



UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI
ELECTRONICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI



PROIECT DE DIPLOMĂ

Marcu Andrei Cristian

COORDONATOR ȘTIINȚIFIC

Asist. drd. ing. Petcușin Felix Alin

Iulie 2021

CRAIOVA



UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI
ELECTRONICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI



Application for Human Resources Management in a Company

Marcu Andrei Cristian

COORDONATOR ȘTIINȚIFIC

Asist. drd. ing. Petcușin Felix Alin

Iulie 2021

CRAIOVA

„Educatia e painea sufletului.”

Proverb popular

DECLARAȚIE DE ORIGINALITATE

Subsemnatul Marcu Andrei Cristian, student la specializarea Calculatoare cu predare în limba engleză din cadrul Facultății de Automatică, Calculatoare și Electronică a Universității din Craiova, certific prin prezenta că am luat la cunoștință de cele prezentate mai jos și că îmi asum, în acest context, originalitatea proiectului meu de licență:

- cu titlul Application for Human Resources Management in a Company,
- coordonată de Asist. drd. ing. Inginer Petcușin Felix Alin,
- prezentată în sesiunea Iulie 2021.

La elaborarea proiectului de licență, se consideră plagiat una dintre următoarele acțiuni:

- reproducerea exactă a cuvintelor unui alt autor, dintr-o altă lucrare, în limba română sau prin traducere dintr-o altă limbă, dacă se omit ghilimele și referința precisă,
- redarea cu alte cuvinte, reformularea prin cuvinte proprii sau rezumarea ideilor din alte lucrări, dacă nu se indică sursa bibliografică,
- prezentarea unor date experimentale obținute sau a unor aplicații realizate de alți autori fără menționarea corectă a acestor surse,
- însușirea totală sau parțială a unei lucrări în care regulile de mai sus sunt respectate, dar care are alt autor.

Pentru evitarea acestor situații neplăcute se recomandă:

- plasarea între ghilimele a citatelor directe și indicarea referinței într-o listă corespunzătoare la sfârșitul lucrării,
- indicarea în text a reformulării unei idei, opinii sau teorii și corespunzător în lista de referințe a sursei originale de la care s-a făcut preluarea,
- precizarea sursei de la care s-au preluat date experimentale, descrieri tehnice, figuri, imagini, statistici, tabele et caetera,
- precizarea referințelor poate fi omisă dacă se folosesc informații sau teorii arhicunoscute, a căror paternitate este unanim cunoscută și acceptată.

Data,

Semnătura candidatului,

20.06.2021



UNIVERSITATEA DIN CRAIOVA
Facultatea de Automatică, Calculatoare și Electronică
Departamentul de Calculatoare și Tehnologia Informației

Aprobat la data de
Șef de departament,
Prof. dr. ing.
Marius BREZOVAN

PROIECTUL DE DIPLOMĂ

| | |
|---|--|
| Numele și prenumele studentului/-ei: | Marcu Andrei Cristian |
| Enunțul temei: | Application for Human Resources Management in a Company Description: Aplicatia ofera o solutie pentru gestionarea angajatilor dintr-o companie. Oferta statistici si o vedere mai detaliata asupra companiei. |
| Datele de pornire: | ASP.NET Core, React, React Material UI, EF Core, Devexpress React, SQL Server |
| Conținutul proiectului: | <i>[Descrierea succintă a conținutului fiecărui capitol al lucrării]</i> |
| Material grafic obligatoriu: | Schema aplicatiei, diagrame UML, screenshot-uri din aplicatie |
| Consultații: | Periodice |
| Conducătorul științific (titlul, nume și prenume, semnătura): | Asist. drd. ing. Petcușin Felix Alin |
| Data eliberării temei: | 27.11.2020 |
| Termenul estimat de predare a proiectului: | 24.06.2021 |
| Data predării proiectului de către student și semnătura acestuia: | |



REFERATUL CONDUCĂTORULUI ȘTIINȚIFIC

Numele și prenumele candidatului/-ei:

Specializarea:

Calculatoare cu predare în limba engleză

Application for Human Resources Management in a Company

Titlul proiectului:

Locația în care s-a realizat practica de documentare (se bifează una sau mai multe din opțiunile din dreapta):

În facultate ☐

În producție ☐

În cercetare ☐

Altă locație: *[se detaliază]*

În urma analizei lucrării candidatului au fost constatate următoarele:

| | | | | | |
|-----------------------------|------------------------|---|---|--|---|
| Nivelul documentării | | Insuficient <input type="checkbox"/> | Satisfăcător <input type="checkbox"/> | Bine <input type="checkbox"/> | Foarte bine <input type="checkbox"/> |
| Tipul proiectului | | Cercetare <input type="checkbox"/> | Proiectare <input type="checkbox"/> | Realizare practică <input type="checkbox"/> | Altul <i>[se detaliază]</i> |
| Aparatul matematic utilizat | | Simplu <input type="checkbox"/> | Mediu <input type="checkbox"/> | Complex <input type="checkbox"/> | Absent <input type="checkbox"/> |
| Utilitate | | Contract de cercetare <input type="checkbox"/> | Cercetare internă <input type="checkbox"/> | Utilare <input type="checkbox"/> | Altul <i>[se detaliază]</i> |
| Redactarea lucrării | | Insuficient <input type="checkbox"/> | Satisfăcător <input type="checkbox"/> | Bine <input type="checkbox"/> | Foarte bine <input type="checkbox"/> |
| Partea grafică, desene | | Insuficientă <input type="checkbox"/> | Satisfăcătoare <input type="checkbox"/> | Bună <input type="checkbox"/> | Foarte bună <input type="checkbox"/> |
| Realizarea practică | Contribuția autorului | Insuficientă <input type="checkbox"/> | Satisfăcătoare <input type="checkbox"/> | Mare <input type="checkbox"/> | Foarte mare <input type="checkbox"/> |
| | Complexitatea temei | Simplă <input type="checkbox"/> | Medie <input type="checkbox"/> | Mare <input type="checkbox"/> | Complexă <input type="checkbox"/> |
| | Analiza cerințelor | Insuficient <input type="checkbox"/> | Satisfăcător <input type="checkbox"/> | Bine <input type="checkbox"/> | Foarte bine <input type="checkbox"/> |
| | Arhitectura | Simplă <input type="checkbox"/> | Medie <input type="checkbox"/> | Mare <input type="checkbox"/> | Complexă <input type="checkbox"/> |

| | | | | | |
|--------------------------|--|--|--|---|---|
| | Întocmirea specificațiilor funcționale | Insuficientă <input type="checkbox"/> | Satisfăcătoare <input type="checkbox"/> | Bună <input type="checkbox"/> | Foarte bună <input type="checkbox"/> |
| | Implementarea | Insuficientă <input type="checkbox"/> | Satisfăcătoare <input type="checkbox"/> | Bună <input type="checkbox"/> | Foarte bună <input type="checkbox"/> |
| | Testarea | Insuficientă <input type="checkbox"/> | Satisfăcătoare <input type="checkbox"/> | Bună <input type="checkbox"/> | Foarte bună <input type="checkbox"/> |
| | Funcționarea | Da <input type="checkbox"/> | Parțială <input type="checkbox"/> | Nu <input type="checkbox"/> | |
| Rezultate experimentale | | Experiment propriu <input type="checkbox"/> | | Preluare din bibliografie <input type="checkbox"/> | |
| Bibliografie | | Cărți | Reviste | Articole | Referințe web |
| Comentarii și observații | | | | | |

În concluzie, se propune:

| | |
|---|---|
| ADMITEREA PROIECTULUI <input type="checkbox"/> | RESPINGEREA PROIECTULUI <input type="checkbox"/> |
|---|---|

Data,

Semnătura conducătorului științific,

PROJECT SUMMARY

The Application for Human Resources Management in a Company is an application designed to simplify the work of the organization on an IT company. Shortly called “*HRDesk*”, the application handles the holiday requests of the employees, offers an overview of all the holidays, has the possibility to book meeting rooms and future plan the organization of the internal calls. The application also generates reports for different types of dates like employees, holiday requests, leave requests, upcoming meetings.

The reason I’ve chose this topic it’s because the application is useful in real world situations and offers multiple ways of expansion. From the point where the application is right now it can be extended to also support polls, salary reports, virtual office management, food menus, and multiple internal company necessities.

The following technologies were used in the development of the website:

Frontend:

- [React](#)
- [React Redux](#)
- [Material UI](#)
- [DevExtreme Reactive](#)

Backend:

- [ASP.NET Core](#)
- [Entity Framework Core](#)

Database:

- [Sql Server](#)

Version control:

- [Github](#)
- [Sourcetree](#)

The application is divided on multiple levels, and each user needs permission in order to access them.

1. Dashboard – Contains general data about the company

2. Leave requests – Possibility to add a leave request(a leave request is a unexpected leave from the work, like going to a doctor, which can be maximum a few hours)
3. Dayoff requests – Possibility to add a holiday request
4. Reports – Charts about the company
5. Meetings – Upcoming meetings of the logged-in user
6. Team – Company employees overview
7. Holiday Calendar – Holiday tracker for the user and it's company(can see his holidays, teams holidays and free national days)
8. Book room – Possibility to create a Meeting which will be assigned to a Team\
9. Manage employees – Add, Edit and Delete employees
10. Manage holidays – Approve or Decline Leave requests, Holidays and create National Days
11. Manage organization – Management data like offices, meeting rooms, teams and functions.

Termenii cheie: website, React, Redux, SQL, ASP.NET Core, Entity Framework, Human resource management, HRDesk, Devextreme Reactive, Material UI

MULȚUMIRI

În această secțiune opțională (în eng., *Acknowledgements*), autorul are ocazia de a face o declarație de recunoștință față de oricine (conducătorul științific/alte persoane apropiate autorului/instituției/organizației/et caetera) a susținut sau a contribuit la realizarea lucrării sale.

CUPRINSUL

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 1 |
| 1.1 | THE OBJECTIVE..... | 1 |
| 1.2 | THE MOTIVATION | 2 |
| 1.3 | DOCUMENT STRUCTURE | 2 |
| 2 | TECHNOLOGIES..... | 3 |
| 2.1 | ASP.NET CORE..... | 3 |
| 2.1.1 | <i>What is ASP.NET Core?</i> | <i>3</i> |
| 2.1.2 | <i>NuGet packages.....</i> | <i>3</i> |
| 2.1.3 | <i>Why should you use ASP.NET Core?.....</i> | <i>3</i> |
| 2.1.4 | <i>ASP.NET Core vs ASP.NET.....</i> | <i>4</i> |
| 2.2 | ENTITY FRAMEWORK CORE | 4 |
| 2.2.1 | <i>What is Entity Framework Core?</i> | <i>4</i> |
| 2.2.2 | <i>LINQ</i> | <i>4</i> |
| 2.2.3 | <i>Code first approach.....</i> | <i>5</i> |
| 2.3 | REACT..... | 7 |
| 2.3.1 | <i>What is React?.....</i> | <i>7</i> |
| 2.3.2 | <i>JSX.....</i> | <i>7</i> |
| 2.3.3 | <i>React Redux</i> | <i>8</i> |
| 2.3.4 | <i>Axios.....</i> | <i>9</i> |
| 2.3.5 | <i>React Material UI.....</i> | <i>10</i> |
| 2.3.6 | <i>Devexpress Scheduler.....</i> | <i>10</i> |
| 2.4 | SQL SERVER | 10 |
| 3 | PROJECT STRUCTURE..... | 12 |
| 3.1 | HRDESK BACKEND SOLUTION | 12 |
| 3.2 | HRDESK FRONTEND SOLUTION | 12 |
| 3.3 | REPOSITORY PATTERN | 13 |
| 3.4 | UNIT OF WORK..... | 14 |
| 3.5 | EXCEPTION MIDDLEWARE..... | 14 |
| 3.6 | DEPENDENCY INJECTION | 15 |
| 4 | DATABASE STRUCTURE..... | 16 |
| 4.1 | DATABASE SCHEMA | 16 |
| 4.2 | TABLES..... | 16 |

| | | |
|----------|--|-----------|
| 4.2.1 | <i>Users</i> | 16 |
| 4.2.2 | <i>Teams</i> | 16 |
| 4.2.3 | <i>Functions</i> | 17 |
| 4.2.4 | <i>Offices</i> | 17 |
| 4.2.5 | <i>Leave requests</i> | 17 |
| 4.2.6 | <i>Daysoff</i> | 17 |
| 4.2.7 | <i>Company Details</i> | 17 |
| 4.2.8 | <i>Personal Details</i> | 17 |
| 4.2.9 | <i>Meeting Rooms</i> | 18 |
| 4.2.10 | <i>Meetings</i> | 18 |
| 4.2.11 | <i>National days</i> | 18 |
| 4.2.12 | <i>Permissions</i> | 18 |
| 4.2.13 | <i>Polls, PollAnswers, UserPollAnswers</i> | 18 |
| 4.2.14 | <i>UserPermissions</i> | 18 |
| 5 | HRDESK MODULES | 19 |
| 5.1 | DASHBOARD MODULE | 19 |
| 5.2 | LEAVE REQUEST MODULE | 19 |
| 5.3 | DAYOFF REQUEST MODULE..... | 20 |
| 5.4 | REPORTS MODULE | 21 |
| 5.5 | MEETINGS MODULE | 22 |
| 5.6 | TEAM MODULE | 23 |
| 5.7 | HOLIDAY CALENDAR MODULE | 24 |
| 5.8 | BOOKING MODULE | 25 |
| 5.9 | MANAGE EMPLOYEES MODULE | 26 |
| 5.10 | MANAGE HOLIDAYS MODULE..... | 29 |
| 5.11 | MANAGE ORGANIZATION..... | 31 |
| 5.12 | UNAUTHORIZED ACCESS | 31 |
| 5.13 | LOGIN PAGE | 31 |
| 6 | UML DIAGRAMS | 32 |
| 6.1 | CERINȚE GENERALE | 33 |
| 6.2 | STRUCTURA DOCUMENTULUI | 33 |
| 6.3 | DIMENSIUNILE LUCRĂRII | 34 |
| 6.4 | ELEMENTE DE TEHNOREDACTARE..... | 35 |
| 6.5 | FORMULELE MATEMATICE..... | 35 |
| 6.6 | ILUSTRĂȚIILE..... | 35 |

| | | |
|-----------|---|-----------|
| 6.6.1 | <i>Figurile</i> | 35 |
| 6.6.2 | <i>Tabelele</i> | 36 |
| 6.6.3 | <i>Legenda (unei figuri/tabele)</i> | 37 |
| 7 | TERMENI DE UTILIZARE | 38 |
| 7.1 | AUTORII | 38 |
| 7.2 | LICENȚA DE UTILIZARE | 38 |
| 8 | CONCLUZII | 39 |
| 9 | BIBLIOGRAFIE | 40 |
| 10 | REFERINȚE WEB | 41 |
| A. | CODUL SURSĂ | 42 |
| B. | SITE-UL WEB AL PROIECTULUI | 43 |
| C. | CD / DVD | 44 |
| | INDEX | 45 |

LISTA FIGURILOR

| | |
|--|---|
| FIGURE 1. ASP.NET CORE LOGO [1] | 3 |
| FIGURE 2. LINQ QUERY SYNTAX | 5 |
| FIGURE 3. LINQ EXTENSION METHODS | 5 |
| FIGURE 4. CODE FIRST APPROACH[6]..... | 5 |
| FIGURE 5. HRDESK USER ENTITY EXAMPLE | 6 |
| FIGURE 6 - REACT LOGO[8] | 7 |
| FIGURE 7. JSX CODE[9]..... | 7 |
| FIGURE 8. HOW JSX REALLY LOOKS BEHIND AFTER BEING PARSED BY BABEL[9] | 8 |
| FIGURE 9. REACT REDUX LOGO[10] | 8 |
| FIGURE 10. AXIOS WRAPPER FROM HRDESK | 9 |

LISTA TABELELOR

| | |
|---|---|
| TABLE 1 - ASP.NET CORE VS ASP.NET | 4 |
|---|---|

1 INTRODUCTION

1.1 The objective

The objective of the Human resource management application in a company called „HRDesk” from now on is to come in help of the companies, granting an alternative of employees management. The application offers different statistics, possibility to export data in Excels and it is designed to be used however the company wants. This was achieved by avoiding creating predefined roles(which would've got access to a certain module) and instead when an user is created it will be assigned to multiple modules.(For each menu item in the left drawer there is a module). This way the users can be splitted on responsibilities(for example there can be a user which accepts/declines the holiday requests of the employees). This user can also be a developer.

Available modules at the time of the presentation:

1. Dashboard – Contains general data about the company and the upcoming meetings of the user
2. Leave requests – Employee/Administrative level – the user can add or delete a leave request. A leave request is an urgent leave, for example going to a doctor
3. Dayoff requests – Employee level – the user can add or delete a day off. Also in here he can keep the track of his number of free days. When we calculate the number of free days we check the overlap with weekends, free national days and another holidays.
4. Reports – Employee/Administrative level – Different charts about the company(Age average chart, Country of provenience chart, Functions chart)
5. Meetings – Employee level – Upcoming meetings between two dates
6. Team – Employee level – Company employees ordered by employment date
7. Holiday Calendar – Employee level – Can see his own holidays, team members holidays and also national days
8. Book room – Administrative level – Reserve a meeting room for a team. This meeting will be displayed on the Meetings module.
9. Manage employees – Administrative level – A more detailed overview of the employees. Can add, delete, update them. In here we can assign the user to a module and also we can generate a report(Excel) of all the employees
10. Manage holidays – Administrative level – This module handles the leave and day off requests created before by the employee. The administration can either accept and decline them. Also on this level we have the national days, which are considered as day offs for all employees.

11. Manage organization – Administrative level – Handles the location management. For example offices, meeting rooms(used for booking), teams and functions.

1.2 The motivation

The main reason for choosing this specific topic is because this is a real world usable application that can be extended with many functionalities. Also I am working in a company of this kind and I've seen the benefits an internal application can provide. Keeping track of employees data, like number of free days, salary, and multiple details can simplify the work of the management team of a company.

Also another reason is that the requirement of the topic was to be a Web application, the domain I'm interested in evolving the most. This way I was able to use all the knowledge acquired during the university and create an usable and opened for extension application.

1.3 Document structure

2 TECHNOLOGIES

2.1 ASP.NET Core

2.1.1 What is ASP.NET Core?



Figure 1. ASP.NET Core logo [1]

ASP.NET Core is the successor of the .NET Framework developed by Microsoft. It is a cross-platform framework targeted to run on the .NET Core platform. It's free, open-source and supports the development of multiple types of applications(Web, Cloud, Mobile etc.).[2]

ASP.NET Core features can be extended by the use of NuGet packages.

2.1.2 NuGet packages

A NuGet package is an essential tool for any modern development platform. It's a mechanism through which developers can create, share and consume useful code. Put simply, a NuGet package is a single ZIP file that contains compiled code. These packages can be private(used by a restrained number of users) or can be public. One example of NuGet package is Entity Framework which will be detailed later on.[3]

2.1.3 Why should you use ASP.NET Core?

- Cross-platform support
- Good performance
- Expandable using NuGet packages
- Cloud support
- Open source

- Can be used for many types of applications(Web, Mobile, Cloud etc)
- Created by Microsoft
- Updated constantly

2.1.4 ASP.NET Core vs ASP.NET

| ASP.NET Core | ASP.NET |
|---|--|
| Supports Windows, Linux and macOS | Supports Windows |
| Bigger CPU usage and performance | Lower CPU usage and performance |
| Can be hosted on multiple platforms | Can be hosted on IIS(Internet Information Server) |
| Supports Docker | Doesn't support Docker |
| It is focused on Web applications development | It is focused on both Desktop and Web applications |

Table 1. ASP.NET Core vs ASP.NET [4]

2.2 Entity Framework Core

2.2.1 What is Entity Framework Core?

Entity Framework Core is an open source and cross platform object-relational mapper. It was initially part of .NET Framework and starting with Entity Framework 6 was delivered separately. This can be acquired from the NuGet packages. [5]

In my application Entity Framework Core was used in order to simplify the relation with the database. Writing LINQ code directly in the main project, without the need of SQL stored procedures makes the code more readable. In my case the database was created using a code first approach with migrations

2.2.2 LINQ

Language-Integrated Query(LINQ) is the technology that allow the user to do queries directly from C# code.

Linq syntax is of two types:

- Query syntax. Similar two classic SQL syntax.

```
0 references
public User QuerySyntaxExample(int id)
{
    var users = GetAll();
    return (from user in users
            where user.Id == id
            orderby user.CreatedDate
            select user).SingleOrDefault();
}
```

Figure 2. Linq query syntax

- Extension method syntax

```
0 references
public User ExtensionMethodExample(int id)
{
    var users = GetAll();
    return users.OrderBy(user => user.CreatedDate).FirstOrDefault(user => user.Id == id);
}
```

Figure 3. Linq extension methods

2.2.3 Code first approach

In entity framework code first approach the focus is on creating entities from C# code. Based on the Context of the application Entity Framework will create the corresponding tables via migrations. Using Data Annotations on Entity fields you can add limitations like in SQL(for example maximum length of string).

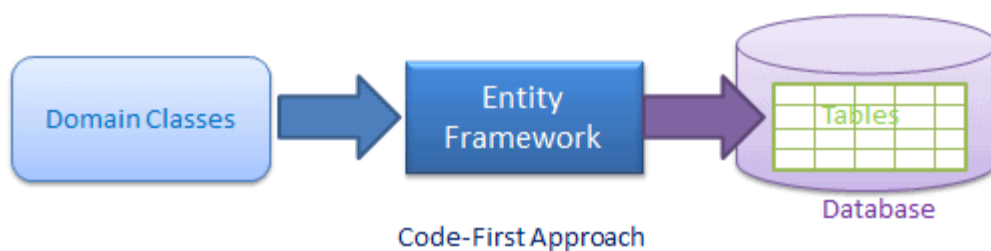


Figure 4. Code first approach[6]

In HRDesk I've used the code first approach. The following picture displays an Entity example from the application. In this case the Entity also inherits a "Base Entity" which is DeleteEntity. This

entity automatically adds some extra fields, which are: IsDeleted(used for soft delete), Id(unique identifier), Creation date, Updated date(if it is updated by anyone).

```
29 references
public class User : DeleteEntity<int>
{
    6 references
    public string WorkEmail { get; set; }
    4 references
    public string Password { get; set; }
    2 references
    public bool Predefined { get; set; }
    8 references
    public int? TeamId { get; set; }

    [ForeignKey("TeamId")]
    6 references
    public Team Team { get; set; }

    4 references
    public int? FunctionId { get; set; }

    [ForeignKey("FunctionId")]
    19 references
    public Function Function { get; set; }

    4 references
    public int? OfficeId { get; set; }

    [ForeignKey("OfficeId")]
    6 references
    public Office Office { get; set; }
    2 references
    public ICollection<UserPermission> Permissions { get; set; }
    1 reference
    public int? CompanyDetailsId { get; set; }

    [ForeignKey("CompanyDetailsId")]
    27 references
    public CompanyDetails CompanyDetails { get; set; }
    1 reference
    public int? PersonalDetailsId { get; set; }

    [ForeignKey("PersonalDetailsId")]
    46 references
    public PersonalDetails PersonalDetails { get; set; }
    7 references
    public byte[] ImageSrc { get; set; }
    4 references
    public string ImageType { get; set; }
}
```

Figure 5. HRDesk User entity example

2.3 React

2.3.1 What is React?

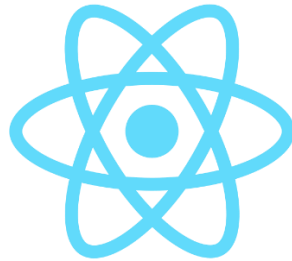


Figure 6 - React logo[8]

React is an open-source front-end JavaScript library created by Facebook. The code is structured on UI Components. The main reason of the component is that the code is modular and one component can be used by multiple other components. This way the user is forced to create reusable code. React is mainly focused on state management and displaying the data in the DOM. There are additional libraries that extend it's functionalities. For example react-router-dom for routing, axios for api calls, material-ui for CSS components.[7]

2.3.2 JSX

“JSX is an XML/HTML-like syntax used by React so that XML/HTML-like text can co-exist with JavaScript/React code. The syntax is intended to be used by preprocessors (i.e., Babel) to transform HTML-like text found in JavaScript files into standard JavaScript objects that a JavaScript engine will parse.” [9]

Basically, you write HTML-like code which is then parsed by Babel or another processor into Javascript code. The differences between normal HTML and JSX are not noticeable.

```
var nav = (  
  <ul id="nav">  
    <li><a href="#">Home</a></li>  
    <li><a href="#">About</a></li>  
  </ul>  
)  
;
```

Figure 7. JSX Code[9]

```

var nav = React.createElement(
  "ul",
  { id: "nav" },
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Home"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "About"
    )
  )
);

```

Figure 8. How JSX really looks behind after being parsed by Babel[9]

2.3.3 React Redux



Figure 9. React Redux logo[10]

In React the state it's placed inside a component and it's available only there. This state can be passed to the Parent Component or the Child component and this causes a limitation. React Redux provides an alternative way to create the state. React Redux creates a global state named store.

Redux is composed of the following components:

- Actions: It's a JavaScript object that contains a type field. The type field describes the action. For example: "getUser".
- Reducer: It's a function that updates the state(if needed) based on the action
- Dispatch: Receives an action as parameter and calls the reducer function of that specific function.
- Store: The actual global state. This can be accessed all over the application using Redux connect.
- Selector: Extracts a field from the state

2.3.4 Axios

Axios is a HTTP client from Node.js. I've created a wrapper using this package for handling api requests. The wrapper automatically creates the request body, attaches the API base url and the authorization token if it's available

```
import axios from "axios";
import { getToken } from "../services/storage";

const axiosWrapper = () => {
  const defaultOptions = {
    baseURL: process.env.REACT_APP_BASE_URL,
    headers: {
      "Content-Type": "application/json",
    },
  };

  let instance = axios.create(defaultOptions);

  instance.interceptors.request.use(function (config) {
    const token = getToken();
    config.headers.Authorization = token ? `Bearer ${token}` : "";
    return config;
  });

  return instance;
};

export default axiosWrapper;
```

Figure 10. Axios wrapper from HRDesk

From this point the wrapper can handle any type of requests, by simply receiving the controller url and the payload.

2.3.5 React Material UI

Material UI is an open-source library that provides a multitude of UI components for React application. Material UI follows the guidelines provided by Google on Material Design. It is similar to Bootstrap, it's also based on a 12 column grid structure. Multiple components from Material UI were used in the development of the HRDesk application. The most encountered ones in my application are: Container, Grid, Data Grid, Card, Button, Transfer List, Drawer, Paper, App bar, Avatar and Material Icons. Using the grid system from Material UI caused the application to be as mobile responsive as possible. There are indeed some limitations on data grid with many columns, which can not be resized on a mobile device and another component should be used for this case.

2.3.6 Devexpress Scheduler

The Devexpress scheduler was used for the holiday and booking calendar. It offers a similar type of calendar to the one in Outlook. This calendar can be customized in any means by the user since it accepts any component to be overwritten. The scheduler supports recurrent events and for a booking system this was ideal. It's user friendly and it's interface is similar to the one from the Material Design.

2.4 SQL Server

SQL Server is a relational database management developed by Microsoft. It is based on the SQL query language. SQL Server can handle database with a huge amount of data. SQL Server has it's own SQL derivation, called T-SQL. T-SQL extends the classic SQL queries by adding additional syntax for stored procedures and transactions. It support multiple types of data, from numbers and strings to binary data.[11][12]

SQL Server is also available as a cloud storage on Microsoft Azure. The Azure SQL Database is based on the last version of SQL Server

2.5 Github



Figure 11. GitHub logo[13]

GitHub is a version control website for software development. It offers multiple services like version control, code management, pull requests, task management, bug tracking. It's free to use, the user can create both public and private repositories. There are some limitations for the free plan, like the number of contributors but that is not an issue for a final exam project. In 2018 GitHub was bought by Microsoft. It the largest host of source code in the world. [14]

While developing HRDesk I've committed after each major change on GitHub, this way whenever an issue occurred I could check the old code, find the cause of the issue easier and or course keep the code safe on the GitHub storage. For my relation with GitHub I've used a Git client available for windows and iOS, called Sourcetree. HRDesk git repository will be found in the how to use chapter.

2.6 Sourcetree

Sourcetree is a free to use Git client developed by Atlassian. It is a middleware between the user and a Git. It's intuitive and there are only a few buttons, which makes it very easy to use. Writing the same commands from git bash would be way harder. Also, in case if something is not supported by it's buttons available in the UI, for example hard resetting a merge, or rollbacking a commit it also offers a git bash directly from the UI.

In my opinion the best thing about the Sourcetree is the history view, where you can see all the commits and branches, each branch with it's own color and line

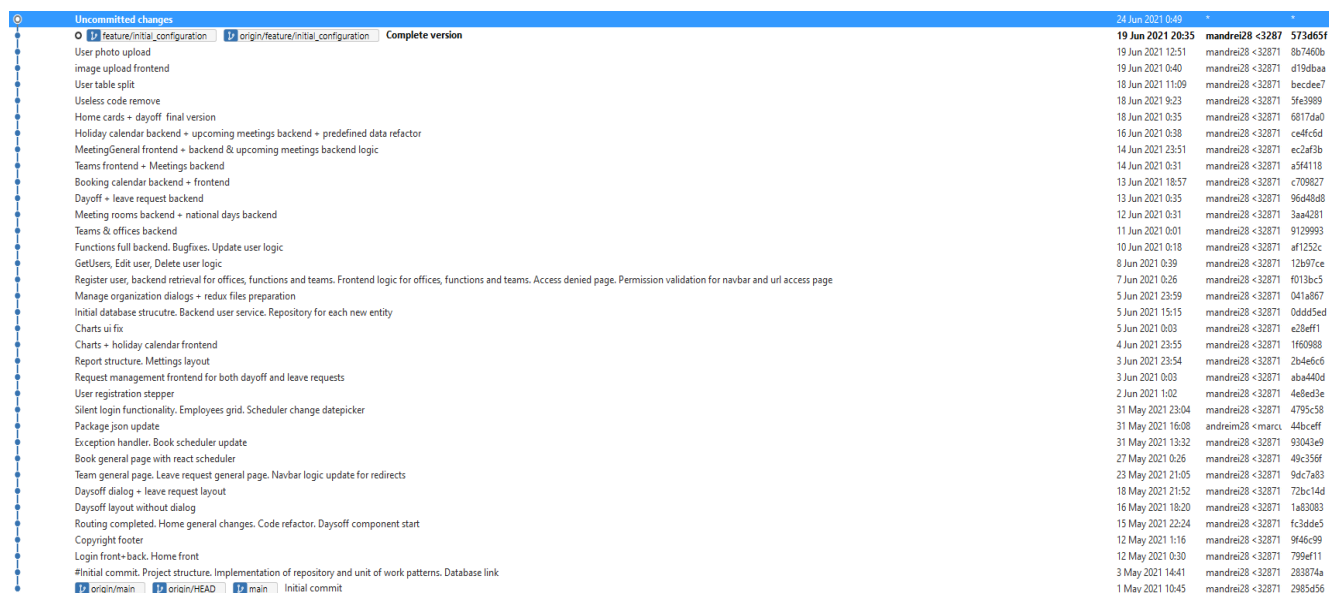


Figure 12. HRDesk sourcetree history

In my case I've used only one branch, so there is only one line. In this view I can see all the changes I've made in that commit and rollback to an exact commit if needed.

3 PROJECT STRUCTURE

3.1 HRDesk backend solution

HRDesk backend solution is found in the WebApi folder

HRDesk solution is splitted on 3 projects:

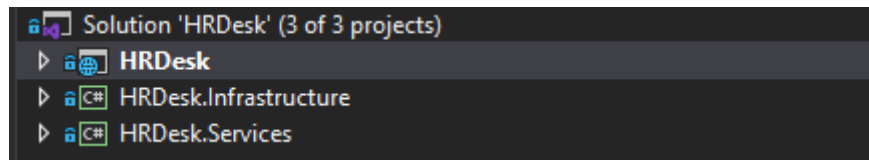


Figure 13. HRDesk solution

- HRDesk contains the controllers and the Startup. Also there is a wrapper for dependency injection, which is simply called on Startup.
- HRDesk.Infrastructure contains everything needed by the database. In this project we can find the entities, repositories, migrations, DbContext, UnitOfWork.
- HRDesk.Services contains the services of the application. All the logic needed all over the application, mappers, error handlers, exception middlewares.

3.2 HRDesk frontend solution

HRDesk frontend solution is found in the WebApp folder

The project is composed of the following structure:

- Api : Contains the Axios wrapper for api calls
- Common: Contains the actual components. A component is composed of a container(which contains the dispatch methods from Redux), general file which contains the actual React code and styles which contains css and material ui styles
- Media: Images, or any file needed in application
- Redux: The actions, reducers and selectors needed by Redux
- Services: Different services that were needed between multiple components, and in order to not duplicate the code this was exported from an external file
- Utils: Constant values used in the application

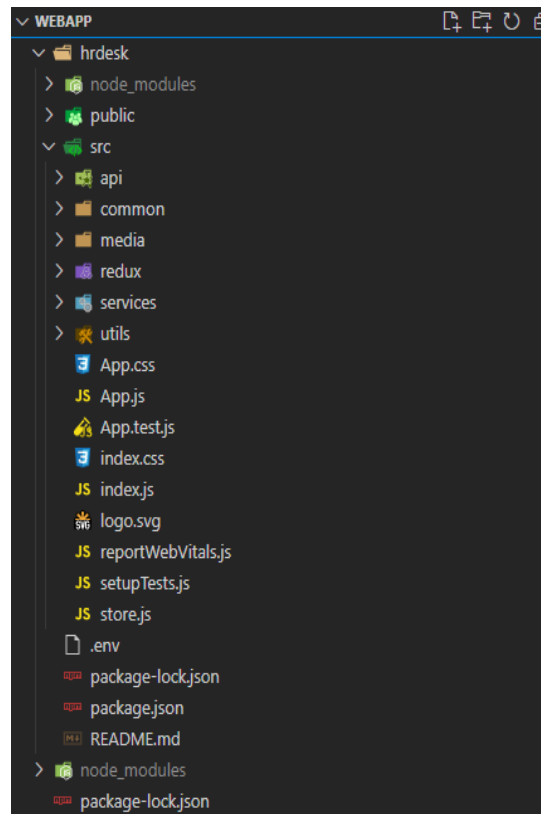


Figure 14. HRDesk frontend solution

3.3 Repository pattern

Repositories are classes that contain the logic to access the database. Most of the code is made out of Linq code using extensions methods. I've also created a Base Repository which contains CRUD operations. For each entity(table in database) there is one repository. The normal repository extends the CRUD operations provided by the BaseRepository(for example GetUserByEmail, GetUserByName etc.).

The repositories contain one extra level of abstraction, created by the Unit of Work pattern. Using repository pattern will minimize duplicate logic and will decouple the services from the actual database framework. The services are independent of the framework, as long as the logic is kept in the repositories. If the database relation framework (Entity Framework in my case) needs to be changed at any time, the code will be modified only in the repositories.

3.4 Unit of Work

An Unit of Work is a class that controls all the repositories, providing an extra layer of abstractisation between the services and the repositories. This way, the user doesn't depend on a framework. For example if entity framework will be replaced in the future the code needs to be changed only in the repositories, leaving the services untouched. This way the application is flexible, maintainable and opened for extension.

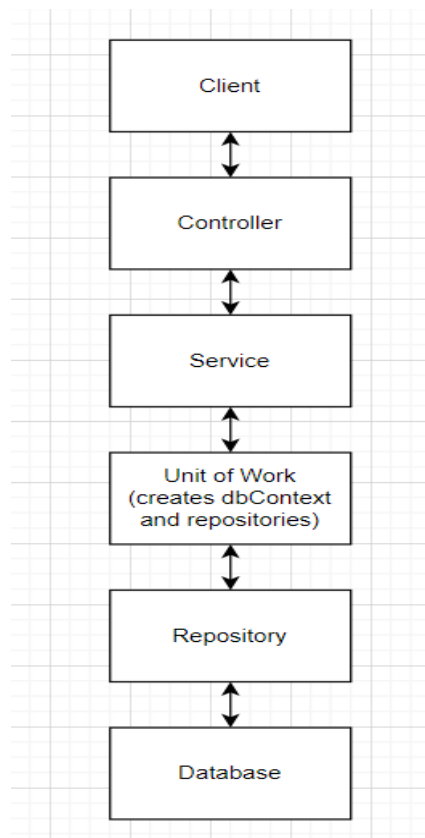


Figure 15. Unit of work pattern flow

3.5 Exception middleware

The application has an automatic exception handler, whenever an exception is thrown the message is captured by the middleware and it's sent to the frontend. In frontend the code is caught by axios and the error message is automatically displayed in a red toaster. This way the user will know whenever a request fails, and a message will indicate the cause of the issue. One example of exception is thrown when the administrator tries to create a user whose email address is already found in the database. The actual user will receive the error message in toaster "Email already in use".

3.6 Dependency Injection

ASP.NET Core supports dependency injection pattern. The classes and their interfaces are registered in the startup of the application. There are three types of lifetimes:

Singleton: Wherever the dependency injection is made the same instance of class will be used for all the calls

Scoped: The instance is the same within the scope of a request, but it's changed when another request is made.

Transient: The instance is always different

Dependency Injection decouples dependencies between services by using abstractions.

4 DATABASE STRUCTURE

4.1 Database schema

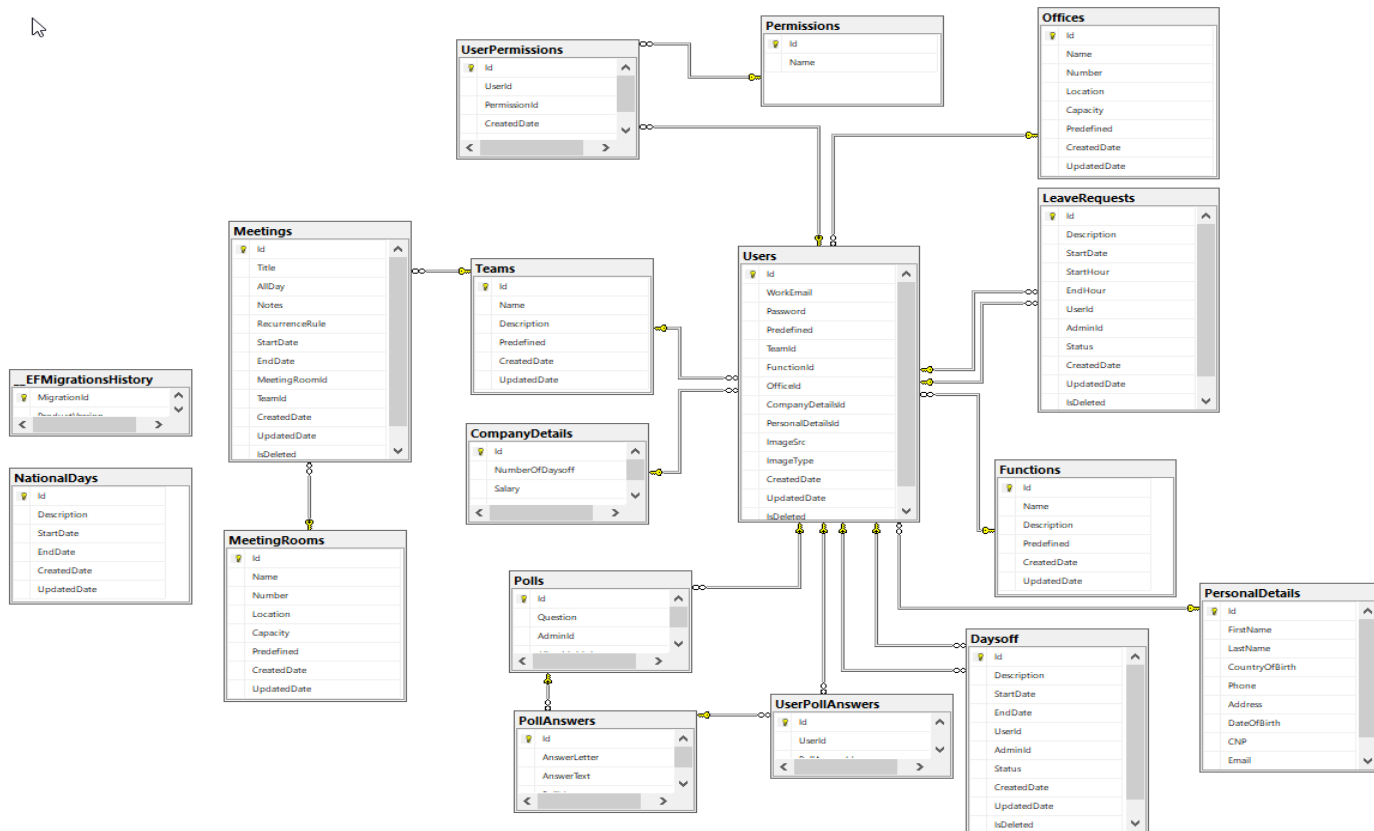


Figure 16. HRDesk database diagram

4.2 Tables

4.2.1 Users

Users is the main table of the application, we can call it the core table. This table is related with most of the other tables. It contains all the data about the employee. The details are splitted in two, personal details and company details. An user needs to be linked to a function, table and office in order to be usable in the application. The account and the password of the user also encoded and stored in this database. The type of the user entity is DeleteEntity, meaning that when an user is marked as deleted but not actually removed from the table.

4.2.2 Teams

Teams table is used to store the teams available in the companies. An user needs to be assigned to a team, and a team is needed when a booking is made.

4.2.3 Functions

Functions are stored in this table. An user needs to be linked to a function. The functions are used all over the application. Also there is a chart related to the functions

4.2.4 Offices

An user needs to be linked to an office. The offices are used for a better organization in the company.

4.2.5 Leave requests

The leave requests contain two FKs to the user table. One points to the user that made the leave request and the other one points the administrator that has to accept or decline it. The status of a leave request is defined by the following enum:

- Status = 0 -> Waiting
- Status = 1 -> Refused
- Status = 2 -> Approved

The type of this entity is DeleteEntity, meaning that a soft delete is performed instead of an actual remove.

4.2.6 Daysoff

The daysoff are similar to the leave request, the main difference between these entities is that a dayoff can be made on multiple days, while a leave request is limited to hours in a day. The status enum is the same one from the leave request.

- Status = 0 -> Waiting
- Status = 1 -> Refused
- Status = 2 -> Approved

4.2.7 Company Details

It's an extension of the user table. Contains user's company related data, like salary, employment date, number of free days.

4.2.8 Personal Details

It's an extension of the user table. Contains user's personal data like birth date, name, country of provenience, email, telephone number.

4.2.9 Meeting Rooms

Meeting rooms are similar to offices. The main use of them is by the booking calendar. For each meeting room there will be a column displayed in the calendar.

4.2.10 Meetings

A meeting is related to a Team and a Meeting room. The recurrence rule is a string defined by the devexpress scheduler. This string will tell the scheduler if the meeting is recurrent, or if it's an all day meeting. The meetings are created from the booking calendar and can be seen in both the Booking calendar, home page and meetings module.

4.2.11 National days

National days are added by an administrator. A national day is similar to a dayoff but it's not assigned to an user and doesn't need to be approved. It's a standalone entity that it's reflected to all the employees in the company.

4.2.12 Permissions

In the permissions table we can find the modules. There is no way to add or remove them. The data is inserted by the seed method at the start of the application. The permissions are linked to an user using a many to many relationship, making the user to see the actual module.

4.2.13 Polls, PollAnswers, UserPollAnswers

Those tables are related.

- Polls contains the actual question of the poll and if the poll accepts multiple answers from an user.
- PollAnswers contains the possible answers for that poll, we can have one or many answers for a poll.
- UserPollAnswers is the many to many relationship between the answers and the user. If the poll accepts multiple answers each user will have multiple entries in this table. Also this table support the case when the poll accepts only one entry

4.2.14 UserPermissions

This table is the many to many table between users and permissions. Based on this data the user gets access to modules

5 HRDESK MODULES

5.1 Dashboard module

The dashboard is a module available to all the user in the application. It's main purpose it's of a home page. Contain details about the number of employees in the company and also an overview about the upcoming meetings.

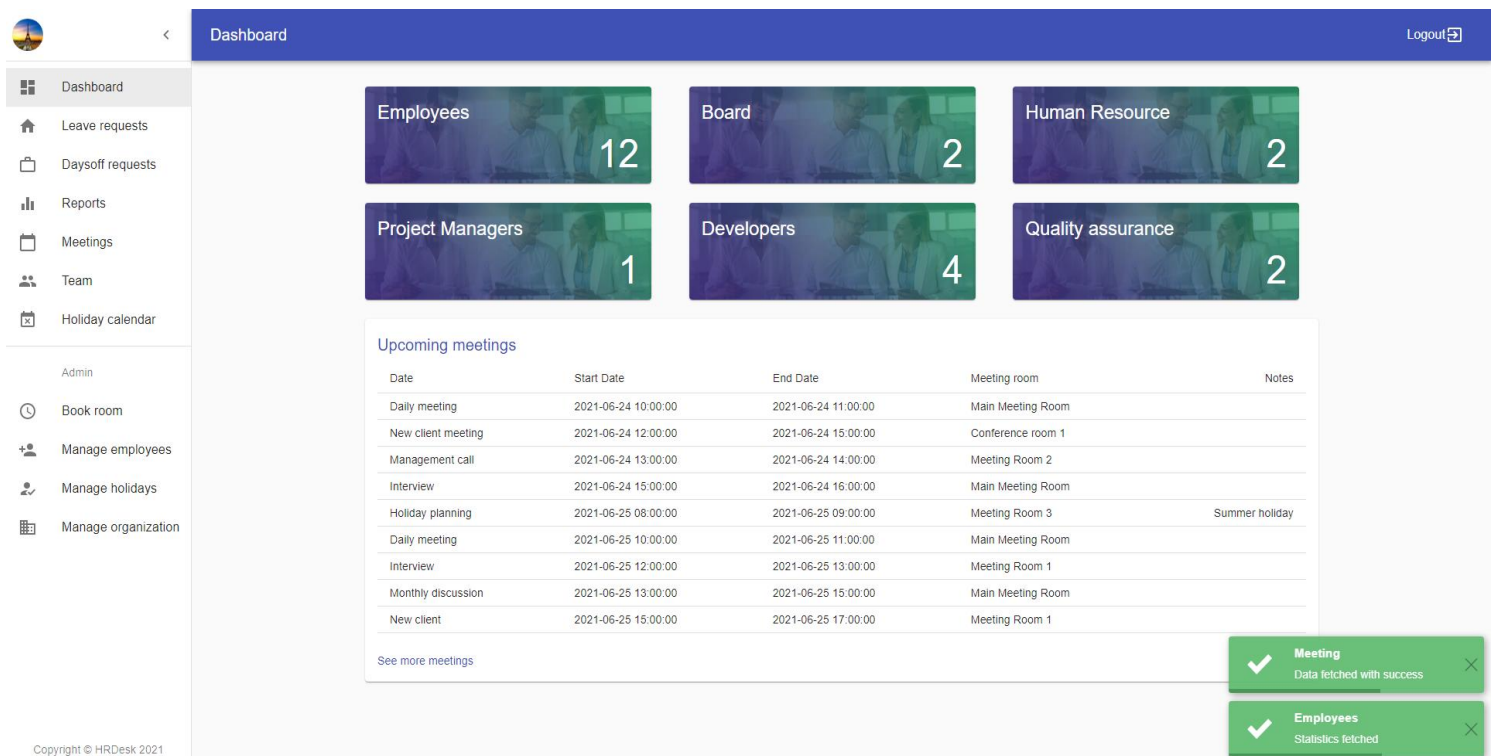



Figure 17. Dashboard module

5.2 Leave request module

As mentioned earlier in the presentation a leave request is a temporary necessity to leave the workplace, the most encountered example is going to a doctor. After the user talks with a person which can approve it's leave request he can come to this module and create a leave request. When creating a leave request he needs to specify the interval he will be gone, and also the person which he spoke with. The leave request will be sent to the person he spoke with, and from here the administrator can accept or decline the leave request. This request doesn't impact the number of free days. Let's call them unexpected holidays.



Dashboard

Leave requests

Daysoff requests

Reports

Meetings

Team

Holiday calendar

Admin

Book room

Manage employees

Manage holidays

Manage organization

Leave requests

Logout

Short description *

Date *06/24/2021

Start Hour11:01 PM

End hour11:01 PM

Spoke with *

ADD

| ID | Description | Date | Start hour | End hour | Status | Verified By | |
|----|----------------|------------|------------|----------|----------|-----------------------|--|
| 31 | laborator | 2021-06-25 | 22:25:00 | 22:25:00 | Waiting | | |
| 24 | Laborator | 2021-06-28 | 13:00:00 | 14:00:00 | Waiting | | |
| 32 | facultate | 2021-06-29 | 14:00:00 | 16:00:00 | Waiting | | |
| 33 | facultate | 2021-06-30 | 14:00:00 | 16:00:00 | Waiting | | |
| 34 | work from home | 2021-06-30 | 08:00:00 | 16:00:00 | Waiting | | |
| 35 | work from home | 2021-07-01 | 08:00:00 | 16:00:00 | Waiting | | |
| 36 | work from home | 2021-07-02 | 08:00:00 | 16:00:00 | Waiting | | |
| 37 | work from home | 2021-07-05 | 08:00:00 | 16:00:00 | Waiting | | |
| 1 | Dentist | 2021-06-24 | 20:00:00 | 21:00:00 | Approved | Marcu Andrei Cristian | |

1-9 of 9

✓

LeaveRequest

Data fetched with success

Copyright © HRDesk 2021

Figure 18. Leave request module

5.3 Dayoff request module

Similar to leave requests, but those are the actual holiday requests of the user. Compared to a leave request a day off can contain multiple days, while a leave request is limited only to hour intervals. Also, when creating a holiday request the user needs to specify the administrator he spoke with, and the administrator will have to accept or decline the dayoff.

If the dayoff is accepted then the number of free days of the user will be automatically adjusted. This number can be noticed on the chart from the top of the page. With blue is the number of free days and with red the number of used days. The number of free days used by a holiday request is calculated without the days that are in weekend, or overlap a national day, or another holiday.

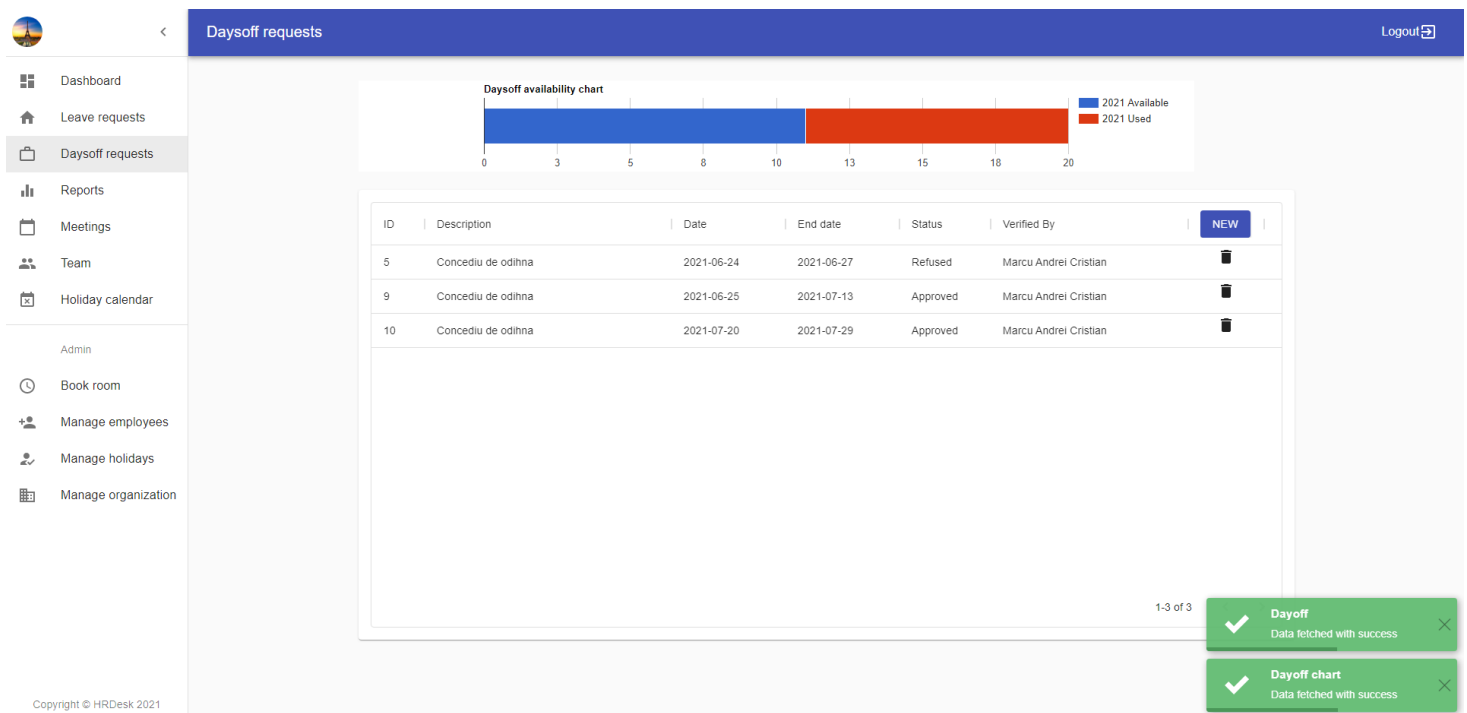


Figure 19. Dayoff module

5.4 Reports module

Is composed of three pie charts at the moment of the presentation. On this module there are already multiple tabs available for future use charts.

Right now there are three charts:

- One which displays the number of users between different ranges of age(ex 20-30, 30-40 etc).
- The second one displays the number of users that come from a certain country.
- And the last chart displays the number of users of each function.

Both the functions and the countries of provenience are dynamically retrieve in function of the data inserted, meaning that whenever a function or another country is added this will be automatically available in the charts.

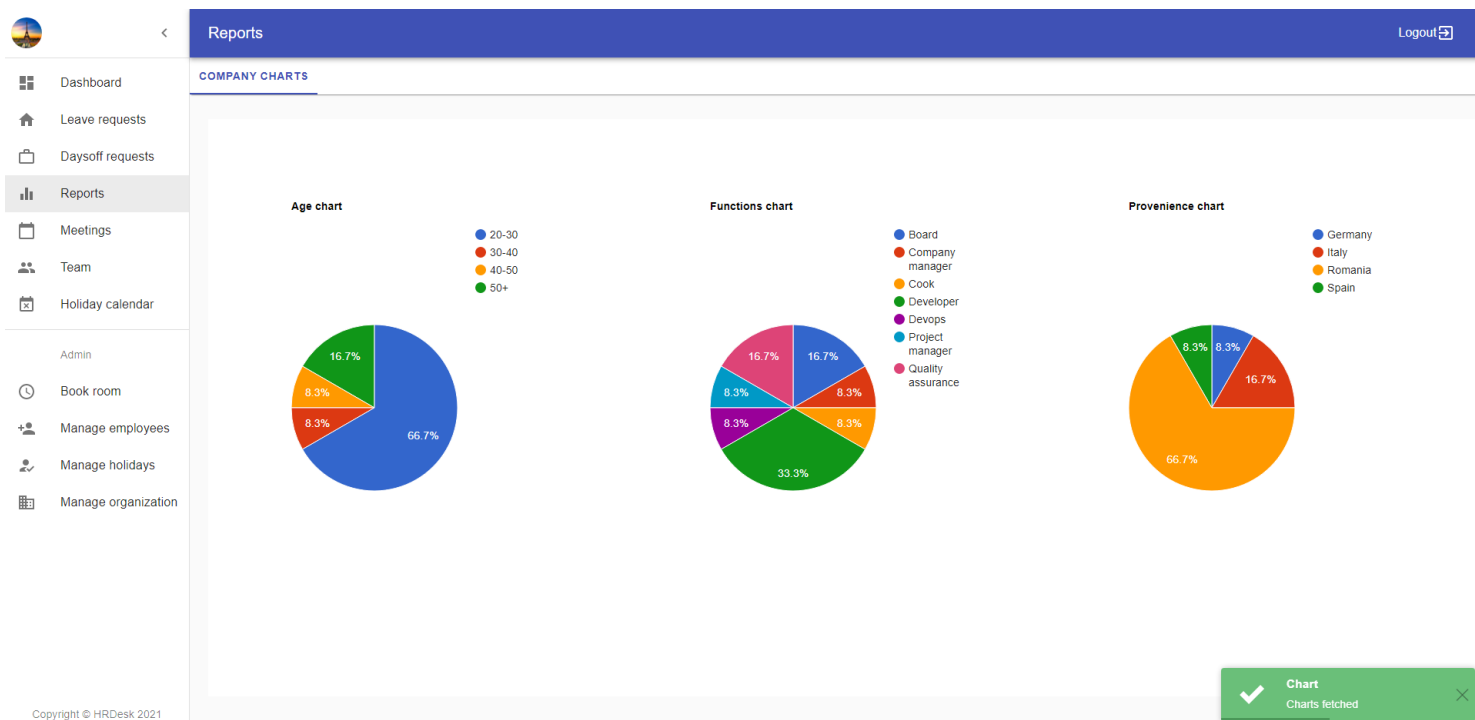


Figure 20. Reports module

5.5 Meetings module

It's more of an informative module, the user can see all the meetings between two dates (start and end) which are assigned to it's team. This module is only for display, seeing the meeting room where the call will take place and the description of the call. The meetings will be added from the Booking module which is on the administrative level

| ID | Title | Start date | End date | Room |
|----|--------------------|---------------------|---------------------|-------------------|
| 1 | Daily meeting | 2021-06-24 10:00:00 | 2021-06-24 11:00:00 | Main Meeting Room |
| 5 | New client meeting | 2021-06-24 12:00:00 | 2021-06-24 15:00:00 | Conference room 1 |
| 6 | Management call | 2021-06-24 13:00:00 | 2021-06-24 14:00:00 | Meeting Room 2 |
| 7 | Interview | 2021-06-24 15:00:00 | 2021-06-24 16:00:00 | Main Meeting Room |

1-4 of 4

Figure 21. Meetings module

5.6 Team module

Displays all the employees of the company ordered by the employment date. The entries are displayed in a mobile friendly way using Cards from Material UI. Each card is customized by the data of the user (picture, function, name, etc.).

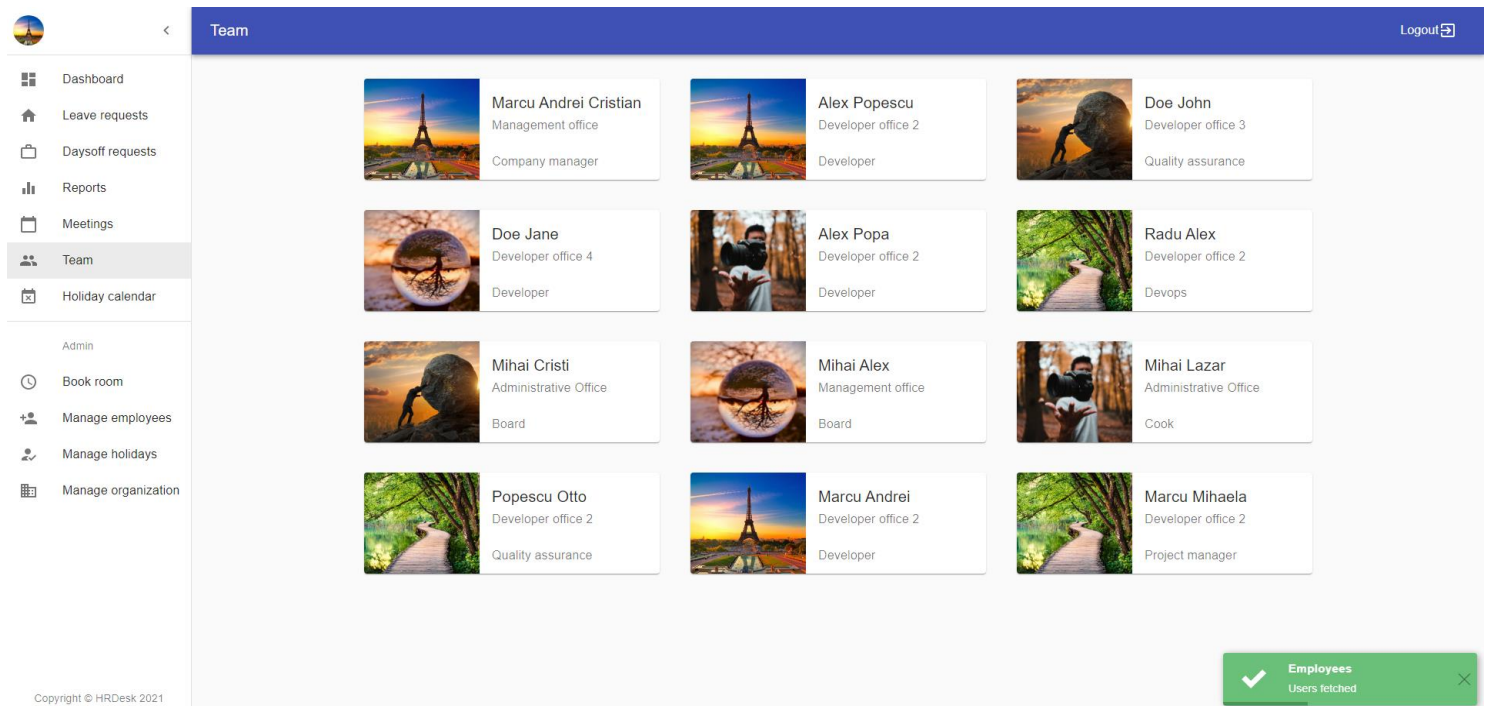


Figure 22. Team module

Clicking a card opens a more detailed view of the employee, where you can see non-sensitive data like Name, Office, Function, Email, Age, Date of Employment and Avatar.

Both the card and the employee dialog contain the picture of the user. The picture of the user is created by the administrator at the same time with the user, and stored in the database as a byte64 array. Then the picture is cropped by the Material-UI Avatar component.

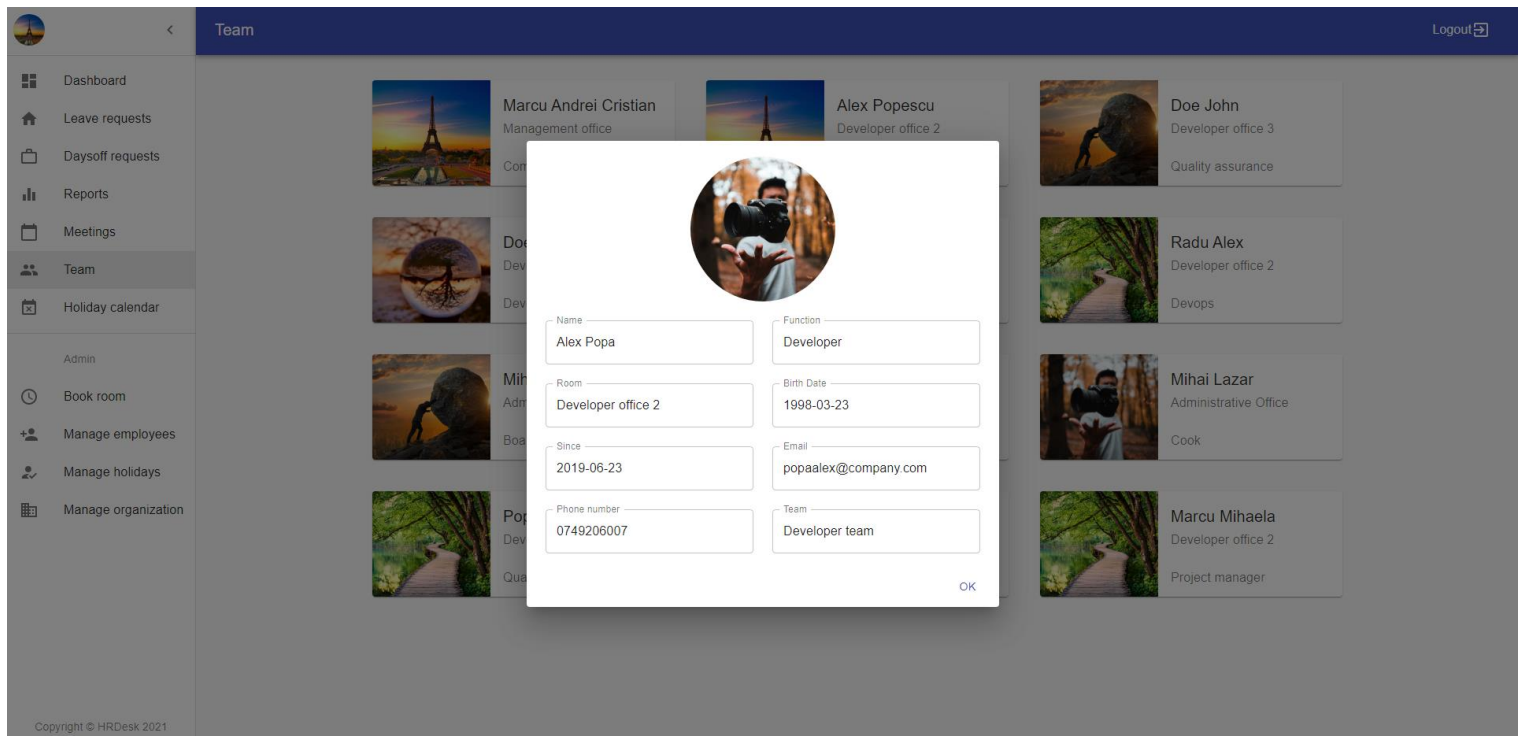
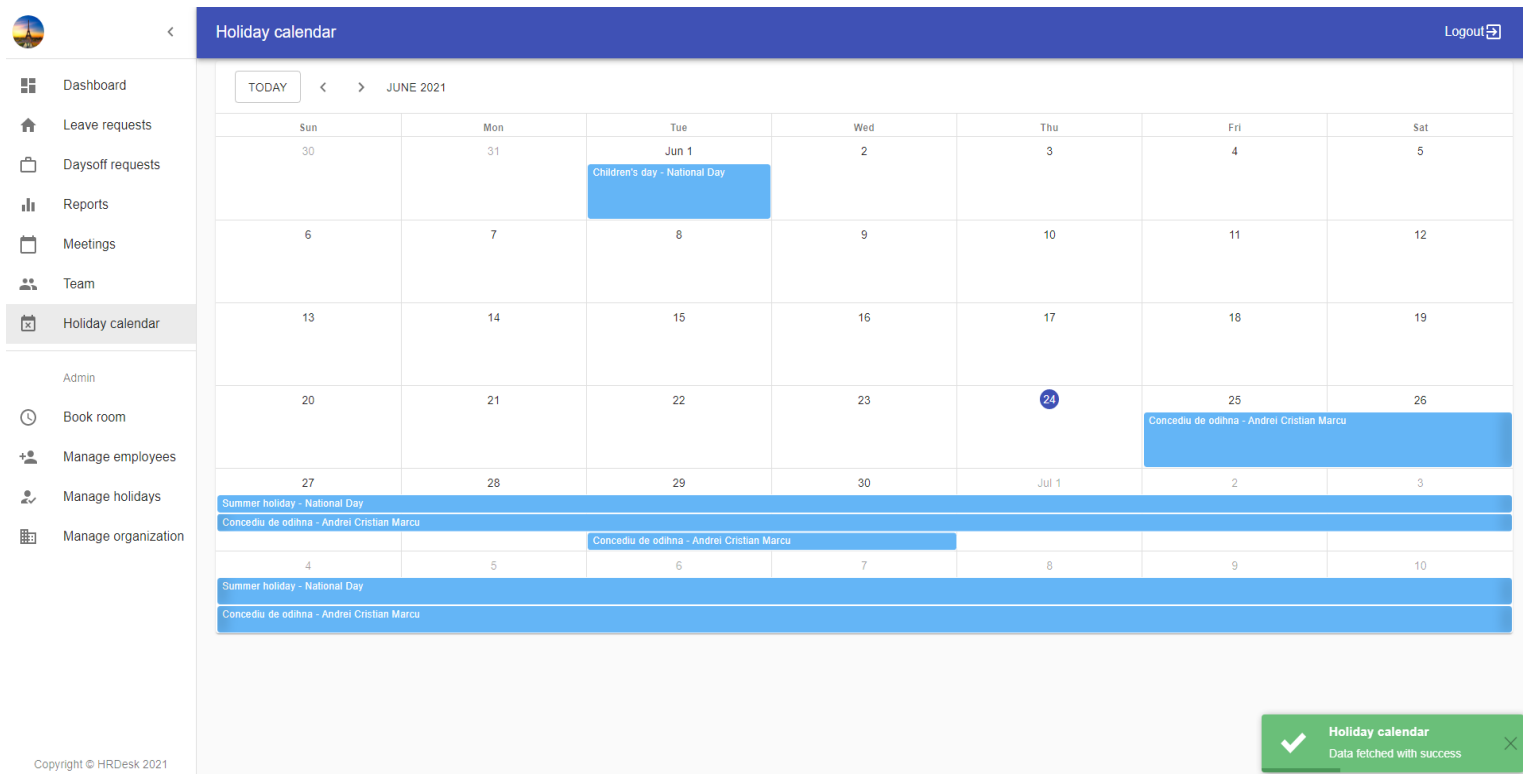


Figure 23. Team module - Employee details

5.7 Holiday Calendar module

The module displays a read-only calendar similar to the one available in Microsoft Outlook. This calendar displays three types of entries.

- The first ones are the days off of the current logged in user.
- This user is assigned to a team, so in the calendar the user will also see all the holidays of it's team members.
- In addition to this the national days defined from the administrative level are also displayed. This way the team can make a better development organization, by knowing everytime when a colleague is going to be missing, or when all the team will be gone and the sprint needs to be shorted.



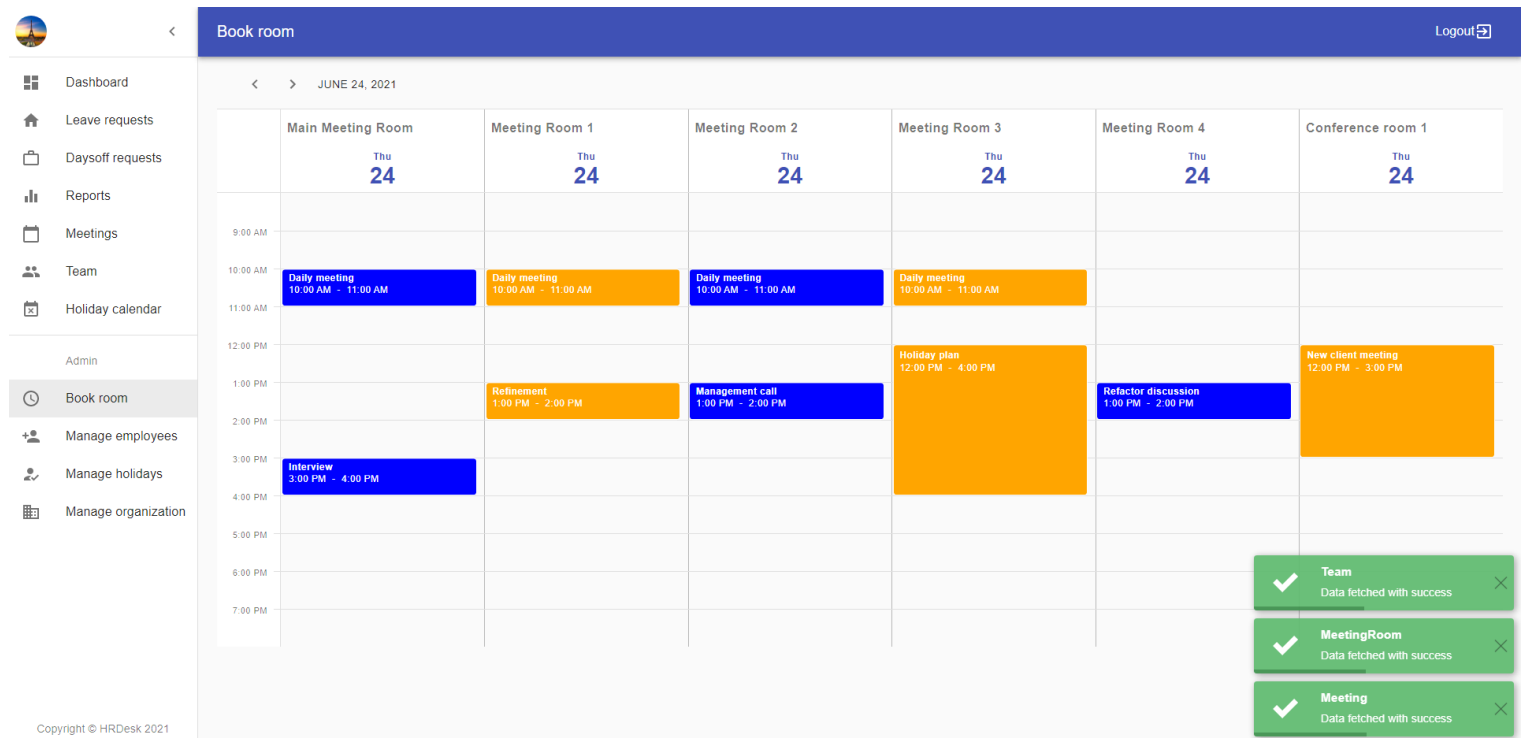


Figure 25. Booking module

5.9 Manage employees module

It's the main administrative level, in here all the employees are created, deleted or updated. The creation of an employee is made out of 4 steps.

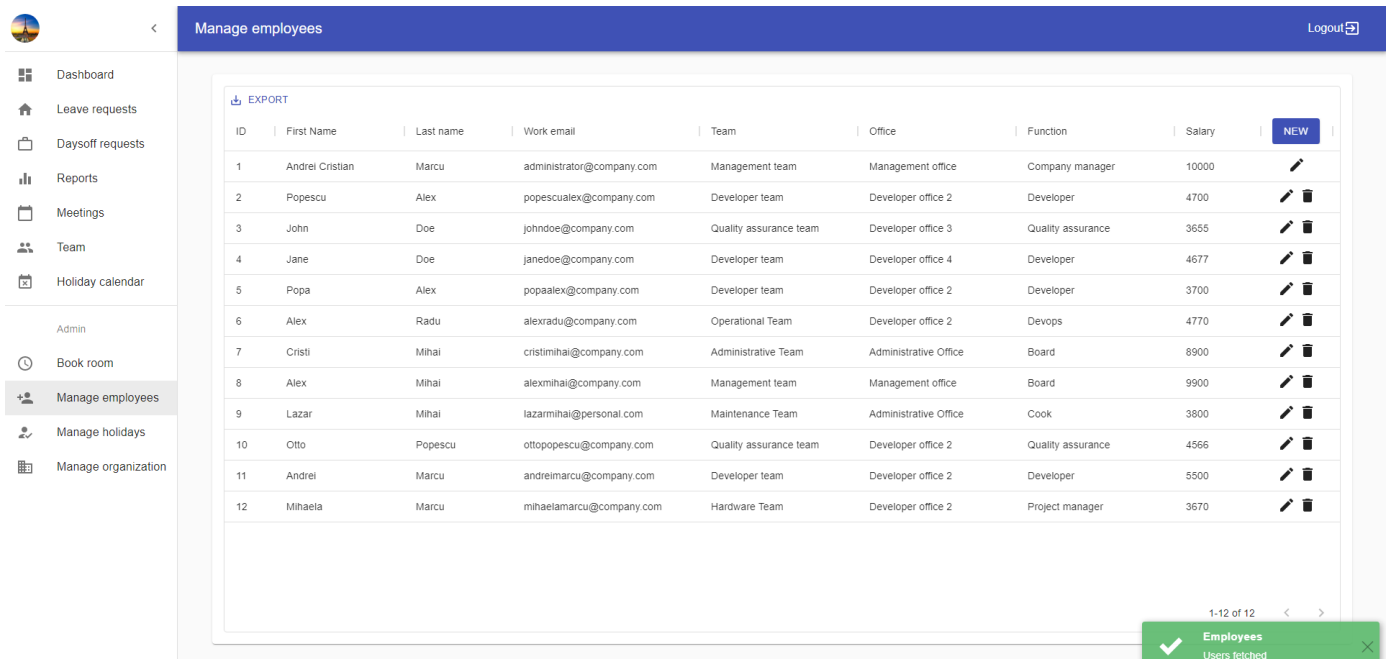


Figure 26. Manage employees module

The first step is adding personal data like Name, Email, Phone Number, Date of Birth, Country of Provenience.

Manage employees

Logout

1 Personal Data 2 Company Data 3 Picture 4 Permissions

First Name *
Andrei Cristian

Last Name *
Marcu

Email *
administrator@admin.com

Country of birth *
Romania

Phone *
0749206007

Address *
Company address

Date of birth *
05/28/1998

CNP *
198052811111

NEXT

Copyright © HRDesk 2021

Figure 27. Manage employees module step 1

The second step contains company specific data like Office, Function, Team, Salary, Number of free days, Work email, Account password.

Manage employees

Logout

1 Personal Data 2 Company Data 3 Picture 4 Permissions

Team *
Management team

Function *
Company manager

Office *
Management office

Password *

Number of daysoff *
20

Salary *
10000

Date of employment *
06/05/2021

Work email *
administrator@company.com

BACK NEXT

Figure 28. Manage employees module step 2

The third step is adding a picture of the user, after a picture is uploaded the user can see a preview of it.

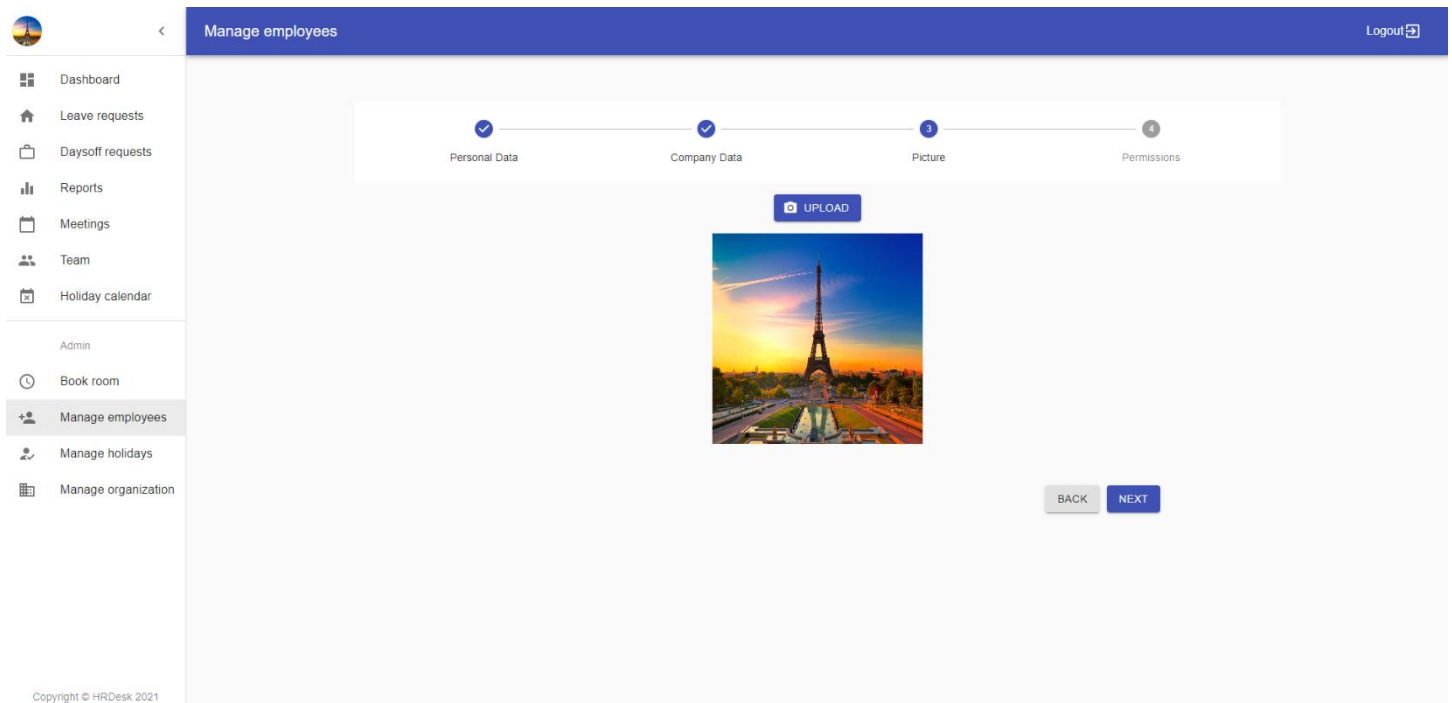


Figure 29. Manage employees module step 3

And the last step, the most important one is the module assigning step. In here the administrator can select which modules of the application the user will be able to see.

If the user is missing a module and he tries to access that page in any way(ex: from url) he will be automatically redirected to the Access Denied page. Also if an used tries to make a request to a module controller that is missing permission he will receive an Unauthorized Http Code.

Between all the steps the data is changed at any time, meaning that if we go back to a step, when we come back the data will always be saved. This way the user can go back and make changes without losing the work.

The modules of an user can be modified at any time by the administrator.

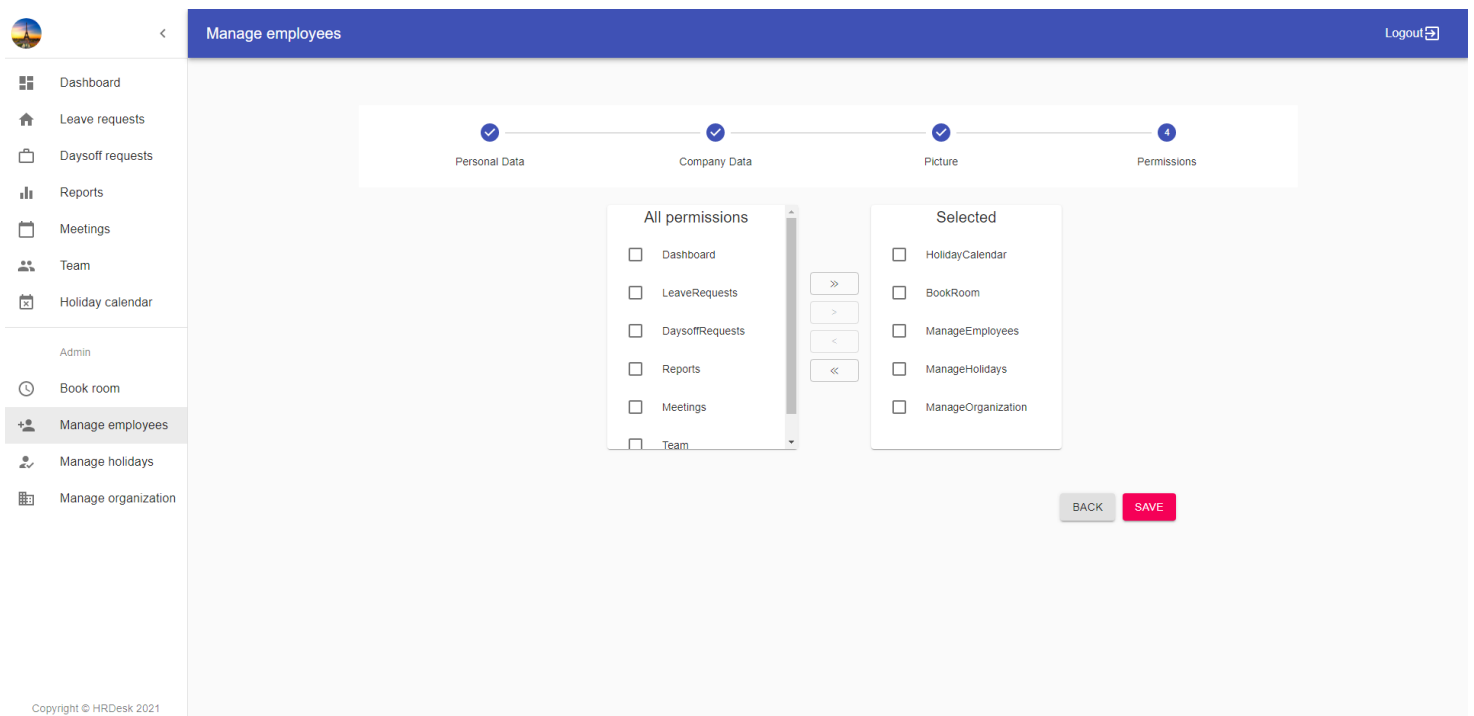


Figure 30. Manage employees module step 4

5.10 Manage holidays module

It's the administrative module of the requests. It strongly related to the Module 2 and Module 3. In here the person marked as "Administrator" when a leave request or day off is created will see all it's pending request.

There are two button for each request: "Accept" and "Decline". If a leave request is accepted or declined this will be automatically reflected in the status of the request from Module 2 and Module 3. If a request was accepted and the user changed his mind he can always change it's option. In addition when a day off is accepted the number of free days of the user is automatically calculated.

| ID | User | Date | Start hour | End hour | Status | Description | | |
|----|-----------------------|------------|------------|----------|----------|----------------|--------|---------|
| 6 | Doe John | 2021-06-21 | 08:30:00 | 17:00:00 | Waiting | Work from home | ACCEPT | DECLINE |
| 7 | Doe John | 2021-06-22 | 08:30:00 | 17:00:00 | Refused | Work from home | ACCEPT | |
| 12 | Doe John | 2021-06-29 | 08:30:00 | 17:00:00 | Refused | Work from home | ACCEPT | |
| 13 | Mihai Alex | 2021-06-24 | 08:30:00 | 17:00:00 | Refused | Work from home | ACCEPT | |
| 14 | Mihai Alex | 2021-06-25 | 08:30:00 | 17:00:00 | Approved | Work from home | | DECLINE |
| 15 | Mihai Alex | 2021-06-23 | 08:30:00 | 17:00:00 | Approved | Work from home | | DECLINE |
| 16 | Mihai Alex | 2021-06-22 | 08:30:00 | 17:00:00 | Approved | Work from home | | DECLINE |
| 17 | Mihai Alex | 2021-06-21 | 08:30:00 | 17:00:00 | Approved | Work from home | | DECLINE |
| 24 | Marcu Andrei Cristian | 2021-06-28 | 13:00:00 | 14:00:00 | Approved | Laborator | | DECLINE |

Figure 31. Manage holidays module leave requests

There is one extra tab on this module, the National Days tab, which defines the free days across all the company employees. This national days are displayed in each user's holiday calendar.

| ID | Description | Start date | End date | CreationDate | |
|----|-------------------|------------|------------|--------------|-----|
| 1 | Easter holiday | 2021-05-02 | 2021-05-04 | 2021-06-24 | NEW |
| 2 | Christmas holiday | 2021-12-22 | 2022-01-04 | 2021-06-24 | NEW |
| 3 | Children's day | 2021-06-01 | 2021-06-02 | 2021-06-24 | NEW |
| 4 | Summer holiday | 2021-06-27 | 2021-07-14 | 2021-06-24 | NEW |

Figure 32. Manage holidays module national days

5.11 Manage organization

It's also an important administrative module. In here the administrator can create, update or delete it's own company details. There are three tabs:

- Teams where all the teams are defined(the teams are used all over the application, for example when creating a meeting or an user).
- Functions tab is also needed when creating an user. Whenever we add a Function the Functions chart from Module 4 is automatically updated
- Rooms tab contains the information about offices and meeting rooms

For all these tabs there are some predefined entries created by the DbContext onModelCreating seed method, which is called the first time when a migration is ran on the database. There is also an user with all the modules enabled created on startup. The credentials for this user will be provided later in the presentation.

5.12 Unauthorized access

The routing is strongly protected by the permissions, meaning that if the user tries to access a module which it's missing from it's own attributions he will always end on the Access Denied page.

5.13 Login page

The login page doesn't have a register possibility, the only person that can create an user is the administrator or a person with Module 9. When the user logs in he has the possibility to check the remember me option which will store it's token to the localStorage instead of sessionStorage. For the time being the token is valid a month. If the token expires the user is logged out of the application.

6 UML DIAGRAMS

- Convenții de redactare

În această secțiune sunt detaliate convențiile de urmat în timpul editării lucrării.

6.1 Cerințe generale

Cerințele generale sunt preluate din [Olt07] (lucrare pe care o recomandăm călduros spre citire candidaților noștri, înainte de demararea redactării tezei): „responsabilitatea tezei este în întregime a candidatului, atât în ceea ce privește conținutul, cât și forma. Lucrarea trebuie să aibă o organizare clară și riguroasă, care să dovedească gândirea inginerească a candidatului. Ideile exprimate în lucrare trebuie să se înlănțuie conform unei logici clare. În acest sens, elementele de coerență și de coeziune a textului trebuie folosite în mod corect. Ideile se organizează în paragrafe, redactate cu indentație și fără spațiu între ele. Nu fraza creează paragraful, ci ideea!

Stilul este extrem de important („Stilul este veșmântul gândului,” spune Samuel Johnson). Lucrarea trebuie redactată într-un limbaj științific adecvat domeniului de cercetare abordat. Se vor evita particularitățile limbajului colocvial. Nu sunt admise greșeli gramaticale de redactare (acord, punctuație, lexic etc.). Lucrarea normativă ce va fi avută în vedere în această privință este Dicționarul ortografic, ortoepic și morfologic al limbii române [DOOM05].

Candidatul are obligația de a verifica dacă datele, termenii folosiți, numele proprii, citatele, titlurile (în limba română și în alte limbi) sunt corecte.

Candidatul trebuie să fie consecvent în exprimarea ideilor, în folosirea termenilor, a numelor proprii, a datelor, precum și a punctuației și a elementelor de structură a lucrării. Consecvența este necesară și în privința tipurilor de evidențieri grafice folosite (litere cursive, litere îngroșate sau sublinieri).

Termenii tehnici de origine străină neadaptați, consacrați de lucrările de specialitate, nu se traduc, dar, dacă folosiți o sursă bibliografică străină, puteți încerca traducerea unor termeni noi, cu condiția ca cei din limba de origine să fie prezenți alături. În ambele cazuri, se recomandă scrierea acestor termeni cu litere speciale (de regulă italice).

La cele menționate mai sus, adăugăm următoarea observație, care nu este deloc lipsită de importanță: notarea semnelor diacritice românești este obligatorie. Un text românesc în care ă se confundă cu a nu face o impresie bună, dă o notă de neglijență.”

6.2 Structura documentului

Lucrarea trebuie să conțină următoarele capitole:

- Coperta tezei;
- Pagina de titlu;
- Declarația de originalitate (completată și semnată de către autor);
- Formularul de înregistrare a enunțului temei lucrării (completat și semnat în solidar de către autor și coordonatorul științific);
- Referatul coordonatorului științific (completat și semnat de coordonator);
- Declarația de mulțumire a autorului (opțională);
- Cuprinsul lucrării;
- Lista figurilor;
- Lista tabelelor;
- Introducere;
- Conținutul propriu-zis al lucrării (capitolele constitutive);
- Concluzii;
- Bibliografia;
- Referințele web;
- Anexele (index, codul sursă, site-ul web al aplicației, etc.).

6.3 Dimensiunile lucrării

Nu pot fi acceptate lucrări cu un număr de pagini mai mic decât 50. Prin urmare se dau, cu titlu de recomandare, următoarele dimensiuni:

- 50 – 80 de pagini pentru o lucrare de licență;
- 60 – 100 de pagini pentru o lucrare de disertație.

În altă ordine de idei, recomandăm următoarea distribuție a paginilor pe diversele secțiuni/capitole [Olt07]:

- Introducerea reprezintă cca. 10 – 15% din lucrare;

- Conținutul reprezintă cca. 75 – 80% din lucrare;
- Concluziile reprezintă cca. 10%.

6.4 Elemente de tehnoredactare

Pe scurt:

- dimensiunea paginilor va fi A4, 21 x 29,7 cm;
- marginile recomandate sunt: sus/jos/stânga/dreapta – 2,54 cm;
- fontul recomandat este Times New Roman;
- corpul literelor va avea dimensiunea de 11 puncte;
- spațiul dintre rânduri va avea dimensiunea de 1 rând și jumătate (1,5);
- indentarea unui paragraf se va face cu 1,27 cm;
- textul paragrafelor trebuie să fie aliniat în mod echilibrat stânga-dreapta (în eng. - *justify*);
- paginile trebuie să fie numerotate conform acestui șablon.

6.5 Formulele matematice

Pentru redactarea formulelor matematice recomandăm utilizarea instrumentului Microsoft Equation Editor și importul lor (o data terminate) în Microsoft Word.

6.6 Ilustrațiile

6.6.1 Figurile

Partea grafică a lucrării are o pondere semnificativă în nota finală acordată lucrării candidatului. Se recomandă prin urmare acordarea unei atenții sporite tehnoredactării figurilor.

Actualizarea Listei Figurilor este obligatorie (procedura este similară cu cea exemplificată în secțiunea 6.6.2).

6.6.2 Tabelele

Exemplificăm aici utilizarea tabelor. Pentru fiecare tabelă adăugată lucrării, autorul trebuie să prevadă și adăugarea unei legende (în eng., *Caption*). La final, este recomandată actualizarea listei figurilor. Pașii necesari pentru actualizarea listei tabelelor sunt:

1. Click dreapta pe Lista Tabelelor. Va apărea un meniu similar cu cel din Figura 1.

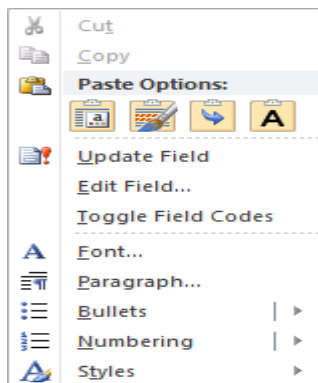


Figura 1. Selectarea prin click dreapta a opțiunii „Update field”

2. Selectarea opțiunii „Update entire table” conform cu Figura 2.

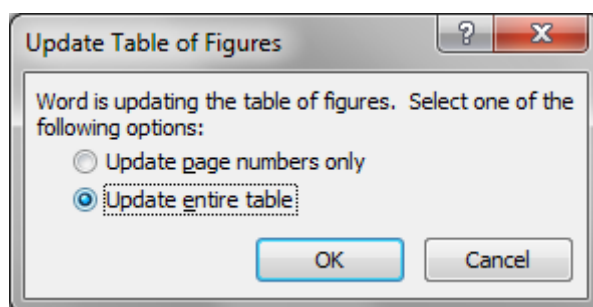


Figura 2. Actualizarea întregului tabel

3. Verificarea fontului folosit pentru conținutul propriu-zis al Listei Tabelelor și alegerea fontului Times New Roman în caz de incongruență.

| Index | Nume utilizator | Valoarea rezumat a parolei (folosind SHA1) |
|-------|-----------------|--|
| 1 | dpopescu | 8fb9e6763269ae7cba85f02668c3c32041bf00ed |
| 2 | eganea | 3b42ba8a586dd1589e949c28c9cf2810f7d65bb4 |
| 3 | mmarian | bfc01d16d1944f3b6caba515556713f4aeeb2d0b |

Tabelul 1. Nume de utilizatori și valorile rezumat ale parolelor acestora

În interiorul lucrării, tabelul poate fi citat folosind eticheta și numărul de ordine ale sale precum în exemplul acesta (vezi Tabelul 1).

6.6.3 Legenda (unei figuri/tabele)

În cele doua secțiuni de mai sus s-au demonstrat două modele de legende atașate fiecărei tabele sau figuri. Microsoft Word permite modificarea, respectiv adăugarea de etichete noi corespunzătoare unui anumit tip de legendă.

7 TERMENI DE UTILIZARE

7.1 Autorii

Acest șablon de document a fost creat de M.Marian pentru colectivul Facultății de Automatică, Calculatoare și Electronică a Universității din Craiova și pentru uniformizarea structurii proiectelor de diplomă ale studenților săi.

7.2 Licența de utilizare

Nu există restricții de utilizare. Documentul nu este constrâns de nicio licență. Sugestiile de îmbunătățire pot fi adresate autorului pe adresa de e-mail: marius.marian@cs.ucv.ro.

8 CONCLUZII

Autorul prezintă concluziile sale...

9 BIBLIOGRAFIE

[ANDR19] – *ASP.NET Core in Action – First edition*, Andrew Lock, Manning Publications, 2019

[JUL10] – *Programming Entity Framework, 2nd Edition*, Julia Lerman, O'Reilly Media, Inc., 2010

10 REFERINȚE WEB

- [1] ASP.NET Core logo - https://commons.wikimedia.org/wiki/File:.NET_Core_Logo.svg
- [2] ASP.NET Core Wikipedia - https://en.wikipedia.org/wiki/ASP.NET_Core
- [3] Microsoft – Introduction to NuGet - <https://docs.microsoft.com/en-us/nuget/what-is-nuget>
- [4] Netsolutions – ASP.NET Core vs ASP.NET - <https://www.netsolutions.com/insights/net-core-vs-net-framework/>
- [5] Entity Framework Core Wikipedia - https://en.wikipedia.org/wiki/Entity_Framework
- [6] Entity Framework Code First Approach – EntityFrameworkTutorial - <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>
- [7] React (JavaScript library) - [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- [8] React logo - [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)#/media/File:React-icon.svg](https://en.wikipedia.org/wiki/React_(JavaScript_library)#/media/File:React-icon.svg)
- [9] What is JSX? - <https://www.reactenlightenment.com/react-jsx/5.1.html>
- [10] React Redux logo - <https://commons.wikimedia.org/wiki/File:Redux.png>
- [11] SQL Server Wikipedia RO - https://ro.wikipedia.org/wiki/Microsoft_SQL_Server
- [12] SQL Server Wikipedia EN - https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [13] GitHub logo - https://en.wikipedia.org/wiki/GitHub#/media/File:GitHub_logo_2013.svg
- [14] GitHub wikipedia - <https://en.wikipedia.org/wiki/GitHub>

A. CODUL SURSĂ

În această anexă se adaugă codul sursă al aplicației...

B. SITE-UL WEB AL PROIECTULUI

Autorul prezintă în această anexă (opțională) site-ul web asociat proiectului său.

C. CD / DVD

Autorul atașează în această anexă obligatorie, versiunea electronică a aplicației, a acestei lucrări, precum și prezentarea finală a tezei.



INDEX

B

Bibliografie..... 9

C

CUPRINSUL..... xi

D

Dimensiuni 3

F

Figuri 4

Formulele matematice 4

I

Ilustrațiile..... 4

L

Legenda..... 6

LISTA FIGURILORxii

LISTA TABELELORxiii

R

Referințe web 10

S

Structura documentului 2

T

Tabele 5