

Špecifikácie – FotGetMeNot

Jedná sa o aplikáciu, ktorá umožní užívateľovi po prihlásení zapisovať si na jednom mieste pozreté / prečítané knihy, filmy a seriály. Má možnosť si zaznačiť kde skončil v pozeraní / čítaní a či je daná séria ukončená, neukončená alebo ešte nie je zverejnená. K jednotlivým položkám si vie pridávať tagy podľa, ktorých vie svoje položky triediť. Zároveň vie pridať hodnotenie a komentár, toto môže byť buď zverejnené verejne alebo len súkromne. Tie vie vidieť aj neprihlásený používateľ. V rámci aplikácie existujú aj administrátori, ktorý majú povolenie vymazávať zverejnené príspevky alebo aj celé účty.

Cieľová skupina používateľov sú nadšenci kníh a filmografie.

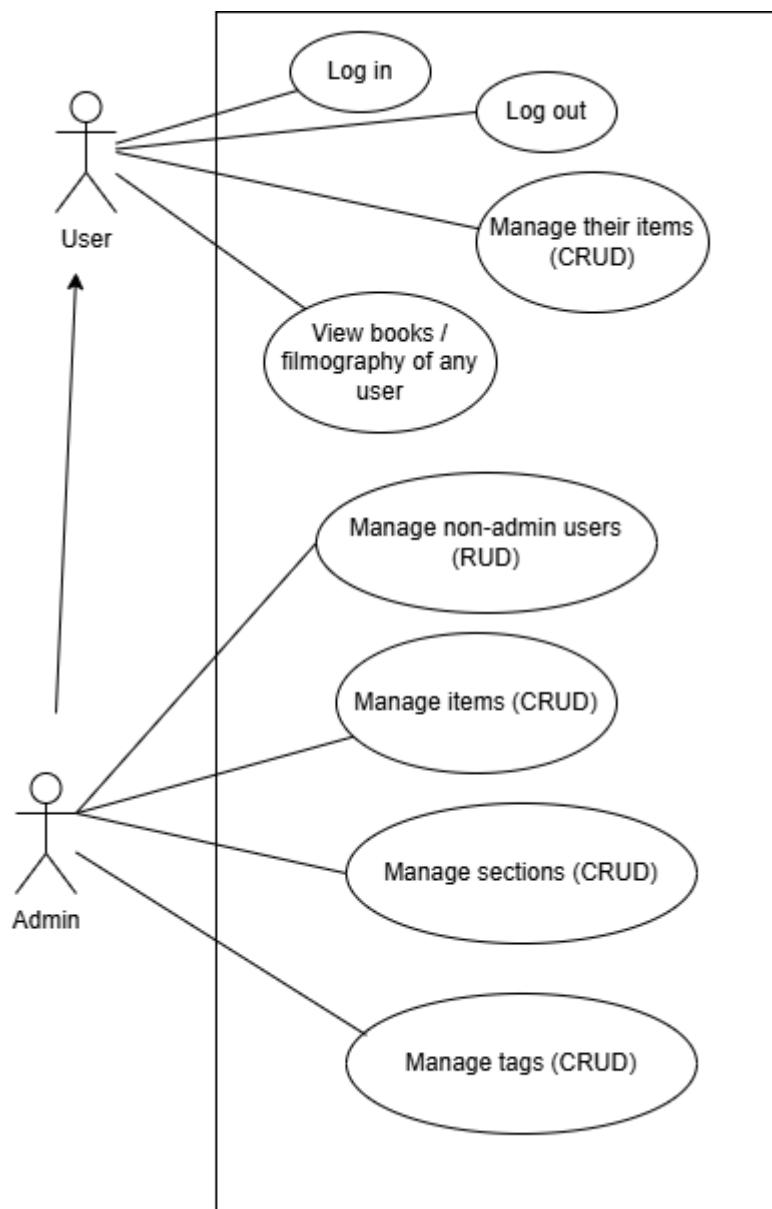
Používateľské požiadavky

Role

- **Neprihlásený užívateľ** - má prístup ku komentárom a hodnoteniam zverejnených položiek
- **Prihlásený užívateľ** – vie si pridávať pod svoje konto položky (knihy, filmy / seriály)
- **Admin** - vymazať zaregistrovaných používateľov, prípadne len konkrétne položky, ak obsahujú nevhodné komentáre alebo sa užívatelia nesprávajú podľa zásad.

Use Case Diagram

- CRUD = Create, Read, Update, Delete
- „Log in“, „Log out“ a „Register“ sú v diagrame na doplnenie úplnosti



Dátový model

Nasledujúci koncepčný dátový model obsahuje entity, atribúty a vzťahy.

Entity a atribúty

Atribúty, ktoré sú jednoznačné nie sú popísané v nasledujúcej časti ale sú len v diagrame.

Užívateľ - User

- Užívateľ je identifikovaný cez jeho priradené id.
- Obmedzenia a pravidlá:
 - užívateľ má vždy jeden neadministratívny účet a nanajvýš jeden administratívny účet.
 - administratívny účet je tvorený cez databázu, nemôže byť vytvorený cez oficiálnu registráciu aplikácie

Účet - Account

- Účet na prihlásenie sa do aplikácie. Prihlasovacie meno - username je unikátne.
- Obmedzenia a pravidlá:
 - každý účet má práve len jedného užívateľa
 - užívateľ je priradený k účtu cez user_id
- Atribúty:
 - password: uložené ako hash hodnota
 - is_admin: vráti true, ak užívateľ je administrátor, inak vráti false

Item

- základné info o knihe, filmu / seriáli - užívateľ vie vybrať z existujúcich na FE - ak zmení nejaký základný údaj o zázname a dané informácie nezodpovedajú existujúcemu záznamu, vytvorí sa nový
- Obmedzenia a pravidlá:
 - existuje len jeden záznam s danou kombináciou údajov
- Atribúty:
 - type: či sa jedná o knihu alebo film / seriál

ItemAccount

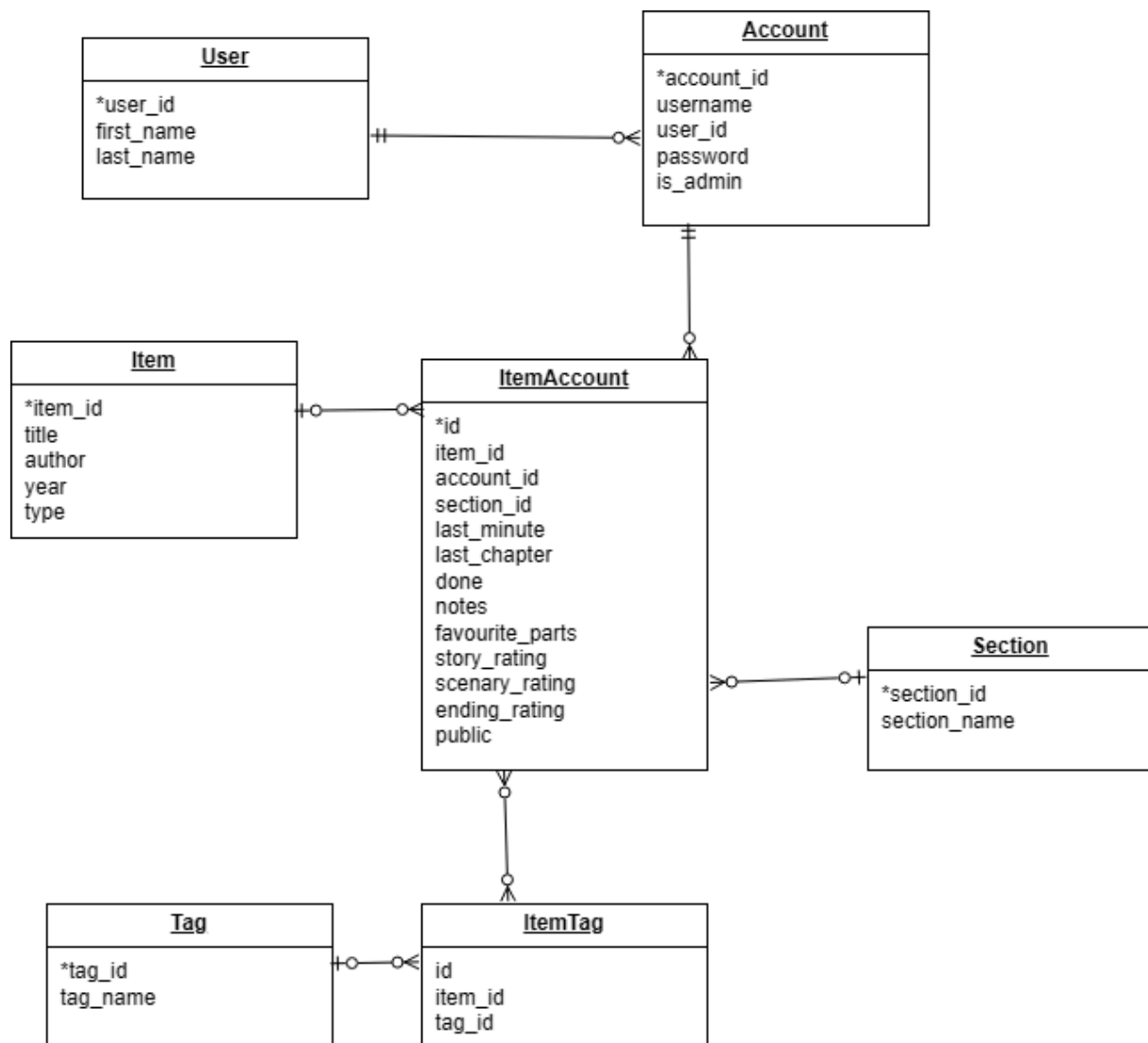
- doplňujúce informácie pre daný záznam z Filmografie a daného užívateľa
- Obmedzenia a pravidlá:
 - existuje len jeden záznam s kombináciou item_id a account_id
- Atribúty:
 - done: bool či užívateľ dokončil film / seriál
 - public: bool či vie zobrazovať záznam aj neprihlásený užívateľ
 - notes: komentáre

Tag

- základné budú z databázy, nové tagy vie vytvárať každý používateľ (pokiaľ nie je daný tag v databáze, vznikne nový záznam), ktoré vie následne používať každý používateľ
- vymazávať tagy vie len admin

Section

- základné budú z databázy, nový section vie vytvárať každý používateľ (pokiaľ nie je daný section v databáze, vznikne nový záznam), ktoré vie následne používať každý používateľ
- vymazávať sections vie len admin



Architektúra

Architektúra bude založená na client-server architektúre a bude používať SPA (Single Page Application) prístup.

Technické požiadavky

- Client-side: React 19.0.0, TypeScript, HTML5, CSS3, Tailwind 3
- Server-side: node.js v20.17.0, JavaScript, Express.js 4.21.2
- Databáza: PostgreSQL 17
- Interface client-server: Rest API
- Hosting: render.com
- Podporované prehliadače: chrome, safari, firefox

Časový plán

obsah Beta verzie

- testovacie dáta v databáze pre mock-frontend
- Základná funkcionálna aplikácia implementovaná end-to-end v oboch smeroch
 - Spracovanie vstupu používateľa - tok dát z frontendu do DB (Controller → API → DB)
 - registrácia nového užívateľa
 - zápis a zmena kníh a filmov / seriálov daného užívateľa
 - Tok dát naspäť k používateľovi (DB → API → View)
 - zobrazenie kníh a filmov / seriálov pre daného užívateľa
 - zobrazenie verejných kníh a filmov / seriálov daných užívateľov
 - zobrazenie tagov, sections, záznamov a užívateľov na manažovanie pre adminov
- Všetky ostatné funkcionality implementované minimálne ako mock frontend
- Implementovaná autentifikácia

4. týždeň

- vytvorenie wireframes
- setup projektu
- vytvorenie subpages + dizajn

5. týždeň

- vytvorenie databázy a tabuliek
- migrovanie tabuliek databázy

6. týždeň

- set up backend
- API pre backend

7. týždeň

- autentifikácie, zabezpečovanie hesla užívateľov

8. týždeň

- deployment beta verzie

9. týždeň (Beta verzia)

- fix buggov a update po spätnej väzbe

10. týždeň

- fix buggov

11. týždeň (Finálna verzia)

- deployment finálnej verzie

Budúca práca

- Rozšírenie o webview aplikáciu prípadne čisto lokálne