Bilkent University

Department of Computer Engineering

Senior Design Project

ErasXchange

Final Report

Group Members:

Ceren Akyar - 22003158
Deniz Mert Dilaverler - 22003530
Hasan Yarkın Kurt - 22003072
Ali Emir Güzey - 22003186
Dağhan Ünal - 22002182

Instructor: Eray Tüzün

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

Date: December 18, 2022

Contents

Final Report

ErasXchange

1 Introduction

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.

1.1 Current Implementation State

In the current implementation state, key features are implemented. Students can add pre approval forms, ISO members can add placement point excel files, and coordinators can start and publish application processes. Students can also cancel their applications in any state. If a student has faulty data, the student can cancel that current application or change that application’s semester. However, some additional features were not implemented. The unimplemented features are as follows:

- Erasmus Information page was not implemented.
- The events page was not implemented.
- The filter options in institutions and courses are not fully developed. The filter options are limited.
- Currently, there are no email notifications.
- For test data, there are only data from Erasmus programs. However, the application is also suitable for Exchange programs.

2 Lessons Learned

In the meeting process, we have learned that setting up meeting goals and a time limit is beneficial. Before putting time limits, the meeting became long and irrelevant. But with goals and limitations, we quickly addressed current issues.

Setting up an “unofficial” due date is important for teams. When there is a due date for the team members to show their work to the group, with the feedback of members, the work quality improves. Also, we stopped trying to put everything together in the last minutes before submission.

For implementation, we have learned that having clear class diagrams and mock-ups is crucial for a fast implementation. Having mock-ups also prevents members from skipping a case in implementation.

Also, using languages or libraries in implementation is also important. We have learned that trying to learn a new language and a different coding style is hard while also trying to implement the program.

3 User’s Guide

The user interface of the design is based on if the user is logged in to the system and to the role of the user. Generally, the design of the pages remains similar for all roles. However, there are small component additions to specific roles.

3.1 Login Page

Non-logged users can only see a sidebar containing institution information and information about courses offered. Users need to log in to see other details, such as their applications and form data. When users enter with their correct ID and password depending on their role assigned during the authentication process, the content of the application will change. Coordinators can see all applications while students can only see their own application. Coordinators have options such as adding a course, an institution, and approve pre-approval forms. Students have options such as canceling their application, sending pre-approval forms to the server. ISO can upload necessary files to the server.

3.2 Institutions Page

The user can see the partnered institutions and easily view them with filter options. To view institutions, a user does not have to log in to the system. However, if a coordinator views the page, there will be an additional component to add a new institution with “Add New Institution” button. When clicked, the coordinator can fill out new institution information and submit it to the system.

3.3 Applications Page

This page is visible to two roles: student and coordinator. On this page, coordinators can view student applications that are assigned to them. If the points are uploaded, the coordinator can start the placement process. After starting, if there are any faulty applications, the coordinator can select to cancel faulty applications or to send application faulty data change to the student. After all faulty data is fixed, the coordinator can publish the placement results.

Students can view their applications and cancel their application process. If placement is started and the student has faulty application data (for instance, wrong semester), the student can change the semester or cancel that application. If published, the student can view the application result and, if placed, can cancel the application.

3.4 Forms Page

Forms page has different content for students and coordinators. This allows users to use their respective functionalities according to their role in the Erasmus/Exchange process. Students can create pre-approval forms in the Forms page. By pressing the add existing course button, which is placed at the right upper corner, a pop-up is opened. In this pop-up by filling the fields in the pop-up and pressing the submit button students can add an existing course to their form. Students can also add non-previously approved courses to their pre-approval form. This is done by pressing the add new course button which is placed next to the add existing course button. Process for adding the new course through the pop-up is the same as existing courses.

Coordinators can review students' pre-approval forms in the Forms Page. In this page student forms are listed. In each form unit there are 3 buttons. These are: review button, approve button and reject button. When the review button, which has an eye icon, is pressed, the student's pre-approval form is automatically downloaded to allow coordinators to review the form. Next to the review button there is an approve button which has a tick icon. This button can be used to approve the pre-approval form. Lastly there is the reject button which has a cross button. When this button is pressed a pop-up opens where the coordinator can write feedback. When feedback is submitted, the form is rejected. On the top of forms page there is a switch button where coordinators can change the page between “forms” and “accepted forms”. In the accepted form section coordinators can see the pre-approval forms which were approved by them. Here they can write a student transfer form for the approved form.

3.5 Courses Page

The user can see the courses they can take on Erasmus and their equivalents and easily view them with filter options. To view courses, a user does not have to log in to the system. However, if a coordinator views the page, there will be an additional component to add a new course. When clicked, the coordinator can fill out new course information and submit it to the system.

3.6 Dashboard

The dashboard is the main page when users log in. If the user is a coordinator, a board member, or a student, they can view the notifications, the forms, and the application process. If it is a member of ISO (Office of International Students), they will have two options on this page: Upload placement points as excel files (if it is not already done) and upload the official transcript of students.

4 Build Instructions

Requirements:

- JDK 17 or higher
- IntelliJ IDEA
- NodeJS (also NPM)
- Docker Desktop

Instructions:

- Make sure ports 8080, 5400, and 3000 are empty.
  Run the following command on the command line:
  ```
  docker run --name postgres-cs319 -p 5400:5432 -e
  POSTGRES_PASSWORD=passwd -e POSTGRES_USER=admin -d
  postgres
  ```
- Verify from docker desktop that to verify that the container is running

- Open the following pathed project from IntelliJ IDEA: "`BilkentErasmus/backend/ErasXchange`" and download the dependencies by clicking "load maven project"
- Run the backend by entering the file `BilkentErasmus/backend/ErasXchange/src/main/java/com/caddy/erasxchange/ErasXchangeApplication.java` and running the project from IDEA.
- Now, run "`npm install`" in the "`BilkentErasmus/frontend`" directory
- Run "`npm start`" in the "`BilkentErasmus/frontend`" directory

5 Work Allocation

- Ali Emir Güzey:
  - State models, Overview in Analysis Report
  - Access Control and Security, Persistence Layer, External Packages, and Class Interfaces of Persistence Layer in Design Report
  - Backend functionality related to forms, file management, database deployment, and authentication in Implementation
- Ceren Akyar:
  - Use Case Diagram and functional requirement elicitation in Analysis reports
  - Web Browser Layer object diagram and Final Object diagram in Design Report
  - Student Application page front-end implementation
  - Institutions page front-end implementation
- Dağhan Ünal:
  - Sequence Diagram and UI mock-ups in Analysis reports
  - Hardware/Software Mapping, trade-offs, frontend layer object diagram explanations in Design Reports
  - Sidebar front-end implementation
  - login, logout functionalities with user authentication front-end implementation

    - Main navigation front-end implementation
    - Courses page front-end implementation
    - Applications (coordinator) front-end implementation
- Deniz Mert Dilaverler:
    - Non-functional requirements and Class diagrams in the Analysis Reports
    - Service Layer diagrams and interfaces, Persistent data management section in the Design Reports
    - Application, University and Course services in the back-end
- Hasan Yarkın Kurt:
    - Activity Diagrams in the Analysis reports
    - Introduction, Boundary Conditions, web browser object layer diagram explanations in the Design Reports
    - Forms (coordinator) page front-end implementation
    - Forms (student) page front-end implementation