

1. Popis aplikácie

Expense tracker je webová aplikácia na sledovanie vývoja hodnoty osobného majetku. Je určená pre bežného človeka, ktorý sa chce naučiť lepšie hospodáriť so svojimi financiami, vyhodnocovať výdaje a sledovať príjmy. Bežný človek funguje tak, že mu príde výplata a následne do ďalšej výplaty bezhlavo mína peniaze bez určitého plánu, keďže uchovávať každú transakciu v hlave je takmer nemožné. Potom sa ľudia spytujú samy seba, kam tie peniaze zmizli. Niektoré banky majú za týmto účelom spravené rozhrania, kde sa dajú transakcie zaraďovať do kategórií, ale sú dostupné iba pre klientov banky. Ja som napríklad skúšal vyriešiť zapisovaním do excelu a predpokladám, že aj viacero iných ľudí. Problém je ale, že Excel ponúka priveľa možností a nemá priateľské rozhranie pre človeka ktorý s ním nerobí dennodenne. Inšpiroval som sa aj myšlienkou aplikácií, ktoré slúžia na trackovanie kalórií alebo od aplikácie na zapisovanie cvikov spolu s hmotnosťami a opakovaniami.

Používateľ sa prihlási na stránke s autentifikáciou pomocou google účtu. Tento spôsob zjednodušuje registráciu tým, že si netreba vymýšľať prihlasovacie meno a heslo. Taktiež sa takto vyhnem tomu aby som musel uchovávať používateľove prihlasovacie údaje. Prihlásenie pretrvá aj po zavretí stránky a opätovnom otvorení, ak si používateľ pri prihlasovaní zaznačí možnosť "Zapamätať prihlásenie". Prihlásenému užívateľovi sa budú ukladať všetky vykonané zmeny. Novému užívateľovi sa automaticky nastaví meno a email z Google účtu.

Tieto údaje si bude môcť zmeniť a môže si nastaviť aj hlavnú menu v ktorej sa bude jeho majetok počítať, pričom predvolené bude euro. V nastavení účtu bude možnosť výberu zasielania pravidelných reportov na email. Táto funkcionálnosť začne fungovať až po vybratí časového intervalu odosielania. Bude možné si vybrať preferovaný formát dátumu a času. V rozhraní stránky si používateľ pridá jednotlivé zdroje financií ktoré vlastní ako sú kreditné karty, hotovosť, krypto peňaženky, bankové účty. Pri vytváraní budú preddefinované typy napríklad pri výbere bankového účtu sa bude dať zapísať IBAN, názov banky, pri kreditnej karte číslo karty, dátum expirácie, pri brokerových stránkach URL stránky. Všetky budú mať spoločné pole aktuálneho stavu, menu v akej má financie držané, poznámku, voliteľný názov, výber ikony. Bude možné pre jeden zdroj pridať viacero stavov a mien v prípade že drží kombináciu rôznych mien.

Na jednej podstránke bude možné vytvárať kategórie transakcií ako napríklad potraviny, výlet, živnosť a zábava. Polia na vyplnenie budú názov a ikona. Voliteľne bude možné nastaviť rozpočet na mesačný alebo ročný interval. Po prekročení množstva výdajov pre kategóriu s budgetom sa používateľovi zobrazí upozornenie.

Následne bude možné vytvárať, vymazať a editovať záznamy transakcií, príjmov, výdajov, nákupov, predajov, dlžôb a transferov z jedného zdroja na druhý. V prípade príjmov a výdajov bude možné vybrať z akého zdroja boli peniaze odpočítané, vybrať kategóriu, deň a čas, poznámku, názov a pripojiť súbor. Nákup je typ transakcie, ktorý sa bude využívať hlavne pri investíciách, kde síce odídu peniaze, ale z transakcie dostanem naspäť akciu alebo krypto. Bude možné vybrať zdroj z ktorého ubudli peniaze, poplatky, deň a čas. Následne bude možné pridať do zoznamu položky ktoré používateľ obdržal. Bude možné vybrať z možností akcie, mena, kryptomeny a NFT. Používateľ zapíše množstvo, názov,

menu, poznámku a cenu jedného kusu. V prípade nákupu akcie za inú menu sa dá nastaviť konverzný kurz. Predaj bude fungovať rovnako, pričom sa bude dať vyplniť cena predaja a cieľový účet kam boli peniaze odoslané. Transfer bude mať na výber: zdroj, cieľ, poplatok. Pri investíciách sa pri akciách a kryptomenách budú načítavať údaje z verejnej API. Ak hodnotu nebude možné zistiť z API, tak si používateľ môže nastaviť manuálne aktuálnu hodnotu alebo sa bude brať hodnota akú mala pri nákupe.

V transakcii príjmu a výdaju bude možné nastaviť opakovanie na báze dňa, týždňa, mesiaca, roka. Transakcie bude možné filtrovať podľa názvu, kategórie, typu transakcie, veľkosti transakcie a časového rozmedzia. V adresári bude možné vytvárať entity dlžníkov a prijímateľov. Tu sa bude dať zapísať meno a číslo účtu. Entita z adresára môže byť pridaná do transakcie.

Pri dlžobe sa bude dať vybrať entita z adresára, množstvo, popis a výber, či je používateľ dlžník alebo naopak. Po splatení dlžoby túto zmenu upraví v transakcii a dlžoba bude vybavená.

Ďalšia funkcionálnosť je nastavenie odosielania e-mailových reportov na mesačnej, ročnej a týždennej báze. Budú obsahovať zhrnutie transakcií za toto obdobie.

Ak používateľ má účet v banke VÚB tak si môže v ich IB exportovať transakcie a následne naimportovať do aplikácie. Dodatočne môže priradovať kategórie a nastaviť ďalšie polia.

Pre zobrazenie štatistiky bude spravená ďalšia podstránka kde bude predvolené časové obdobie od prvej transakcie až do prítomnosti. V grafe bude zobrazený vývoj hodnoty majetku v tomto časovom období. V zhrnutí bude zobrazené, akú hodnotu mali jednotlivé typy transakcií.

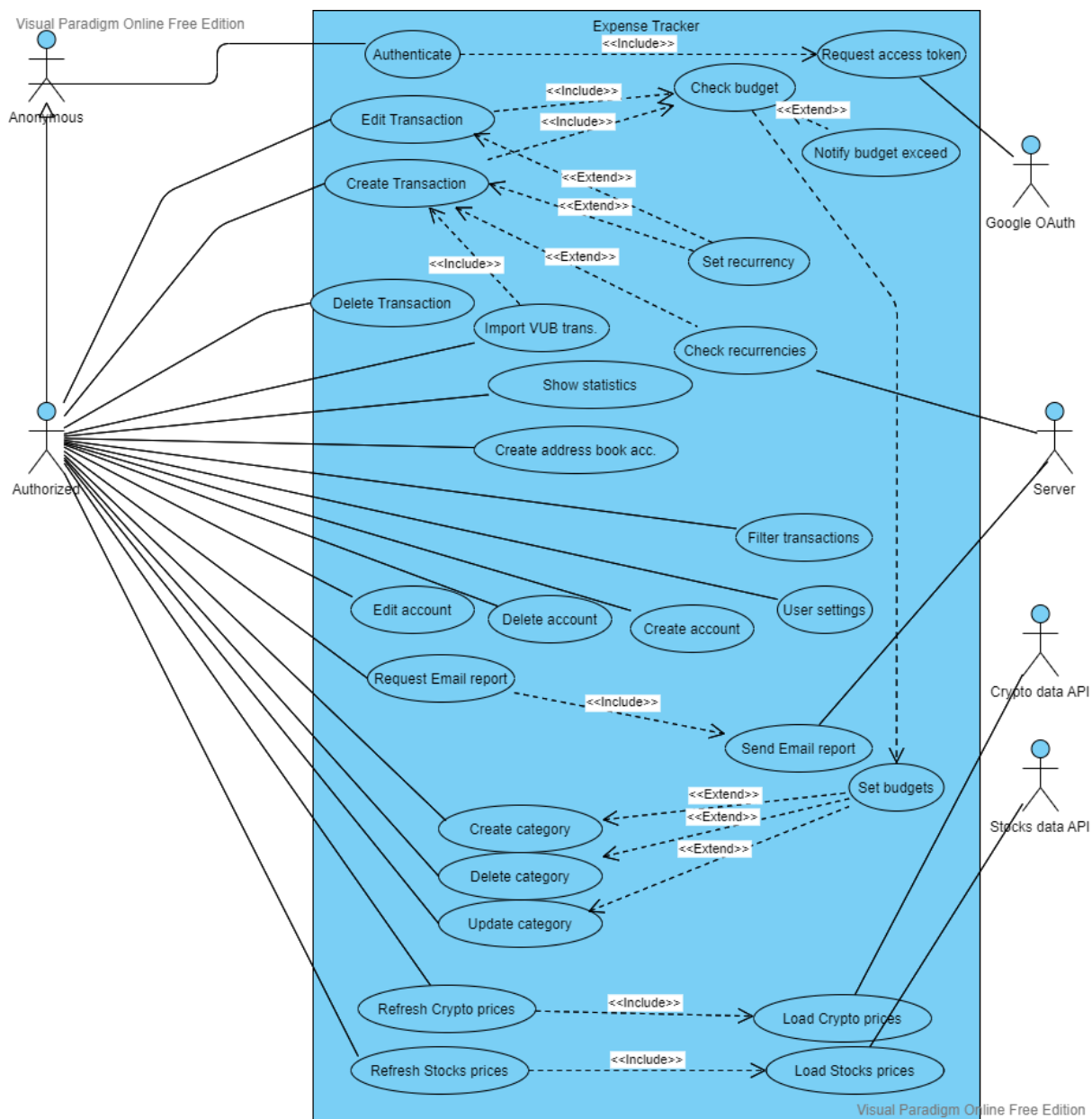
2. Typy používateľov

Používateľ, ktorý nie je prihlásený cez Google autentifikáciu nebude môcť využívať stránku. Po prihlásení sú všetci používatelia rovnocenní, čiže existuje iba jedna rola *Authorized user*. Obsah ktorý tento používateľ vytvorí môže vidieť a meniť iba on. Dostupné funkcionality sú popísané v sekcii 1.

3. Ostatné systémy

Pri zisťovaní hodnoty akcií sa budú brať údaje z REST API marketstack.com. Hodnoty kryptomien sa budú získať z coingecko.com. V prípade že by údaje boli nedostatočné, tak by som skúsil iné stránky, ktoré poskytujú API rozhranie. Používateľ bude mať call to action tlačítko na prepočítanie hodnoty akcií. Po kliknutí sa spustí asynchrónny job v súlade s limitami API a aktualizuje aktuálne ceny akcií a kryptomien pre používateľa. Pri prihlásení budem používať Google OAuth.

4. Use case diagram



5. Databáza

Databázu budem používať MySQL 8. MySQL má dobrú podporu ORM Eloquent a migrácií tabuliek z Laravelu. Eloquent zjednoduší a sprehľadní manipuláciu a získavanie údajov v databáze.

- FK *transaction_type_id* - ID typu transakcie
- *transaction_time* - Timestamp vykonania transakcie
- *value* - Množstvo peňazí použitých v transakcií
- FK *currency_id* - Mena použitá v transakcií
- FK *adress_book_id* - ID záznamu z adresára použitého v transakcií
- *repeating* - Názov spôsobu opakovania transakcie
- FK *source_account_id* - ID účtu z ktorého ubúdajú položky
- FK *end_account_id* - ID účtu na ktorý pribudnú položky

Tabuľka má Many-to-One vzťah s tabuľkou [Address Book](#), [Transaction Type](#), [Category](#)

6.3 Account

Obsahuje účty používateľa ako napríklad peňaženka, kreditná karta, bankový účet.

- FK *currency_id* - Hlavná mena účtu
- *name* - Názov účtu
- *value* - Predpočítaná hodnota účtu pomocou asynchrónneho jobu
- *icon* - Vybraná ikona účtu
- FK *user_id* - ID používateľa

Tabuľka má One-to-Many vzťah s tabuľkou [Account Item](#), a [Transaction](#).

6.4 Category

Obsahuje vytvorené kategórie používateľom.

- *name* - Názov kategórie
- *icon* - Ikona priradená ku kategórií
- FK *user_id* - ID používateľa, ktorý kategóriu vytvoril

Tabuľka má One-to-Many vzťah s tabuľkou [Budget](#)

6.5 Budget

Tabuľka obsahuje budgety nastavené používateľom a priradené ku kategórií.

- FK *user_id* - ID používateľa
- *budget_period* - Časový interval na ktorý sa bude budget kontrolovať pri vytvorení transakcie (týždeň, deň, mesiac, rok)
- *amount* - Hodnota rozpočtu, ktorá by sa nemal presiahnuť
- *start_time* - Čas vytvorenia pravidla budgetu
- FK *category_id* - ID kategórie na ktorú sa aplikuje budget

6.6 Account Item a Transaction Item

Tabuľky obsahujú vlastnené položky účtov alebo transakcií, ako sú akcie, kryptomeny.

- FK *account_id* - ID účtu, ktorý vlastní položku
- FK *transaction_id* - ID transakcie, ktorá vlastní položku
- FK *item_type_id* - ID z tabuľky ItemType, výber definovaného typu položky
- *metadata* - JSON pole uchovávajúce údaje položky: predajnú cenu, aktuálnu cenu, množstvo, poplatok
- *name* - názov alebo ticker

- quantity - množstvo
- currency_id - mena v ktorej sa položka primárne obchoduje

Transaction Item obsahuje price - predajná alebo nákupná cena, fees - poplatky, "fees_currency_id" - mena použitá pri poplatku.

Account Item obsahuje "average_buy_price" - priemerná nákupná cena, "current_price" - aktuálna cena položky získaná z API

Tabuľka má Many-to-One vzťah s tabuľkou [ItemType](#)

6.7 Item Type

Tabuľka obsahuje definované typy položiek (akcie, kryptomeny) ktoré sú implementované na Backende.

6.8 Address Book

Adresár osôb alebo objektov. Obsahuje ID, názov a IBAN číslo. Položky z adresára je možné vyberať pri transakciách.

6.9 Transaction Type

Typy transakcií ktoré sú implementované na backende a to výdaj, príjem, prevod, nákup a predaj. Typ transakcie sa vyberá pri vytváraní záznamu transakcie. Tabuľka obsahuje iba názov.

6.11 File

Obsahuje súbory priradené k transakciám. Má polia: FK "transaction_id", name - vygenerovaný názov súboru a "original_name" je pôvodný názov súboru.

6.11 Reports

Obsahuje pravidlá posielania e-mailových reportov. V tabuľke sa vytvára záznam v prípade že si používateľ v nastavení účtu nastaví zasielanie e-mailov. Má polia: FK PK user_id, sending_period. "sending_period" môže mať hodnotu: mesačné, ročné, týždenne. "created_at" je timestamp kedy bolo pravidlo vytvorené.

6.10 Jobs a Failed_Jobs

Samostatné tabuľky, ktoré vytvorí Laravel pri vykonávaní Queued Job-ov prostredníctvom databázy.

- id - ID jobu
- queue - Názov fronty
- payload - Objekt, ktorý bude skonsumovaný pri vykonaní úlohy
- attempts - Počet pokusov na vykonanie úlohy
- reserved_at, available_at, created_at - Timestamp údaje

6.11 Currency

Názvy často používaných mien. Má OneToMany vzťah s Transaction, [User](#), [TransactionItem](#) a [AccountItem](#)

7. Technologické požiadavky

Aplikácia bude mať na backende PHP framework Laravel s knižnicou Laravel Livewire ktorá je určená na vytváranie moderných, reaktívnych, dynamických rozhraní využívaním Laravel Blade čo je technológia template viewov. V kombinácii s Livewire budem používať Javascriptovú knižnicu Alpine.js. Na štylovanie frontendu budem používať CSS framework Tailwind. Databáza bude použitá MySQL 8 prostredníctvom ORM Laravel Eloquent. Prehliadače budú plne podporované všetky okrem Internet Exploreru a Safari. Aplikácia je primárne určená pre desktopové zariadenia.

8. Časový plán

Týždeň 5. (15.3 - 21.3)

- Vytvorenie základnej kostry projektu a nastavenie vývojového prostredia - 2h
- Vytvorenie modelov, ciest, tried - 4h
- Vytvorenie migrácii tabuliek s príslušnými poľami a ich importovanie do databázy - 4h
- Prihlasovanie pomocou OAuth, zistenie fungovania, implementácia - 2h
- Ukladanie prihlásenia po ukončení stránky - 2h

Týždeň 6. (22.3 - 29.3)

- Metódy na vytváranie transakcií, účtov, adresára a ukladanie nastavení - 6 h
- Vytvorenie frontendu pre podstránky Účty, Nastavenia, Kategórie, Transakcie - 6h
- Prihlasovanie pomocou OAuth, zistenie fungovania, implementácia - 2h
- Ukladanie prihlásenia po ukončení stránky - 2h

Týždeň 7. (30.3 - 4.4)

- Implementácia typov položiek Akcia, Kryptomena - 4 h
- Implementácia typov transakcií Výdaj, Prijem, Prevod, Nákup, Dlžoba - 6 h
- Frontend na pridávanie položiek k účtom a transakciám - 2h
- Stránka so štatistikami - 3h
- Filtrovanie transakcií - 2 h

Týždeň 8. (5.4 - 11.4)

- Úprava a vymazanie záznamov transakcií, účtov, kategórií - 3h
- Implementácia pridávania položiek k účtom a transakciám spolu s podporou kombinácii mien - 3 h
- Nastavenie budgetov pri kategóriách a ich kontrola pri vytváraní transakcií - 3h
- Nahrávanie súborov pri vytváraní transakcie - 1h

Týždeň 9. (12.4 - 18.4)

- Odhlasovanie sa - 2h
- Nastavenie preferovaného zobrazovania času a následná aplikácia na stránkach - 1h

- Opakujúce sa transakcie - 3h
- Import excelu z VÚB exportu a vytvorenie transakcií k jednotlivým položkám - 3h
- Vyladenie Frontendu - 1h

Týždeň 10. (19.4 - 25.4)

- Implementácia načítania údajov o akciách z API - 3h
- Implementácia načítania údajov o kryptomenách z API - 3h
- Implementácia spustenia asynchrónneho procesu na získanie údajov po kliknutí používateľom - 1 h
- Odosielanie e-mailových reportov - 2h

Týždeň 11. (26.4 - 2.5)

- Dokončovanie - 5 hodín
- Testovanie - 6 hodín

Spolu by to malo byť 87 hodín na dokončenie všetkých zaumienených funkcionalít.