

# Zápisnica č. 4

Tím 17 – EVP  
08. novembra 2022

**Prítomní:** všetci

**Miesto stretnutia:** 3.38, FIIT STU

## TODO:

- ☐ Dokončiť retrospektívu
- ☐ Rozdeliť user stories na tasky
  - ☐ Ohodnotiť tasky, dať estimates
  - ☐ Priradiť tasky k šprintu
  - ☐ Rozdeliť si tasky

## Body stretnutia:

- Prebratie si výsledkov šprintu (jednotlivých user stories [US])
  - Analýza technológií
  - Prípady použitia + hierarchie obrazoviek
  - CERIF
- Zadefinovanie nových user stories [new US]

## Poznámky zo stretnutia:

- **[US] Analýza technológií**
  - Potrebný technickejší popis — minimálne snippets; ako sa s tým robí
    - Dopísať do dokumentu – všetky technológie čo máme v tabuľkách
  - Zatiaľ sa žiadne verziovanie riešiť nebude – vedúci to vidí na Postgres (s ktorým vraj robí aj druhý tím)
  - Výbrané technológie:
    - Django, React
    - Databáza – buď nejaká grafová alebo Postgres
- **[new US] Analýza možností využitia grafových databáz**
  - *Alternatívna grafová db v spojení s Djangom*
    - Iné možnosti čo sa týka grafových db
  - *Dokument, v rámci ktorého bude mapovanie hlavných entít modelu CERIF (vrátane snippetov)*
    - Zamerať sa na to, čo sa dá zadefinovať v rámci vzťahov
  - Menšia US – max dvaja ľudia
  - Dopísať benchmarks databáz – pozor na porovnania (či v príkladoch nie sú queries výhodnejšie pre konkrétne db)

- **[new US] Implementácia základného prototypu**
  - Nasadiť backend, spraviť základný frontend v Reacte, prepojiť to s databázou
    - Len nech to niečo robí a máme prvú verziu (klúčne aj hello world, najlepšie niečo potiahnuť z db)
    - S backendom a frontendom sa nič nebude meniť – jedine s db
  - Zatiaľ to klúčne môžeme cez Neo4j
    - Dá sa to prepojiť s analýzou – vyskúšať neo4j alebo ďalšie alternatívy
    - Neo4j **testing** nemusíme pushovať – ale treba to zdokumentovať
      - Testovane db dopísať do docu – netreba do GitLab
  - Testovacia vzorka pre db? Export zo SK CRIS alebo získať od nich
  - Zatiaľ **lokálne** zdockerizované, ale nech v **Gite** už máme projekt
    - Git → Docker → Azure/AWS student plan
- **[US] Prípady použitia + hierarchie obrazoviek**
  - Finančný riešiteľ ani sysadmin nás netrapí
    - Pokiaľ dokážeme prejsť hlavným activity diagramom (so schváľovaním žiadostí etc), tak je to úplne jedno, ako to dáme
- **[new US] Návrh obrazoviek z dokumentu**
  - Mid-fidelity
  - V čom spravíme návrh obrazoviek? Figma
  - Nech sa to dá preklikávať etc
  - Nech to zadávatelia schvália alebo povedia, čo zmeniť
  - Vyhľadávanie vložiť podľa [existujúceho projektu SK CRIS](#)
    - Pozrieť si výstupy + register projektov, keď budeme riešiť obrazovky
  - Bolo by dobré pozrieť si iný portal okrem SK CRIS (aby sme vedeli ako vyzerajú nejaké obrazovky – čo všetko tam dať)
    - PortalVS → Kega projects → Overview → Click na project
- **[US] CERIF**
  - Buď vytvoriť vlastné class id alebo nebudeme podporovať
  - Keď to nie je, teoreticky pracovať s tým, čo máme – zatiaľ pracovať priamo s nimi a potom new US do backlogu
- **[new US] Došpecifikovanie CERIF fieldov**
  - Vytvorenie vlastných fieldov
  - Dopísať k tomu dokumentáciu
- **Ďalšie poznamky**
  - Tasky su v našej réžii – vieme ich vytvoriť aj na stretnutí tímu
  - Retrospektíva
    - Netreba písať, že “stihli sme všetky tasky, stále pravidelné stretnutia”
    - Skôr sa zamerať na “Niečo sa udialo v tíme a malo to pozitívny dopad”
    - Nemusíme dávať nič do plusov (“čo bolo dobré na šprinte”)