Bilkent University

Department of Computer Engineering

Internship Report Management System

Project short-name: Bilport

Final Report

Arda İynem, Atika Zeynep Evmez, Ege Çenberci, Yağız Özkarahan, Zeynep Naz Sevim

Instructor: Eray Tüzün
Teaching Assistant(s): Tolga Özgün, Muhammad Umair Ahmed, Yahya Elnouby

Final Report
May 28, 2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object-Oriented Software Engineering course CS319.

Contents

1. Introduction

Project Bilport aimed to create an easy and reliable internship management system to handle the internship processes of Engineering Faculty undergraduate students at Bilkent University. It was only partially successful in reaching its goal. Bilport was able to achieve a web application with different user interfaces depending on the user's role, upload and download pdf files and offer a database to store these files relieving the registrar's office from its file sorting work. However, some main functionalities, such as filling out student evaluation forms online, were not achieved due to problems encountered during the implementation.

2. Lessons Learned

Throughout the project, we have understood the importance and the difficulty of team coordination and clear communication. We discovered managing meetings without making excessive amounts of it or keeping the meetings time effective was more challenging than one would think. It is safe to say that we all have a newfound appreciation for team managers as we were struggling with proper work distribution and coordination.

Another important takeaway from this project was the importance of requirement analysis. It is only possible to implement some of the features initially thought of in projects restricted by time, such as this one. Even though we had decided not to implement many ideas that we initially thought of in order to have a relatively smaller, more manageable project, there were still features we could not complete on time. Through this experience, we have learned the true importance of making a thorough requirement analysis and rigorous planning.

On the technical side of things, we have learned the importance of extracting only the necessary information from the technologies we have used. Our project mainly utilizes Spring and React technologies, which were new to almost all our

group members. Therefore, while building this project, we practiced finding the necessary technical insight among an endless sea of manuals, tutorials, and questions asked by others; to complete the task at hand.

3. User’s Guide

3.1. Student

The student users can access the website after their accounts have been created by the super-admin user with the username and password that they receive through email. Upon login, the student user can view their uploaded reports in the Report List tab and submit a new report on the Submit Report tab. They can view their information by pressing the “Profile” button from the side menu, which displays their full name, student ID, and email. The Change Password section of the Profile tab is not functional at the time and should be ignored. The student can navigate through the side menu to the Evaluator/TA Info tab, where their evaluator’s and TA’s full name and email are visible. The Supervisor Info button of the side menu leads the user to the Supervisor Information tab, where the student's supervisor for either internship can be seen. If the supervisor's information has not yet been entered into the system by the student, it can be done so by using the "Set Supervisor Information" button at the bottom of the page. In the screen that pops up, after entering and submitting the required information, an informative email will be sent to the student's supervisor. Finally, the Log Out button on the side menu can be pressed to log the user out of the system.

3.2. Evaluator

The evaluator users can access the website after the initialization of their accounts with the username and password that they receive through email. Upon login, the evaluators face the assigned students window, where they can see the

students they are responsible for grading. The evaluators can sort the student list by their name in ascending or descending order. They can also search for a student using the search bar at the top of the page. The evaluators can click on a student, which will direct them to a tab that displays information about the student's coursework, like report iteration count, and helpful information like the student's email and supervisor id. When the evaluator changes to the Report tab from the student info tab, they can see the three parts of the internship evaluation form (A, B, and C). Through the Part A tab of the Report tab, the evaluator can fill out the internship form's first part. Through the Part B tab, the evaluator can download the student's report(s), upload the document containing their feedback, and set a due date for the next iteration of the report. The evaluator can fill out the internship form's last part through the Part C tab and submit their choices. After clicking on the "Profile" button on the side menu, the evaluator can view their information. The change password section of this screen is not functional and should be ignored. When the "Course TA Info" button is clicked from the side menu, information on the TAs of the courses CS299 and CS399 are displayed if applicable. Finally, the evaluator can log out via the "Log out" button on the side menu.

3.3. Teacher’s Assistant

The TA users can access the website after the initialization of their accounts with the username and password that they receive through email. Upon login, they will be greeted with the Students tab, where they can view the students that are assigned to them, similarly to the evaluators. When a student from the list is clicked upon, a screen displaying the student's ID, email, and report status will be displayed. The TA user can download the report submitted by the student from the "Download Student's Report" button if applicable and can upload their feedback through the dedicated section at the bottom of the screen. Similarly to other users, the TA user can click the "Profile" button from the side menu to view their full name, email, ID, and assigned course. The change password section of this screen is not functional and should be ignored. Finally, the TA can log out via the "Log out" button on the side menu.

3.4. Supervisor

The supervisor users can access the website after the initialization of their accounts with the username and password they receive through email. Upon logging in, the supervisor user views the Confidential Summer Training Form page. On this page, the supervisor can check the appropriate boxes for the displayed questions about their intern's performance. They can and must also give their intern's internship's starting and ending dates through the interface, upon which the total duration of the training is calculated. At the bottom of this screen, the supervisor user can give a general evaluation grade to the trainee and add additional comments if desired. Similarly to all the other users, the supervisor can view their information via the "Profile" button on the side menu. The change password section of this screen is not functional and should be ignored. Lastly, the "Log Out" button can be pressed on the side menu to log out from the system.

3.5. Admin

The admin users have to be created by the super admin user or the developers. Upon login, the admin users face the Students screen of the Management tab. Through this screen, the admins can add a new student to the system via the "Add a new Student" button. The admins can also view a list of students on this screen. Through this student list, they can easily view whether or not the students are assigned to evaluators. They can click on any student from the list, which will direct the admin user to a screen that displays the student's general information and an "Assign student to evaluator" button. Upon clicking the "Assign student to evaluator" button, the admin can assign/change a student's evaluator through the "Choose and evaluator" section that drops down. On the Management tab, the admins can change the screen to the evaluators list screen by clicking the "Evaluators" button beside the "Students" button at the top of the screen. Here, the admin user has a list view of the existing evaluators with an "Add new evaluator" button below. Upon clicking this "Add new evaluator" button, the admin can add a new evaluator to the system using the section that appears. When the admin clicks

on any evaluator from the list, a new section titled "Evaluator Info" appears, displaying the evaluator's general information, such as name, email, ID, student limit, and assigned student count. In the "Evaluator Info" section, the admin can use the "Change Student Limit" section to change the selected evaluator's student limit to assign them more or fewer students. Moreover, the admin user can click the "Profile" button from the side menu to view their information. The change password section on this screen is not functional and should be ignored. Finally, the admin can log out after clicking the "Log Out" button from the side menu.

3.6. SuperAdmin

The user information of the superadmin is hard-coded to the system. Superadmin can enter their account after the login information is provided by the developers. Superadmins can create student, evaluator and teaching assistant accounts one-by-one by entering the necessary information specified on the webpage. Also, superadmin can import excel files to create student and evaluator accounts. The file selection on the page accepts .xlsx and .xls files. There should be no empty cells in the files. The format of the excel file for students should be like below:

Id	Name	Surname	Email	Courses Taken
22006789	Kemal	Kılıçdar	kemal.kk@ug.bilkent.edu.tr	CS299,CS399
21308768	Servet	Sayar	servet.sayar@ug.bilkent.edu.tr	CS399
21903845	Oğuz	Akrep	oguz.akrep@ug.bilkent.edu.tr	CS299

Table 1: Student Input File Format Example

Cells' types should be plain text. If both CS299 and CS399 courses are taken, they should be separated with a comma and there should not be any blank space before and after the comma. After the file selection, the students will be created in the database and their account information will be sent to entered emails by clicking

on the import button. The format of the excel file for students should be like the table below:

Id	Name	Surname	Email	Student Limit
1211	Levent	Seymen	kaptan@bilkent.edu.tr	1
1013	Caroline	Akarsu	caroline@bilkent.edu.tr	39
9345	Hürrem	Sultan	hürrem@bilkent.edu.tr	41

Table 2: Evaluator Input File Format Example

The type of the student limit column should be integer. The other cells' types should be plain text. After the file selection, the evaluators will be created in the database and their account information will be sent to entered emails by clicking on the import button.

4. Build Instructions

The Backend and Frontend folders of our project are easily available on our GitHub page.

Backend:

Prerequisite: This is a Java project, therefore you need to have Java (JRE) installed on your device. Project runs on Java 17, but we have also tested it on Java 11 and it works without any problems.

1. Open the project directory "Backend" on your preferred IDE.
2. Run the DemoApplication.java file from your IDE. It is located in src/main/java/com.bilport.demo/

Frontend:

Prerequisite: The frontend of this project runs on React, so you need to have Node.js installed on your device to run the application.

1. Open the project directory “bilport-webapp” on your preferred IDE. This directory is located inside our main “Frontend” folder.
2. Open the terminal. Current directory of the terminal should be looking like “.../bilport-webapp”
3. Type “npm install” on the terminal and enter. This should install the dependencies of the project.
4. After all the dependencies are installed, type “npm run dev” on the terminal and enter. This should build and run the application.
5. If all the previous steps are successful, the console should display the link of the website (localhost:5173/). Ctrl+click on this link to go to the webpage.

Once both the backend and frontend build steps are completed, you should be able to navigate our website without any problem.

5. Work Allocation

Arda İynem

Analysis Report:

- Participated and helped to make state, activity and use case diagrams.
- Prepared the mock-ups of UI from scratch.

Design Report

- Written the access and security parts as the one who implemented it. Summarized the Java Spring Security architecture and JWT system for authentication and authorization.

Final Report

- Written the work allocation part for my own participation.

Implementation

- Learned and Implemented the Spring Boot Architecture and REST API Architecture and started the project.
- Learned and Implemented Spring Security for authentication and authorization system.
- Learned and Implemented the JWT Token & Cookie system for token based time-expired authentication with refresh cookies and access cookies.
- Learned and Implemented the React hooks and React Route and axios for role based authentication and authorization and persisted login.
- Implemented the class diagram on a project within REST API architecture rules.
- I have participated as a full-stack coder and provided the connection with the frontend while implementing the backend.
    - **Backend:**
        - Configured Spring Boot, Spring Security and Spring Web and Spring Data and implemented related configuration classes and Filtret (Custom Spring Boot Filters for authentication and authorization) classes.
        - Implemented the JWT token logic both in the back-end and in front-end
        - Created and implemented Repository Classes for the objects in the class diagram from scratch. Implemented the required query statements and methods for the connection between database layer and service layer.
        - Created and implemented Service Classes for the objects in the class diagram from scratch. Implemented the required business logic after the queries and prepared the related data to be served in the controller layer.
        - Created and implemented Controller Classes for the objects in the class diagram from scratch. Implemented the endpoints for the connection between the controller layer and the frontend.

        - Created and implemented DTO (Data Transfer Object) and Model Classes for the objects in the class diagram from scratch.
    - **Frontend:**
        - Implemented the role based authentication and authorization logic with React Route, axios and JWT access tokens.
        - Implemented login persistence logic with React Route and cookies for refresh tokens.
        - Implemented the whole connection with the backend (Get and Post requests for Reports, User related data, and pdf upload & download) with axios fetch asynchronous functions.
        - Implemented JWT refresh and access token logic.
        - Implemented many custom hooks and contexts in React for auth, login, persistence and logout functionalities. Implemented logic with useState, useEffect hooks and custom axios components with interceptors for fetch operations.

Atika Zeynep Evmez

Analysis Report:

- **Use Case Diagram:** Prepared the use case diagram for the first iteration from scratch. Edited it on the second iteration.
- **Mock-ups:** Recreated the UI mock-ups for the second iteration from scratch.
- **Functional Requirements:** Wrote functional requirements but it is understood that it should not be in the analysis report, so removed it on second iteration.
- **State Diagrams:** Took part in preparing the student state diagram (created during video call together with the group) (First iteration).

Recreated report state diagram, edited student state diagram (second iteration).

Design Report

- **Subsystem Decomposition Diagram:** Prepared and edited subsystem decomposition diagram for first and second iterations.

Final Report

- Wrote my work allocation part.

Implementation

- In the implementation, I've mostly worked on endpoints and React hooks, in other words, I have made connections between database and user interface for users that are specified below.
- Supervisor:
    - Created supervisor endpoints (related controller, model, repository, and service classes) in backend and made connection with the related pages on frontend.
    - I connected "Student Supervisor Info Page" to supervisor endpoints and let the student create a supervisor and also change their info.
    - Created SupervisorForm object for the "Summer Training Evaluation Form", and created its endpoints. Created the connections with "Supervisor Form Page" and enabled the supervisor to fill the training form and store its data on the database.
- Ta:
    - Created TA endpoints (related controller, model, repository, and service classes) in backend and made connection with the related pages on frontend.
- Evaluator:
    - Created evaluator endpoints (related controller, model, repository, and service classes) in backend and made connection with the related pages on frontend.

    - Created EvaluatorForm object for the “Summer Training Evaluation Form”, and created its endpoints. Created the connections with “Evaluation Form Page” and enabled the evaluator to fill the evaluation form and store its data on the database.
- Admin:
    - Created admin endpoints (related controller, model, repository, and service classes) in backend and made connection with the related pages on frontend.
- Superadmin:
    - Created superadmin endpoints (related controller, model, repository, and service classes) in backend and made connection with the related pages on frontend.

Ege Çenberci

Analysis Report:

First Iteration:

- I have written the first two sections of the report
- I have contributed while creating the activity and state diagrams and led the group, as we got easily distracted on the voice call

Second Iteration:

- I have updated the first two sections that I was responsible of according to the feedback
- Created the old system activity diagram
- Created the class diagram from scratch as the first one was heavily lacking
- Formatted the entire document (indentation, page numbers, figure naming, proper spacing, and minor tweaks to language on sections I wasn’t originally responsible for)

Design Report

First Iteration:

- I have written every section excluding sections 2.1 - 2.4.
- Created the object design diagram
- Formatted the entire document (indentation, page numbers, figure naming, proper spacing, and minor tweaks to language on sections I wasn't originally responsible for)

Second Iteration:

- Updated all the sections I was responsible for according to the feedback received (all sections excluding 2.1 - 2.4)
- Updated the object design diagram
- Added references sent by my group member to section 2.3
- Added the calculations to section 2.2
- Formatted document where changes were needed (indentation, figure/table naming, proper spacing, and minor tweaks to language on sections I wasn't originally responsible for)

Final Report

- I have created the document and provided a template for section 5
- I have written the sections through 1 - 3.5
- I have formatted the entire document (indentation, page numbers, figure naming, proper spacing, and minor tweaks to language on sections I wasn't originally responsible for)
- Wrote the work I was responsible for on section 5

Implementation

- I have created the MailController service to provide mail sending utility
- Created the MailConfig class to configure the JavaMailSender as a Spring bean for the MailController service
- Implemented the MailController at proper locations to send emails to the newly created accounts' email addresses with the correct format

- Created the PasswordGenerator class to generate random passwords for users during account initialization

Yağız Özkarahan

Analysis Report:

- I have written the non-functional requirements and pseudo-requirements parts of the report.
- We have made the system models (the diagrams) together with the entire group, on video calls. I took an active part in it, particularly in making the activity and state diagrams.

Design Report

- I have written the hardware/software mapping and design goals parts of the report.

Final Report

- I have written the build instructions part of the report, and I have also written my own work allocation part.

Implementation

- I was responsible for the entire frontend design of the project. Using Javascript with React and Bootstrap, I have written all the pages for all of the users, and made them responsive and interactive with temporary placeholder values.
- After the frontend design was done, I also took part in making the backend-frontend connection work. My other teammates were responsible for setting up the endpoints and getting/posting data between backend and frontend and I was mainly responsible for improving this process through fixing the bugs and making the website look more responsive (such as displaying loading screens in some particularly slow areas, refreshing some pages when new data is

received, giving warning or success messages when particular actions were done and particular buttons were pressed, etc.)

- I was also responsible for handling the implementation of automatic page navigation (when the student user enters, they are directed to the student page; when the evaluator user enters, they are directed to the evaluator page etc.)

Zeynep Naz Sevim

Analysis Report:

- I wrote the functional requirements (use-case scenarios) for the first iteration and changed the use-case diagram for the second iteration.
- As Yağız said we made the system models (the diagrams) together with the entire group, on video calls. I took an active part in it for all diagrams.

Design Report

- I wrote the persistent data management section.

Final Report

- I wrote the user's guide of the super admin and my own work allocation.

Implementation

- I implemented excel import functionality in the super admin page for students and evaluators. I made changes on the frontend and implemented the backend of the functionality. I connected those two.