



Bilkent University

Department of Computer Engineering

CS319 Term Project

Section 3

Group 3A

Bilasmus

Analysis Report

Group Members:

Alp Afyonluoğlu (22003229)
Emre Karataş (22001641)
Erkin Aydın (22002956)
Parsa Keihan (22001422)
Selin Bahar Gündoğar (22001514)

Instructor: Eray Tüzün

Teaching Assistant: Muhammad Umair Ahmed

1. Contents

1.	Introduction	3
2.	Proposed System	3
2.1.	Overview	3
2.2.	Functional Requirements	4
2.3.	Non-functional Requirements	5
3.	System Models	7
3.1.	Use-Case Diagram	7
3.2.	Class Diagram	19
3.3.	Dynamic Models Diagram	22
3.3.1.	State Machine Diagrams	22
3.3.1.1.	Application for Outgoing Students	22
3.3.1.2.	Course Request Process	22
3.3.2.	Activity Diagrams	23
3.3.2.1.	Course Transfer Form Evaluation	23
3.3.2.2.	Course Identification	23
3.3.3.	Sequence Diagrams	24
3.3.3.1.	Course Requesting	24
3.4.	User Interface	24

Analysis Report

Bilasmus

1. Introduction

Our application Bilasmus is a mode to communicate the Erasmus/Exchange mobility processes of outgoing Bilkent students and incoming students from partner universities with Bilkent Erasmus/Exchange coordinators and personnel. The previous mode of carrying out this procedure included all parties of Erasmus/Exchange communicating through emails and handing paperwork by hand. Thus, as the members of Erasmus, we decided to create an application that would help all people associated with the Erasmus/Exchange process.

This application will contribute to the efficiency of processing official and non-official Erasmus documents within Bilkent. The application will eradicate the paper-based communication and will minimize the need to send emails to related users. Thus, as the group, Bilasmus, we created this application to be used by Erasmus/Exchange coordinators, incoming and outgoing students, Faculty Committee Board, International Student Office, and Department Secretaries.

All users will be able to upload the necessary documents they need to upload as for their role and e-sign the documents through the application as well. The users will be getting notifications both on the application and through email to be reminded of other related users' progress. Furthermore, each user will be having a to-do list on the application. All users will be able to send messages to related users. Information of all users will be given on their profile pages. Moreover, the students can also send their course request forms through the application to both instructors, coordinators, and department secretaries.

2. Proposed System

The website application will manage the process of submitting and reviewing mobility documents related to Erasmus/Exchange of outgoing Bilkent students and incoming students.

2.1. Overview

The website application will manage the process of submitting and reviewing mobility documents related to Erasmus/Exchange of outgoing Bilkent students and incoming students.

The application will be serving all Erasmus users by offering different user type logins to the application. The outgoing students will be able to upload their necessary documents such as the preapproval form, language proficiency exam results, and course request forms. Moreover, outgoing student users can also see a list of all previously accepted courses before creating their course request forms. The incoming students will be able to send their course requests to the department secretary. Additionally, all Erasmus/Exchange coordinators will be able to review the documents uploaded by the outgoing students and e-sign it or reject the documents and request change from the students. Followingly, the coordinator can also upload the course transfer form through the application to the Faculty Committee Board for the approval of such document. The Faculty Committee Board will have the ability to make their assessments and state the outcome over the application of both the Pre-Approval form and the Course Transfer Form. On the other hand, the department secretary will be

responsible for receiving the course request from the incoming students and notify the students the status of registering them to their courses. The secretary has the right to reject a course if he/she redeems it as too difficult or too easy. Moreover, International Student Office is responsible for receiving the language proficiency exam results from outgoing students and uploading the excel sheet of students who applied for Erasmus/Exchange and their placement information. Followingly, all related users can message each other and add the deadlines of upload and sign dates of documents as reminders to other users on their to do lists which would also be sent as emails too. Additionally, outgoing students may not register to the courses stated in their Pre-Approval Form and in such case the student must re-submit their Pre-Approval Form again to the coordinator with the registered courses. The Faculty Committee Board will get to decide if the new course will be counted toward credit during the Course Transfer Form.

Only the students who are accepted to Erasmus/Exchange can use this platform. Every accepted outgoing student will receive an email from BCC containing the link to access Bilasmus for the first time. Once the student opens the website, then the student can designate a password and login with that password for future access. Moreover, Erasmus/Exchange coordinator can add a new course or remove a course if the coordinator decides so and does it through the application.

2.2. Functional Requirements

Authentication:

All users must be able to login to their account by using their email address and password. In case they forget their password, they must be able to reset their password via the login page, in which case the system must automatically send a password reset link to the user.

File upload:

System must allow users to upload files to the cloud system. The system must allow coordinators to upload course transfer forms, international office to upload transcripts, and outgoing students to upload pre-approval forms and learning agreement documents.

Viewing Application Status:

Incoming and outgoing students must be able to view their application status, which must provide information about whether their documents are approved or not. Approval actions taken by the staff must update the application status of students.

Managing Profiles:

Profile information of all users must be manageable by BCC. In addition to updating communication email addresses of users, BCC must be able to delete users. Each user must have an option to view their own saved profile information, and staff users must be able to add their signature to their own profile. Profile information of all students must be viewable by staff.

Messaging:

A messaging service must allow all users to send in-app messages to one another. Each user must be able to view incoming messages, reply to messages, search all users by name, and select a user to send a message. If a message is not seen by the receiver for a long time, the system will send an automated email to the message receiver to notify and remind about the message. Users must be able to add a message to their to-do list.

To-do Tracking:

A to-do panel must be accessible by all users for everyone to save and view their to-do tasks. To-do tasks must be added either manually by user or assigned depending on application status of students, such as whether someone sent a pre-approval form for approval or not. To-do items must have option to add remainders, in which case the system sends automated reminders at pre-determined times. Users must be able to set to-do items as completed or delete a chosen to-do item.

Notifications:

The system must send automated notifications about events and actions, such as reminders or approval of documents. Users must be able to receive and view notifications. Notifications must be delivered to the user as email as well.

Course Requests:

Incoming and outgoing students must be able to send course requests. Department secretary must have access to an interface to register courses of incoming students, and instructors must be able to view and approve requests of outgoing students. Approval status of requested courses must be saved.

Cancelling and Replacement:

After an outgoing student decides to cancel their application, the system must assign a new outgoing student from the waitlist. The system must notify the BCC for them to register the user to the system manually.

Pre-approval Form:

Outgoing students must be able to send a pre-approval form. The coordinator and faculty committee board must access an interface to approve or reject the document. If an outgoing student fails to take a pre-planned course during the program, a new pre-approval form must be sent by student. The coordinator must be able to view this form and create a course transfer form accordingly, which must be approved by the faculty committee board later.

User Registration:

BCC must have access to user management to be able to register new users to the system. After a new user account is registered, a welcome email must be sent to the email address of user by the system, which must include a link to set a password for the first time.

Signing and approving:

Bilkent staff users must be able to approve or reject documents waiting for approval. Their signature saved in their profile page must be used when they sign a document digitally. Approval status of documents must be saved for viewing later.

Help:

A help page must be accessible by all users. Frequently asked questions and general information about how to use the system must be found on the help page.

2.3. Non-functional Requirements

Performance:

- The website should be loaded in less than 3 seconds.
- After each interaction on the website, any output or result of interaction should be shown in less than 3 seconds.

- The user should be logged-in in less than 5 seconds.
- The minimum time to collect and display the data needed for a page from database should be less than 1 second.
- The system should change status of a student's document status after each approval in less than 20 seconds.
- The user information, name and email edited by BCC, should be updated in less than 3 seconds.
- All of the data in the system should get backed up every 1 hour.
- The in-app messages should be sent and received in less than 3 seconds.

Usability:

- The system should be user-friendly and basic for common use.
- The buttons, pop-ups, and notifications should be useful and aligned.
- Buttons and toolbar text fields should be clickable.
- The colors used in the background and foreground of each page should be consistent and not eye-straining.
- The sections and fields in each page should be related to the user based on which actor is using it.
- The help button should let the user contact admins for any technical problem.

Reliability:

- The system should be reliable such that no unexpected and unwanted events happen to the documents in the system, such as vanishing.
- System should not be down for more than a few minutes.
- The data should be preserved to prevent data loss.
- Any incident should be recorded and documented.
- System should be able to handle corner cases appearing due to user activities, such as invalid inputs and requests.
- The system should be secure.

Supportability:

- The UI should appear the same to every user, that is, images, icons, etc. shouldn't change as browser changes.
- The website should be reachable by WebKit and Gecko using Browsers, such as Chrome and Safari for WebKit and Mozilla Firefox for Gecko.
- The website should be reachable on devices with at least 2 GB RAM.

3. System Models

3.1. Use-Case Diagram

The actor hierarchy can be seen in the below diagram.

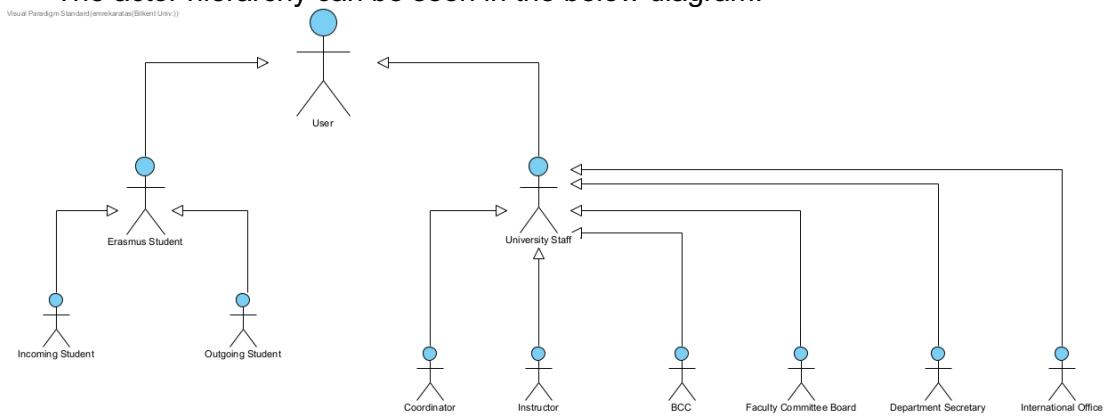


Figure 1: Actor Hierarchy

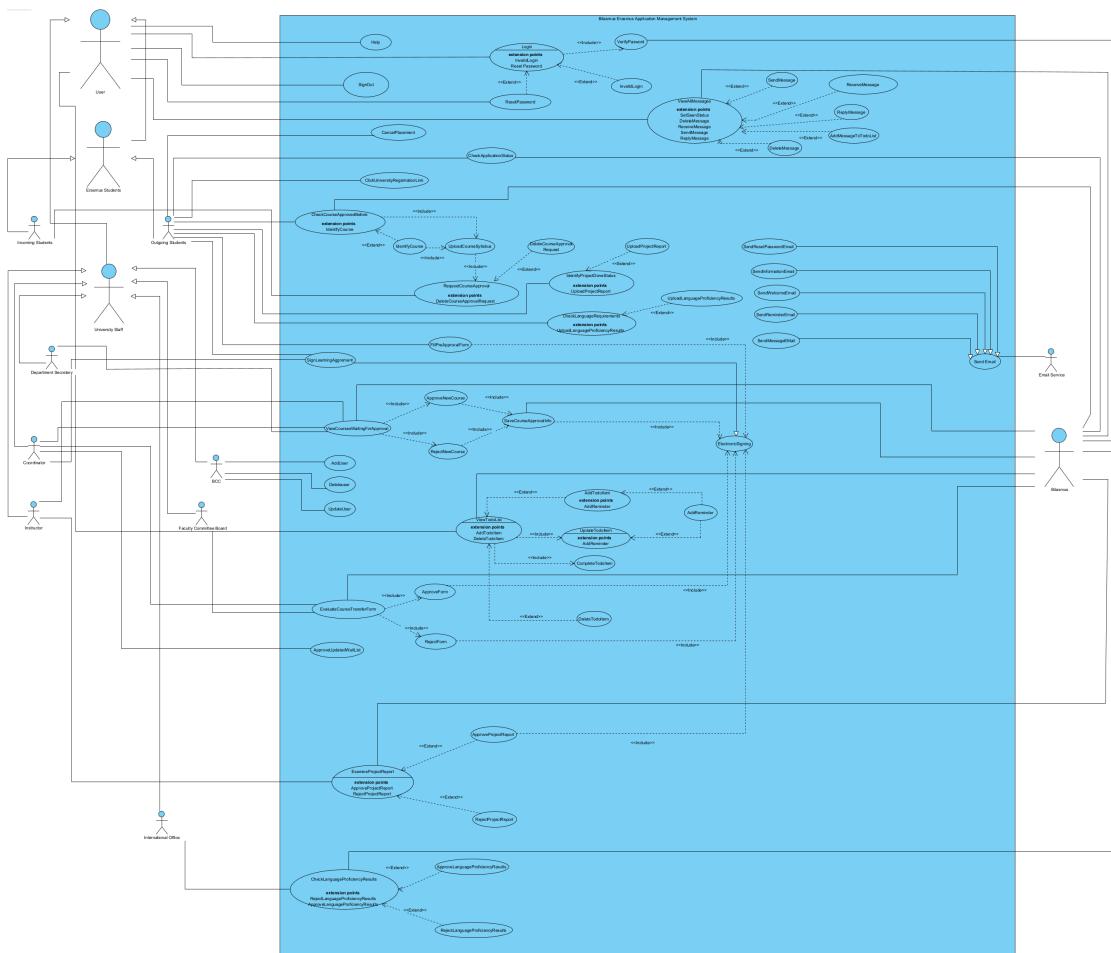


Figure 2: Use Case Diagram

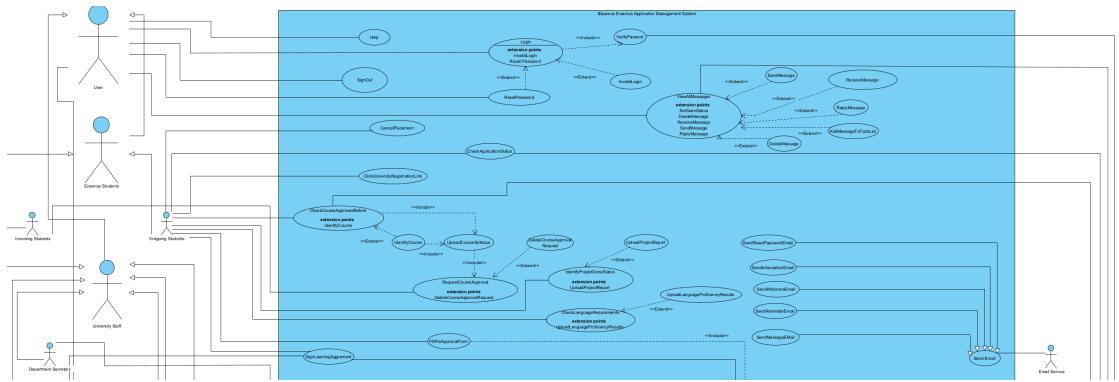


Figure 3: Close up of Use Case Diagram

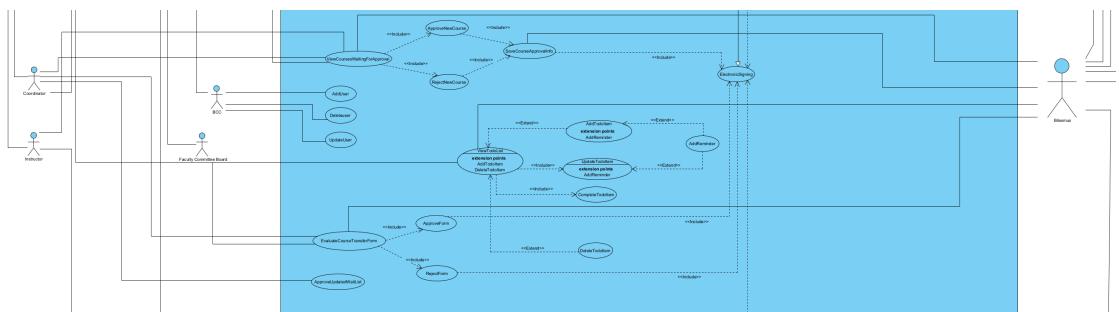


Figure4: Close up of Use Case Diagram

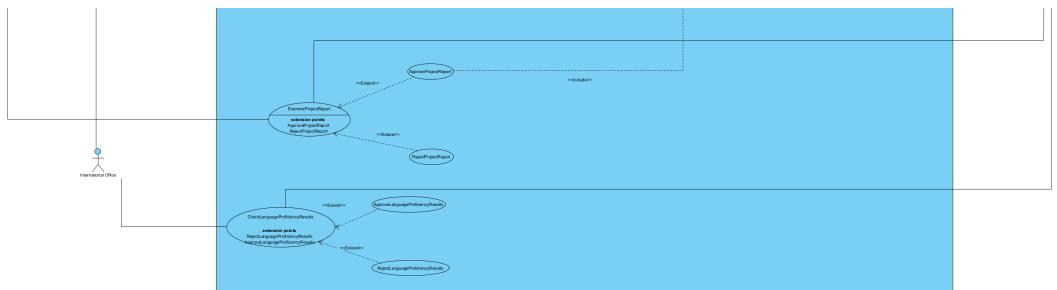


Figure5: Close up of Use Case Diagram

Use Case: Login

Participating Actor(s): User

Entry Conditions: Actor is not logged in.

Exit Conditions: Actor provides ID and password.

The Flow of Events:

- Actor clicks on ID/password fields
- Actor provides necessary information
- If information is valid actor

Use Case: Verify Password

Participating Actor(s): User, Bilasmus

Entry Conditions: User provided ID and password.

Exit Conditions: User verifies his/her password correctly.

The Flow of Events:

- User gets verification message from Bilasmus.
- User provides verification code to the Bilasmus.
- If code is valid User logs into to Bilasmus.

Use Case: Invalid Login

Participating Actor(s): User

Entry Conditions: Actor provided wrong password or ID.

Exit Conditions: This use case extends Login use case. It is initiated by the system whenever User provides wrong ID and/or password to the system.

The Flow of Events:

- Actor tries to log in.
- Actor enters invalid information to the required fields.
- The system generates an error message, and the actor must re-enter the faulty information.

Use Case: Reset Password

Participating Actor(s): User

Entry Conditions: Actor provided wrong password.

Exit Conditions: User resets his/her password successfully.

The Flow of Events:

- User gets reset password e-mail from E-mail service.
- User provides the code in the mail.
- User creates new password.
- User's password has been changed.

Use Case: Help

Participating Actor(s): User

Entry Conditions: Actor is in main page.

Exit Conditions: Actor gets helpful information and returns to the main page.

The Flow of Events:

- User clicks Help button at the top of menu bar.
- User goes help page.
- User may investigate Q&A section from help page.
- User returns to main page.

Use Case: Sign Out

Participating Actor(s): User

Entry Conditions: Actor has not signed out.

Exit Conditions: Actor signs out and leaves the program.

The Flow of Events:

- User clicks on sign out button.
- Confirmation pop-up page appears, and user selects yes or no from pop-up screen.
- If yes is selected, user signs out and leaves the program.
- If no selected, user will return to main page and stays logged in.

Use Case: ViewAllMessages

Participating Actor(s): User

Entry Conditions: Actor is in main page.

Exit Conditions: Actor can see all messages sent to him/her.

The Flow of Events:

- User encounters with message section in the main page.
- User can see all messages incoming/outgoing.

Use Case: Send Message

Participating Actor(s): User

Entry Conditions: Actor is in View All Messages section.

Exit Conditions: Actor sends message to another actor in the system.

The Flow of Events:

- Actor clicks on send message button.
- Actor selects the person who is going to receive the message.
- Actor writes his/her message in provided area.
- By clicking on “Send” button, message has gone to another actor.

Use Case: Receive Message

Participating Actor(s): User

Entry Conditions: Another actor has sent a message to main actor.

Exit Conditions: Main actor receives message from sender actor.

The Flow of Events:

- When a message received, notification will be pushed.
- Actor clicks on the person who has sent the message.
- Actor receives the message sent by another actor.

Use Case: Reply Message

Participating Actor(s): User

Entry Conditions: Actor receives a message.

Exit Conditions: Actor sends a reply message to the sender actor.

The Flow of Events:

- Actor clicks on the message bubble in the conversation.
- Actor writes his/her message as a reply to received message.
- Actor clicks on “Send” button to send reply message.
- Notification is sent to the receiver actor.

Use Case: Add Message to Todo List

Participating Actor(s): User

Entry Conditions: Actor receives a message.

Exit Conditions: A to-do list item has been added to the actor's to-do list.

The Flow of Events:

- Actor clicks on the message bubble in the conversation.
- The actor selects the message that he/she wants to add to his/her to-do list.
- The actor selects “Add to to-do List” button.
- A new to do list item has been added to the actor's to do list.

Use Case: Reply Message

Participating Actor(s): User

Entry Conditions: Actor receives a message.

Exit Conditions: Message sent to the actor has been deleted from the system.

The Flow of Events:

- Actor clicks on the message bubble in the conversation.
- Actor selects the message he/she wants to delete.

- Actor clicks on “Delete Message” button to delete message.
- That message has been deleted from the system.

Use Case: Cancel Placement

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been placed to a university in Bilasmus system.

Exit Conditions: Actor has been removed his/her application and deleted from Bilasmus app.

The Flow of Events:

- Actor clicks on the “Cancel Placement” button.
- Confirmation pop-up which includes “Yes” and “No” buttons has been sent to the actor.
- If the actor clicks on “Yes” button, the application process will be terminated, and the actor will be deleted from the Bilasmus app.
- Information e-mail will be sent to the International Office.
- If actor clicks on “No” button, he/she will be directed to main page.

Use Case: Check Application Status

Participating Actor(s): Outgoing Student, Bilasmus

Entry Conditions: Actor has been registered to Bilasmus app.

Exit Conditions: Actor learns about the details of his/her application process.

The Flow of Events:

- Actor is in the main page dedicated to himself/herself.
- The application process information will be checked from Bilasmus app.
- Outgoing student learns about the details of his/her application process.

Use Case: Click University Registration Link

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been registered to Bilasmus app.

Exit Conditions: Actor clicks on university's registration link and registers to target university

The Flow of Events:

- Actor is in the main page dedicated to himself/herself.
- Actor hits “Click University Registration Link” button.
- Actor is directed to target university's registration page.

Use Case: Check Course Approved Before

Participating Actor(s): Outgoing Student, Bilasmus

Entry Conditions: Actor starts to decide to the courses he/she will take in Erasmus.

Exit Conditions: Actor learns whether the course is an approved course before or not.

The Flow of Events:

- Actor is in the add course section.
- The information regarding his/her Erasmus university and the courses approved in that university are given in a table.
- If the course is one of the approved courses, he/she will be directed to “Upload Syllabus” page.
- Else, actor will be directed to “Identify Course” section.

Use Case: Identify Course

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has intended to take a course which is not approved by Bilkent University before.

Exit Conditions: Details about the course is submitted.

The Flow of Events:

- Actor has been redirected from “Check Course Approved Before” section.
- Actor gives information about course name, course code and the department of the course, whether the course is must, elective or additional course in the provided area.
- Actor hits “Submit Course Details” button.
- The actor is directed to “Upload Course Syllabus” section.

Use Case: Upload Course Syllabus

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has identified not-approved course **OR** actor has an intend to take a course which is approved before.

Exit Conditions: Actor uploaded course syllabus in pdf format and this document has been sent to related coordinator of the course.

The Flow of Events:

- Actor hits “Select File” button from the system.
- The actor selects related course syllabus from the computer.
- Actor previews the document.
- Actor hits “Submit” button and the document has been saved to Bilasmus in pdf format.

Use Case: Request Course Approval

Participating Actor(s): Outgoing Student, Incoming Student

Entry Conditions: Actor has uploaded the course syllabus successfully.

Exit Conditions: Actor makes official request to the coordinator who is related to the course.

The Flow of Events:

- The actor previews previous steps (the course information, course syllabus document) if he/she is an outgoing student.
- If the actor is an incoming student, the actor requests the courses he/she applied so far.
- Actor hits “Request Course” button.

Use Case: Delete Course Approval Request

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has already sent an official request about the course.

Exit Conditions: Actor’s waiting approval course list will be updated, and the dedicated course request will be deleted from the system.

The Flow of Events:

- Actor hits “Delete Course Request” button.
- Confirmation pop-up screen, which includes “Yes” and “No” buttons will appear.
- If “Yes” is selected, the course request will be deleted, and actor’s waiting approval course list will be updated.
- If “No” is selected, actor will be directed back to the course approval section.

Use Case: Identify Project Done Status

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been returned from Erasmus to his/her main university.

Exit Conditions: If actor has done a project in the Erasmus program, actor will be directed to “Upload Project Report” page.

The Flow of Events:

- Actor hits “Identify Project Done Status” button.
- Information screen, which includes “Yes” and “No” buttons will appear.
- If “Yes” is selected, actor will be directed to “Upload Project Report” page.
- If “No” is selected, actor will be directed back to the main page.

Use Case: Upload Project Report

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has identified that he/she has done project in Erasmus program.

Exit Conditions: Actor’s project report will be uploaded in pdf format successfully.

The Flow of Events:

- Actor hits “Select File” button from the system.
- Actor selects related project report from the computer.
- Actor previews the document.
- Actor hits “Submit” button and the document has been saved to Bilasmus in pdf format.

Use Case: Check Language Requirements

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been registered to Bilasmus app.

Exit Conditions: Actor’s will be directed to “Upload Language Proficiency Results” page if he/she is placed a university whose education language(s) is/are not English.

The Flow of Events:

- Actor checks language requirements of the university which he/she has been placed.
- If university’s education language(s) is/are different from English, the actor will be directed to “Upload Language Proficiency Result” section.

Use Case: Upload Language Proficiency Result

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been placed to a university whose education language(s) is/are different from English.

Exit Conditions: Actor’s language proficiency results will be uploaded in pdf format successfully.

The Flow of Events:

- Actor hits “Select File” button from the system.
- Actor selects related language proficiency result from the computer.
- Actor previews the document.
- Actor hits “Submit” button and the document has been saved to Bilasmus in pdf format.

Use Case: Fill Pre-approval Form

Participating Actor(s): Outgoing Student

Entry Conditions: Actor has been registered to Bilasmus app.

Exit Conditions: Actor has filled the pre-approval form successfully.

The Flow of Events:

- Actor fills required areas regarding his/her pre-approval.
- Actor hits “Submit” button.
- Since this form is an official document, the actor will be directed to “Electronic Signing” section.

Use Case: Sign Learning Agreement

Participating Actor(s): Outgoing Student, Coordinator

Entry Conditions: Actor (Outgoing Student) has completed all steps before going to Erasmus.

Exit Conditions: Actors have signed the learning agreement successfully.

The Flow of Events:

- Actors preview the document.
- Actor hits “Electronic Signing” button.
- The document has been signed by actors electronically.

Use Case: View Courses Waiting for Approval

Participating Actor(s): Department Secretary, Coordinator, Instructor

Entry Conditions: Actors have been registered to the system.

Exit Conditions: Actors have seen the course approval waitlist.

The Flow of Events:

- Actors hit “View Courses Waiting for Approval” button.
- New courses waiting for approval will be listed to the actors.
- Actors will be directed to “Approve New Course” or “Reject New Course” depending on their behavior.

Use Case: Approve New Course

Participating Actor(s): Department Secretary, Coordinator, Instructor

Entry Conditions: Actors have seen the course approval waitlist.

Exit Conditions: Actors approve the course.

The Flow of Events:

- Actors hit “Approve New Course” button.
- Courses waiting for approval will be listed to the actors.
- Actors will select the course they want to approve.
- The actor will see the details of the course (including Syllabus in the pdf format).
- Actors may hit “Approve” button.

Use Case: Reject New Course

Participating Actor(s): Department Secretary, Coordinator, Instructor

Entry Conditions: Actors have seen the course approval waitlist.

Exit Conditions: Actors reject the course.

The Flow of Events:

- Actors hit “Reject New Course” button.
- Courses waiting for approval will be listed to the actors.
- Actors will select the course they want to reject.
- The actor will see the details of the course (including Syllabus in the pdf format).
- Actors hit “Reject” button.

Use Case: Save Course Approval Info

Participating Actor(s): Department Secretary, Coordinator, Instructor

Entry Conditions: Actors have already rejected or approved the course.

Exit Conditions: Actors’ behavior will be saved to Bilasmus.

The Flow of Events:

- Actors hit “Save” button.
- The behavior has been saved to Bilasmus.

Use Case: Add User

Participating Actor(s): BCC

Entry Conditions: Dedicated user have not been registered to the system.

Exit Conditions: Dedicated user will be registered to the system.

The Flow of Events:

- Actor hits “Add User” button.
- The actor identifies whether the user is a student or a staff member.
- Actor fills identity information regarding the user.
- Actor hits “Save” button
- Dedicated user has been registered to the system.
- Welcome e-mail will be sent to dedicated user.

Use Case: Delete User

Participating Actor(s): BCC

Entry Conditions: Dedicated user has already been registered to the system.

Exit Conditions: Dedicated user has been deleted from the system.

The Flow of Events:

- Actor hits “Delete User” button.
- Actor identifies whether user a student or a staff member.
- Actor selects the to be deleted user from list of users.
- Actor hits “Save” button
- Dedicated user has been deleted from the system.

Use Case: Update User

Participating Actor(s): BCC

Entry Conditions: Dedicated user has already been registered to the system.

Exit Conditions: Dedicated user's information has been updated.

The Flow of Events:

- Actor hits “Update User” button.
- Actor identifies whether user a student or a staff member.
- Actor selects the to be updated user from list of users.
- Actor fills necessary information to be updated.
- Actor hits “Save” button
- Dedicated user's information has been updated in the system.

Use Case: View Todo List

Participating Actor(s): User

Entry Conditions: User has already been registered to the system.

Exit Conditions: User has seen his/her to-do list.

The Flow of Events:

- The actor hits “View to-do list” button.
- Actor has seen his/her to-do list.

Use Case: Add Todo Item

Participating Actor(s): User

Entry Conditions: User has already viewed the to-do list.

Exit Conditions: User has added new item to to-do list.

The Flow of Events:

- The actor hits “add to-do item” button.
- Actor identifies to-do list and its' deadline.
- Actor hit “Save” button.

Use Case: Update Todo Item

Participating Actor(s): User

Entry Conditions: User has already viewed the to-do list.

Exit Conditions: User has updated to-do item.

The Flow of Events:

- The actor hits “update to-do item” button.
- Actor changes the information to be updated
- Actor hit “Save” button.

Use Case: Complete Todo Item

Participating Actor(s): User

Entry Conditions: User has already viewed the to-do list.

Exit Conditions: User has completed to-do item.

The Flow of Events:

- Actor hits “complete to-do item” button.
- Todo item has been labeled as done.

Use Case: Delete Todo Item

Participating Actor(s): User

Entry Conditions: User has already viewed the to-do list.

Exit Conditions: User has deleted to-do item.

The Flow of Events:

- Actor hits “delete to-do item” button.
- A pop-up screen, which includes “Yes” and “No” buttons, will appear for confirmation.
- If “Yes” is selected, the regarding to-do item will be deleted.

Use Case: Add Reminder

Participating Actor(s): User

Entry Conditions: User has added a new to-do item or updated the to-do item.

Exit Conditions: A reminder about to-do item will be created.

The Flow of Events:

- Actor hits “Add Reminder” button.
- The actor selects the to-do item which will have the reminder.
- Reminder e-mail will be sent to the user by E-mail service.

Use Case: Evaluate Course Transfer Form

Participating Actor(s): Coordinator, Faculty Committee Board

Entry Conditions: Student has submitted his/her course transfer form.

Exit Conditions: Course transfer form will be sent to approval or rejection.

The Flow of Events:

- Actors hit “Evaluate Course Transfer Form” button.
- Actors select the course transfer form.
- Course transfer form will be evaluated by previewing necessary documents (syllabus)

Use Case: Approve Form

Participating Actor(s): Coordinator, Faculty Committee Board

Entry Conditions: Student’s course transfer form has been evaluated.

Exit Conditions: Course transfer form will be approved.

The Flow of Events:

- Actors select the course transfer form from the list.
- Actors hit “Approve Form” button.

Use Case: Approve Form

Participating Actor(s): Coordinator, Faculty Committee Board

Entry Conditions: Student’s course transfer form has been evaluated.

Exit Conditions: Course transfer form will be rejected.

The Flow of Events:

- Actors select the course transfer form from the list.
- Actors hit “Reject Form” button.

Use Case: Approve Updated Waitlist

Participating Actor(s): Coordinator

Entry Conditions: A student has cancelled his/her placement and waiting list is updated.

Exit Conditions: Updated waitlist will be approved.

The Flow of Events:

- Actor can see the change has been done by Bilasmus automatically.
- Actor approves updated waitlist

Use Case: Examine Project Report

Participating Actor(s): Instructor

Entry Conditions: A student has uploaded his/her project report.

Exit Conditions: Project report will be examined by the instructor.

The Flow of Events:

- Actor can see project reports.
- Actor can examine the project report by viewing the project report in a pdf format.

Use Case: Approve Project Report

Participating Actor(s): Instructor

Entry Conditions: Project report has been examined.

Exit Conditions: Project report has been approved.

The Flow of Events:

- Actor selects related project report.
- Actor hits “Approve” button.

Use Case: Approve Project Report

Participating Actor(s): Instructor

Entry Conditions: Project report has been examined.

Exit Conditions: Project report has been rejected.

The Flow of Events:

- Actor selects related project report.
- Actor hits “Reject” button.

Use Case: Electronic Signing

Participating Actor(s): Outgoing Student, Coordinator, Faculty Committee Board, Instructor, Department Secretary

Entry Conditions: Related documents are submitted.

Exit Conditions: Document has been signed electronically.

The Flow of Events:

- Actors select related document.
- Actor hits “Sign” button.

Use Case: Check Language Proficiency Results

Participating Actor(s): International Office

Entry Conditions: Actor has been registered to the system.

Exit Conditions: Actor has viewed language proficiency results.

The Flow of Events:

- Actor selects “Check Language Proficiency Results” button.

- Actor can see the list of language proficiency results.
- Actor can preview language proficiency results in pdf format.

Use Case: Approve Language Proficiency Result

Participating Actor(s): International Office

Entry Conditions: Actor has checked language proficiency result.

Exit Conditions: Actor has approved language proficiency result.

The Flow of Events:

- Actor selects the result which should be approved.
- Actor can hit “Approve” button.

Use Case: Reject Language Proficiency Result

Participating Actor(s): International Office

Entry Conditions: Actor has checked language proficiency result.

Exit Conditions: Actor has rejected language proficiency result.

The Flow of Events:

- Actor selects the result which should be rejected.
- Actor can hit “Reject” button.

Use Case: Send Reset Password E-mail

Participating Actor(s): E-mail Service

Entry Conditions: User has reset his/her password.

Exit Conditions: Randomly generated password will be sent to the user.

The Flow of Events:

- Actor receives the user's e-mail from Bilasmus app

Use Case: Send Information E-mail

Participating Actor(s): E-mail Service

Entry Conditions: User(s) have uploaded documents

Exit Conditions: Information e-mail will be sent to related users in the system.

The Flow of Events:

- Actor receives the user's e-mail from Bilasmus app

Use Case: Send Welcome E-mail

Participating Actor(s): E-mail Service

Entry Conditions: BCC added a new user.

Exit Conditions: This new user's password and related information have been sent to the new user.

The Flow of Events:

- Actor receives the user's e-mail from Bilasmus app

Use Case: Send Reminder E-mail

Participating Actor(s): E-mail Service

Entry Conditions: User has created a new reminder.

Exit Conditions: Reminder event's information will be sent to the user.

The Flow of Events:

- Actor receives the user's e-mail from Bilasmus app

Use Case: Send Message E-mail

Participating Actor(s): E-mail Service

Entry Conditions: User have received a new message.

Exit Conditions: Message information will be sent to the user.

The Flow of Events:

- Actor receives the user's e-mail from Bilasmus app

Use Case: Send E-mail

Participating Actor(s): E-mail Service

Entry Conditions: E-mail Service has received the e-mail of target user.

Exit Conditions: Related E-mail will be sent.

The Flow of Events:

- Actor sends related E-mail.

3.2. Class Diagram

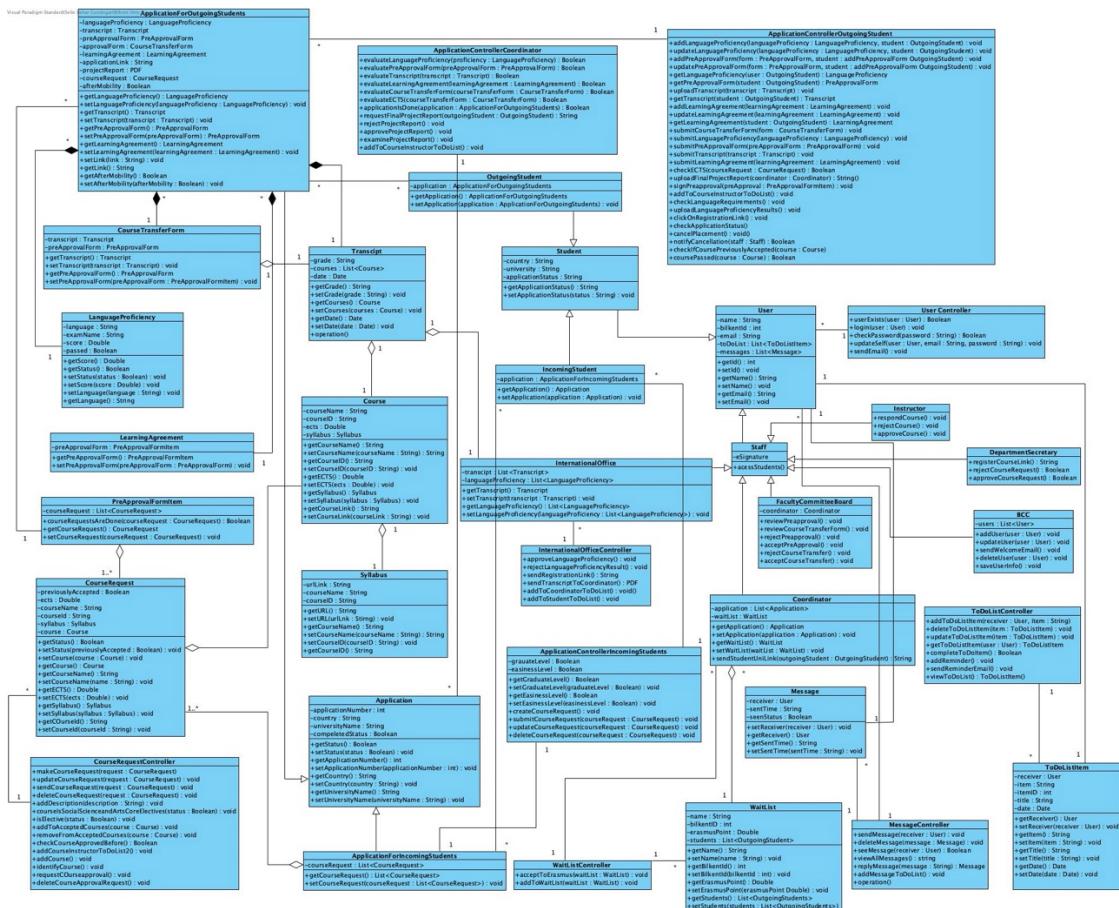


Figure 6: Bilkimus Class Diagram

User Class: User class is the parent class of Student and Staff classes. It contains the common properties of all users such as name, bilkentid, email, <List> toDoList, <List> messages of all users.

Student Class: Student class is the parent class of OutgoingStudent and IncomingStudent classes. Student class has common properties of other types of users such as the name of the Erasmus/Exchange university, the country the student is going to, and the application status.

IncomingStudent Class: This class inherits the Student Class and is a class for students coming to Bilkent University as Erasmus/Exchange students. This class has

an application of the type ApplicationForIncomingStudents. This class aggregates ApplicationForIncomingStudents Class.

OutgoingStudent Class: This class inherits the Student Class and has an application for ApplicationForOutgoingStudents. Bilkent university students who will go to Exchange/Erasmus universities are of this type. This class aggregates ApplicationForOutgoingStudents Class.

Staff Class: This class inherits the User class and has the property of e-signing documents and accessing students.

UserController Class: This controller class has basic methods for all user types such as checking the user password and user email for login page and also the ability to send emails.

Instructor Class: This class extends the Staff class. Instructor class can view course request forms from outgoing students and reject or approve the courses. This class aggregates the Transcript Class.

DepartmentSecretary Class: This class is a child class of the Staff Class. It includes the properties of reviewing course request forms of incoming students to Bilkent and registering them to Bilkent courses.

BCC Class: This class is a child class of Staff Class and is the admin user of the application. It has the properties of adding, deleting, and updating users and user information. This class controls the other users in an administrative way.

Coordinator Class: This class extends the Staff Class and has the properties of viewing through the application process of all outgoing students. It also has the WaitList class to access the list and update the list accordingly.

FacultyCommitteeBoard Class: This class extends the Staff Class and has the properties of reviewing preapproval forms and the course transfer forms of all outgoing students.

InternationalOffice Class: This class extends the Staff Class and has the properties transcripts and language proficiencies of all outgoing students.

InternationalOfficeController: This is a controller class of the InternationalOffice Class, and the user has the properties of approving language proficiencies of outgoing students, and receiving the transcripts of outgoing students from their host universities after returning from their Erasmus/Exchange program. They also send the transcript to the coordinators.

Message Class: This class creates messages for users to send each other and has the properties of receiver, sent time, and seen status.

MessageController Class: This class manages the Message Class and sends, deletes, and views all messages. It also can add messages to the ToDoList of other users.

ToDoListItem Class: This class creates a ToDoListItem class, and it has the properties of receiver, item, item id, title, and date.

ToDoListController Class: This class is a control class of the ToDoListItem class and can add, and delete, update a ToDoListItem. It can also get all to do list items.

WaitList Class: This class creates a list of all students who are in the waitlist of Erasmus applications. It also has the property of Erasmus point.

WaitListController Class: This class controls the WaitList Class and either can accept students into the Erasmus program or an outgoing student back to the WaitList if the student cancels their program.

Syllabus Class: This class creates the syllabus object which has the properties a URL link, course name, and the course id.

Application Class: This class is the parent class of the other application children's classes. The class consists of common properties such as the application number, country, university name, and the completed status.

ApplicationForIncomingStudents Class: This class is a container for the course requests of incoming users, and it extends the Application Class. This class aggregates CourseRequest Class.

ApplicationControllerForIncomingStudents: This class is a controller of the ApplicationForIncomingStudents Class. This class has the properties of graduate level or easiness level status for their courses. The incoming student can create course request, update it, and delete it and send it to the Department Secretary to have them register their courses. The courses can be accepted or refused based on the status of the properties of the courses listed in the class.

ApplicationForOutgoingStudents Class: This class consists of a language proficiency exam result, transcript, pre-approval form, learning agreement, the registration link of the host university, the project report of any project courses taken in the host university, a course request, and the status of the mobility. This class has composition with LanguageProficiency, Transcript, CourseTransferForm, and LanguageAgreement classes.

ApplicationControllerForOutgoingStudents: This class is the control class of the ApplicationForOutgoingStudents Class. The class has methods to add, update, submit and delete language proficiency tests, preapproval forms, learning agreement, and course transfer form. It can also upload transcript and check ECTS of courses. It also can add things to the ToDoList, sign documents, check the application status, cancel placement, notify cancellation, add a course to the accepted courses list, and check if a course has been accepted before.

ApplicationControllerCoordinator: This class is a controller for the Coordinator Class. The class includes methods such as evaluating language proficiency, pre-approval forms, transcript, learning agreement, course transfer form, evaluate ECTS, and project reports. It can also close an application and send course requests to instructors.

Course Class: This class creates a course, and it has the properties of a Syllabus, course name, course id, and ECTS value. This class aggregates Syllabus Class.

CourseRequest Class: This class creates a course request and has the properties of previously accepted status, ECTS, course name, course id, syllabus, and course. This class aggregates the Course Class and the Courses Class.

CourseRequestController Class: This class is a controller of the CourseRequest Class. It can create, update, and delete a course request. It can also add description, identify a course, check if course is approved before, remove from accepted courses,

add a description to the course, state if it is a mandatory or an elective course, and state if it is an arts or social science elective.

Transcript Class: This class consists of the GPA, the courses as a list, and the date. This class aggregates the Course Class.

LanguageProficiency Class: This class includes the properties of language, exam name, score, and the passed status of the result.

PreApprovalFormItem Class: This class contains a list of course requests. This class aggregates the CourseRequest Class.

LearningAgreement Class: This class includes the property of a PreApprovalFormItem.

CourseTransferForm Class: This class has a transcript and the pre-approval form of the outgoing student. This class aggregates the Transcript Class.

3.3. Dynamic Models Diagram

3.3.1. State Machine Diagrams

3.3.1.1. Application for Outgoing Students

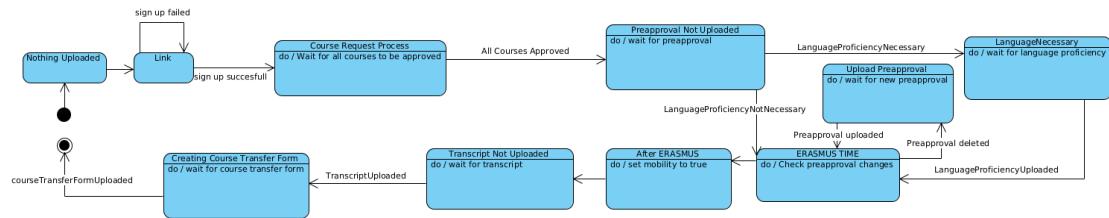


Figure 7: Application for Outgoing Students State Machine Diagram

This state machine diagram is chosen because it contains extensive, detailed, and one of the most crucial parts of the program. It models all states of the outgoing students' application process before and after mobility. It starts with registering via a link and ends with the course transfer form being uploaded.

3.3.1.2. Course Request Process

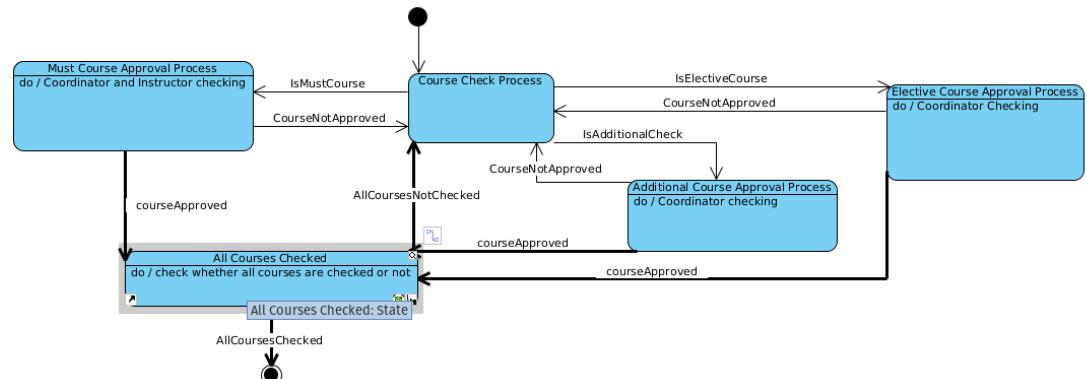


Figure 8: Course Request Process State Machine Diagram

This state machine diagram is a sub-state machine diagram of the previous state machine diagram. It models the Course Request Process state of the previous diagram, starting with identifying the course type and ending with all courses being checked.

3.3.2. Activity Diagrams

3.3.2.1. Course Transfer Form Evaluation

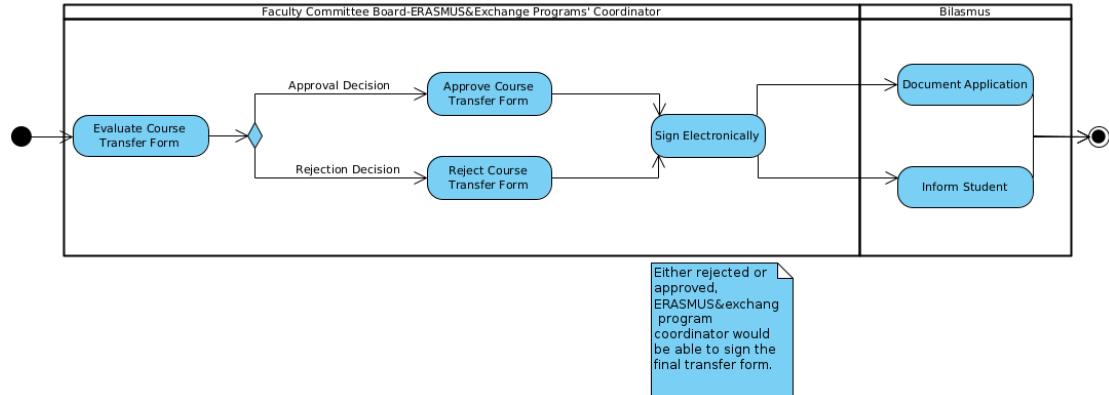


Figure 9: Course Transfer Form Evaluation Activity Diagram

This diagram represents how the course transfer form evaluation should proceed. We implemented this diagram to describe the process necessary for the application to end. It also gives an idea of what the Bilasmus will and will not do.

3.3.2.2. Course Identification

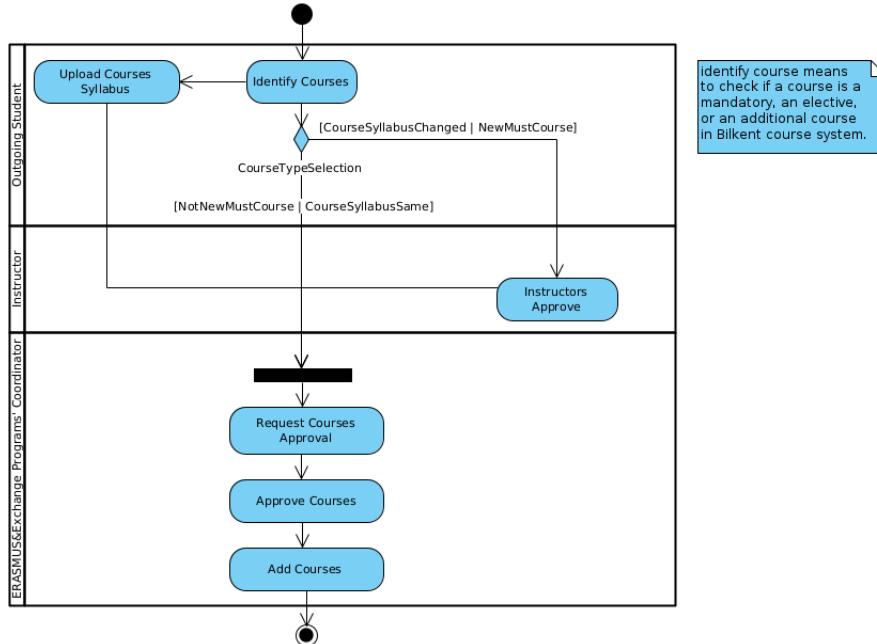


Figure 10: Course Identification Activity Diagram

This diagram represents how the courses are approved, and additionally, it shows how different courses are treated in the process. The primary motivation behind implementing this diagram was to visualize each step in the course approval process.

3.3.3. Sequence Diagrams

3.3.3.1. Course Requesting

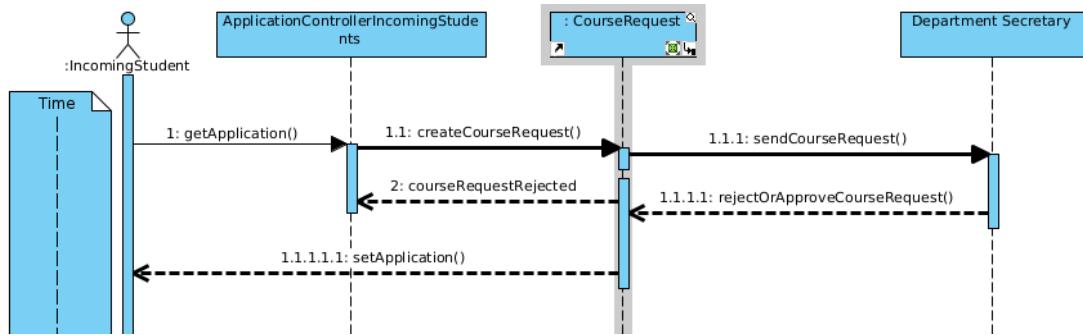


Figure 11: Course Requesting Sequence Diagram

This diagram represents requesting course approval will work in low-level design.

3.4. User Interface

The UI Flow is shown below.

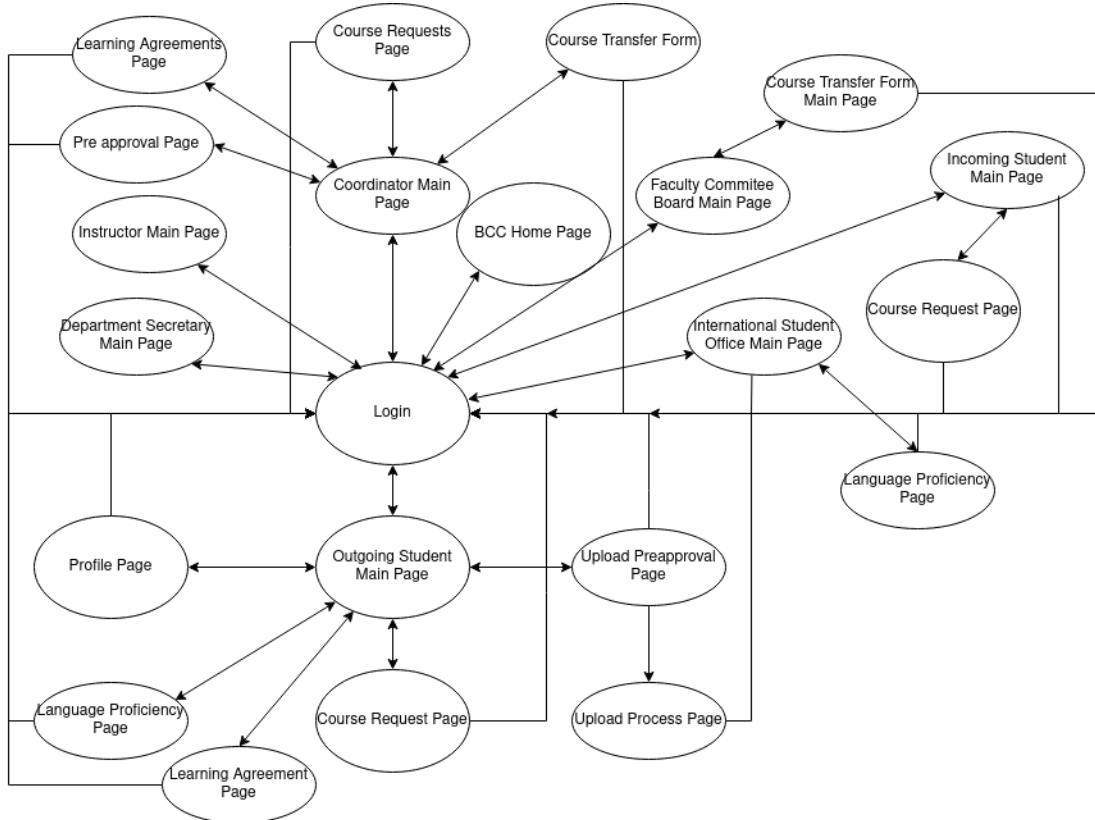


Figure 12: The UI Flow

Login Page

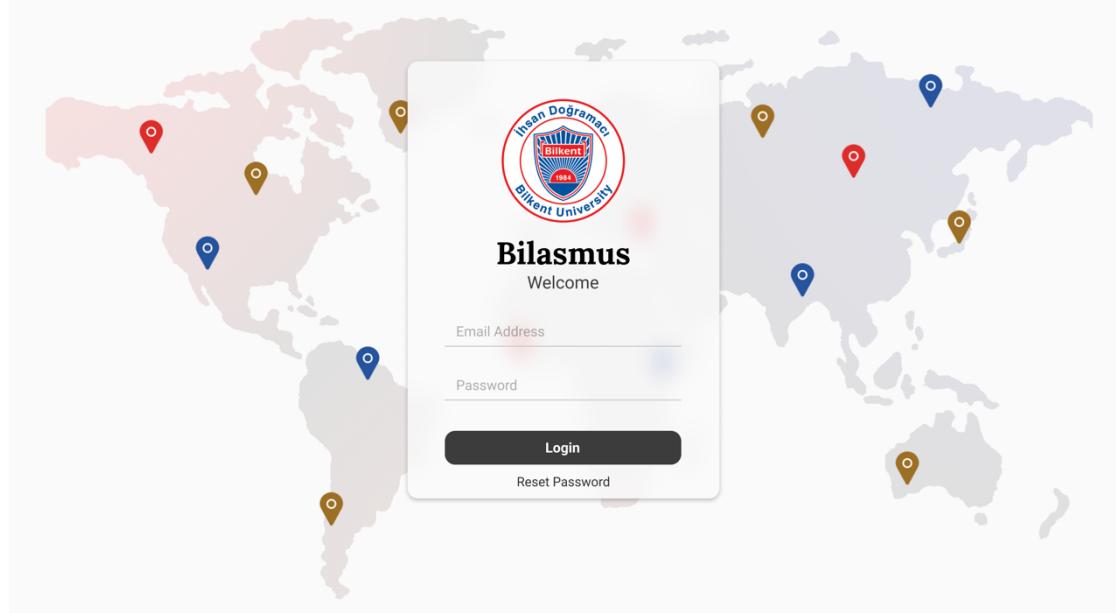


Figure 13: Login page

Login Failed Page

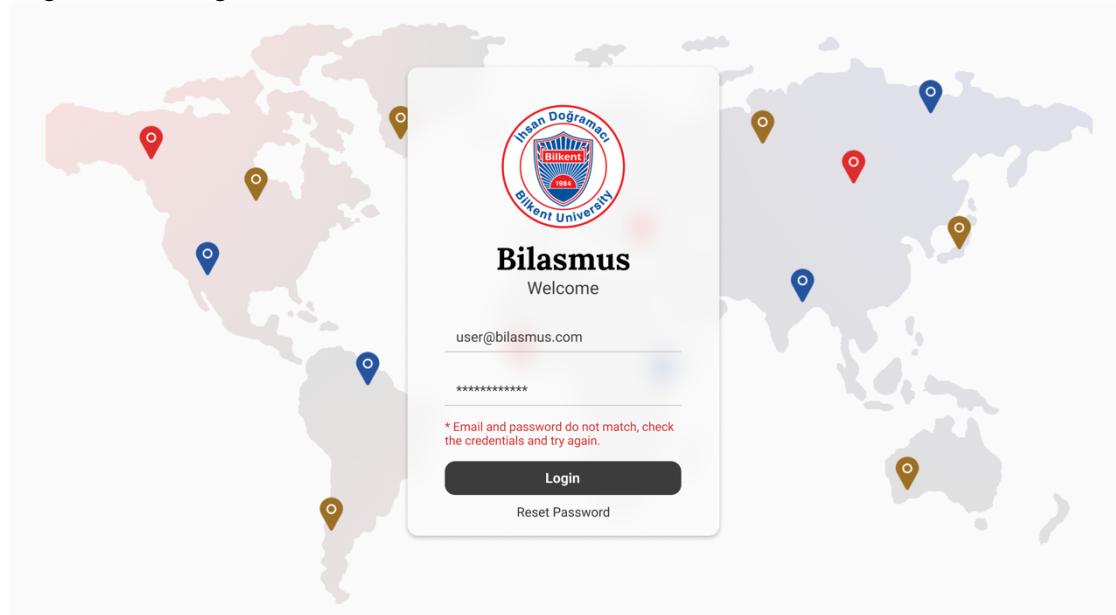


Figure 14: Login failed page

Reset Password Page



Figure 15: Login failed page

Profile Page

A screenshot of a profile page template. At the top, there is a red header bar with a logo on the left, followed by navigation links: Home, Course Request, Pre-Approval Form, Learning Agreement, Language Proficiency, and Help. On the right side of the header are links for Log out, a message icon, a notification bell, and a user profile icon. The main content area has a light gray background. In the center, there is a profile section with a large photo of a person holding a book. To the left of the photo, the word "Profile" is written in bold. Below the photo is a box containing the following information:

Name: Student1
BilkentID: 22xxxxxx
Phone Number: +90 xxx xxx xx xx
Citizen ID: xxxxxxxxxxxx
Email: student1.ug.bilkent.edu.tr

At the bottom of the page, there are two buttons in a light gray box: "Contact BCC to update info" and "Contact BCC" (in a darker button). To the right, there are two more buttons: "Add signature" and "Upload Signature".

Figure 16: Profile page template of all users

Notifications Page

The screenshot shows the Notifications Page template. At the top, there is a red header bar with a logo, navigation links (Home, Course Request, Preapproval Form, Learning Agreement, Language Proficiency, Help), and user account options (Log out, notifications, profile). Below the header, there is a section titled "Learning Agreement" with a placeholder for uploading a document. A large button labeled "Choose File" is present. To the right, a sidebar titled "Notifications" displays three items: a green checkmark indicating approval of a pre-approval form, a purple reminder for re-uploading the learning agreement, and a red X indicating rejection of a learning agreement.

Figure 17: Notifications page template of all users

Help

The screenshot shows the Help page template. At the top, there is a red header bar with a logo, navigation links (Home, Course Request, Preapproval Form, Learning Agreement, Language Proficiency, Help), and user account options (Log out, notifications, profile). Below the header, the word "Welcome" is displayed. A search bar with the placeholder "How can we help you?" and a text input field with "Type here..." are shown. In the bottom left corner, there is a section titled "Application Status".

Figure 18: Help page template of all users

Messages

The screenshot shows a 'Messages' interface. At the top right, there are icons for 'Log out', a speech bubble, a bell, and a user profile. Below the header, a 'Welcome' message is displayed. A 'Application Status' button is highlighted with a blue border. To the right, a 'Messages' panel lists several messages from different users:

- person1: The form is still needed
- person2: I will let you know.
- person3: below you can find add
- Me: Thanks hocam.
- Me: Where can I find the inform
- Me: I will send them as soon as..
- person4: Contact Mr. person5 a..

A green arrow points upwards towards the 'Log out' button. A text input field at the bottom right of the messages panel contains the placeholder 'Type here..'. There is also a small 'Person5' icon above the input field.

Figure 19: Message template of all users

Detailed Info

The screenshot shows a 'Detailed Info' page for a user. At the top right, there are icons for 'Log out', a speech bubble, a bell, and a user profile. The main content area includes:

- Profile:** Student Name 1, Department: Computer Engineering, Phone Number: +90 xxx xxx xx xx, Bilkent ID: 22xxxxxx, Email: name.surname@ug.bilkent.edu.tr
- Signed Documents:** A list of three documents: Document 1 (dd/mm/yyyy), Document 2 (dd/mm/yyyy), and Document 3 (dd/mm/yyyy).
- All Uploaded Documents:** A list of seven documents: Document 1 (dd/mm/yyyy), Document 2 (dd/mm/yyyy), Document 3 (dd/mm/yyyy), Document 4 (dd/mm/yyyy), Document 5 (dd/mm/yyyy), Document 6 (dd/mm/yyyy) which is highlighted with a blue border, and Document 7 (dd/mm/yyyy).
- Course List:** A section stating 'Course List is Empty'.

Figure 20: Detailed info page of all users

Main Page- Outgoing Student

Figure 21: Main page of outgoing student users

Course Request Page- Outgoing Student

Figure 22: Course Request page of outgoing student users

Language Proficiency Page- Outgoing Student

The screenshot shows the Language Proficiency page for outgoing student users. At the top, there is a red navigation bar with links for Home, Course Request, Preapproval Form, Learning Agreement, Language Proficiency, and Help. On the far right of the bar are Log out, a message icon, a bell icon, and a user profile picture. Below the navigation bar, the page title "Welcome" is displayed. To the right of the title is a file upload interface showing a progress bar labeled "Loading..." and a green "Upload" button. A blue arrow points upwards from the "Upload" button towards the "Language Proficiency" link in the navigation bar. To the right of the upload area is a "To-Do List" section with two items: "Upload preapproval form" and "Upload learning agreement", each with an unchecked checkbox. In the center of the page, there is a message box with a blue border containing the text "Successfully uploaded" and an "Ok" button.

Figure 23: Language Proficiency page of outgoing student users

Pre-Approval Form Page- Outgoing Student

The screenshot shows the Pre-Approval Form page for outgoing student users. At the top, there is a red navigation bar with links for Home, Course Request, Pre-Approval Form, Learning Agreement, Language Proficiency, and Help. On the far right of the bar are Log out, a message icon, a bell icon, and a user profile picture. Below the navigation bar, the page title "Pre-Approval Form" is displayed. A sub-header text says "Upload your Pre-Approval Form document by either dragging and dropping your file to the box below, or clicking on the Choose File button." Below this text is a large rectangular file upload area with a central icon of a document with an upward arrow. Below the icon is the text "Drop your file here...". Underneath the upload area is a dark grey "Choose File" button. Further down the page, there is a section titled "Previously Uploaded Files" with a table. The table has one row showing a file named "Name Surname Preapproval Form.pdf" with a download icon, a timestamp "Uploaded: 1 November 2022, 13:35", and a status "Status: Approved". To the right of the file name are "Download" and "Delete" buttons.

Figure 24: Pre-Approval Form page of outgoing student users

Pre-Approval Form Page- Upload Process- Outgoing Student

The screenshot shows the 'Pre-Approval Form' section of a web application. At the top, there is a red header bar with the university logo and navigation links: Home, Course Request, Pre-Approval Form, Learning Agreement, Language Proficiency, and Help. On the right side of the header are links for Log out, a message icon, a bell icon, and a user profile icon. Below the header, the main content area has a title 'Pre-Approval Form' and a sub-instruction: 'Upload your Pre-Approval Form document by either dragging and dropping your file to the box below, or clicking on the Choose File button.' A large, light-gray rectangular input field is provided for file uploads, containing a small document icon and the text 'Name Surname Preapproval Form.pdf'. To the right of this field, it says 'Uploaded: 1 November 2022, 15:01' and 'Status: Pending Upload'. Below the input field is a dark gray button labeled 'Upload'. Further down, under the heading 'Previously Uploaded Files', there is a list of a single file: 'Name Surname Preapproval Form.pdf', uploaded on '1 November 2022, 13:35', with status 'Approved'. This list includes a 'Download' button (dark gray) and a 'Delete' button (red).

Figure 25: Pre-Approval Form page- file upload process- of outgoing student users

Learning Agreement Page- Outgoing Student

The screenshot shows the 'Learning Agreement' section of a web application. It features a red header bar with the university logo and navigation links: Home, Course Request, Pre-Approval Form, Learning Agreement, Language Proficiency, and Help. On the right side of the header are links for Log out, a message icon, a bell icon, and a user profile icon. The main content area has a title 'Learning Agreement' and a sub-instruction: 'Upload your Learning Agreement document by either dragging and dropping your file to the box below, or clicking on the Choose File button.' A large, light-gray rectangular input field is provided for file uploads, featuring a central icon of a laptop with an upward arrow and the placeholder text 'Drop your file here...'. Below this field is a dark gray button labeled 'Choose File'. Underneath the input field, there is a section titled 'Previously Uploaded Files' showing a single file entry: 'Name Surname Learning Agreement.pdf', uploaded on '1 November 2022, 13:00', with status 'Approved'. This entry includes a 'Download' button (dark gray) and a 'Delete' button (red).

Figure 26: Learning Agreement page of outgoing student users

Main Page-Incoming Student

The screenshot shows the main dashboard for incoming students. At the top, there is a navigation bar with links for Home, Course Request, and Help, along with user profile and log-out options. The main content area is divided into two sections: 'Course List' on the left and 'To-Do List' on the right.

Course List: This section displays a list of courses offered by İhsan Doğramacı Bilkent University. Each course entry includes the course name, university logo, and two buttons: 'Details' and 'Syllabus'.

- History of Turkey
- Digital Design
- Fundamentals of Computer Science I
- General Physics I
- Cultures, Civilizations and Ideas I
- Orientation for Incoming Students

To-Do List: This section is currently empty, displaying the message "Your To-Do List is Empty".

Figure 27: Main page of incoming student users

Course Request- Incoming Student

The screenshot shows the 'Course Request' page for incoming students. The top navigation bar is identical to Figure 27. The main content area is titled 'Course Request' and contains a form for adding courses to the request list.

Add a course to the course request list: This section includes input fields for 'Course ID' and 'Course Name', and a large green 'Add' button.

Added courses: This section lists five courses that have been added to the request list, each with an 'Update' and a 'Delete' button.

- CS319- Object-Oriented Software Engineering
- LNGI71- Turkish I
- FAT71- Introduction to Art Design, and Culture I
- CS465- Computer Graphics
- CS426- Parallel Computing

A green 'Submit' button is located at the bottom right of the page.

Figure 28: Learning Agreement page of outgoing student users

Main Page- Coordinator

Figure 29: Main page of coordinator users

Course Request Page- Coordinator

Figure 30: Course Request page of coordinator users

Course Request Page-View Check Request- Coordinator

Student 1		Course Type	Evaluated by Instructor	Course Link and Syllabus	Final Decision
Course Name		Mandatory	Approved	Link Syllabus	Approve Reject
1. CE319- Object-Oriented Software Engineering		Elective	-	Link Syllabus	Approve Reject
2. LNG170- German I		Elective	-	Link Syllabus	Approve Reject
3. FA71- Introduction to Art Design, and Culture I		Mandatory	-	Link Syllabus	Approve Reject
4. CE465- Computer Graphics		Mandatory	Rejected	Link Syllabus	Approve Reject
5. CE426- Parallel Computing		Mandatory	-	Link Syllabus	Approve Reject

Student 2		Course Type	Evaluated by Instructor	Course Link and Syllabus	Final Decision
Course Name		Mandatory	Approved	Link Syllabus	Approve Reject
1. ART203- Renaissance Art		Elective	-	Link Syllabus	Approve Reject
2. LANG120- German II		Elective	-	Link Syllabus	Approve Reject
3. MAT360- Linear Algebra		Mandatory	Rejected	Link Syllabus	Approve Reject
4. IEI01- Introduction to Industrial Engineering		Mandatory	-	Link Syllabus	Approve Reject
5. PSYC320- Psychology of Consumerism		Mandatory	-	Link Syllabus	Approve Reject

Student 3		Course Type	Evaluated by Instructor	Course Link and Syllabus	Final Decision
Course Name		Mandatory	Pending	Link Syllabus	Approve Reject
1. HIST150- History of Germany		Elective	-	Link Syllabus	Approve Reject
2. LNG140- Urdu I		Elective	-	Link Syllabus	Approve Reject
3. FA101- Introduction to Fashion		Mandatory	Rejected	Link Syllabus	Approve Reject
4. MAN220- Business Law		Mandatory	-	Link Syllabus	Approve Reject
5. ECON410- Microeconomic Theory		Mandatory	-	Link Syllabus	Approve Reject

Figure 31: Course Request page of coordinator users

Pre-Approval Page- Coordinator

 Name Surname Preapproval Form.pdf	Student: Name Surname	Uploaded: 1 November 2022, 13:35	View	Approve	Reject
 Name Surname Preapproval Form.pdf	Student: Name Surname	Uploaded: 2 November 2022, 14:25	View	Approve	Reject
 Name Surname Preapproval Form.pdf	Student: Name Surname	Uploaded: 3 November 2022, 15:15	View	Approve	Reject
 Name Surname Preapproval Form.pdf	Student: Name Surname	Uploaded: 4 November 2022, 16:05	View	Approve	Reject
 Name Surname Preapproval Form.pdf	Student: Name Surname	Uploaded: 5 November 2022, 17:55	View	Approve	Reject

Figure 32: Pre-Approval Form page of coordinator users

Learning Agreement Page- Coordinator

The screenshot shows a list of five learning agreements for review:

- Name Surname Learning Agreement.pdf
Student: Name Surname
Uploaded: 1 November 2022, 13:35
View Approve Reject
- Name Surname Learning Agreement.pdf
Student: Name Surname
Uploaded: 2 November 2022, 14:25
View Approve Reject
- Name Surname Learning Agreement.pdf
Student: Name Surname
Uploaded: 3 November 2022, 15:15
View Approve Reject
- Name Surname Learning Agreement.pdf
Student: Name Surname
Uploaded: 4 November 2022, 16:05
View Approve Reject
- Name Surname Learning Agreement.pdf
Student: Name Surname
Uploaded: 5 November 2022, 17:55
View Approve Reject

Figure 33: Learning Agreement page of coordinator users

Learning Agreement Page-No Requests- Coordinator

The screenshot shows a message indicating there are no pending requests:

No Pending Approval

There is not any Learning Agreement that needs your attention. New approval tasks will appear here once they are assigned to you.

Figure 34: Learning Agreement page-no requests- of coordinator users

Course Transfer Forms Page-Coordinator

The screenshot shows a list of 11 students (Person1 to Person11) with their names and emails. Each student entry includes a small profile picture, the name, the email address, and a 'Download Forms' button. Above the list is a search bar with the placeholder 'Type here...' and a green 'search' button. A red arrow points to the search bar.

#		Type here...	search	Download Forms
A	Person1	Person1@gmail.com		
B	Person2	Person2@gmail.com		
C	Person3	Person3@gmail.com		
D	Person4	Person4@gmail.com		
E	Person5	Person5@gmail.com		
F	Person6	Person6@gmail.com		
G	Person7	Person7@gmail.com		
H	Person8	Person8@gmail.com		
I	Person9	Person9@gmail.com		
J	Person10	Person10@gmail.com		
K	Person11	Person11@gmail.com		

Figure 35: Course Transfer Forms page of coordinator users

Main Page- International Student Office

The screenshot shows the main interface for international student office users. It includes a 'Send Transcript' section with a search bar and a 'Send Transcript to Coordinator' button. Below this are two 'Add Todo List' sections: one for 'Coordinator's todo List' and one for 'Student's todo List', both with fields for Title, Due Date, and an 'Add Reminder' toggle. To the right is a 'To-Do List' sidebar with checkboxes for 'Check Language Proficiency Results' and 'Send Registration Link'. At the bottom is a 'Registration Link' section with a 'Student ID' input field and a 'Send Registration Link' button.

Send Transcript

Send Transcript to Coordinator

Add Coordinator's todo List

Title:

Due Date:

Add Reminder Add

Add Student's todo List

Title:

Due Date:

Add Reminder Add

Registration Link

Send Registration Link to ALL Students

Send Registration Link

To-Do List

- Check Language Proficiency Results
- Send Registration Link

Figure 36: Main page of international student office users

Language Proficiency Page- International Student Office

Approved Students

- Student1: exam name: IELTS, language: English, score: 7.5, passed: Yes No
- Student2: exam name: TOEFL, IBT, language: English, score: 115, passed: Yes No
- Student3: exam name: TEF, language: French, score: 72, passed: Yes No
- Student4: exam name: DSD, language: German, score: C1, passed: Yes No
- Student5: exam name: IELTS, language: English, score: 6.5, passed: Yes No

Rejected Students

- Student1: (No data shown)
- Student2: (No data shown)
- Student3: (No data shown)
- Student4: (No data shown)
- Student5: (No data shown)
- Student6: (No data shown)
- Student7: (No data shown)
- Student8: (No data shown)
- Student9: (No data shown)
- Student10: (No data shown)
- Student11: (No data shown)
- Student12: (No data shown)
- Student13: (No data shown)

Figure 37: Language Proficiency page of international student office users

Main Page- Department Secretary

Welcome

Course Requests

Student Name	Course Name	Status
Student Name 1	Course Name 1	✓ ✗
Student Name 1	Course Name 2	✓ ✗
Student Name 2	Course Name 3	✓ ✗
Student Name 2	Course Name 4	✓ ✗
Student Name 2	Course Name 5	✓ ✗

To-Do List

Your To-Do List is Empty

Figure 38: Main page of department secretary users

Course Requests Page- Department Secretary

Course Requests

Student 1

- 1. CS319- Object-Oriented Software Engineering
- 2. LNC171- Turkish I
- 3. FAI71- Introduction to Art Design, and Culture I
- 4. CS465- Computer Graphics
- 5. CS426- Parallel Computing

Student 2

- 1. CS315- Programming Languages
- 2. CS342- Operating Systems
- 3. FAI71- Introduction to Art Design, and Culture I
- 4. CS475- Data Privacy
- 5. CS476- Automata Theory and Formal Languages

Student 3

- 1. SOC101- Introduction to Sociology
- 2. LNC172- Turkish II
- 3. PSYC100- Introduction to Psychology
- 4. ECON101- Introduction to Economics
- 5. COMD341- Media and Society

Figure 39: Course Request page of department secretary users

Main Page-Faculty Committee Board

Welcome

Preapproval List

UNIVERSITY NAME 1 Student Name 1	View
UNIVERSITY NAME 2 Student Name 2	View
UNIVERSITY NAME 3 Student Name 3	View
UNIVERSITY NAME 4 Student Name 4	View
UNIVERSITY NAME 5 Student Name 5	View

Course Transfer Forms

UNIVERSITY NAME 1 Student Name 1	View
UNIVERSITY NAME 2 Student Name 2	View
UNIVERSITY NAME 3 Student Name 3	View
UNIVERSITY NAME 4 Student Name 4	View
UNIVERSITY NAME 5 Student Name 5	View
UNIVERSITY NAME 6 Student Name 6	View

To-Do List

Your To-Do List is Empty

List of Signatures

List of Signatures

Figure 40: Main page of faculty committee board users

Course Transfer Form Page- Faculty Committee Board

The screenshot shows a web application interface for managing course transfer forms. At the top, there is a blue header bar with a logo, 'Home', 'Help', and 'Log Out' links, and a user profile icon. Below the header, the title 'Faculty Committee Board' is displayed. The main content area is titled 'Course Transfer Form'. It contains a list of six student entries, each with a text input field labeled 'Student' followed by a number (1 through 6), a 'Check Form' button, and two action buttons: 'Approve' (green) and 'Reject' (red). To the right of this list is a sidebar titled 'Search Course Transfer Form' with a search input field and a 'Search' button.

Figure 41: Course Transfer Form page of faculty committee board users

Pre-Approval Forms Page- Faculty Committee Board

The screenshot shows a web application interface for managing pre-approval forms. At the top, there is a blue header bar with a logo, 'Home', 'Help', 'Log out' link, and a user profile icon. Below the header, the title 'Pre-Approval Forms' is displayed. The main content area lists five pre-approval forms, each represented by a card. Each card contains a document icon, the file name 'Name Surname Preapproval Form.pdf', the student name 'Surname', and the upload date ('Uploaded: 1 November 2022, 13:35'). To the right of each card are three buttons: 'View' (grey), 'Approve' (green), and 'Reject' (red).

Figure 42: Pre-Approval Forms page of faculty committee board users

Main Page- Instructor

The screenshot shows the main page for instructor users. At the top, there is a header with a logo, 'Home', 'Help', 'Log out', and a user profile icon. Below the header, the title 'Course Requests' is displayed. The page is divided into two sections: 'Student 1' and 'Student 2'. Each section contains a table with course details, course type, and action buttons for 'Link', 'Syllabus', 'Approve', and 'Reject'. In 'Student 1', three courses are listed: CE319- Object-Oriented Software Engineering, CE465- Computer Graphics, and CE426- Parallel Computing, all marked as Mandatory. In 'Student 2', one course, CS359-Operating Systems, is listed as Mandatory.

Figure 43: Main page of instructor users

Main Page- BCC

The screenshot shows the main page for BCC users. At the top, there is a header with a logo, 'Home', 'Help', 'Log Out', and a user profile icon. A welcome message 'Welcome, BCC!' is displayed. The page features several interactive components:

- Add User:** A form with fields for 'Name', 'Bilkent ID', and 'E-mail', and a 'Staff' or 'Student' selection switch. An 'Add User' button is present.
- To-Do List:** A sidebar with checkboxes for 'Add new incoming students' and 'Delete students who cancelled'.
- Delete User:** A form with a 'Bilkent ID' field and a red 'Delete User' button.
- Update User:** A form with fields for 'Bilkent ID', 'E-mail', and 'New E-mail', and an 'Update User' button.

Figure 44: Main page of BCC