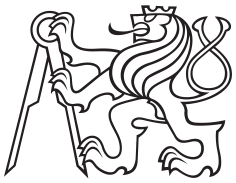
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Systém pro sběr zpětné vazby od uživatelů na základě existujícího issue trackeru

Timofei Likharev

Školitel: Ing. Jan Vrátník
Květen 2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Likharev** Jméno: **Timofei** Osobní číslo: **456901**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Softwarové systémy**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro sběr zpětné vazby od uživatelů na základě existujícího issue trackeru

Název bakalářské práce anglicky:

System for user feedback based on existing issue tracker

Pokyny pro vypracování:

Z dostupné dokumentace nastudujte existující API pro systém Jetbrains Youtrack. Důsledně analyzujte informace poskytované pomocí API. Navrhněte a implementujte samostatný systém využívající toto API pro import vybraných úkolů z Youtrack. Nad importovanými úkoly implementujte systém synchronizace s původním systémem. V implementovaném systému navrhněte a realizujte uživatelsky přívětivé uživatelské prostředí pro vizualizaci informací z importovaných úkolů. Implementujte systém, který zprostředkuje vybrané úkoly ze systému Youtrack pro veřejnost. Nad systémem implementujte základní diskuzi. Pomocí API udržujte informace a změny mezi původním úkolem v Youtrack a vaším systémem aktuální.

Seznam doporučené literatury:

[1] YouTrack REST API Reference [online]. [cit. 2018-01-16]. Dostupné z: https://www.jetbrains.com/help/youtrack/standalone/YouTrack-REST-API-Reference.html
[2] KALIN, Martin. Java Web services: up and running. Second edition. Sebastopol, California: O'Reilly, 2013. ISBN 1449365116.
[3] RICHARDSON, Leonard a Sam. RUBY. RESTful web services. Farnham: O'Reilly, c2007. ISBN 0596529260.
[4] SANDOVAL, José. RESTful Java web services: master core REST concepts and create RESTful web services in Java. Birmingham, U.K.: Packt Pub., 2009. From technologies to solutions.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jan Vrátník, katedra počítačů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2018** Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Jan Vrátník podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji vedoucímu své bakalářské práce pánu Ing. Janu Vrátníkovi za cenné rady, nápovědy a konzultace.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Abstrakt

Tato práce se zabývá návrhem a implementací aplikace pro sběr zpětné vazby od uživatelů na základě existujícího issue trackeru. Hlavním cílem je vytvoření takového systému, který usnadní vývojářům, kteří používají konkrétní issue tracker (YouTrack), komunikaci se zákazníky a zároveň poskytne zákazníkům jednoduchý způsob sledovat práci na projektech a ovlivnit ji svými komentáři. Aplikace byla implementována v programovacích jazycích TypeScript (s použitím Angular frameworku) a Java.

Klíčová slova: Angular, YouTrack, Java, Spring Framework, zpětná vazba

Školitel: Ing. Jan Vrátník

Abstract

This thesis deals with the design and implementation of a user feedback collection based on an existing issue tracker. The main goal is to create a system that will make it easier for developers who use a specific issue tracker (YouTrack) to communicate with customers and provide customers with an easy way to track project work and influence it with comments. The application was implemented in TypeScript programming languages (using Angular Framework) and Java.

Keywords: Angular, YouTrack, Java, Spring Framework, feedback

Title translation: System for user feedback based on existing issue tracker

Obsah

Obrázky

Tabulky

Kapitola 1
Úvod

Aplikace představuje systém pro sběr zpětné vazby od uživatelů na základě existujícího issue trackeru YouTrack, se kterým bude propojena. Uživatel je v zadání chápán jako zákazník softwarové firmy, která vyvíjí nějaký produkt.

Cílem je navrhnout a implementovat systém, který umožní firmám dostávat zpětnou vazbu od svých zákazníků, např. hlášení o bugu v systému nebo nový požadavek. Další možnosti pro zákazníky jsou sledování aktuálních issues na projektu (v jakem jsou stavu, kdo na tom pracuje atd.) a diskuze nad nimi.

Aplikace musí poskytovat co nejjednodušší způsob komunikace se zákazníkem, který bude pohodlný jak pro uživatele tohoto systému (pro zákazníky), tak i pro vývojáře, kteří budou reagovat na zpětnou vazbu od zákazníků přímo z YouTrack. Tuto aplikaci lze využívat po dobu vývoje i po dobu údržby.

1.1 Motivace

Toto zadání lze splnit vhodným nastavením uživatelských skupin a rolí přímo v YouTrack, ale existuje několik důvodů, proč je výhodnější vytvořit samostatný systém:

1. YouTrack je pro neprogramátora extrémně komplikovaný.
2. YouTrack je profesionální technický systém, proto není úplně zaměřen na krásné a nejmodernější grafické uživatelské rozhraní, což by mohlo vadit obyčejnému zákazníkovi.
3. Dává smysl nedávat zákazníkovi přístup do interního firemního systému, i když tam neuvidí žádná soukromá data.
4. Je zbytečné dávat zákazníkovi k dispozici všechny nástroje co poskytuje YouTrack.
5. Automatické nastavení, které udělá aplikace, ušetří vývojářům čas.

Kapitola 2
Analýza

Uživatel je v názvu práce chápán jako zákazník softwarové firmy, která vyvíjí nějaký produkt. V tomto významu bude slovo zákazník používáno v celé práci. Slovo uživatel se používá ve významu obecný uživatel.

2.1 Požadavky

2.1.1 Funkční požadavky

Funkční požadavky popisují jaké akce se dají v systému provádět. Pro přehlednost jsou rozdělené do několika částí.

- Zákazník
  - Projekty
    - **FR1:** Zobrazení seznamu povolených projektů.
    - **FR2:** Zobrazení informací o zvoleném projektu.
  - Issues
    - **FR3:** Zobrazení seznamu issues ve zvoleném projektu.
    - **FR4:** Hledání issues podle zadaných parametrů.
    - **FR5:** Zobrazení informací o zvoleném issue.
    - **FR6:** Vytváření issue.
    - **FR7:** Editace issue.
  - Komentáře
    - **FR8:** Zobrazení komentářů ke zvolenému issue.
    - **FR9:** Přidání nového komentáře k issue.
    - **FR10:** Editace komentáře.
    - **FR11:** Mazání komentáře.
  - Ostatní
    - **FR12** Podpora formátu markdown u komentáře a popisu issue.

- Administrátor
  - Zákazníci
    - **FR13:** Zobrazení seznamu zákazníků.
    - **FR14:** Zobrazení informací o zvoleném zákazníkovi.
    - **FR15:** Vytváření nového zákazníka.
    - **FR16:** Mazání zákazníka.
  - Projekty
    - **FR17:** Zobrazení seznamu projektů.
    - **FR18:** Zobrazení informací o zvoleném projektu.
    - **FR19:** Zapnutí sběru zpětné vazby pro projekt.
    - **FR20:** Přidání zákazníkovi přístupu k projektu.
    - **FR21:** Zrušení zákazníkovi přístupu k projektu.
    - **FR22:** Přepínaní viditelnosti issues z původního projektu.
  - Ostatní
    - **FR23:** Registrace administrátora.
- Obecné
  - **FR24:** Aktivace uživatele.
  - **FR25:** Přihlášení uživatele.
  - **FR26:** Odhlášení uživatele.
  - **FR27:** Obnovení zapomenutého hesla.
  - **FR28:** Změna hesla v profilu.

2.1.2 Nefunkční požadavky

Nefunkční požadavky definují omezení služeb nebo funkcí, které systém poskytuje.

- **NFR1:** Uživatelské rozhraní bude implementované jako webová aplikace s použitím některého z moderních frontendových frameworků (viz *4.1. Klientská část*).
- **NFR2:** Využití rozhraní YouTrack REST API, Hub REST API ke práci s daty.
- **NFR3:** Zákazník nesmí mít možnost přihlásit se přímo do systému YouTrack.
- **NFR4:** Zákazník nesmí vidět interní vývojářské komentáře u původních issues, které jsou vytvořené vývojáři.

2.2 JetBrains YouTrack

YouTrack je aplikace od JetBrains, která se používá pro issue-tracking a projektový management. V této části je uživatel chápán jako uživatel YouTrack/Hub.

Mezi hlavní funkce aplikace patří:

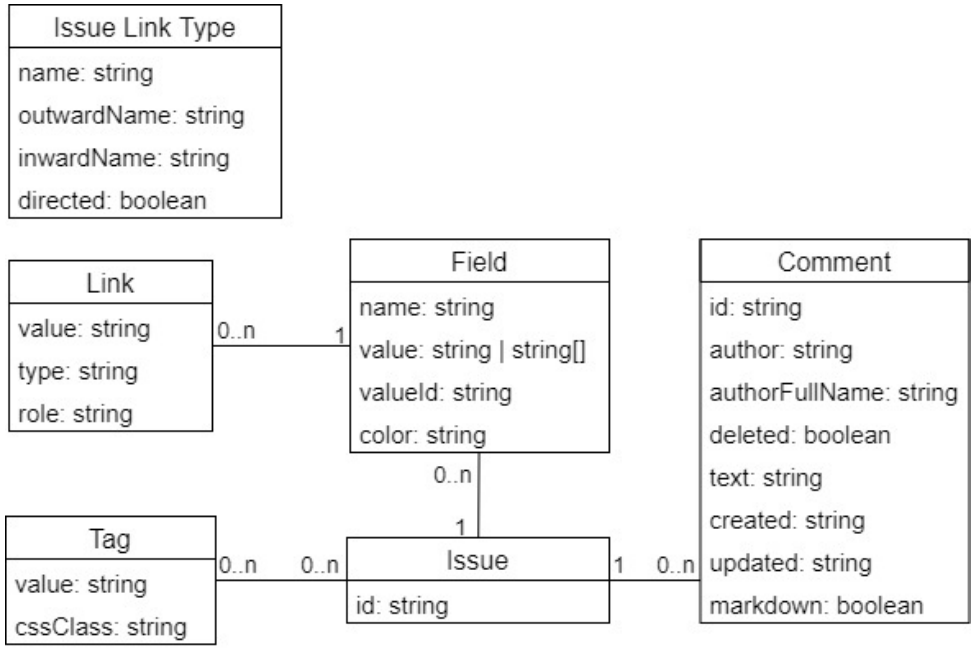
- vytvořit nový projekt,
- přidat uživatele do projektového týmu,
- vytvořit nové issue, nastavit parametry (priorita, typ, stav atd.),
- napsat komentář k issue.

Pro přístup k aplikaci bude použito rozhraní YouTrack REST API.

2.2.1 YouTrack REST API

YouTrack poskytuje rozhraní YouTrack REST API, které se především používá ke správě issues. Umožňuje provádět všechny základní CRUD-operace nad issues a komentáři.

Model



Obrázek 2.1: Zjednodušený datový model YouTrack REST API

Diagram reprezentuje zjednodušený datový model, k němuž poskytuje přístup rozhraní YouTrack REST API. Skutečný model je složitější, ale není potřeba ho používat celý.

Popis modelu:

- **Field** reprezentuje většinu atributů issue (např. prioritu, stav, typ, popis, odkaz na jiné issue).
- **Link** je odkaz určitého typu (**IssueLinkType**) na jiné issue.
- **IssueLinkType** popisuje typ odkazu. Kdyz je **directed**, vazba mezi issues je oboustranná. To znamená, že když uživatel přidá odkaz **outwardName** z issue A do issue B, pak se automaticky přidá odkaz **inwardName** z B do A. Když není **directed** zpáteční odkaz z B do A se nepřidává.

Použití

Aplikace bude používat rozhraní jako obyčejné REST API, tj. zasláním HTTP-dotazů a obdržením odpovědí.
Ukázka použití rozhraní pro přidání nového issue:

- **Request:**

```
PUT <YouTrack REST API URL>/issue?project=P1&summary=newSummary
&description=newDescription
```

- **Request parameters:**

Název	Typ	Popis
project	string	ID projektu
summary	string	Krátký popis issue (název issue)
description	string	Delší popis issue
attachments	soubor ve formatu "multipart/form-data"	Přílohy k novému issue
permittedGroup	string	Uživatelská skupina, která bude mít právo vidět toto issue

Tabulka 2.1: HTTP parametry pro přidání nového issue

- **Response:**

```
HTTP/1.1 201 Created
Content-Type: application/xml;charset=UTF-8
Location:<YouTrack REST API URL>/rest/issue/P1-1
Content-Length: 0
```

2.3 JetBrains Hub

Hub je integrační a administrační služba, která se používá pro řízení uživatelů, skupin, rolí, práv, autorizaci a spojení s jinými týmovými aplikacemi od JetBrains. V teto části uživatel je chápán jako uživatel YouTrack/Hub.

Mezi hlavní funkce aplikace patří:

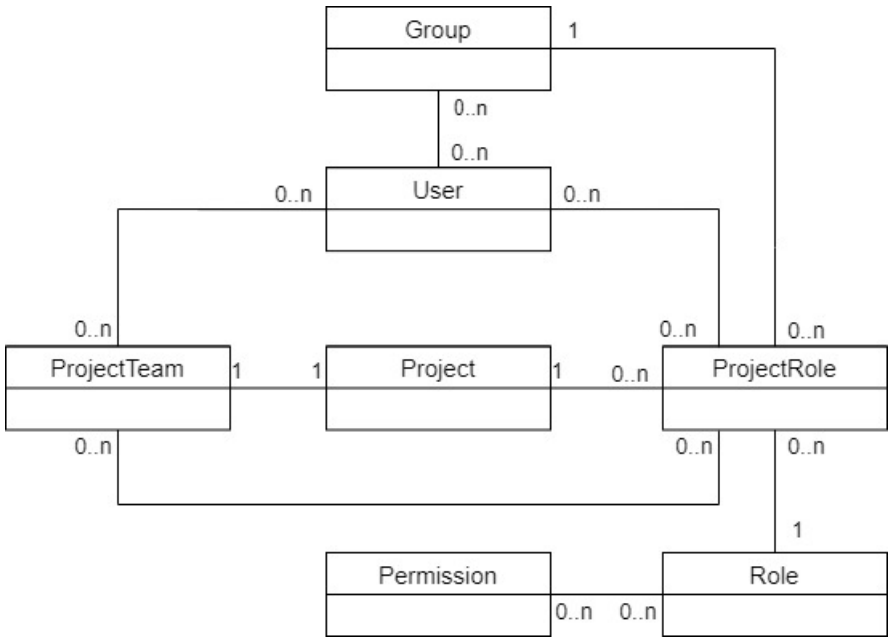
- vytvořit nového uživatele,
- vytvořit novou skupinu,
- přidat uživatele do skupiny,
- přidat roli do skupiny.

Pro přístup k aplikaci bude použito rozhraní Hub REST API.

2.3.1 Hub REST API

Hub REST API je podobné rozhraní jako YouTrack REST API. Umožňuje provádět všechny základní CRUD-operace nad projekty, uživateli, uživatelskými skupinami, projektovými týmy, rolemi a právy.

Model



Obrázek 2.2: Zjednodušený datový model Hub REST API

Diagram reprezentuje zjednodušený datový model, k němuž poskytuje přístup rozhraní Hub REST API. Skutečný model je složitější, ale není potřeba ho používat celý.

Popis modelu:

- Každý uživatel může být členem uživatelské skupiny, která obsahuje role pro určité projekty.
- Každá role je definována právy, které poskytuje uživatelům.
- Projekt má svůj projektový tým pro vývojáře, který je podobný uživatelské skupině (může obsahovat role pro určité projekty).

Použití

Aplikace bude používat rozhraní jako obyčejné REST API, tj. zasláním HTTP-dotazů a obdržením odpovědí.
Ukázka použití rozhraní pro vytváření projektu:

- **Request:**

```
POST <Hub REST API URL>/projects?fields=id,key,name
```

- **Request parameters: fields** (string) - Nepovinný parametr který definuje jaké atributy projektu se mají vrátit v těle odpovědi.
- **Request Body (ve formátu JSON):**

```
{
   "key": "NP",
   "name" : "NewProject",
   "description" : "NewProject"
}
```

- **Response:** 200 OK
- **Response Body (ve formátu JSON):**

```
{
   "id": "1cd022a9-6e7a-4ade-b495-01cbd7e5063d",
   "key": "NP",
   "name" : "NewProject",
}
```

2.3.2 Role a práva

Je několik předdefinovaných rolí, které je možné v této aplikaci využít:

- **Observer** - vidí issues, informaci o issues a komentáře k nim.
- **Reporter** - má stejná práva jako **Observer**, navíc může vytvářet a editovat své vlastní issues, a psát komentáře ke všem issues.
- **Developer** - má stejná práva jako **Reporter**, navíc může editovat všechna issues.

2.3.3 Autorizace

Jsou dva způsoby autorizace:

1. **Oauth 2.0 autorizace:**
   a. Při přihlášení aplikace přesměruje uživatele na přihlašovací stránku.
   b. Uživatel zadá přihlašovací jméno a heslo od YouTrack.
   c. Vygeneruje se dočasný přihlašovací token.
   d. Aplikace uloží token do paměti a bude používat pro komunikaci se systémy (YouTrack a Hub) přes REST rozhraní (ve hlavičce Authorization při HTTP dotazu).
   e. Aplikace smaže token po odhlášení.
2. **Autorizace pomocí permanentních tokenů:**
   a. Uživatel vytvoří permanentní token v YT a zkopíruje ho.
   b. Uživatel vloží zkopírovaný token do aplikace.
   c. Aplikace uloží token do databáze.
   d. Po správném přihlášení aplikace načte token z databáze do paměti a bude používat pro komunikaci se systémy (YouTrack a Hub) přes REST rozhraní (ve hlavičce Authorization při HTTP dotazu).

Podle požadavku **NFR3** nesmí zákazník znát login a heslo k YT, proto v aplikaci bude použit druhý způsob autorizace (pomocí permanentních tokenů).

Kapitola 3
Návrh

Z analyzovaných požadavků vyplývá, že aplikace musí poskytovat různé rozhraní pro uživatele-zákazníky a uživatele-administrátory.

3.1 Administrátorská část

Administrátorská část je určená pro administrátory firemních systémů YouTrack. Každý administrátor se může zaregistrovat v tomto systému a připojit ho k YouTrack pomocí API tokenu, který vytvoří přes nastavení YouTrack (viz *3.6.1 Administrátorská část - Registrovat se*). Administrátor musí přidat adresu této aplikace do seznamu CORS-povolených adres v nastavení YouTrack.

Po přihlášení aplikace zobrazí všechny existující projekty a všechny vytvořené zákazníky. Hlavní možnosti jsou zapnutí zpětné vazby pro projekt, vytváření zákazníků a přidání/zrušení zákazníkům přístupu k projektu.

3.2 Zákaznická část

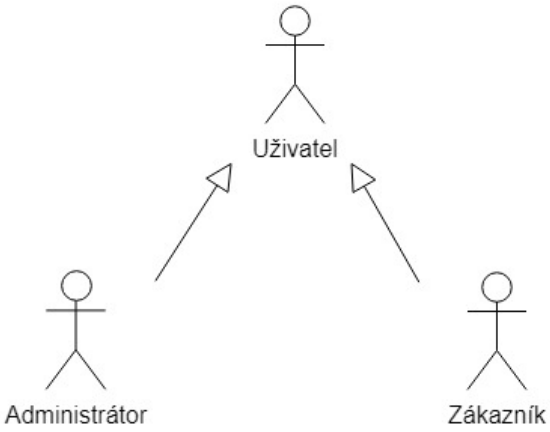
Zákaznická část je určena pro reálné zákazníky, pro které firma vyvíjí nějaký produkt. Firma chce jednoduchým a pohodlným způsobem dostat zpětnou vazbu od zákazníků (např. hlášení o bugu v systému, který se vyvíjí, nějaký nový požadavek nebo komentář k aktuálnímu issue z projektu). Administrátor firemního YouTrack přidá zákazníka do tohoto systému, který po přihlášení zobrazí seznam povolených projektů, včetně issues a dalších informace (viz *3.4.5 Zobrazení seznamu issues*).

Dále aplikace umožňuje přidávat nové issue, případně sledovat práci na projektu, tj. dostávat informaci o současných issues a komentovat je.

Zákazník má přístup jenom k tomuto systému, nikoliv k YouTrack (podle požadavku **NFR3**).

3.3 Role

Uživatelské role definují jaké akce může uživatel s konkrétní rolí provádět.



Obrázek 3.1: Uživatelské role

1. **Uživatel** - libovolný uživatel systému. Role je určena k použití společných funkcí jako přihlášení nebo obnovení zapomenutého hesla.
2. **Zákazník** - reálný zákazník firmy, uživatel zákaznické části systému.
3. **Administrátor** - administrátor firemního YouTrack, uživatel administrátorské části systému.

Další informace o rolích jsou popsány v části *3.4.1 Uživatelský účet*.

3.4 Logika aplikace

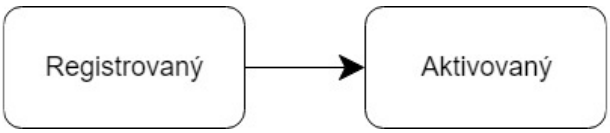
3.4.1 Uživatelský účet

Každý uživatel má účet v YouTrack a záznam v databázi teto aplikaci (viz *3.7.2 Databázový model*).

Administrátor

Při registraci má uživatel zadat svůj API token, jehož majitel má administrátorská práva v YouTrack, jinak aplikace nebude fungovat. Administrátor má již předem účet v YouTrack, po registraci aplikace uloží informaci o administrátorovi do databáze. Při registrace musí zadat stejný login jako v YouTrack.

Následující stavový diagram popisuje možné stavy uživatele s rolí administrátor:



Obrázek 3.2: Stavy administrátorů

- Administrátor je **registrovaný**, když se zaregistroval (viz *3.6.1 Administrátorská část - Registrovat se*), ale ještě neaktivoval svůj účet pomocí odkazu zaslanému na email.
- Administrátor je **aktivovaný**, když aktivoval svůj účet (viz *3.6.2 Společná část - Aktivovat účet*).

Zákazník

Zákazník nemá předem účet v YouTrack, vytváří se administrátorem přes tento systém (v administrátorské části).

Přestože aplikace může provést všechny akce v zákaznické části pomocí administrátorského tokenu (pro většinu akcí, např. správu uživatelských skupin nebo uživatelů, potřebuje pravě administrátorský token), musí každý zákazník mít svůj vlastní, protože jenom na základě tokenu YouTrack přidává autora k issues a komentářům při jejich vytváření.

Zákazník nesmí mít přímý přístup k YouTrack (podle požadavku **NFR3**), ale je potřeba vytvořit API token přes YouTrack. Z tohoto důvodu, zákaznické účty v YouTrack a API tokeny vytváří administrátor (viz *3.6.1 Administrátorská část - Vytvořit nového zákazníka*), zákazník pak jenom aktivuje svůj účet pomocí linku zaslanému na email (viz *3.6.2 Společná část - Aktivovat účet*).

Každý zákazník je v YouTrack členem speciální skupiny 'Clients' (tuto skupinu aplikace vytvoří při počátečním nastavení (viz *3.4.2 Počáteční nastavení aplikace*). V rámci jednoho projektu může být více zákazníků (např. ředitel zákaznické firmy, technický konzultant zákaznické firmy atd.)

Následující stavový diagram popisuje možné stavy uživatele s rolí zákazník:

Obrázek 3.3: Stavy zákazníků

- Zákazník je **vytvořený**, když ho vytvořil administrátor (má jenom účet v YouTrack) (viz *3.6.1 Administrátorská část - Vytvořit nového zákazníka*).
- Zákazník je **registrovaný**, když administrátor přidal mu API token (má účet v YouTrack a záznam v databázi) (viz *3.6.1 Administrátorská část - Vytvořit nového zákazníka*).
- Zákazník je **aktivovaný** když aktivoval svůj účet (viz *3.6.2 Společná část - Aktivovat účet*).

3.4.2 Počáteční nastavení aplikace

Po prvním přihlášení administrátora aplikace provede v YouTrack počáteční nastavení:

1. Vytvoří uživatelskou skupinu 'Clients'. Každý zákazník v systému je její členem.
2. Vytvoří roli 'Client' s právy na vytváření tokenů, tagů, přidání odkazů na issue. Umožňuje administrátorovi vytvořit API token přes zákaznický účet. Umožňuje aplikaci přidávat tagy a odkazy.
3. Přidá roli 'Client' do skupiny 'Clients'.
4. Vytvoří **IssueLinkType** (viz *2.2.1 Youtrack REST API - Model*) s následujícími hodnotami:

```
{"name" : "Feedback",
"outwardName": "origin for",
"inwardName": "feedback for",
"directed": true}
```

   Používá se pro vytváření issue typu feedback (viz *3.4.4 Typy přístupu k issue*).

3.4.3 Samostatný projekt pro feedback

Ke každému projektu (s názvem <ProjectName>), ke kterému bude zapnut sběr zpětné vazby (viz *3.6.1 Administrátorská část - Zapnout sběr zpětné vazby pro projekt*), vytvoří aplikace v YouTrack nový projekt s názvem <ProjectName (feedback)> z důvodu oddělení interních issues od nových issues které vytvoří zákazník (viz *3.4.4 Typy přístupu k issue*) a oddělení komunikace se zákazníkem od komunikace mezi vývojáři (viz *3.4.8 Zobrazení a přidání komentářů k issue*), aby se aktuální projektové úkoly a diskuze nemíchaly s jakoukoliv zpětnou vazbou od zákazníků.

Každý feedback-projekt má svoji uživatelskou skupinu '<ProjectKey>-clients' pro zákazníky s nastavenými rolemi **Observer** pro původní projekt a **Reporter** pro feedback-projekt (viz *2.3.2 Role a práva*). Takovým způsobem, každý uživatel z této skupiny dostane možnost vidět issues z původního projektu, vidět issues z feedback-projektu a přidávat tam svoje issues. Aplikace přidá přístup s rolí **Developer** všem vývojářům s příslušného původního projektu.

Po zapnutí zpětné vazby pro projekt aplikace uloží příslušný záznam do databáze, který obsahuje ID původního projektu, feedback-projektu, uživatelské skupiny '<ProjectKey>-clients' a logickou hodnotu 'Show origin issues', která určuje viditelnost issues z původního projeku pro zákazníky.

Dále se v textu používají názvy původní projekt (nebo origin-projekt), tj. původní vývojářský projekt z YouTrack a feedback-projekt, tj. projekt pro zpětnou vazbu od zákazníků.

3.4.4 Typy přístupu k issue

Rozlišují se tři různé typy issue podle přístupu k nim v teto aplikaci:

1. **Origin** - issues z původního projektu. Issues tohoto typu mají odkaz 'origin for' na příslušné feedback-issue (typ feedback), když takové issue existuje. Tento odkaz potřebuje aplikace pro načtení a přidání komentářů zákazníků k původnímu issue (typ origin). Zároveň vývojáři vidí u jakého issue probíhá konverzace se zákazníkem ohledně původního issue.

2. **New** - nové issues vytvořené zákazníkem. Přidávají se do příslušného feedback-projektu z důvodu oddělení od interních původních issues. Issues tohoto typu mají tag 'New'.

3. **Feedback** - speciální typ issue, místo pro diskuzi mezi zákazníky a vývojáři nad původním issue (příslušné issue typu origin). Tento typ se používá pro oddělení komunikace se zákazníky od komunikace mezi vývojáři. Vytváří se automaticky ve feedback-projektu v moment, když zákazník chce napsat první komentář k issue s typem origin. Aplikace zkontroluje zda existuje odkaz 'origin for' na příslušné feedback-issue, ve případě když neexistuje vytvoří ho, pak přidá komentář k tomuto feedback-issue. (viz *3.4.8 Zobrazení a přidání komentářů k issues*). Issues tohoto typu mají název '<OriginIssueSummary> (feedback)' a odkaz 'feedback for' na původní origin-issue (typ origin).

Následující tabulka stručně popisuje výše uvedené typy issue:

Typ	Projekt	Autor	Jak odlišit
origin	Původní	Vývojář	nic nebo odkaz 'origin for' na feedback-issue
new	Feedback	Zákazník	tag 'New'
feedback	Feedback	Aplikace	odkaz 'feedback for' na origin-issue

Tabulka 3.1: Typy přístupu k issue

Typ přístupu se nikdy nezobrazí v uživatelském rozhraní, je používán pouze aplikací. Typ přístupu není to samé jako typ issue, který je jedním atributem issue v YouTrack. Dále v textu ve významu typ přístupu se používá zkrácený název typ. Je vždycky jasné z kontextu, o jakém typu jde.

Použití typů přístupu je popsáno dále.

3.4.5 Zobrazení seznamu issues

Při defaultním nastavení aplikace zobrazí issues dvou typů: origin a new. Administrátor má možnost vypnout zobrazení issues z původního projektu (typ origin), aby zákazník mohl vidět jenom nové issues vytvořené zákazníky (typ new) (viz *3.6.1 Administrátorská část - Přepnout viditelnost issues z původního projektu*).

3.4.6 Zobrazení informací o issue

Informace o zvoleném issue (název, popis, parametry) se vždycky načtou z tohoto issue (bez závislosti na typu issue, buď origin nebo new). Podle požadavku **FR12** aplikace podporuje zobrazení popisu issue ve formátu markdown (pomocí knihovny, viz *4.1.3 Použité knihovny*).

3.4.7 Editace issue

Editovat issue může jenom jeho autor.

3.4.8 Zobrazení a přidání komentářů k issue

Podle požadavku **NFR4** zákazník nesmí vidět interní vývojářské komentáře k issues z původního projektu (typ origin), místo toho aplikace zobrazí komentáře k příslušnému feedback-issue.

Dělá se to z důvodu oddělení interní komunikace mezi vývojáři od komunikace se zákazníkem.

Takový způsob má dvě výhody:

- Z pohledu zákazníka to vypadá, jako by psal komentář přímo k původnímu issue.
- Vývojáři tyto komentáře vidí a odpovídají na ně u příslušného 'feedback' issue z názvem '<OriginIssueSummary> (feedback)', na které má původní issue odkaz 'origin for'.

Nemíchají se původní komentáře u interních issues se zákaznickými komentáři k těmto issues, a odpověďmi na ně.

Komentáře k issue se zobrazí a přidávají v závislosti na typu issue:

- **Origin:**
  - zobrazí se komentáře z příslušného issue typu feedback,
  - komentáře se v YouTrack přidávají k příslušnému feedback-issue. Když feedback-issue neexistuje, aplikace ho vytvoří (s názvem <OriginIssueSummary> (feedback)) ve feedback-projektu a přidá odkaz 'feedback for' na origin-issue.
- **New:**
  - zobrazí se komentáře z tohoto issue,
  - komentáře se přidávají k tomuto issue.

Aplikace podporuje zobrazení a přidání komentářů ve formátu markdown (podle požadavku **FR12**).

3.4.9 Editace a mazání komentářů

Editovat nebo smazat komentář může jenom jeho autor.

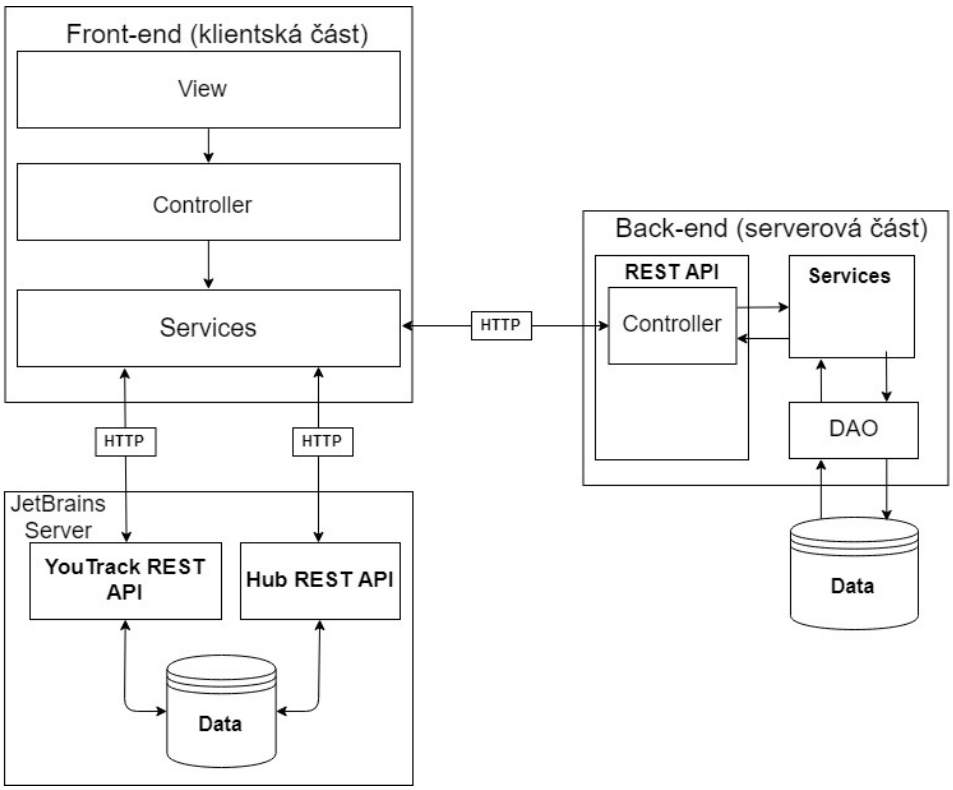
3.5 Struktura aplikace

Podle požadavku **NFR2** aplikace přistupuje k datům pomocí rozhraní YouTrack a Hub REST API. Všechna data z YouTrack musí být aktuální, proto nemá smysl ukládat do vlastní databáze data, která aplikace může načíst přímo z YouTrack.

Z důvodu použití způsobu autorizace pomocí permanentních tokenů (viz *2.3.3 Autorizace*) aplikace potřebuje vlastní databázi, kam tokeny bude ukládat spolu s příslušnými uživatelskými údaji.

Aplikace potřebuje někam uložit informaci o závislosti mezi původním projektem, příslušným feedback-projektem, příslušnou uživatelskou skupinou a logickou hodnotou 'Show origin issues for clients'. Není vhodný způsob, když tato vazba bude zaručena jenom tím, že jméno feedback-projektu je '<ProjectName (feedback)>' a jmeno skupiny je '<ProjectKey>-clients', protože ta jména může někdo změnit. Je potřeba přidat silnější vazbu, proto aplikace bude ukládat tato data do databáze.

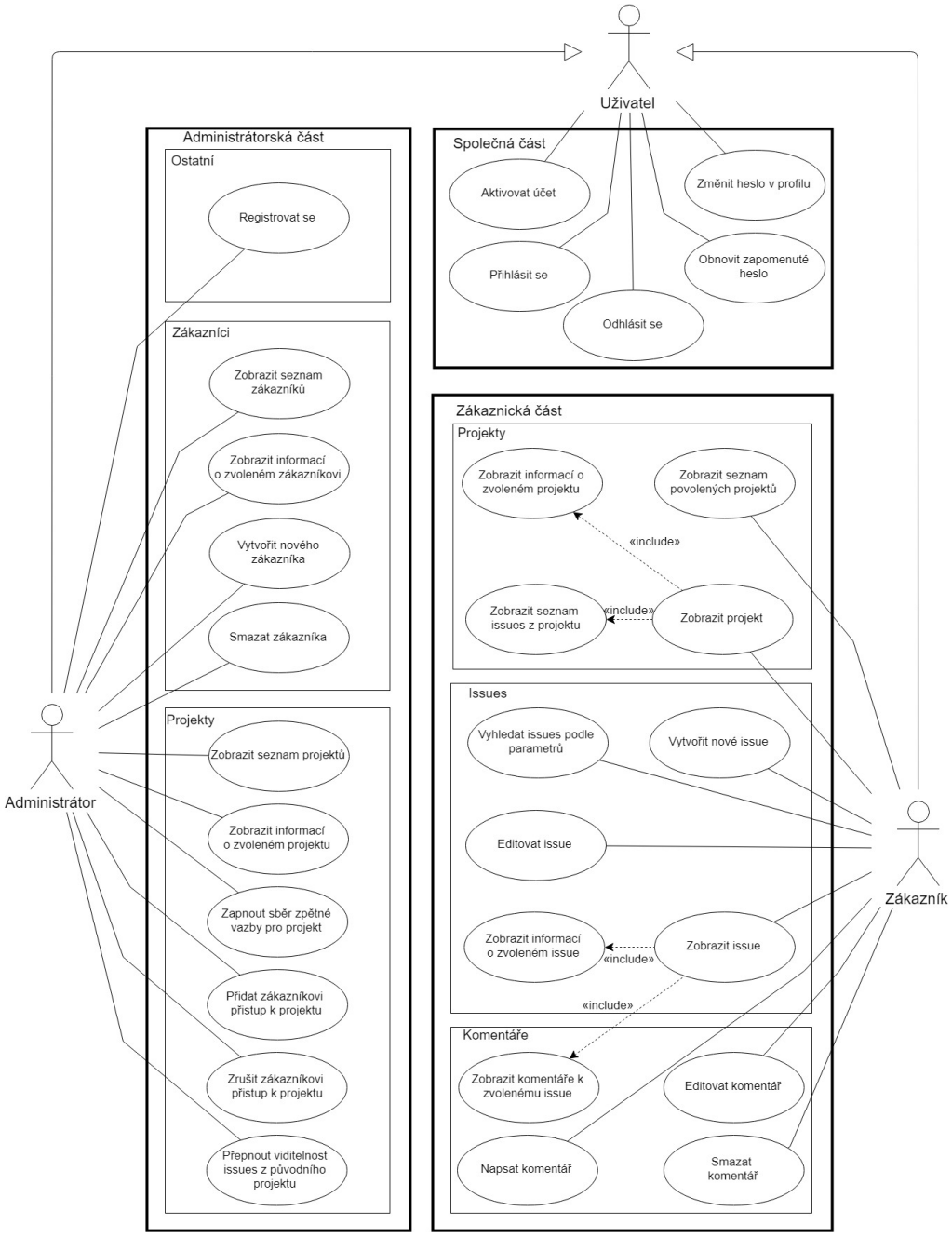
Celá aplikace bude rozdělena na dvě častí: webový klient a backend server. Serverová část se bude používat pro uložení uživatelských údajů, informací o projektech, a funkce související se správou těchto údajů, např. autorizace, změna hesla nebo mazání uživatele, načtení projektů.



Obrázek 3.4: Struktura aplikace

3.6 Případy užití

Možná interakce mezi uživatelem (s konkrétní rolí) a aplikací se definuje seznamem případů užití. Každý případ užití je využíván aspoň jednou rolí a splňuje aspoň jeden funkční požadavek.



Obrázek 3.5: Diagram případů užití

Dále jsou popsány složitější případy užití.

3.6.1 Administrátorská část

Registrovat se

1. Uživatel klikne na tlačítko 'Sign up for admin'.
2. Aplikace zobrazí formulář.
3. Uživatel zadá potřebné údaje (login, name, email, URL YouTrack, API token) a stiskne tlačítko 'Sign up'. Login musí být stejný jako v YouTrack.
4. Aplikace vytvoří uživatele v databázi a odešle aktivační link na email.

(Dále viz *3.6.2 Společná část - Aktivovat účet*)

Vytvořit nového zákazníka

1. Po přihlášení administrátor rozklikne formulář na přidání nového zákazníka.
2. Administrátor zadá potřebné údaje(login, name, email a heslo od YouTrack) a stiskne tlačítko 'Create'.
3. Aplikace vytvoří uživatele se zadanými daty v systému YouTrack (Hub), přidá ho do uživatelské skupiny 'Clients' a zobrazí ho v seznamu zákazníků v aplikaci.
4. Administrátor použije login a heslo pro přihlášení do zákaznického účtu YouTrack.
5. Administrátor vytvoří API token pro zákazníka přes nastavení YouTrack a zkopíruje ho.
6. Administrátor se vrátí do aplikace, klikne na nového zákazníka v seznamu.
7. Aplikace zobrazí formulář na vložení API tokenu pro zvoleného zákazníka.
8. Administrátor vloží token do formuláře a stiskne tlačítko 'Register'.
9. Aplikace vytvoří zákazníka v databázi a pošle mu aktivační link na email.

(Dále viz *3.6.2 Společná část - Aktivovat účet*)

Zapnout sběr zpětné vazby pro projekt

1. Administrátor zvolí projekt ze seznamu projektů.
2. Administrátor stiskne tlačítko 'Start collecting feedback'.
3. Aplikace vytvoří v YouTrack samostatný projekt pro zpětnou vazbu s názvem '<ProjectName> (feedback)'.

4. Aplikace vytvoří v YouTrack uživatelskou skupinu s názvem '<ProjectKey>-clients' pro zákazníky (s rolemi **Observer** pro <ProjectName> a **Reporter** pro <ProjectName> (feedback)).

5. Aplikace přidá do příslušného projektového týmu roli **Developer** pro projekt <ProjectName> (feedback) .

6. Aplikace přidá záznam do databáze, který obsahuje ID původního projektu, feedback-projektu, uživatelské skupiny '<ProjectKey>-clients' a logickou hodnotu 'Show origin issues'.

Přidat zákazníkovi přístup k projektu

Zákazník po přihlášení vidí jenom ty projekty, ke kterým má přístup. Administrátor může přidat zákazníkovi přístup následujícím způsobem:

1. Administrátor zvolí projekt ze seznamu.

2. Administrátor klikne na rozbalovací seznam.

3. Zobrazí se všechny zákazníky (zaškrtnutí = přístup k projektu).

4. Administrátor přidá zákazníka kliknutím na příslušný checkbox a stiskne tlačítko 'Update'.

5. Aplikace přidá zákazníka do uživatelské skupiny v YouTrack příslušící zvolenému projektu.

To, že zákazník má přístup k projektu neznamená, že vidí issues z původního projektu (toto se přepíná administrátorem, viz *3.4.5 Zobrazení seznamu issues* a *3.6.1 Administrátorská část - Přepnout viditelnost issues z původního projektu*).

Zrušit zákazníkovi přístup k projektu

Podobným způsobem administrátor může zrušit zákazníkovi přístup k projektu následujícím způsobem:

1. Administrátor zvolí projekt ze seznamu.

2. Administrátor klikne na rozbalovací seznam.

3. Zobrazí se všechny zákazníky (zaškrtnutí = přístup k projektu).

4. Administrátor odebere zákazníka kliknutím na příslušný checkbox a stiskne tlačítko 'Update'.

5. Aplikace odebere zákazníka z uživatelské skupiny v YouTrack příslušící zvolenému projektu.

Přepnout viditelnost issues z původního projektu

Administrátor může přepínat viditelnost issues z původního projektu (typ origin):

1. Administrátor zvolí projekt ze seznamu.
2. Administrátor přepne viditelnost kliknutím na checkbox 'Show origin issues for clients'.
3. Aplikace uloží změnu do databáze.

3.6.2 Společná část

Aktivovat účet

Po registraci uživatel obdrží email s aktivačním odkazem, následující postup je:

1. Uživatel klikne na aktivační odkaz a přesměruje se na stránku aktivace.
2. Aplikace zkontroluje, zda aktivační odkaz je platný.
3. Aplikace zkontroluje roli uživatele:
   a. Když je administrátor, pokračuje bodem 7.
   b. Když je zákazník, pokračuje bodem 4.
4. Aplikace zobrazí formulář na zadání nového hesla.
5. Zákazník zadá nové heslo a stiskne tlačítko 'Submit'.
6. Aplikace uloží heslo zákazníka do databáze.
7. Aplikace přesměruje uživatele na přihlašovací stránku s předvyplněným loginem.

Přihlásit se

1. Po správném přihlášení aplikace zkontroluje roli uživatele.
   a. Když je zákazník, pokračuje bodem 3.
   b. Když je administrátor, pokračuje bodem 2.
2. Aplikace provede počáteční nastavení, jestliže ještě nikdy nebylo provedeno (viz *3.4.2 Počáteční nastavení aplikace*).
3. Zobrazí se uživatelský profil (v závislosti na roli uživatele).

Obnovit zapomenuté heslo

1. Uživatel klikne na odkaz 'Forgotten password' na přihlašovací stránce.
2. Aplikace zobrazí formulář na zadání loginu.
3. Uživatel zadá login a stiskne tlačítko 'Send'.
4. Aplikace zkontroluje, zda takový uživatel existuje.
5. Aplikace vygeneruje odkaz s hashi na obnovení hesla, uloží hash a čas expiraci v databázi a pošle odkaz na email uživatele.
6. Uživatel klikne na odkaz a přesměruje se na stránku zadání nového hesla.
7. Uživatel zadá nové heslo a stiskne tlačítko 'Submit'.
8. Aplikace porovná hash z odkazu a hash z databáze.
9. Aplikace zkontroluje, jestli platnost odkazu ještě není expirovala.
10. Aplikace uloží heslo uživatele do databáze a přesměruje uživatele na přihlašovací stránku s předvyplněným loginem.

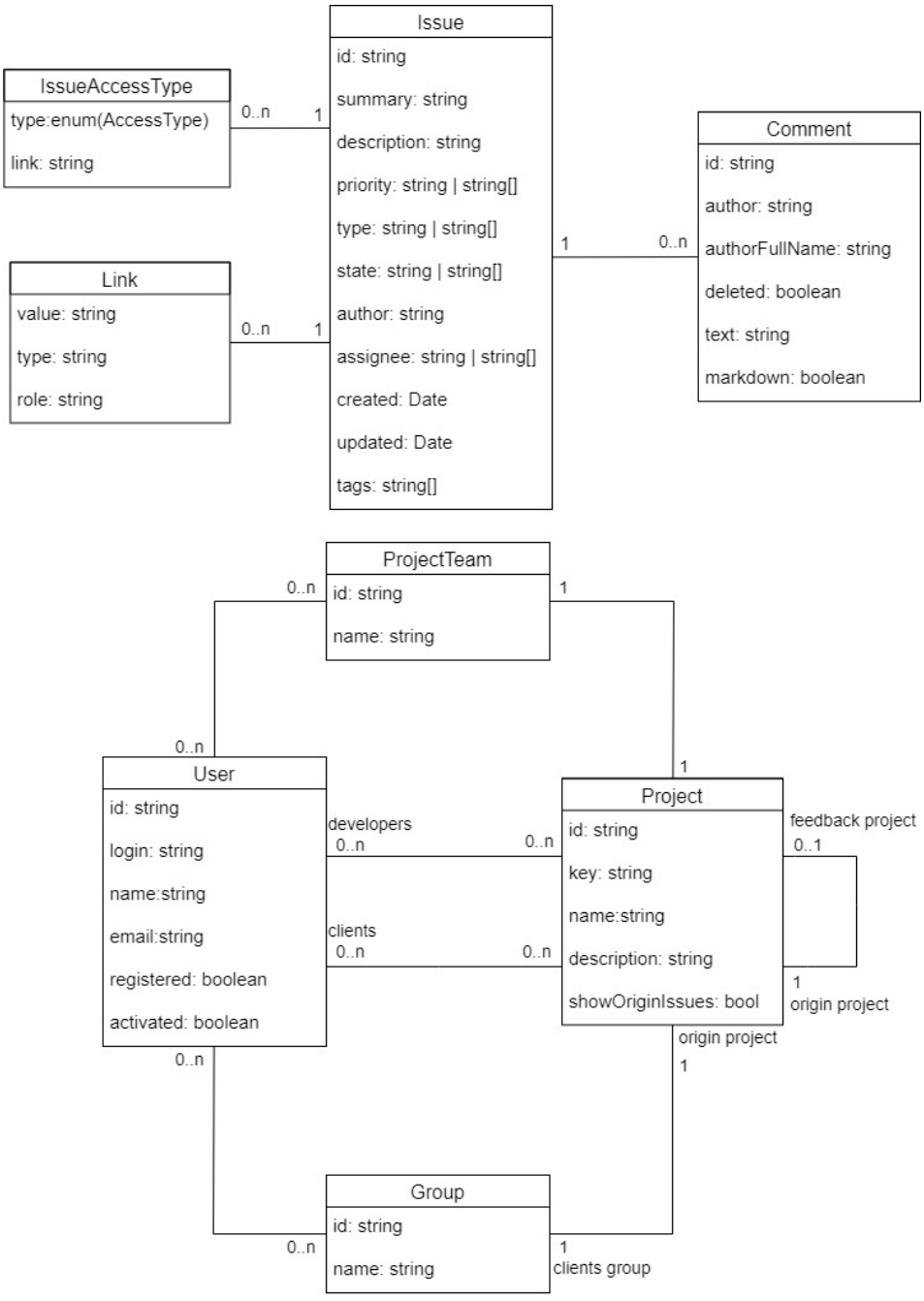
3.7 Model

Aplikace je rozdělena na dvě části, proto jsou dva různé datové modely, které jsou popsány dále.

3.7.1 Klientská část aplikace

Všechna potřebná data aplikace vždy načte z YouTrack při zobrazení jednotlivých stránek aplikace. Aplikace uloží tato data jenom do paměti a znovu je načte po obnovení stránky.

Následující diagram reprezentuje datový model na klientské straně aplikace.



Obrázek 3.6: Datový model klientské častí aplikace

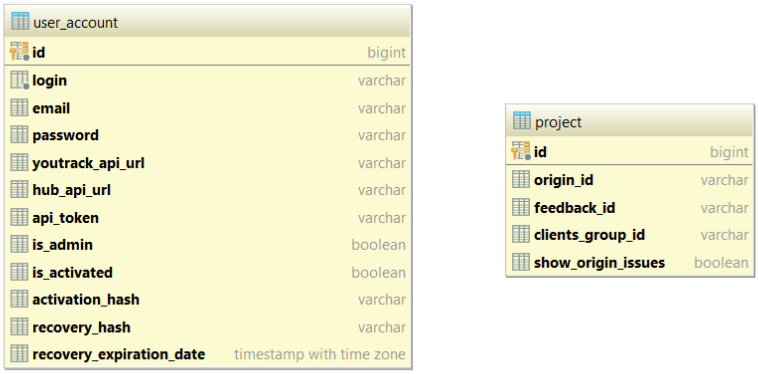
Popis modelu:

- Logika použití entit **User**, **Group**, **Project** a **ProjectTeam** je podobná jako v YouTrack (viz *2.3.1 Hub REST API - Model*), navíc každý původní projekt obsahuje odkazy na feedback-projekt, uživatelskou skupinu, seznam vývojářů a seznam zákazníků na projektu.

- **IssueAccessType** - typ přístupu k issue, obsahuje **type** (origin, new nebo feedback, viz *3.4.4 - Typy přístupu k issues*) a **link** - odkaz na příslušné issue (pro typy origin a feedback).
- **Link** - odkaz na jiné issue.

3.7.2 Databázový model

Následující diagram reprezentuje databázový model na serverové straně aplikace. Obsahuje jenom dvě tabulky, protože serverová část se používá pouze pro správu uživatelských účtů a informací o projektech.



Obrázek 3.7: Databázový model aplikace

Součástí tabulky **user_account** jsou:

- **login, email, password** - základní údaje,
- **youtrack_api_url, hub_api_url** - odkazy na příslušné systémy YouTrack a Hub,
- **api_token** - permanentní API token vytvořený přes nastavení v YouTrack,
- **is_admin, is_activated** - logické hodnoty, určující stav uživatele,
- **activation_hash** - aktivační hash (viz *3.6.2 Společná část - Aktivovat účet*),
- **recovery_password_hash, recovery_expiration_date** - hash na obnovení hesla a její expirační datum (viz *3.6.2 Společná část - Obnovit zapomenuté heslo*).

Součástí tabulky **project** jsou:

- **origin_id** - ID původního projektu,
- **feedback_id** - ID příslušného feedback-projektu,
- **clients_group_id** - ID příslušné uživatelské skupiny pro zákazníky,
- **show_origin_issues** - logická hodnota, která určuje zda zákazníci vidí issues z původního projektu (viz *3.6.1 Administrátorská část - Přepnout viditelnost issues z původního projektu*).

Kapitola 4

Implementace

4.1 Klientská část

Tato část popisuje technologie použité k implementaci klientské strany aplikace.

4.1.1 Volba technologie

Pro implementaci klientské strany aplikace je potřeba zvolit vhodnou moderní frontend-technologii, odpovídající těmto požadavkům:

- jednoduchost se naučit,
- dobrá dokumentace a podpora,
- hodně hotových řešení (ušetří čas).

Porovnání třech moderních frontend-technologií:

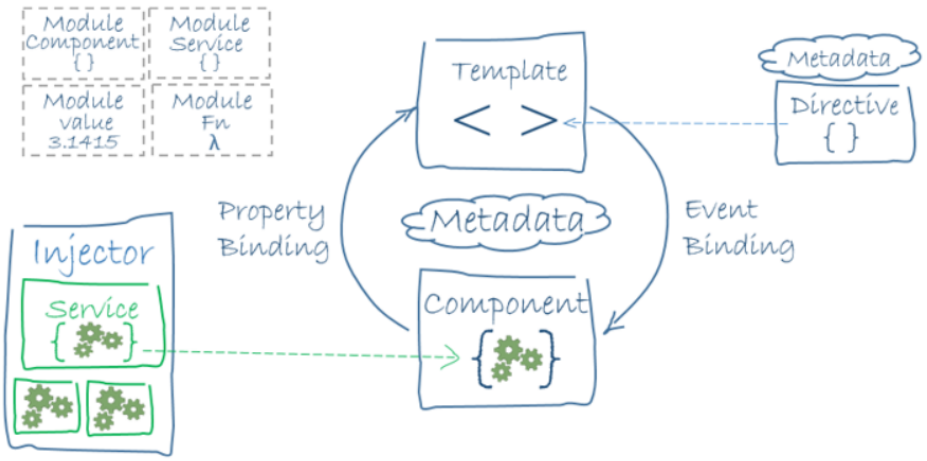
- **React:** JavaScript knihovna, která poprvé byla vydána jako open source v roce 2013 pod BSD licencí. Lze k ní najít spoustu zdrojů, výukových programů a knihoven/komponent. Flexibilní, hodí se pro microservice structure. Používá se obrovskými firmami, např. Facebook nebo Instagram. Poslední update (verze 16.3.2) byl 16. dubna 2018.
- **Angular:** TypeScript framework, který byl vydán v roce 2016, je kompletní přepsání frameworku AngularJS od stejného týmu. Má velmi kvalitní dokumentaci a podporu od Google. Obsahuje hodně hotových řešení. Podporuje dvoucestnou synchronizaci dat. Používá se obrovskými firmami, např. Google a Microsoft. Poslední update (verze 6.0) byl 3. května 2018.
- **Vue:** JavaScript framework, který poprvé byl vydán v roce 2014 jako open source Framework pod MIT licencí. Vue je relativně nový framework, který získává spoustu nových vývojářů. Pracuje na MVVM vzoru, má velmi jednoduché API, je minimalistický. Vue.js je inspirováno frameworky Angular.js, React.js, Knockout.js a Rivets.js a pracuje na dvoucestné synchronizaci dat. Používá se např. v Adobe a GitHub. Poslední update (verze 2.5.16) byl 14. března 2018.

Na základě výše uvedeného porovnání byl zvolen framework **Angular**, z důvodu, že je to kompletní framework s velkým množstvím hotových řešení, kvalitní dokumentací a podporou, má velké rozšíření.

4.1.2 Angular 5

Používá se poslední verze (v době výběru) - **Angular 5**.

Struktura



Obrázek 4.1: Architektura Angular-aplikaci

Výše uvedený obrázek z oficiální dokumentace Angular [9] popisuje strukturu programu napsáného s použitím frameworku Angular.

Základní elementy jsou:

- **Components** - zodpovídají za prezentační část aplikace. Komponenty dostávají vstupní data, na jejichž základě vrací zpracovanou šablonu (**Template**), tj. konkrétní HTML-stránku. Používá se 'two-way binding', tj. na stránce se zobrazí data z modelu a uživatelské akce (např. zadání údajů nebo stisknutí tlačítka) změní model. Závislosti na services jsou injektovány v parametrech konstruktoru podle principu Dependency injection.
- **Services** - servisní vrstva, zodpovídá za business-logiku aplikace. Pro posílání HTTP dotazů na serverovou část se používá HttpClient.

HttpClient

HttpClient je součástí frameworku, určen k posílání HTTP dotazů. V této aplikaci posílá dotazy na YouTrack/Hub REST API a vlastní serverovou část. Je potřeba injektovat HttpClient do servisu.

Ukázka kódu (načtení issues z projektu):

```
...
  private url = 'http://localhost:80/youtrack/rest/issue';
  private max = 100;

  constructor(private http: HttpClient) {
  }

 getIssues(project: string): Observable<Issue[]> {
    return this.http.get<IssueJSON[]>(`${this.url}/byproject/
    ${project}?max=${this.max}`, {headers: HttpService.headers})
      .pipe(
        map(issues => this.issueConverter.toIssueArray(issues))
      );
  }
...
```

Pomocí metody *http.get* aplikace posílá HTTP dotaz na YouTrack REST API. Metoda *getIssues()* vrátí objekt **Observable**, ve kterém se objeví pole issues, po získání odpovědi z YouTrack.

4.1.3 Použité knihovny

V klientské části aplikace se používají následující knihovny:

- **Angular Material** - oficiální knihovna komponent ve stylu 'Material Design'(oficiální designový jazyk od Google).
- **Angular Bootstrap Material Design** - knihovna komponentů ve stylu 'Material Design', má složitější šablony.
- **angular-paginator** - knihovna pro stránkování seznamů.
- **angular2-markdown** - podpora formátu markdown u komentářů a popisů issue (požadavek **FR12**).

4.2 Serverová část

4.2.1 Java Spring Framework

Serverová část je obyčejný web-server, který nemá žádné specifické požadavky, proto pro implementaci této části byl zvolen **Java Spring Framework** jako jedna z nejpopulárnějších technologii a framework, se kterým má autor nejvíc zkušeností.

Struktura

Serverová část aplikace je rozdělena na tři úrovně (viz *3.5 Struktura aplikace*):

- **REST** - obsahuje controllery, které komunikují s klientskou částí aplikace. Ukázka kódu (UserController):

```
...
@PostMapping("/users/register")
public ResponseEntity<Message> register(@RequestBody UserTo userTo) {
    RegistrationMessage response = registrationService.register
    (userConverter.toAdminInfo(userTo));
    switch (response) {
        case USER_ALREADY_EXISTS:
            return new ResponseEntity<>(new Message(USER_ALREADY_EXISTS),
            HttpStatus.CONFLICT);
        case USER_IS_NOT_ACTIVATED:
            return new ResponseEntity<>(
            new Message(USER_IS_NOT_ACTIVATED), HttpStatus.CONFLICT);
        case EMAIL_COULD_NOT_BE_SEND:
            return new ResponseEntity<>(new Message(COULD_NOT_BE_SEND),
            HttpStatus.INTERNAL_SERVER_ERROR);
        default:
        case REGISTRATION_IS_OK:
            return new ResponseEntity<>(HttpStatus.CREATED);
    }
}
...
```

  Tento controller přijímá dotazy na registraci uživatele. Controller pošle uživatelské údaje do service, odkud se pak vrátí RegistrationMessage, tj zpráva, která určuje výsledek dotazu. Na základě teto zprávy controller vytvoří odpověď ResponseEntity, kterou odešle zpátky na klientskou stranu aplikace.

- **Services** - servisní vrstva, zodpovídá za business-logiku aplikace.
- **DAO** - komunikuje s databází pomocí speciální knihovny (viz *4.2.2 MyBatis*).

Spring Boot Application

Serverová část aplikace je implementována pomocí Spring Boot, protože je to rychlý a jednoduchý způsob jak postavit web-server. Požaduje minimum konfigurace.

Service pro posílání emailů

Aplikace potřebuje posílat emaily uživatelům (např. aktivační link). Pro tyto účely se používá rozhraní **EmailService**.

Dále je ukázka jeho implementace:

```
@Service
public class EmailServiceImpl implements EmailService {

    @Autowired
    private JavaMailSender emailSender;

    private final String from = "cfsystem.bachelor@gmail.com";


    public boolean sendEmail(String to, String subject, String text) {
        SimpleMailMessage message = new SimpleMailMessage();
        message.setFrom(from);
        message.setTo(to);
        message.setSubject(subject);
        message.setText(text);

        try {
            emailSender.send(message);
        } catch (MailException exc) {
            return false;
        }
        return true;
    }
}
```

Implementace je založena na použití třídy **JavaMailSender**, která je součástí Spring Framework. Je potřeba do souboru **application.properties** přidat nastavení SMTP-serveru (používá se SMTP Gmail Server) a Google accountu, ze kterého se budou odesílat maily. Tento SMTP Server se používá pouze pro účely bakalářské práce.

4.2.2 MyBatis

Cílem bylo zvolení Java knihovny pro práci s databází, která je 'lightweight' a se kterou má autor nejvíc zkušeností. K porovnání byly vybrány dvě knihovny: Hibernate a MyBatis.

Porovnání MyBatis a Hibernate

	MyBatis	Hibernate
+	1) Jednoduchost a rychlejší vývoj. 2) Je dobrá pro psaní čistých SQL.	1) Databázový agnostický jazyk. 2) Automatická generace SQL, podporované koncepce OOP. 3) Vysoká škálovatelnost.
-	1) SQL příkazy mohou být svázány s konkrétním dodavatelem databáze. 2) Nepodporuje vestavěné ukládání do mezipaměti, stránkovací mechanismus.	1) Vysoká složitost. 2) Změny schématu mohou způsobit problémy.

Tabulka 4.1: Porovnání MyBatis a Hibernate

Na základě tohoto porovnání byla zvolena knihovna **MyBatis**, díky své jednoduchosti.

Použití

Pro práci s konkrétní entitou je potřeba vytvořit:

- DAO-rozhraní:

```
@Mapper
public interface UserDao {
...
    User getById(Long id);

    User getByLogin(String login);
...
}
```

- XML-soubor, ve kterém jsou definované funkce z rozhraní ve tvaru přímých SQL-dotazů:

```
...
<select id="getById" resultMap="userResult">
    SELECT *
    FROM t_user
    WHERE t_user.id =  #{id}
</select>


<select id="getByLogin" resultMap="userResult">
    SELECT *
    FROM t_user
    WHERE t_user.login = #{login}
</select>
...
```

4.2.3 Ostatní knihovny

- **Project Lombok** - obsahuje užitečné anotace, např. v následujícím kódu anotace @Data umožňuje používat getter a setter metody na všechny argumenty:

```
...
@Data
public class User implements Serializable {

    private Long id;
    private String login;
    private String email;
...
```

- **Apache Commons Lang 3** - používá se pro generaci hashů pro aktivaci účtu a obnovení hesla.

4.3 Databáze

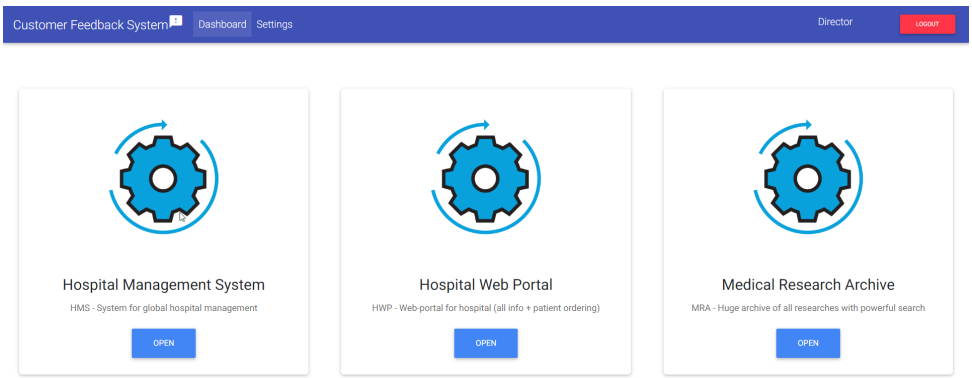
Podle vlastních zkušeností autora byla zvolena databáze **PostgreSQL**.

4.4 Ukázka UI

Dále jsou screenshoty jednotlivých stránek aplikace.

4.4.1 Zákaznická část

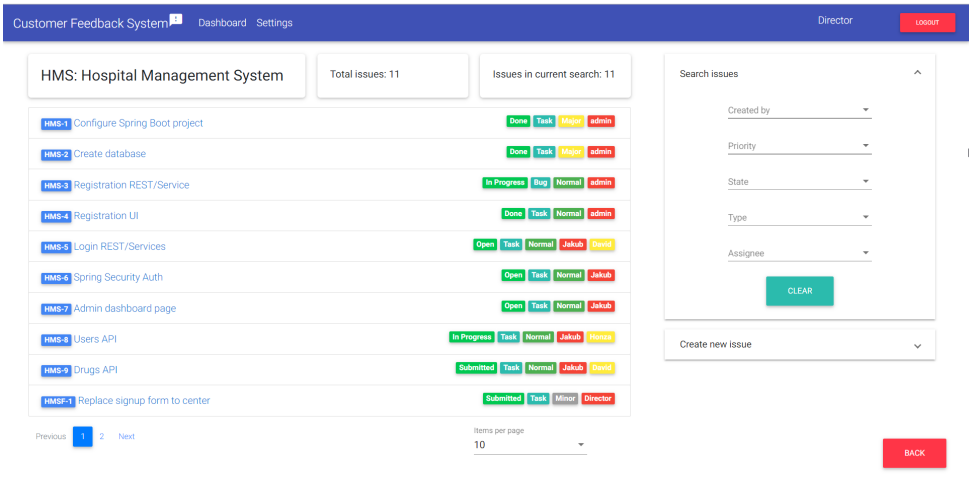
Dashboard



Obrázek 4.2: Dashboard

Na dashboardu zákazník vidí všechny projekty, ke kterým má přístup. Muže otevřít libovolný projekt stisknutím tlačítka 'Open'.

Stránka projektu

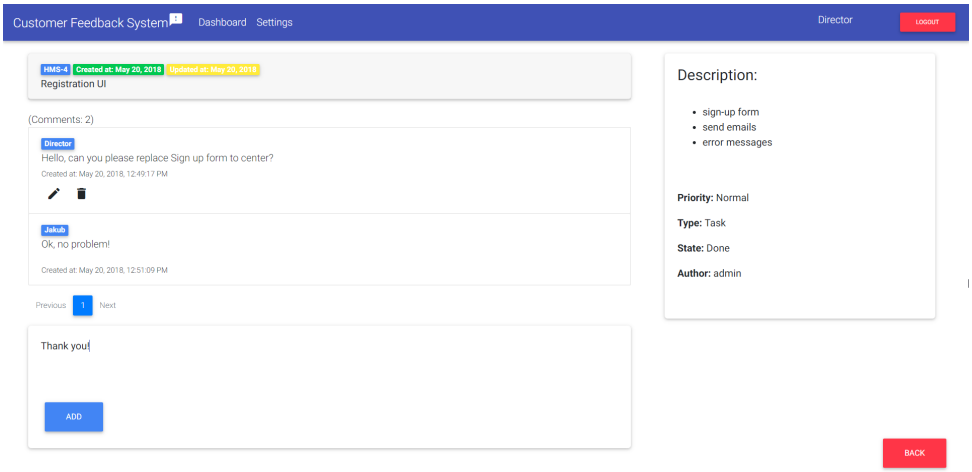


Obrázek 4.3: Stránka projektu

Vlevo je seznam issues z původního projektu a feeedback-projektu (typy **origin** a **new** viz *3.4.4 Typy přístupu k issue*), kde barevné labely reprezentuji hodnoty atributů issue: stav, typ, prioritu, autora a assignee. Administrátor může přepnout viditelnost issues (viz *3.6.1 Administrátorská část - Přepnout viditelnost issues z původního projektu*).

Vpravo jsou filtry pro hledání issues a formulář na přidání nového issue (je na obrázku skryt - otevři se kliknutím). Kliknutím na issue ze seznamu zákazník se přesměruje na stránku zvoleného issue.

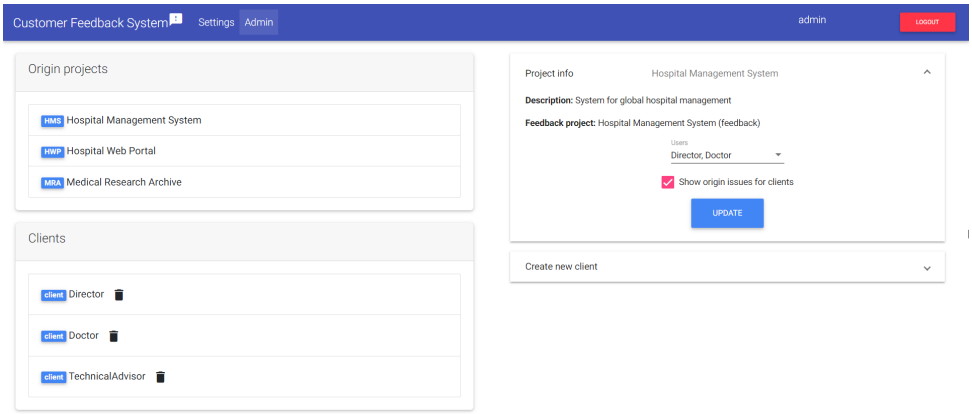
Stránka issue



Obrázek 4.4: Stránka issue

Vlevo jsou informace o issue (id, summary, datum vytvoření, datum obnovení) a komentáře k issue. Vpravo jsou další informace o issue: popis, priorita, stav, typ, autor a případně assignee, odkazy na jiné issues, tagy.

4.4.2 Administrátorská část



Obrázek 4.5: Stránka administrátoru

Vlevo je seznam původních projektů z YouTrack a seznam zákazníků, kteří byli v této aplikaci vytvořeni.

Vpravo na obrázku po kliknutí na projekt ze seznamu se zobrazí informace o původním projektu: popis, název feedback-projektu, rozbalovací seznam pro přidání/zrušení zákazníkovi přístupu k projektu a checkbox 'Show origin issues'. Kliknutím na zákazníka ze seznamu se na stejném místě zobrazí informace o něm. Vpravo dolu je také formulář na přidání nového zákazníka (je na obrázku skryt - otevři se kliknutím).

Kapitola 5

Závěr

5.1 Zhodnocení

V rámci této bakalářské práce byla navrhnuta a implementována webová aplikace, která usnadní komunikaci mezi vývojáři (uživatele YouTrack) a zákazníky. Využití REST API pro zápis a načtení dat přímo z YouTrack zaručuje, že data budou vždy aktuální. Aplikace má jednoduché a uživatelsky přívětivé grafické rozhraní. Všechny požadavky byly splněny.

5.2 Možné budoucí rozšíření aplikace

5.2.1 Použití pro jiné issue-trackery

Aplikaci lze rozšířit na použití dalších trackerů kromě YouTrack. Je očekávané, že jiné trackery budou mít podobnou datovou strukturu, proto Angular-komponenty a šablony, které tvoří prezentační vrstvu klientské části aplikace mohou být znovupoužitelné. Servisní vrstva klientské části také může být rozšířena pro komunikaci s jiným trackerem, jenom bude potřeba přidat speciální konvertory, které budou převádět data z formátu, ve kterém je poskytuje API trackeru do interního formátu aplikace.

5.2.2 Použití pro obecné uživatele

Další možnost rozšíření je použití pro obecné uživatele. Uživatel současné verze aplikace je reálný zákazník firmy, nemusí být koncovým uživatelem produktu, který pro něho firma vyvíjí. V případě rozšíření pro obecné uživatele bude potřeba vyřešit, kam ukládat uživatelské účty. V současné verzi aplikace má každý uživatel účet v YouTrack a záznam v databázi aplikace, a těch zákazníků není až tak moc. Pokud by bylo uživatelů aplikace o mnoho více, může to způsobit problém, protože YouTrack má omezený počet uživatelů a maximální počet závisí na tarifním plánu.

Literatura

[1] YouTrack REST API Reference [online]. [cit. 2018-01-16]. Dostupné z: https://www.jetbrains.com/help/youtrack/standalone/YouTrack-REST-API-Reference.html

[2] Hub REST API Reference [online]. [cit. 2018-05-17]. Dostupné z: https://www.jetbrains.com/help/hub/HUB-REST-API.html

[3] KALIN, Martin. Java Web services: up and running. Second edition. Sebastopol, California: O'Reilly, 2013. ISBN 1449365116.

[4] RICHARDSON, Leonard a Sam. RUBY. RESTful web services. Farnham: O'Reilly, c2007. ISBN 0596529260.

[5] SANDOVAL, José. RESTful Java web services: master core REST concepts and create RESTful web services in Java. Birmingham, U.K.: Packt Pub., 2009. From technologies to solutions.

[6] Top 10 frameworků pro moderní frontend [online]. [cit. 2018-05-20]. Dostupné z: https://www.wdt.cz/inovace/top-10-frameworku-pro-moderni-frontend-a589ddee72867455fd9c17a2d

[7] Angular vs. React vs. Vue: A 2018 Comparison [online]. [cit. 2018-05-17]. Dostupné z: http://www.cuelogic.com/blog/angular-vs-react-vs-vue-a-2018-comparison/

[8] Angular vs. React vs. Vue: A 2017 comparison [online]. [cit. 2018-05-17]. Dostupné z: https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176

[9] Angular Framework Reference. [online]. [cit. 2018-05-17]. Dostupné z: https://angular.io/

[10] Angular Material [online]. [cit. 2018-05-17]. Dostupné z: https://material.angular.io/

[11] Angular Bootstrap with Material Design [online]. [cit. 2018-05-17]. Dostupné z: https://mdbootstrap.com/angular/

[12] Spring Framework Reference [online]. [cit. 2018-05-17]. Dostupné z: https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/index.html

[13] Hibernate vs MyBatis [online]. [cit. 2018-05-17]. Dostupné z: https://rajivrnair.github.io/hibernate-vs-mybatis

[14] MyBatis Reference [online]. [cit. 2018-05-17]. Dostupné z: http://www.mybatis.org/mybatis-3/

Příloha A

Zkratky

API	Application programming interface
BSD	Berkeley Software Distribution
CORS	Cross-Origin Resource Sharing
CRUD	Create, Read, Update, Delete
DAO	Data Access Object
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
MVVM	Model-View-ViewModel
OOP	Object Oriented Programming
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locator
XML	eXtensible Markup Language

Příloha B

Nasazení aplikace

B.1 Softwarové požadavky

Pro běh aplikace je potřeba mít:

- Java (verze 8 nebo vyšší),
- Maven (verze 3.5.2 nebo vyšší),
- PostgreSQL server,
- Node.js (verze 5.6.2 nebo vyšší).

B.2 Nasazení aplikace

B.2.1 Databáze

Nejprve je potřeba vytvořit databázi a spustit skript **BachelorProject/tables.sql** pro vytvoření tabulek. Defaultní nastavení databáze jsou:

```
Port: 5432
Database name: BP
Username: postgres
Password: password
```

Tyto parametry lze změnit v souboru
BachelorProject/backend/src/main/resources/application.properties

B.2.2 Serverová část

Pro rozběhání serverové části je nutné provést následující posloupnost příkazů přes příkazový řádek:

1. cd backend
2. mvn clean install
3. java -jar target/bachelor_project-0.0.1-SNAPSHOT.jar

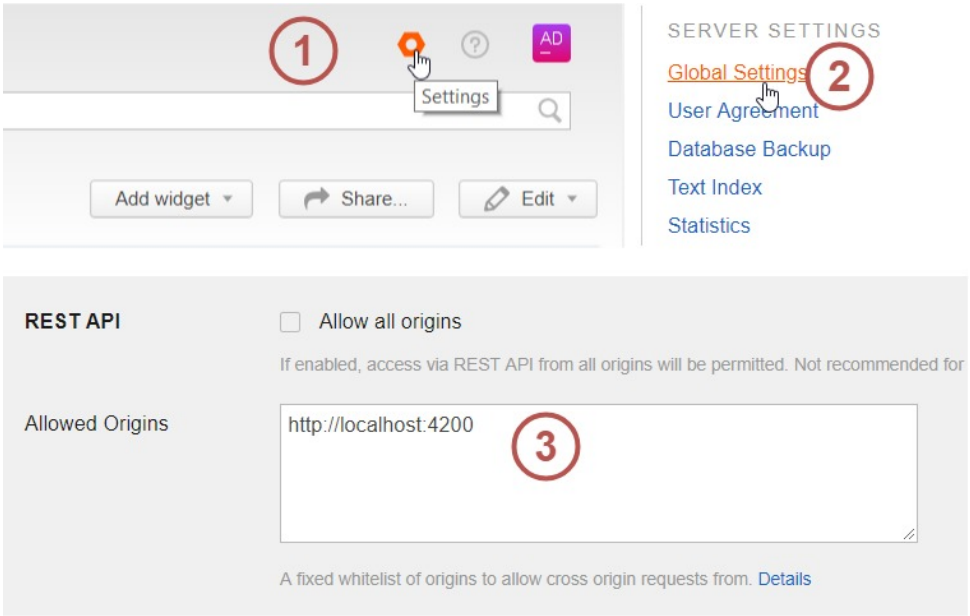
B.2.3 Klientská část

Pro rozběhání klientské části je nutné provést následující posloupnost příkazů přes příkazový řádek:

1. cd frontend
2. npm install -g @angular/cli
3. npm install
4. ng serve

B.3 Nastavení YouTrack

Pro povolení aplikaci posílat dotazy na YouTrack/Hub REST API je nutné přidat adresu aplikaci do CORS-povolených adres v YouTrack:



Obrázek B.1: CORS nastavení