

**DELTA** – Střední škola informatiky a ekonomie,  
Základní škola a Mateřská škola s.r.o.

**Ke Kamenci 151, PARDUBICE**



## **Agenda pro rezervaci sportovišť**

Příjmení, jméno: Elsnic Tomáš

Studijní obor: Informační technologie

Třída: 4.B

Školní rok: 2018-2019

# Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: Tomáš Elšnic  
Školní rok: 2018/2019  
Třída: 4. B  
Obor: Informační technologie 18-20-M/01

Téma práce: **Agenda pro rezervaci sportovišť**  
Vedoucí práce: Bc. Vlad'ka Janů

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Student vytvoří informační systém, systém bude sloužit na rezervaci sportovních hřišť. Uživatel se bude moci přihlásit pod svým účtem na kterém bude mít kredit díky kterému si pak zarezervuje hřiště, každé hřiště bude mít jinou cenu a kredit si bude moci uživatel zakoupit od admina který vlastní hřiště. Systém bude webová aplikace, která bude obsahovat registrace uživatelů, rezervace, možnost zrušit rezervaci, přidat kredit a také administrátorský účet který bude moci přidávat měnu, měnit rezervace, přidávat nová hřiště nebo banovat uživatele. Student použije k realizaci technologie jako jsou např. SP.NET MVC, MS SQL.

Stručný časový harmonogram (s daty a konkretizovanými úkoly)

Do 28.9

Student navrhne všechny usecase (GUI).

Do 10.10

Student dokončí db model.

Do 25.11

Student dokončí GUI pro uživatele, rezervace, čas pro rezervace, registraci.

Do 16.12

Student dokončí GUI pro kredit a administrátorský účet

Prohlašuji, že jsem maturitní projekt vypracoval samostatně s použitím uvedené literatury.

V Pardubicích 27.03.2019

Upřímně děkuji Bc. Vladěce Janů za studijní materiál a odborné vedení při zpracování maturitního projektu.

## Anotace

### Klíčová slova

React, Nestjs, MYSQL, informační systém, web

### Keywords

React, Nestjs, MYSQL, information system, web

### Resumé

Jako maturitní práci byl zvolen informační systém nazvaný agenda rezervace hřišť. Systém by měl sloužit jako stránka kde se uživatel zaregistruje a pod svým účtem si může vybrat hřiště které by chtěl zarezervovat, na jaký čas a také za kolik. Je zde i administrátorský režim který umožňuje administrátorovi přidávat, upravovat, mazat nebo editovat hřiště.

As a final project, an information system called the agenda pro rezervaci sportovišť was chosen. The system should serve as a site where the user registers and under his account can choose a field he would like to book, at what time and for how much. There is also an administrator mode that allows the administrator to add, edit and delete a field.

## Obsah

|                                                             |    |
|-------------------------------------------------------------|----|
| Zadání maturitního projektu z informatických předmětů ..... | 2  |
| Anotace.....                                                | 5  |
| Klíčová slova .....                                         | 5  |
| Keywords .....                                              | 5  |
| Resumé.....                                                 | 5  |
| Úvod .....                                                  | 7  |
| 1 Cíle práce .....                                          | 8  |
| 2 Výběr technologií pro řešení .....                        | 9  |
| 2.1 Frontend.....                                           | 9  |
| 2.1.1 React.....                                            | 9  |
| 2.1.2 Nodejs.....                                           | 9  |
| 2.1.3 JSX.....                                              | 9  |
| 2.1.4 Webpack.....                                          | 9  |
| 2.2 Backend .....                                           | 9  |
| 2.2.1 Nestjs.....                                           | 9  |
| 2.3 MYSQL .....                                             | 10 |
| 2.4 Visual Studio Code.....                                 | 10 |
| 2.5 XAMP .....                                              | 10 |
| 2.6 CSS .....                                               | 10 |
| 2.7 Npm .....                                               | 10 |
| 3 Pojmy .....                                               | 11 |
| 3.1 databáze .....                                          | 11 |
| 3.2 Framework .....                                         | 11 |
| 3.3 server-side aplikace .....                              | 11 |
| 3.4 Typescript .....                                        | 11 |
| 3.5 Middleware .....                                        | 11 |
| 4 Způsoby řešení a postupy .....                            | 12 |
| 4.1 Vytvoření Frontendu .....                               | 12 |
| 4.2 Formulář .....                                          | 12 |
| 4.1.1 Register formulář.....                                | 12 |
| 4.1.2 Login formulář .....                                  | 13 |

|                                        |    |
|----------------------------------------|----|
| 4.3 Vytvoření backendu.....            | 13 |
| 4.4 Struktura backendu .....           | 14 |
| 4.4.1 Controller.....                  | 14 |
| 4.4.2 Model .....                      | 14 |
| 4.5 Vytvoření uživatele.....           | 15 |
| 4.5.1 User entity .....                | 15 |
| 4.6 Autentizace uživatele .....        | 16 |
| 4.6.1 Passport strategy.....           | 16 |
| 4.7 Hřiště a vše kolem nich.....       | 17 |
| 4.7.1 Výpis a zobrazení.....           | 17 |
| 4.7.2 Přidávání nového hřiště.....     | 17 |
| 5 Příručka k použití .....             | 18 |
| 5.1 Registrace a přihlášení.....       | 18 |
| 5.2 Hřiště a ovládání.....             | 19 |
| 5.2.1 Přidání nového hřiště .....      | 20 |
| 5.2.2 Editace hřiště.....              | 20 |
| 6 Zhodnocení dosažených výsledků ..... | 21 |
| 6.1 Splněné cíle .....                 | 21 |
| 6.2 Nesplněné cíle .....               | 21 |
| Závěr .....                            | 22 |
| Zdroje.....                            | 23 |
| Seznam obrázků.....                    | 25 |
| Bibliografie.....                      | 25 |

## Úvod

V tomto dokumentu bude představen webový informační systém, který žák vypracoval. Účelem toho dokumentu je tedy uživateli představit projekt a přinést informace o použitých technologiích. Projekt byl zkrácen, protože autor byl kvůli závažné nemoci 2 měsíce nepřítomen, je také psán pomocí jiných technologií, než které byli zvoleny na začátku projektu, těmito technologiemi jsou React a Nestjs a pro ukládání dat databáze MYSQL. React byl zvolen, protože ASP.Net se v dnešní době už skoro nepoužívá a většina webů vzniká právě pomocí Reactu. Na následujících stránkách bude rozebrána tvorba projektu od tvorby formuláře až po programování zobrazení, přidávání a úpravu dat.

## 1 Cíle práce

Vytvoření webové informačního systému s využitím technologií React a Nestjs.

Navržení struktury databáze

Navržení struktury stránky a zobrazení dat

Navržení principů filtrování a úpravy dat.

## 2 Výběr technologií pro řešení

Zde je popis využitých programů a technologií, které byli použity.

### 2.1 Frontend

#### 2.1.1 React

React je JavaScriptová knihovna pro vytváření webových komponent která řeší pouze UI vrstvu a data drží v jednotlivých komponentách. Obsahuje JSX což je rozšíření pro JavaScript podobné HTML, které zpřjemňuje čitelnost komponent a Virtual DOM, který reprezentuje reálné prvky v paměti DOM, které jsou generovány komponentami React ještě předtím, než se provedou změny na stránce. Zvolen pro frontend díky jeho dostupnosti, jednoduchosti a možnostem dalšího rozšíření.

#### 2.1.2 Nodejs

Node.js je prostředí runtime jazyka JavaScript na straně serveru, který se spustí JavaScript na straně serveru.

#### 2.1.3 JSX

JSX je rozšíření syntaxe jazyka JavaScript, obvykle používaný k popisu prvků uživatelského rozhraní pomocí React.

#### 2.1.4 Webpack

Webpack zabaluje Javascriptové soubory, aby se dali spustit prohlížeči. Často se používá k určení kompilátoru, jako je například Babel nebo TypeScript, transpiluje kód JSX nebo TypeScript pro prostý JavaScript.

### 2.2 Backend

#### 2.2.1 Nestjs

Nestjs je framework pro vytváření server-side aplikací. Je vestavěn typescriptem a kombinuje elementy OOP (objektově orientovaného programování) a také nabízí možnosti využití dalších knihoven díky jeho struktuře. Využívá také dependenci injection což je technika pro vkládání závislostí mezi jednotlivými komponentami programu tak, aby jedna komponenta mohla používat druhou, aniž by na ni měla v době sestavování programu referenci. V projektu je využíván pro řešení backendu.

## 2.3 MYSQL

MySQL je systém řízení báze dat uplatňující relační databázový model, je to multiplatformní databáze a komunikace s ní probíhá pomocí jazyka SQL. Podobně jako u ostatním SQL databázích se jedná o dialekt toho jazyka s některými rozšířeními. MySQL byl vybrán pro svou snadnou implementovatelnost, výkon, a především protože se jedná o volně šiřitelný software.

## 2.4 Visual Studio Code

Visual Studio Code je bezplatný nástroj pro vývojáře. Jedná se o editor zdrojového kódu, který doplňuje rodinu produktů Visual Studio. Vybrán pro editaci veškerého kódu.

## 2.5 XAMP

Multiplatformní softwarový balíček vyvinutý firmou Apache Friends. Obsahuje volně dostupný otevřený software a serverovou aplikaci (Apache), databázi (MySQL), a skriptovací jazyk (PHP). Tento balíček byl vybrán pro vytvoření lokálního serveru pro testování frontendu a backendu.

## 2.6 CSS

Jazyk, který udává způsob zobrazení a vzhled stránek. Použit na stylování stránek a formuláře.

## 2.7 Npm

Npm (Node.js package manager) je správce balíčků pro Javascript a výchozí správce balíčků pro prostředí Node.js.

## 3 Pojmy

### 3.1 databáze

Databáze je systém souborů s pevnou strukturou záznamů. Tyto soubory jsou mezi sebou navzájem propojeny pomocí klíčů. Tento software se v české odborné literatuře nazývá systém řízení báze dat.

### 3.2 Framework

Framework je softwarová struktura, která slouží jako podpora při programování, vývoji a organizaci jiných softwarových projektů.

### 3.3 server-side aplikace

Aplikace, která běží na straně serveru a uživatel k ní má obvykle přístup pouze přes aplikační rozhraní (API), tudíž nevidí zdrojový kód.

### 3.4 Typescript

TypeScript je open-source programovací jazyk vytvořený a spravovaný firmou Microsoft. Rozšiřuje javascript o statické typování a další atributy.

### 3.5 Middleware

Specializovaný softwarem spojují softwarové komponenty nebo podnikové aplikace.

## 4 Způsoby řešení a postupy

Tato kapitola pojednává o tom, jak projekt vznikal a jaké byly použity řešení.

### 4.1 Vytvoření Frontendu

Začátkem bylo vytvoření klasického React projektu tím, že se založila složka, kde se bude projekt vytvářet. Projekt se pak nainstalovat pomocí jednoduchého příkazu který se zadal do příkazového řádku.

***Npm install -g react-native-cli***

Pro spuštění frontendu se pak používá příkaz ***Npm start***.

### 4.2 Formulář

Před začátkem programování byl navrhnout formulář, přes který se uživatel zaregistruje a přihlásí. Pro formulář byl vybrán zeleno-černý styl od kterého se bude následovně odvíjet i celá aplikace. Formulář je vlastnoručně nastýlován nepoužívá tedy žádný bootstrap.

#### 4.1.1 Register formulář

Formulář pro registraci obsahuje jméno uživatele, e-mailovou adresu a heslo.

```
render() {
  return (
    <div className="App">
      <div className="App_Aside"></div>
      <div className="App_Form">
        <div className="PageSwitcher">
          <NavLink to="/" activeClassName="PageSwitcher_Item--Active" className="PageSwitcher_Item">Sign In</NavLink>
          <NavLink exact to="/signup" activeClassName="PageSwitcher_Item--Active" className="PageSwitcher_Item">Sign Up</NavLink>
        </div>

        <div className="FormTitle">
          <NavLink to="/" activeClassName="FormTitle_Link--Active" className="FormTitle_Link">Sign In</NavLink> or <NavLink exact to="/" activeClassName="FormTitle_Link--Active" className="FormTitle_Link">Sign Up</NavLink>
        </div>
        <div className="FormCenter">
          <form className="FormFields">
            <div className="FormField">
              <label className="FormField_Label" htmlFor="email">E-Mail Address</label>
              <input type="email" id="email" className="FormField_Input" placeholder="Enter your email" name="email" value={this.state.email} onChange={this.handleChangeEmail}/>
            </div>

            <div className="FormField">
              <label className="FormField_Label" htmlFor="password">Password</label>
              <input type="password" id="password" className="FormField_Input" placeholder="Enter your password" name="password" value={this.state.password} onChange={this.handleChangePassword}/>
            </div>

            <div className="FormField">
              <button onClick={() => this.handleSubmit()} type="button" className="FormField_Button mr-20">Sign In</button>
              <Link to="/signup" className="FormField_Link">Create an account</Link>
            </div>
          </form>
        </div>
      </div>
    </div>
  );
}

export default withRouter(SignInForm);
```

Obrázek 1- Technická část register formuláře [1]

### 4.1.2 Login formulář

Přihlašovací formulář obsahuje stejná data jako registrační kromě toho, že uživatel nemusí pro přihlášení zadávat jméno.

```
render() {
  return (
    <div className="App">
      <div className="App_Aside"></div>
      <div className="App_Form">
        <div className="PageSwitcher">
          <NavLink to="/" activeClassName="PageSwitcher_Item--Active" className="PageSwitcher_Item">Sign In</NavLink>
          <NavLink exact to="/signup" activeClassName="PageSwitcher_Item--Active" className="PageSwitcher_Item">Sign Up</NavLink>
        </div>

        <div className="FormTitle">
          <NavLink to="/" activeClassName="FormTitle_Link--Active" className="FormTitle_Link">Sign In</NavLink> or <NavLink exact to="/" activeClassName="Fo
        </div>

        <div className="FormCenter">
          <form onSubmit={this.handleSubmit} className="FormFields">
            <div className="FormField">
              <label className="FormField_Label" htmlFor="name">Full Name</label>
              <input type="text" id="name" className="FormField_Input" placeholder="Enter your full name" name="name" value={this.state.name} onChange={this.handleCha
            </div>
            <div className="FormField">
              <label className="FormField_Label" htmlFor="password">Password</label>
              <input type="password" id="password" className="FormField_Input" placeholder="Enter your password" name="password" value={this.state.password} onChange=
            </div>
            <div className="FormField">
              <label className="FormField_Label" htmlFor="email">E-Mail Address</label>
              <input type="email" id="email" className="FormField_Input" placeholder="Enter your email" name="email" value={this.state.email} onChange={this.handleCha
            </div>
            <div className="FormField">
              <button className="FormField_Button mr-20">Sign Up</button> <Link to="/sign-in" className="FormField_Link">I'm already member</Link>
            </div>
          </form>
        </div>
      </div>
    </div>
  );
}

export default SignUpForm;
```

Obrázek 2-Struktura login formuláře [1]

### 4.3 Vytvoření backendu

Ve složce projektu byla vytvořena další složka backend do které se pomocí podobného příkazu následovně nainstalovat Nestjs pomocí následujících příkazů.

***Npm i -g @nestjs/cli***

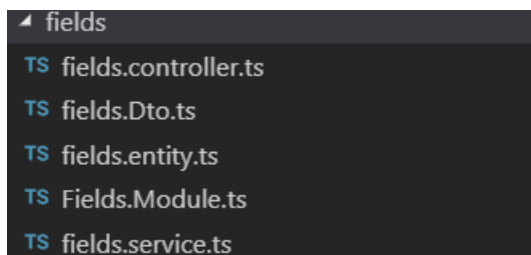
***Nest new backend***

***Git clone https://github.com/nestjs/typescript-starter.git backend***

***Npm install***

## 4.4 Struktura backendu

Pro hřiště byla vytvořena složka pod názvem fields která obsahuje 5 komponent.



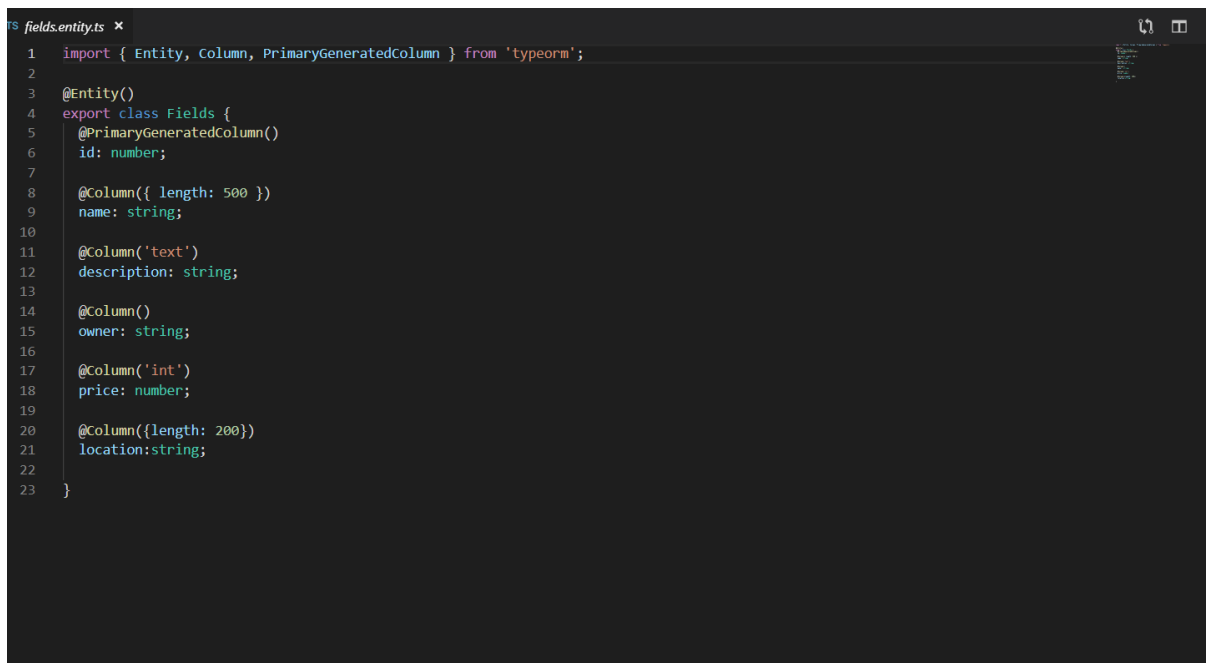
Obrázek 3-Backend [1]

### 4.4.1 Controller

Controller má na starost, aby se aktualizoval model je tedy jakousi ústřední výkonnou jednotkou, která se stará o celkové provázání a funkčnost aplikace.

### 4.4.2 Model

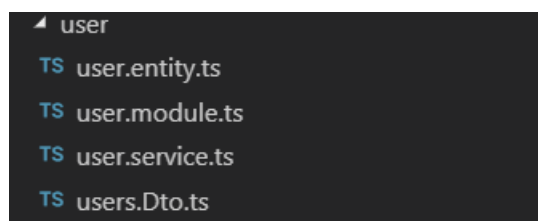
Model slouží jako datová struktura, která odpovídá databázi (abstrakt databázové tabulky). Pro vytvoření modelu byl použit TypeORM. TypeORM je ORM které může běžet na Node.js, React-native a na spoustu dalších platformách dá se také používat s TypeScriptem nebo JavaScriptem. Umožňuje vytvářet databázové tabulky, mazat, vkládat, updatovat a hledat data bez používání těžce udržitelných SQL dotazů.



Obrázek 4-Model [1]

## 4.5 Vytvoření uživatele

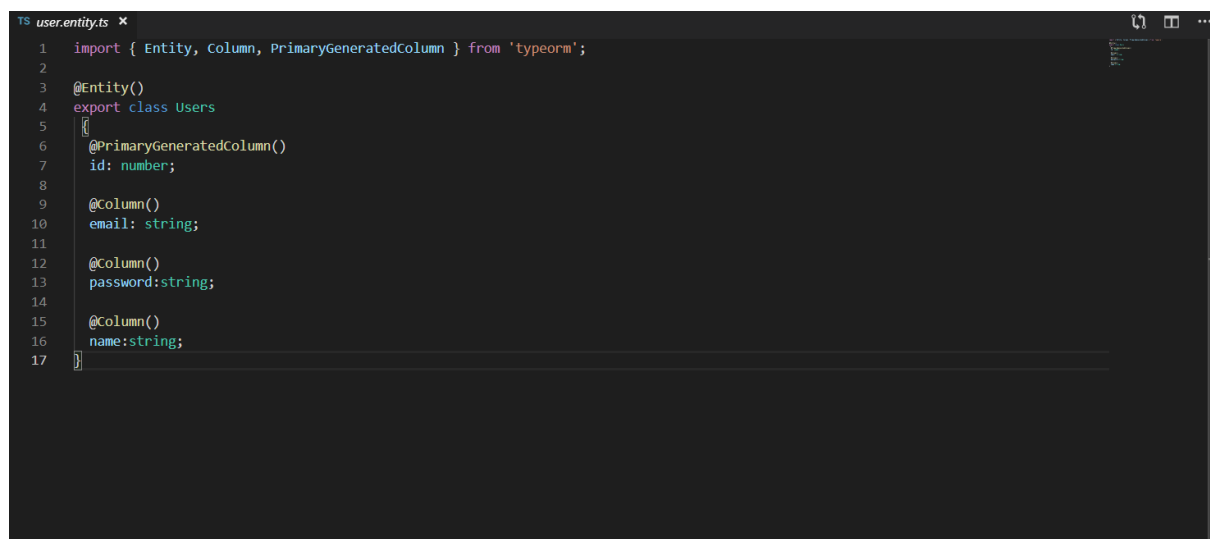
Pro vytvoření uživatele byla vytvořena další samostatná složka users obsahující 4 komponenty.



Obrázek 5-Uživatel [1]

### 4.5.1 User entity

Určuje, jaké bude mít uživatel hodnoty, jak se bude zapisovat do databáze.



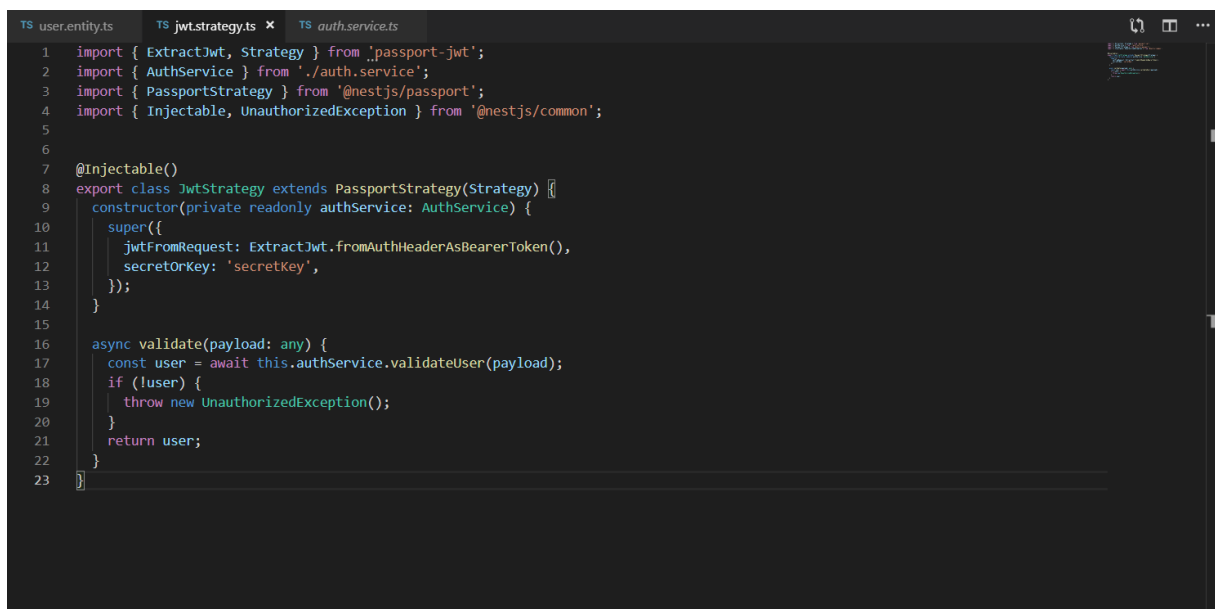
Obrázek 6-User entity [1]

## 4.6 Autentizace uživatele

U aplikace bylo důležité, aby při přidávání hřišť měl uživatel vlastní identitu a nemohl se vydávat ze někoho jiného, jinak by uživatel mohl přidávat nová pole a měnit si ceny podle sebe i za jiné uživatele, a proto byla vytvořena autentizace uživatele, která mu zajišťuje ochranu před falšováním identity.

### 4.6.1 Passport strategy

Pro autentizaci byla použita JWT strategie knihovny passport. JWT (JSON Web Token) je standard který definuje samostatný bezpečný způsob pro přenos informací mezi stranami jako JSON objekt. Passport je ověřovací middleware který je v projektu použit pro ověřování požadavků. Jwt strategii si tedy můžeme vysvětlit tak že pro autorizaci bude passport používat pouze jwt token.



```
TS user.entity.ts TS jwt.strategy.ts x TS auth.service.ts
1 import { ExtractJwt, Strategy } from 'passport-jwt';
2 import { AuthService } from './auth.service';
3 import { PassportStrategy } from '@nestjs/passport';
4 import { Injectable, UnauthorizedException } from '@nestjs/common';
5
6
7 @Injectable()
8 export class JwtStrategy extends PassportStrategy(Strategy) {
9   constructor(private readonly authService: AuthService) {
10     super({
11       jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
12       secretOrKey: 'secretKey',
13     });
14   }
15
16   async validate(payload: any) {
17     const user = await this.authService.validateUser(payload);
18     if (!user) {
19       throw new UnauthorizedException();
20     }
21     return user;
22   }
23 }
```

Obrázek 7-Passport strategy [1]

## 4.7 Hřiště a vše kolem nich

### 4.7.1 Výpis a zobrazení

Ve frontendu byla vytvořena nová stránka fields kde se vypisují data z databáze a ukládají se do předem nastýlované tabulky ReactTable. ReactTable je jednoduchý, rychlý a rozšiřitelný datagrid build pro React, pomocí kterého jsou všechna hřiště vypísána a uspořádána do tabulky.

```
<div>
  <Navbar/>
  <ReactTable
    data={this.state.fields}
    columns={[{ Header: "Hřiště",
    columns: [
      { Header: "ID", accessor: "id" },
      { Header: "Správce", accessor: "owner" },
      { Header: "Název", accessor: "name" },
      { Header: "Cena za hodinu", accessor: "price" },
      { Header: "Lokace", accessor: "location" },
      { Header: "Informace", accessor: "description" },
      {
        id: 'edit',
        accessor: '[row identifier to be passed to button]',
        Cell: ({original}) => (<button class="btn" onClick={() => this.handleDelete(original.id)}><i class="fa fa-close"></i>&#10005;</button>)
      },
    ]
  ]]}
/>
</div>
```

Obrázek 8-Zobrazení [1]

### 4.7.2 Přidávání nového hřiště

Pro přidávání nového hřiště byla vytvořena nová stránka NewField která obsahuje formulář pro přidávání nových hřišť.

```
render() {
  return(
    <React.Fragment>
      <Navbar/>
      <h1>Zadejte nové hřiště</h1>
      <div className="wrapper">
        <div className="formInputsWrapper">
          <form onSubmit={this.handleSubmit}>
            <label htmlFor="name">Jméno</label>
            <input id="name" className="formInput" type="text" placeholder="field name" />
            <label htmlFor="owner">Vlastník</label>
            <input id="owner" className="formInput" name="owner" onChange={this.handleChange} type="text" placeholder="field owner" />
            <label htmlFor="price">Cena</label>
            <input id="price" className="formInput" name="price" onChange={this.handleChange} type="number" placeholder="field price" />
            <label htmlFor="description">Informace</label>
            <textarea id="description" className="formInput" name="description" onChange={this.handleChange} placeholder="field description" />
            <label htmlFor="location">Lokace</label>
            <textarea id="location" className="formInput" name="location" onChange={this.handleChange} type="text" placeholder="field location" />
            <input className="formInput submitButton" type="submit" value="odeslat" />
          </form>
        </div>
      </div>
    </React.Fragment>
  )
}
```

Obrázek 9-Přidání nového hřiště [1]

## 5 Příručka k použití

Tato kapitola obsahuje plný tutoriál k řízení aplikace.

### 5.1 Registrace a přihlášení

Pro přístup k hřištím a dalším prvkům aplikace je nutné se zaregistrovat pomocí Registračního formuláře tím že zadáme přihlašovací jméno, e-mailovou adresu, heslo a stlačíme Sign-Up.

Sign In or Sign Up

FULL NAME  
Vaše\_jméno

PASSWORD  
.....

E-MAIL ADDRESS  
váš\_email@neco.cz

Sign Up I'm already member

Pořídít snímek

Obrázek 10-Registrace [1]

Hned po stisknutí budeme přesměrování na Přihlašovací formulář, kde vyplníme naši e-mailovou adresu a heslo a pak stlačíme Sign-in.

Sign In or Sign Up

E-MAIL ADDRESS  
tomaselsnic@seznam.cz

PASSWORD  
...

Sign In Create an account

Pořídít snímek

Obrázek 11-Přihlášení [1]

## 5.2 Hřiště a ovládání

Po přihlášení se dostaneme na menu, kde máme na výběr zobrazení seznamu hřišť nebo odhlášení. Když zobrazíme hřiště přesměrujeme se na novou stránku, která obsahuje už přidaná hřiště a ovládací prvky.

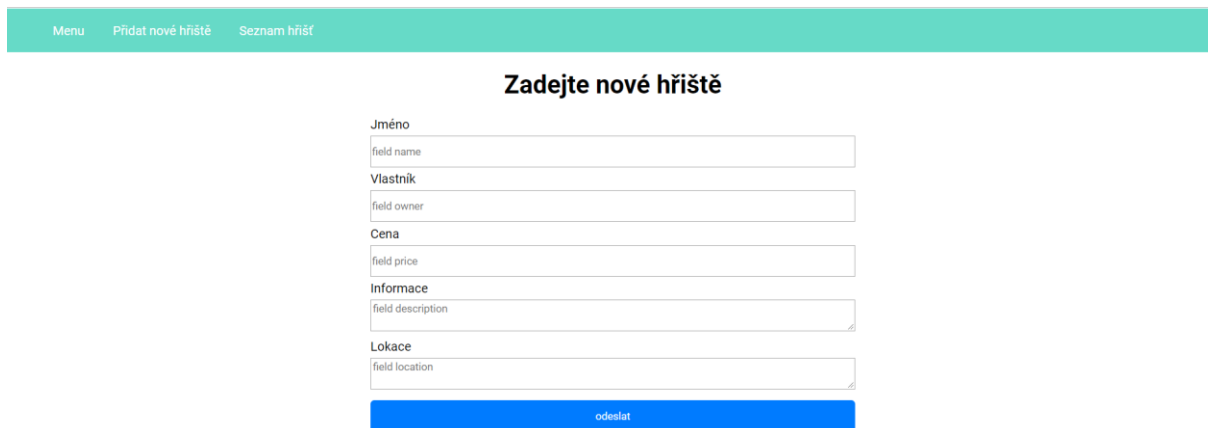
[illegible]

Obrázek 12-Ovládání tabulky hřišť [1]

Pro smazání hřiště lze použít červený křížek a pokud si chce uživatel zobrazit jen určitý počet hřišť může použít scrollmenu.

### 5.2.1 Přidání nového hřiště

Pro přidání nového pole si uživatel v navbaru musí zvolit Přidat nové hřiště a bude přesměrován na novou stránku s formulářem.



Menu Přidat nové hřiště Seznam hřišť

### Zadejte nové hřiště

Jméno  
field name

Vlastník  
field owner

Cena  
field price

Informace  
field description

Lokace  
field location

odeslat

Obrázek 13-Formulář pro přidání nového hřiště [1]

Do formuláře pak už jen uživatel vyplní svoje informace a dá odeslat a tím se pole přidá do seznamu hřišť.

### 5.2.2 Editace hřiště

Kliknutím na hřiště se zobrazí dodatečné informace a také možnost editovat pole. Pole zedituju pomocí tlačítka edit, které po kliknutí zobrazí editovací formulář kde lze změnit všechna data hřiště. Po úspěšné editaci stačí jen zmáčknout odeslat a hřiště se ihned upraví podle dat které uživatel zadal.

## 6 Zhodnocení dosažených výsledků

### 6.1 Splněné cíle

Projekt obsahuje dokončené rozhraní pro uživatele, stránku pro zobrazení, úpravu, přidávání a mazání hřišť. Úspěšně vytvořena i autentizace uživatele pro bezpečnost a jedinečnost uživatele.

### 6.2 Nesplněné cíle

Kvůli dvojměsíční nepřítomnosti se nepodařilo udělat administrátorský režim ani možnost uživatelům přidat kredit pro objednávání hřišť.

## Závěr

Jelikož autor nikdy nepracoval s javascriptovou knihovnou React ani s frameworkem Nestjs musel se k dokončení práce tyto techniky nastudovat. Hlavní bylo naučit se snadno orientovat v těchto knihovnách a naučit se s nimi pracovat jak na frontendu tak i na backendu. Největší problém dělala autorovi počáteční instalace a zorientování se v kódu. Projekt má mnoho možností na rozšíření jako je administrátorský mód nebo přidání kreditů za kterou by se dali hřiště propůjčovat.

## Zdroje

DžejEs - JavaScript pro web. DžejEs - JavaScript pro web [online]. Dostupné z: <https://www.dzejes.cz/react-uvod.html>

An Introduction to NestJS for Ionic Developers | joshmorony - Learn Ionic & Build Mobile Apps with Web Tech. joshmorony - Learn Ionic & Build Mobile Apps with Web Tech [online]. Copyright © Mobirony 2019 [cit. 24.03.2019]. Dostupné z: <https://www.joshmorony.com/an-introduction-to-nestjs-for-ionic-developers/>

Visual Studio Code - Code Editing. Redefined. Visual Studio Code - Code Editing. Redefined [online]. Copyright © 2019 [cit. 24.03.2019]. Dostupné z: <https://code.visualstudio.com/>

Visual Studio Code - ke stažení zdarma - Stahuj.cz. Stahuj.cz - Download shareware a freeware (programy zdarma) [online]. Copyright © 1998 [cit. 24.03.2019]. Dostupné z: [https://www.stahuj.cz/vyvojove\\_nastroje/multifunkcnieditory/visual-studio-code/](https://www.stahuj.cz/vyvojove_nastroje/multifunkcnieditory/visual-studio-code/)

Naučte se stavět moderní weby s React JS a Redux (školení) - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 24.03.2019]. Dostupné z: <https://www.root.cz/zpravicky/naucte-se-stavet-moderni-weby-s-react-js-a-redux-skoleni/>

Documentation | NestJS - A progressive Node.js web framework. Documentation | NestJS - A progressive Node.js web framework [online]. Dostupné z: <https://docs.nestjs.com/>

Getting Started · React Native. [online]. Copyright © 2019 Facebook Inc. [cit. 24.03.2019]. Dostupné z: <https://facebook.github.io/react-native/docs/getting-started>

API Authentication with JSON Web Tokens and Passport — Scotch.io. Top Shelf Web Development Training — Scotch.io [online]. Copyright © Scotch.io, LLC [cit. 24.03.2019]. Dostupné z: <https://scotch.io/@devGson/api-authentication-with-json-web-tokensjwt-and-passport>

Vytvoření aplikace Node.js a React - Visual Studio | Microsoft Docs. [online]. Dostupné z: <https://docs.microsoft.com/cs-cz/visualstudio/javascript/tutorial-nodejs-with-react-and-jsx?view=vs-2017>

MySQL – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/MySQL>

XAMPP – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/XAMPP>

Software – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/Software>

Databáze – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/Datab%C3%A1ze>

Framework – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/Framework>

Framework – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: [https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

Typescript – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: <https://cs.wikipedia.org/wiki/TypeScript>

Vkládání závislostí – Wikipedie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-24]. Dostupné z: [https://cs.wikipedia.org/wiki/Vkl%C3%A1d%C3%A1n%C3%AD\\_z%C3%A1vislost%C3%AD](https://cs.wikipedia.org/wiki/Vkl%C3%A1d%C3%A1n%C3%AD_z%C3%A1vislost%C3%AD)

GitHub - typeorm/typeorm: ORM for TypeScript and JavaScript (ES7, ES6, ES5). Supports MySQL, PostgreSQL, MariaDB, SQLite, MS SQL Server, Oracle, WebSQL databases. Works in NodeJS, Browser, Ionic, Cordova and Electron platforms.. The world's leading software development platform · GitHub [online]. Copyright © 2019 [cit. 25.03.2019]. Dostupné z: [https://github.com/typeorm/typeorm?fbclid=IwAR3uXrNSc9D7dKoKlcDxp9NPQBKfFfxZlB\\_LQqf8pUlhryxzWJEsLykZvo](https://github.com/typeorm/typeorm?fbclid=IwAR3uXrNSc9D7dKoKlcDxp9NPQBKfFfxZlB_LQqf8pUlhryxzWJEsLykZvo)

## Seznam obrázků

|                                                         |    |
|---------------------------------------------------------|----|
| Obrázek 1- Technická část register formuláře [1] .....  | 12 |
| Obrázek 2-Struktura login formuláře [1] .....           | 13 |
| Obrázek 3-Backend [1] .....                             | 14 |
| Obrázek 4-Model [1].....                                | 14 |
| Obrázek 5-Uživatel [1] .....                            | 15 |
| Obrázek 6-User entity [1] .....                         | 15 |
| Obrázek 7-Passport strategy [1].....                    | 16 |
| Obrázek 8-Zobrazení [1] .....                           | 17 |
| Obrázek 9-Přidání nového hřiště [1] .....               | 17 |
| Obrázek 10-Registrace [1] .....                         | 18 |
| Obrázek 11-Přihlášení [1] .....                         | 18 |
| Obrázek 12-Ovládání tabulky hřišť [1].....              | 19 |
| Obrázek 13-Formulář pro přidání nového hřiště [1] ..... | 20 |

## Bibliografie

[1] T. Elsnic, 2019.