INTERNATIONAL BURCH UNIVERSITY
FACULTY OF ENGINEERING AND NATURAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGIES

CEN 334 SOFTWARE ENGINEERING
PROJECT DOCUMENTATION

# EMPLOYEE TRACKER

Prepared by:
Faris Bektaš &
Amina Mehić

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching assistant

June 6, 2021

# Table of contents

# 1. Introduction

## 1.1 About the project

The application consists of 2 parts: a list of Employees and records for the presence of employees at daily scrum meetings. The list of employees contains the ability to add, delete, update users and search. Records for daily scrum meetings contain a table that records the time employees have arrived for the meeting. The user has the ability to add, modify, delete and search as well.

## 1.2 Project functionalities and screenshots

List of main features:

Since the functions of our basic features are surely clear by names, we are not going to explain them in detail, but only to list them:

- Add user
- Search
- Delete user
- Edit user

Also, on Scrum Page:

- Add arrivals for specific user
- Delete arrivals for specific user
- Edit arrivals for specific user
- Search

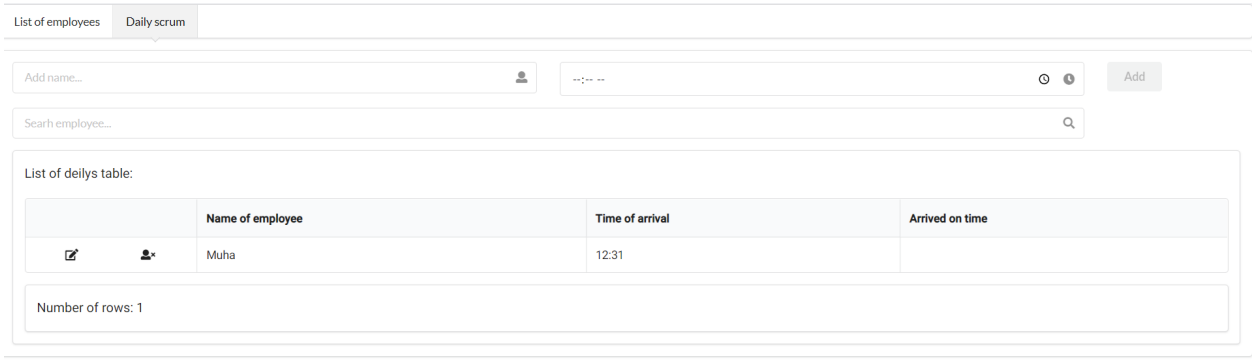


Fig. 1: Employees Page

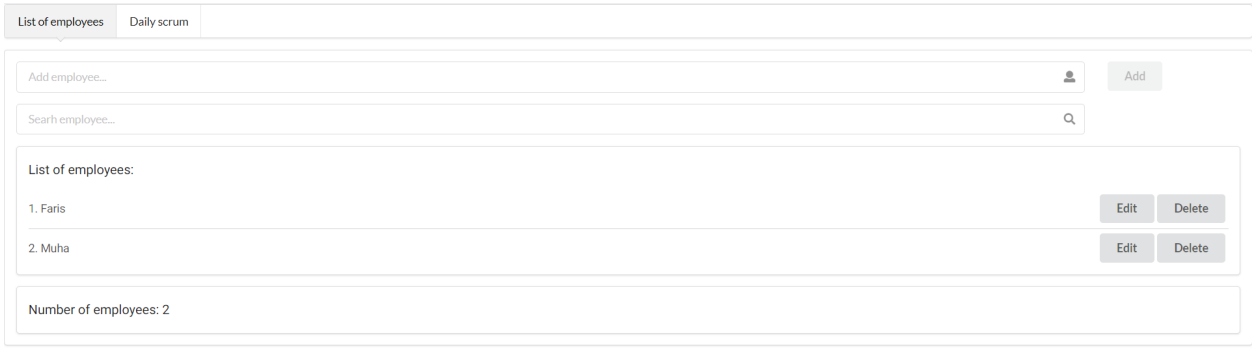


Fig 2: Employees Edit Page

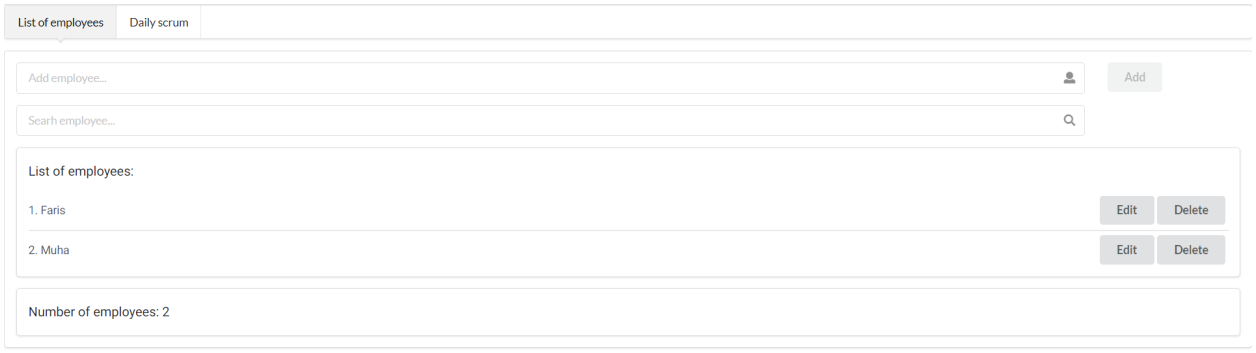


Fig 3: Daily Scrum Page

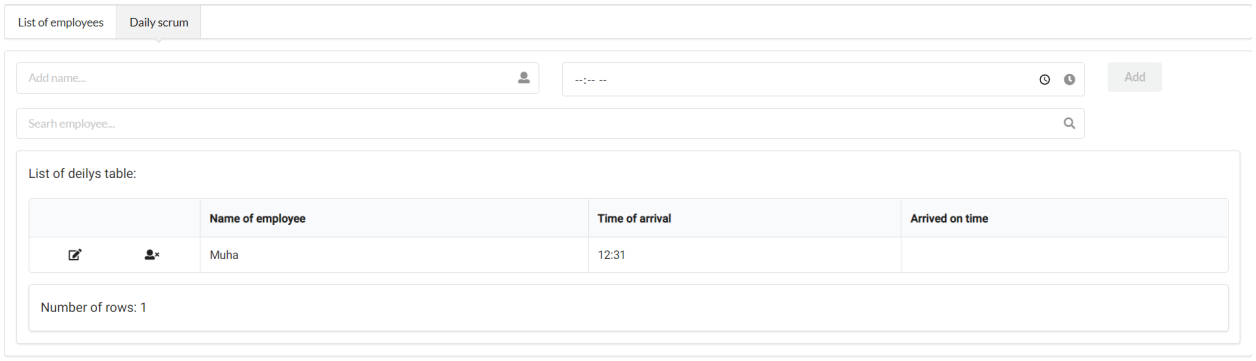


Fig 4: Daily Scrum Edit Page

# 2. Project Structure

## 2.1 Technologies

The technologies that we used for this project are:

- Angular with Redux - for implementing the frontend part
- NodeJS - for implementing the backend part
- mongoDB - for implementing the database

## 2.2 Coding Standard

We used Excel Micro coding standards which happen to be official for Typescript and Node.js scripts along with Google standard as well. We also used an Angular style guide for this code structure.

- Use const for all of your references; avoid using var.
- If you must mutate references, use let instead of var.
- Use the literal syntax for object creation.
- Use readable synonyms in place of reserved words.
- Use computed property names when creating objects with dynamic property names.

- Use arrow functions for object methods instead of shorthand properties or an anonymous function.
- Use property value shorthand.
- Use the literal syntax for array creation.
- Use Array#push instead of direct assignment to add items to an array.
- Use array spreads ... to copy arrays.
- Use array destructuring.
- Strings longer than 80 characters should be written across multiple lines using string concatenation.
- Use function expressions instead of function declarations.
- Never use arguments, opt to use rest syntax ... instead.
- In arrays with enumeration we wrote commas
- We wrote parentheses in the same line

## 2.3 Database Entities

Table names in our Mongo Database are :

- Users
- Arrivals

## 2.4 Architectural Pattern

The Architectural Pattern that we used is Angular Architectural Pattern. Angular architecture is defined in the main three layers: Core layer, Abstraction layer, and Presentation layer. This is also called High-Level Architecture.

Presentation Layer — This layer is mainly responsible for the design and user interface. Here we are defining all our angular components.

Abstraction layer — The abstraction layer decouples the presentation layer from the core layer. This layer will behave as a mediator and the facade for the presentation layer.For example, here we call services to have data communication between template and core layer.

Core layer — This layer is the innermost layer of an application, though all outside communication happens here. In this layer, one can place all state management, core modules, and services. In this layer, we also placed validators, mappers, or more advanced use-cases that require manipulating many slices of our UI state.

## 3. Conclusion

To conclude everything, we are not really satisfied with the project we made. The idea was so much better, but the implementation itself lacked a bit of skill in Node js/Angular. We chose this frameworks to work with for the reason because we wanted to learn new things in a short period of time.  Also, we tried and gave it all to fulfill the requirements for this project.

In the future, we will keep working on this project because we think that the idea behind it is great. We will try to make it fully functional with additional database entities, components, forms etc. Also, the most challenging part in this project was implementing rest methods on our frontend.