BTO

Final Report

CS319 Section 1

Sami Bora Akoğuz 22202184
Ege Ertem 22202433
Can Kütükoğlu 22202619
Kerem Cindaruk 22201907
Ertuğrul Malkoç 22102737

23 December 2024

Contents:

Introduction

The final version of BTO System offers a highly-functional and easy-to-use experience for both BTO members as well as external school representatives.

The process of requesting tours from Bilkent is very easy as only a form is needed to be filled. The system assigns a custom code for the applicant which can be used to track the progress of the application. The applicant can view via the custom code which state their application is in. An Email notification is also sent regarding any updates to the application.

The internal process of assigning guides & managing tours is also very easy. The Advisors accept the incoming tour requests via internal dashboards. The conflicts are displayed on the dashboard and advisors can easily resolve them by choosing which ones to move forward with. The guides can apply to these tours and advisors can accept the guide applications to the tours. When a guide is assigned, the tour is confirmed and an Email is sent to all parties. Advisors can swap the guides of the tour by viewing available guides at the desired date. At any time, both advisor and tour applicant can cancel the tour. Individual tours follow the same flow as well.

Inviting Bilkent to fairs is also very straightforward. The school submits a request and again receives a custom code alongside the emails to track the process. The Coordinator is responsible for fair management and can assign Guides to these fairs. Every step can be tracked via emails.

Data is very important in our system. The tours have a Survey assigned to them which can be accessed via custom Survey codes from the main page. The guides may share the quiz code for the tour at the end and the attendant can fill out the form. Every data is graphed in a meaningful manner in Data Panels which can be viewed by Admin and Coordinator.

Work hours of the BTO members are also tracked in the system. Users can log their work hours and Admins/Coordinators can view the work hours to make necessary payments.

Member registration is also integrated into the system. Any Bilkent student can apply to be a member. Coordinator can view these applications and decide to accept them into the system. An Email is sent accordingly.

Advisors can also be assigned to certain days which can be viewed by Guides to have a last resort contact.

Learning Experience

We, as a group, learned a lot from this term-long project, we learned a lot of technical knowledge about software engineering, improved our analytical thinking skills, learned how to comply with customer requests and requirements with a real-life example and we learned how to be and work as a team.

At first, we had a presentation from our customer (BTO) explaining what they wanted from this project. We took notes and discussed what we needed to do, what we could use, what was necessary and how we could form the system depending on BTO's requirements. This discussion and brainstorming process was essential to form the baseline for the project, as it made us have a rough idea on how we were going to approach this project moving forward.

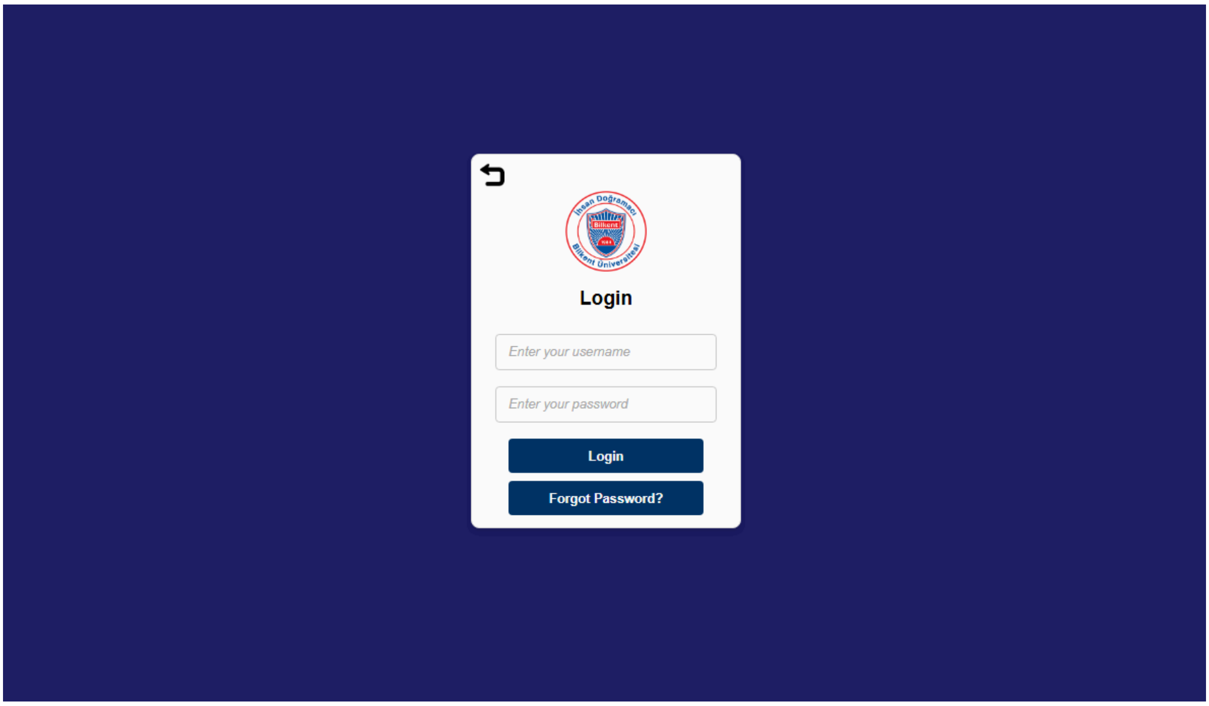
We did research and decided on how to best implement the code using what technologies according to the needs. We decided on the tech stack and how to use them. We made diagrams and schematics to fully realize the complete program and its workflow. As we progressed with the project, the need for these diagrams became clearer. Every time we worked on a diagram, we had a more clear understanding on how we were going to implement the necessary functionalities and which actor was going to have which functionalities. Moreover, while implementing the code, the diagrams helped us while writing the code and how classes interact with each other clearly. Also we could break down the functionalities to implement so that we could assign a portion to each of us without intervening or interrupting each other.

This project was an opportunity to experience real life on a smaller scale. We learned the process of finding a balance between what we want to deliver and what the customer wants and can be satisfied with. We experienced what it is like to work as a group, as much as it lifts a weight and catalyses the developing process, it is also a challenge. We had disagreements on how to implement things, and what we should use to implement them, and also we had times where our codes did not work well together, which is a challenge to overcome and is part of team experience. Even though this project was somewhat of a small scale project, this is one of the first steps going forward for us. When we work on grand-scale projects in the future, this project will be one of the stepping stones on how we got there.

User’s Guide

BTO Member

BTO Member is the general name of people who are part of BTO. This consists of Guide, Candidate Guide, Advisor, Coordinator and Admin. These actors have user accounts and they access the BTO system via the login page. They can reset their passwords in case they forgot it. All of the users have panels they can access accordingly to their role. All users can log their work hours which can be viewed by Coordinators.

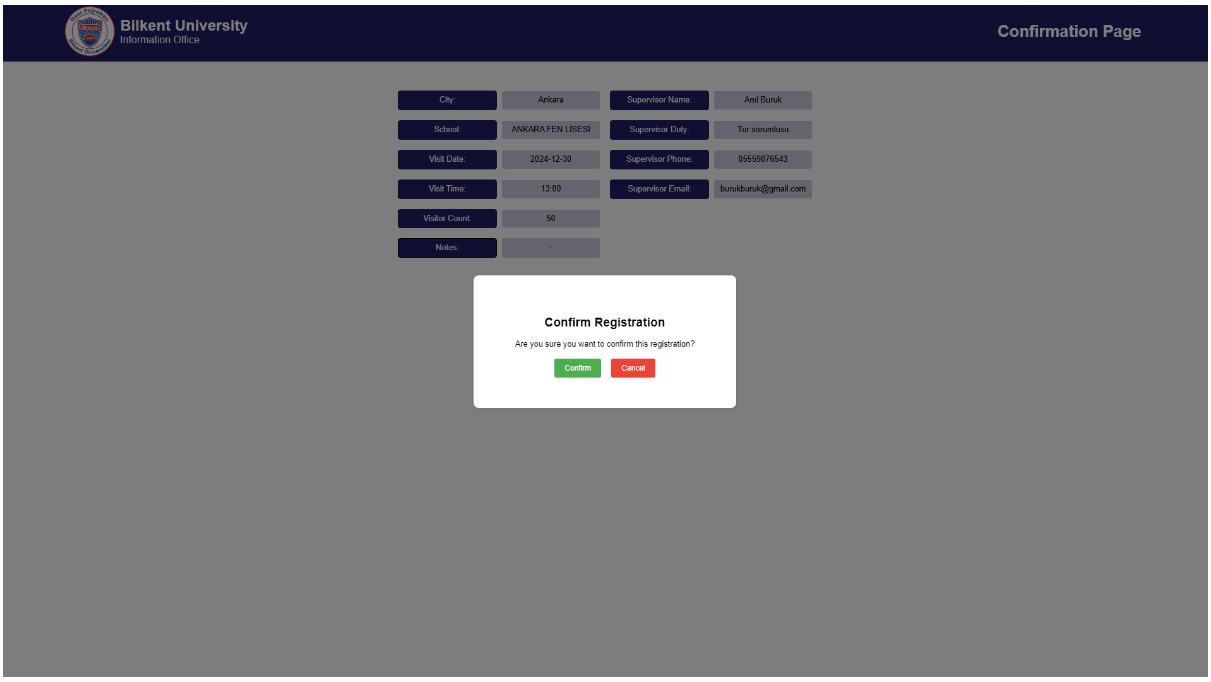


High School Representative

High School Representatives can invite BTO to their high school for a fair by filling up the fair request form. All the required parts should be filled. The answer to the request will be answered to the mail address provided in the form.

Head of a Tour

Head of a tour, who are organizing tours to Bilkent from their high school, can request a tour at the desired day, in the time slots 9 am., 11 am., 1.30 pm. or 4pm by filling up the tour request form. All the requested parts should be filled. The progress of the request will be sent to the mail address provided in the form. The tour request may be cancelled after getting accepted, in which case Head of a Tour will be requested to select a new date and/or time slot from the system. The tour is expected to arrive at time as the schedule can be tight at times. Guide/s will meet the tour and at the end of the tour students will be given a code to join a survey to express their opinions about the tour.



City:	Ankara	Supervisor Name:	Anıl Buruk
School:	ANKARA FEN LİSESİ	Supervisor Duty:	Tur sorumlusu
Visit Date:	2024-12-30	Supervisor Phone:	05559876543
Visit Time:	13:00	Supervisor Email:	burukburuk@gmail.com
Visitor Count:	50		
Notes:	-		

Confirm Registration

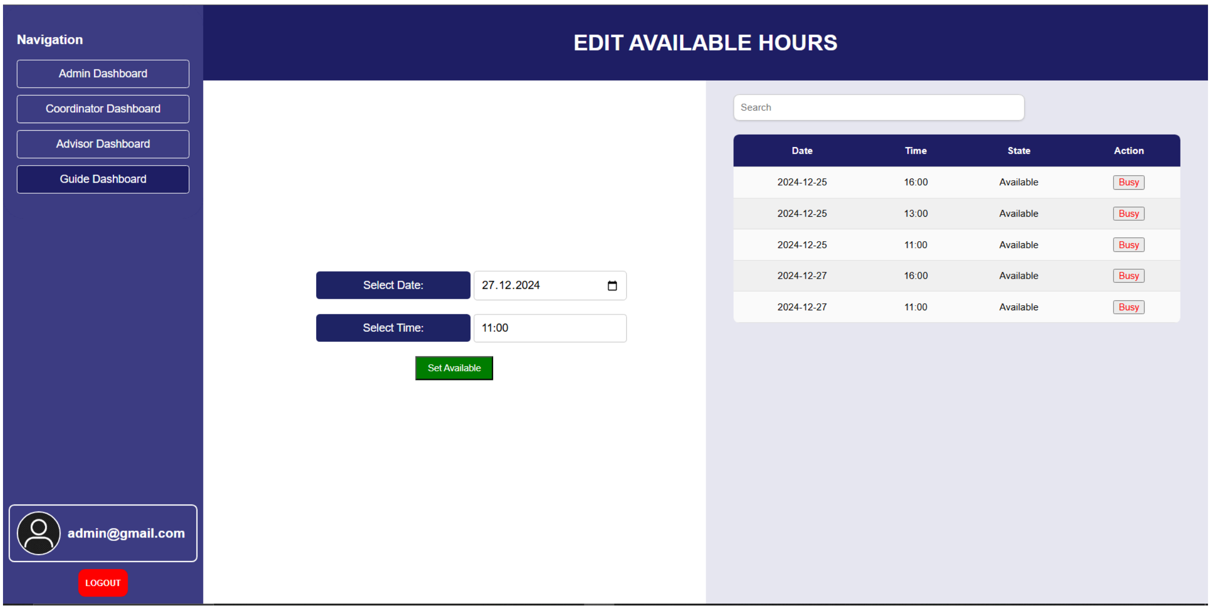
Are you sure you want to confirm this registration?

Bilkent Student

Bilkent students can apply to join BTO as a guide by filling up the BTO member request form. All required parts should be filled. The answer to the request will be answered to the mail address provided in the form. If the coordinator decides to accept them to BTO, a random generated password will be mailed to their email which they can use to access the system.

Guide

Guides are BTO members with access to the guide panel. They can enter and edit their available hours, view account information and view tours assigned to them. Guides can list all available tours according to their available hours, and they can apply to be a guide on that tour. If this request is accepted, this will be shown in the Assigned Tours section. In this section, Guides can see the details of the tour and the school of the tour. Guides can remove themselves from the tour, in which case another guide can fill up this role. Guides have to show up prior to the tour's time and greet the tour. At the end of the tour, they can start the survey assigned to this tour.



EDIT AVAILABLE HOURS

Select Date: 27.12.2024

Select Time: 11:00

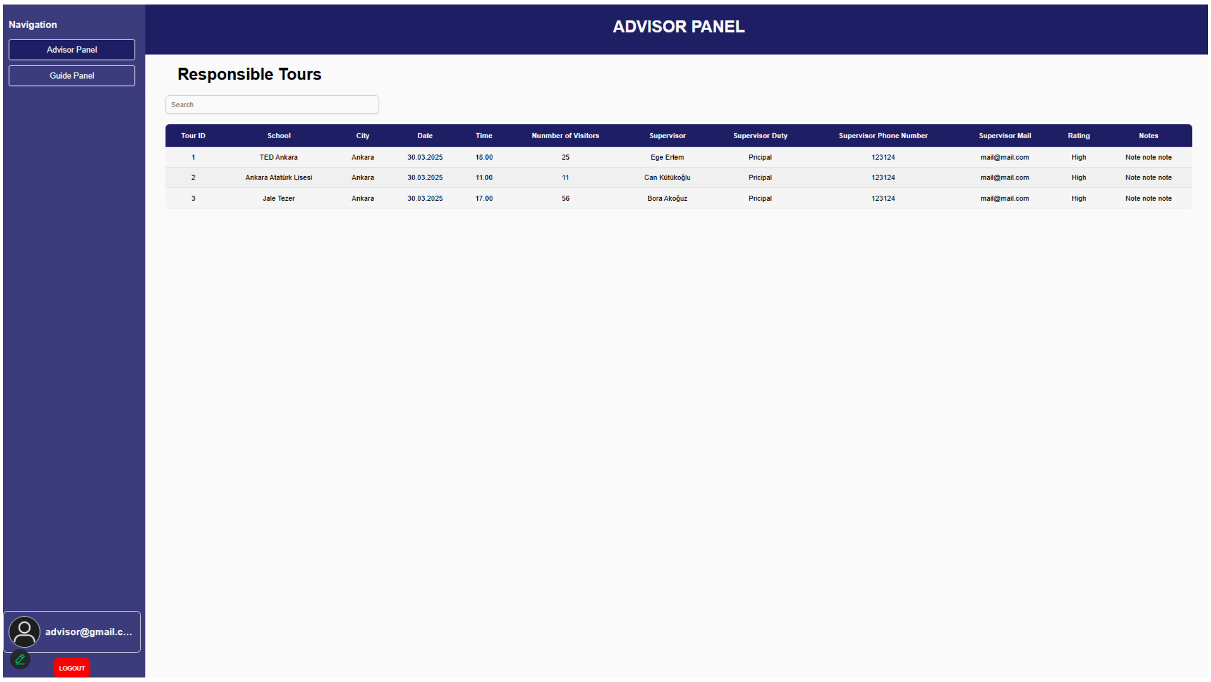
Date	Time	State	Action
2024-12-25	16:00	Available	Busy
2024-12-25	13:00	Available	Busy
2024-12-25	11:00	Available	Busy
2024-12-27	16:00	Available	Busy
2024-12-27	11:00	Available	Busy

Candidate Guide

New BTO Guides need to gain some experience in order to become a guide. This process requires Guides to join tours and fairs a number of times. Candidate Guides are BTO members with access to the candidate guide panel. They can view their account information, list all available tours, check the details of the tour, and register to any of them. They can list the tours they joined, and they can remove themselves from the tour. Candidate Guides do not have to wait for tour request approval.

Advisor

Advisors are BTO members with access to the advisor panel. They can view their account information. Advisors manage tours. They can accept or decline tour requests, fair requests, guides' tour registration requests. Advisors can List tours, view detailed information about the tours, view/add/remove/change the guides assigned to the tour. They can also accept alternative tours instead of registered tours, which will cancel the priorly accepted tour. Advisors can also make exceptions, and accept more tours than previously set constraints. They can also cancel tours without replacement. Accepting, declining and cancelling a tour will send an Email to the head of the tour.



ADVISOR PANEL

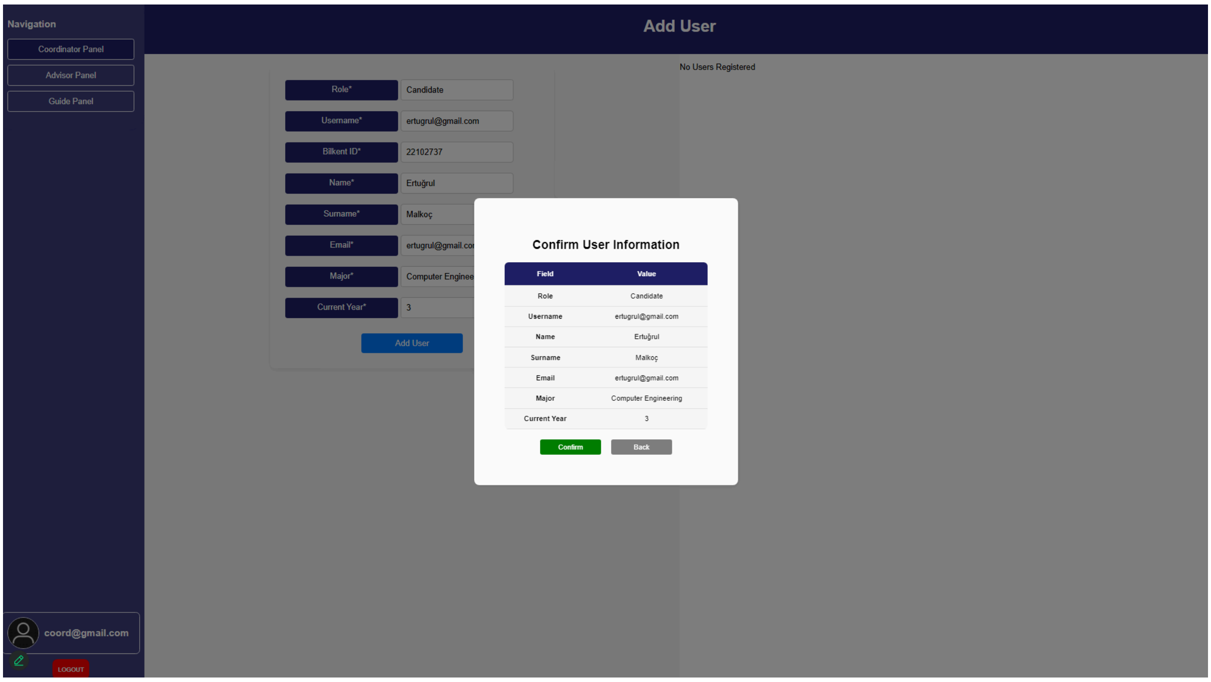
Responsible Tours

Search

Tour ID	School	City	Date	Time	Nunmber of Visitors	Supervisor	Supervisor Duty	Supervisor Phone Number	Supervisor Mail	Rating	Notes
1	TED Ankara	Ankara	30.03.2025	18.00	25	Ege Ertem	Pricipal	123124	mail@mail.com	High	Note note note
2	Ankara Atatürk Lisesi	Ankara	30.03.2025	11.00	11	Can Kütükoğlu	Pricipal	123124	mail@mail.com	High	Note note note
3	Jale Tezer	Ankara	30.03.2025	17.00	56	Bora Akoğuz	Pricipal	123124	mail@mail.com	High	Note note note

Coordinator

Coordinator is a BTO member with access to the coordinator panel. They can view the account information of other users. Coordinators manage the fairs and the user system. They can accept/decline fairs and assign guides to these fairs. And they can manage users of the system. Coordinators can register candidate guides, guides and advisors to the system. They can search for a user or list all users and select one of them to remove them from the system. They can also upgrade candidate guides to guides. Coordinators can also access the Advisor panel, and use all the functionality of an Advisor as well as all the functionality of the Guides.



Add User

No Users Registered

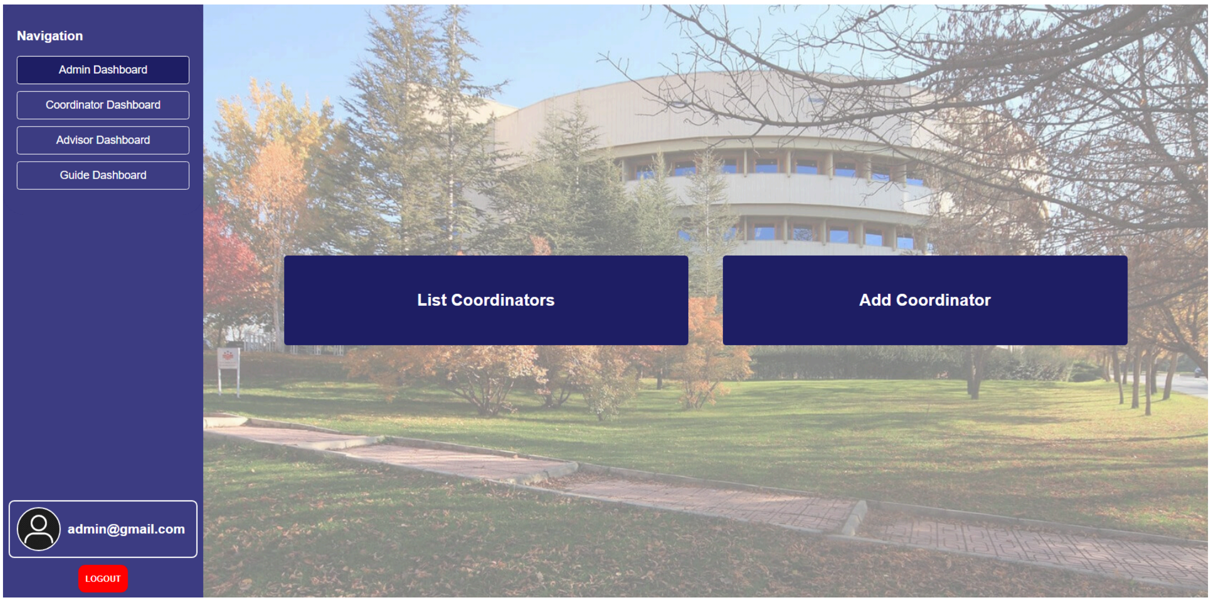
Role*	Candidate
Username*	ertugrul@gmail.com
Bilkent ID*	22102737
Name*	Ertuğrul
Surname*	Malkoç
Email*	ertugrul@gmail.co
Major*	Computer Enginee
Current Year*	3

Confirm User Information

Field	Value
Role	Candidate
Username	ertugrul@gmail.com
Name	Ertuğrul
Surname	Malkoç
Email	ertugrul@gmail.com
Major	Computer Engineering
Current Year	3

Admin

Admin is a BTO member with access to the admin panel. They can view their account information of all users including other Admins and Coordinators. They can register a new coordinator to the system, list all coordinators in the system and delete coordinators from the system. They can also access Coordinator and Advisor's panels, and use all of their functionalities.



Build Instructions

1) Docker Desktop and Docker Compose need to be installed in the machine. If that’s not the case, install Docker Desktop from the link provided below:

https://www.docker.com/products/docker-desktop/

Installing Docker Desktop will automatically install Docker Compose.

2) Clone the repository “https://github.com/BilkentBTO/BTO.git”.

```
git clone https://github.com/BilkentBTO/BTO.git
```

3) Navigate to the directory “BTO/App”

```
cd ./BTO/App
```

4) Run Docker Compose while the Docker Engine is running (have Docker Desktop running in the background). Ensure that no other containers are running as they can block the ports used: 5173 & 8080.

```
docker compose up
```

Or in the development process, use “compose watch” to enable hotloading.

```
docker compose watch
```

5) Access the localhost port printed on the terminal. Example output:

```
frontend-1  |   VITE v6.0.3  ready in 159 ms
frontend-1  |
frontend-1  |   ➜  Local:   http://localhost:5173/
frontend-1  |   ➜  Network: http://172.18.0.3:5173/
```

The Project can be viewed via the link given in Local. The default is: http://localhost:5173/

Implementation and Reasoning

Tech Stack

For this projects' tech stack we used ASP .NET, React.js, PostgreSQL, Docker and GitHub. There were many choices and combinations we could do, but it all came down to how comfortable we were implementing the code. ASP .NET is not the number one choice when it comes to backend but C# is very reliable and can be very useful for maintainability. And it is used for grand scale projects in real life use cases as well. C#'s syntax is also similar to Java or C++, which we learned in first and second year of CS classes respectively. Unlike ASP .NET, React.js is very popular, it is very useful and flexible. Because of this popularity, there are many documentations and tutorials about React.js, which made it easier to debug any problems and enhance and speed up the development process. As for PostgreSQL, the reason why we preferred it was because it was useful, secure and reliable. For a data-heavy project like this, it was important to not face many problems with the database so that we can focus on other parts. With this many moving parts in the project, it could be hard to connect all of them easily. Which is why we used Docker. Docker is a useful tool as it offers a common machine and environment to work on for the developers of the team. This eliminates the dependency, version conflict, operating system and portability problems. And the last part of the tech stack is Github, which is a necessity in a group project. It is a tool that handles version control and allows for developers to develop a program together without interrupting each other, and if they do, it is easy to take it back. All of these combined provided a very powerful development environment and also seamless workflow.

Not Implemented

We did not implement the Messaging System we showed in the Use Case Diagram as we did not think it was necessary at the later stages. The Messaging System was a system that enabled text communication between BTO members. This was not really necessary as the Email system already automatically sends crucial information. People will not check their BTO account's messages regularly, and similar to how nobody uses Moodle's messaging system, if anyone wants to communicate with each other, they can use Emails. Everyone can access each other's Email account. For that, we did not want to put resources into implementing such a system when we can focus on other parts of the project that are more important, more necessary and need more time.

Email / SMS System

We used MailJet as our Email provider, as not only its API had C# wrapper we could use in the backend, but also it allowed for 200 free Email’s to test out our Email system. We wanted to implement both the Email system as well as the SMS system, however after searching for a while, we could not find an SMS sender that provides free SMS that works. We figured it is not a problem as Bilkent already has an Email system and an SMS system. BTO can use Bilkent’s system with slight changes to the Email System code. They can move forward with MailJet API if they prefer, but using MailJet API is mostly for the testing purposes and it is not a necessity. The Email functionality is currently paused due to the limit of the free plan. All the infrastructure for it is ready and functional as we have tested it in our development process.

Work Allocation

Sami Bora Akoğuz - 22202184

Deliverable 1:

- Writing Tech Stack
- Determining Use Cases
- Drawing Use Case Diagram

Deliverable 2:

- Determining Classes
- Drawing Class Diagram
- Determining Relations
- Drawing Sequence Diagram
- Writing the report

Deliverable 3:

- Drawing Subsystem Decomposition Diagram
- Determining relations between subsystems
- Writing the report

Deliverable 4:

- Drawing Detailed Class Diagram
- Determining extra relations
- Writing the report

Implementation:

- Implemented backend of Login, User System, Scheduling System, Survey System, Tour Registration System, Fair Registration System, Individual Tour Registration System, Tour Accept/Reject & Tour Apply/Dismiss System, Error Handling, Docker Containers and their relations, Database setup, Scraping School Data.

Final Report:

- Writing work allocation
- Changing introduction

Ege Ertem - 22202433

Deliverable 1:

- Determining Use Cases
- Drawing Use Case Diagram
- Writing Textual Descriptions

Deliverable 2:

- Drawing Mock-ups
- Writing report

Deliverable 3:

- Determining Relations between subsystems
- Writing report

Deliverable 4:

- Determining design patterns
- Writing report

Implementation:

- Implemented frontend of Login, Tour Registration form, Fair Registration form, Scheduled Tours, Change Assigned Guide panel, Join BTO registration page, Assign Guide to Fairs/Tours pages, List All Users page, User profile, Left sidebar, Added the Individual Tour option to all view tour pages.

Final Report:

- Writing work load
- Checking the information

Can Kütükoğlu - 22202619

Deliverable 1:

- Determining Use Cases
- Drawing Use Case Diagram
- Writing Textual Descriptions

Deliverable 2:

- Drawing User Page Mock-ups
- Writing the report

Deliverable 3:

- Determining Relations between subsystems
- Drawing the diagram
- Writing report

Deliverable 4:

- Determining relations between classes
- Writing report

Implementation:

- Implemented Frontend of Main page, Data panel, Survey System, Tour Registration View/Cancel pages, View Work Hours page, Edit Available Hours page, List All Users page, Navigation between all UI pages, View Advisor Info Page, Overall user experience quality of the pages, making each page look better and more appealing to the eye.

Final Report:

- Writing my work allocations part
- Overlooking the screenshots and structure of report.

Kerem Cindaruk - 22201907

Deliverable 1:

- Contributing to Use-Case Diagram
- Writing Use-Case Diagram Textual Descriptions for actors; Admin, Candidate Guide
- Writing the descriptions of Tech Stack; ASP .NET, PostgreSQL

Deliverable 2:

- Contributing to Non-Functional Requirements and Sequence Diagram
- Making the Class Diagram

Deliverable 3:

- Writing SubSystem Decomposition Textual Descriptions

Implementation:

- Writing Classes, database controllers and API's for Tour, Fair and TimeBlock
- Creating the Constraints Class, and Some Other Minor Classes
- Collecting The Data of Distances Between Ankara and Other Provinces
- Setting up MailJet API and Implementing the Mail System
- General Contribution and Improvements to Other Classes

Final Report:

- Writing the Parts; Contents, Introduction, Learning Experience, Implementation and Reasoning
- Writing My Own Part of Work Allocation
- Adding a demo video

Ertuğrul Malkoç - 22102737

Deliverable 1:

- Drawing the Use Case Diagram
- Determining the use cases
- Analyzing the use cases
- Checking the report

Deliverable 2:

- Drawing the Activity and State Diagrams
- Analyzing the states and activities in the system
- Writing the report

Deliverable 3:

- Drawing the Subsystem Decomposition Diagram
- Analyzing and determining the relations between subsystems
- Writing the report

Deliverable 4:

- Drawing the Detailed Class Diagram
- Analyzing the relations between classes
- Checking the report

Implementation:

- Working on the backend of the Survey System, Quiz System, and Visitor System
- Adding descriptions to files, classes, and methods
- Making minor changes to other classes

Final Report:

- Writing the work allocation section
- Taking screenshots from the program
- Checking the report
- Adding a demo video