

01. Výber témy a popis aplikácie

a) Vyberte si vlastnú (VHODNÚ a primerane náročnú) tému aplikácie.

Téma je aplikácia na organizovanie konferencií alebo iných akcií.

b) Vytvorte stručný popis tejto témy a funkcionálit výslednej aplikácie v súbore README.txt.

README nebude bude toto. Aplikácia umožní sa registrovať, prihlásiť sa do svojho účtu. Vybrať si konferenciu a ktorej prezentácie sa zúčastníš. Admin bude vidieť kto sa kde prihlásil.

02. Databázový návrh

V databázovom návrhu treba zabezpečiť, aby databáza obsahovala:

a) minimálne 5 hlavných tabuliek, ktoré budú reprezentovať rôzne entity.

Hlavne:

1. Conferences 2. Stages 3. Presentations 4. Users 5. Sponsors 6. Notes

Pilot:

1. PresentationsHasParticipants 2. PresentationsHasSpeakers
3. sponsorsHasConferences

b) minimálne 15 atribútov naprieč všetkými tabuľkami (PK, FK a časové pečiatky ako created_at, updated_at sa nerátajú). Z toho treba použiť 5 rozdielnych dátových typov (napr. INT a LONG sa počítajú ako jeden typ, ale VARCHAR a TEXT sa počítajú ako dva rôzne typy),

1. Conferences = 4; 2. Stages = 2; 3. Presentations = 7; 4. Users = 10;
5. Sponsors = 4; 6. Notes = 2

Pilot:

1. PresentationsHasParticipants = 0; 2. PresentationsHasSpeakers = 1

Hlavne $4+2+7+10+4+2 = 29$

Pilot $0+1 \quad 1$

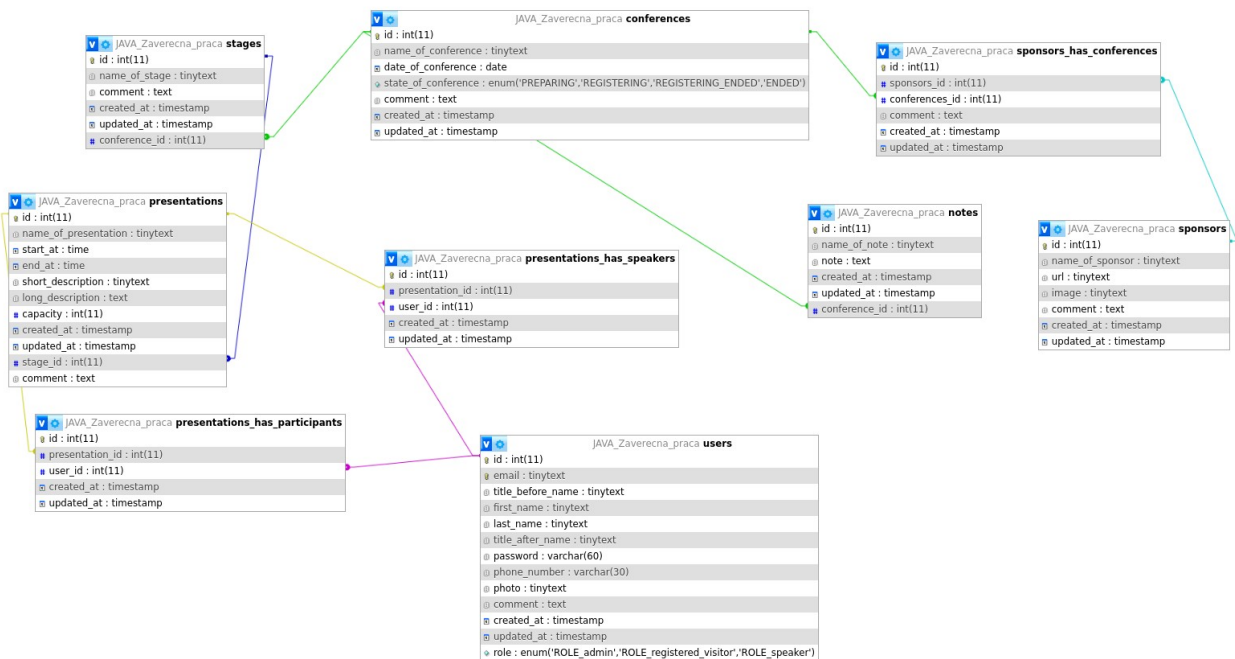
Dokopy = 30

Typy udajov 1. tinyText, 2. date, 3. enum, 4. text, 5. timestamp, 6. int,
7. varchar

c) minimálne 1 vzťah M:N,

Až 3 máme. 1. PresentationsHasParticipants 2. PresentationsHasSpeakers
3. sponsorsHasConferences

Súčasťou odovzdania musí byť **export databázy v SQL formáte a E-R diagram** ako PNG. Hodnotí sa jej správnosť (atribúty + údajové typy) a vyplnené DEMO údaje (dostatočný počet nechám na Vás, nech to po spustení nejako vyzerá).



03. Java entity a ORM

Vytvorte Java entity, ktoré budú mapované na databázové tabuľky pomocou JPA (Hibernate). Dbajte na správne mapovanie vzťahov medzi entitami. Správne mapovanie môžete vyskúšať cez REST API (nie je to povinné). Dajte si pozor na stratu dát pri manipulácii s dátami.

Mapovanie je urobené na testovanie si môžete nainštalovať do postmana restPríkazy v súbore "Postman.json".

V config/secutity treba odkomentovať `csrf().disable()` a `requestMatchers("/api/**").permitAll()`

04. Validácia

Implementujte validáciu vstupných dát vo všetkých formulároch aplikácie. Minimálne musíte mať 2 formuláre v rámci aplikácie.

Minimálne 2 formuláre – Pre verejnosť Registrácia, Login.

Pre Admina : Konferencie – Vytvoriť, Aktualizovať

Scény – Vytvoriť, Aktualizovať

Prezentácie – Vytvoriť, Aktualizovať

Používateľ – Vytvoriť

Sponzory - Vytvoriť, Aktualizovať

Hodnotia sa správne nastavené validačné kritéria v spolupráci s databázou (napr. ak má mať meno max. 64 znakov v DB, tak ho obmedzím na veľkosti zhora touto veľkosťou, ale pozor aj zdola, nikto sa nevolá jednoznakovo ...). Výsledky validácie by

mali byť spracované v Controlleri a chybové správy by mali byť korektne zobrazené v Thymeleaf šablóne.

Validácie su nastavené, spodné hodnoty som nenastavoval.

05. Bezpečnosť

Pridajte autentifikáciu a autorizáciu. Aplikácia musí obsahovať prihlásenie, registráciu, odhlásenie a minimálne 3 rôzne používateľské role, pričom jedna z nich musí byť ADMIN, ktorý dohliada na všetko a dokáže meniť práva ostatným používateľom, prípadne niečo schvaľovať, upravovať, mazať,... .

Registrácia, Prihlásenie a Logout je. V databáze c Users commenty sú heslá.

Role sú: ROLE_admin, ROLE_registered_visitor, ROLE_speaker.

Zabezpečte prístup k rôznym častiam aplikácie na základe rolí. Rovnako zabezpečte zobrazenie Not Found stránky, keď používateľ zadá stránku, ktorá neexistuje. Každý používateľ musí mať možnosť zmeniť svoje informácie a heslo cez používateľské rozhranie. Zmena hesla musí byť oddelená od úpravy ostatných osobných údajov. Zabezpečte, aby bola implementovaná správna validácia pri zmene hesla (napr. minimálny počet znakov, kontrola zhody)

Jedine /MVC/amin** je dostupné iba pre admina. Používatelia nemajú možnosť meniť svoje údaje.

06. Zobrazenie údajov na stránke

Implementujte Thymeleaf šablóny, ktoré zobrazia dáta z databázy.

Podmienkou je zobrazit' všetky dáta, prípadne využi dáta (ako napr. heslo pri prihlásení a podobne). Všetko by malo mať logickú štruktúru, vlastný dizajn nie je potrebný, pokojne môžete využiť Bootstrap.

Používateľ s príslušnými právami musí mať možnosť vytvárať, upravovať a mazať dáta v aplikácii, pričom tieto operácie budú dostupné iba pre používateľov s určitými rolami. V Thymeleaf šablónach zobrazit' tieto možnosti len pre autorizovaných používateľov na základe ich prístupových práv.

V adminovy je pri väčšine dát create, update, delete.

07. Bonus: Stránkovanie (10 bodov) a obrázky (10 bodov)

Stránkovanie je urobené pre admin/presentations.