Bilkent University

Department of Computer Engineering

CS319 Term Project

ErasXchange

Group 3B

Analysis Report

Group Members:

Ceren Akyar - 22003158

Deniz Mert Dilaverler - 22003530

Hasan Yarkın Karakurt - 22003072

Ali Emir Güzey - 22003186

Dağhan Ünal - 22002182

Instructor: Eray Tüzün

Teaching Assistant(s): Emre Sülün, Muhammed Umair Ahmed, Mert Kara, İdil Hanhan

Date: November 6, 2022

Contents

Analysis Report

1 Introduction

Tracking Erasmus/Exchange applications have always been a hassle in Bilkent University. The application process includes too many actors and too many documents for each part to handle manually. The lack of a portal to track one's application creates a great amount of ambiguity to the Erasmus/Exchange application process.

Currently, the Erasmus/Exchange applications are followed by mail chains between students and coordinators. This is an ineffective and time-consuming way to handle the many steps of an Erasmus/Exchange application. There is also the problem of forms with the current system. There are many forms in the Erasmus/Exchange process. There are preapproval forms, course approval forms, and course transfer forms. All of these forms have to be filled out and scanned and sent to the next person. This causes a tremendous amount of paper waste.

ErasXchange is an Erasmus and Exchange application tracker for Bilkent University Students, Erasmus/Exchange coordinators, and board members. The application will allow students to follow their application process and see what they need to do before going to the host school. ErasXchange will also be useful for coordinators and board members to track student applications, handle the paperwork digitally and automatically divide the workload of schools. ErasXchange will provide a platform for both students and coordinators to get the pre-Erasmus/pre-Exchange met, to handle the required forms, and changes in applications.

2 Overview

ErasXchange is a web application for performing outgoing Erasmus and exchange students' placements and related paperwork. The workflow of ErasXchange starts with an Excel table file obtained from the International Students Office, which contains the student information gathered from the applications done through Bilkent University

Erasmus+ Application System [1]. ErasXchange places the students according to their Erasmus scores and preferred schools. After that, it provides a system for placed students and exchange coordinators to use to complete the paperwork needed for the exchange procedures.

2.1 Overview of the System Prior to the Placements

Before the information from Bilkent University Erasmus+ Application System arrived at the system, ErasXchange acts as a general information website for the Erasmus program at Bilkent University. The information is currently spread across several web pages, which complicates gathering information for students [3, 4, 5, 6]. Tables for available universities, requirements, previously accepted courses for each university, and frequently asked questions are combined into a single web application where students can find their desired information easily.

2.2 Algorithm for the Placements

Erasmus placements at Bilkent University are done the same way as ÖSYM [2]. Each student indicates up to 5 preferred schools, the system places students according to their preferences and available quotas. When all quotas are full, students that are not placed yet are moved to the waiting bin to be contacted in case a selected student cancels their application.

2.3 Overview of the System After the Placements

Students are informed through email notifications about whether they are placed. Actions related to post-placement steps, such as filling in pre-approval forms, registering host universities' courses to the system, or canceling the application, are visible to the placed students. Coordinators can view the list of placed students, empty quotas if any, and the students in the waiting bin. They may make announcements through the system or send messages to a student privately. They review courses registered by students and approve or reject their transfer. Also, they evaluate the pre-approval forms through the system.

2.4 Overview of the System After the Erasmus/Exchange Programme

After the students' Erasmus/Exchange program ends, courses taken by students need to be registered to their Bilkent University Transcript. This is called the Course Transfer

Process. The International Students Office receives students' transcripts from the host university and sends these documents to the coordinator. Later coordinator prepares the Course Transfer From according to the transcript and Pre-Approval Form of the student. Then the coordinator sends this form to the Faculty Board Members in order to be signed.

3 Functional Requirements

3.1 Interactions of Student Role

- Students can view schools and the assigned coordinator to those schools.
- Students can view previously approved courses which would help to fill preapproval form.
- If a student is placed in a school, the student can fill out a preapproval form. While filling preapproval form, students can select a project course and upload their project report after attending the course. Students can also choose to add a new course. In that case, the student uploads the syllabus link of the course, course name, equivalent course name from Bilkent, and if it is a must-course or an elective.
- Students can cancel their Erasmus application before placements are announced.
- Students can cancel their Erasmus application after placements are announced. In that case, the student chooses to decide between canceling their application altogether or being placed at the top of the waiting bin.

3.2 Interactions of International Students Office Role

- International Students Office (ISO) can upload excel files that contain Erasmus points (or exchange points) by choosing the program type, and if it is Erasmus, by choosing the department.
- ISO can send acceptance links to the related students when the host organization sends a link to ISO.
- ISO can send an official transcript to coordinators after the host organization sends the transcript to ISO. ISO sends the transcript by uploading it to the system with related student information.

3.3 Interactions of Board Member Role:

- A board member can sign documents which is a course transfer form, after a coordinator sends the form to the board member. A board member can sign it with an e-signature or by printing and scanning back the document (it is flexible due to security reasons).
- A board member can also sign the transcript sent by ISO by e-signature or by printing and scanning back and uploading the document.

3.4 Interactions of Coordinator Role

- Coordinators can fill out a course transfer form by viewing the student's preapproval form and filling out information on the template course approval form.
- Coordinator can view events that are pending preapproval forms, pending course applications, etc. Coordinator can filter event types (only preapproval forms, only waiting list placements, etc.)
- Coordinator can enter the information of assigned universities by selecting an organization, filling in organization information (applicable semester, quota, etc.), and submitting information.
- Coordinator can start placements. After that, the coordinator's assigned department placements will be done. Hence, this is a one-time event for all departments. If there are multiple coordinators in a department, after a coordinator starts the placements, other coordinators cannot start the process again. (In exchange, this is generally a one-time event.) If a student has incorrect application information, coordinator will choose to notify the student (or multiple students) through the system.

3.5 Interactions of Instructor Role

- The related instructor can view the course approval forms of students. After viewing, they can accept or deny the approval form for a new course. If instructor denies it, the instructor can write feedback or a reason for denial and submits the form again for the student's view.

4 Non-functional Requirements

4.1 Maintainability

ErasXchange will be implemented with maintainability in mind. Maintainability will be crucial for supporting ErasXchange throughout semesters of use. Hence during development, the codebase will be well-documented to ensure easier debugging and testing.

4.2 Reliability

ErasXchange will work with important user and education data. The integrity of the said data will be important. Hence the database will be regularly backed up in order to prevent data loss from data corruption and physical factors.

4.3 Intuitiveness

The ErasXchange will be used by students from all departments with significantly differing technical knowledge. Hence, the user experience of the ErasXchange will be well thought out. The UX will be tested for misuse and designed as intuitively as possible.

4.4 Extendability

Erasmus/Exchange application procedure can be changed in the future. New features may need to be added or existing features may be needed to be removed. Hence extendability of ErasXchange is a matter of great importance. That is why ErasXchange will be developed in accordance with Object Oriented design patterns to make sure that new features can be added easily with minimal debugging and testing required.

4.5 Load Handling

Unlike other web applications, ErasXchange will have more concentrated traffic in comparison to spread-out traffic. This is because Erasmus/Exchange applications will have specific times for events that concern every applicant like application result announcements. During those kinds of announcements, the ErasXchange platform will be flooded with requests and the servers need to either handle such loads or limit user access during peak usage times.

4.6 Low Response Time

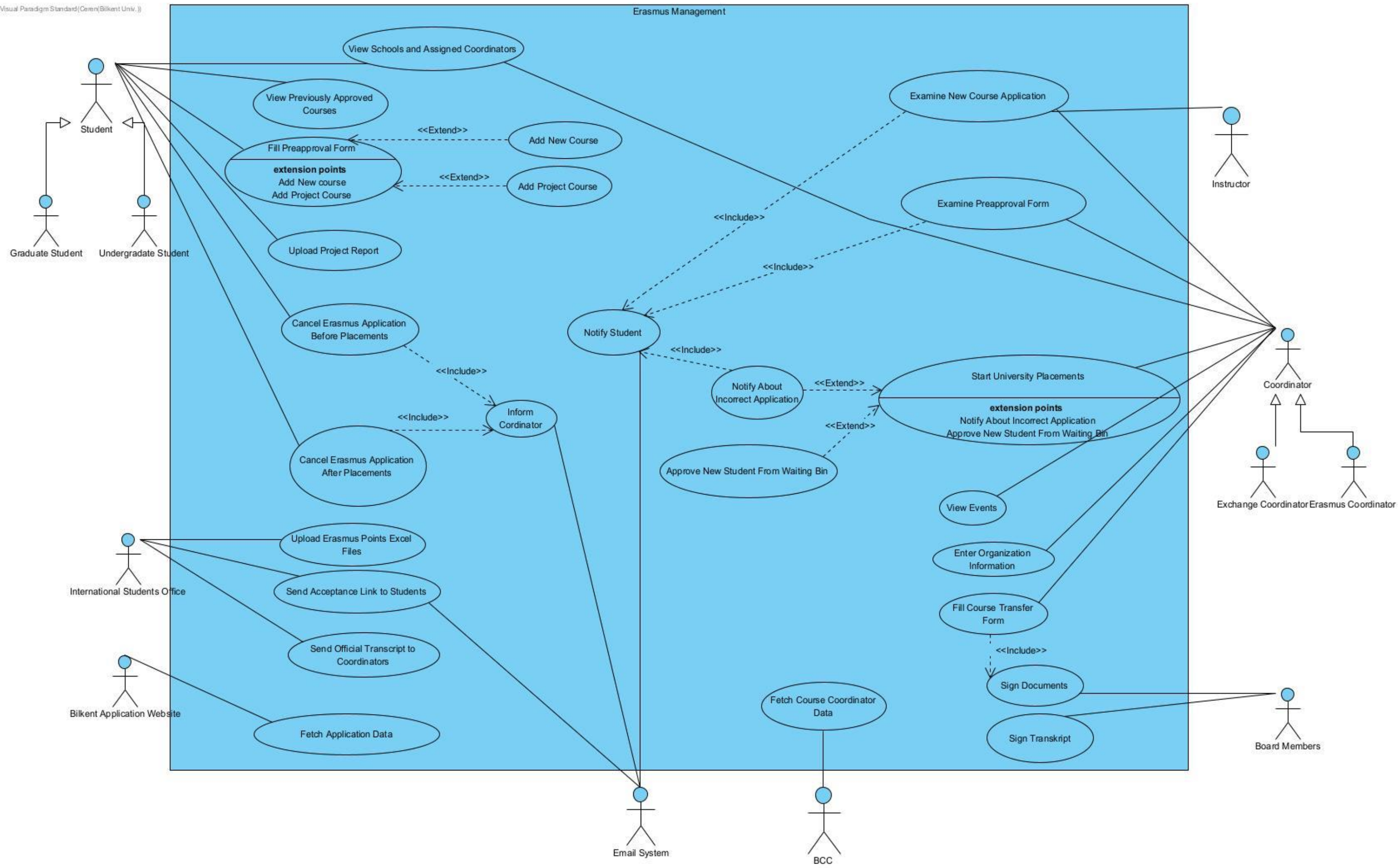
One of the benefits of ErasXchange to the existing application system is that the proposed system allows users to send forms and do changes in their applications faster. However, this benefit is lost if the service requires long and frustrating request times. Hence, Eraschange/Buddy will try to meet sub-second request times and will provide visual effects on the front end to make the user feel like the wait times are less than they are.

4.7 Usage of Existing Systems

The proposed project will be required to interact with existing systems for data requests. Hence, access from the existing Erasmus/Exchange application systems will be requested. Also access from BCC will also be requested for requesting course, coordinator, and/or student data.

5 System Models

5.1 Use-Case Model


Visual Paradigm Standard(Ceren(Bilkent Univ.))
Erasmus Management
View Schools and Assigned Coordinators
View Previously Approved Courses
Fill Preapproval Form
extension points
Add New course
Add Project Course
<<Extend>>
Add New Course
<<Extend>>
Add Project Course
Upload Project Report
Cancel Erasmus Application Before Placements
<<Include>>
Inform Cordinator
<<Include>>
Cancel Erasmus Application After Placements
Upload Erasmus Points Excel Files
Send Acceptance Link to Students
Send Official Transcript to Coordinators
Fetch Application Data
Notify Student
<<Include>>
<<Include>>
<<Include>>
Examine New Course Application
Examine Preapproval Form
Notify About Incorrect Application
<<Extend>>
<<Extend>>
Start University Placements
extension points
Notify About Incorrect Application
Approve New Student From Waiting Bin
Approve New Student From Waiting Bin
View Events
Enter Organization Information
Fill Course Transfer Form
<<Include>>
Sign Documents
Sign Transkript
Fetch Course Coordinator Data
Student
Graduate Student
Undergradate Student
International Students Office
Bilkent Application Website
Email System
BCC
Instructor
Coordinator
Exchange Coordinator
Erasmus Coordinator
Board Members

Use Case Name: Upload Erasmus Points Excel File

Participating Actor: International Students Office

Entry Condition: User with International Students Office role logins to Erasmus Website and clicks on the relevant section.

Exit Condition:

- User uploads excel file successfully, or

- User exits the website, or
- User cancels upload, or
- Process is unsuccessful

Flow of Events:

1. User selects the excel file with Erasmus points to upload
2. User selects program type (Erasmus or Exchange)
3. If program type is Erasmus, user selects department
4. User selects education status (graduate or undergraduate)
5. User presses submit button
6. Upload status is demonstrated on the screen at the end. (Successful or unsuccessful)

Use Case Name: Send Official Transcript to Coordinators

Participating Actor: International Students Office

Entry Condition: After the host organization sends an official transcript to International Students Office, a user with International Students Office role logins to Erasmus Website and clicks on the relevant section.

Exit Condition:

- User sends official transcript successfully, or
- User cancels process, or
- User exits the website, or
- Process is unsuccessful

Flow of Events:

1. User selects the transcript's school
2. User selects the student's department
3. The system checks the school and department and finds the assigned coordinator
4. User selects the transcript file to upload
5. User presses submit button
6. Upload status is demonstrated on the screen (Successful or unsuccessful)
7. System sends the transcript to the assigned coordinator if successful

Use Case Name: View Schools And Assigned Coordinators

Participating Actor: Student, Coordinator

Entry Condition: User with Student role or Coordinator role logins to Erasmus Website and clicks on the relevant section.

Exit Condition:

- User exits viewing, or
- User exits website

Flow of Events:

1. User views schools and assigned coordinators
2. User can filter viewing by program type, by semester type, by location
3. User can view information until they decide to go back or exit

Use Case Name: View Previously Approved Courses

Participating Actor: Student

Entry Condition: User with Student role logins to Erasmus Website and clicks on relevant section.

Exit Condition:

- User exits viewing, or
- User exits the website

Flow of Events:

1. User views previously approved or denied courses
2. User can filter viewing by department, school, approve status (denied or approved)
3. User can view information until they decide to go back or exit

Use Case Name: Cancel Erasmus Application Before Placements

Participating Actor: Student

Entry Condition: User with Student role logins to Erasmus Website and clicks on the relevant section before placements are announced.

Exit Condition:

- User exits cancellation process, or
- User exits the website, or
- Process is completed

Flow of Events:

1. User clicks on “Cancel Application” button

2. The user approves that they are indeed canceling

3. The system is being notified that the user will not be taken place in placements

4. The coordinators will be informed through the system

5. Cancellation status is demonstrated on the screen at the end. (Successful or unsuccessful)

Use Case Name: Cancel Erasmus Application After Placements

Participating Actor: Student

Entry Condition: User with Student role logins to Erasmus Website and clicks on the relevant section if placements are announced.

Exit Condition:

- User exits cancellation process, or

- User exits the website, or

- Process is completed

Flow of Events:

1. User clicks on the “Cancel Application” button

2. The user approves that they are indeed canceling

3. User selects between canceling all applications or canceling the placed school

4. If the user selects canceling all applications, their application process will be terminated, and the next student from the waiting list will be assigned

5. If the user selects to only cancel the placed school, the next student from the waiting list will be assigned, and the user will be placed on the waiting list again

6. The coordinators will be informed through the system

7. Cancellation status is demonstrated on the screen at the end. (Successful or unsuccessful)

Use Case Name: Fill Preapproval Form

Participating Actor: Student

Entry Condition: User with a Student role logs in to Erasmus Website and clicks on the relevant section after placements are done.

Exit Condition:

- User exits process, or
- User exits the website, or
- Process is completed

Flow of Events:

1. User chooses relevant courses and their equivalent from previously approved courses

2. User can filter the courses by choosing department-specific courses, or user can search for a course by entering the course code

3. User submits the form by pressing the “submit” button

4. Form status is demonstrated on the screen at the end. (Sending is successful or unsuccessful)

5. If the form status is successful, the system sends the form to the assigned coordinator

Use Case Name: Add New Course

Participating Actor: Student

Entry Condition: It is an extension of the fill preapproval form. User can add new courses while they are in the state of filling preapproval form.

Exit Condition:

- User exits process, or
- User exits the website, or
- Process is completed

Flow of Events:

1. User adds a link to the syllabus of the course from the host organization
2. User selects if the course is similar to a must course or elective course
3. If it is a must course, the user selects the equivalent must course from Bilkent
4. If it is an elective course, the user selects the equivalent elective course from Bilkent
5. Form status is demonstrated on the screen at the end. (Sending is successful or unsuccessful)
6. If it is a must course, the form of adding a new course is sent to the assigned coordinator. If coordinator approves, the course will be sent to the assigned instructor
7. If it is an elective course, the form of adding a new course is sent to the assigned coordinator

Use Case Name: Add Project Course

Participating Actor: Student

Entry Condition: It is an extension of the fill preapproval form. User can add project courses while they are in state of filling preapproval form.

Exit Condition:

- User exits process, or
- User exits the website, or
- Process is completed

Flow of Events:

1. User selects a project course
2. System notes the users selected project course for uploading after returning from the host organization
3. Status is demonstrated on the screen at the end. (Sending is successful or unsuccessful)

Use Case Name: Upload Project Report

Participating Actor: Student

Entry Condition: It is extension of adding a project course. User adds their project report after returning from host organization.

Exit Condition:

- User exits process, or

- User exits the website, or
- Process is completed

Flow of Events:

1. If user had a project course, user selects project report to upload
2. User clicks on submit button to submit the form
3. Status is demonstrated on the screen at the end. (Upload is successful or unsuccessful)
4. System sends the project report and the course information to the assigned coordinator

Use Case Name: Examine New Course Application

Participating Actor: Instructor, Coordinator

Entry Condition: User with Instructor role or Coordinator role logs in to Erasmus Website and clicks on the relevant section if there is a new course application form assigned to them.

Exit Condition:

- User exits process, or
- User exits the website, or
- User accepts or denies form

Flow of Events:

1. User checks the form that contains student information, syllabus link, and equivalent course from Bilkent
2. User accepts or denies the course application form

3. If user wants to accept the course application form, user clicks on accept button.

4. If user denies the course application form, user writes feedback and reason.

5. User presses submit button.

6. Status is demonstrated on the screen at the end. (Successful or unsuccessful)

7. System sends feedback to assigned student

Use Case Name: Examine Preapproval Form

Participating Actor: Coordinator

Entry Condition: User with Coordinator role logs in to Erasmus Website and clicks on relevant section if there is a preapproval form.

Exit Condition:

- User exits process, or
- User exits website, or
- User accepts or denies form

Flow of Events:

1. User checks the form that contains student information, selected courses, equivalent course from Bilkent

2. If user accepts form, student will be informed

3. If there are elective courses needed to be merged, user merges the courses (to fulfill ECTS criteria)

4. If user wants to accept the form, user presses on accept button

5. If user denies preapproval form, user writes feedback and reason

6.. User presses submit button

7. Status is demonstrated on the screen at the end. (Successful or unsuccessful)

8. System sends feedback to assigned student

Use Case Name: View Events

Participating Actor: Coordinator

Entry Condition: User with Coordinator role logs in to Erasmus Website and clicks on the relevant section.

Exit Condition:

- User exits process, or
- User exits the website, or

Flow of Events:

1. User enters the view events section
2. User views current events (pending preapproval forms, pending course applications, etc.)
3. User can filter event types (only preapproval forms, only waiting list placements, etc.)
4. User views events until they exit the process

Use Case Name: Enter Organization Information

Participating Actor: Coordinator

Entry Condition: User with Coordinator role logs in to Erasmus Website and clicks on relevant section.

Exit Condition:

- User exits process, or
- User exits the website, or
- User completes the process

Flow of Events:

1. Assigned organizations that the user should fill will be displayed
2. User selects an organization
3. User fills in organization information (applicable semester, quota, etc.)
4. User submits information

Use Case Name: Start University Placements

Participating Actor: Coordinator

Entry Condition: User with Coordinator role logs in to Erasmus Website and clicks on the relevant section after International Students Office uploads Erasmus points and all school information is filled in.

Exit Condition:

- User exits after pressing start placements, or
- User exits process, or
- User exits the website

Flow of Events:

1. User presses the start placements button

2. After that, system starts placements of coordinator’s department (or if it is Exchange, starts general placement)

3. System notifies students about their placements after the placement process finishes

Use Case Name: Notify About Incorrect Application

Participating Actor: Coordinator

Entry Condition: It is an extension of Start University Placements use case. After placements are done, there may be incorrect applications (e.g. incorrect semester). User enters “Notify About Incorrect Application” use case if that’s the case.

Exit Condition:

- User exits after pressing start placements, or
- User exits process, or
- User exits website

Flow of Events:

1. Coordinator is being informed about incorrect information
2. System asks coordinator if system should warn students about their incorrect application
3. If proceed, system notifies students
4. Students can either cancel their application or edit the information

Use Case Name: Approve New Student From Waiting Bin

Participating Actor: Coordinator

Entry Condition: It is an extension of the “Start Placements” use case. Entry condition is that the placements must be started, and a student should cancel their application.

Exit Condition:

- User exits process, or
- User exits the website, or
- User completes the process

Flow of Events:

1. User is notified about the top student from the waiting bin
2. User sees student information and the student’s assigned organization information
3. User approves or denies the student
4. If approved, student gets notified

Use Case Name: Fill Course Transfer Form

Participating Actor: Coordinator

Entry Condition: User with the Coordinator role logs in to Erasmus Website and clicks on the relevant section.

Exit Condition:

- User exits process, or
- User exits the website, or
- User completes the process

Flow of Events:

1. User selects an assigned student (students are assigned from assigned schools)
2. User can view the student’s preapproval form to compare
3. User fills out information to prepare course transfer form
4. User presses submit button to submit course transfer form
5. System sends the form to the board

Use Case Name: Sign Documents

Participating Actor: The Board

Entry Condition: It is included in the “Fill Course Transfer Form” use case. To enter, the Fill Course Transfer Form use case should be completed first.

Exit Condition:

- User exits process, or
- User exits the website, or
- User completes the process

Flow of Events:

1. User can view the course transfer form
2. User signs the form by e-signature or by printing, signing and uploading the form back (for security if users do not prefer having an e-signature)
3. User can deny signing. If so, the reason for denial is typed, and the form with feedback is sent back to the coordinator

Use Case Name: Sign Transcript

Participating Actor: The Board

Entry Condition: User with The Board role logs in to Erasmus Website and clicks on the relevant section if International Students Office uploaded any transcript.

Exit Condition:

- User exits process, or
- User exits the website, or
- User completes the process

Flow of Events:

1. User selects a transcript to view
2. User views the official transcript that International Students Office uploaded
3. User signs the transcript by e-signature or by printing, signing, and uploading the form back (for security if users do not prefer having an e-signature)
4. User can deny signing. If so, the reason for denial is typed, and the transcript with feedback is sent back to International Students Office

Use Case Name: Notify Student

Participating Actor: Email System

Entry Condition: Enters if there is an incorrect application, a form denial with feedback, or a course denial with feedback.

Exit Condition:

- Exits after email is sent

Flow of Events:

1. System receives student’s email information

2. System receives a topic

3. System sends an email to the student

Use Case Name: Inform Coordinator

Participating Actor: Email System

Entry Condition: Student cancels their application before placements.

Exit Condition:

- Exits after email is sent

Flow of Events:

1. System receives the coordinator’s email information

2. System receives a topic

3. System sends an email to the coordinator about the cancellation

Use Case Name: Fetch Application Data

Participating Actor: Bilkent Application Website

Entry Condition: Applications should be completed.

Exit Condition:

- Data fetching is completed

Flow of Events:

1. Fetches application data of students to create accounts

Use Case Name: Fetch Course Coordinator Data

Participating Actor: BCC

Entry Condition: Course coordinators should be finalized.

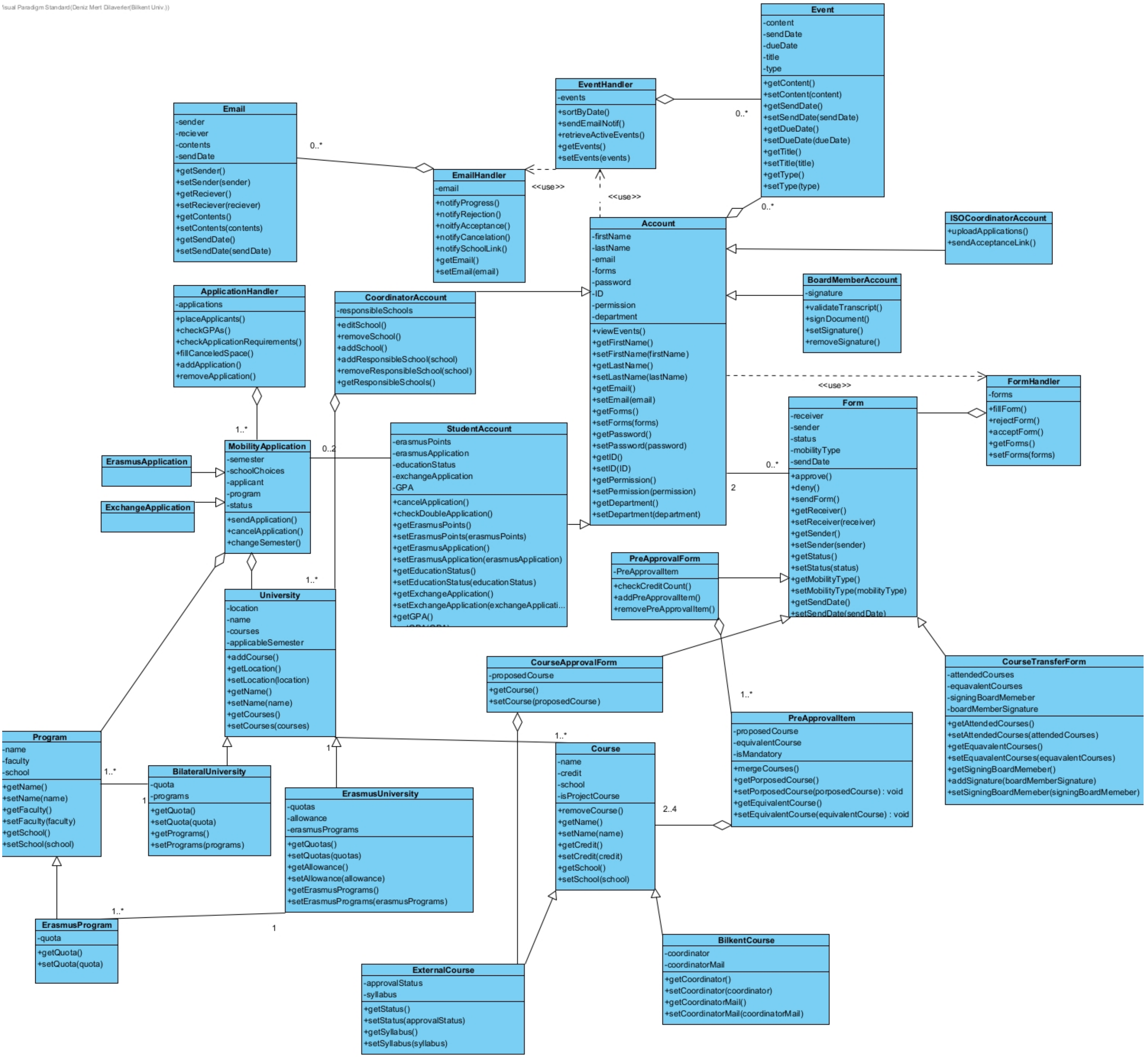
Exit Condition:

- Data fetching is completed

Flow of Events:

1. System fetches course name, department of the course, course code, and assigned course coordinator
2. System matches approved courses with fetched course coordinator information

5.2 Object and Class Model



Main classes and their relations with each other of ErasXchange are shown here. The classes are more focused on data encapsulation and the relation of said data rather than functionality which will be handled more in-depth in the Design Report. For the back end, once the request pathings are designed, controller classes will be implemented to direct the requests to the relevant service class. Service classes will also be designed to handle incoming requests using the model classes in the Object and Class Model. Thus ErasXchange will be implemented in accordance to MVC patterns. The classes will also be used to create the database schema.

5.2.1 Account Classes

In ErasXchange, there must be a class that encapsulates a user's credentials and functionalities. There are varying user types but all of them have certain properties and functionalities in common. Hence an Account class will be extended by the following classes: StudentAccount, CoordinatorAccount, BoardMemberAccount, and ISOCoordinatorAccount. All of these classes have properties such as firstName, lastName, email, and an AccountFormHandler class to handle the form functionalities of all account classes. Accounts will also be connected to EventHandler classes that will handle events for the user's account.

5.2.1.1 StudentAccount

StudentAccounts will be accounts owned by students. They will have the original properties of Bilkent ID, education status (graduate or undergraduate) Erasmus points, and CGPA. There will also be the 2 applications (Erasmus and Bilateral) applications that will be connected to the StudentAccount. The StudentAcount class will be able to use application classes to do actions on Applications. There will also be a placedSchool school property for all students that will be initialized after the application results.

5.2.1.2 CoordinatorAccount

CoordinatorAccount classes will be for accounts of Erasmus/Exchange coordinators who will have properties like responsibleSchools which will be the schools that the said coordinator will be responsible for. This will allow the balancing of events between coordinators with respect to school. The coordinatorAccount will also be able to add universities to the system.

5.2.1.3 BoardMemberAcoount

Since the board members have to sign a lot of papers. Their signature will be stored in this class. However, if the BoardMember doesn’t want their signature on the system due to security reasons the member can send their signature to the system during document signings. The BoardMemberAccount will validate transcripts and upload and remove signatures.

5.2.1.4 ISOCoordinatorAccount

ISO coordinators will be able to upload the applications to the system with uploadApplications() function. They will also be able to send the acceptance link that was received from the host school with sendAcceptanceLink() function.

5.2.2 MobilityApplication Classes

Despite having the same properties, MobilityApplication classes will be separated as Erasmus and Bilateral. This is designed so that it is clearer to the developer during implementation. The MobilityApplicationClass has properties like the semester for the applied semester, choices for the chosen schools, applicant to connect the application to its StudentAccount, and program to show which program the student is applying to. The student will also be able to use changeSemester() to change the application semester and cancelApplication() to cancel their application. The MobilityApplication classes will be handled by ApplicationHandler class.

5.2.2.1 ApplicationHandler

ApplicationHandler class will have an ordered list of applications that will be ordered by the Exchange’s/Erasmus’s ordering criteria. The handler will check for edge cases and requirements of applications. It will also handle canceled applications and fill in their replacements.

5.2.3 University Classes

University classes are for encapsulating necessary information about schools that the students have applied to. The University classes will be used for connecting the Course classes to their respective institutions

.

5.2.3.1 ErasmusUniversity and BilateralUniversity Classes

The University class is extended by ErasmusUniversity and BilateralUniversity class to handle differences in properties like ErasmusUniversity having ErasmusPrograms, individual programQuotas, and allowances, whereas BilateralUniversity classes will have a general quota and program classes.

5.2.3.2 ErasmusProgram and Program classes

They are used to show which programs the universities offer. ErasmusProgram extends Program because unlike Exchange universities, Erasmus universities have quotas for their programs and not their school as a whole

5.2.4 Course Classes

Course classes will be used for forms like course transfer forms or preapproval forms. Course class is extended as BilkentCourse and external course. Course classes have base information about a course like a name, school, credit and etc.

5.2.4.1 BilkentCourse and ExternalCourse class

These classes had to be separated for the additional properties each needed. BilkentCourse class requires the coordinator to the approval of equivalent courses, and the ExternalCourse class requires the status of acceptance of the said course.

5.2.5 Form Classes

Form classes are a crucial part of ErasXchange’s vision to eliminate hard-copy paperwork from Erasmus/Exchange applications. Hence Form class is used for the base requirements of all forms like sender, receiver and etc. Form classes are extended by the following classes: CourseApprovalForm, PreApprovalForm, CourseTransferForm.

5.2.5.1 CourseApprovalForm

Courses of host universities first need to be registered to the system. To approve courses first, CourseApprovalForm classes will be used to send the proposed course to be approved to the coordinator. The coordinator can then approve or deny and send the form back.

5.2.5.2 PreApprovalForm

PreApprovalForm class will be used for students to encapsulate the proposed and equivalent courses. The courses will be in PreApprovalItem class to make the proposed and equivalent courses each other's indices. The PreApprovalItem class will also support two or more courses that are merged into a singular course.

5.2.5.3 CourseApprovalForm

CourseApprovalForm class is for registering the transcript that the host university has sent after the mobility of the students. The class is filled with the taken courses, the grade the course was passed with, and the equivalent course. The form will also require a signature from the BoardMember and the signing BoardMember information.

5.2.6 Event Classes

Another crucial feature of ErasXchange is that it can handle many events like application cancellations, form approvals, placement results and etc. Event classes are used for the notification of certain news to the user. They are needed for the user to see what events have occurred in their events tab.

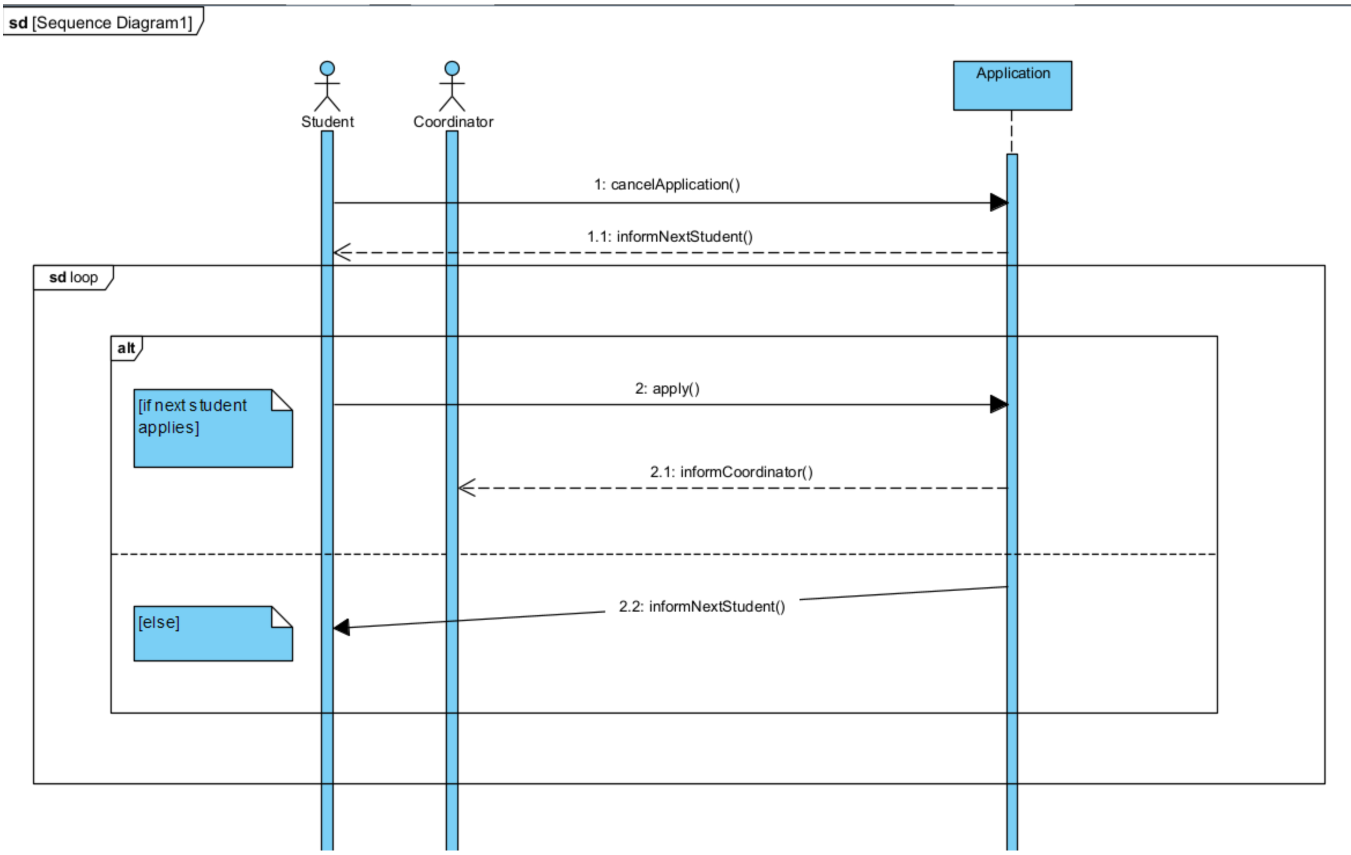
5.2.6.1 EventHandler

EventHandler class is used for handling all the events of users. The EventHandler also helps with the assignment of events to user accounts, like the assignment of events with respect to the responsible school of the coordinator. EventHandler also uses EmailHandler to send notifications of events as e-mails.

5.2.6.2 EmailHandler

EmailHandler is the class that interacts with mailing functionalities and handles the notification of events through email.

5.3 Sequence Model

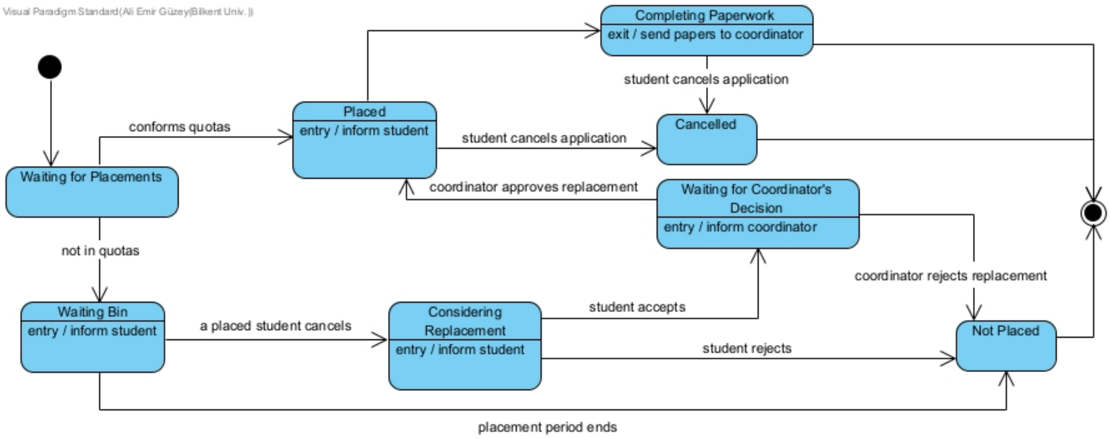


In this model, a student cancels their application. When a student cancels their application, system automatically informs next student. If the next student wants to apply, system informs the coordinator. If the next student does not apply in a given amount of time, system informs the next student in a loop until the student gap is filled or there are no more students left to inform.

5.4 State Model

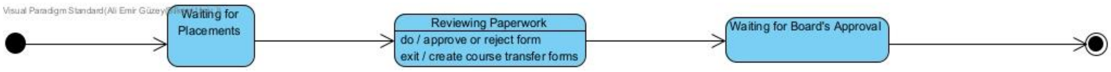
The state machine models below describe the different states of some objects in the system.

5.4.1 State Model of a Student



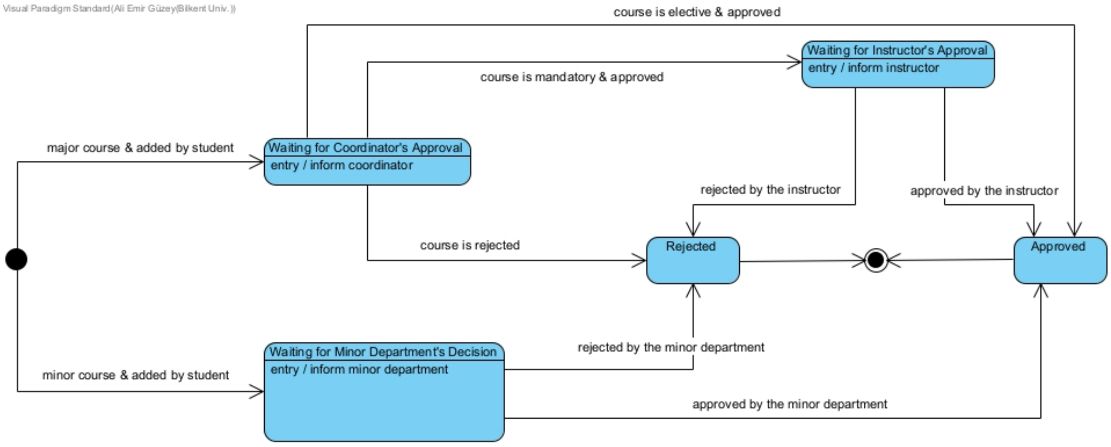
All students start the process in the 'waiting' state. When the placements are done by the system, each student goes into either the 'placed' state or 'waiting bin' state according to the quotas and their school preferences. A student in the waiting bin can consider replacing a placed but later canceled student's exchange placement. If the student accepts, he/she waits for the coordinator's decision. He/she becomes a placed student and goes to the 'completing paperwork' state where he/she will be filling his/her preapproval form and proposing it to the coordinator.

5.4.2 State Model of a Coordinator


Visual Paradigm Standard(Ali Emir Güzey(
Waiting for Placements
Reviewing Paperwork
do / approve or reject form
exit / create course transfer forms
Waiting for Board's Approval

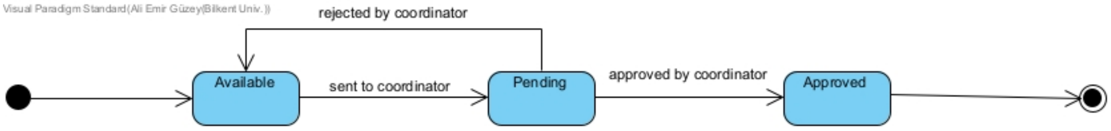
A coordinator waits until the placements are done. After that, he/she goes into 'reviewing paperwork' state, where he/she will review students' preapproval forms and create course transfer forms for returning students. Then, he/she presents the forms to the board's approval.

5.4.3 State Model of a Course


Visual Paradigm Standard(Ali Emir Güzey(Bilkent Univ.))
course is elective & approved
Waiting for Instructor's Approval
entry / inform instructor
course is mandatory & approved
major course & added by student
Waiting for Coordinator's Approval
entry / inform coordinator
rejected by the instructor
approved by the instructor
course is rejected
Rejected
Approved
Waiting for Minor Department's Decision
entry / inform minor department
rejected by the minor department
minor course & added by student
approved by the minor department

A course pair (a course at the host university and its counterpart at Bilkent) can be added to the system by a student. If the added course is a course from the student's major program, the course waits for the coordinator's approval. If the counterpart at Bilkent is a technical elective from the student's curriculum, only the coordinator's approval is enough to add the course to the system. If the course is being added in place of a mandatory course at Bilkent, then that course's instructor's approval is needed to add the course. If the course is from the minor program of the student, the department of the minor program is needed to add the course to the system.

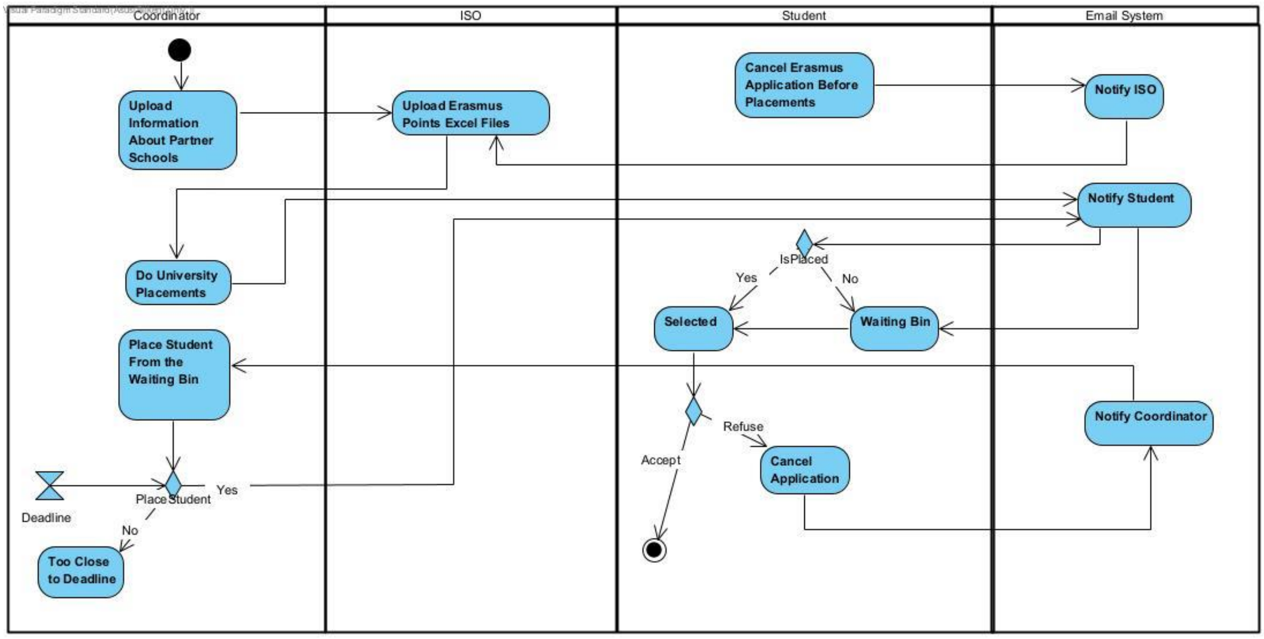
5.4.3 State Model of a Preapproval Form



A preapproval form is available to the placed students to fill in with the courses they plan to take at the host university. When they fill out and send the form to the coordinator, the coordinator either approves or rejects it. If the form is rejected, it becomes available again to the student to make the changes according to the coordinator's feedback. If the coordinator approves the form, the process ends.

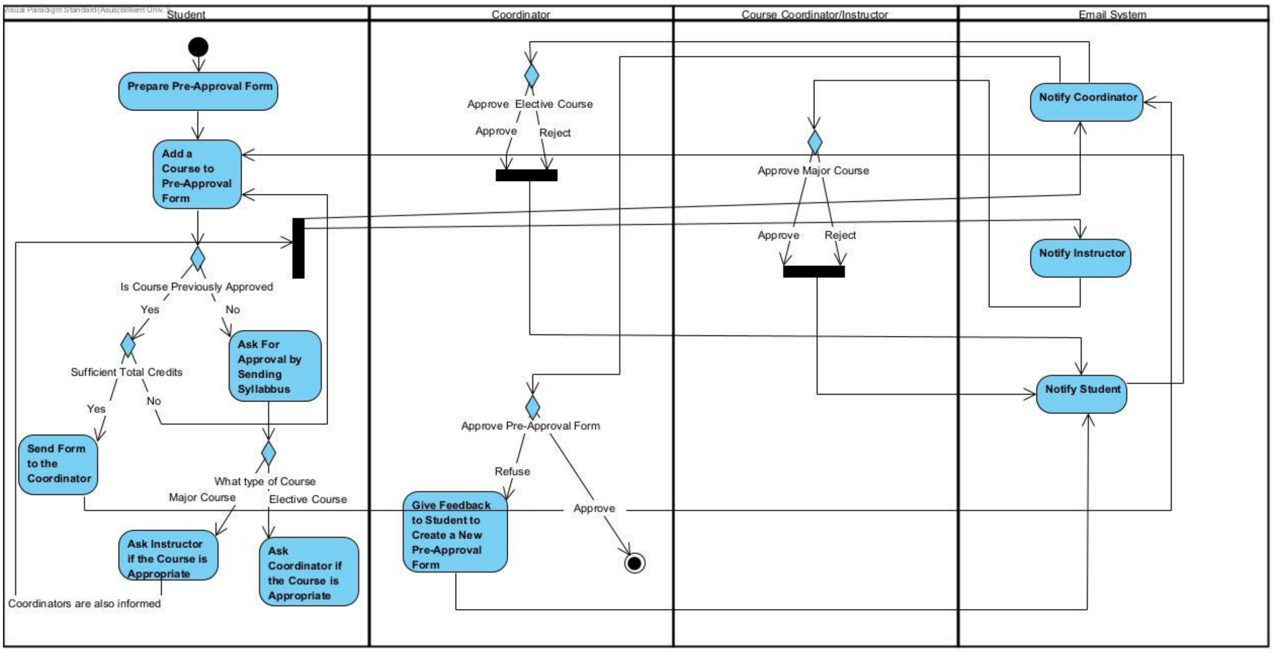
5.5 Activity Model

5.5.1 Activity Model of Student Placements



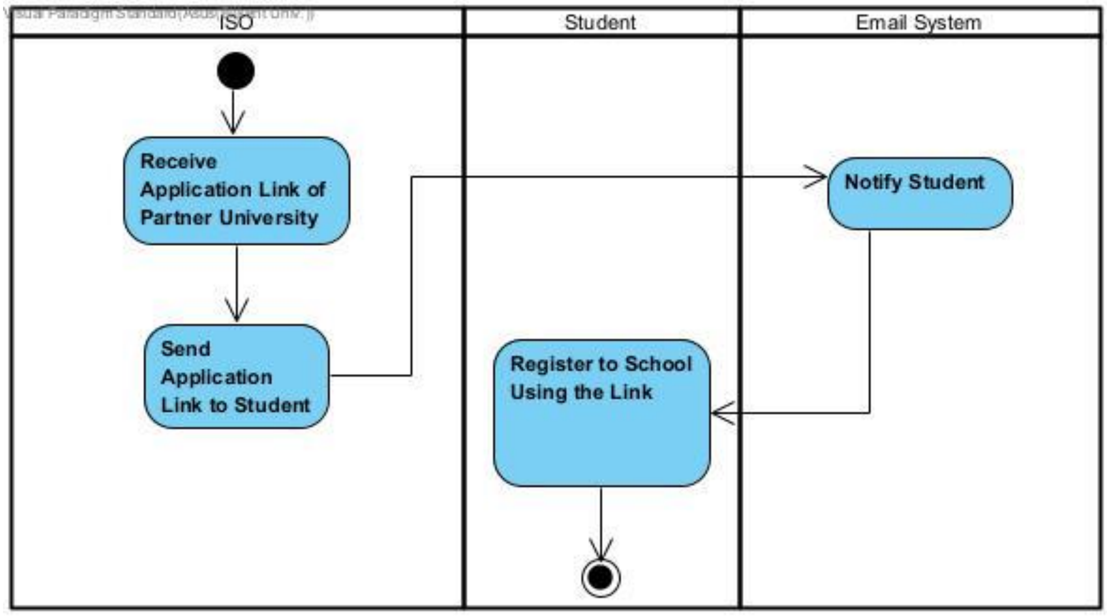
In this activity diagram, the process of student placements is shown. The process starts with coordinators as they upload information about partner schools such as the location, language requirements, available semesters etc. After this, students can apply for the Erasmus Programme and cancel their applications. Then the International Students Office (ISO) uploads the Excel sheets containing Erasmus points of each Applicant. Coordinators then can make placements by using the system according to the Excel sheets. After the placements students are notified if they are selected or not. Students who are not selected are placed in a waiting bin. If a selected student cancels their application, the coordinator is notified. Coordinator then may decide to select the top student from the waiting bin as a replacement. If time is too close to the deadline then the coordinator may also decide not to choose a replacement. This activity diagram finishes as selected students accept their schools.

5.5.2 Activity Model of Preparing Pre-Approval Forms


Visual Paradigm Standard(Asus(Bilkent Univ.))
Student
Coordinator
Course Coordinator/Instructor
Email System
Prepare Pre-Approval Form
Add a Course to Pre-Approval Form
Is Course Previously Approved
Yes
No
Sufficient Total Credits
Yes
No
Ask For Approval by Sending Syllabus
Send Form to the Coordinator
What type of Course
Major Course
Elective Course
Ask Instructor if the Course is Appropriate
Ask Coordinator if the Course is Appropriate
Coordinators are also informed
Approve Elective Course
Approve
Reject
Approve Pre-Approval Form
Refuse
Approve
Give Feedback to Student to Create a New Pre-Approval Form
Approve Major Course
Approve
Reject
Notify Coordinator
Notify Instructor
Notify Student

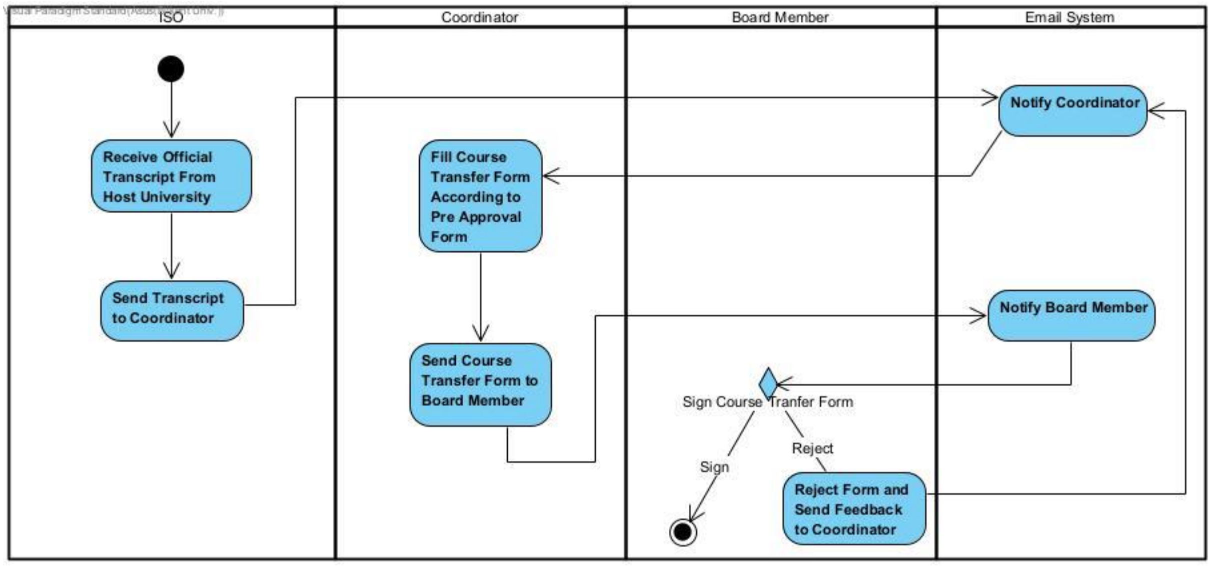
In this activity diagram, the process of preparing the Pre-Approval form is shown. Pre-Approval Forms are an important part of the Erasmus Programme as they indicate the courses students will take at the partner university and how these courses will be counted towards students' own university's curriculum. Process starts as students add courses to the Pre-Approval Form. Some of the courses are previously approved as mentioned in [7]. However, if the course is not previously approved, students need to ask for approval by sending the syllabus to the official. If it is an elective course coordinators are notified and they approve or reject the course. If it is a major course both Course Coordinators/Instructors and Erasmus/Exchange Coordinators are notified. Instructors can approve or reject the course. Depending on the approved courses, students finalize their Pre-Approval Form and if it has sufficient total credits (30-33 ECTS) [2] they can send the form to coordinators. Coordinator can reject the Pre-Approval form and give feedback to the students or they can accept the form which ends the Pre-Approval form process.

5.5.3 Activity Model of Registering to Partner School


ISO
Student
Email System
Receive Application Link of Partner University
Send Application Link to Student
Notify Student
Register to School Using the Link

In this diagram, the process of registering to a partner school is shown. The process starts as the International Students Office receives an application link from the partner university. Then ISO sends this link to the student by using the system. Students receive the link and register to the school by using the link which ends the process.

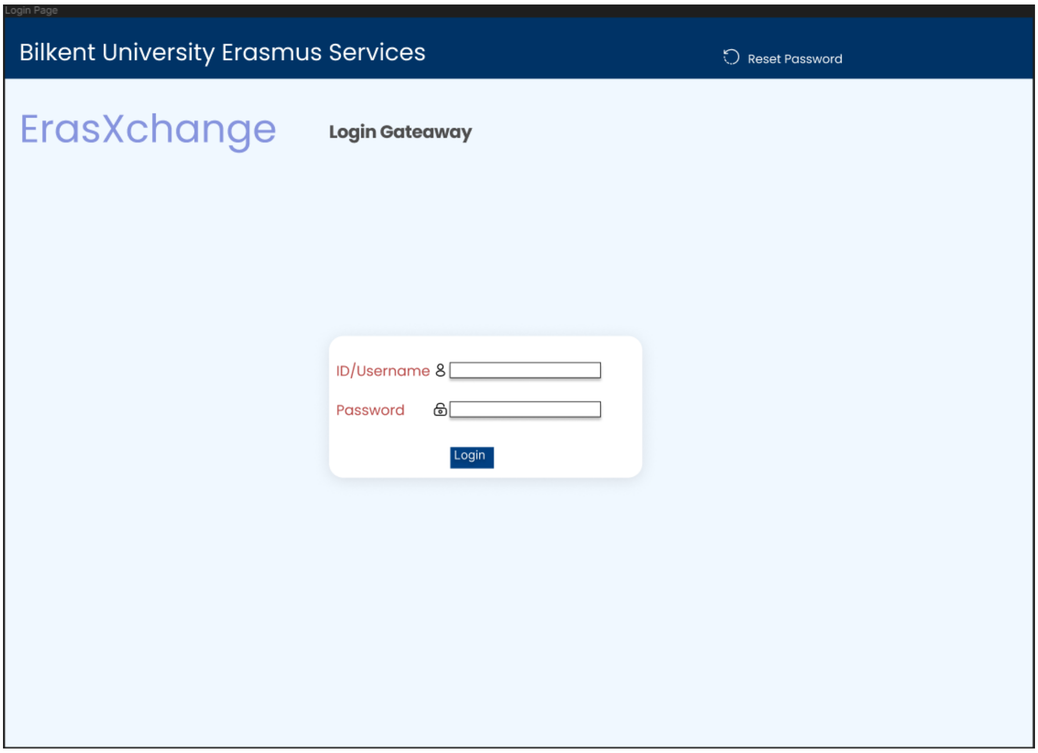
5.5.4 Activity Model of Course Transfer Procedure


ISO
Coordinator
Board Member
Email System
Receive Official Transcript From Host University
Send Transcript to Coordinator
Fill Course Transfer Form According to Pre Approval Form
Send Course Transfer Form to Board Member
Notify Coordinator
Notify Board Member
Sign Course Tranfer Form
Sign
Reject
Reject Form and Send Feedback to Coordinator

In this activity diagram, the course transfer procedure is shown. This procedure takes place after the student's Erasmus Program ends. Procedure starts as the International Students Office receives students' official transcript from the host university. ISO then sends this transcript to the coordinator. Coordinator prepares a Course Transfer Form based on the transcript and Pre-Approval Form of the student. Then the coordinator sends the Course Transfer Form to the Board in order to be signed. Board may choose to reject the form and send a feedback to the coordinator or they can sign the document which ends the process.

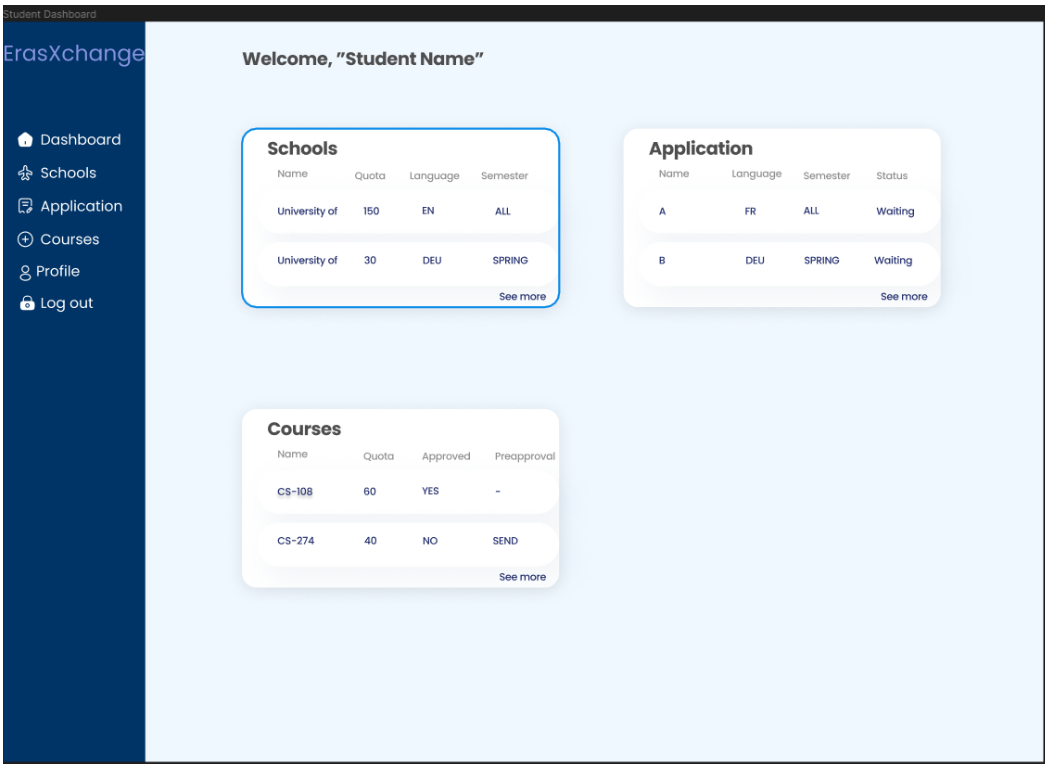
5.6 User Interface

5.6.1 Login Gateway



The web application starts with a login getaway that logs a user into a coordinator or student role. Both roles have different UIs.

5.6.2 Student Dashboard



Schools

Name	Quota	Language	Semester
University of	150	EN	ALL
University of	30	DEU	SPRING

Application

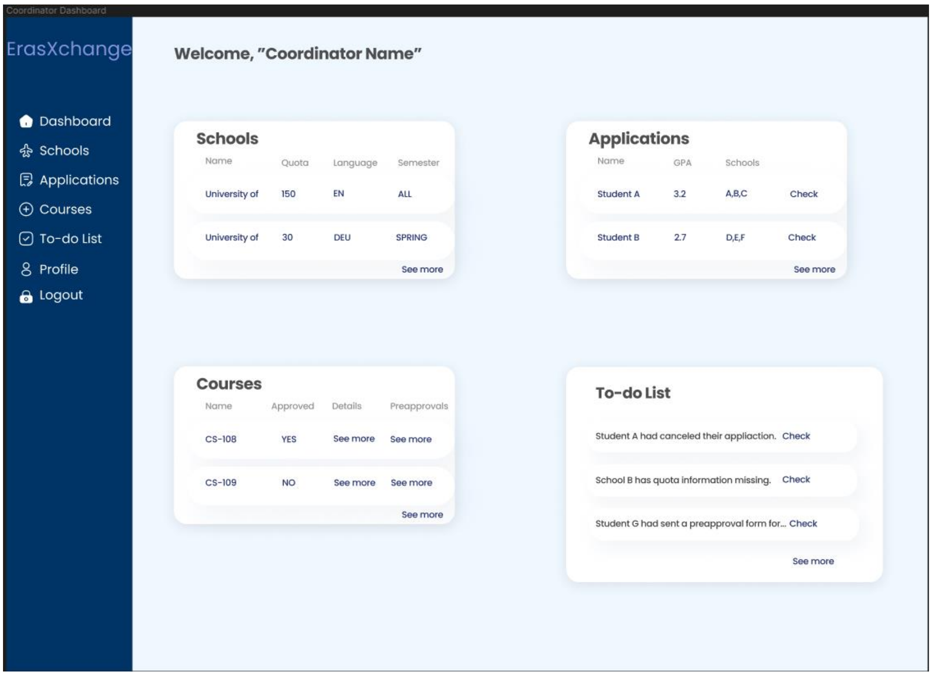
Name	Language	Semester	Status
A	FR	ALL	Waiting
B	DEU	SPRING	Waiting

Courses

Name	Quota	Approved	Preapproval
CS-108	60	YES	-
CS-274	40	NO	SEND

Students will be directed to the dashboard of their profile designed as above. The dashboard is designed in such a way that navigation between different tabs are simple and smooth. There is a sidebar and also elements seen on the dashboard that can help navigating for the needs of the student.

5.6.3 Coordinator Dashboard



Welcome, ”Coordinator Name”

Schools

Name	Quota	Language	Semester
University of	150	EN	ALL
University of	30	DEU	SPRING

See more

Applications

Name	GPA	Schools	
Student A	3.2	A,B,C	Check
Student B	2.7	D,E,F	Check

See more

Courses

Name	Approved	Details	Preapprovals
CS-108	YES	See more	See more
CS-109	NO	See more	See more

See more

To-do List

Student A had canceled their appliaction. Check

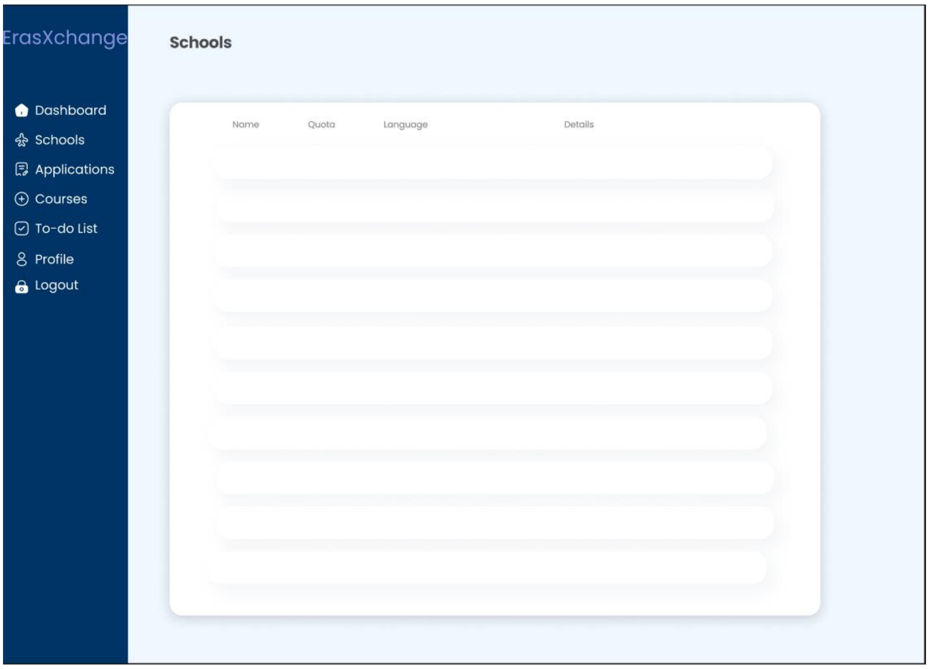
School B has quota information missing. Check

Student G had sent a preapproval form for... Check

See more

Coordinators will be directed to the dashboard of their profile designed as above. The dashboard is designed in such a way that navigation between different tabs are simple and smooth. There is a sidebar and also elements seen on the dashboard that can help navigating for the needs of the coordinator.

5.6.4 Schools

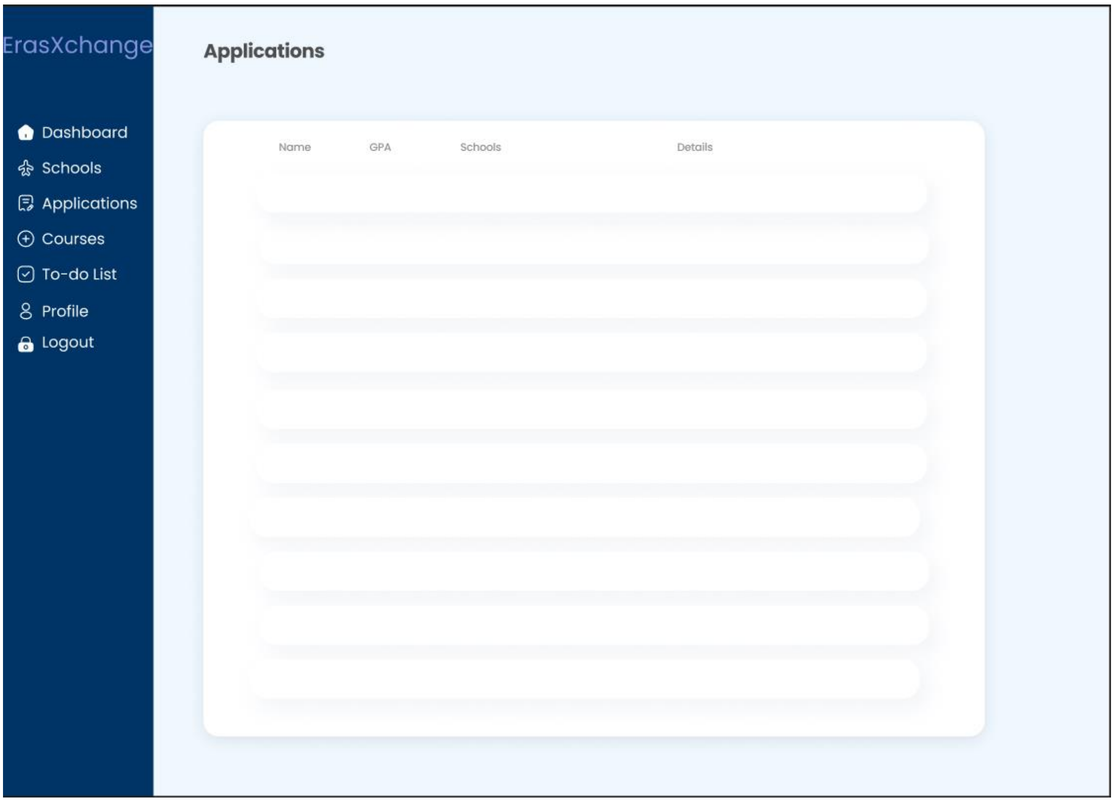


Schools

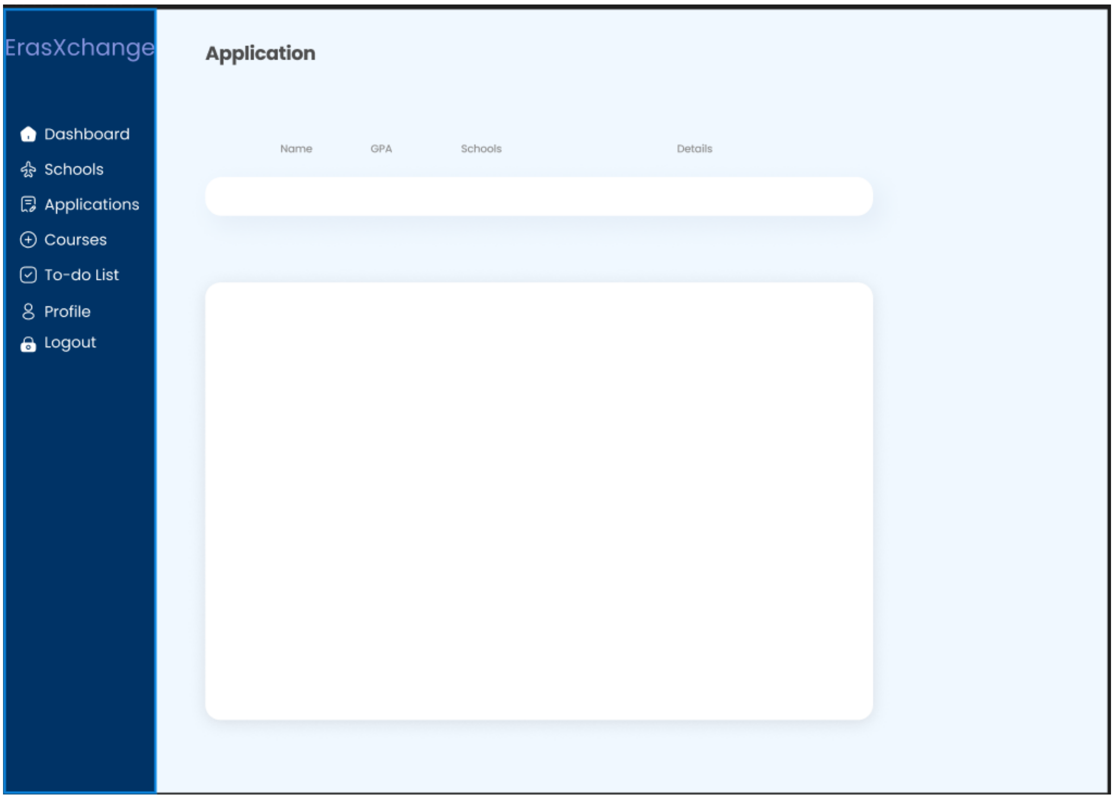
Name	Quota	Language	Details

In the schools tab, the user can see any school's information. Coordinators can edit the information.

5.6.5 Applications/Application

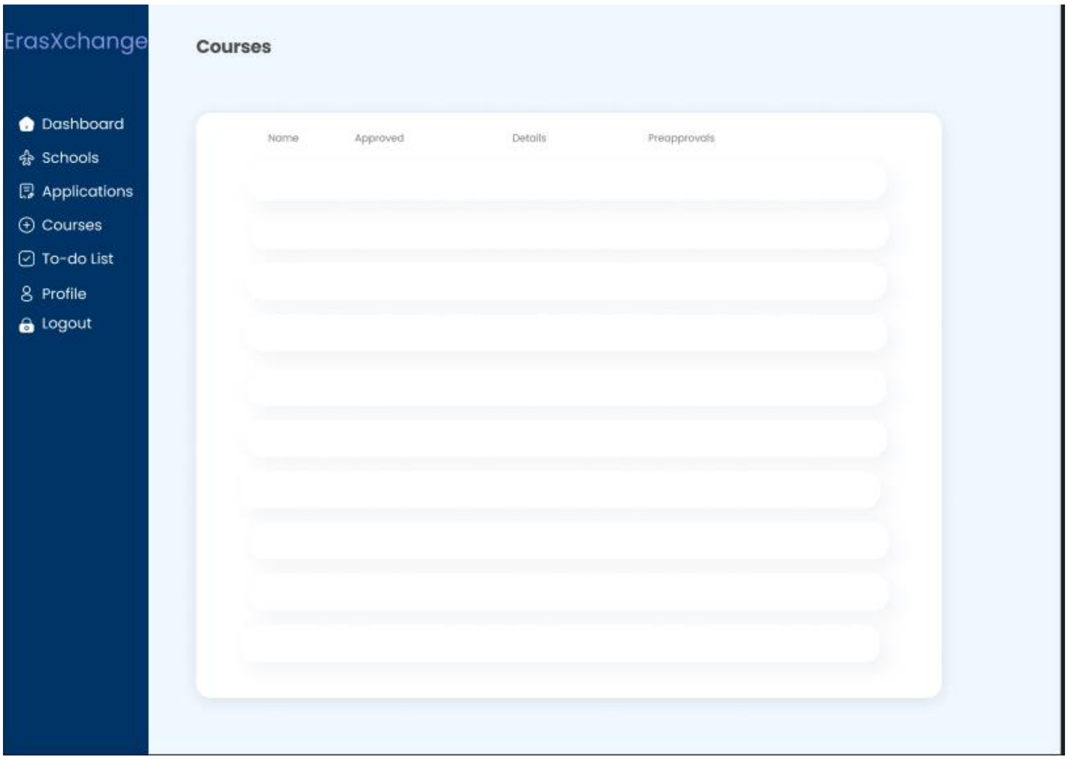


In the applications tab, a coordinator can see all the application data for each student, like above.



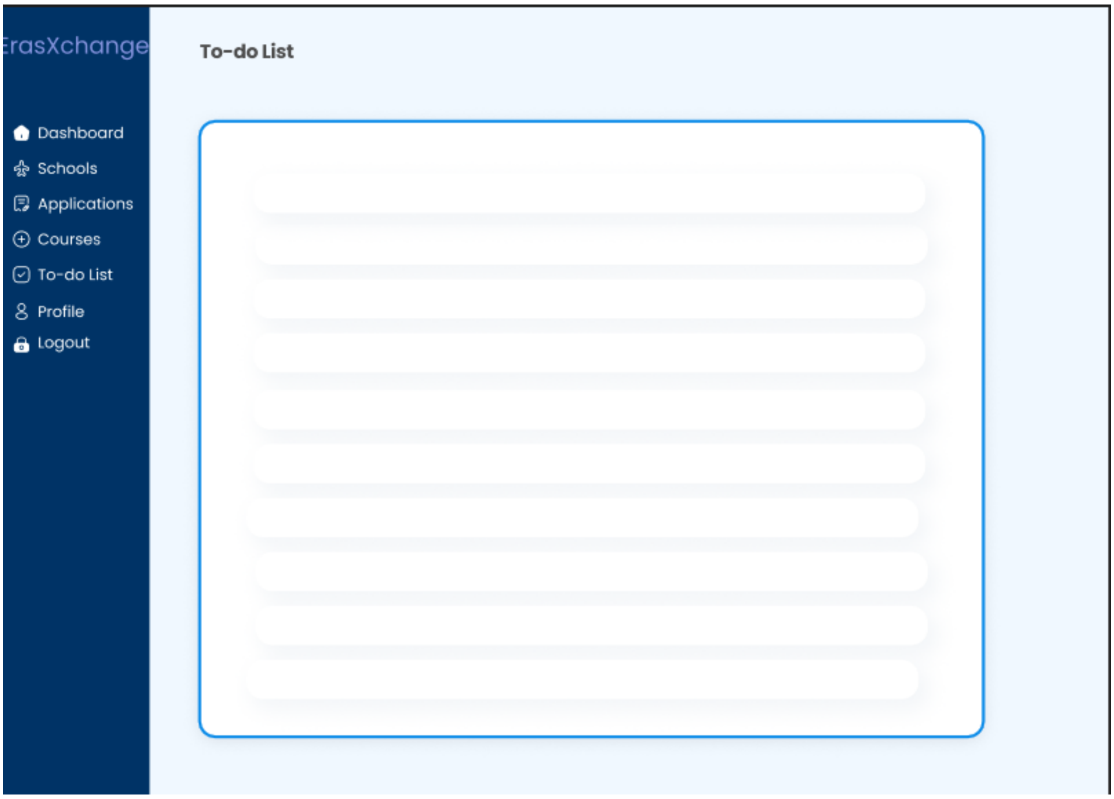
Students can only see their application data. They have an option to cancel with a simple icon and a warning pop-up.

5.6.6 Courses



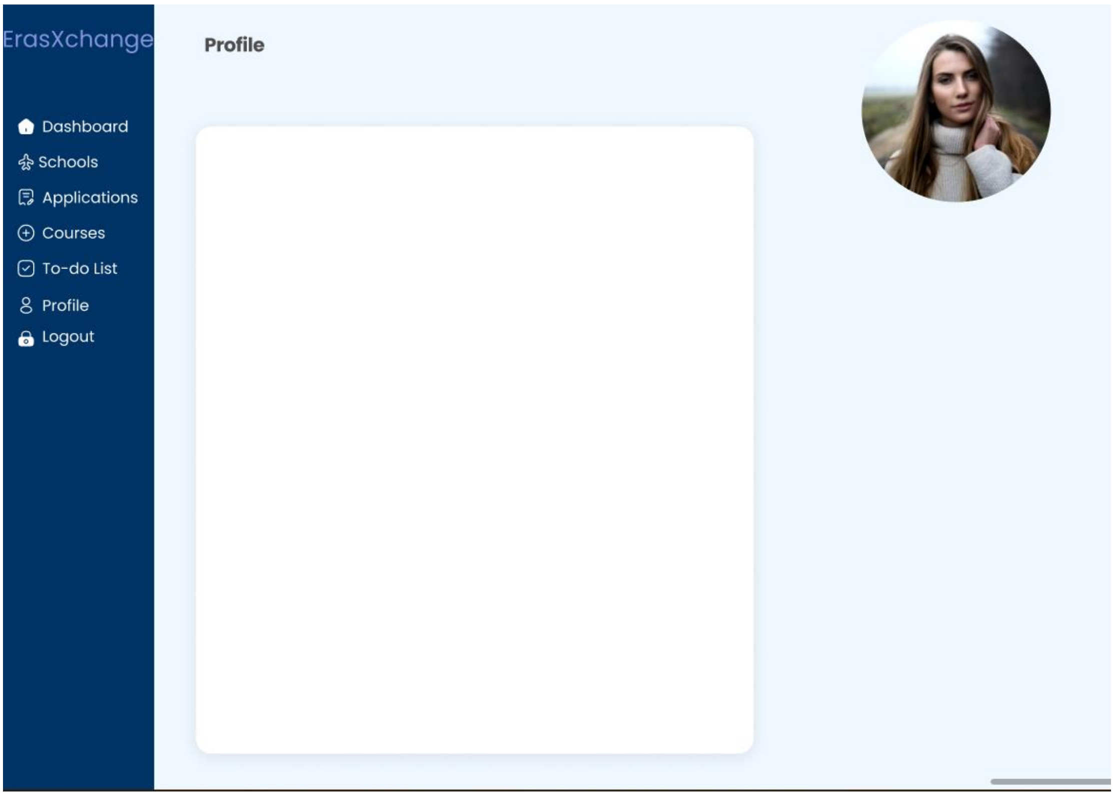
Students can see their course data, send a preapproval form. Coordinators can see every course and receive preapproval forms uploaded by students.

5.6.7 To-do List



Coordinators will have a to-do list tab reminded to them on the main dashboard page. They can easily keep track of the work that needs to be done or notifications that they might want to know.

5.6.8 Profile



Users will have a profile where they can change their information and the avatar. Their avatars will be seen in coordinators to-do list related to them, further smoothing the web application experience.

6 Conclusion

In conclusion, the ErasXchange web application aims to facilitate the Erasmus application process at Bilkent University on a single platform. The goal of ErasXchange is to reduce paper and email usage by digitalizing preapproval forms and course transfer forms. The application plans to automate most of the Erasmus/Exchange application process and make the pre-mobility period more seamless. ErasXchange aims to be a beneficial platform from which students can view organizations and courses from a single platform rather than assessing multiple websites and platforms. It will also benefit in terms of accessibility since students will no longer need to send or submit forms to multiple departments.

Finally, the proposed project has been broken down into six models to analyze its requirements. The system consists of the following models:

1. Use-case model
2. Object-class model
3. Sequence model
4. State model
5. Activity model
6. User Interface of ErasXchange

The models were created by doing a requirement analysis. Requirement analysis was done by assessing clients’ opinions, investigating students’ approach to the current process, and analyzing the current Erasmus/Exchange placement system.

7 References

[1] “Erasmus+ online application and Management System - Exchange Office - Bilkent University,” *Bilkent University | Erasmus+ Application System*. [Online]. Available: https://app.erasmus.bilkent.edu.tr/. [Accessed: 03-Nov-2022].

[2] “Erasmus and Exchange @ Bilkent,” *Google Slides*. [Online]. Available: https://docs.google.com/presentation/d/1WexlR_2Z5RNcLjnAGniQtPoU7oQOhszQ3zpipC_1eIo [Accessed: 03-Nov-2022].

[3] “Değişim Programları,” *TR / Bilkent Universitesi* –. [Online]. Available: https://w3.bilkent.edu.tr/www/degisim-programlari/. [Accessed: 03-Nov-2022].

[4] “İkili Değişim Anlaşmaları,” *Bilkent Universitesi*. [Online]. Available: https://www.bilkent.edu.tr/bilkent-tr/academic/ikili_degisim_anlasmalari.html. [Accessed: 03-Nov-2022].

[5] “Erasmus Değişim Anlaşmaları,” *Bilkent Universitesi*. [Online]. Available: https://www.bilkent.edu.tr/bilkent-tr/academic/erasmus_degisim_anlasmalari.html. [Accessed: 03-Nov-2022].

[6] “Exchange Programs,” *Bilkent University Department of Computer Engineering*. [Online]. Available: http://www.cs.bilkent.edu.tr/~exchange/ [Accessed: 03-Nov-2022].

[7] Alkan, Can. “courses-webpage” *Google Sheets*. [Online]. Available: courses-webpage [Accessed: 03-Nov-2022].