

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Informačný systém pre podporu riadenia projektu



2023

Vedoucí práce:  
Mgr. Martin Trnečka, Ph.D.

Michaela Siekelová

Studijní program: Aplikovaná informatika,  
kombinovaná forma

## **Bibliografické údaje**

Autor: Michaela Siekelová  
Název práce: Informačný systém pre podporu riadenia projektu  
Typ práce: bakalárska práca  
Pracovište: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2023  
Studijní program: Aplikovaná informatika, kombinovaná forma  
Vedoucí práce: Mgr. Martin Trnečka, Ph.D.  
Počet stran: 32  
Přílohy: elektronická data v úložišti katedry informatiky  
Jazyk práce: slovenský

## **Bibliographic info**

Author: Michaela Siekelová  
Title: Information system for project management  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2023  
Study program: Applied Computer Science, combined form  
Supervisor: Mgr. Martin Trnečka, Ph.D.  
Page count: 32  
Supplements: electronic data in the storage of department of computer science  
Thesis language: Slovak

## Anotácia

*Cielom tejto bakalárskej práce bolo vytvoriť univerzálny webový informačný systém, realizovaný ako responzívna webová stránka, pre riadenie a správu projektov. Systém umožňuje vytvárať a spravovať projekty, zadávať úlohy a tieto úlohy následne priradovať zamestnancom. Všetky atributy projektu, úlohy a zamestnanca sú konfigurovateľné podľa aktuálnych požiadavkov. Systém umožňuje prehľadné zobrazenie zadaných úloh, vyťaženie zamestnancov a ďalšie kľúčové prvky z oblasti riadenia projektu, a to včetně zobrazenia kritickej cesty.*

## Synopsis

*The goal of this bachelor's thesis was to create a universal information web system, implemented as a responsive website, for project management and administration. The system will allow to create and manage projects, assign tasks and then assign these tasks to employees. All project, task and employee attributes will be configurable according to the current ones requirements. The system will enable a clear display of assigned tasks, workload employees and other key elements from the field of project management, including display critical paths.*

**Kľúčové slová:** řízení projektu; webová aplikace; závěrečná práce;

**Keywords:** project management; web application; bachelor's thesis;

V tejto časti by som rada poďakovala v prvom rade vedúcemu bakalárskej práce páňovi Mgr. Martinu Trnečkovi, Ph.D. za jeho trpezlivosť, pochopenie a mnohé rady a doporučenia, ktoré mi stihol za krátku dobu poskytnúť. Taktiež by som sa rada poďakovala mojim blízkym za podporu a vytvorenie podmienok pre úspešné dokončenie tejto práce.

*Odevzdaním tohto textu jeho autor/ka cestne vyhlasuje, že celú prácu vrátane príloh vypracoval/a samostatne a za použitia iba zdrojov spomínaných v texte práce a uvedených v zozname literatúry.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
1.1	Existujúce riešenia . . . . .	8
1.1.1	Atlassian Jira Software . . . . .	8
1.1.2	Atlassian Trello . . . . .	9
1.1.3	Porovnanie a záver . . . . .	9
<b>2</b>	<b>Technická dokumentácia</b>	<b>10</b>
2.1	Použité technológie . . . . .	10
2.1.1	Spring boot . . . . .	10
2.1.2	Keycloak . . . . .	10
2.1.3	Angular . . . . .	11
2.1.4	PostgreSQL . . . . .	11
2.2	Popis riešenia . . . . .	11
2.2.1	Architektúra implementácie . . . . .	12
2.2.2	REST API . . . . .	13
2.2.3	Diagram databáze . . . . .	14
2.2.4	Adresárová štruktúra zdrojového kódu . . . . .	15
<b>3</b>	<b>Užívateľská príručka</b>	<b>20</b>
3.1	Hlavné menu . . . . .	20
3.2	Projekty . . . . .	20
3.2.1	Prehľad projektov . . . . .	20
3.2.2	Vytvorenie projektu . . . . .	20
3.2.3	Detail projektu . . . . .	21
3.2.4	Atribúty pre tickety v projekte . . . . .	22
3.2.5	Gantt diagram . . . . .	23
3.3	Tickety . . . . .	23
3.3.1	Prehľad ticketov . . . . .	23
3.3.2	Vytvorenie ticketu . . . . .	24
3.3.3	Detail ticketu . . . . .	25
3.3.4	Tasky a progres ticketu . . . . .	25
3.3.5	Hodnoty atribútov ticketu . . . . .	25
3.4	Užívatelia . . . . .	26
3.4.1	Prehľad užívateľov . . . . .	26
3.4.2	Vytvorenie užívateľa . . . . .	26
3.4.3	Zmena hesla . . . . .	27
3.4.4	Detail užívateľa . . . . .	27
<b>4</b>	<b>Možné rozšírenia</b>	<b>28</b>
4.1	Komentáre užívateľov . . . . .	28
4.2	História zmien . . . . .	28
4.3	Prílohy . . . . .	28

<b>Záver</b>	<b>29</b>
<b>Conclusions</b>	<b>30</b>
<b>5 Obsah elektronických dat</b>	<b>31</b>
<b>Bibliografia</b>	<b>32</b>

## Zoznam obrázkov

1	Klienti keycloaku . . . . .	11
2	Diagram architektury aplikácie . . . . .	13
3	Diagram databáze . . . . .	15
4	Adresárová štruktúra REST API . . . . .	18
5	Adresárová štruktúra UI . . . . .	19
6	Hlavné menu . . . . .	20
7	Dialog pre vytvorenie projektu . . . . .	21
8	Detail projektu . . . . .	21
9	Dialog pre pridanie užívateľa do projektu . . . . .	22
10	Vytvorenie atribútu v projekte . . . . .	23
11	Dialog pre vytvorenie ticketu . . . . .	24
12	Detail ticketu . . . . .	25
13	Vytvorenie hodnoty atribútu . . . . .	26
14	Vytvorenie užívateľa . . . . .	26
15	Detail užívateľa . . . . .	27

## Zoznam tabuliek

1	Identifikátory kontrolerov v REST API . . . . .	13
2	Obsah tabuľky s prehľadom projektov . . . . .	22
3	Obsah tabuľky s prehľadom ticketov . . . . .	24

## Zoznam zdrojových kódov

1	Service pre dialog vytvorenia projektu . . . . .	12
2	ProjectPostDto . . . . .	14
3	Kontroler pre projekty . . . . .	14
4	KeyCloakService . . . . .	16
5	AttributeDto . . . . .	16

# 1 Úvod

Projektový manažment nie je doménou len vo svete informačných technológií. Jeho využitie nájdeme v rôznych odvetviach či už priemyslu, marketingu, logistiky a mnohých ďalších. Dnes existuje množstvo aplikácií, ktoré pomáhajú tímom ľudí organizovať svoju prácu a projektovým manžerom vytvárať plán na úspešné dokončenie zadaného projektu. Mnohé z týchto nástrojov sú platené alebo nedostatočné pre prehľadný proces, v ktorom každý člen vie aká je jeho úloha v danej chvíli. V rámci tejto bakalárskej práce som vytvorila aplikáciu, ktorá umožní projekt a jeho priebeh prehľadne naplánovať, následne priradiť dostupné kapacity členov tímu a nakoniec aj previesť členov tímu jednotlivými úlohami projektu.

Text práce som rozdelila na tri sekcie. V úvodnej sekcii sa venuje existujúcim riešeniam, ich výhodám, nevýhodám a využiteľnosti v rôznych odvetviach. V ďalšej sekcii prezentujem aplikáciu vytvorenú v rámci tejto bakalárskej práce a jej použitie. Posledná sekcia je technická dokumentácia a použité technológie.

## 1.1 Existujúce riešenia

Pre účely porovnania existujúcich riešení som vybrala dva rozdielne nástroje od firmy Atlassian ako zástupcov dvoch skupín nástrojov projektového manažmentu. Jednou skupinou sú komplexné riešenia primárne vyvinuté pre tímy softwarových vývojárov. Druhou skupinou sú jednoduchšie nástroje využívajúce takzvané todo listy.

### 1.1.1 Atlassian Jira Software

Jira<sup>1</sup> od firmy Atlassian je jednoznačne jedným z najviac používaných nástrojov pre projektový manažment v oblasti informačných technológií. Ponúka množstvo nástrojov, ktoré nutne užívateľom nevnučuje, a to tomuto nástroju dáva širokú škálu využiteľnosti. Tento software umožňuje zamestnancom jednej organizácie pracovať na viacerých projektoch súčasne. Ponúka základné funkcionality, ako napríklad vytváranie projektov, úloh a ich rozdeľovanie medzi členov tímu, ale aj tie komplexnejšie, napríklad nástienky kanban, gantt diagram, spárovanie softwaru s verzovacím systémom a ďalšie funkcionality využívané nie len v agilných tímoch.

Software Jira je skvelým nástrojom, no je to zároveň komplexný software, ktorý je primárne určený pre tímy softwarových vývojárov. V iných odvetviach sa využíva len zriedka, pretože množstvo funkcionalít, ktoré sú tu nevyužiteľné, je taktiež zahrnuté v cene za túto službu a okrem iného znižujú prehľadnosť a intuitívnosť použitia.

---

<sup>1</sup>Jira (<https://www.atlassian.com/software/jira>)



### 1.1.2 Atlassian Trello

Trello<sup>2</sup>, taktiež od firmy Atlassian, ponúka využitie na rôznych operačných systémoch. Aplikáciu je možné využiť v mobilných zariadeniach ale i cez webový prehliadač. Užívateľ vytvára zoznamy jednotlivých úloh, ktoré je potrebné spraviť a má možnosť tieto zoznamy nazdieľať svojim spolupracovníkom. Úlohy sú zobrazené v kanbanových nástenkách, čo zvyšuje prehľadnosť aktuálneho stavu práce vykonávanej na projekte.

Aplikácia je prehľadná a používanie je intuitívne i pre málo skúsených používateľov. Nevýhodou tohto riešenia ale zostáva absencia nástrojov pre pokročilejšie plánovanie práce a kapacít v rámci projektu.

### 1.1.3 Porovnanie a záver

Obe skupiny existujúcich riešení majú svoje pozitívne i negatívne stránky. Kým kanbanové aplikácie založené na to-do listoch ako je Trello sú prívetivé pre málo skúsených a nenáročných užívateľov, neposkytujú dostatočnú funkcionálnosť pre plánovanie projektu za hranice rozdelenia úloh. Naopak komplexné riešenia podobné Jira software poskytujú nadmernú niekedy až prebytočnú funkcionálnosť, v ktorej sa aj skúsený užívateľ niekedy stráca. Mojim cieľom teda bolo vytvoriť aplikáciu ktorá bude poskytovať jednoduchosť a intuitívne používanie a zároveň bude poskytovať nástroje na pokročilejšie plánovanie projektu a kapacít.

---

<sup>2</sup>Trello (<https://trello.com/>)

## 2 Technická dokumentácia

Táto sekcia sa venuje technickej stránke projektu. V krátkosti predstavuje použité technológie, štruktúru databáze, adresárové štruktúry zdrojového kódu aplikácie a popisuje architektúru implementácie.

### 2.1 Použité technológie

Pre vývoj REST-ového API pomocou frameworku springboot som použila vývojové prostredie IntelliJ Idea<sup>3</sup>. Pre vývoj užívateľského prostredia som využila vývojové prostredie Visual Studio Code<sup>4</sup>.

#### 2.1.1 Spring boot

Spring<sup>5</sup>, ako taký, je framework, ktorý slúži pre jednoduchý vývoj webových aplikácií v Jave. Poskytuje nám veľa funkcií, ktoré nám uľahčia život a niekoľko tisíc riadkov kódu. Spring Boot<sup>6</sup> je nadstavba Spring. Implementuje teda všetky funkcie Spring, ale ešte k tomu nám poskytuje jednu veľkú funkciu navyše, ktorá nám tvorbu projektu veľmi zrýchli a uľahčí. Spring Boot má totiž v sebe server Tomcat<sup>7</sup>, ktorý nám pri spustení projektu vytvorí server, ktorý nastaví prístupný pre prehliadače.

#### 2.1.2 Keycloak

Keycloak<sup>8</sup> je open sourcový software, ktorý umožňuje jednotné prihlásenie (IdP) pomocou správy identít a správy prístupu pre aplikácie. Tento software je implementovaný v jazykoch Java a vo východnom nastavení podporuje protokoly federácie identity SAML v2 a OpenID Connect (OIDC) / OAuth2.

Prostredníctvom administrátorskej konzoly môžu správcovia centrálnie spravovať všetky aspekty servera Keycloak. Môžu povoliť a zakázať rôzne funkcie. Môžu nakonfigurovať sprostredkovanie identity a federáciu používateľov. Môžu vytvárať a spravovať aplikácie a služby a definovať podrobnú autorizačnú politiku. Môžu tiež spravovať používateľov vrátane povolení a relácií.

Keycloak stojí nad oboma koncami REST-ovej komunikácie. Overuje, že požiadavky, ktoré prišli do REST-ového API pochádzajú od klienta užívateľského prostredia teda angularovej časti aplikácie. Zároveň autorizuje užívateľa na základe údajov poslaných v hlavičke requestu. REST API, ktoré tvorí springová aplikácia, pomocou keycloaku zisťuje aký užívateľ je prihlásený a môže s jeho pomocou vytvárať nových užívateľov. Obe tieto spojenia majú v administrácii keycloaku svojho klienta (obr. 1).

---

<sup>3</sup>IntelliJ Idea (<https://www.jetbrains.com/idea/>)

<sup>4</sup>Visual Studio Code (<https://code.visualstudio.com/>)

<sup>5</sup>Spring (<https://spring.io/>)

<sup>6</sup>Spring Boot (<https://spring.io/projects/spring-boot>)

<sup>7</sup>Tomcat (<https://tomcat.apache.org/>)

<sup>8</sup>Keycloak (<https://www.keycloak.org/>)

Client ID	Enabled	Base URL
account	True	<a href="http://localhost:8180/auth/realms/pato/account/">http://localhost:8180/auth/realms/pato/account/</a>
account-console	True	<a href="http://localhost:8180/auth/realms/pato/account/">http://localhost:8180/auth/realms/pato/account/</a>
admin-cli	True	Not defined
angular-springboot	True	Not defined
broker	True	Not defined
realm-management	True	Not defined
security-admin-console	True	<a href="http://localhost:8180/auth/admin/pato/console/">http://localhost:8180/auth/admin/pato/console/</a>
springboot-keycloak	True	Not defined

Obr. 1: Klienti keycloaku

### 2.1.3 Angular

Angular<sup>9</sup> je vývojová platforma, ktorá je postavená na TypeScript<sup>10</sup>. TypeScript je silne typovaný programovací jazyk, ktorý stavia na JavaScripte. Platforma Angular zahŕňa:

- Rámec založený na komponentoch na vytváranie škálovateľných webových aplikácií.
- Zbierku dobre integrovaných knižníc, ktoré pokrývajú širokú škálu funkcií vrátane smerovania, správy formulárov, komunikácie klient-server a ďalších.
- Sadu vývojárskych nástrojov, ktoré pomáhajú vyvinúť, zostaviť, testovať a aktualizovať kód.

K vývoju aplikácie som použila angularové knižnice Angular Material<sup>11</sup> a Syncfusion<sup>12</sup>.

### 2.1.4 PostgreSQL

PostgreSQL<sup>13</sup> je objektovo-relačný databázový systém. PostgreSQL podporuje SQL (relačné) aj JSON (nerelačné) dotazovanie.

## 2.2 Popis riešenia

Táto sekcia popisuje samotnú implementáciu aplikácie. V prvej podsekcii vysvetľuje architektúru aplikácie. Druhá sekcia dokumentuje REST-ové API. Ďalej zobrazuje a opisuje databázovú štruktúru a nakoniec približuje adresárovú štruktúru pre priblíženie stavby implementácie.

<sup>9</sup>Angular (<https://angular.io/>)

<sup>10</sup>TypeScripte (<https://www.typescriptlang.org/>)

<sup>11</sup>Angular Material (<https://material.angular.io/>)

<sup>12</sup>Syncfusion (<https://www.syncfusion.com/angular-components>)

<sup>13</sup>PostgreSQL (<https://www.postgresql.org/>)

### 2.2.1 Architektúra implementácie

Na obr.2 je vyobrazená architektúra aplikácie. Aplikácia sa skladá zo štyroch samostatných komponent. Užívateľ pracuje len s aplikáciou užívateľského prostredia, ktorá na užívateľskú interakciu reaguje vyvolaním udalosti v angularovej komponente. Každá takáto komponenta má svoj service, ktorý zabezpečuje odosielanie HTTP žiadostí na adresu REST-ového API. Príklad takého servisu pre komponentu je zobrazený v zdrojovom kóde 1. Pred odoslaním žiadosti si service vytiahne hlavičku pre žiadosť od service služby pokrývajúcej komunikáciu s aplikáciou Keycloak. Keycloak je spustený ako samostatná aplikácia využívajúca aplikačný server Wildfly<sup>14</sup>.

Zdrojový kód 1: Service pre dialog vytvorenia projektu

```
@Injectable({
    providedIn: 'root'
})
export class CreateProjectDialogRestService {

    private baseUrl: string = environment.API_URL;

    constructor( private _http: HttpClient, private kc: KeycloakService ) { }

    createProject(project: ProjectDto) {
        let headers = new HttpHeaders().set('Content-Type', 'application/json'
        );
        this.kc.addTokenToHeader(headers)
        return this._http.post(this.baseUrl + 'projectManagement/project/
        create', project, {headers:headers});
    }
}
```

Žiadosť prijme REST API podľa adresy URL danej žiadosti v kontroleri spravujúcim danú oblasť. REST API aplikácie bližšie popisuje sekcia 2.2.2. Tieto kontrolery sú v aplikácii štyri. Kontrolery spravujú tickety, projekty, atribúty a užívateľov. Kontroler slúži k prijatiu žiadosti, verifikácii hlavičky na Keycloak, zavolanie potrebnej metody servisu a ak je to potrebné k odoslaniu odpovede. V REST API spravujú triedy s názvom Service logiku aplikácie. Ak je potrebné spraviť úpravy v databázi alebo sa databáze dotázať na dáta, využije service služieb triedy Repository. Triedy s názvom Repository rozširujú triedu JpaRepository frameworku Spring a zabezpečujú komunikáciu s databázou. Každá entita má svoju Repository triedu a tabuľku v databáze.

---

<sup>14</sup>Wildfly (<https://www.wildfly.org/>)

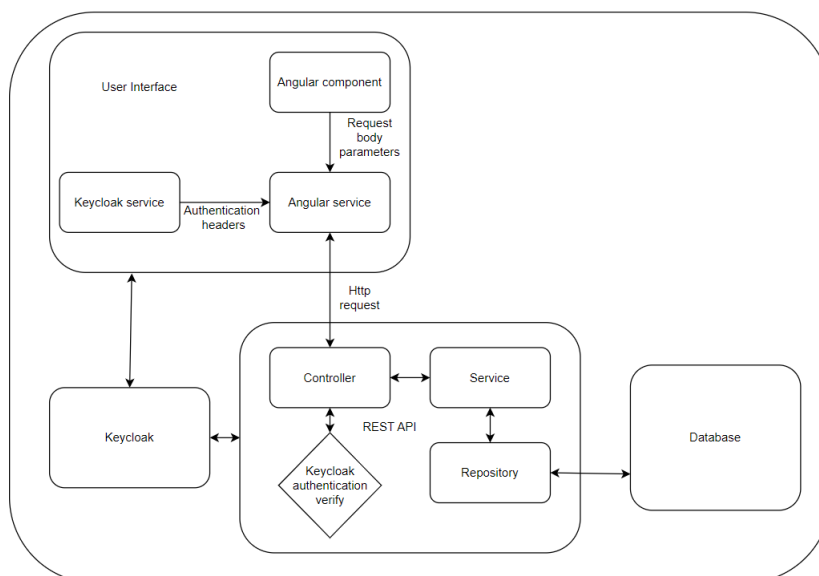
## 2.2.2 REST API

Koncové body REST-ového API identifikujú metódy vykonávajúce požadované úkony. Pre správne vytvorenie žiadosti na daný koncový bod, je potrebné do hlavičky žiadosti vložiť potrebné údaje pre autentifikáciu užívateľa a do tela žiadosti vložiť vstupné parametre vo formáte JSON. Koncové body sú identifikované pomocou URL adresy, ktorá sa skladá z adresy kde je spustený aplikačný server, kľúčového slova `api`, identifikátoru kontrolera, ktorý prijíma žiadosť a identifikátoru metódy, ktorá sa má vykonať. V tabuľke 1 sú zobrazené identifikátory kontrolerov obsluhujúcich jednotlivé oblasti aplikácie.

Tabuľka 1: Identifikátory kontrolerov v REST API

Projekty	projectManagement
Tickety	ticketManagement
Používatelia	userManagement
Atribúty	attributeManagement

Identifikátory metód a telo žiadosti sa líšia vzhľadom k povahe potrebného úkonu. K vytvoreniu danej entity je identifikátorom metódy názov entity a kľúčové slovo `create`. Telo žiadosti pre vytvorenie entít je špecifické pre každú entitu. Ako príklad uvádzam *ProjectPostDto*, zobrazený v zdrojovom kóde 2. Toto DTO je vstupným parametrom pre metódu určenú na vytvorenie projektu (zdrojový kód 3).



Obr. 2: Diagram architektury aplikácie

### Zdrojový kód 2: ProjectPostDto

```
public class ProjectPostDto {  
  
    @JsonProperty("name")  
    private String name;  
  
    @JsonProperty("shortcut")  
    private String shortcut;  
  
    @JsonProperty("deadline")  
    private Date deadline;  
  
}
```

### Zdrojový kód 3: Kontroler pre projekty

```
@RestController  
@RequestMapping("api/projectManagement")  
public class ProjectController {  
  
    @Autowired  
    private ProjectService projectService;  
  
    @CrossOrigin(origins = "*")  
    @PostMapping(path = "project/create")  
    public void createProject(@RequestBody ProjectPostDto dto){  
        // ...  
    }  
  
    // ...  
}
```

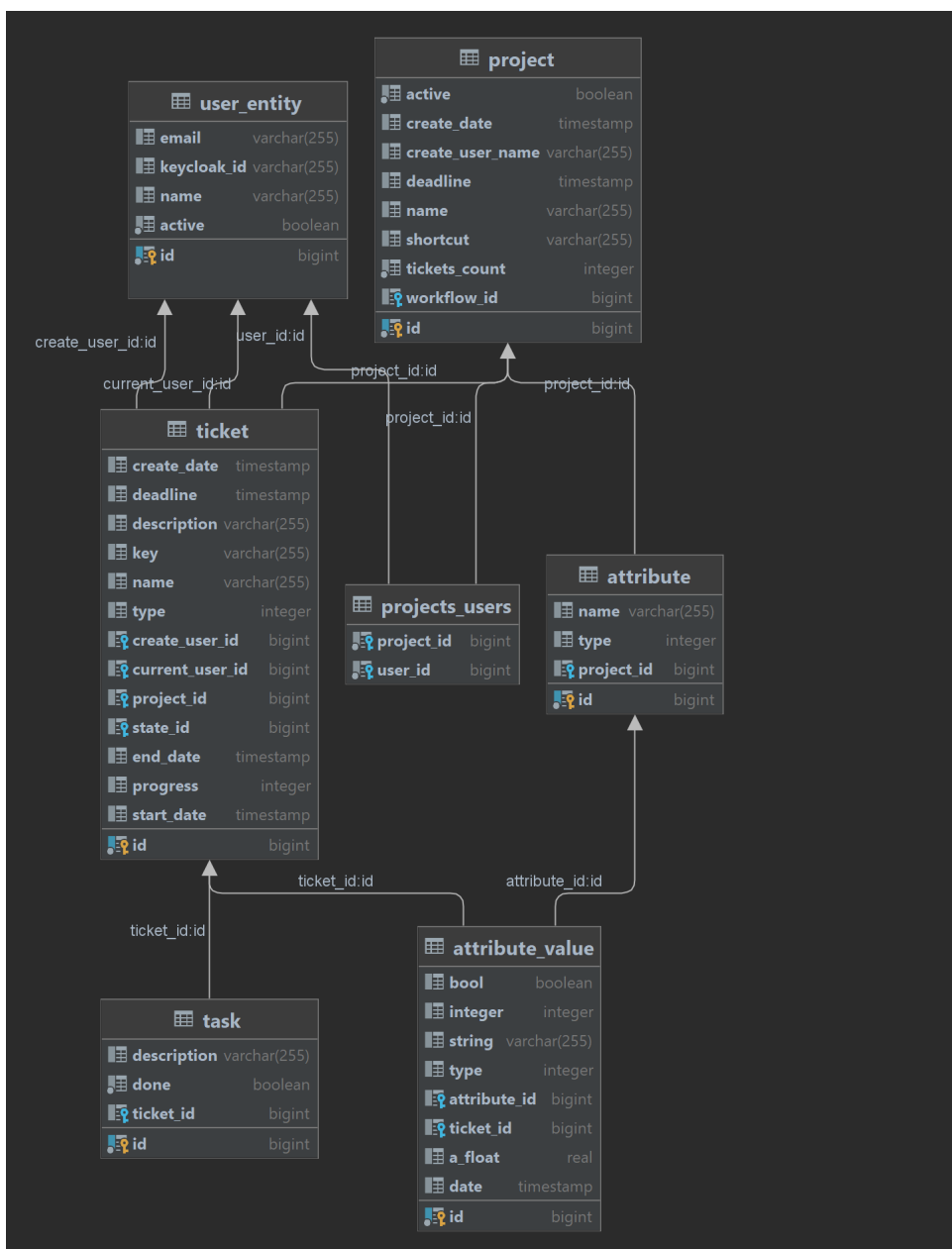
### 2.2.3 Diagram databáze

Diagram databáze (obr. 3) je vygenerovaný pomocou vývojového prostredia IntelliJ Idea. Hlavnými triedami na ktorých je aplikácia založená, sú triedy užívateľa (*user\_entity*), projektu (*project*) a ticketu (*ticket*). Zvyšné triedy sú pomocné pre rozšírenie informácií o hlavných triedach. Trieda *project\_users* uchováva informáciu o tom, ktorý užívateľia pracujú na akom projekte. Trieda *attribute* je viazaná iba na projekt a definuje iba povahu atribútov teda typ a názov. Každý projekt môže mať neobmedzené množstvo atribútov. Trieda *attribute\_value* predstavuje už konkrétnu hodnotu atribútu priradenú k ticketu a jej povaha je definovaná triedou *attribute*. Pre detailnejšie rozdelenie úloh v rámci ticketu je využitá trieda *task*. Každý ticket môže mať neobmedzené množstvo taskov.

## 2.2.4 Adresárová štruktúra zdrojového kódu

Aplikácia sa skladá z dvoch samostatne bežiacich aplikácií, ktoré spolu komunikujú pomocou HTTP dotazov.

REST-ové API implementované v frameworku springboot prijíma dotazy z aplikácie užívateľského prostredia, obstaráva logiku aplikácie, komunikáciu s keycloak a mazať databáze. V zložke `src/main/java/cz/upol/pato` sa nachádzajú zložky manažmentu jednotlivých častí aplikácie. Každá entita



Obr. 3: Diagram databáze

má svoj manažment, ktorý obsahuje zložky `ctrl`, `service`, `repository` a `entity`. V zložke `ctrl` sa vždy nachádza kontroler pre prijímanie dotazov a odosielanie odpovedí. Tento kontroler volá pre vykonanie dotazu triedu `servisu`, ktorá sa nachádza v zložke `service`. Servisa má k dispozícii `repository` triedy, ktoré pracujú s databázou. Každá entita má svoje vlastné `repository` rozhranie. V zložke `keycloak` sa nachádzajú konfiguračné súbory a `service` pre prácu s Keycloak-om. V `KeycloakService.java` je implementovaná komunikácia so službou Keycloak, ktorá je potrebná k získaniu informácií o aktuálne prihlásenom užívateľovi. Táto metóda (zdrojový kód 4) je použitá všade, kde je potrebné dáta v aplikácii vyselktovať priamo pre prihláseného užívateľa.

#### Zdrojový kód 4: KeyCloakService

```
@Service
public class KeyCloakService {

    @Autowired
    private KeycloakAdminClientConfig keycloakAdminClientConfig;

    public String getCurrentUserId(){
        Authentication auth = SecurityContextHolder.getContext()
            .getAuthentication();
        log.info(auth.getName());
        log.info(SecurityContextHolder.getContext().getAuthentication().
            getPrincipal().toString());
        @SuppressWarnings("unchecked")
        KeycloakPrincipal<RefreshableKeycloakSecurityContext> principal =
            (KeycloakPrincipal<RefreshableKeycloakSecurityContext>)
            SecurityContextHolder.getContext()
                .getAuthentication().getPrincipal();
        return principal.getName();
    }

    // ...
}
```

V zložke `mapstruct` sú uložené DTO štruktúry potrebné pre mapovanie z a do dát vo formáte JSON. Atribúty tried sú mapované podľa hodnoty vyplnenej v použitej anotácii `@JsonProperty`. Ako príklad je v zdrojovom kóde 5 zobrazená trieda `AttributeDto`.

#### Zdrojový kód 5: AttributeDto

```
public class AttributeDto {
    @JsonProperty("type")
    private String type;

    @JsonProperty("projectId")
```



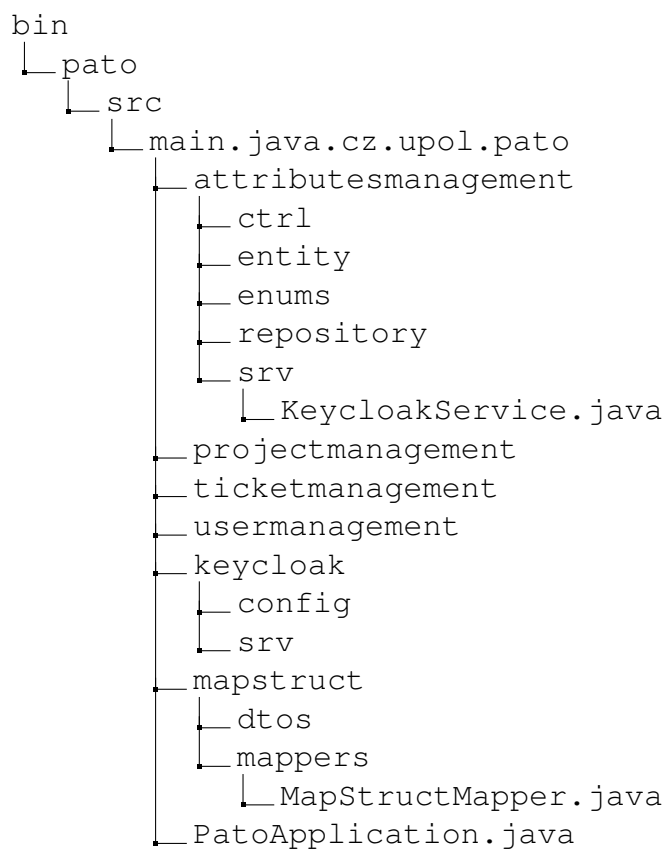
```
private long projectId;

@JsonProperty("name")
private String name;

@JsonProperty("id")
private String id;
}
```

Užívateľské prostredie implementované ako angularová aplikácia sa skladá z jednotlivých komponent, ktoré sú poskladané tak, aby vytvárali jednostránkovú aplikáciu. To spôsobuje, že je možné vždy načítať len potrebnú časť prostredia a nie celú stránku. Koreňová zložka `src` obsahuje koreňový html súbor `index.html`, ktorý obsahuje potrebnú html hlavičku, a v tele tohto súboru, je vložená komponenta `app/app.component.html`. Tento súbor obsahuje komponentu hlavného menu `app/main-menu`. Hlavné menu využíva `app/AppRoutingModule`, v ktorom je nadefinované navigovanie v aplikácii. Zložka `app` obsahuje komponenty roztriedené na dialogy, prehľady, tabuľky a grafy. Každá komponenta sa skladá z html súboru, css súboru a suborov v jazku TypeScript, ktoré zabezpečujú logiku komponenty. Súbor s koncovkou `.service.ts` zabezpečujú komunikáciu komponenty s REST-ovým API.

Obr. 4: Adresárová štruktúra REST API



Obr. 5: Adresárová štruktúra UI



## 3 Uživatelská příručka

Táto sekcia slúži ako uživatelská príručka k aplikácii vyvinutej v rámci tejto bakalárskej práce. Úvodná obrazovka po otvorení aplikácie je určená na prihlásenie užívateľa, pretože neregistrovaným užívateľom aplikácia nie je sprístupnená.

### 3.1 Hlavné menu

Po prihlásení je užívateľ presmerovaný na úvodnú stránku, a v hornej časti obrazovky sa nachádza lišta s hlavným menu. Menu má v pravej časti ikonu pre zobrazenie vlastného profilu užívateľa, ikona zámky pre zmenu hesla a tlačítko pre odhlásenie. Po kliknutí na tlačítko odhlásenia je užívateľ presmerovaný znovu na obrazovku prihlásenia. Zmena hesla je realizovaná cez Obrazovku Keycloak-u. V ľavej časti lišty sa nachádzajú tri tlačidlá *Projects*, *Tickets* a *Users*. Každé z týchto tlačidiel otvorí užívateľovi prehľad daného obsahu.



Obr. 6: Hlavné menu

### 3.2 Projekty

Jedným zo základných prvkov pri plánovaní projektu je samotný projekt. Je to štruktúra, ktorá v sebe nesie všetky potrebné informácie pre konfiguráciu a plánovanie úloh potrebných k úspešnému splneniu projektových požiadavok.

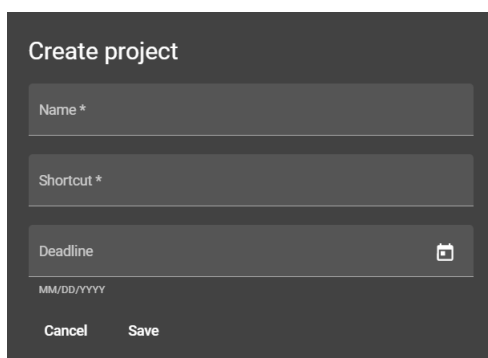
#### 3.2.1 Prehľad projektov

Prehľad projektov tvorí tabuľka, ktorej jednotlivé riadky predstavujú projekty, v ktorých je užívateľ účastníkom. Stĺpce tabuľky zobrazujú názov projektu, skratku projektu a dátum do kedy pránuje byť projekt dokončený. Tieto údaje sú bližšie vysvetlené v tabuľke 2. Prvým stĺpcom tabuľky sú akčné tlačítka. V hlavičke tabuľky je tlačítko so symbolom +, pre otvorenie dialógového okna na vytvorenie užívateľa. Tento dialog je bližšie popísaný v sekcii 3.2.2. V tomto stĺpci je v riadkoch ikona dvojitej šípky, ktorá užívateľa presmeruje na stránku detailu projektu a ikona smetného koša pre odstránenie projektu. Detail projektu je bližšie popísaný v sekcii 3.2.3.

#### 3.2.2 Vytvorenie projektu

Dialogové okno pre vytvorenie nového projektu je vyobrazené na obr.7. Dialog obsahuje povinné polia pre zadanie názvu projektu, skratky projektu a nepovinné pole pre zadanie dátumu plánovaného dokončenia projektu. Názov projektu môže mať maximálne 50 znakov a skratka projektu môže mať maximálne 10 znakov.

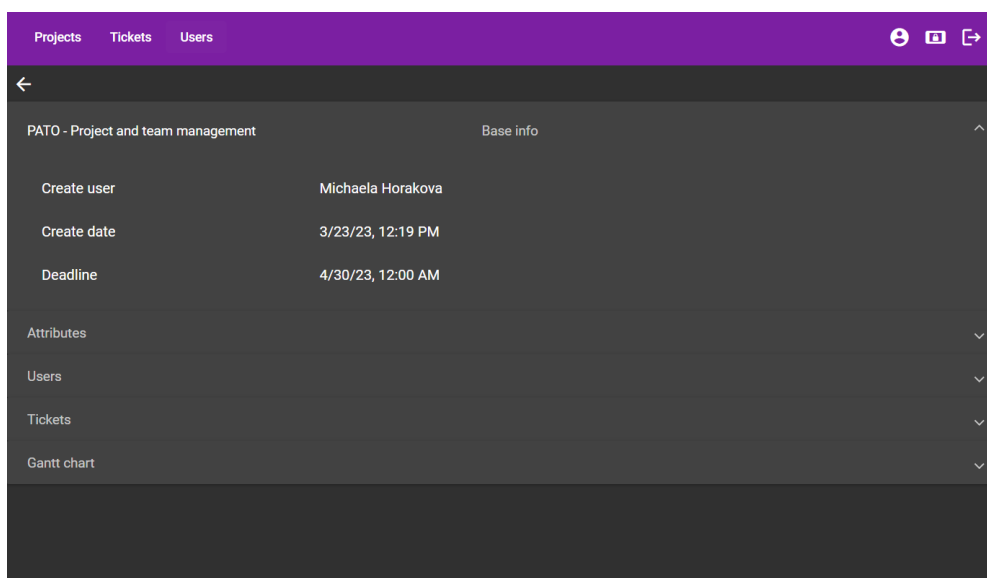
Nie je možné vytvoriť projekt s dátumom plánovaného dokončenia projektu v minulosti. Po kliknutí na tlačidlo *Save* v dolnej časti okna, sa projekt vytvorí a užívateľa, ktorý tento projekt vytvoril pridá do zoznamu účastníkov projektu. Toto okno sa zatvorí a užívateľovi sa znovu zobrazí prehľad projektov, v ktorom sa teraz zobrazuje aj novo vytvorený projekt. V okne sa nachádza aj tlačidlo *Cancel*, ktoré operáciu zruší a okno zavrie.



Obr. 7: Dialog pre vytvorenie projektu

### 3.2.3 Detail projektu

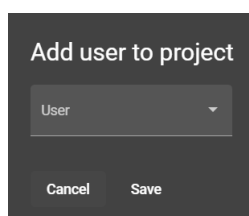
V ľavom hornom rohu detailu projektu je tlačidlo šípky, ktorá vráti užívateľa späť na prehľad projektov. Detail projektu zobrazuje 5 osobitných častí. Viz obr. 8.



Obr. 8: Detail projektu

Prvou časťou sú základné údaje projektu. V hlavičke tejto časti je skratka

projektu a jeho názov. Po rozbalení môže užívateľ vidieť údaje o tom kto projekt vytvoril, kedy ho vytvoril a dátum plánovaného ukončenia. V druhej časti detailu sa nachádza rozbalovací panel pre atribúty, ktorým sa venuje sekcia 3.2.4. V rozbalovacom paneli s titulom *Users* sa nachádza prehľad užívateľov, ktorí sú účastníkmi projektu. V ľavom hornom rohu tabuľky užívateľov je symbol +, ktorá otvára dialog pridania užívateľa do projektu, viz obr. 9.



Obr. 9: Dialog pre pridanie užívateľa do projektu

V dialogovom okne (obr. 9) je k dispozícii výber užívateľov, ktorí sú už v aplikácii aktívni, teda vytvorení, a zároveň v čase otvorenia okna nie sú účastníkmi projektu. Po uložení okna sa okno zavrie a užívateľ je stále v detaile projektu, a prehľad účastníkov projektu je rozšírený o vybraného užívateľa. Rozbalovací panel s titulom *Tickets* obsahuje tabuľku tiketov, ktoré spadajú pod daný projekt. Tejto tabuľke sa bližšie venuje sekcia 3.3.1. V poslednej časti detailu sa nachádza rozbalovací panel s titulom Gantt diagram. Tejto časti sa bližšie venuje sekcia 3.2.5.

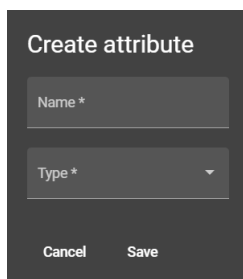
Tabuľka 2: Obsah tabuľky s prehľadom projektov

Name	Celý názov projektu
Shortcut	Skratka projektu využívaná pri tvorbe kľúču ticketu
Deadline	Plánovaný dátum dokončenia projektu
Create user	Užívateľ ktorý projekt vytvoril
Create date	Dátum vytvorenia projektu

### 3.2.4 Atribúty pre tickety v projekte

Projekty rôznych zameraní, môžu mať potrebu konfigurácie údajov, ktoré je užitočné zobrazit' pri jednotlivých úlohách, i keď nie sú bežne používané. Pre uvedenie príkladu, to môže byť napríklad dátum prezentácie splnenej úlohy zákazníkovi. Tento údaj aplikácia neumožňuje nastaviť vo východných vlastnostiach ticketu, ale má na tento účel takzvané atribúty. Užívateľ môže v tejto časti vytvoriť pre projekt atribút, ktorý sa neskôr dá nastaviť v každom z ticketov projektu. Hodnotám atribútov pre konkrétne tickety sa venuje sekcia 3.3.5. Po otvorení rozbalovacieho panelu s titulom *Attributes*, ktorý sa nachádza v detaile projektu,

môže užívateľ vidieť tabuľku všetkých atribútov vytvorených pre daný projekt. V ľavom hornom rohu tabuľky je tlačítko so symbolom +, ktoré po kliknutí otvára dialog pre vytvorenie atribútu. Prvým políčkom pre vyplnenie je názov atribútu, a druhým políčkom je výber typu atribútu. Atribút môže byť typu text, číslo alebo dátum. Po uložení dialogového okna bol do tabuľky atribútov pridaný nový atribút. Jednotlivé atribúty projektu je možné zmazať ikonou v ľavej časti každého záznamu. Vymazanie atribútu zmaže i všetky hodnoty tohto atribútu priradené k ticketom.



Obr. 10: Vytvorenie atribútu v projekte

### 3.2.5 Gantt diagram

Gantt diagram je nástroj, ktorý je pri plánovaní projektu nápomocný pre prehľad o úlohách rozvrhnutých v čase a vyťažených kapacitách. Rozbalovací panel s titulom *Gantt diagram* obsahuje grafické znázornenie ticketov projektu v čase. V ľavej strane môže užívateľ vidieť o aký ticket ide, a v pravej časti je vyobrazený tento ticket na časovej osi. Z grafu je možné vyčítať na koľko percent je ticket hotový, kto na ňom pracuje a kedy je naplánované splnenie danej úlohy. Percentuálny progres ticketu je vypočítaný podľa celkového počtu taskov ticketu a počtu splnených taskov ticketu.

## 3.3 Tickety

Každý projekt má svoje úlohy, ktoré je potrebné splniť pre jeho dokončenie. K tomuto účelu v aplikácii slúžia tickety.

### 3.3.1 Prehľad ticketov

V prehľade ticketov sa nachádzajú všetky tickety ktoré sú priradené k riešeniu prihlásenému užívateľovi. Stĺpce tabuľky zobrazujú kľúč ticketu, názov ticketu, typ, mená užívateľa, ktorý ticket vytvoril a aktuálneho riešiteľa, dátumy vytvorenia a predpokladaného splnenia a nakoniec aj skratku projektu, pod ktorý ticket spadá. Tieto údaje sú bližšie popísané v tabuľke 3.

Tabuľka 3: Obsah tabuľky s prehľadom ticketov

Key	Kľúč ticketu ktorý sa skladá zo skratky projektu a poradového čísla ticketu
Name	Celý názov ticketu
Type	Typ - teda či ide o úlohu alebo opavu chyby
Create user	Užívateľ ktorý ticket vytvoril
Assignee	Užívateľ ktorý je aktuálne priradený na prácu na tickete
Create date	Dátum vytvorenia ticketu
Deadline	Plánovaný dátum dokončenia ticketu
Project	Skratka projektu pod ktorý ticket spadá
Start date	Odhadovaný dátum začatia prác na tickete
End date	Odhadovaný dátum dokončenia ticketu
Progress	Percentuálne vyjadrenie koľko podúloh ticketu je hotových

### 3.3.2 Vytvorenie ticketu

V ľavom hornom rohu tabuľky ticketov sa nachádza tlačítko s ikonou +, ktoré otvára dialógové okno (obr. 11) pre vytvorenie ticketu. V tomto okne sú povinné polia ako názov ticketu, projekt, typ, predpokladaný dátum dokončenia úlohy a popis. Typ ticketu môže byť task (úloha) alebo bug (chyba). Po uložení dialógového okna, sa okno zavrie a v prehľade pribudne jeho záznam.

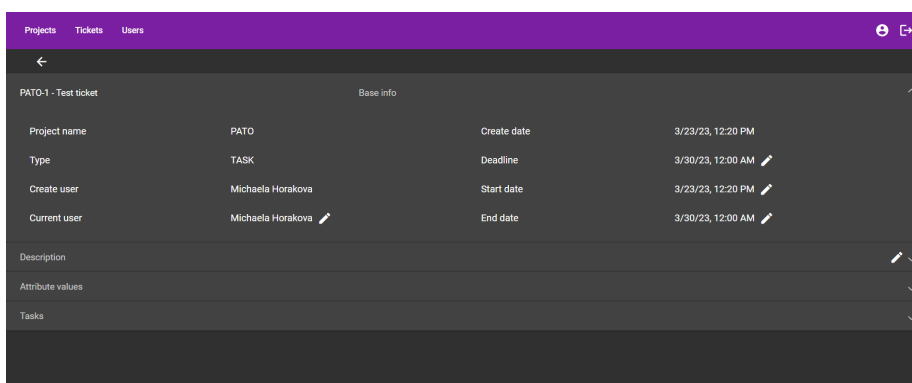
Obr. 11: Dialog pre vytvorenie ticketu



### 3.3.3 Detail ticketu

Detail ticketu (obr. 12) pozostáva zo štyroch častí, z ktorých každá sa nachádza v rozbaľovacom paneli.

Prvá časť je panel so základnými údajmi o tickete, ako napríklad projekt, typ, priradený užívateľ atď. Všetky parametre ticketu sú uvedené v tabuľke 3. V tejto časti je možné priradiť ticket inému užívateľovi pomocou ikony ceruzky vedľa mena v kolonke *Current user*. Rozšírenie informácií o tickete je možné v časti *Attribute values*. Táto časť je bližšie popísaná v sekcii 3.3.5. Ďalšou časťou je popis ticketu, ktorý je možné upraviť pomocou ceruzky v pravom hornom rohu panelu. V časti s titulom *Tasks* je zoznam podúloh, ktoré je potrebné splniť pre dokončenie ticketu. Tasky sú bližšie popísané v sekcii 3.3.4.



Obr. 12: Detail ticketu

### 3.3.4 Tasky a progres ticketu

Tasky predstavujú jednotlivé úkony ktoré je potrebné vykonať pre splnenie ticketu. Podľa týchto podúloh je vypočítaný percentuálny progres pre účely zobrazenia aktuálneho stavu v ganttovom diagrame projektu. Podúlohy je možné vytvárať pomocou tlačítka s ikonou + a mazať pomocou ikony smetného koša. Zaškrtnutím check-boxu na ľavej strane označí užívateľ splnenie úlohy.

### 3.3.5 Hodnoty atribútov ticketu

V sekcii 3.2.4 je popísané ako je možné v aplikácii vytvoriť atribúty. V tejto sekcii sa budem venovať tomu, ako tieto atribúty využiť, na vytvorenie hodnôt atribútu. V prehľade *Attribute values* nájde užívateľ tlačítka s ikonou +, ktoré otvorí dialógové okno (obr. 13) pre vytvorenie hodnoty atribútu. Toto dialógové okno má niekoľko vstupných polí no nie všetky sa dajú vyplniť. Ako prvé je nutné zvoliť atribút ktorého hodnotu chceme pridať. Podľa typu tohto atribútu sa odblokuje dané pole pre vyplnenie. Po uložení sa okno zavrie, a v prehľade pribudne vytvorený záznam.

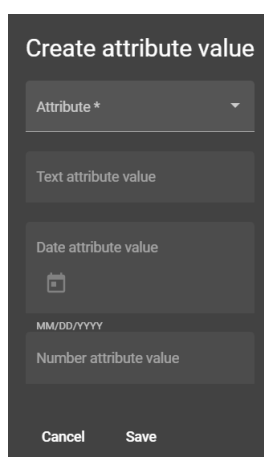
## 3.4 Uživatelia

### 3.4.1 Prehľad užívateľov

V prehľade užívateľov sa nachádzajú všetci aktívni užívatelia. V ľavom stĺpci je možnosť vytvorenia nového užívateľa, zobrazenia jeho detailu, a jeho odstránenie.

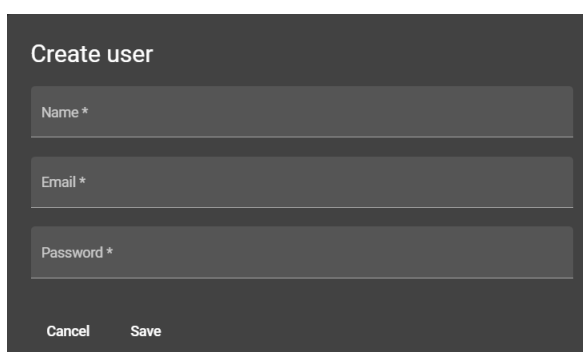
### 3.4.2 Vytvorenie užívateľa

Vytvoriť užívateľa môže iba užívateľ ktorý už účet založený má. V hlavnom prehľade užívateľov sa nachádza tlačítko s ikonou +, ktoré otvorí dialógové okno pre vytvorenie užívateľa. V tomto okne je potrebné vyplniť meno, email a heslo nového užívateľa. Ihned po uložení je užívateľ vytvorený a môže sa prihlásiť.



The image shows a dark-themed dialog box titled "Create attribute value". It contains a dropdown menu labeled "Attribute \*", followed by three input fields: "Text attribute value", "Date attribute value" (with a calendar icon and "MM/DD/YYYY" format), and "Number attribute value". At the bottom, there are "Cancel" and "Save" buttons.

Obr. 13: Vytvorenie hodnoty atribútu



The image shows a dark-themed dialog box titled "Create user". It contains three input fields: "Name \*", "Email \*", and "Password \*". At the bottom, there are "Cancel" and "Save" buttons.

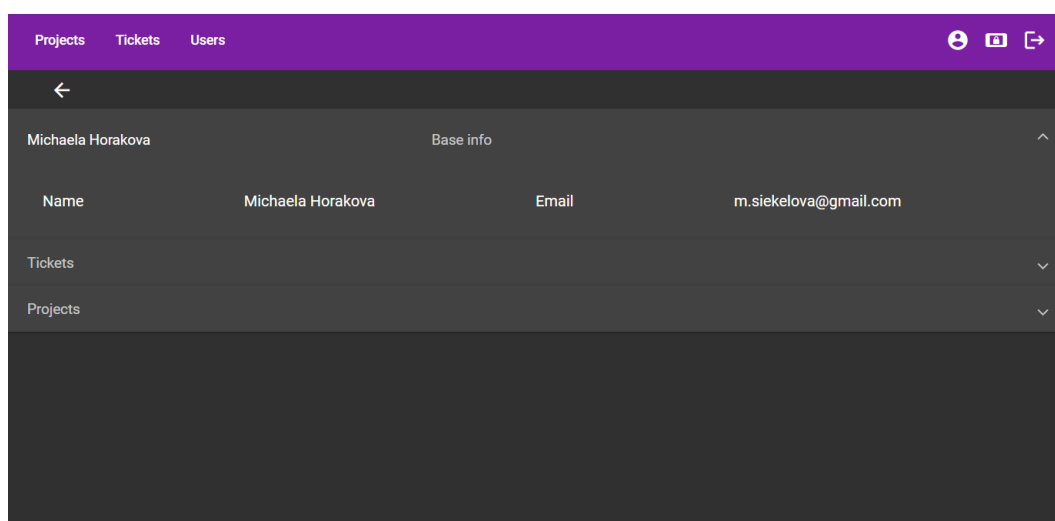
Obr. 14: Vytvorenie užívateľa

### 3.4.3 Zmena hesla

Keďže východzie heslo užívateľa je vytvárané kolegom, ktorý mu zakladá účet, je nutné po prvom prihlásení heslo zmeniť. Užívateľ je vyzvaný na vytvorenie nového hesla ihneď po prvom prihlásení. Neskôr si môže užívateľ zmeniť heslo pomocou ikony zámky v pravom hornom rohu hlavného menu.

### 3.4.4 Detail užívateľa

Detail užívateľa (obr. 15) je poskladaný rovnakým spôsobom ako detaily projektu a ticketu. Aplikácia nezobrazuje veľa osobných údajov užívateľov. Nie je to pre projekt potrebné. Preto v prvej časti základných informácií je iba email a meno užívateľa. Ďalej tento detail obsahuje prehľad ticketov ktorých je daný užívateľ riešiteľom a prehľad projektov ktorých je daný užívateľ účastníkom.



Obr. 15: Detail užívateľa

## 4 Možné rozšírenia

Aplikáciu by v budúcnosti bolo možné rozšíriť napríklad o komentáre užívateľov, históriu zmien na projekte a úlohách alebo možnosťou pridať k ticketom alebo projektom súbory rôzneho typu.

### 4.1 Komentáre užívateľov

Pri implementácii komentárov by vznikol nový panel v užívateľskom prostredí v detaile ticketu. Panel by obsahoval menšie sekcie s výpisom mena užívateľa, ktorý ticket komentoval, dátumom komentára a jeho samotným obsahom. V hornej časti novo vzniknutého panelu by bolo pole pre vloženie textu a tlačítko pre uloženie. V REST API by vznikol nový kontroler a service pre komentáre. Komentáre by bolo možné vytvárať, mazať a upravovať.

### 4.2 História zmien

V histórii zmien by bolo možné evidovať napríklad zmeny pridelenia ticketu, plánovaných dátumov plnenia a zmeny v taskoch, ktoré ticket obsahuje. Bolo by tak možné sledovať či sa pôvodné predstavy o plnení úloh zhodujú s prvotnou predstavou.

Pri implementácii by vznikla nová tabuľka v detaile ticketu ktorá by zobrazovala dátum zmeny, typ zmeny, povodnú hodnotu a novú hodnotu. V REST API by vznikol nový kontroler a service pre udalosti. Úprava by vyžadovala aj doplnenie stávajúcej implementácie, pretože by musela byť metódoda pre zaznamenanie zmeny zavolaná z každého miesta kde sa evidencia udalosti bude vyžadovať.

### 4.3 Prílohy

Implementácia príloh vložených do aplikácie by vyžadovala vytvorenie novej komponenty v užívateľskom prostredí, kde by bolo možné napríklad súbory typu pdf zobrazíť, stiahnuť alebo nahrať. Bolo by potrebné zvážiť akým spôsobom budú súbory spracovávané a ukladané. Samotná implementácia riešenia by bola náročnejšia ako predošlé navrhované rozšírenia, no bola by aj veľmi prínosná pre aplikáciu pretože by bolo možné priamo do aplikácie uložiť napríklad dokumenty potrebné k riešeniu úlohy alebo návrhy riešenia vypracované do súboru externe.

## Záver

V rámci tejto bakalárskej práce bola vytvorená webová aplikácia, ktorá umožní užívateľom jednoducho a prehľadne naplánovať projekt. Aplikácia je použiteľná v rôznych odvetviach. Umožňuje projektovým manažerom projekt naplánovať, a ostatným členom tímu sa prehľadne zorientovať v aktuálnych úlohách.

Vytvorenie aplikácie ma obohatilo o mnohé poznatky predovšetkým z oblasti konfigurácie REST-ového API a implementácie užívateľského prostredia, pretože tejto časti vývoja sa v praxi nevenujem. Táto skúsenosť bola pre mňa veľmi prínosná, pretože som mala možnosť realizácie projektu podľa vlastnej predstavy a zistila som čo všetko obnáša vývoj aplikácie úplne od začiatku až po hotový produkt.

V budúcnosti by som chcela aplikáciu rozšíriť o komentáre a históriu práce. Bolo by tak možné pri úlohách vidieť poznatky členov tímu získané v priebehu vykonávania úlohy a taktiež informácie o zmenách vykonaných na jednotlivých úlohách.

## Conclusions

As part of this bachelor's thesis, a web application was created that will allow users to easily and clearly plan the project. The application is applicable in various industries. It allows project managers to plan the project, and other team members to clearly navigate current tasks.

Creating the application enriched me with a lot of knowledge, especially in the field of REST API configuration and implementation user environment, because I do not deal with this part of development in practice. This experience was very beneficial for me because I had the possibility of realizing the project according to my own idea and I found out what the development of the application entails from the very beginning to the finished product.

In the future, I would like to expand the application with comments and work history. It would thus be possible to see knowledge in the tasks of team members obtained during the execution of the task, as well as information about changes made to individual tasks.

## 5 Obsah elektronických dát

Obsah elektronických dát odovzdaných v systéme katedry informatiky:

### **text/**

Táto zložka obsahuje všetky súbory potrebné pre vygenerovanie PDF dokumentu textu, tj. zdrojový text textu a príloh.

### **bin/**

Táto zložka obsahuje textový súbor README.txt, ktorý popisuje postup spustenia aplikácie. Virtuálny stroj, v ktorom sú pripravené súbory pre spustenie aplikácie.

### **src/**

Táto zložka obsahuje zdrojové kódy aplikácie. V podzložke `pat0` sú zdrojové kódy REST-ového API. V podzložke `pat0-ui` sú zdrojové kódy aplikácie užívateľského prostredia.

## Bibliografia

- [1] Getting started with Angular. Super-powered by Google [online]. 2022-02-28. [cit. 2023-04-20]. Dostupné z: <https://angular.io/start>
- [2] Getting Started with Angular Material. Super-powered by Google [online]. [cit. 2023-04-20]. Dostupné z: <https://material.angular.io/guide/getting-started>
- [3] Střecha, Tomáš. Úvod do Spring Boot frameworku pre Javu. [online]. [cit. 2023-04-20]. Dostupné z: <https://www.itnetwork.sk/java/spring-boot/uvod-do-spring-boot-frameworku-pre-javu>
- [4] PostgreSQL 15.2 Documentation. The PostgreSQL Global Development Group [online]. [cit. 2023-04-20]. Dostupné z: <https://www.postgresql.org/docs/current/>
- [5] Keycloak Admin REST API. [online]. [cit. 2023-04-20]. Dostupné z: <https://www.keycloak.org/docs-api/21.1.1/rest-api/index.html>