Flatten the curve

# Základní popis:

* Náš projekt bude backendem znalostní databáze o koronaviru Covid-19. Bude obsahovat popis nemoci, příznaky, léčby, vědecké články, potenciální léčbě, kde se dát otestovat...
* Hlavním účelem aplikace je šířit pravdivé informace o nemoci Covid-19 mezi obyčejné lidi. Má být jednoduchá, intuitivní, lehká na použití a hlavně škálovatelná a co nejlépe optimalizovaná. Naše databáze bude obsahovat všechny potřebné informace, aby se dále mohlo předcházet šířením misinformací.

# Pro koho je projekt určen?

* Projekt je určen pro širokou veřejnost (guests), kde články jsou upravovány a přidávány našimi editory (editors). Účty jsou vytvářeny a spravovány administrátory (admins)

# Role

**USERS** se tedy dělí na **GUESTS**, **ADMINS** a **EDITORS**.

* Guests mají přístup ke všem informacím, nemají ale vytvořený účet a nemohou vytvářet/mazat/upravovat příspěvky. Nejsou tedy v naší aplikaci nijak definováni.
* Admins mají nejen přístup ke všem informacím, mohou ale vytvářet, mazat a upravovat články a vytvářet a mazat uživatelské účty. Do aplikace se musí přihlašovat.
* Editors poté mají přístup ke všem informacím, mohou editovat, vytvářet a mazat články, ale nemají právo jakkoli upravovat/vytvářet/mazat uživatelské účty. Do aplikace se musí přihlašovat.

# Entity

* Uživatelský účet v sobě obsahuje křestní jméno a příjmení, heslo a uživatelské jméno. Dále se každému uživateli přiřadí role (admin/editor). Hosté uživatelský účet nemají.
* Hlavní obsah aplikace se dělí na články, symptomy a léčby. To vše spadá pod příspěvky (vizte class diagram), které obsahují základní informace - nadpis a datum vytvoření. Článek navíc obsahuje tělo článku. Symptom obsahuje popis a procento výskytu u pacientů. Léčba obsahuje popis, efektivitu, používané léky a dobu trvání léčby.
* Bude také obsahovat takzvaný Tool, který bude zveřejňovat statistiky, články třetích stran atd.

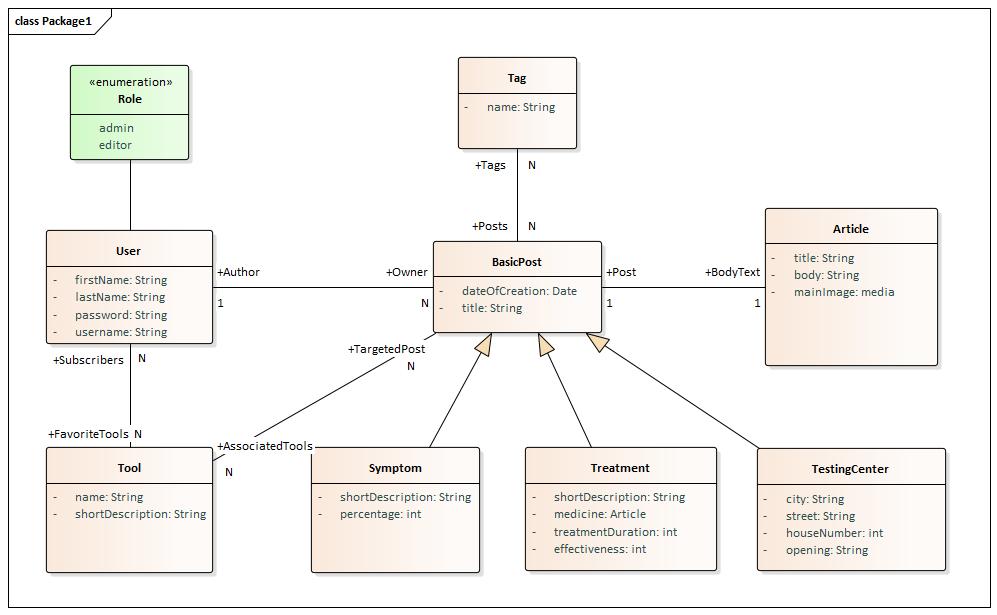
# Omezení

* Uživatelé nebudou moci do aplikace přispívat a ptát se na dotazy ohledně onemocnění.
* Aplikace nebude náhradou za vyšetření od lékaře, slouží pouze jako informace! Pokud má uživatel podezření, že je nakažen a pociťuje aplikací popsané symptomy, měl by jít na testy a ne aplikaci brát jako diagnózu.

# Implementace

Aplikaci píšeme v jazyce Java.

Využívat budeme SQL databázi (JPA), framework Springboot a případně JWT pro zabezpečení. Rádi bychom využívali API třetích stran pro získávání dat o Covidu-19 z internetu. Vše poběží na operačním systému Windows. Do budoucna se k projektu bude moci přidělat webový frontend.



# Návod k instalaci

(Předpokládáme, že máme nainstalovaný Maven a prostředí podporující Javu 8)

* Projekt naklonujeme z gitu:
* git clone [git@gitlab.fel.cvut.cz:B201\_B6B36EAR/vevarluk.git](mailto:git@gitlab.fel.cvut.cz:B201_B6B36EAR/vevarluk.git)
* Přesuneme se do adresáře:
* mkdir vevarluk
* Spustíme příkazy
* mvn install
* mvn clean package

Vygenerovaný soubor, se nachází v adresáři target.

Databáze se nachází na osobním serveru. Není nutné ji tedy zprovoznit pro běh aplikace.

* Databázový server je: 89.24.243.161:5432/covid
* Přihlašovací jméno je: admin
* Heslo pro přihlášení je: root

Pokud se chceme na databázi podívat v grafickém prostředí, je to možné na adrese <https://adminer.devdom.org/>

Stačí pak ještě zadat do kolonky Systém – PostgreSQL a do kolonky Server – localhost.

Postman testy jsou přiložené zvlášť v gitlabu.

# Co jsme si vyzkoušeli

Většina získaných zkušeností se točí kolem frameworku **Spring** a Spring security. S nimi jsme se naučili pracovat nejvíce. Výhodou Springu oproti předešlým technologiím, které jsme použili je možnost tvorby Beanů, ulehčující celkovou funkcionalitu projektu a jeho rychlejší tvorbu. Také jsme měli možnost využít spoustu balíčků a knihoven, které hodně práce řešily za nás.

Spring si také automaticky vytváří tabulky v databázi podle vytvořeného modelu, což se nám také velmi líbilo.

Mezi další významné nové znalosti patří:

* Generování a formátování **JSON**u
* Práce s **Postman**em
* Využili jsme virtuální embedded databázi pro testování

# Zajímavosti

* Pokus o vytvoření škálovatelné implementace Tools. Program může obsahovat řadu nástrojů, které například tahají data z externích API. Stačí vytvořit jejich příslušnou implementaci do package service.tools a potom vytvořit do databáze záznam s jejich jménem.
* Škálovatelné typy BasicPost. Program se dá rozšířit o řadu dalších položek k ukládání a automaticky si vytvoří tabulku do databáze.
* Při vracení JSONu v Controllerech zabraňujeme nekonečné rekurzi (způsobenou vazbami mezi entitami) pomocí pouhého vypsání ID dané entity z databáze. Klient poté může provést Get dotaz na dané ID.

# Nečekané problémy

* Nešlo vytvořit pouze jedno Dao a jeden Service pro BasicPost (rodiče všech zděděných položek). Spring neumí jednodušše rozlišit typ daného potomka při práci s databází. Museli jsme tedy vytvořit Dao a Service třídy pro každého potomka třídy BasicPost.