditions:** application is not canceled
- **Exit Conditions:** application is canceled
- **The Flow of Events:**
 1. Actor views the application
 2. Clicks on cancel

DiscardPlacement

- **Use Case:** DiscardPlacement
- **Participating Actor:** Student
- **Entry Conditions:** placement is not discarded, placement is active
- **Exit Conditions:** placement is discarded
- **The Flow of Events:**
 1. Actor clicks on discard placement

DisplayPreviousCourses

- **Use Case:** DisplayPreviousCourses

- **Participating Actor:** Student
 - **Entry Conditions:** previous courses are uploaded to the system
 - **Exit Conditions:** operation is completed
 - **The Flow of Events:**
1. Actor views the previous courses

CreatePreApprovalForm

- **Use Case:** CreatePreApprovalForm
 - **Participating Actor:** Student
 - **Entry Conditions:** wishlist is not created
 - **Exit Conditions:** wishlist is created
 - **The Flow of Events:**
1. Actor enters the needed information
 2. Clicks on create

AddToPreApprovalForm

- **Use Case:** AddToPreApprovalForm
 - **Participating Actor:** Student
 - **Entry Conditions:** wishlist is created
 - **Exit Conditions:** addition is completed
 - **The Flow of Events:**
1. Actor selects and enters the needed information
 2. Clicks on add

RemoveFromPreApprovalForm

- **Use Case:** RemoveFromPreApprovalForm
 - **Participating Actor:** Student
 - **Entry Conditions:** wishlist is created and not empty
 - **Exit Conditions:** removal is completed
 - **The Flow of Events:**
1. Actor selects the course to be removed
 2. Clicks on remove

SendPreApprovalForm

- **Use Case:** SendPreApprovalForm
 - **Participating Actor:** Student
 - **Entry Conditions:** wishlist is created and complete
 - **Exit Conditions:** wishlist is sent
 - **The Flow of Events:**
1. Actor click on send

RequestNewCourse

- **Use Case:** RequestNewCourse

- **Participating Actor:** Student
- **Entry Conditions:** the course to be requested is not in the accepted courses list before
- **Exit Conditions:** course is requested
- **The Flow of Events:**
 1. Actor enters the information of the new course
 2. Clicks on request

SeeConfirmedCourses

- **Use Case:** RequestNewCourse
- **Participating Actor:** Student
- **Entry Conditions:** confirmed courses are uploaded to the system
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actor views the confirmed courses

SubmitPreApprovalForm

- **Use Case:** SubmitPreApprovalForm
- **Participating Actor:** Student
- **Entry Conditions:** Form is complete
- **Exit Conditions:** submit is successful
- **The Flow of Events:**
 1. Actor clicks on submit

SubmitLearningAgreement

- **Use Case:** SubmitLearningAgreement
- **Participating Actor:** Student
- **Entry Conditions:** Form is complete
- **Exit Conditions:** submit is successful
- **The Flow of Events:**
 1. Actor clicks on submit

ReportProblem

- **Use Case:** ReportProblem
- **Participating Actor:** Student
- **Entry Conditions:** none
- **Exit Conditions:** problem is reported
- **The Flow of Events:**
 1. Actor types an explanation about the problem
 2. Clicks on report

ViewApplicationStatus

- **Use Case:** ViewApplicationStatus
- **Participating Actor:** Student
- **Entry Conditions:** application is not canceled
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actor views their application status

ViewFormerlyAcceptedCourses

- **Use Case:** ViewFormerlyAcceptedCourses
- **Participating Actor:** Student
- **Entry Conditions:** accepted courses are uploaded to the system
- **Exit Conditions:** operation is completed
- **The Flow of Events:**
 1. Actor views the accepted courses

GeneratePreApprovalForm

- **Use Case:** GeneratePreApprovalForm
- **Participating Actor:** Student
- **Entry Conditions:** Form is not generated yet
- **Exit Conditions:** Form is generated
- **The Flow of Events:**
 1. Actor clicks on generate

ReviewUniversity

- **Use Case:** ReviewUniversity
- **Participating Actor:** Student
- **Entry Conditions:** Actor studied in that university and have not reviewed it yet
- **Exit Conditions:** review is completed
- **The Flow of Events:**
 1. Actor rates and comments on the university
 2. Clicks on submit

ApproveNominatedCourses

- **Use Case:** ApproveNominatedCourses
- **Participating Actor:** Faculty Administration Committee Member
- **Entry Conditions:** courses are nominated
- **Exit Conditions:** courses are approved or rejected
- **The Flow of Events:**
 1. Actor views the courses
 2. Clicks on approve

RejectsNominatedCourses

- **Use Case:** RejectsNominatedCourses
- **Participating Actor:** Faculty Administration Committee Member
- **Entry Conditions:** courses are nominated
- **Exit Conditions:** courses are approved or rejected
- **The Flow of Events:**
 1. Actor views the courses
 2. Clicks on approve

ApproveCourseTransferForm

- **Use Case:** ApproveCourseTransferForm
- **Participating Actor:** Faculty Administration Committee Member
- **Entry Conditions:** form is uploaded
- **Exit Conditions:** form is approved or rejected
- **The Flow of Events:**
 1. Actor views the form and clicks on approve

RejectCourseTransferForm

- **Use Case:** RejectCourseTransferForm
- **Participating Actor:** Faculty Administration Committee Member
- **Entry Conditions:** form is uploaded
- **Exit Conditions:** form is approved or rejected
- **The Flow of Events:**
 1. Actor views the form and clicks on reject

2.3.2. Object Models and Class Diagram

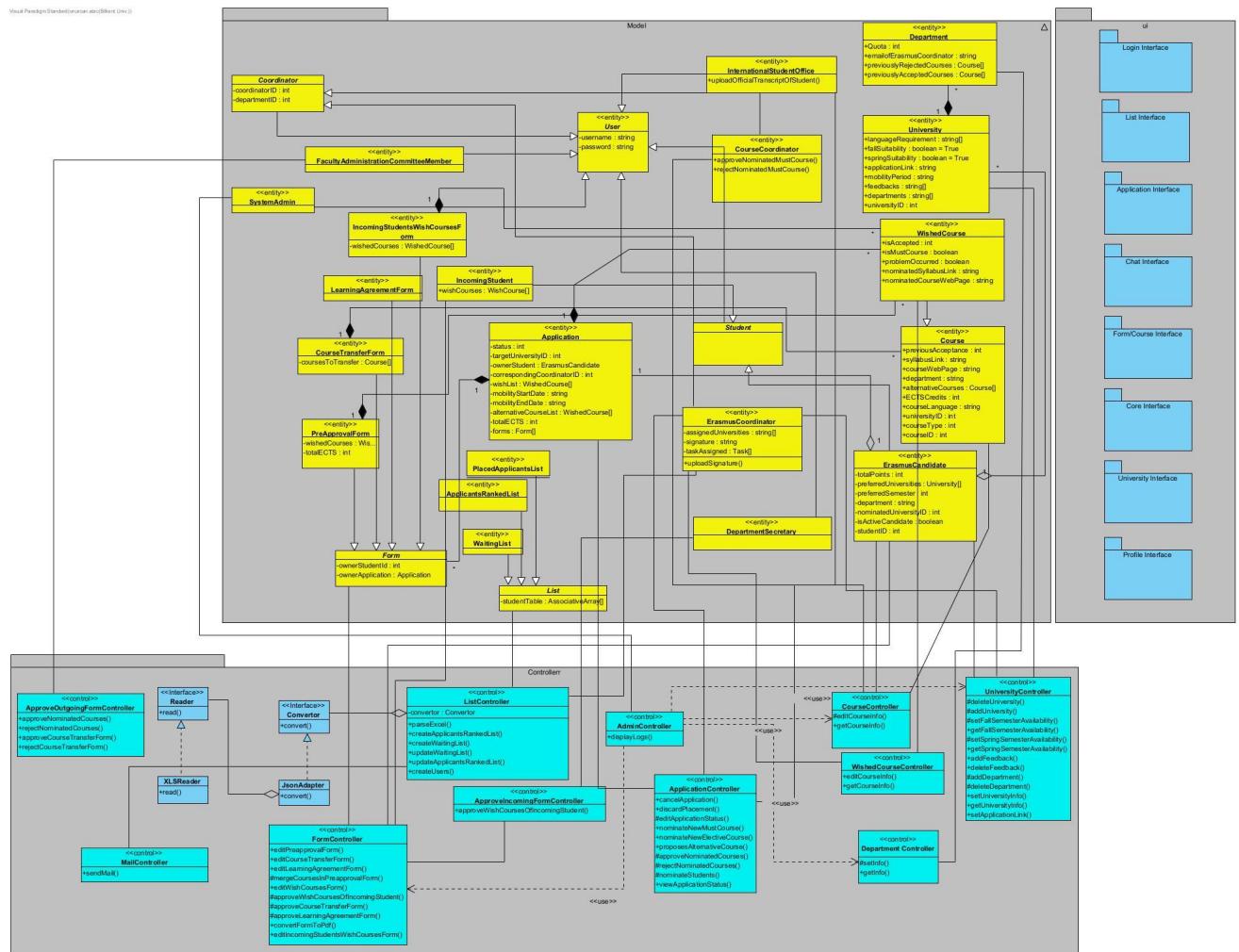


Fig. 2: Class Diagram

The full diagram can be accessed through:

<https://github.com/trelans/319-Project/blob/main/Diagrams/ClassDiagram.jpg>

The class diagram is constructed in a Model Controller View manner.

Model, Controller, and View classes for a specific class are connected with each other.

As mentioned before, the system has different kinds of users. Therefore there exists an abstract user class that realizes all student classes, coordinator classes, international student office, faculty administration committee member, and a department secretary. User classes are connected with the corresponding controller classes specific to that user class which holds the operations which that user is allowed to and can perform.

Student classes are connected with their controller classes where their allowed operations are

present. Every student registered in the system is an instance of this class. Their properties are stored and used to fill in information during the procedure. Outgoing students and incoming students are the two instances of the student class. Their properties and controller class connections are made according to the operations these students must perform during the whole Erasmus procedure.

Erasmus candidate model class extends the Student class. This class represents the outgoing students in the system.

Another model class that extends the Student class is the Incoming Students. This model class represents the international students who applied to Erasmus at Bilkent.

Student classes have lower access to the system compared to Coordinator classes. For example, while coordinators can view other students or approve forms, students can not perform these operations.

Course class is the class representing courses. It holds the properties and information of a class which is then used in viewing courses and other relevant pages, which includes courses. To manipulate course classes/instances, there is a course controller class that includes the needed methods like editCourseInfo to operate and maintain the system healthily and sustainably.

Wished Course class is the child of the course class, which represents a course instance that is requested to be taken by an Erasmus Candidate. The coordinators determine the acceptance of the wished course by the course controller class. Since this class is the child of the Course class, it also holds all the needed information and properties of the class.

The coordinator class is an abstract class which then realizes as Erasmus and Course coordinators. Erasmus Coordinator is the user type that mainly runs the Erasmus procedure, approves forms, and other similar operations. It is connected to various controller classes since they should be able to edit/manipulate nearly everything in the system. Their overall access level to the system is naturally higher than the students and course coordinators.

Course Coordinator is the user type that manipulates applications. Therefore it is connected to the application controller class. Even though they have higher access than a student, they do not have the same access as an Erasmus coordinator instance.

International Student Office is a model class that deals with applications overall. They provide the transcripts and other official documents. Their controller class allows them to manipulate the status of the students and applicants in the system to a certain level.

Department Secretary is a model class that deals with incoming international students. They check and look for any inconveniences in the forms and wished courses of the incoming students. Their

controller class allows them to manipulate the status of applications.

Faculty Administration Committee Member is a model class that deals with the course forms of outgoing Erasmus candidates. They check the course-related forms and approve/reject them accordingly. Their controller class allows them to control and manipulate form class instances.

The system Admin class is a model class that has access to everything. They are the admins who can control and manipulate everything. They are needed to interfere with any problem or difficulty and help any type of user with all possible problems. The controller classes to which they are connected allow them to manipulate/edit everything in the system.

University class is the class for the university instances. Universities that are Erasmus partners are registered to the system as university class instances. They hold relevant and needed information about the universities. Students can later view information to ease their wonderings. Relevant and useful links are also stored for reaching needed information. To manipulate/edit the properties of a university, this class is connected to a university controller class which holds the methods to operate needed changes.

Form class is an abstract class that realizes all needed forms for the Erasmus procedure. Pre-Approval form and course transfer form are some examples of those. Forms are generated in the system by the students with the appropriate UI. Students also upload their forms to the systems. Coordinators can view and approve/reject the forms with the controller classes. Forms all have relevant controller classes to manipulate them.

Application class is a model class that extends the Form class. It is the model class for an Erasmus application. An application can be viewed and edited by the students. Erasmus coordinators can additionally approve and edit these applications. To edit this model class, an application controller class enables users who have permission to manipulate the application.

The list abstract class is a model class for all lists used by coordinators or the system itself to perform operations related to listing applicants during the procedure. The list class controller includes methods for updating the lists in the system. Waiting List, ApplicantsRankedList, and PlacedApplicantsList classes are the model classes that extend from the List class. List instances are mainly used to rank/order the applicants before the mobility period.

2.3.3. Dynamic Models

2.3.3.1 Activity Diagrams

2.3.3.1.1 Create User Activity

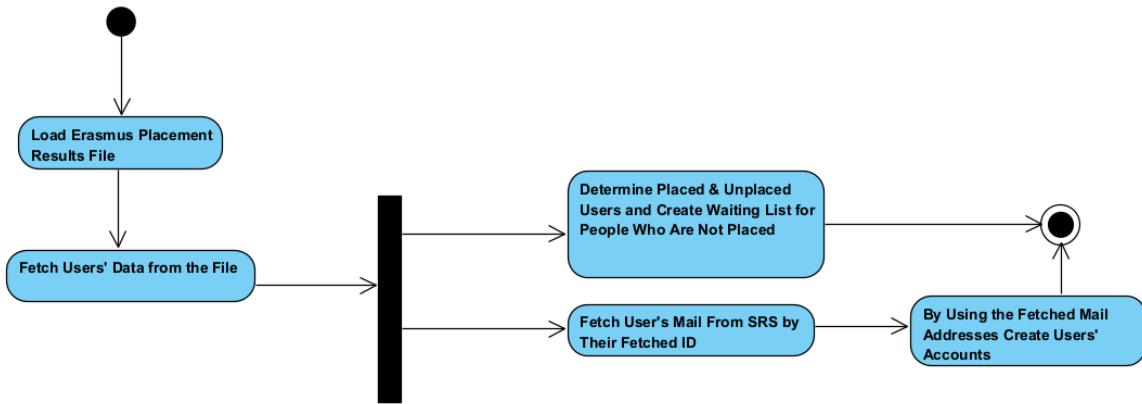


Fig. 3: Create User Activity

When the Erasmus coordinator uploads the Erasmus replacement results, the data, like the user's id number, is fetched from the file. Then, by using the fetched IDs of the users' mail addresses are fetched from SRS. In the next step, all the fetched data is gathered and used to create user accounts with default passwords. Then, the emails to change the default passwords are sent to the users' mail addresses. Also, in parallel to these activities, a waiting list was created from the parsed data from the replacement results file.

2.3.3.1.2 Submit Pre-approval Form Activity

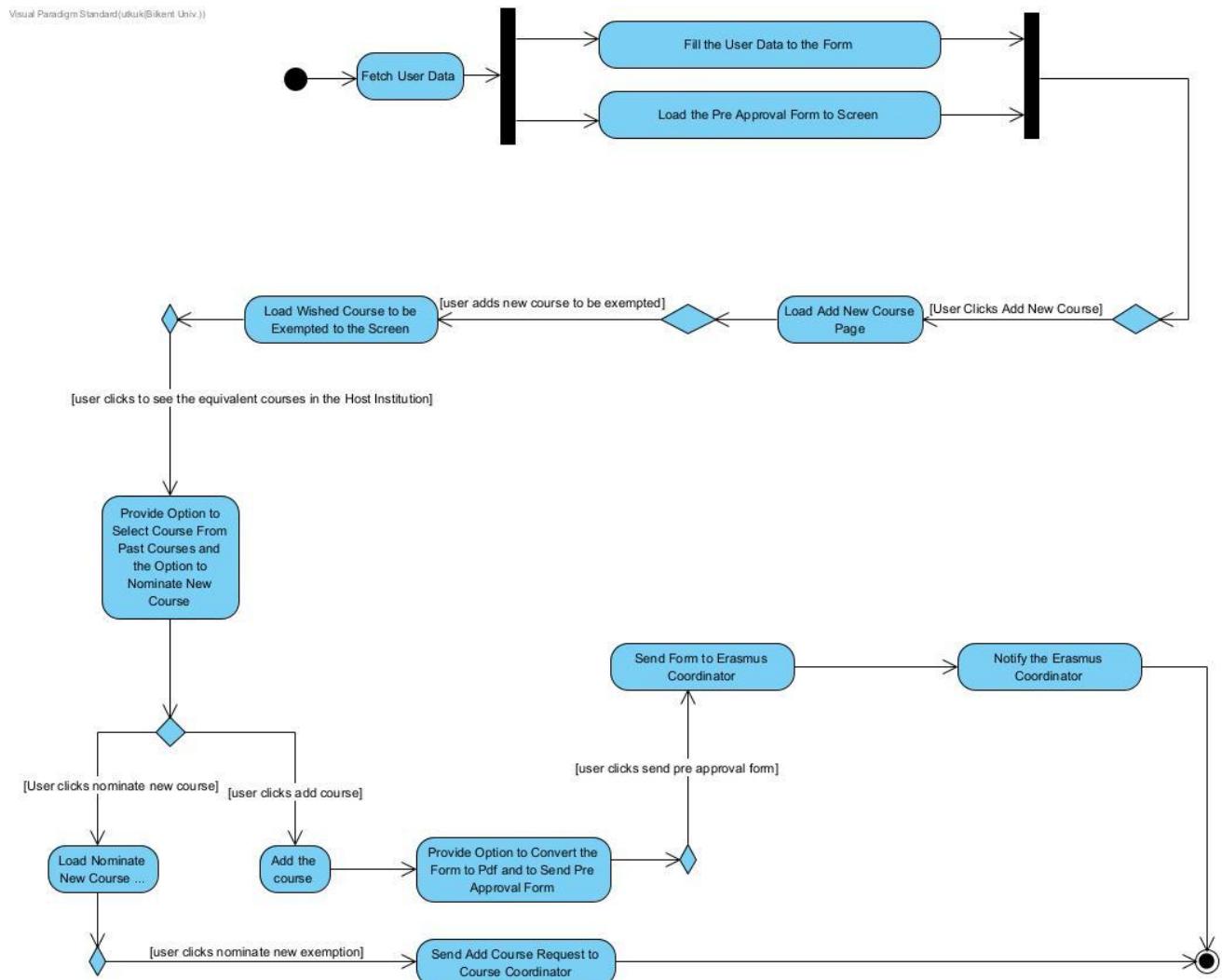


Fig. 4: Pre-approval form activity

In this activity, students' critical information such as name, surname, and id will be fetched from the database. The fetched information will be filled in the pre-approval form and cannot be changed. After that, the student will be able to select the desired courses and submit them. After filling the form, the system will provide an option to convert the form as a pdf to students. If the user selects to send the form, the application will notify and send the document to the Erasmus Coordinator.

2.3.3.2 Sequence Diagram

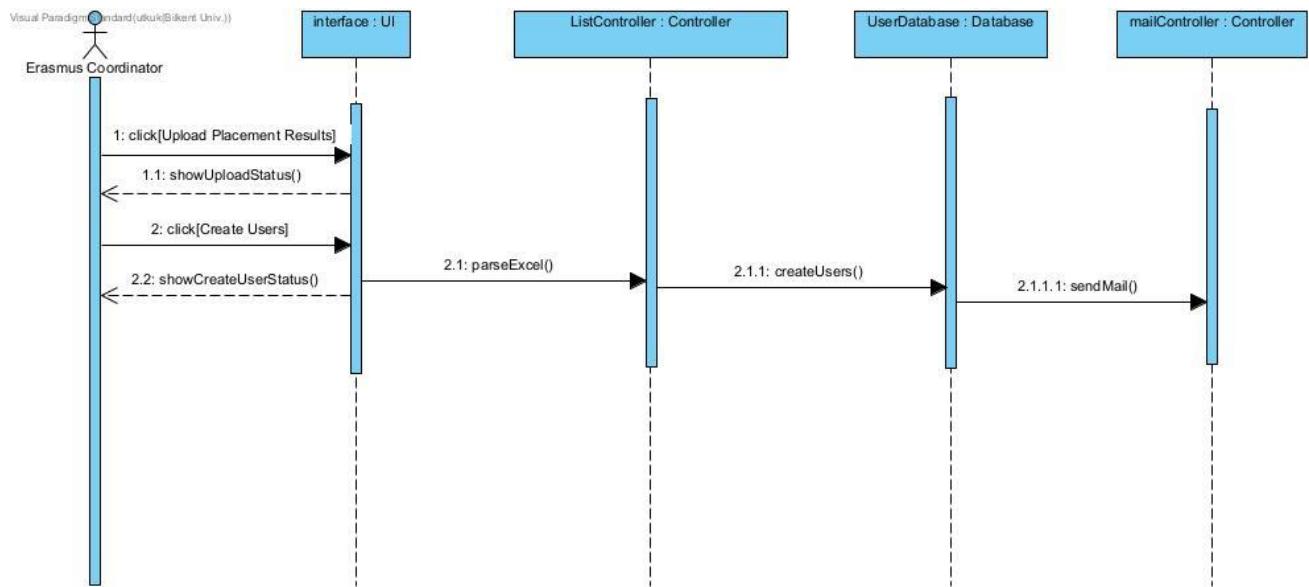


Fig. 5: Create User Sequence Diagram

The Erasmus coordinator uploads the placement results to the system by clicking the upload placement results. If the coordinator clicks the create users option, the placement result file is parsed, and user accounts are created.

2.3.3.3 University State Diagram

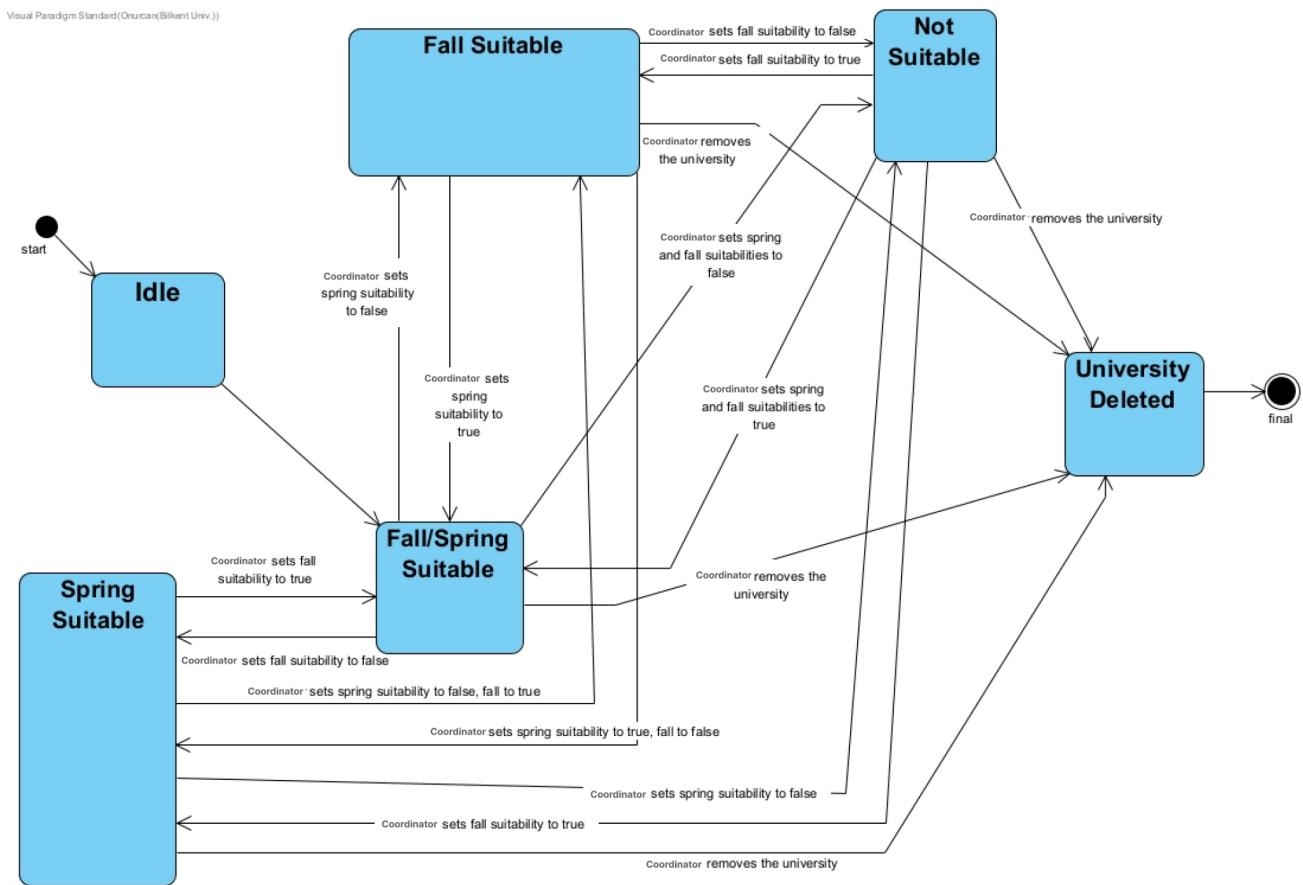


Fig. 6: University State Diagram

This diagram illustrates the state of a University object. Firstly, when a University is added to the system by the Erasmus Coordinator, it is set to be available for the students to go on both spring and fall semesters. The coordinator can then change the suitability of the universities to fall suitable, spring suitable, and not suitable using the interface. The coordinator can also delete a University, and once deleted; there is no possibility that that University could be available for any semester. Hence, the diagram concludes there.

2.3.3.4 WishedCourse and Course State Diagram

Visual Paradigm Standard (Onurcan/Bilkent Univ.)

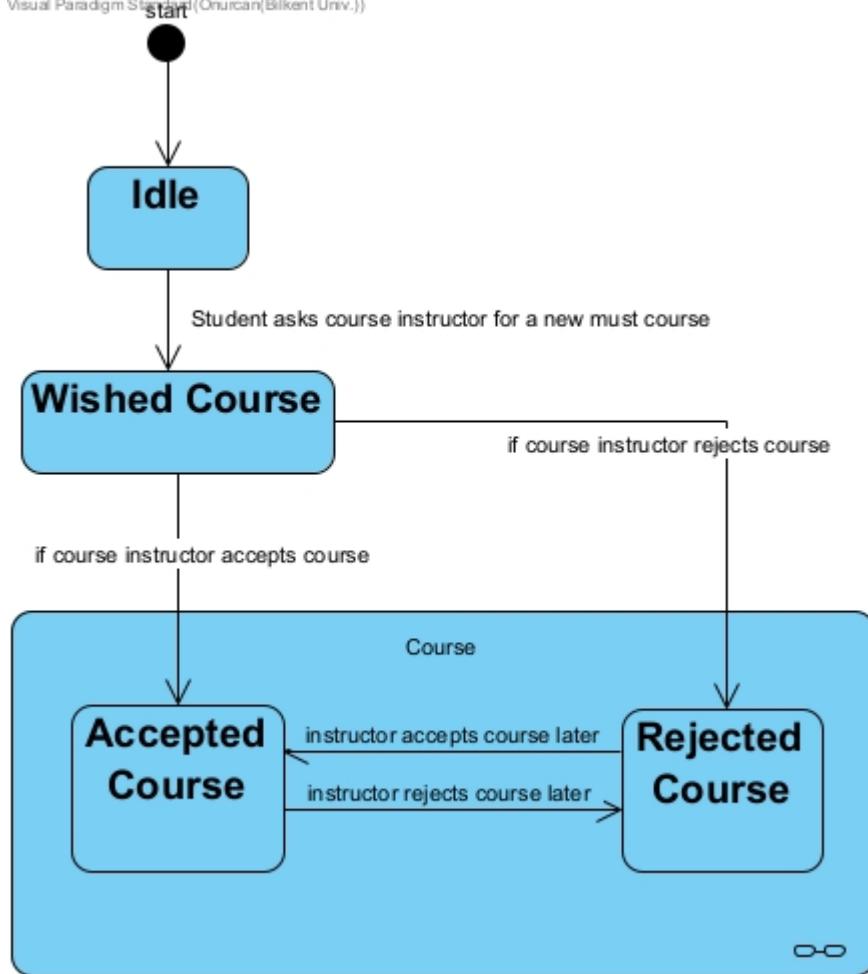


Fig. 7: WishedCourse and Course State Diagram

This diagram illustrates the states and relations of WishedCourse and Course objects and aims to clarify the course qualification system. Once a student asks a course instructor to take a course from his/her respective university that any other student has not taken before as a must course, a Wished Course is created. Then, if the instructor accepts the course as equivalent to the must course that he/she teaches at Bilkent University, that course is saved as an Accepted Course. If the instructor rejects the course, it is saved as a Rejected Course. The instructor can also accept a Rejected Course or reject an Accepted Course later.

2.3.3.5 Application State Diagram

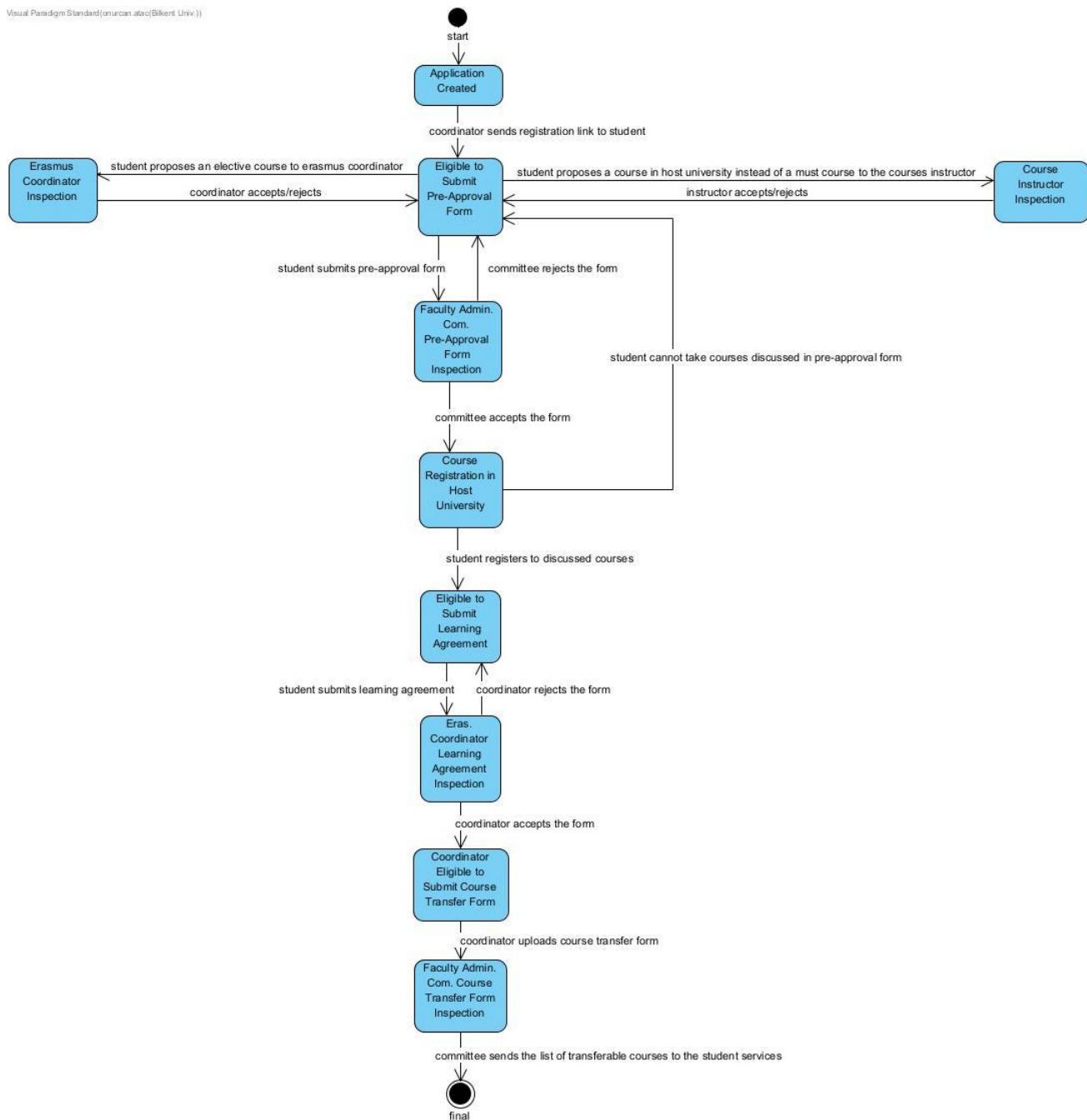


Fig. 8: Application State Diagram

This diagram illustrates the different stages of status in the Application object. Firstly, with the first Erasmus placements being announced, an Application is created for each student on the list, either placed to a university or on the waiting list. After the coordinator sends the registration links to the placed students, they become eligible to submit their Pre-Approval Form, which is going to be approved or disapproved by the committee. Students can also suggest must courses to the respective instructors at that stage. If the committee accepts the form, students are eligible to submit the Learning Agreement Form, which is going to be approved by the coordinator. At that stage, students can contact the coordinator about the courses that are not taken due to conflicts or other issues, while also suggesting new courses. In order to make sure that the student takes the contracted courses beforehand, students are required to upload their transcripts once the term starts. Only then will their state be fully approved. In the end, the Course Transfer Form, which the coordinator prepares, will be inspected by the committee.

2.3.4. User Interface

2.3.4.1 User Authentication UIs

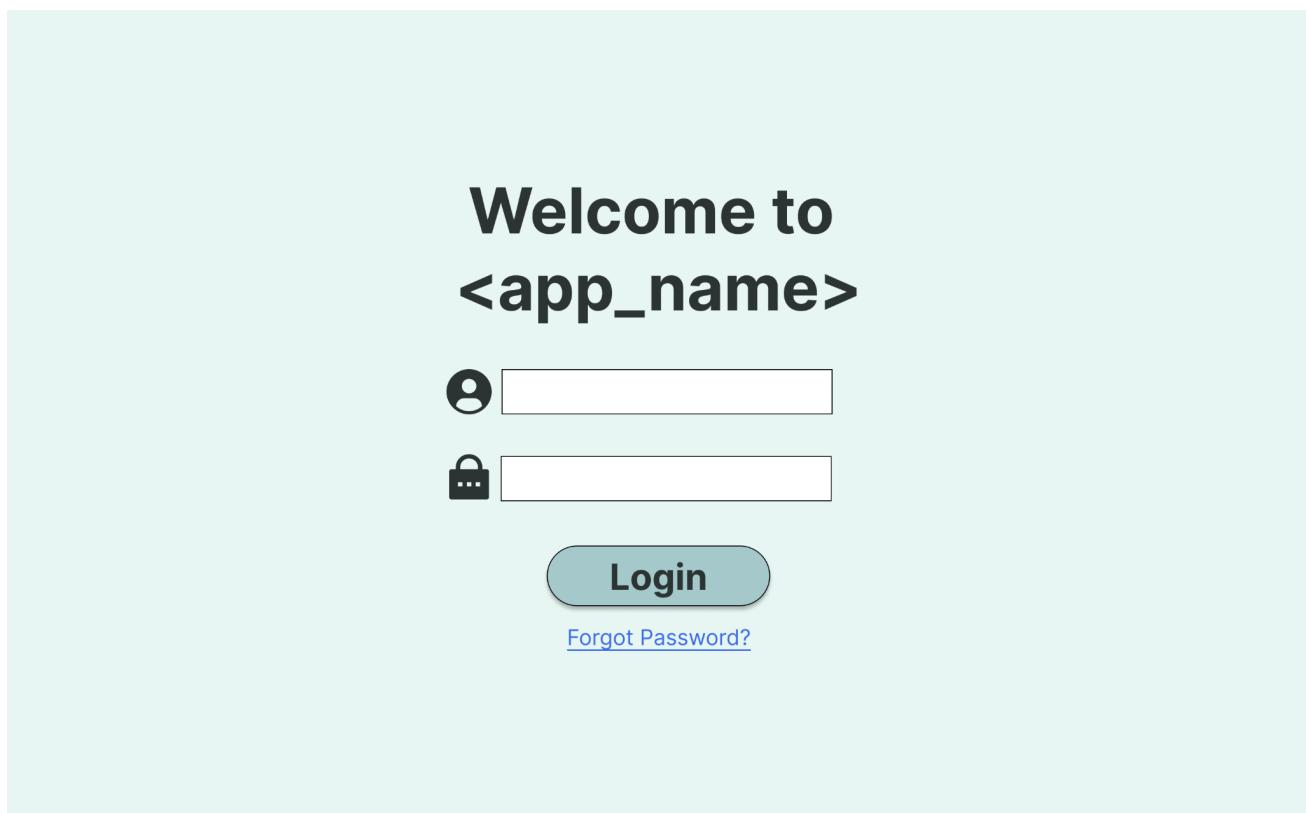


Fig. 9: Login Page UI

Forgot Your Password?

That's okay, it happens.

Click on the button to get a password reset link.
The link will be sent to your e-mail.

Reset Password

Fig. 10: Forgot Password Page UI

Verification Code Sent to <e-mail>

Please enter the code below

•	•	•	•
---	---	---	---

Submit Code

Fig. 11: Reset Password Confirm Page UI

**Please enter your
new password**

Reset Password

Fig. 12: Reset Password Page UI

**Please enter your
new E-Mail**



Reset E-Mail

Fig. 13: Reset E-Mail Page UI

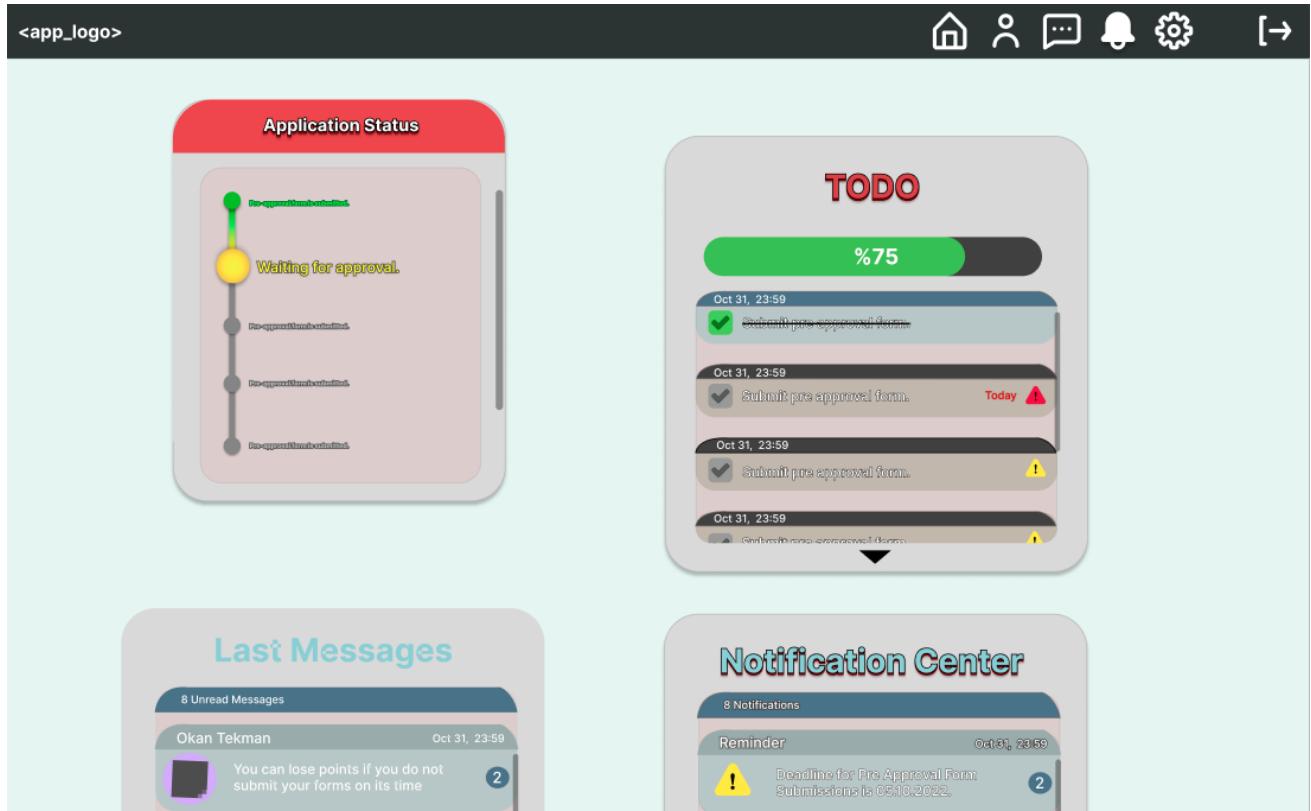


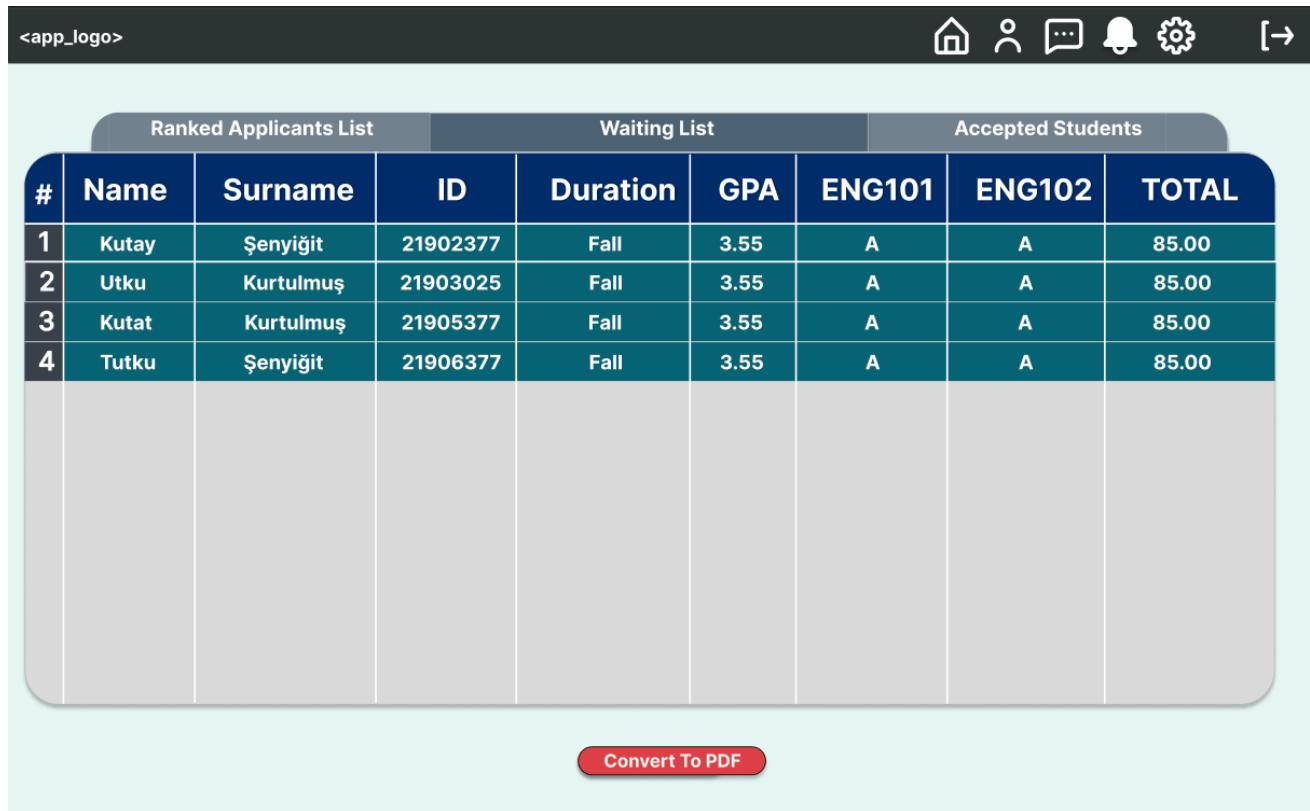
Fig. 14: Dashboard UI (Only Erasmus Candidate see Application Status popup, others reach through TODO list)

Application Status: Students can see the stage of their application.

Todo: User can see what needs to be done and their due dates, it also shows the operations that are completed recently.

Last Messages: Users can see their recent messages.

Notification Center: User can see their recent notifications.



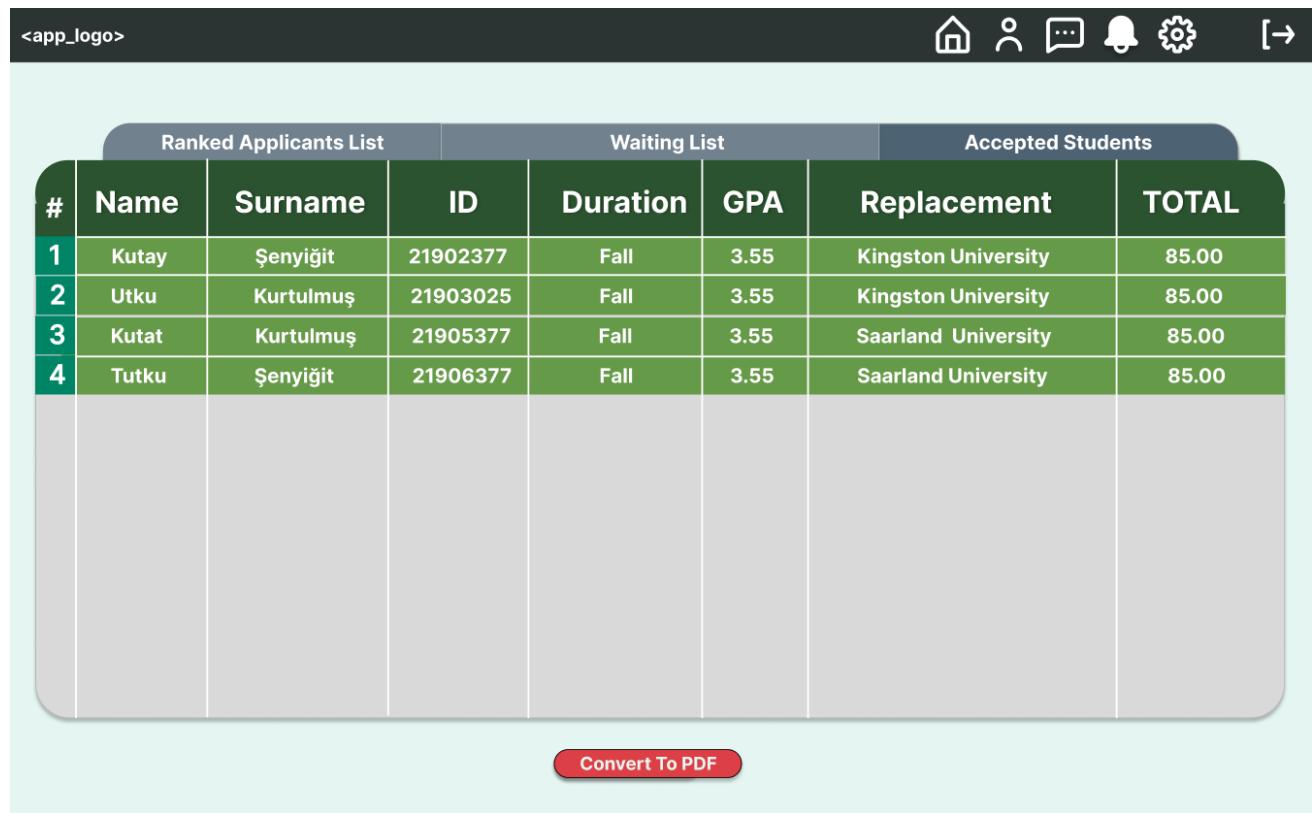
The screenshot shows a user interface for managing student applications. At the top, there is a dark header bar with icons for home, user profile, settings, and navigation. Below this is a light blue header with three tabs: "Ranked Applicants List", "Waiting List" (which is currently selected), and "Accepted Students". The main content area contains a table with the following data:

#	Name	Surname	ID	Duration	GPA	ENG101	ENG102	TOTAL
1	Kutay	Şenyiğit	21902377	Fall	3.55	A	A	85.00
2	Utku	Kurtulmuş	21903025	Fall	3.55	A	A	85.00
3	Kutat	Kurtulmuş	21905377	Fall	3.55	A	A	85.00
4	Tutku	Şenyiğit	21906377	Fall	3.55	A	A	85.00

Below the table is a red button labeled "Convert To PDF".

Fig. 15: Waiting List UI

Shows the list of the students that are not placed and their information.



The screenshot shows a mobile application interface for managing student placements. At the top, there is a dark header bar with icons for home, user profile, message, notifications, settings, and a menu. Below the header is a light blue navigation bar with three tabs: "Ranked Applicants List" (selected), "Waiting List", and "Accepted Students". The main content area is a table titled "Accepted Students" with the following columns: #, Name, Surname, ID, Duration, GPA, Replacement, and TOTAL.

#	Name	Surname	ID	Duration	GPA	Replacement	TOTAL
1	Kutay	Şenyiğit	21902377	Fall	3.55	Kingston University	85.00
2	Utku	Kurtulmuş	21903025	Fall	3.55	Kingston University	85.00
3	Kutat	Kurtulmuş	21905377	Fall	3.55	Saarland University	85.00
4	Tutku	Şenyiğit	21906377	Fall	3.55	Saarland University	85.00

[Convert To PDF](#)

Fig. 16: Accepted Students UI

Shows the list of the students that are placed and their information.

The screenshot shows a mobile application interface for managing student applications. At the top, there is a dark header bar with icons for home, user profile, messages, notifications, settings, and a menu. Below the header is a light blue navigation bar with tabs for "Ranked Applicants List", "Waiting List", and "Accepted Students".

The main content area displays a table titled "Ranked Applicants List" with the following columns: #, Name, Surname, ID, Duration, GPA, Replacement, and TOTAL.

The table contains the following data:

#	Name	Surname	ID	Duration	GPA	Replacement	TOTAL
1	Kutay	Şenyiğit	21902377	Fall	3.55	Kingston University	85.00
2	Utku	Kurtulmuş	21903025	Fall	3.55	Kingston University	85.00
3	Kulay	Şenyiğit	21903377	Fall	3.55	Waiting	85.00
4	Kupay	Şenyiğit	21904377	Fall	1.55	-----	85.00

At the bottom right of the table area is a red button labeled "Convert To PDF".

Fig. 17: Ranked Applicants List UI

Shows the list of all of the students and their information.

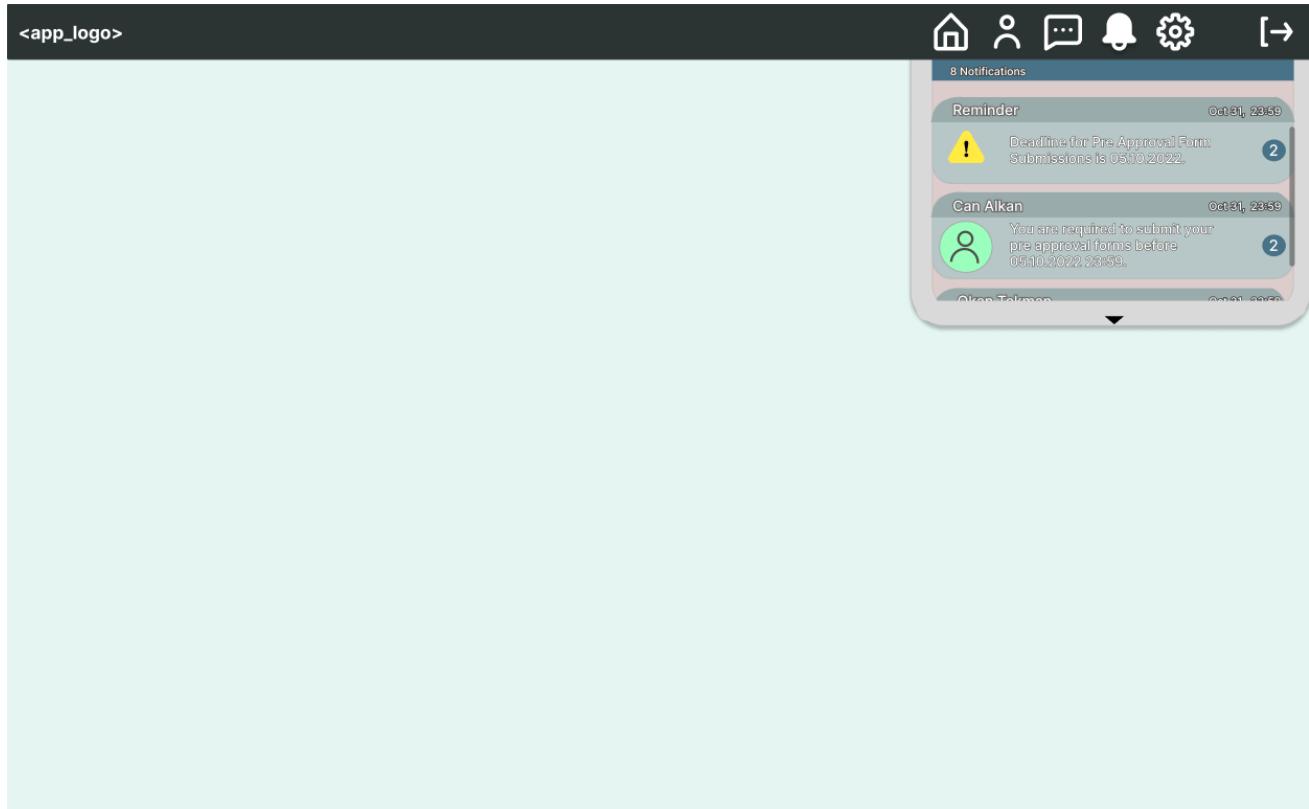


Fig. 18: Notifications UI

Users can see their recent notifications. Such as reminders of due dates, announcements. Students are also notified about the progress about their application.

The screenshot shows a mobile application interface for managing student applications. At the top, there is a toolbar with icons for home, user profile, messages, notifications, settings, and a search bar. Below the toolbar, the student's name, Tutku Şenyiğit, is displayed. The application status is shown as "Waiting for Academic Coordinator Inspection". The applied institution is listed as "<University Name>". The responsible Erasmus coordinator is Can Alkan. The mobility period is set from 01.09.2023 to 01.01.2024. The preapproval form is approved, and the learning agreement form has not been uploaded yet. There are three "View" buttons next to the learning agreement form. The course transfer form is not yet available. At the bottom, there are two red buttons: "Cancel Application" and "Discard Placement".

Student Name:	Tutku Şenyiğit
Application Status:	Waiting for Academic Coordinator Inspection
Applied Institution:	<University Name>
Responsible Erasmus Coordinator:	Can Alkan
Mobility Period:	01.09.2023 - 01.01.2024
Preapproval Form:	Approved (Deadline: Passed)
Learning Agreement Form:	Not Uploaded (Deadline: 31.03.2023)
Course Transfer Form:	Not Yet Available

Fig. 19: Application Page for Outgoing Students

Students can see and manage their application process, they also have the option to either cancel their application completely or discard their placement if they don't like their host university.

Toolbar
<app_logo>

Student Name:	Tutku Şenyiğit		
Application Status:	Waiting for Academic Coordinator Inspection		
Applied Institution:	<University Name>		
Responsible Erasmus Coordinator:	Can Alkan		
Mobility Period:	01.09.2023 - 01.01.2024		
Preapproval Form:	Approved	(Deadline: Passed)	
Learning Agreement Form:	Not Uploaded	(Deadline: 31.03.2023)	
Course Transfer Form:	Not Yet Available		

Fig. 20: Application Page for Erasmus Coordinators, International Student Office

This page is similar to the application page for students, but Erasmus Coordinators will be able to cancel the application of a student or view the form submissions of the . International Student Office can also cancel the application since they check if the GPA of the student satisfies the requirements.

Toolbar
<app_logo>

The screenshot shows a mobile application interface for managing student applications. At the top, there's a dark header bar with icons for home, user profile, messages, notifications, settings, and a search function. Below this is a light blue header section containing the title 'Erasmus+ Application' and a subtitle 'Erasmus+ Application'. The main content area has a light green background and displays the following information:

Student Name:	Tutku Şenyiğit		
Application Status:	Waiting for Academic Coordinator Inspection		
Applied Institution:	<University Name>		
Responsible Erasmus Coordinator:	Can Alkan		
Mobility Period:	01.09.2023 - 01.01.2024		
Preapproval Form:	Approved	(Deadline: Passed)	View
Learning Agreement Form:	Not Uploaded	(Deadline: 31.03.2023)	View
Course Transfer Form:	Not Yet Available		View

Fig. 21: Application Page for Faculty Administration Committee Members

This page is similar to the application page for Erasmus Coordinators, however FAC Members will only be able to view the submitted forms of the student.

Toolbar: <app_logo>

<u>University Name</u>	<u>Semester</u>	<u>Quota</u>	<u>Mobility Period</u>	<u>Country</u>		
1. Kingston University	Fall	25	22.01.2022 - 1.02.2022	England	Edit	X
2. MIT	Spring	25	22.01.2022 - 1.02.2022	England	Edit	X
3. MZU	Both	12	22.01.2022 - 1.02.2022	England	Edit	X
4. Cambridge University	Fall	6	22.01.2022 - 1.02.2022	England	Edit	X
5. Cambridge University	Fall	6	22.01.2022 - 1.02.2022	England	Edit	X
6. Cambridge University	Fall	15	22.01.2022 - 1.02.2022	England	Edit	X
7. Cambridge University	Fall	45	22.01.2022 - 1.02.2022	England	Edit	X
8. Cambridge University	Fall	74	22.01.2022 - 1.02.2022	England	Edit	X
9. Cambridge University	Spring	1	22.01.2022 - 1.02.2022	England	Edit	X
10. Cambridge University	Spring	4	22.01.2022 - 1.02.2022	England	Edit	X

Add University

Fig. 22: University Page for Erasmus Coordinator

Erasmus Coordinator can see, modify and add the universities available for Erasmus.

<app_logo>
[→]

Applicant Info:

Name:	<Applicant Name>
Surname:	<Applicant Surname>
ID Number:	<Applicant ID>
Department:	<Applicant Department>
Host Institution Name:	<Host Institution Name>
Academic Semester:	<Applied Academic Year> <Applied Academic Semester>

Host Institution Courses to be Transferred Upon Approval Course or Requirement to be exempted

Course Code	Course Name	ECTS Credits	isElective	Course Code	Course Name	Course Group	Part No
1. X_400614	Data Structures and Algorithms	6.0	No	CS 473	Algorithms I	None	N X
2. L_GCBAALG003	Imagining The Dutch: themes in Dutch History	6.0	Yes	None	None	Arts Core Elective	N X
3. X_401020	Statistical Methods	6.0	Yes	Math 260	None	Technical Elective	N X
4. INF4032	Computer Networks	3.0	Yes	CS 421	None	Technical Elective	1 X
5. INF4036	Applied Cryptography	1.0	Yes	CS 421	None	Technical Elective	2 X

[Add New Course](#)

Total ECTS: 22

I have read [info](#)
Approved By
Name
Signature
Date

[Convert To PDF](#)
[Send Preapproval Form](#)
Not Approved

Fig. 23: Pre-Approval Form Page Student UI

Outgoing students can enter required information in order to complete Pre-Approval Form and submit it for inspection. There is an option to convert it to PDF as well. Add New Course button leads to Fig. 27.

<app_logo>

Applicant Info:

Name:	<Applicant Name>
Surname:	<Applicant Surname>
ID Number:	<Applicant ID>
Department:	<Applicant Department>
Host Institution Name:	<Host Institution Name>
Academic Semester:	<Applied Academic Year> <Applied Academic Semester>

APPROVE PRE-APPROVAL FORM

Host Institution Courses to be Transferred Upon Approval Course or Requirement to be exempted*

Course Code	Course Name	Credits	isElective	Course Code	Course Name	Course Group	Part No
1. X_400614	Data Structures and Algorithms	6.0	No	CS 473	Algorithms I	None	N
2. L_GCBAALG003	Imagining The Dutch: themes in Dutch History	6.0	Yes	None	None	Arts Core Elective	N
3. X_401020	Statistical Methods	6.0	Yes	Math 260	None	Technical Elective	N
4. INF4032	Computer Networks	3.0	Yes	CS 421	None	Technical Elective	1
5. INF4036	Applied Cryptography	1.0	Yes	CS 421	None	Technical Elective	2

Total ECTS: 22

Approved By	<u>Name</u>	<u>Signature</u>	<u>Date</u>
Not Approved			

Convert To PDF **Edit Form** **Give Feedback**

Fig. 24: Pre-Approval Form Page Erasmus Coordinator UI

<app_logo>

Applicant Info:

Name:	<Applicant Name>
Surname:	<Applicant Surname>
ID Number:	<Applicant ID>
Department:	<Applicant Department>
Host Institution Name:	<Host Institution Name>
Academic Semester:	<Applied Academic Year> <Applied Academic Semester>

APPROVE PRE-APPROVAL FORM

REJECT PRE-APPROVAL FORM

Host Institution Courses to be Transferred Upon Approval Course or Requirement to be exempted*

Course Code	Course Name	Credits	isElective	Course Code	Course Name	Course Group	Part No
1. X_400614	Data Structures and Algorithms	6.0	No	CS 473	Algorithms I	None	N
2. L_GCBAALG003	Imagining The Dutch: themes in Dutch History	6.0	Yes	None	None	Arts Core Elective	N
3. X_401020	Statistical Methods	6.0	Yes	Math 260	None	Technical Elective	N
4. INF4032	Computer Networks	3.0	Yes	CS 421	None	Technical Elective	1
5. INF4036	Applied Cryptography	1.0	Yes	CS 421	None	Technical Elective	2

Total ECTS: 22

Approved By	<u>Name</u>	<u>Signature</u>	<u>Date</u>
Not Approved			

Convert To PDF **Give Feedback**

Fig. 25: Pre-Approval Form Page Faculty Administration Committee UI

Pre-Approval Form Information Page:

I ACKNOWLEDGE THAT

Failure to prepare and submit pre-approval form before the mobility period starts may result in courses not to be transferred.

The forms will be discussed in the Faculty Administration Committee and courses will be pre-approved or rejected.

In case the pre-approved courses could not be taken due to schedule conflict or prerequisites, the student immediately emails the coordinators and proposes alternative courses. Pre-approval form is amended as necessary. The Faculty Administration Committee do not discuss the amendment, therefore the transfer is not "guaranteed" unless the changed course was previously accepted in an earlier semester.

Pre-Approval and Learning Agreement Forms SHOULD BE COMPLETED DURING THE SEMESTER BEFORE THE MOBILITY STARTS

A transferred course may provide exemption from a requirement in the curriculum if deemed to be equivalent by the Faculty/School Executive Board. It is possible for one transferred course to provide exemption from one or more curriculum courses or vice versa.

Course Group in Pre-Approval Form Page is applicable only if there is a directly equivalent course in the elective group that the student is exempted from. The student will be considered to have taken this course by the STARS system.

Total ECTS credits of courses transferred upon approval SHOULD BE AT LEAST 30 to be ELIGIBLE TO THE ERASMUS AID.

I Accept The Terms

Fig. 26: Info Page UI (Visible to All)

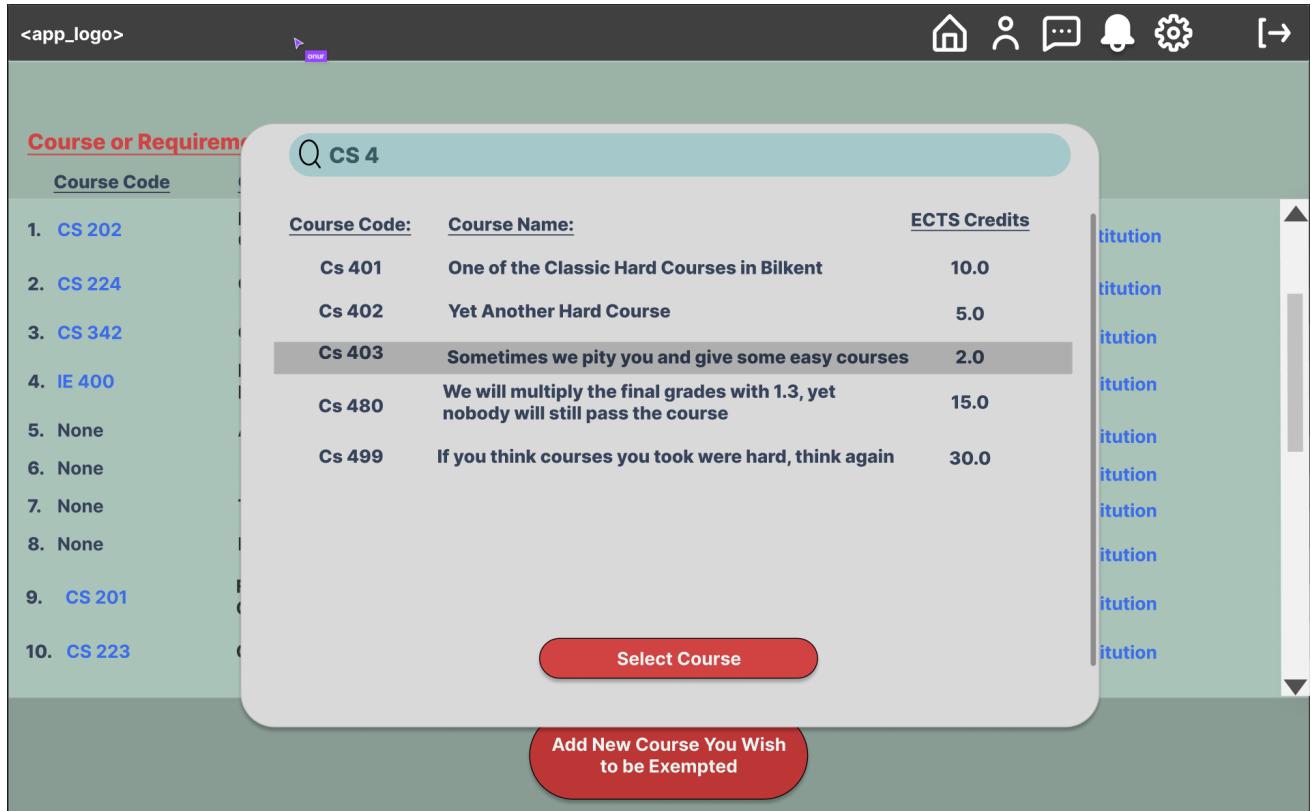


Fig. 27: Add Course Page Adding New Course to be Exempted popup UI (Visible to Erasmus Candidate)

Students need to select a Bilkent University course to replace it with another course from the host university later on in Fig. 29. Moreover, a new page Fig. 28 opens up in the background.

Course or Requirement to be exempted

Course Code	Course Name	ECTS Credits	
1. CS 202	Fundamental Structures of Computer Science I	5.0	Click to see the equivalent courses in the Host Institution
2. CS 224	Computer Organization	6.5	Click to see the equivalent courses in the Host Institution
3. CS 342	Operating Systems	6.5	Click to see the equivalent courses in the Host Institution
4. IE 400	Principles of Engineering Management	5.0	Click to see the equivalent courses in the Host Institution
5. None	Arts Core Elective	-	Click to see the equivalent courses in the Host Institution
6. None	Technical Elective	-	Click to see the equivalent courses in the Host Institution
7. None	Technical Elective	-	Click to see the equivalent courses in the Host Institution
8. None	Project Elective	-	Click to see the equivalent courses in the Host Institution
9. CS 201	Fundamental Structures of Computer Science II	5.0	Click to see the equivalent courses in the Host Institution
10. CS 223	Computer Organization 1	6.5	Click to see the equivalent courses in the Host Institution

Add New Course You Wish to be Exempted

Fig. 28: Add Course Page UI (Visible to Erasmus Candidate)

Students can see the list of courses from Bilkent University that they have added. Those are needed for finding or creating add request equivalent courses from the host university. Clicking to the links on the right side of the courses leads to a popup (Fig. 29). Add New Course button on the bottom leads to Fig. 27, enabling the user to add a new Bilkent course. Clicking to course codes lead to Course Profile Page (Fig. 52)for that course.

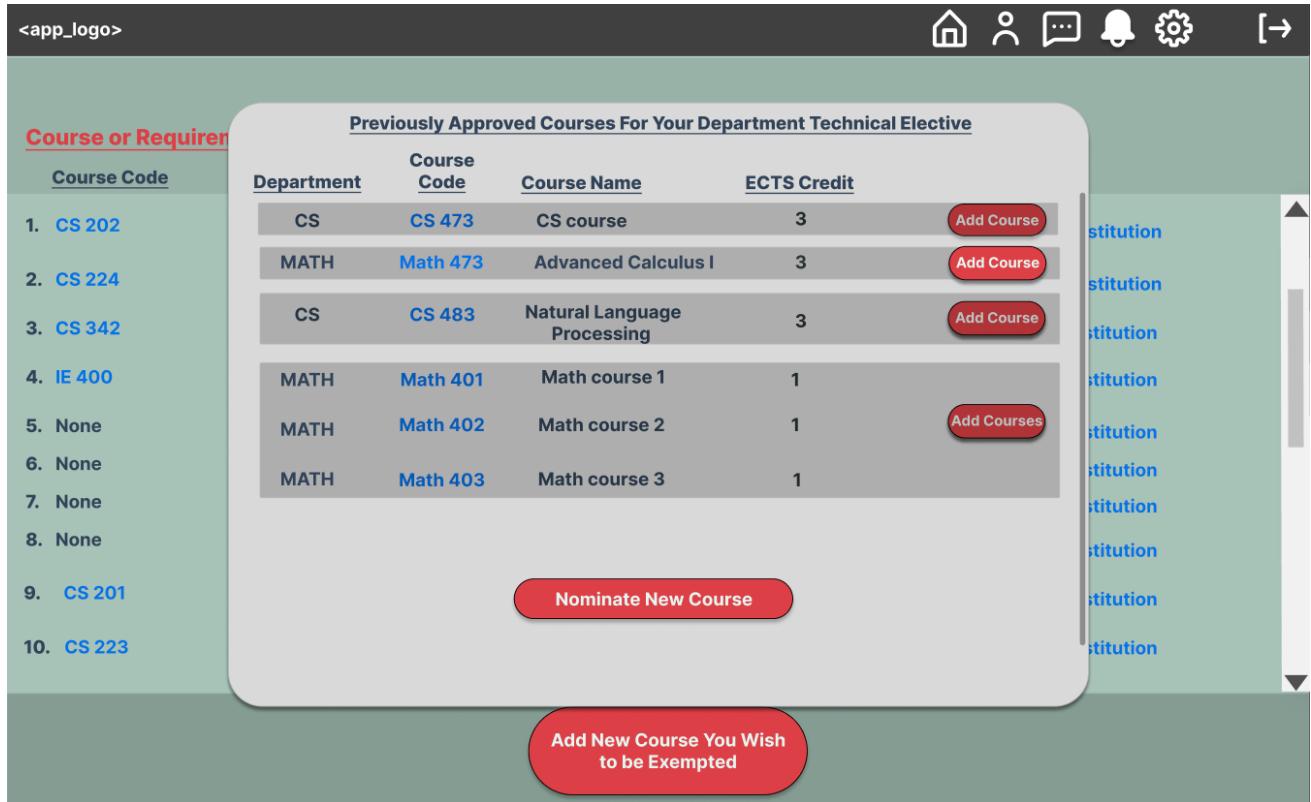


Fig. 29: Add Course Page Courses Giving In the Host popup (for Elective Course) UI

Students need to decide whether they want to nominate a new course or select one or multiple courses from previously accepted courses, while replacing the Bilkent course. Previously added courses can be seen as a list. There can be multiple courses for satisfying a course from Bilkent University. If a student wants to nominate a new course, a new popup opens (Fig. 31).

The screenshot shows a user interface for managing course requirements. On the left, a sidebar lists 'Course or Requirement' items numbered 1 through 10. Item 1 is 'CS 202', item 2 is 'CS 224', item 3 is 'CS 342', item 4 is 'IE 400', items 5-8 are 'None', item 9 is 'CS 201', and item 10 is 'CS 223'. A central modal window titled 'Previously Approved Courses For CS 224' displays a table with columns: Department, Course Code, Course Name, and ECTS Credit. One row is shown: Department 'CS', Course Code 'CS 473', Course Name 'CS course', and ECTS Credit '3'. A red 'Add Course' button is at the bottom right of the table. Below the table is a red 'Nominate New Course' button. At the bottom of the modal is a red button with the text 'Add New Course You Wish to be Exempted'.

Previously Approved Courses For CS 224				
	Department	Course Code	Course Name	ECTS Credit
	CS	CS 473	CS course	3

Nominate New Course

Add New Course You Wish to be Exempted

Fig. 30: Add Course Page Courses Giving In the Host popup (for Must Course) UI

Has the same functionality with Fig. 29, that is given for an example technical elective course. This page is given for a must course example. Nominate New Course button leads to Fig. 31.

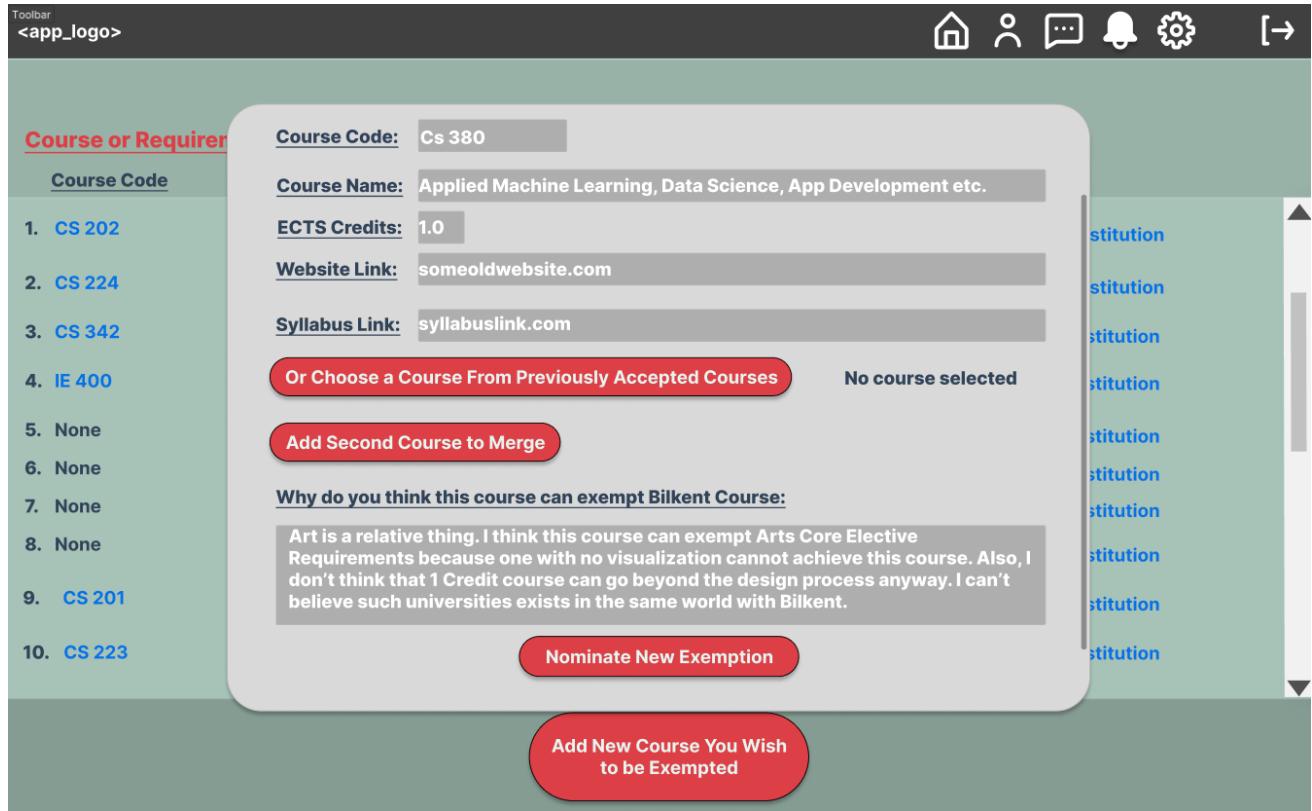


Fig. 31: Add Course Page Nominate Course(Erasmus Candidate) UI

They can either select a course from previously accepted courses , or add one or multiple courses by filling form. If they send the nomination request by clicking Nominate New Exemption, nomination request will be sent to course coordinator or erasmus coordinator.(Fig. 32 and Fig. 33)

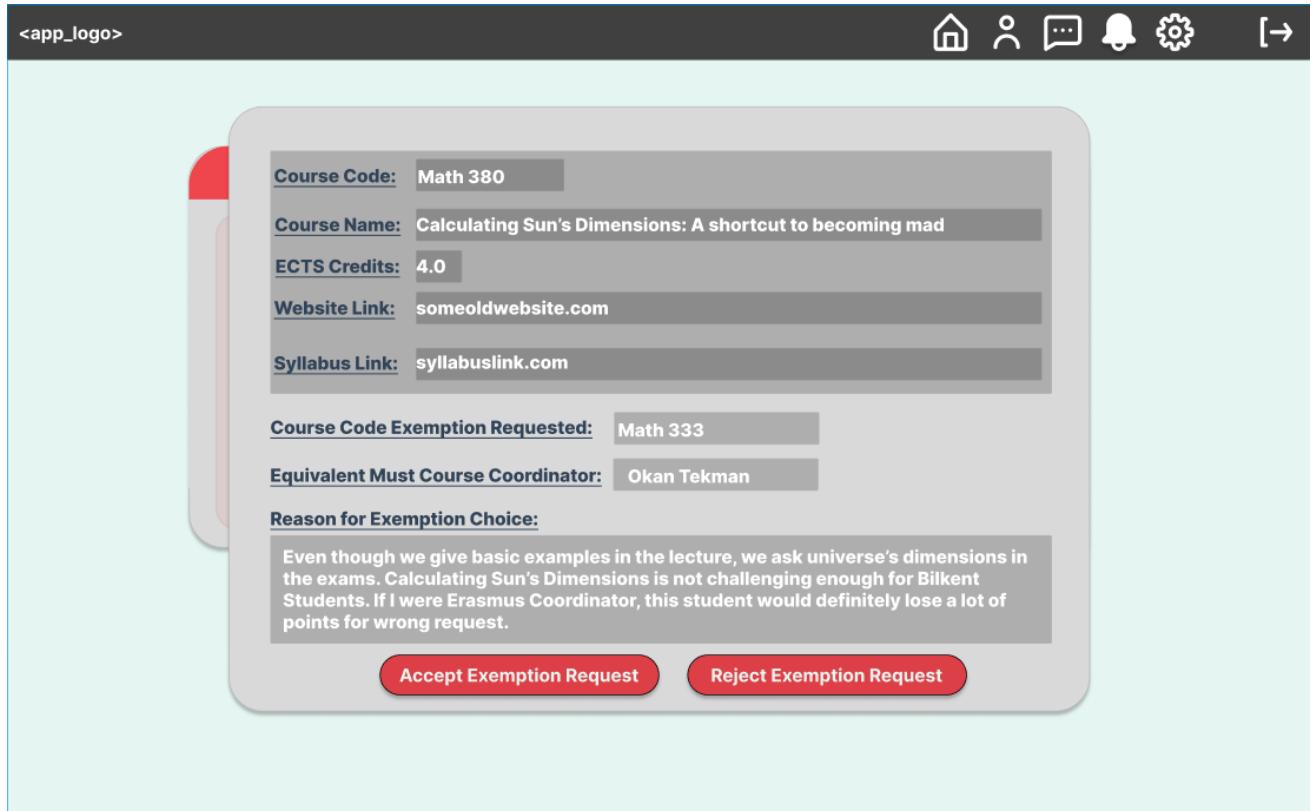


Fig. 32: Add Course Page Accept Exemption Request (Course Coordinator) UI

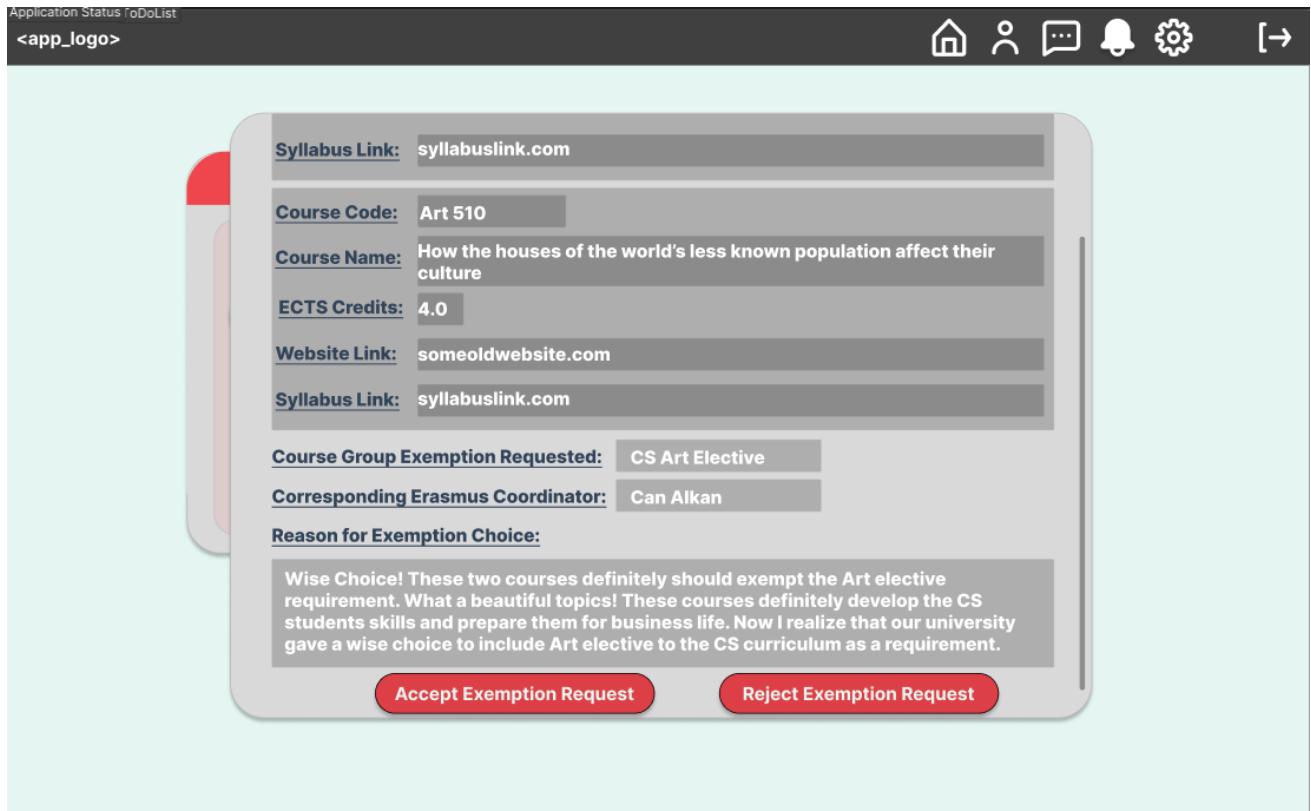


Fig. 33: Add Course Page Accept Exemption Request (Erasmus Coordinator) UI

Learning Agreement Form Before Mobility Part 1 of 3:

Section 2							
	Last Name	First Name's	Date of Birth	Nationality	Sex[M/F]	Academic Year	Study Cycle
Student Info	<Last Name>	<First Name's>	<Date of Birth>	<Nationality>	<Sex[M/F]>	<Academic Year>	<Study Cycle>

Section 3							
	Name	Faculty	Erasmus Code	Department	Address	Country, Country Code	Contact Person Name
Sending Institution	<Instution Name>	<Faculty>	<Erasmus Code>	<Department>	<Address>	<Country, Country Code>	<ContactPerson Name>

Section 4							
	Name	Faculty	Erasmus Code	Department	Address	Country, Country Code	Contact Person Name
Receiving Institution	<Instution Name>	<Faculty>	<Erasmus Code>	<Department>	<Address>	<Country, Country Code>	<ContactPerson Name>

←
→

Convert To PDF
Apply Changes

Fig. 34: Learning Agreement Form Page (1/3) Before Mobility All Views UI

The required information for the learning agreement form is fetched by the application. However, user can edit related informations if there is a mistake in the learning agreement form. After the edit operation, the user can apply the changes and convert the edited form to the PDF. The user also can go to other parts of the form by clicking the arrow buttons.

<app_logo>

Learning Agreement Form Before Mobility Part 2 of 3:

Study Programme at Receiving Institution			
Planned period of the mobility: from [month/year] <month/year> to [month/year] <month/year>			
<u>Component Code</u>	<u>Component Title at the receiving institution (as indicated in the course catalog)</u>	<u>Semester</u> [e.g. autumn / spring / term]	<u>Number of ECTS credits (or equivalent) to be awarded by the receiving institution upon successful completion</u>
Total:			

Section 2 TABLE A

Component Code Component Title at the receiving institution (as indicated in the course catalog) Semester [e.g. autumn / spring / term] Number of ECTS credits (or equivalent) to be awarded by the receiving institution upon successful completion

Total:

Section 3 TABLE B

Component Code Component Title at the receiving institution (as indicated in the course catalog) Semester [e.g. autumn / spring / term] Number of ECTS credits (or equivalent) for the group of educational components in the student's degree that would normally be completed at the sending institution and which will be replaced by the study abroad

Total:

Edit Add Row

← →

Convert To PDF Apply Changes

Fig. 35: Learning Agreement Form Page (2/3) Before Mobility All Views UI

<app_logo>

Learning Agreement Form Before Mobility Part 3 of 3:

By signing this document, the student, the sending institution and the receiving institution confirm that they approve the proposed Learning Agreement and that they will comply with all the arrangements agreed by all parties. Sending and receiving institutions undertake to apply all the principles of the Erasmus Charter for Higher Education relating to mobility for studies (or the principles agreed in the inter-institutional agreement for institutions located in Partner Countries). The sending institution and the student should also commit to what is set out in the Erasmus+ grant agreement. The receiving institution confirms that the educational components listed in Table A are in line with its course catalogue and should be available to the student. The sending institution commits to recognise all the credits gained at the receiving institution for the successfully completed educational components and to count them towards the student's degree as described in Table C. Any exceptions to this rule are documented in an annex of this Learning Agreement and agreed by all parties. The student and receiving institution will communicate to the sending institution any problems or changes regarding the proposed mobility programme, responsible persons and/or study period.

TABLE C						
<u>Commitment</u>	<u>Name</u>	<u>Function</u>	<u>Phone Number</u>	<u>Email</u>	<u>Date</u>	<u>Signature</u>
Student	<Name>	Student	<Phone No>	<email>	<date>	<signature>
Responsible Person at Sending Institution	<Name>	<Function>	<Phone No>	<email>	<date>	<signature>
Responsible Person at Receiving Institution	<Name>	<Function>	<Phone No>	<email>	<date>	<signature>

Edit

← →

Convert To PDF Apply Changes

Fig. 36: Learning Agreement Form Page (3/3) Before Mobility All Views UI

Toolbar <app_logo>

Learning Agreement Form During Mobility Part 1 of 2:

Section 2 Exceptional changes to Table A (to be approved by e-mail or signature by the student, the responsible person in the sending institution and the responsible person in the receiving institution)					
<u>Component Code</u>	<u>Component Title at the receiving institution (as indicated in the course catalog)</u>	<u>Deleted Component [Tick if applicable]</u>	<u>Added Component [Tick if Applicable]</u>	<u>Reason For Change</u>	<u>Number of ECTS credits (or equivalent) to be awarded by the receiving institution upon successful completion</u>

Edit **Add Row**

Section 3 Exceptional changes to Table B (to be approved by e-mail or signature by the student and the responsible person in the sending institution)					
<u>Component Code</u>	<u>Component Title at the receiving institution (as indicated in the course catalog)</u>	<u>Deleted Component [Tick if applicable]</u>	<u>Added Component [Tick if Applicable]</u>	<u>Reason For Change</u>	<u>Number of ECTS credits (or equivalent) for the group of educational components in the student's degree that would normally be completed at the sending institution and which will be replaced by the study abroad</u>

Edit **Add Row**

← →

Convert To PDF **Apply Changes**

Fig. 37: Learning Agreement Form Page (1/2) During Mobility All Views UI

Toolbar <app_logo>

Learning Agreement Form During Mobility Part 2 of 2:

By signing this document, the student, the sending institution and the receiving institution confirm that they approve the proposed Learning Agreement and that they will comply with all the arrangements agreed by all parties. Sending and receiving institutions undertake to apply all the principles of the Erasmus Charter for Higher Education relating to mobility for studies (or the principles agreed in the inter-institutional agreement for institutions located in Partner Countries). The sending institution and the student should also commit to what is set out in the Erasmus+ grant agreement. The receiving institution confirms that the educational components listed in Table A are in line with its course catalogue and should be available to the student. The sending institution commits to recognise all the credits gained at the receiving institution for the successfully completed educational components and to count them towards the student's degree as described in Table C. Any exceptions to this rule are documented in an annex of this Learning Agreement and agreed by all parties. The student and receiving institution will communicate to the sending institution any problems or changes regarding the proposed mobility programme, responsible persons and/or study period.

<u>Commitment</u>	<u>Name</u>	<u>Function</u>	<u>Phone Number</u>	<u>Email</u>	<u>Date</u>	<u>Signature</u>
Student	<Name>	Student	<Phone No>	<email>	<date>	<signature>
Responsible Person at Sending Institution	<Name>	<Function>	<Phone No>	<email>	<date>	<signature>
Responsible Person at Receiving Institution	<Name>	<Function>	<Phone No>	<email>	<date>	<signature>

Edit

← →

Convert To PDF **Apply Changes**

Fig. 38: Learning Agreement Form Page (2/2) During Mobility All Views UI

Toolbar
 <app_logo>

Learning Agreement Form After Mobility:

Section 2				
Academic Outcomes at Receiving Institution				
Component Code	Component Title at the receiving institution (as indicated in the course catalog)	Was the component successfully completed by the student? [Yes/No]	Number of ECTS credits (or equivalent)	Receiving Institution Grade
Total:				

Section 3

Recognition Outcomes at Sending Institution			
Component Code	Title of recognised component (as indicated in the course catalogue) at the sending institution	Number of ECTS credits (or equivalent)	Sending Institution Grade (if applicable)
Total:			

Edit **Add Row**

Edit **Add Row**

Convert To PDF **Apply Changes**

← **→**

Fig. 39: Learning Agreement Form Page After Mobility UI

Toolbar
<app_logo>

The screenshot shows the first page of a course transfer form. At the top, there's a toolbar with icons for home, user profile, message, notifications, settings, and a search bar. Below the toolbar, the title 'Course Transfer and Exemption Form for Undergraduate Students Part 1 of 2' is displayed. The form is divided into several sections:

- Section 2:** Contains fields for Name (Name: and Surname:), ID Number (Department:), Academic Year (Semester:), and a large 'Edit' button.
- Section 3:** Contains fields for 'External Transfer Student' and 'Outgoing Exchange Student' (checkboxes), 'Name of the institution from which courses are transferred' (text input), and checkboxes for 'Internal Transfer Student', 'Transfer Via DGS', and 'Re-registered Student'. It also includes a field for '<Name of previous department>'.
- Section 4:** Titled 'Inferred Courses', it has a table with columns: Course Code, Course Name, ECTS Credits for Erasmus Exchange Students, Grade, Course or requirement to be exempted if transferred course is completed with a passing grade (with sub-columns for Course Code and Name for a Required Course, Elective Group Name for an Elective Requirement, and Credits), and Elective Requirement Exemptions only: Course code(s) of directly equivalent course(s), if any. There are 7 rows numbered 1 to 7. To the right of the table are 'Edit' and 'Add Row' buttons.

At the bottom, there are buttons for 'Convert To PDF', 'Apply Changes', and navigation arrows.

Fig. 40: Course Transfer Form Page (1/2) UI

Some of the required informations for the course transfer form is fetched by the application. However, user can edit these informations if there is a mistake in the fetched information. Moreover, the user can add the courses they had taken during the erasmus by clicking the add row button. If the user makes a mistake while adding the course, they can easily edit the information they have provided. Then, the user can apply the changes they have made, or they can convert the course transfer form to the PDF. They can move on the second page of the course transfer form using the arrow buttons.

Toolbar
<app_logo>

Course Transfer and Exemption Form for Undergraduate Students Part 2 of 2

Approved By	Name	Signature	Date
Exchange Coordinator (exchange students only)			
Chair			
Dean / Director			

Convert To PDF Apply Changes ← →

Fig. 41: Course Transfer Form Page (2/2) Student View UI

The student can see the approvement informations of the course transferm form on the second page. However, they cannot edit the second part of the course transfer form.

Toolbar
<app_logo>

Course Transfer and Exemption Form for Undergraduate Students Part 2 of 2

Section 2			
Approved By	Name	Signature	Date
Exchange Coordinator (exchange students only)			
Chair			
Dean / Director			

Edit

Convert To PDF **Apply Changes** **←** **→**

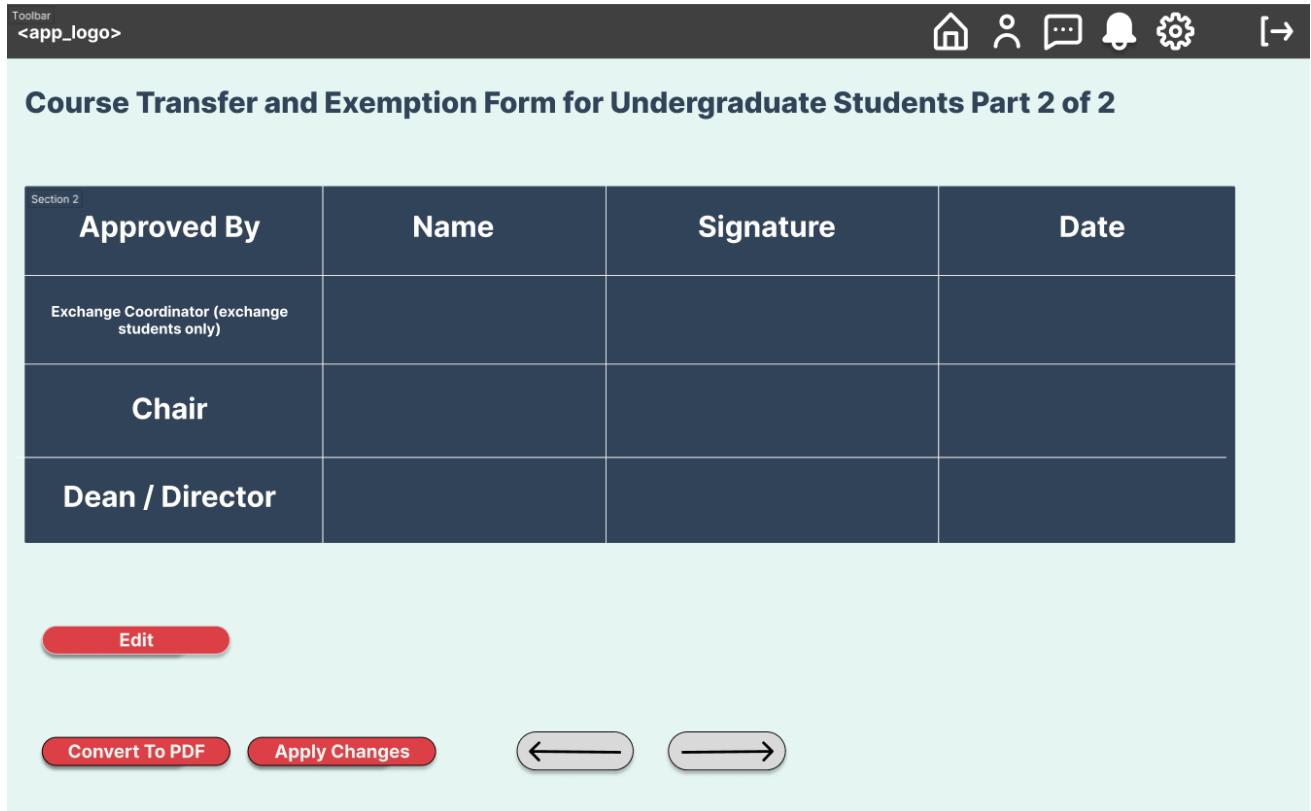


Fig. 42: Course Transfer Form Page (2/2) Authorized View UI

Authorized users can both see and edit the second part of the course transfer form by clicking on the edit button.

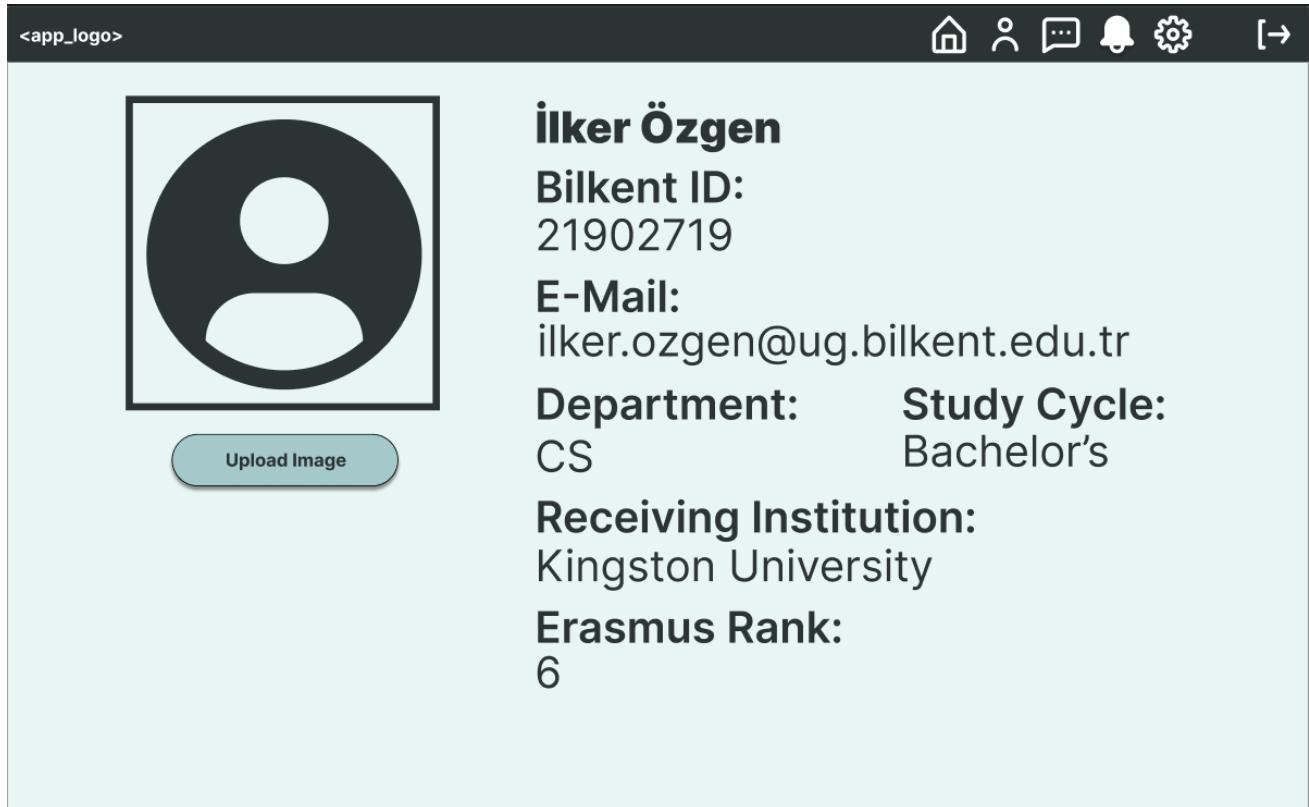


Fig. 43: Profile Page (Own, Student) UI

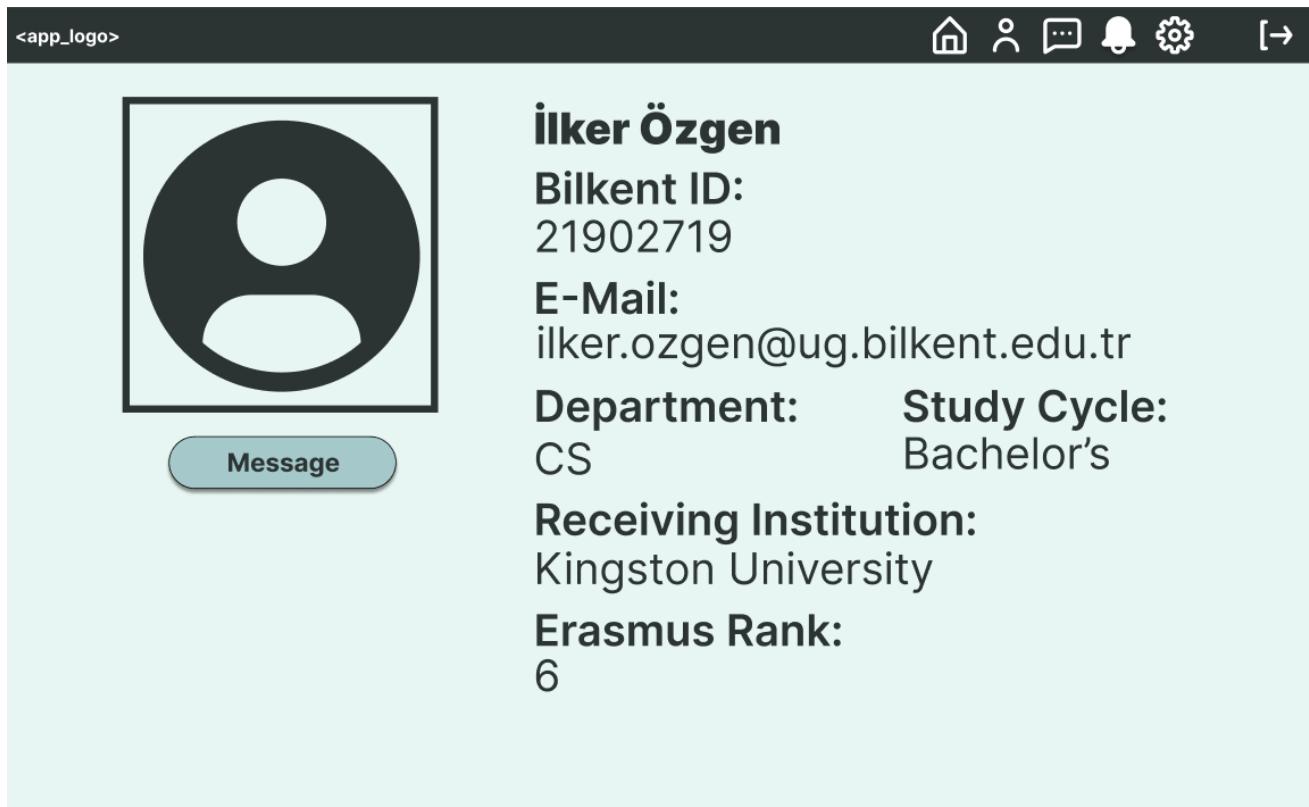


Fig. 44: Profile Page (Others, Student) UI

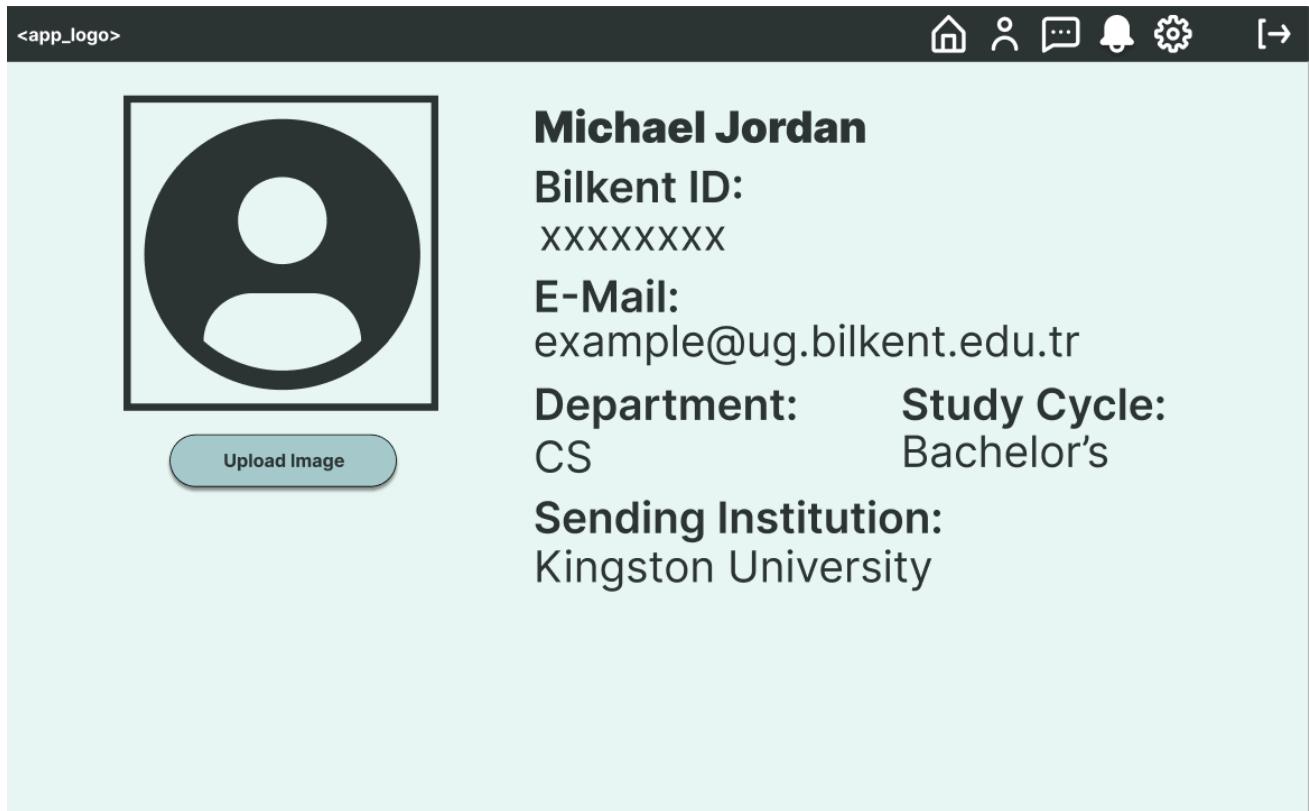


Fig. 45: Profile Page (Own, Incoming Student) UI

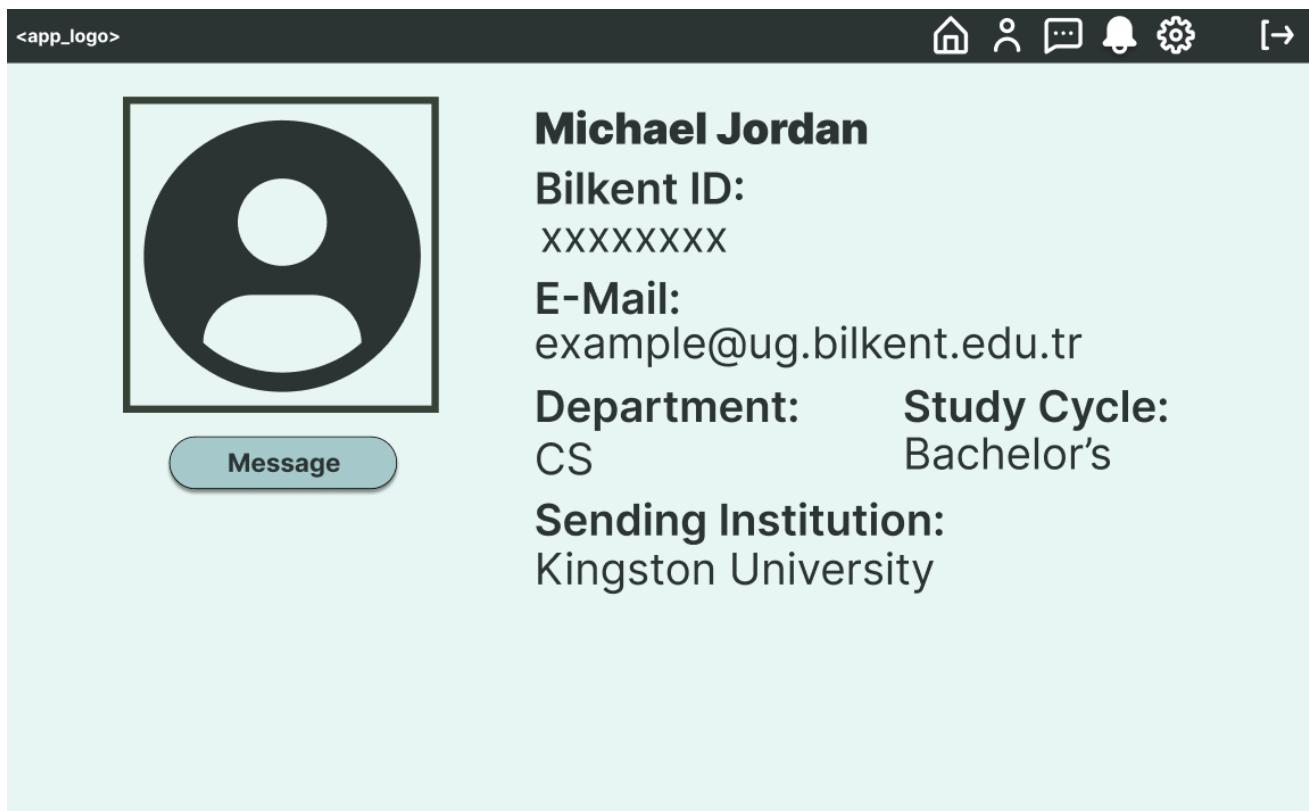


Fig. 46: Profile Page (Others, Incoming Student) 2 UI

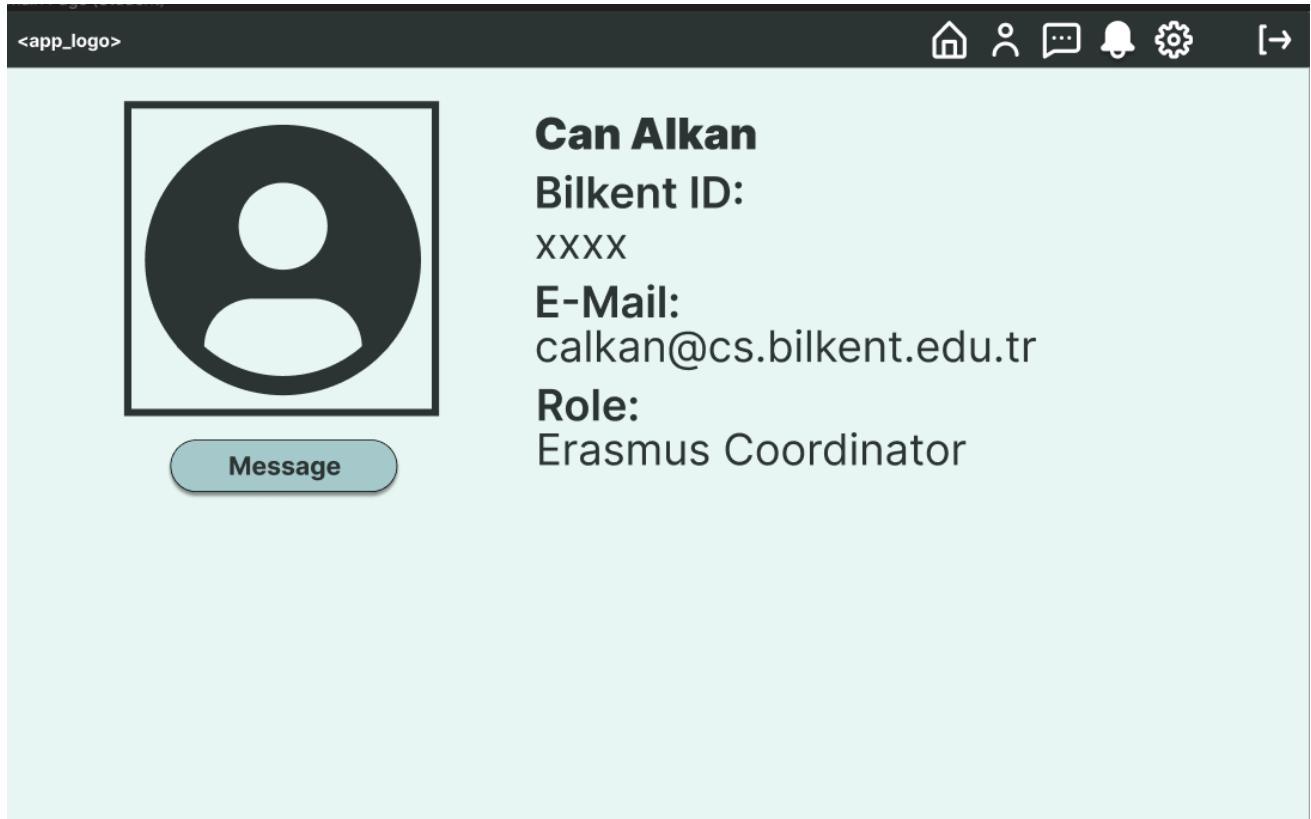


Fig. 47: Profile Page (Others, Coordinator & Instructor) UI

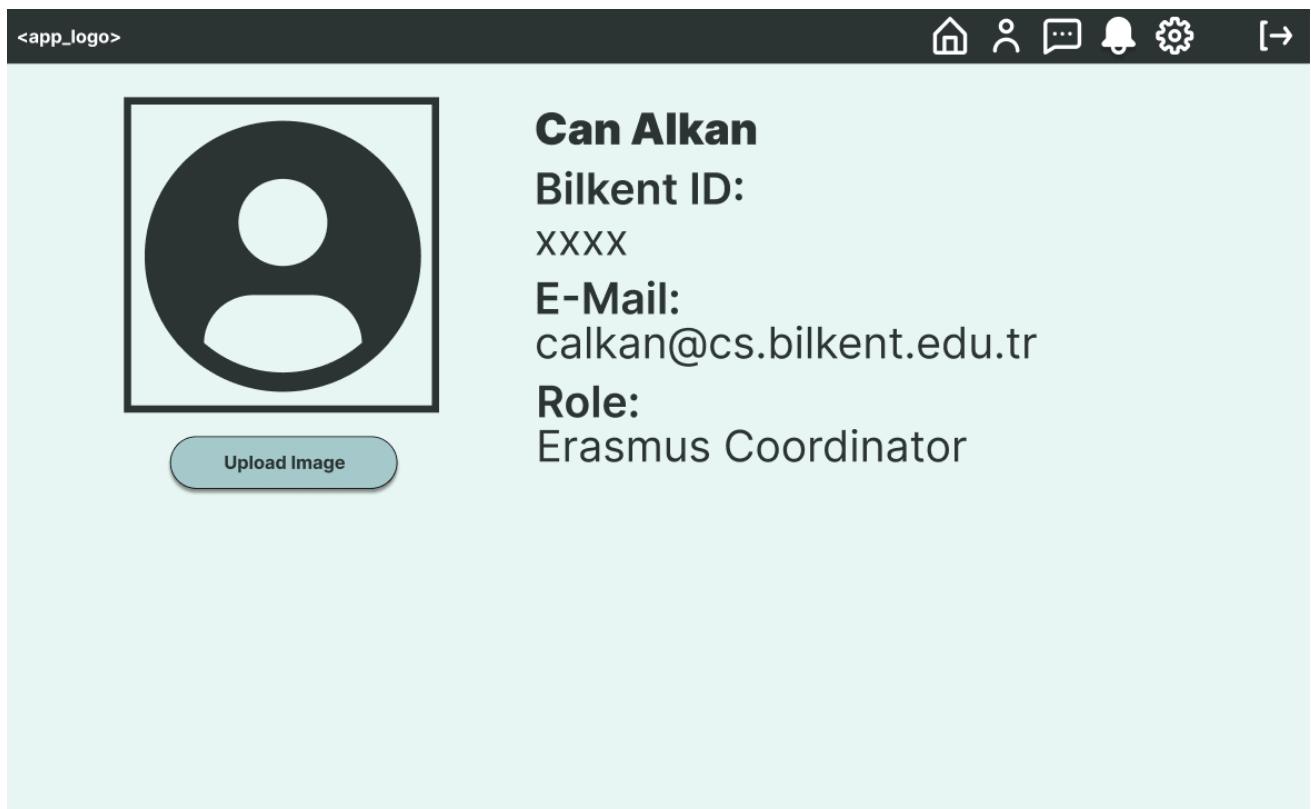


Fig. 48: Profile Page (Own, Coordinator & Instructor) UI

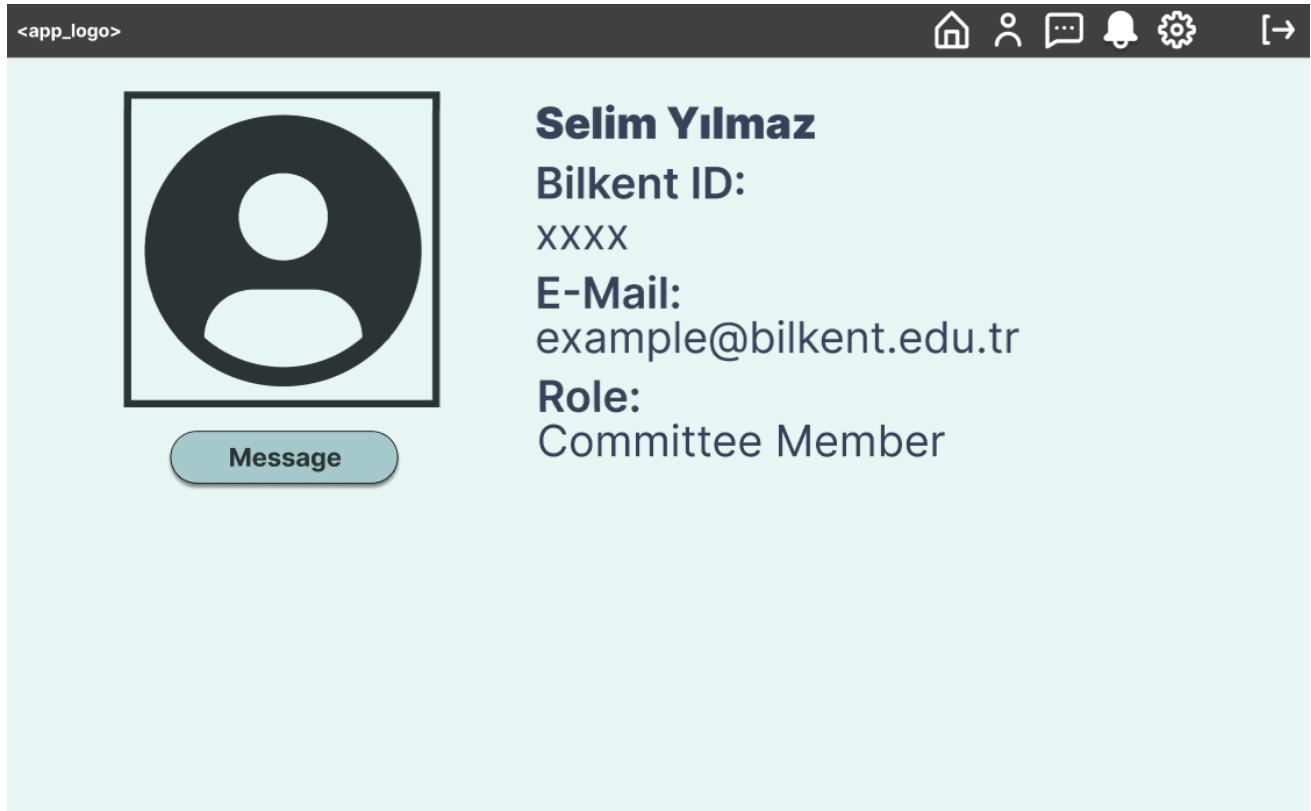


Fig. 49: Profile Page (Others, Others) UI

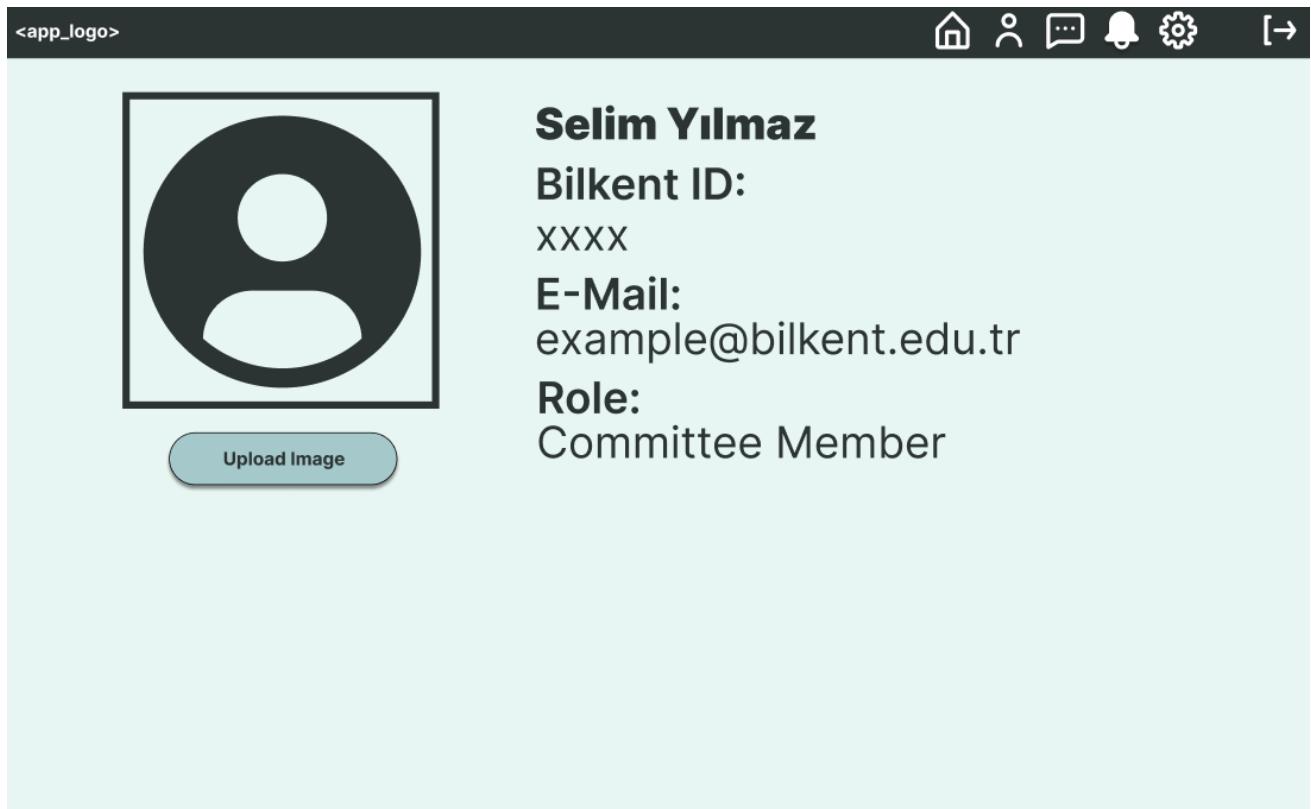


Fig. 50: Profile Page (Own, Others) UI

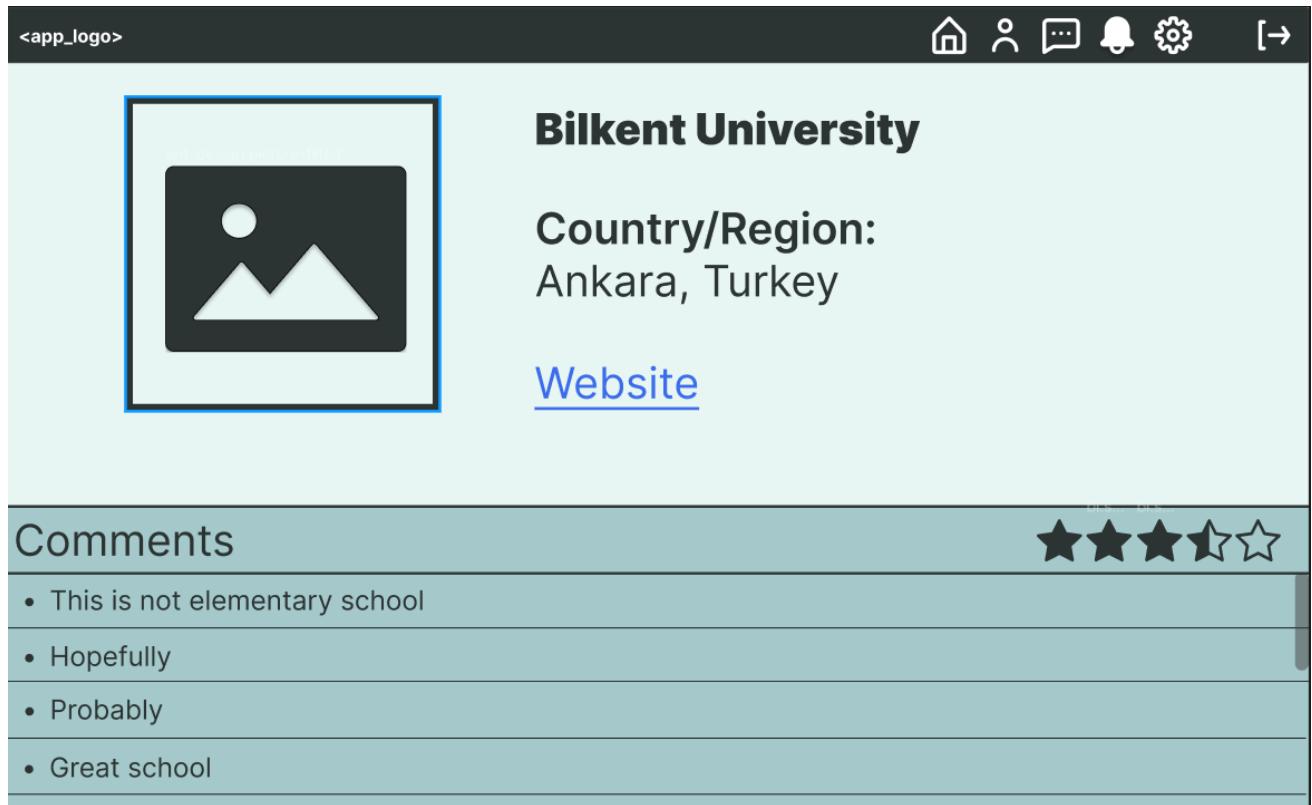


Fig. 51: Profile Page (University) UI

Students can see the comments and rates of the host universities that has been made by previous students.

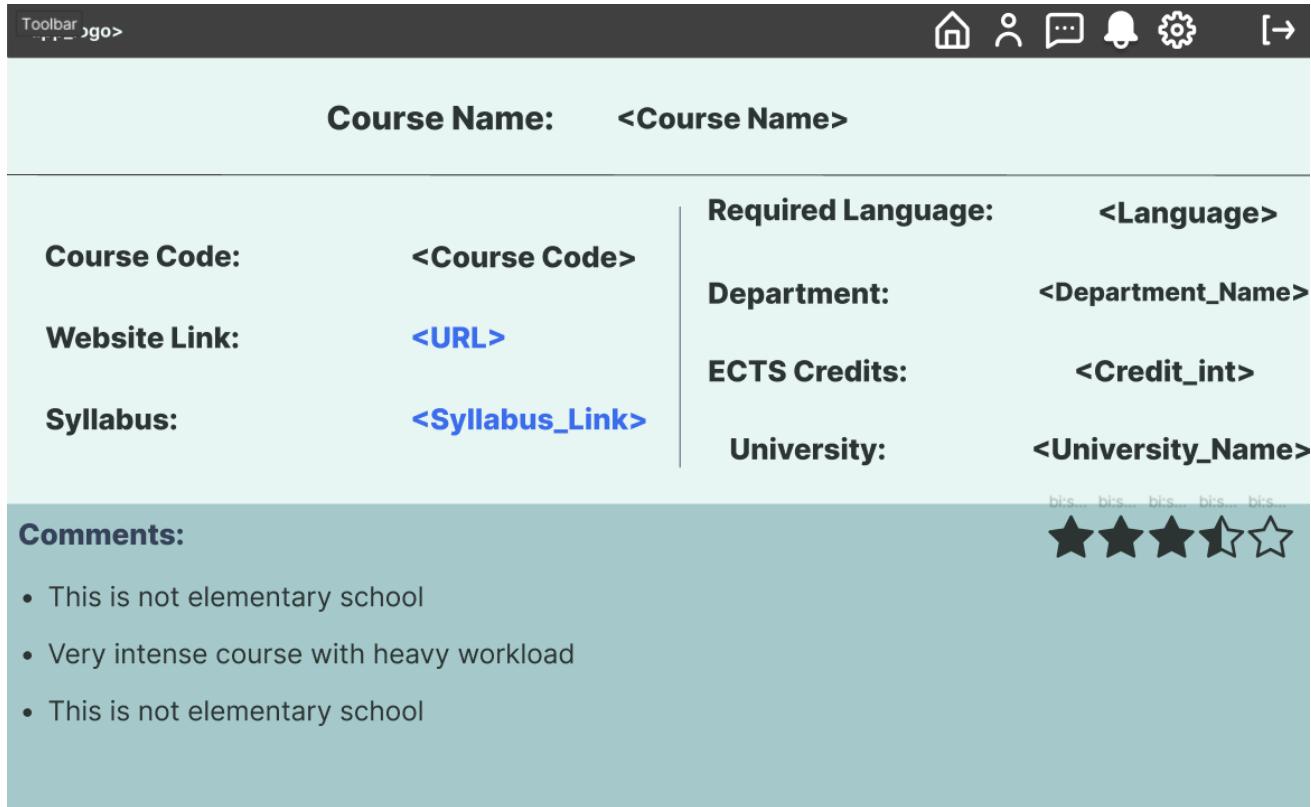


Fig. 52: Profile Page (Course) UI

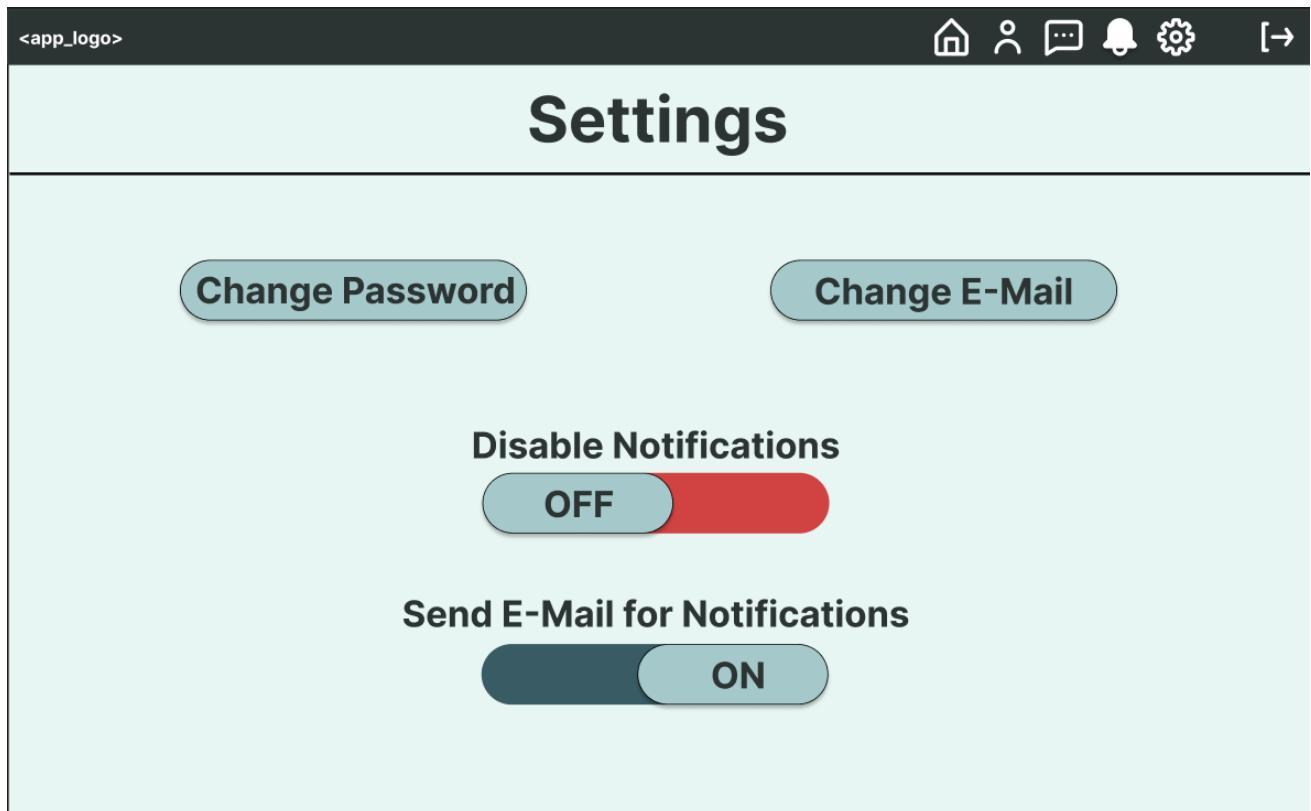


Fig. 53: Settings Page UI

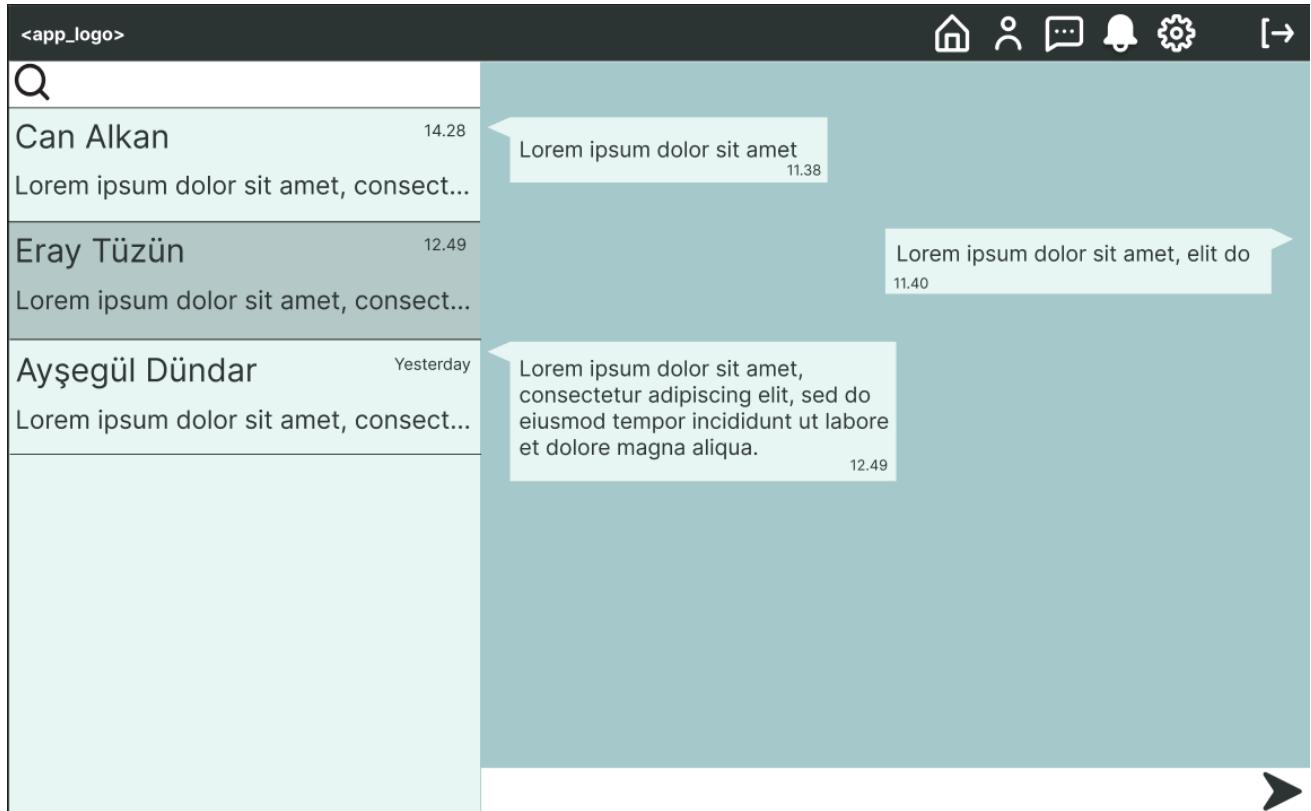


Fig. 54: Chat Page UI

3. References

- [1] 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.