



Bilkent University

Department of Computer Engineering

CS319 Term Project

Section 2

Group 2G

Erasmus App

FINAL REPORT

Group Members:

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

Instructor:

Eray Tüzün

TA's:

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

Table of Contents

- 1. INTRODUCTION**
- 2. LESSONS LEARNED**
- 3. USERS GUIDE**
- 4. BUILD INSTRUCTIONS**
- 5. WORK ALLOCATION**

1. INTRODUCTION

We have implemented nearly all the features that we promised in the earlier stages of the project. Of course, there are little changes, additions and deletions in the project throughout the semester due to lots of different parameters such as the restriction of the tools we have chosen. But we can say that our application, Toera, can do nearly all the things that one can expect from an Erasmus application. Here are the main features of Toera.

Forms:

Toera provides practical and user-friendly interfaces to generate all forms that are necessary throughout the whole Erasmus process (Course Transfer Form, Pre-Approval Form, Learning Agreement Form). An erasmus candidate can easily fill in the blanks with the relevant information without worrying about the form format since Toera generates the forms in the correct format with the input of the candidate. Candidates and coordinators can also edit the forms if there are any changes necessary. In addition to generating the form, Toera can also produce a PDF of any form which users can download easily. Also, university infos and other available datas will be available in Learning Agreement form, which is the most complicated form, that minimizes the research and mistakes that Erasmus Candidate can possibly do.

Chat:

Toera provides a chatting interface between its users. With this feature, communication between any coordinator or other authorized person and Erasmus candidates becomes much easier and immune to miscommunications. With all the communication between students and coordinators being done from here, it is much less likely for anyone to miss a new message or update compared to communicating with emails.

To-Do:

Toera assigns tasks to its users. This feature simplifies the whole Erasmus process for both candidates and other authorities who need to view or approve any kind of form in any stages of the process since they can constantly check whether they have any tasks to do. Users can tick the task as done when they are done with that task.

Courses:

Toera contains the list of previously accepted courses from the universities that Bilkent is Erasmus partners with by using the feature that parse the previously accepted courses from the excel sheet Bilkent provides.. Bilkent courses are provided with the list of their equivalent courses from other universities, so that an Erasmus candidate can easily see which courses they can take instead of the one that they would take in Bilkent. In addition, these courses have feedback sections which can be viewed to have a more clear idea about the course and help candidates make a healthier decision. Other than the feedback, a user can view all necessary information about the course such as ECTS credits, language requirements, etc. Also Erasmus Candidate can easily nominate a new course for a Bilkent course by providing the necessary informations such as syllabus of the course which immediately create notification for the responsible coordinator (electives for course coordinator and must courses for erasmus coordinator). When the coordinator approves the course, the course becomes available for other candidates to select.

Universities:

Toera has a capability of having all the partner universities stored in it's database with the relevant information (in test database there is 15). Users can view universities with the necessary information about them such as their location, Erasmus Code(which is necessary for the forms), mobility periods and feedbacks. As for the courses, feedbacks are also very helpful for a candidate in terms of making a suitable decision for them.

Logs:

Toera allows the erasmus coordinators to access the log data. Erasmus coordinators can navigate to the Logs Page from their main page and display the logs.

Naturally, due to limited time and other constraints, we were not able to stick to our initial ui which we designed at the start of the semester. Therefore we tried to focus on implementing the main functionalities which makes this app useful for anyone who is relevant to Erasmus. We can say that all the functionalities related to the main process are working nearly error-less. Therefore our app is kind-of ready to be used by actual users almost without problems.

2. LESSONS LEARNED

We are proud of the product that we created and also proud of the work that we put in. All the members of our team gave their best efforts to contribute to this project. Of course, looking back to this semester, there are things that we think we could have handled better.

Scheduling:

Even though we thought that we had started this project comparatively early, we have struggled with the deadlines. Making a more realistic, feasible schedule was the key to a healthier developing process. Also due to assignments of other courses, being able to work synchronously as a team was very difficult throughout the semester. Because that working asynchronously in a group project creates numerous obstacles, which delays our development process, it was very important to make progress all together. Therefore, we should have analyzed the assignments and picked the dates which we could work together as a team to be able to seize those days fully.

Tools and Technology:

For most of our group members, this project was our first web application task. Therefore, the tools and technologies which are used for web development was not

a field which we were familiar with. Because of this, choosing the libraries to use, deciding on how to implement some features and other similar obstacles were really difficult to deal with. During the implementation part, there were some points which we realized that we cannot possibly implement a certain feature as the way we planned due to the restrictions of the technology we have chosen. This wastes lots of time and effort. Therefore, what we learned is, we should have analyzed the requirements of the project very detailed, strictly and carefully. Then with the analysis we made, we should have done a very detailed research on the technologies that we might possibly use, to determine whether that technology will allow us to implement in the way we designed. And we should have chosen everything accordingly.

3. USERS GUIDE

All users will log in from the same screen with the initial credentials given by Toera. They will be able to change their password afterwards if they want to by clicking the change password section in the settings. After logging in, all user types face the main page where they can navigate into other pages by clicking the related fields in the navigation bar or by searching from the search bar which is also in the navigation bar.

Every user can view other users' profiles by searching and finding them from the search bar.

When the app initialized, admin user create an Erasmus Coordinators for the Bilkent departments that will be responsible for uploading ranked applicants excel coming from the Bilkent Authorities. Then toera creates all users by parsing the excel, and email them their initial credentials which is users responsibility to change later. Then the app becomes fully functional for the Erasmus process. User guide of the app is as follows:

3.1 Erasmus Coordinator

- Logs in to the application and app navigates to the main page.
- User can see the Todo List on the main page.
- User can navigate to Nomination Requests Page from the main page and see the course nominations made by the students. User can click the review button in that page to approve or reject the request.

- User can navigate to the university list page from main page. In this page, user can add new universities for their department, edit their information, or remove the universities.
- User can navigate to logs page from the main page. In this page, user can view the logs.
- User can search universities and users using the search bar in the main page. When clicked to the result of search, the user is navigated to the profile page of the result.
 - In the profile page of a university, user can view the university information and reviews on the university.
 - In the profile page of a student, user can view the student's personal information, application status, and send a message.
 - In the application status of a student, user can view the relevant information about the student's application, cancel the application, and access the forms.
 - In the pages of the forms, user can view or conditionally edit the form information, convert the form to PDF, give feedback, approve the form.
- By clicking the profile button on the navigation bar, user can see their personal information in their profile page, upload signature.
- By clicking to the DM button on the navigation bar, user will be navigated to chat page in which they can communicate with other user in real-time. In chat, user can select between contacts and send text messages.
- By clicking to the notifications button on the navigation bar, user can view their latest notifications as a popup.
- By clicking to the settings button on the navigation bar, user will be navigated to the settings page. In settings page, user can change their password or E-Mail.
- User can log out using the relevant button on the navigation bar.

3.2 Erasmus Candidate

- Logs in to the application and app navigates to the main page.
- User can see the Todo List on the main page.

- User can search universities and users using the search bar in the main page. When clicked to the result of search, the user is navigated to the profile page of the result.
 - In the profile page of a university, user can view the university information and reviews on the university.
 - In the profile page of a student, user can view the student's personal information and send a message.
- User can navigate to their application status from the main page.
 - In application status, they can view the relevant information, edit and view the forms, cancel application, and discard placement.
 - In the form pages, user can edit the form, submit form, and convert the form to PDF.
- By clicking the profile button on the navigation bar, user can see their personal information in their profile page, upload signature, and navigate to application status page.
- By clicking to the DM button on the navigation bar, user will be navigated to chat page in which they can communicate with other user in real-time. In chat, user can select between contacts and send text messages.
- By clicking to the notifications button on the navigation bar, user can view their latest notifications as a popup.
- By clicking to the settings button on the navigation bar, user will be navigated to the settings page. In settings page, user can change their password or E-Mail.
- User can log out using the relevant button on the navigation bar.
- User can rate their host university and write comments from the main page.

3.3 Course Coordinator

- Logs in to the application and app navigates to the main page.
- User can see the Todo List on the main page.
- User can navigate to Nomination Requests Page from the main page and see the course nominations made by the students. User can click the review button in that page to approve or reject the request.

- User can search universities and users using the search bar in the main page. When clicked to the result of search, the user is navigated to the profile page of the result.
 - In the profile page of a university, user can view the university information and reviews on the university.
 - In the profile page of a student, user can view the student's personal information and send a message.
- By clicking the profile button on the navigation bar, user can see their personal information in their profile page, upload signature.
- By clicking to the DM button on the navigation bar, user will be navigated to chat page in which they can communicate with other user in real-time. In chat, user can select between contacts and send text messages.
- By clicking to the notifications button on the navigation bar, user can view their latest notifications as a popup.
- By clicking to the settings button on the navigation bar, user will be navigated to the settings page. In settings page, user can change their password or E-Mail.
- User can log out using the relevant button on the navigation bar.

3.4 Faculty Member

- Logs in to the application and app navigates to the main page.
- User can see the Todo List on the main page.
- User can navigate to Nomination Requests Page from the main page and see the course nominations made by the students. User can click the review button in that page to approve or reject the request.
- User can search universities and users using the search bar in the main page. When clicked to the result of search, the user is navigated to the profile page of the result.
 - In the profile page of a university, user can view the university information and reviews on the university.
 - In the profile page of a student, user can view the student's personal information, application status, and send a message.
 - In application status page, user can navigate to form pages and approve t

- By clicking the profile button on the navigation bar, user can see their personal information in their profile page, upload signature.
- By clicking to the DM button on the navigation bar, user will be navigated to chat page in which they can communicate with other user in real-time. In chat, user can select between contacts and send text messages.
- By clicking to the notifications button on the navigation bar, user can view their latest notifications as a popup.
- By clicking to the settings button on the navigation bar, user will be navigated to the settings page. In settings page, user can change their password or E-Mail.
- User can log out using the relevant button on the navigation bar.

3.5 Incoming Student

- Logs in to the application and app navigates to the main page.
- User can rate their host university and write comments from the main page.
- User can search universities and users using the search bar in the main page. When clicked to the result of search, the user is navigated to the profile page of the result.
 - In the profile page of a university, user can view the university information and reviews on the university.
 - In the profile page of a student, user can view the student's personal information and send a message.
- By clicking the profile button on the navigation bar, user can see their personal information in their profile page, upload signature, and navigate to application status page.
- By clicking to the DM button on the navigation bar, user will be navigated to chat page in which they can communicate with other user in real-time. In chat, user can select between contacts and send text messages.
- By clicking to the notifications button on the navigation bar, user can view their latest notifications as a popup.

- By clicking to the settings button on the navigation bar, user will be navigated to the settings page. In settings page, user can change their password or E-Mail.
- User can log out using the relevant button on the navigation bar.

4. BUILD INSTRUCTIONS

1. Install node.js.
2. Pull our source code in github/master repository
3. Open the source on you favorite ide.
4. Go to server folder from the command line and use npm i command to build the required node module packages.
5. In 319-Project-Master folder, ((../server) make sure that you can see the server and src folder in 319-Project-Master folder, there might be 2 319-Project-Master folder) use npm i - -force command.
6. In server use npm run dev command to run the server
7. In ../server folder use npm start to start the front-end server
8. You are ready to use the Toera!
9. We use cloud database, hence you don't need to create a new database. We also use postman but you don't need to install postman to run the application

5. WORK ALLOCATION

Utku Kurtulmuş:

Code:

I mainly worked in the back-end part of the project with Mert and Ömer. I implemented the chat feature, implemented and connected its UI and router. Helped creating the database. Implemented the LogIn, ForgetPassword and LogOut features with Ömer. Created the router classes with Ömer and Mert. Implemented the SearchBar and connected its UI with the related router. Connected SearchBar with users which are stored in the database.

Implemented the NavBar with Ömer. Helped with solving problems during connecting the front-end and back-end.

Reports:

Created Sequence, Activity Diagrams with Kutay. Helped others with their parts in both reports. Fixed the issues according to the feedbacks for second iterations for both reports.

Ömer Oktay Gültekin:

Code:

I mainly worked in the back-end part of the project with Mert and Utku. I also fixed some parts in the front-end according to the needs of the back-end. Decided on the properties of model classes with Mert. Implemented the Login, ForgetPassword and Logout features with Utku. Created the database. Created the router classes with Utku. Connected the Form Pages UI with their router classes with Mert. Implemented the NavBar with Utku. Implemented the Application router, connected its UI with the router. Helped other back-end members with their problems. Helped front-end members with their back-end related questions.

Reports:

Created the class diagram with Mert also I wrote and made parts and diagrams that contains most Business Logics in the reports.

Mert Ünlü:

Code:

I mainly worked in the back-end part of the project with Ömer and Utku. Created all the model classes. Worked on router classes of the models. Connected all Profile Pages UI's with the related router classes, connected the Form Pages with related router classes. I helped creating the database for the model instances. Through the end of the project, I updated the model and router classes in the needed way. There were lots of issues during connecting back-end and front-end, I helped with solving those problems.

Reports:

For the analysis report, I created the class diagram with Ömer. For the design report, I created the model classes diagram. Also helped other members with their parts and fixed minor issues according to the feedbacks for both second iteration of the reports. Wrote the final report.

Kutay Şenyiğit**Code:**

I mainly worked in the frontend part of the project with Onurcan and İlker. I implemented Add Course Popup and its related features. I took a role in making connections between backend and frontend. I implemented rendering of the fetched data on tables. Helped teammates when needed. Implemented the user interfaces of the following pages.

1. Learning Agreement Pages
2. Popups Pre-Approval Pages
3. Course Transfer Form Pages
4. Applicants List Page
5. Logs Page
6. Nominate New Course Page

Reports:

Created Sequence, Activity Diagrams with Utku. Helped others with their parts in both reports. Fixed the issues according to the feedback for second iterations for both reports. Also worked on UI designs in the first report. Worked on functional requirements in the design report.

İlker Özgen:**Code:**

I mainly worked on the frontend of the project with Onur and Kutay. Arranged the routing between the pages. Implemented the PDF Conversion and Download feature along with its UI. Added the PDF Feature to form pages. Implemented the UI of the Upload Signature Feature. Made styling, changes on several pages and React components. Implemented the Popup

for Notifications Feature. Implemented a dynamic React table that takes an array as input and construct the table, which is used in several pages. Implemented the Navigation Bar Component, which is used in every page. Implemented Popup Feature for several pages. Implemented Loading Page Animations. Helped the other team members when they needed. Implemented the user interface of the following pages:

1. Application Pages
2. Convert to PDF Pages
3. Forgot Password Page
4. Login Page
5. Profile Pages
6. Reset E-Mail Page
7. Reset Password Confirm Page
8. Reset Password Page

Reports:

For the first iteration of analysis report, I created the use-case diagram, dealt with document formatting, and worked on UI designs. For the second iteration of analysis report, I made the necessary changes according to the feedback, added use-case table with explanations. I helped the others when needed for both iterations.

For the first iteration of design report, I created the Subsystem Decomposition, Object Design, and UI Subsystem Diagrams of UI with Onurcan. For the second iteration of design report, I made the necessary changes to the Subsystem Decomposition, Object Design, and UI Subsystem Diagrams of UI with Onurcan. I dealt with document formatting. I helped the others when needed for both iterations.

Onurcan Ataç:

Code:

I mainly worked on the frontend part of the project with İlker and Kutay. Created the Rating Popup and its mechanism in order to take rating and comments on host universities and courses taken from the students. Also added the rating table to course and university profile pages. Also connected the frontend part of Profile pages and Preapproval Page. Implemented Search Bar on the main page. Created

the table used for university and course lists. Implemented Popup feature for several pages and contributed to the styling of the Navigation Bar. Implemented the frontend of the following pages.

1. University Rating Table
2. Course Rating Table
3. Info Page
4. Pre-Approval Pages
5. Search Bar
6. University List Table
7. University List Page
8. Settings Page

Reports:

For the analysis report, I created the State Diagrams. Also worked on UI design with teammates. For the design report, I created the Subsystem Decomposition, Object Design, and UI Subsystem Diagrams of UI with İlker and Ömer.