

Tímový projekt 1

F. Samuhel, A. Mucska, S. Kobera, I. Budinský, L. Daniš, A. Dolan, T. Drus

5. mája 2022

Obsah

1 Úloha a analýza	1
1.1 Modul Nomenklátor	2
1.2 Časť - Klientská aplikácia - manažovanie kľúčov	2
1.3 Časť - Statistics	3
2 Predstavenie tímu	3
3 Funkcionálne požiadavky	4
3.1 Administrátorské rozhranie	4
3.2 Používateľské rozhranie	4
3.3 Štatistiky	5
4 Návrh systému	5
4.1 User flow diagram	5
4.2 Use-case diagramy	7
4.3 Databázový diagram	8
4.4 API	10
5 Používateľské prostredie	10
5.1 Úvodná stránka	10
5.2 Registrácia a prihlásenie	11
5.3 Administračná stránka	11
6 Vybrané technológie	12
7 Implementácia	13
7.1 API	13
7.2 Úvodná stránka	13
7.3 Prihlasovanie a registrácia	13
7.4 Moje pridané kľúče	15
7.5 Pridávanie kľúčov	15
7.6 Úprava kľúčov	15
7.7 Zmena stavu kľúča	15
7.8 Administrácia kľúčov	15

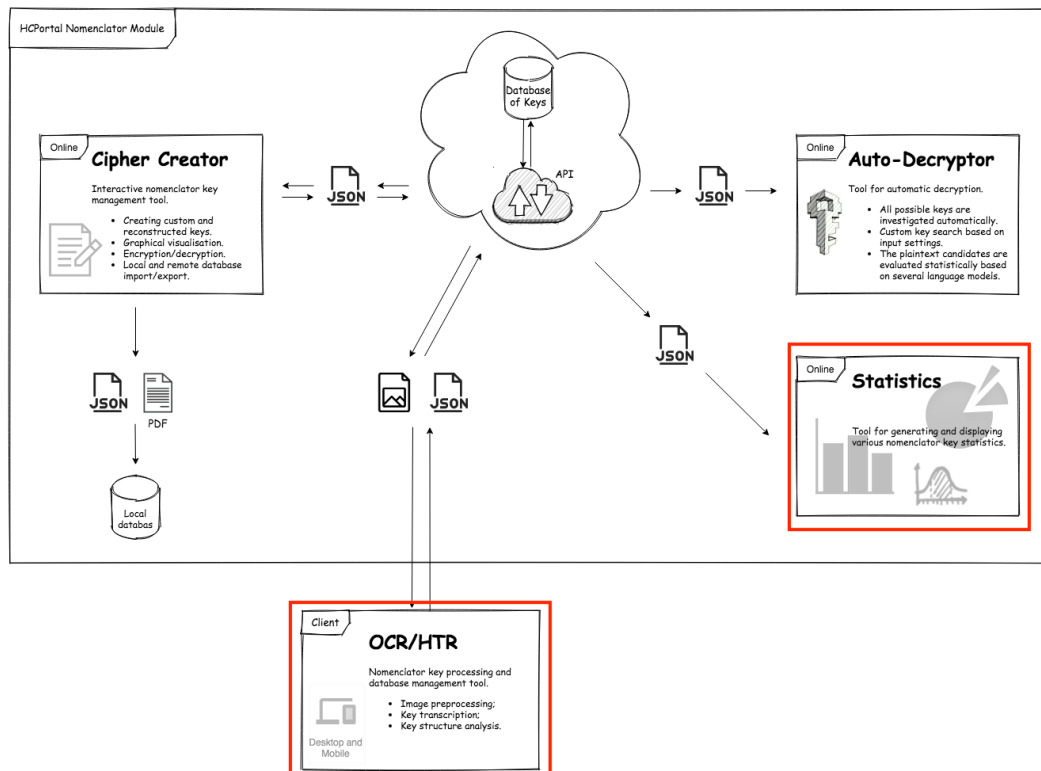
Obsah

1 Úloha a analýza

Úlohou v tímovom projekte je rozšíriť existujúci modul s názvom Nomenklátor, ktorý je súčasťou HCPortal (ďalej modul Nomenklátor), ktorý zahŕňa v sebe už existujúce časti. V nasledujúcej sekcii sa budeme zaoberať analýzou existujúceho modulu, ktorý sa skladá z viacerých častí.

1.1 Modul Nomenklátor

Existujúci modul Nomenklátor sa skladá z viacerých častí. Nasledujúci obrázok ilustruje jednotlivé časti celého modulu.



Obr. 1: Diagram modulu Nomenklátor a jeho častí

Existujúce časti modulu Cipher Creator, Database of Keys a API sú už hotové a pripravené na použitie. Cipher Creator a Auto Decryptor sú priamo prepojené cez API rozhranie s databázou nomenklátorových kľúčov a komunikujú navzájom prostredníctvom API rozhrania. Z tohto dôvodu je potrebné zachovať existujúcu štruktúru databázy a prípadné úpravy štruktúry databázy je nutné vykonávať opatrne s prihliadaním na kompatibilitu jednotlivých častí. Súčasťou modulu sú aj časti modulu OCR/HTR a Statistics, ktoré sú v diagrame zvýraznené červeným obdĺžnikom. Práve tieto dve časti budú predmetom našich úloh v tímovom projekte.

1.2 Časť - Klientská aplikácia - manažovanie kľúčov

V aktuálnom stave je manažovanie nomenklátorových kľúčov obtiažne, pretože sa vykonáva priamo v databáze. Obtiažnosť manažovania kľúčov a absencia rozhrania na manažovanie a zobrazenie kľúčov sú dôvodmi prečo prvou úlohou na ktorú sa zameriame bude tvorba webovej aplikácie. Webová aplikácia bude prepojená s databázou nomenklátorových kľúčov prostredníctvom API rozhrania. Účelom webovej aplikácie bude poskytnúť odbornej verejnosti prehľad aktuálne dostupných nomenklátorových kľúčov z databázy ako aj umožniť ich jednoduché manažovanie. Z toho dôvodu samotné manažovanie nomenklátorových kľúčov bude umožnené prostredníctvom administrátorského rozhrania. Nomenklátorové kľúče budú môcť pridávať prihlásení užívatelia ale aj administrátori. Súčasťou webovej aplikácie bude aj schvaľovací systém, ktorý dovoľí administrátorom schvaľovať a zamietajú už pridané kľúče užívateľov. Až po schválení pridaného kľúča administrátorom bude kľúč zverejnený pre verejnosť. V prípade, ak sa kľúč neschváli, administrátor cez administrátorské rozhranie notifikuje užívateľa s dôvodom jeho zamietnutia. Následne užívateľovi bude povolené kľúč upraviť a opätovne zaslať na schválenie.

Všetky vyššie spomenuté procesy budú napojené na koncové body API rozhrania, ktoré komunikujú s databázou. V diagrame spomenutú komunikáciu ilustrujú šípky z časti OCR/HTR do časti API a z časti API do časti Database of Keys. Bližšej špecifikácií systému sa budeme venovať v kapitole

Funkcionálne požiadavky.

1.3 Časť - Statistics

Jednou z ďalších úloh bude vytvorenie webovej stránky, ktorá bude zobrazovať štatistiky z už existujúcich kľúčov v databáze a novo pridaných kľúčov práve prostredníctvom vyššie spomenutej webovej aplikácie. Rovnako ako pri webovej aplikácii bude webová stránka komunikovať s databázou prostredníctvom API rozhrania.

2 Predstavenie tímu

Atila Mucska

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. 4 ročné skúsenosti v oblasti vývoja webových technológií. Vedúci tímu, ktorý sa staral o správny chod tímu a pracoval na vývoji veľkej časti aplikácie.

Samuel Kobera

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. Skúsenosti s vývojom full stack aplikácií a informačných systémov, v tímovom projekte sa zameral hlavne na vývoj backendu a správne navrhnutiu databázy.

Tomáš Drus

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. Dlhoročná prax v oblasti frontend technológií. V projekte sa venoval hlavne návrhu dizajnu systému a frontendovej časti projektu.

Alan Dolán

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. Skúsenosti z oblastí web developmentu v projekte pomáhal pri tvorbe backendu a taktiež mal na starosti tímovú stránku projektu.

Lukáš Daniš

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. Skúsenosti ako cloud engineer v projekte sa pričínal hlavne pri návrhu systému z pohľadu používateľa a pomohol pri práci s databázami.

Filip Samuhel

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. Skúsenosti z oblasti databázových systémov, kde sa podieľal na správnom návrhu systému a taktiež pracoval aj na frontend časti aplikácie.

Imrich Budinský

Študent FEI STU v odbore Aplikovaná informatika. Študuje v 1. ročníku inžinierskeho štúdia so zameraním Modelovanie a simulácia udalostných systémov. So skúsenosťami ako pri vývoji frontendu aj backendu sa zaslúžil ako pomoc pri návrhu oboch častí a taktiež vypomohol v obidvoch častiach.

3 Funkcionálne požiadavky

V tejto kapitole sa zamierame na špecifikovanie funkcionálnych požiadaviek webovej aplikácie, ktoré vyplývajú z analýzy.

3.1 Administrátorské rozhranie

Administrátorské rozhranie bude súčasťou webovej aplikácie a má slúžiť ako schvaľovací systém nomenklátorových kľúčov. Po prihlásení administrátora (pomocou API rozhrania) sa zobrazí zoznam všetkých pridaných kľúčov nad ktorými administrátor bude môcť vykonať nasledovné operácie: Meniť stav kľúča (Čaká sa na schválenie, Schválený, Vyžiadaná zmena, Zamietnutý), upraviť a zmazať kľúč. Všetky procesy budú komunikovať s databázou prostredníctvom API rozhrania.

ID	Požiadavka	Popis požiadavky
FPA1	Administrátorské rozhranie a autentifikácia	Vyžaduje sa administrátorské rozhranie, do ktorého sa administrátori budú vedieť autentifikovať a následne v ňom schvaľovať nomenklátorové kľúče. Autentifikácia musí prebiehať cez API rozhranie.
FPA2	Zoznam kľúčov v admin rozhraní	Zobrazenie všetkých pridaných kľúčov v systéme v zozname. Filtrácia kľúčov podľa schválenia, zoradovanie, paginácia.
FPA3	Schválenie kľúča	Umožniť schválenie kľúča. Kľúč nadobudne stav "SCHVÁLENÝ"
FPA4	Zamietnutie kľúča	Umožniť zamietnutie kľúča a možnosť uvedenia dôvodu pre ktorý bol kľúč zamietnutý. Kľúč nadobudne stav "ZAMIETNUTÝ" s odôvodnením.
FPA5	Vyžiadanie zmeny kľúča	Možnosť vyžiadania zmeny v pridanom kľúči. Kľúč nadobudne stav "ČAKÁ SA NA ZMENU" s poznámkami, ktoré užívateľ bude musieť vykonať, aby kľúč bol schválený.
FPA6	Zmazanie kľúča	Umožniť administrátorovi zmazať kľúč.

3.2 Používateľské rozhranie

Používateľské rozhranie bude slúžiť na pridávanie a upravovanie kľúčov prihláseným užívateľom. Naopak verejnosť bude môcť prezerať všetky schválené nomenklátorové kľúče.

ID	Požiadavka	Popis požiadavky
FPU1	Prihlasovací systém a registrácia	Vytvoriť rozhranie pre prihlásenie užívateľov. Autentifikácia bude prebiehať prostredníctvom API rozhrania. Registrácia bude rovnako prebiehať cez API endpoint.
FPU2	Pridávanie, editácia nomenklátorových kľúčov	Umožniť pridávanie a editovanie nomenklátorových kľúčov prihláseným užívateľom. Pri pridávaní a editovaní sa bude rovnako komunikovať s API rozhraním. Pri upravovaní kľúča bude úprava fotiek znemožnená kvôli zachovaniu integrity dát.
FPU3	Zoznam pridaných kľúčov prihláseného užívateľa	Zoznam všetkých kľúčov, ktoré pridal prihlásený užívateľ. Pri jednotlivých pridaných kľúčoch sa zobrazí flag, či kľúč bol schválený, čaká sa na schválenie, vyžaduje sa zmena alebo bol zamietnutý. V prípade ak administrátor označil kľúč flagom "Vyžaduje sa zmena", tak automaticky užívateľ uvidí akú zmenu musí spraviť, aby kľúč bol schválený. Analogicky aj pri stave "Zamietnutý" sa zobrazí správa teda dôvod zamietnutia.
FPU4	Zoznam schválených kľúčov	Zobrazenie zoznamu všetkých schválených kľúčov verejnosti s thumbnailmi kľúčov a najdôležitejšími metadátami.
FPU5	Detail schváleného kľúča	Zobrazenie detailu schváleného kľúča so všetkými metadátami a obrázkami.

3.3 Štatistiky

Zobrazenie štatistík nomenklátorových kľúčov z databázy bude na separátnej webovej stránke.

ID	Požiadavka	Popis požiadavky
FPS1	Zobrazenie štatistík	Zviditeľniť odbornej verejnosti zaujímavé štatistiky z nomenklátorových kľúčov prostredníctvom rôznych grafov.

4 Návrh systému

V tejto kapitole sa zameriame na návrh systému webovej aplikácie a taktiež návrhu úprav do existujúcej časti API rozhrania modulu HCPortal. Návrh systému a úpravy API rozhrania budeme zobrazovať v rôznych diagramoch, ktoré následne detailne opíšeme a vysvetlíme.

4.1 User flow diagram

User flow diagram nám slúži na grafické znázornenie jednotlivých krokov, postupov alebo nejakého procesu vrámci webovej aplikácie. Znázorňuje všetky podstránky ktoré bude môcť užívateľ prehliadať, akým spôsobom a za akých podmienok.

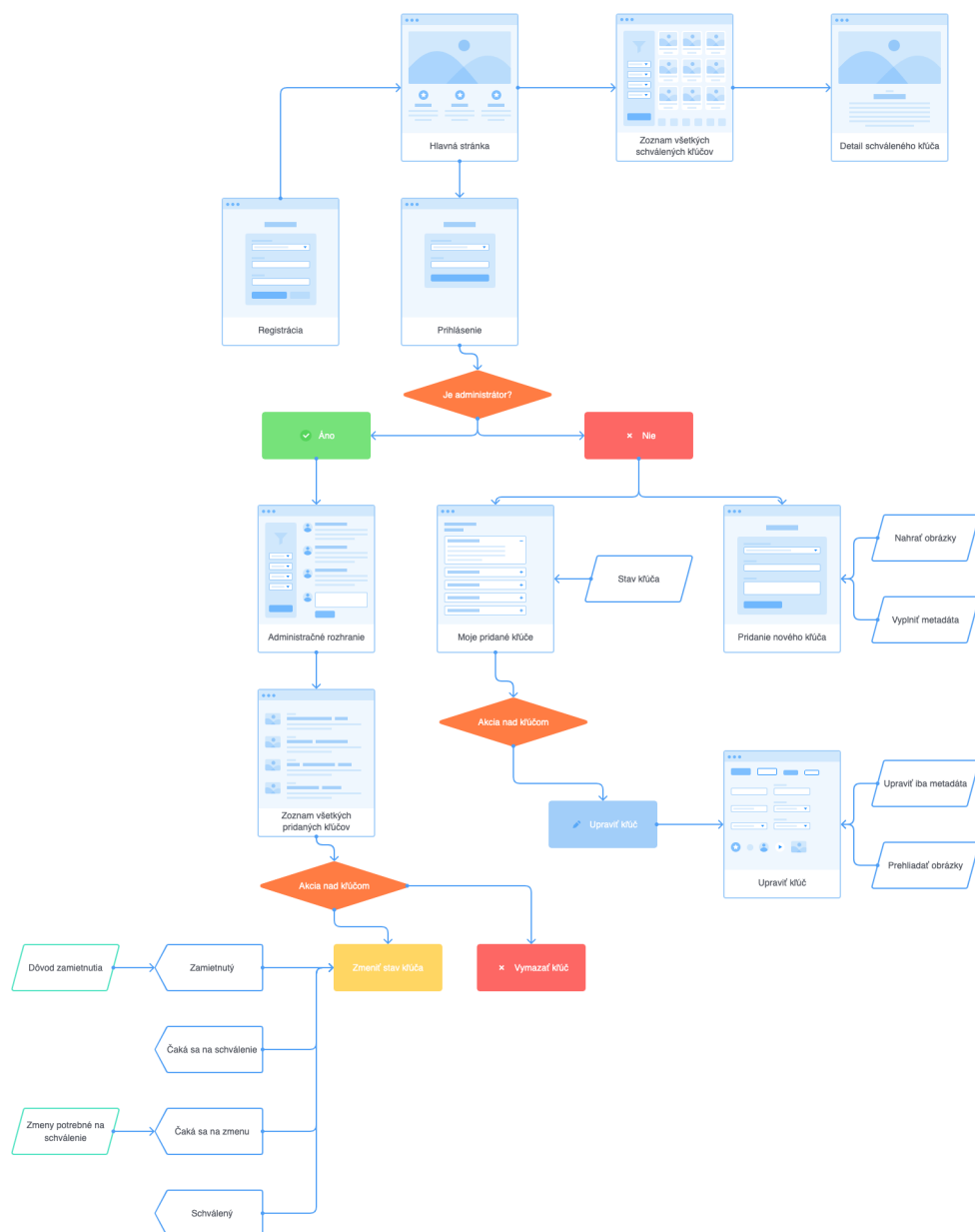
V našej webovej aplikácii (Obr. 2) ako prvé užívateľ uvidí úvodnú stránku, ktorá ho bude informovať o ciele webovej aplikácii. Z hlavnej stránky sa užívateľ bude vedieť dostať z hlavného menu na ďalšie podstránky. Bez potreby prihlásenia bude môcť užívateľ prehliadať aj podstránku na ktorej budú zobrazené všetky schválené nomenklátorové kľúče z databázy. Po kliknutí na nomenklátorový kľúč sa mu otvorí detail kliknutého nomenklátorového kľúča. Detail bude obsahovať obrázky nomenklátorového kľúča so všetkými jeho metadátami.

Okrem vyššie spomenutých podstránok sa užívateľ bude vedieť preklikať aj na podstránku s registračným formulárom. Rovnako z hlavnej stránky sa registrovaní užívatelia budú môcť prihlásiť na separátnej podstránke. Pri prihlasovaní užívateľa systém automaticky vyhodnotí, či účet má rolu administrátora alebo klasického užívateľa. Ak sa bude jednať o účet administrátora, tak systém presmeruje užívateľa do administrátorského rozhrania. V administrátorskom rozhraní sa zobrazí podstránka so všetkými pridanými nomenklátorovými kľúčmi, ktoré sú už schválené, zamietnuté alebo čakajú na schválenie. Prihlásenému administrátorovi bude umožnené nad týmito kľúčmi vykonávať dve akcie. Prvá akcia, ktorú administrátor bude môcť vykonať je vymazanie kľúča z databázy. Druhá akcia je zmena stavu kľúča. Kľúč bude môcť nadobudnúť tieto stavy:

1. Schválený - kľúč bol úspešne schválený a zobrazí sa verejnosti
2. Zamietnutý - kľúč nebol schválený a nezobrazí sa verejnosti
3. Čaká sa na zmenu - kľúč nebol schválený, ale je vyžiadaná zmena od administrátora, ktorú užívateľ, ktorý pridal kľúč musí vykonať, aby kľúč bol následne schválený
4. Čaká na schválenie - kľúč, ktorý ešte nebol administrátorom preverený

Ak nastane prípad, že prihlásený užívateľ nie je administrátorom, tak mu bude umožnené pridať nomenklátorový kľúč a zobrazí jeho doteraz pridané nomenklátorové kľúče. Na podstránke s formulárom na pridanie nomenklátorového kľúča sa zobrazia polia, ktoré vyplní metadátami pridávaného kľúča. Následne bude nevyhnutné k pridávanému kľúču pripojiť aj k nemu súvisiace obrázky. Po úspešnom pridaní kľúča sa kľúč uloží do databázy a zobrazí sa v administrátorskom rozhraní medzi kľúčmi, ktoré čakajú na schválenie.

Ak sa užívateľ rozhodne niektorí pridaných nomenklátorových kľúčov zo zoznamu všetkých jeho doteraz pridaných kľúčov zmeniť, tak sa mu zobrazí podstránka s formulárom, kde už polia budú predvyplnené s metadátami. V tomto formulári bude môcť vykonať zmeny iba v metadátach. Obrázky nebude mať povolené zmeniť iba prehliadať a to z dôvodu zachovania integrity dát.

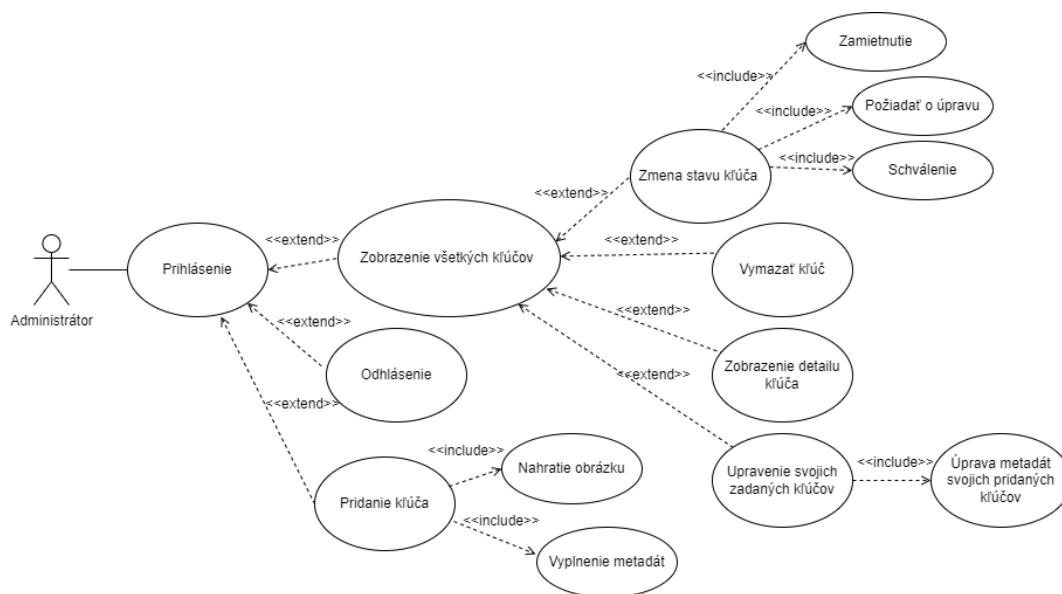


Obr. 2: User flow diagram

4.2 Use-case diagramy

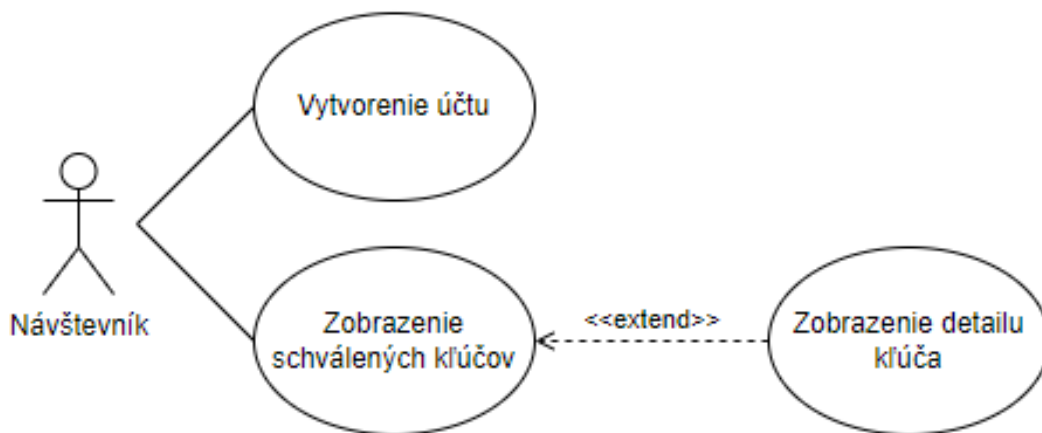
Diagram prípadov použitia (angl. Use-case diagram) je dynamický diagram navrhnutý v modelovacom jazyku UML. Modeluje funkcionálnosť celého systému pomocou aktérov a prípadov použitia. Prípady použitia sú súborom akcií a funkcií, s ktorými má navrhovaný systém disponovať. V tejto súvislosti rozumieme pod pojmom „systém“ niečo, čo sa vyvíja alebo prevádzkuje. Napríklad môže ísť o webovú stránku. Aktérmi (angl. Actors) sú ľudia alebo subjekty, ktoré majú v systéme určité role.

Pre lepšiu predstavu modelácie databázových tabuliek sme spracovali Use-case diagramy, ktoré nám vizualizovali jednotlivých aktérov, prípady použitia a vzťahy medzi nimi. V našom systéme figurujú traja aktéri. Prvým aktérom je administrátor, ktorý reprezentuje fyzickú osobu s dostatočnými znalosťami na schválenie alebo zamietnutie zadaných nomenklátorových kľúčov. Primárne má na starosti správu všetkých kľúčov nahraných na našej webovej stránke, ale taktiež vie nahráť kľúče, ktoré už nie je potrebné schvaľovať.



Obr. 3: Use-case administrátora

Druhý aktér je návštevník stránky, ktorý potrebuje len prezerať existujúce schválené kľúče a má možnosť zobrazenia detailu každého z týchto kľúčov. V prípade záujmu o pridávanie kľúčov, má možnosť vytvorenia si nového účtu.



Obr. 4: Use-case návštevníka

Posledným aktérom je používateľ, ktorý má už vytvorený účet. Má rovnaké možnosti akcií ako návštevník a taktiež niektoré nové. Po prihlásení má možnosť pridávania vlastných kľúčov. Pre pridanie kľúča je potrebné zadať obrázok a metadáta. Následne si môže zobrazíť zoznam všetkých kľúčov, ktoré daný používateľ vytvoril. Z tohto zoznamu má možnosť vykonať úpravu metadát niektorého z kľúčov. Pre ukončenie vykonávaní akcií v režime používateľa je možnosť vykonať akciu odhlásenia sa.



Obr. 5: Use-case používateľa

4.3 Databázový diagram

Ako vidíme na databázovom diagrame (obr. 6), zobrali sme pôvodný návrh Ing. Gona a pozmenili s kľúčovými značkami, aby bolo jasné, kde nastanú zmeny. Použili sme značky **RM**, čo znamená "removed", teda vymazané. Druhá značka je pre nové, **NEW**. Tieto značky v diagrame budú v databáze ignorované, takže napr. nová tabuľka, ktorá má vzniknúť, sa bude volať **nomenclatorKeyState**. Keďže nechceme robiť veľké zmeny do pôvodného návrhu databázy, pokúsili sme sa ich urobiť čo najmenej a aby boli ľahko zakomponovateľné.

Prvá zmena slúži pre riadenie stavu kľúča. Z tabuľky **nomenclatorKeys** vymažeme stĺpec **uploadedBy** a pridáme **stateId**, ktorý je cudzím kľúčom ukazujúcim na novo vzniknutú tabuľku **nomenclatorKeyState**. Táto tabuľka bude obsahovať stĺpce:

1. **id** - Stav bude zachovaný aj po zmazení **nomenclatorKey**, takže táto tabuľka potrebuje vlastné id
2. **state** - Tu budeme ukladať stavy z enumerovaného zoznamu.
3. **createdBy** - Premiestnenie stĺpca z tabuľky **nomenclatorKeys**
4. **createdAt** - Timestamp vytvorenia nomenklátorového kľúča
5. **updatedAt** - Timestamp poslednej úpravy nomenklátorového kľúča, dá nám informáciu, či vôbec bol upravený vďaka porovnaniu s **createdAt**
6. **note** - Textové pole, kde môže admin uviesť dôvod zmeny stavu, napr. na **Neschválené** alebo pre zmazanie, aby používateľ mohol chybu zo zamietnutého kľúča odstrániť, alebo namiesto zmazaného, nový kľúč nahradiť správne

Stĺpec **date** mažeme v prospech stĺpcov **usedFrom** a **usedTo**. Oba sú nulovateľné a zadáme z nich všetky hodnoty, ktoré poznáme. Pridávame aj možnosť **usedAround**, keď nepoznáme dostatočnú odpoveď na predchádzajúce dva stĺpce. Uvádzame značku **POSS**, pretože tento stĺpec možno po dohode nepridáme a určíme si pravidlo zápisu pri nepresnom údaji aj s dvoma stĺpcami.

Ďalšími stĺpcami, ktoré pridávame do tabuľky **nomenclatorKeys** sú:

- **usedChars** - Tu budú zhrnuté všetky použité znaky z viacerých obrázkov. Každý použitý bude uvedený iba raz.
- **cipherType** - V tabuľke sa nachádzajú hodnoty ako **nomenklator**, **kod**, **special/polyalfabeticka substitucia?**, **homofonna s klamacmi**, **substitucia jednoduchá**, **homofonna substitucia**, **homofonna substituca a klamace**. Bud' vytvoríme konštantné pole s možnosťami, alebo z databázy vyberieme z tejto tabuľky iba tento stĺpec navyše pomocou kľúčového slova **DISTINCT**, aby sme používateľovi okrem možnosti zadať text dali aj zoznam možností z už použitých typov šifier.
- **keyType** - Typy kľúča sú z konštantne zadaného poľa možností **e**, **d**, **ed** alebo **null**. Pričom **e** znamená encryption, **d** decryption a hodnota **null** sa použije keď používateľ neviem.
- **groupId** - Pre jeden nomenklátorový kľúč môžu vzniknúť viaceré záznamy, napr. keď máme inú fotku, ktorú chceme tiež použiť. Tieto kľúče samozrejme budú mať rôzne **id**, ale rovnaké **group_id**. Iná verzia rovnakého kľúča sa vytvorí bud' pomocou kopírovania z existujúceho, kedy vytváranie zobrazí formulár podľa predlohy, alebo sa bude vytvárať klasicky ale poskytneme **select**, kde si používateľ vyberie, ku ktorému kľúču patrí.

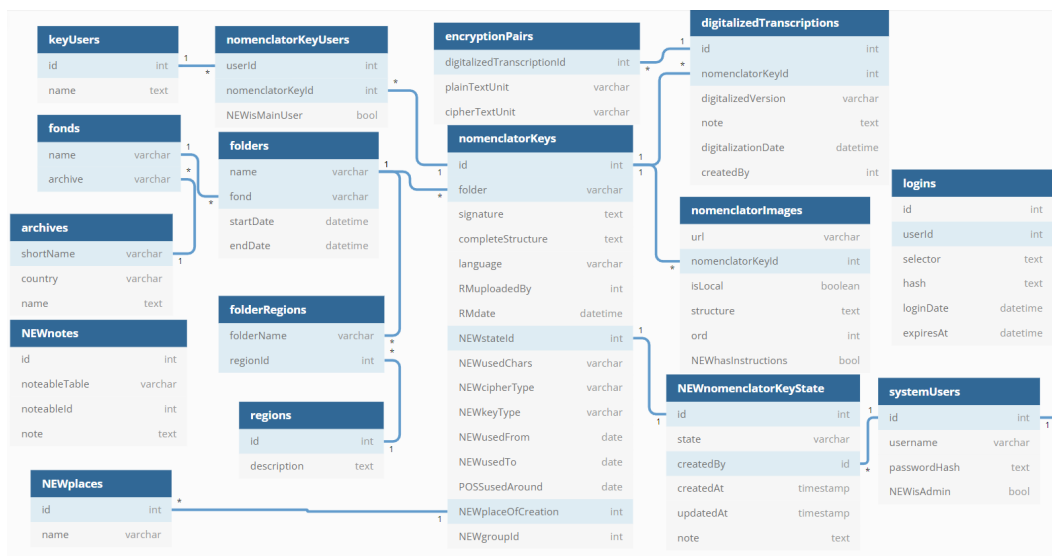
Cudzí kľúč **placeOfCreation** sa síce nachádza v tabuľke **nomenclatorKeys**, ale vzniká k nemu aj nová tabuľka **places**. Tu opäť ponúkame možnosť výberu zo zoznamu už existujúcich miest, alebo napísanie názvu nového.

Do tabuľky **nomenclatorKeyUsers** sme pridali **boolean** (v MySQL **tinyint(1)**) stĺpec **isMainUser**, pretože kľúč môže mať viacerých používateľov, tak aby sme ich vedeli deliť. Zároveň sú v tejto tabuľke a nie v **keyUsers**, pretože daný používateľ kľúča môže byť pri jednom hlavný a pri inom bežný používateľ.

hasInstructions je tiež typu **boolean** a hovorí o tom, či má daný obrázok v sebe návod. Ak ho necháme v tabuľke **nomenclatorImages**, budeme vedieť o každom obrázku zvlášť, či návod obsahuje. Môžeme zväziť presun do tabuľky **nomenclatorKeys**, kedy bude táto informácia ľahšie filtrovateľná, ale nebudeme vedieť kde konkrétne sa nachádza.

Ďalej vzniká tabuľka **notes**, ktorá umožní pridať poznámku ku záznamu z každej tabuľky. Docielime to tak, že do stĺpca **noteableTable** pridáme názov tabuľky, do ktorej sa ideme pozeráť a do stĺpca **noteableId** id, resp. hodnotu primárneho kľúča záznamu v tejto tabuľke.

Napokon pri používateľoch potrebujeme odlíšiť administrátora od bežného používateľa. Do **systemUsers** pridávame **boolean** stĺpec **isAdmin**.



Obr. 6: Databázový diagram po úprave

4.4 API

Api si popíšeme z pohľadu ciest a kde dané cesty na aký účel budeme používať.

POST /login

Parametre

- username
- password

Používateľ sa môže chcieť prihlásiť, použijeme na to API volanie /login, cez ktoré pošleme používateľov login a heslo. Ak budú tieto údaje autorizované, dostaneme naspäť token, vďaka ktorému bude mať používateľ rozšírené možnosti. Ďalej musíme dostať informáciu, či ide o admina, alebo o bežného používateľa, aby sme vedeli o koľko možnosti rozšíriť. Keďže ide o citlivé dáta, uložíme ich do **session** na strane servera.

POST /register

Parametre Parametre

- username
- password

Vracia rovnaké hodnoty ako /login, ale pre novo vytvoreného používateľa na základe poslaných dát.

poznámky: logout len vymazeme tokeny, nemusi byt api call, ale api token ma expiration date, takže by mal byt testovací call, ci je token este funkcný aby sme usera vedeli presmerovat na login page ak nie je.

GET /nomenclatorKeys?state&token

Na túto tabuľku sa vždy automaticky right joinuje **nomenclatorKeyState**. Toto api volanie budeme používať na hlavnej stránke, kde bude zoznam všetkých kľúčov. To, či dostaneme iba schválené, všetky, alebo používateľom vytvorené nomenklátorové kľúče by mal backend určiť podľa tokena. Adminovi by mal vrátiť všetky, prihlásenému používateľovi schválené a ním vytvorené a neprihlásenému používateľovi iba schválené.

GET /nomenclatorKey

K tabuľke je potrebné pripojiť všetky ostatné, pretože to budeme používať v pohľade **show**, kde má používateľ prístup ku všetkým informáciám - k fotkám, miestam, párom, prepisom.

POST /nomenclatorKey

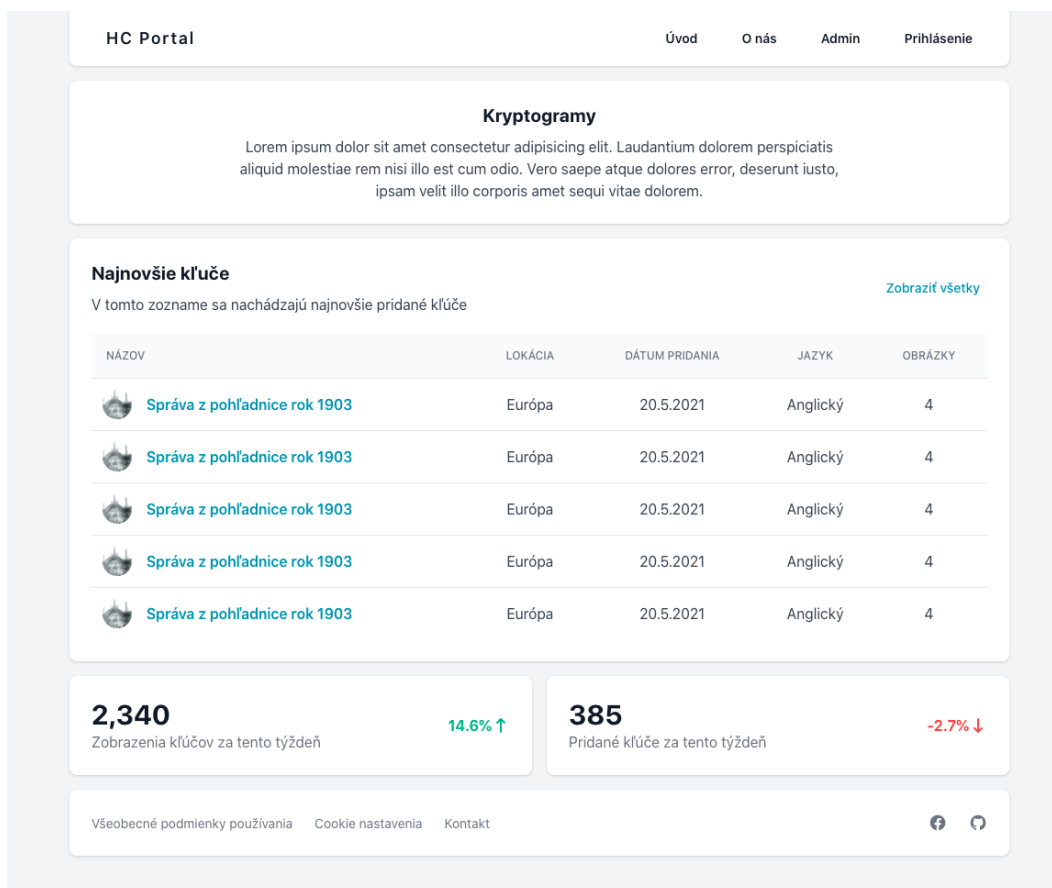
Cesta by mala prijímať všetky parametre, ktoré používateľ pri zadávaní vyplní, vrátane obrázkov.

5 Používateľské prostredie

Základom dobrej webovej stránky je navrhnuť vhodné používateľské prostredie, v ktorom sa budú používatelia ľahko orientovať. Preto je potrebné jednotlivé elementy na webovej stránke rozmiestniť intuitívne aby používateľ vždy našiel to čo hľadal. Grafický návrh webovej stránky musí pôsobiť príjemným dojmom aby používateľa neodradil od jeho používania. Toto zahŕňa vybratie vhodného štýlu textu a použitie správnych farieb. V našom prípade sme ako základné farby použili odtiene šedej, bielu a ako hlavnú kontrastnú farbu sme zvolili tyrkysovú. Dôraz sa tiež kladie na moderný vizuálny vzhľad webovej stránky.

5.1 Úvodná stránka

Úvodná stránka je prvá vec, ktorú používateľ uvidí pri návšteve našej webovej stránky. Na tejto stránke je potrebné zobraziť základné informácie aby používateľ vedel zameranie webovej stránky. Na úvodnej stránke sa ako prvé nachádza navigačné menu pre pohyb na webovej stránke. Ďalej je zobrazený základný popis webovej stránky a pod týmto popisom sa nachádza výpis najnovšie pridaných kľúčov a ich jednotlivé informácie. Na záver máme zobrazené základné štatistiky stránky a pätku stránky.



Obr. 7: Úvodná stránka

5.2 Registrácia a prihlásenie

Používateľ vie využívať základnú funkcionálnu webovú stránku bez potreby registrácie a prihlásenia. Táto funkcionálna stránka zahŕňa prístup k hlavným stránkam a zobrazeniu zoznamu všetkých kľúčov ako aj ich detailov.

Ak chce používateľ využívať rozšírenú funkcionálnu stránku je potrebné aby sa registroval pomocou registračného formuláru, ktorý je zobrazený na obrázku č. 8. V registračnom formulári je potrebné aby vyplnil všetky povinné polia ako sú meno, email, heslo a overenie hesla. Následne po úspešnej registrácii je možné sa prihlásiť cez prihlasovací formulár, ktorý je zobrazený na obrázku č. 8.

Po prihlásení vie používateľ pridávať nové kľúče, zobrazovať zoznam jeho pridávaných kľúčov alebo upravovať jeho existujúce kľúče.

5.3 Administračná stránka

V prípade ak má prihlásený používateľ pridelené administračné právomoci tak má prístup k administračnej stránke, ktorá je zobrazená na obrázku č. 9.

Na tejto stránke vidí administrátor zoznam všetkých kľúčov ako aj informáciu kto daný kľúč pridal, ktorá nie je normálne prístupná. Zobrazuje sa aj aktuálny stav jednotlivých kľúčov. Administrátor má možnosť úpravy kľúča a zmeny jeho stavu podľa potreby. V prípade ak je kľúč nevyhovujúci má možnosť takýto kľúč odstrániť.

Registrácia

Meno *

Email *

Heslo *

Heslo overenie *

* označené polia sú povinné

Registrácia

[Prihlásenie](#) [Domov](#)

Prihlásenie

Email

Heslo

[Zabudnuté heslo](#)

Prihlásenie

[Registrácia](#) [Domov](#)

(a) Registračný formulár

(b) Prihlasovací formulár

Obr. 8: Registrácia a prihlásenie













HC Portal

Úvod O nás Admin Prihlásenie

Všetky klúče

Vyhľadať v klúčoch



+ Pridať klúč

NÁZOV / AUTOR	LOKÁCIA	DÁTUM PRIDANIA	STAV	AKCIE
 Správa z pohľadnice rok 1903 jankomrkvicka@gmail.com	Európa	20.5.2021	Schválený	 
 Správa z pohľadnice rok 1903 jankomrkvicka@gmail.com	Európa	20.5.2021	Na schválenie	 
 Správa z pohľadnice rok 1903 jankomrkvicka@gmail.com	Európa	20.5.2021	Na zmenu	 
 Správa z pohľadnice rok 1903 jankomrkvicka@gmail.com	Európa	20.5.2021	Zamietnutý	 

< >

Zobrazené 1-20 zo 100

Všeobecné podmienky používania Cookie nastavenia Kontakt

Obr. 9: Administračná stránka

6 Vybrané technológie

Ako zadanie sme dostali urobiť frontendovú aplikáciu pre už existujúce api na základe diplomovej práce Ing. Gona. Toto v nás hneď evokovalo myšlienky vytvoriť jednoduchú Vuejs aplikáciu, ktorá by dávala používateľovi pocit modernosti, aj vďaka Vue Routeru, vďaka ktorému netreba načítavať podstránky pri pohybe v rámci aplikácie.

Narazili sme na problém autentifikácie. Vo Vuejs máme prístup iba k Vuex state manažéru, `window.cookie` alebo `localStorage`. Všetky tieto možnosti sú čitateľné používateľom, takže nemáme

bezpečné miesto, kde by sme uložili používateľov token a iné informácie o ňom.

Aby sme mohli poskytnúť bezpečnú autentifikáciu s čisto frontendovou aplikáciou, potrebovali by sme implementovať štandard OAuth 2.0. Tento štandard by musel byť implementovaný aj na backende, teda v API rozhraní a dá sa povedať, že zbytočne by to skomplikovalo prácu.

Preto sme sa rozhodli pre jednoduchšie riešenie a to použiť Laravel, v rámci ktorého dokážeme využiť výhody Javascript-ových komponentov cez Vuejs, aj keď prideme o moderný Vue Router. Výhoda je, že máme aj PHP kód, ktorý beží na strane servera a informácie o používateľovi, spolu s tokenom, môžeme ukladať do premennej `$_SESSION`, v Laraveli presnejšie pomocou globálneho helpera `session()`.

7 Implementácia

V tejto kapitole sa budeme venovať implementácií a opisu funkcionality, ktoré vyplynuli z návrhu systému. Opis jednotlivých modulov systému sme rozdelili logicky do podkapitol.

7.1 API

API request-y a ich spracovanie realizujeme prevažne v serverovej časti aplikácie (backend). Na vykonávanie requestu v Laraveli používame HTTP fasádu, ktorá je súčasťou frameworku Laravel. Odpovede resp. dáta, ktoré obdržíme z API requestov spracúvame rovnako v serverovej časti aplikácie.

7.2 Úvodná stránka

Na rozdiel od dizajnu úvodnej stránky z kapitoly Návrh systému (Obr. 7) sme pri implementácii funkcionality graficky upravili užívateľské prostredie. Okrem sekcie, kde sa nachádza opis účelu webovej stránky sme pridali okná so štatistikami z webu. Presnejšie sa jedná o (zmeniť):

- počet všetkých schválených nomenklátorových kľúčov administrátorom
- počet nomenklátorových kľúčov čakajúcich na schválenie
- počet vyriešených nomenklátorových kľúčov

Pod štatistickými oknami sa nachádza dátová tabuľka, ktorá zobrazuje najnovšie schválené nomenklátorové kľúče (ďalej iba kľúče). Na zobrazovanie kľúčov používame javascriptovú knižnicu DataTables. Kľúče získavame pomocou API requestu GET na URL `/nomenclatorKeys`. GET request vykonávame v controlleri **DashboardController** a to presnejšie v metóde `guest()`.

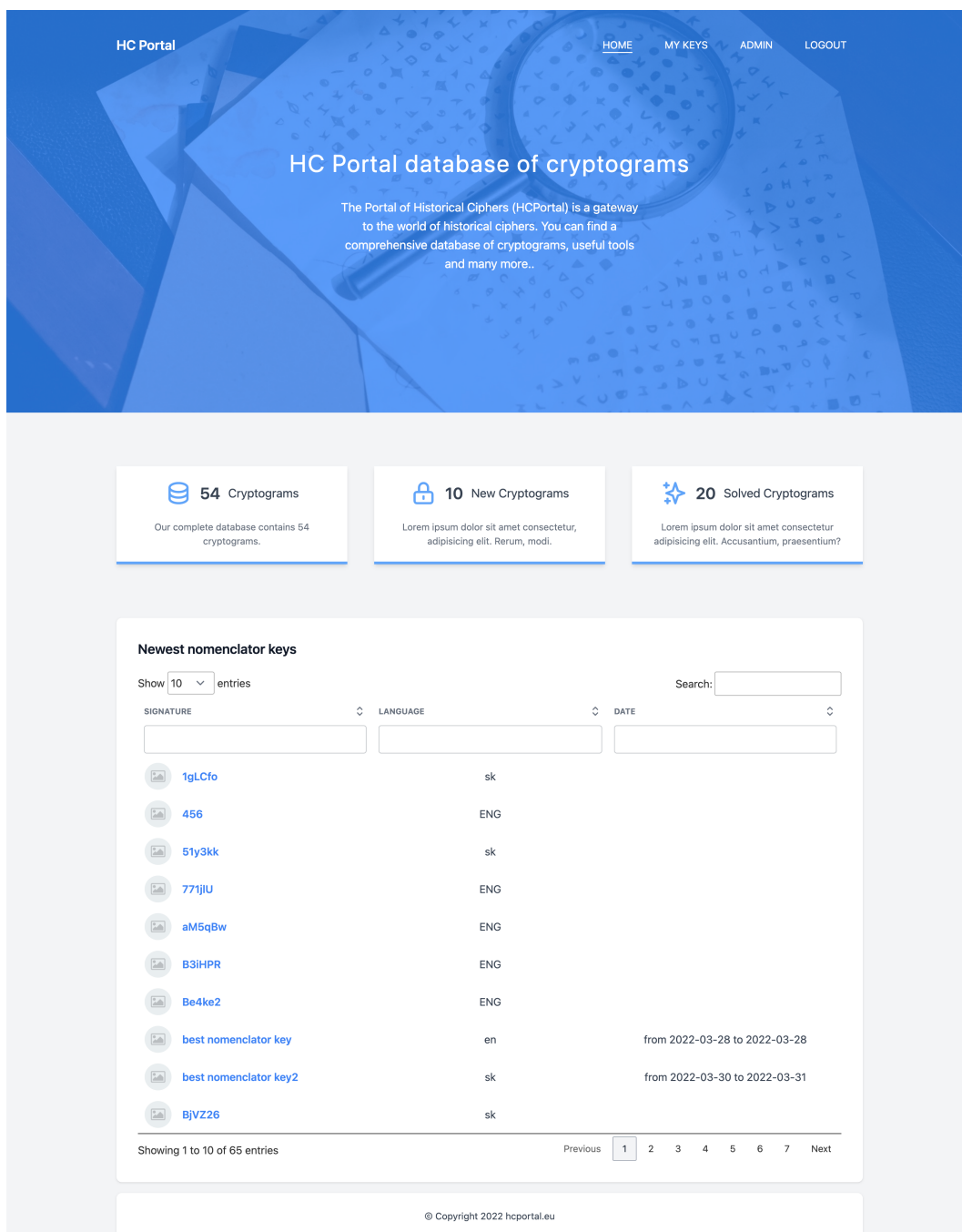
Odpoveď z API requestu následne zobrazíme v spomenutej dátovej tabuľke. V dátovej tabuľke je možnosť vyhľadávania kľúčov:

- vyhľadávanie vo všetkých stĺpcoch dátovej tabuľky
- vyhľadávanie na základe stĺpca "Signature"
- vyhľadávanie na základe stĺpca "Language"
- vyhľadávanie na základe stĺpca "Date"

Súčasťou dátovej tabuľky je aj paginácia zobrazených kľúčov.

7.3 Prihlasovanie a registrácia

Prihlasovanie je realizované cez API endpoint **login**. To be continued



Obr. 10: Úvodná stránka

```

public function guest()
{
    $response = Http::accept('application/json')->post($this->api_base_url . '/nomenclatorKeys');
    $keys = $response->json();

    return view('guest.index', [
        'keys' => $keys
    ]);
}

```

Obr. 11: GET request na získanie všetkých schválených kľúčov z API

- 7.4 Moje pridané kľúče
- 7.5 Pridávanie kľúčov
- 7.6 Úprava kľúčov
- 7.7 Zmena stavu kľúča
- 7.8 Administrácia kľúčov

Zápisnica 8.10.2021

Prítomní:

- Alan Dolán
- Samuel Kobera
- Attila Mucska
- Imrich Budinský
- Lukáš Daniš
- Filip Samuhel
- Tomáš Drus (neskorší príchod)

Poznámky:

- prvé stretnutie tímu, načrtlo sa ako bude vyzerat plán na ďalší týždeň, do ďalšieho týždňa treba potvrdiť a vytlačiť vyhlásenie

Zápisnica 15.10.2021

Prítomní:

- Tomáš Drus
- Alan Dolán
- Samuel Kobera
- Attila Mucska
- Imrich Budinský
- Lukáš Daniš
- Filip Samuhel

Poznámky:

- dva druhy používateľov: user, admin
- admin musí najprv schváliť
- všetko sa musí nachádzať na hostingu (v prípade migrácie), veľkosť disku 50GB
- nomenclator key je logickým celkom v DB, obsahuje metadata, 1:N obrázky a 1:N transkripcie
- nomenclator key metadata sú v podobe textu: obdobie, krajina, jazyk, mená
- transkripcie sa nebudú schvalovať, iba meta dáta a obrázky kľúča
- transkripcie sa budú pridávať inde ako v aplikácii (transkripcie sa robia až po schválení superuserom)
- jeden kluc môže mať viac transkripcií a viac obrázkov a treba zadať pri každej transkripcii z ktorých obrázkov sa skladá
- základom sú obrázky pri prvom pridávaní, po schválení nieje možné už meniť obrázky
- kópie kľúčov môžu byť odlišné a preto sa z nich robí nový objekt
- za prvý semester do 100 záznamov kľúčov treba nahráť na testovanie, druhý semester bude zameraný viac na nahrávanie údajov
- fotka kľúča má veľkosť priemerne 1-2mb(pár môže byť aj 50mb) okolo 200 obrázkov -> 62 až 100 unikátnych kľúčov
- cipher creator -> angular app aby človek mohol vyskladať svoj vlastný kľúč, pridať ďalšie metadata
- môže byť aplikácia, ktorá stiahne obrázok kľúča rozpozná dáta a z toho doplní údaje do databázy

Splnené úlohy:

- všetci členovia tímu podpísali a odovzdali čestné vyhlásenie

Zápisnica 12.11.2021

Prítomní:

- Alan Dolán
- Attila Mucska
- Tomáš Drus
- Samuel Kobera
- Filip Samuhel
- Imrich Budinský

Poznámky:

- Laravel projekt bol inicializovaný a pridán na github.
- Do Databázovej tabuľky bol pridán stĺpec „approved“ – bude značiť pri kľúčoch ich stav
- Na základe tejto zmeny bude treba modifikovať aj už existujúcu API, aby vracala ten stĺpec
- Do tabuľky logins bude potrebné tiež doplniť niekoľko stĺpcov
- Vytvorenie používateľského rozhrania pomocou nadstavby Craftable pre Laravel
- Namiesto Craftable by bolo možno lepšie použiť Vue.js, v prípade ak nie je potrebné použiť PHP, a ak nie je potrebné zmeniť štruktúru databázy
- POST request na API, obdržanie tokenu usera
- Po prihlásení by sa robili GET requesty na získavanie kľúčov
- Administrátor musí schváliť iným používateľom pridáný záznam
- Stav sa budú odlišovať pomocou flag-ov, či sa napríklad nejaký záznam už má zobrazovať alebo nie
- Stĺpec „approved“ bude typu string, ktorý bude zaznamenávať status daného záznamu
- Admina netreba schvalovať nikomu
- Admin nemôže editovať záznamy používateľov, len schváliť, prípadne nahlásiť problém používateľovi, ktorý nahral daný záznam – napríklad pomocou notifikácií
- Dátum pri záznamoch, kedy sa záznam zmenil alebo sa pridal atď.
- Používateľ vie editovať záznam aj po schválení, potom sa záznam musí znova schváliť napr.: preklepy v názve alebo v iných textoch, menšie zmeny

- mazanie záznamov by sa malo schvaľovať adminom, alebo len admin bude môcť záznam vymazať

Úlohy do budúceho stretnutia:

- založiť dokument - návrh, kde bude zhrnuté, čo chceme dosiahnuť, požiadavky, zmeny vykonané v pôvodnom systéme, diagramy atď...
- pozrieť si dokumenty, ktoré sú na trelle

Zápisnica 3.12.2021

Prítomní:

- Attila Mucska
- Tomáš Drus
- Samuel Kobera
- Filip Samuhel
- Imrich Budinský

Poznámky

- dokument má pozostávať minimálne z 10-15 strán do konca semestra
- najprv si treba premyslieť čo chceme dosiahnuť a implementovať a potom vybrať daný jazyk v ktorom sa to bude programovať
- analýza problematiky: na začiatok dokumentu treba napísať text cca. na jednu stranu, že čo už daný systém obsahuje a aké veci ešte plánujeme pridať
- požiadavky sú dobre napísané len treba doplniť niekoľko vecí
- treba popísať use-casy detailne, napr. čo môže robiť používateľ ak je prihlásený alebo ak je administrátorom, atď.
- ak je prihlásený, ale nie je admin, tak vie pridávať záznamy; vie si zobrazíť vlastné záznamy, kde bude značené v akom stave je daný záznam – schválené, čakajúce na schválenie a zmazané
- ak je schválené, môže to editnúť a následne zase počkať na schválenie
- takisto flow pre admina popísať
- treba trochu upraviť DB diagram aby to bolo prehľadnejšie
- pomocná tabuľka nomenclator_key_state by mala obsahovať: referencia na ID stavu, v tabuľke state bude uploadaded_by, upload_date, change_date, enum pre approved, deleted; note v prípade zmazania a prípade editu
- admin si vie vylistovať všetky kľúče, ktoré čakajú na schválenie
- nie je odporúčané použiť json, kvôli tomu, ak sa zmenia nejaké dáta, bolo by komplikované zmeniť celú štruktúru, a nesedeli by niektoré veci v budúcnosti

Úlohy do budúceho stretnutia

- rozšíriť návrh o text a diagramy
- upraviť diagram (prihlásenie namiesto registrácie)
- rozdeliť diagram aj na logické časti
- stanoviť milestony – ktoré časti kedy plánujeme dokončiť

Zápisnica 20.1.2022

Prítomní:

- Atilla Mucska
- Tomáš Drus
- Samuel Kobera
- Filip Samuhel
- Imrich Budinský
- Alán Dolán
- Lukáš Daniš

Poznámky:

- Názov "Existujúci modul nomenklátorových kľúčov HC Portál" je príliš dlhý, používať niečo kratšie, napr. "modul nomenklátor".
- Obr. 1: Diagram modulu Nomenklátor. Na časti OCR/HTR my nepracujeme. Našou úlohou je manažment kľúčov.
- Bude spoločné stretnutie oboch tímov ohľadom zjednoteného návrhu UI a iné.
- Use case Administrátora: chýba možnosť pridávania jeho vlastných prvkov. Admin by mal mať rovnakú možnosť ako používateľ.
- Front end sa robí postupne podľa toho, na akej časti sa práve pracuje. Nerobí sa celý dizajn dopredu.
- Databáza vyzerá byť v poriadku, netreba ju prekomplikovať.
- Poriestia sa ešte veci s ďalšími ľuďmi ohľadom databázy a návrhu. (Gono, ...) Následne budeme informovaný o prípadných zmenách.
- Jasnejšie definovať povinné údaje. Určiť, ktoré údaje môžu byť null a ktoré sú povinné.
- Pri registrácii treba ošetriť ochranu proti vytváraniu účtov botmi. Zakomponovať nejakú ochranu, napr. captcha, konfirmačný email alebo iné.
- Pridanie zápisníc na koniec dokumentu.
- Dohodnúť milestones pre nejaký prototyp aplikácie.

Milestones:

- Do konca januára nahodenie na live server, registrácia a login
- Do konca februára nejaký základný manažment: insert, edit. Doladenie API a DB
- Do konca marca testovanie a mať nahraté kľúče.
- Približne koniec mája / začiatok júna finalizácia a odovzdanie.

Zápisnica 28.2.2022

Prítomní:

- Atilla Mucska
- Alan Dolán
- Samuel Kobera
- Lukáš Daniš
- Imrich Budinský
- Filip Samuhel
- Tomáš Drus

Poznámky:

- Administrácia funguje na stránke, čakáme od Pána Gona v najbližších dňoch na dokumentáciu k API endpointom.
- Do testovania nepôjdu reálne kľúče z dôvodu práv.
- Systém vie identifikovať či je používateľ admin alebo bežný používateľ.

Milestones:

- Spojazdnené minimum, po prihlásení.
- Niečo nahráť.
- Minimálne UI, nemusí to byť 100%.
- Do budúceho týždňa mať minimálnu verziu.
- Štatistiky z databázy.

Zápisnica 14.3.2022

Prítomní:

- Atilla Mucska
- Tomáš Drus
- Samuel Kobera
- Filip Samuhel
- Imrich Budinský
- Alán Dolán
- Lukáš Daniš

Poznámky:

- Urobený dashboard pre admina a prihláseného používateľa.
- Ukážka frontendu stránky pre používateľa / admina.
- Netreba paging, môže sa načítať všetko a upraviť cez frontend.
- Upraviť zobrazenie na úvodnej stránke. Nezobrazovať niektoré informácie pri zozname kľúčov
- Stránka bude celá v angličtine.
- Pri pohľade admin / používateľ pridať zobrazenie dátumu upravenia kľúča.
- Možnosť zvolenia hodnoty 'od' 'do' roku, aby bola možnosť pridať aj hodnotu len s rokom.
- Prejdenie a porovnanie databázy s API

Plány do budúcnosti:

- Pridanie, vyhľadávanie a zobrazenie kľúča

Zápisnica

Prítomní:

- Alan Dolán
- Atilla Mucska
- Imrich Budinský
- Lukáš Daniš
- Filip Samuhel
- Lukáš Daniš
- Tomáš Drus

Poznámky:

- GroupID, nejaký model kde budú kľúče pridané.
- Place of creation: bud si zvolim alebo si pridam nový.
- Opraviť buggy.
- Urobiť finálne UI