
Podrobný návrh

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO

BRATISLAVA, 2016–2017

PORTÁL PRE VERNOSTNÉ PROGRAMY

VYPRACOVALI

LADISLAV BILISICS
MARTIN BOHUMEL
MARTIN KELLNER
IVAN LATTÁK

Obsah

1	Úvod	3
2	Analýza používateľov aplikácie	4
2.1	Používatelia	4
2.2	Use-case diagram	5
3	Generované dáta	6
4	Testovanie scenáre	7
5	Model stavového diagramu	8
6	Analýza používateľského rozhrania aplikácie pre používa- teľa	9
6.1	Vzhľad grafu	9
7	Triedny diagram	12
8	Sekvenčný diagram	13
9	Diagram toku dát	14
10	Data model	15
11	Entitno-relačný diagram	16
12	Object design	17
13	Komponenty aplikácie	19
13.1	Diagram rozmiestnenia komponentov	19
13.1.1	Používateľský počítač	19
13.1.2	Web server	20
13.1.3	Databázový server	20

Kapitola 1

Úvod

Tento dokument obsahuje podrobný popis a komplexný návrh webovej aplikácie pre vernostné programy, ktorá umožňuje jej používateľom prezerat' svoje zisky z ich synov ktorých do hierarchie pozvali. Dokument obsahuje:

- Popis modelu stavového diagramu
- Analýzu používateľov aplikácie a use-case diagram
- Analýzu generovaných dát pre účel rôznych testov
- Popis a zhrnutie testovacích scenárov
- Popis stavového diagramu
- Analýza používateľského rozhrania aplikácie pre používateľa a vzhľad zobrazeného grafu
- Popis triedneho diagramu
- Popis data model diagramu
- Presný opis funkcionality - Object design
- Popis entitno-relačného diagramu a komponenty aplikácie

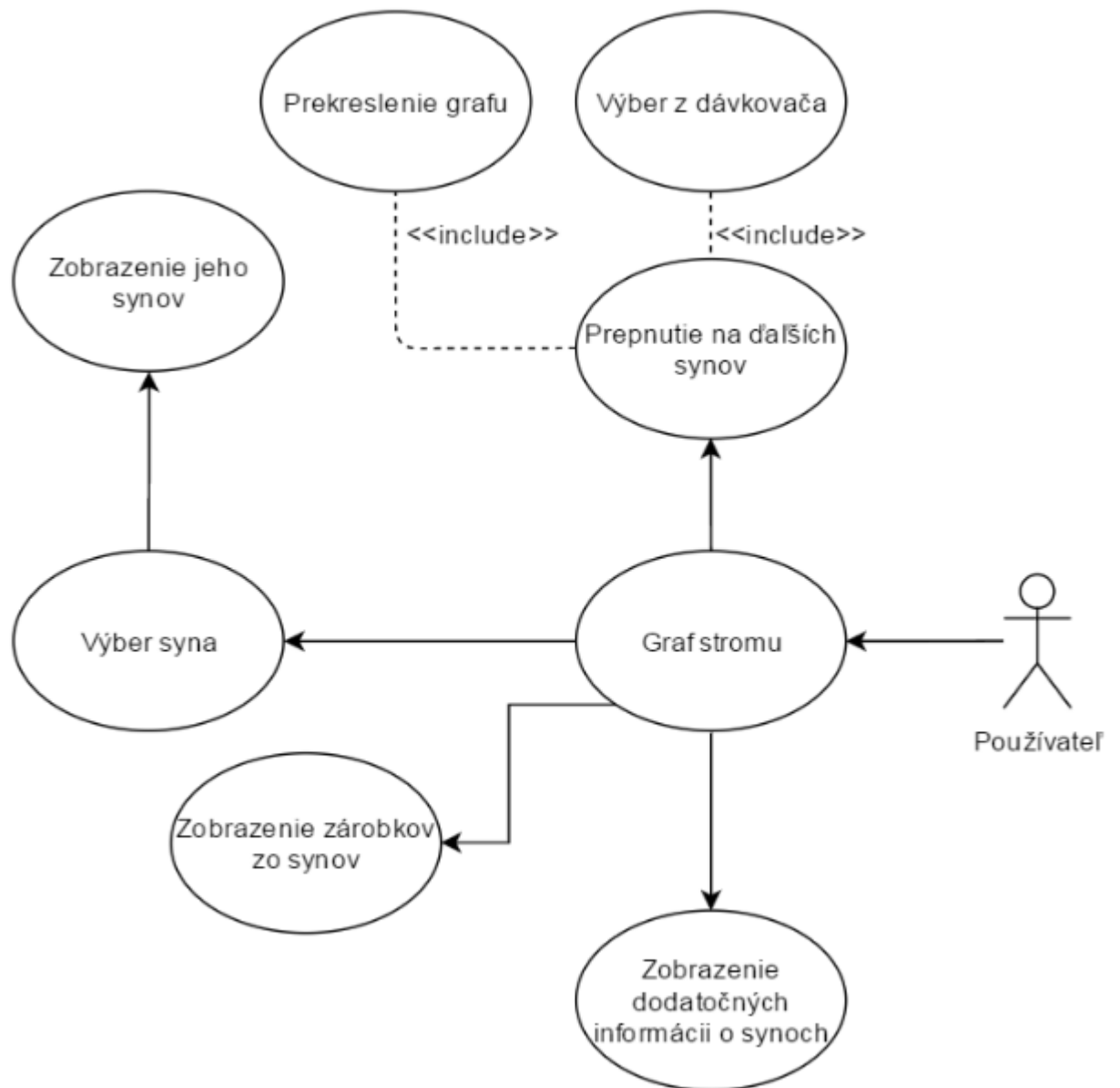
Kapitola 2

Analýza používateľov aplikácie

2.1 Používatelia

Používateľmi aplikácie sú podnikatelia a zákazníci, ktorí pozývajú ďalších používateľov. Majú možnosť si prezerať svoje zisky z používateľov ktorých pozvali.

2.2 Use-case diagram



Používateľ vie pomocou grafového zobrazenia zistiť zárobky z používateľov spolu s dodatočnými dátami (ich program, dátum kedy sa pridali, kedy zaplatili svoj program a na akú dobu), ktorí sú zaregistrovaní vďaka nemu(jeho synovia a ich synovia rekurzívne). Sú zobrazení po desiatich (potenciálne prenastaviteľný počet), pričom pri kliknutí pravým tlačítkom myši sa prepne na ďalších desať. Vie prepínať medzi nimi, aby videl ich synov a svoj zárobok z nich.

Kapitola 3

Generované dáta

Budeme náhodne generovať okolo 40 000 jedincov v strome, ktorí budú zoradení do stromovej štruktúry. Údaje budú meno, zárobok, id a id otca. Jedinci budú rovnomerne rozmiestnení do rôznorodých stromov s rôznymi šírkami a hĺbkami.

Kapitola 4

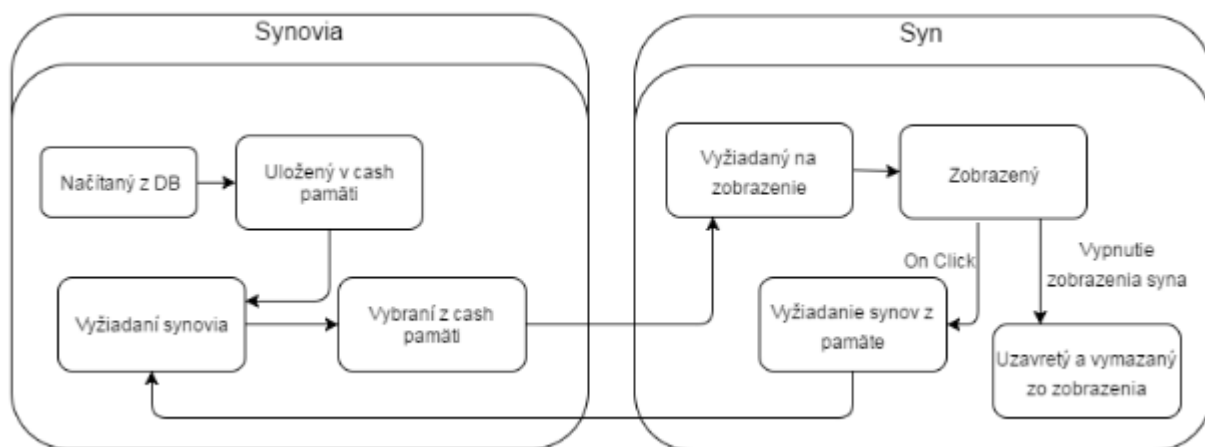
Testovanie scenáre

Testovanie bude prebiehať nad nami generovaných dátach. Ďalej bude testovaná rýchlosť vykreslovania veľkého počtu prvkov a ich zobrazenia a rozmiestnenia na stránke a prekreslenia celého grafu nanovo pri prepnutí na ďalších synov. V rámci testovania budeme musieť sledovať či náš rekurzívny dopyt naozaj vráti každému synovi správne vypočítanú sumu zárobkov jeho synov.

Kapitola 5

Model stavového diagramu

Stavový diagram znázorňuje: Najprv sa podľa otca načítajú len jeho synovia z databázy do pamäte a jediný vykreslený je samotný otec. Následne po kliknutí na otca sa zavolá funkcia, ktorá dávkuje jeho synov. Zobrazí najprv prvých desať a pri prepnutí sa uloží stav otvoreného otca a vygeneruje sa nanovo strom s ďalšími jeho desiatimi synami, ak ich má. V prípade že nie, prepne sa na prvých desať synov. Po kliknutí na jedinca ktorý už má roztvorených synov sa musí zavolať ich deštrukcia spolu s deštrukciou ich synov do hĺbky.

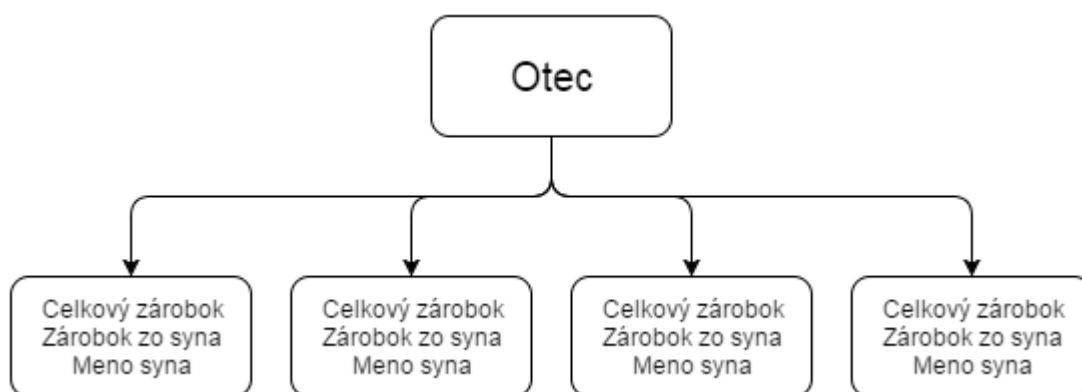


Kapitola 6

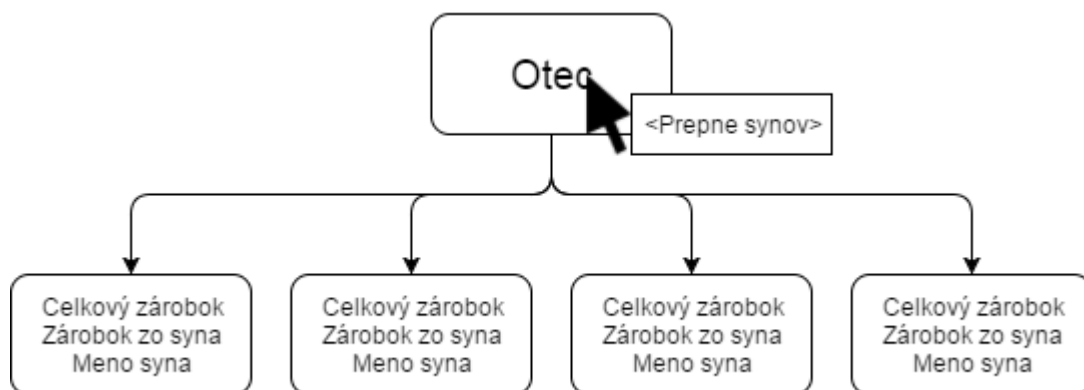
Analýza používateľského rozhrania aplikácie pre používateľa

Táto časť definuje, ako aplikácia vyzerá z pohľadu používateľa. Taktiež neexistuje administrátor ani náhľad pre osobu ktorá nie je zaregistrovaný používateľ.

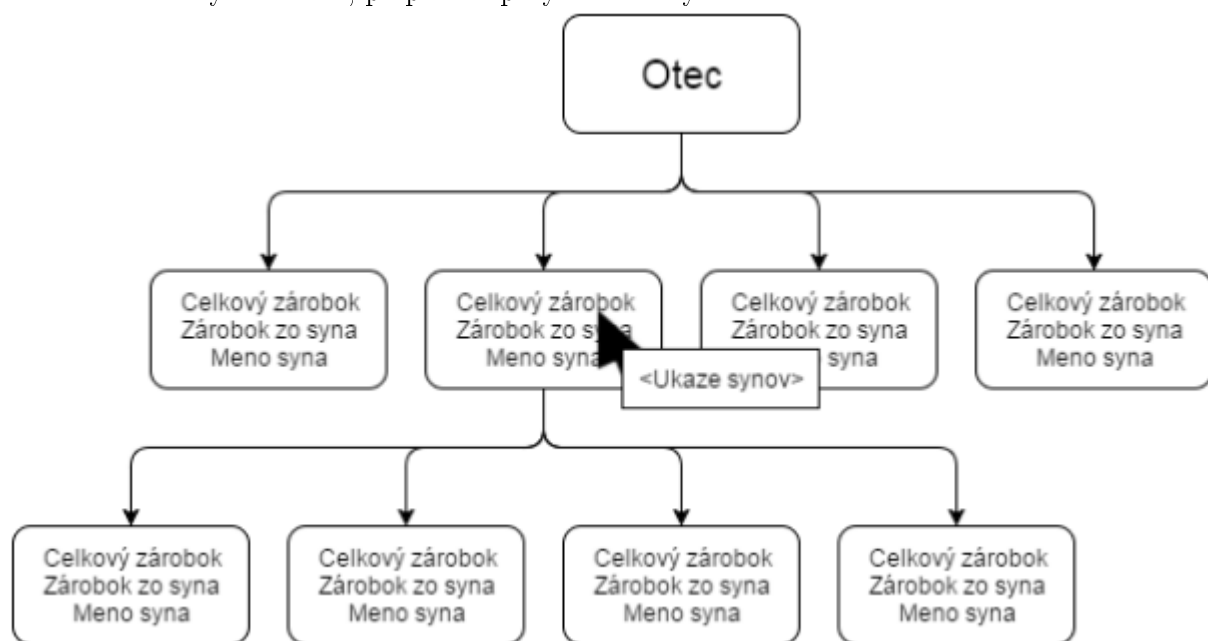
6.1 Vzhľad grafu



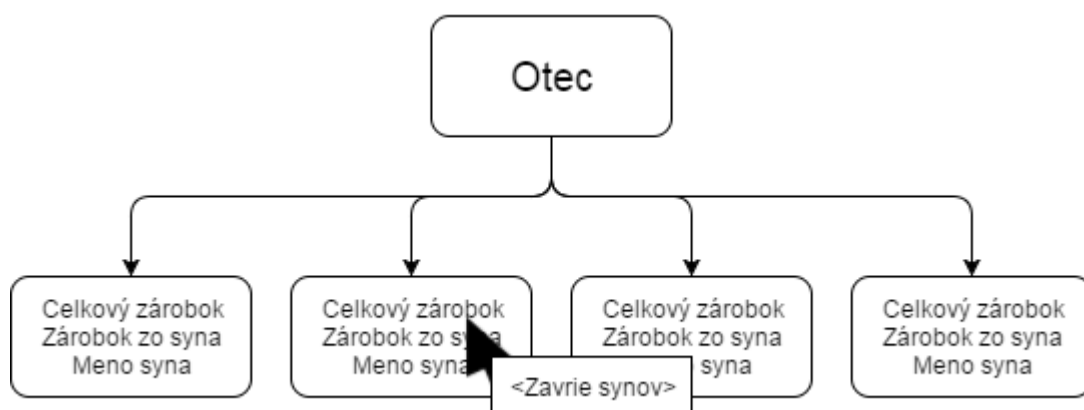
Základný vzhľad grafu je otec (prihlásený používateľ), ktorý prezerá svojich synov a celkový zárobok (z neho a jeho synov a ich synov rekurzívne) ale aj zárobok z jednotlivého syna.



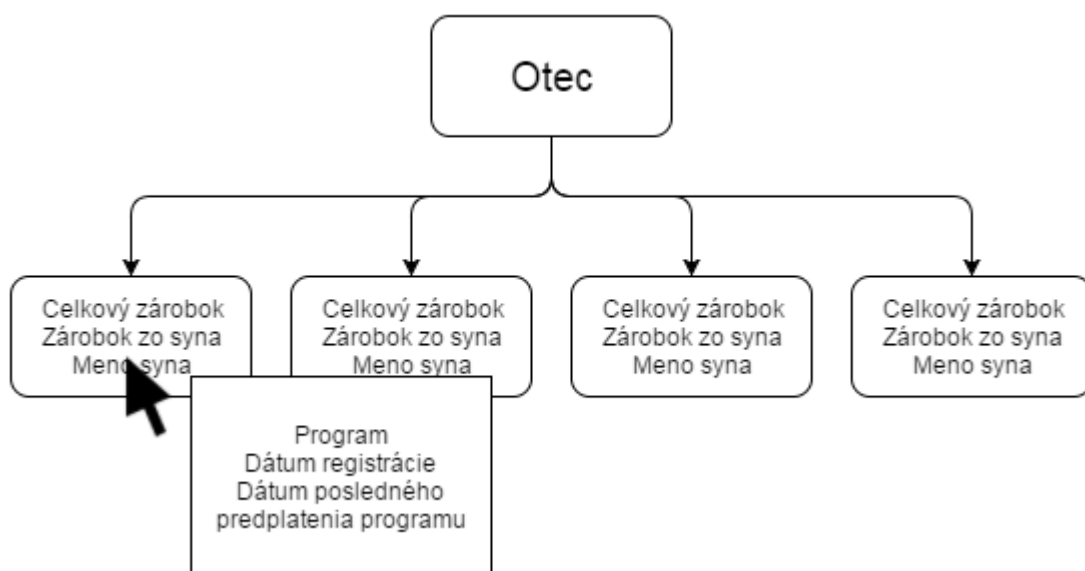
Pri pravom kliknutí na otca alebo syna (ktorý je otcom svojich synov) sa prepne na ďalších desať (alebo vopred nastavený počet) synov, ak ich má. Ak už ďalších synov nemá, prepne na prvých desať synov.



Po ľavom kliknutí na otca sa zobrazia jeho synovia a identické údaje ako v prvom grafe.



Keď používateľ klikne ľavým tlačíkom myši na už otvoreného syna, zavrie sa zobrazenie všetkých jeho synov a ich synov rekurzívne.

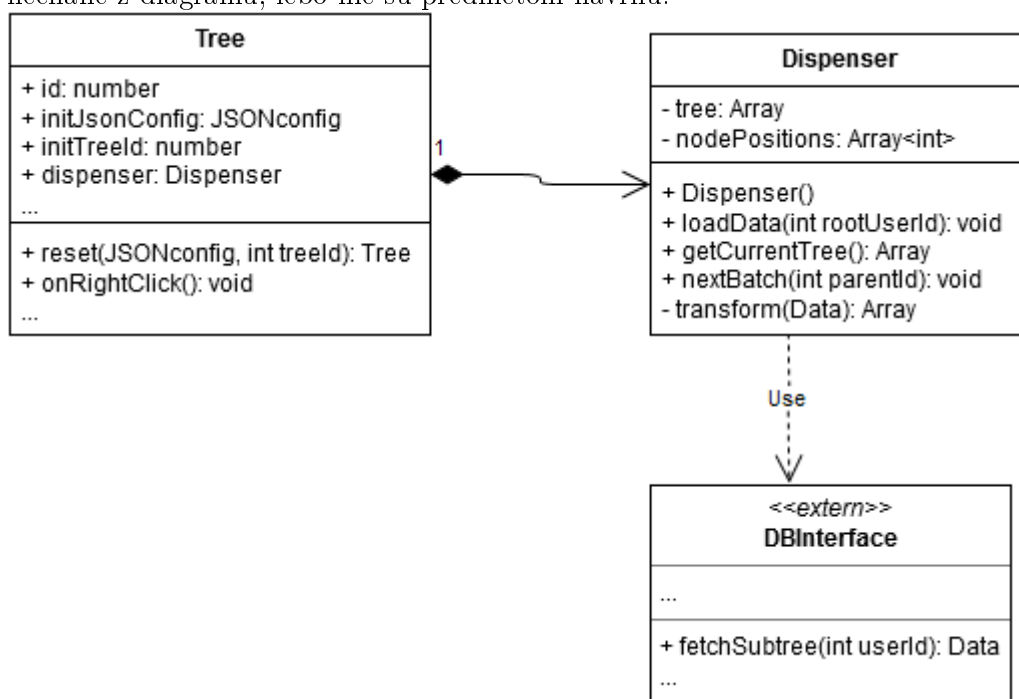


Pri prejdení myšou nad syna sa zobrazí dočasné okno s ďalšími podrobnosťami ohľadom syna.

Kapitola 7

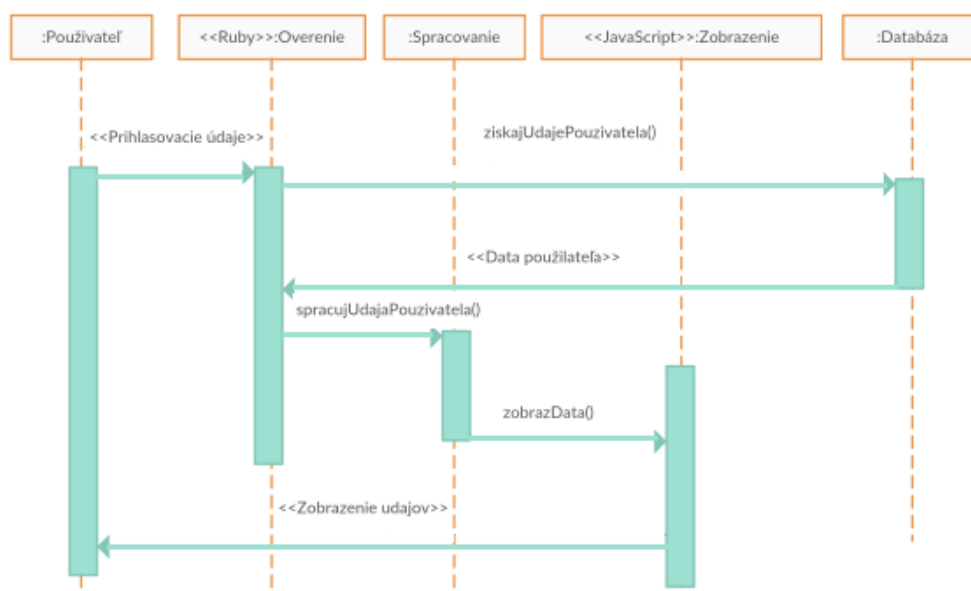
Triedny diagram

Tento triedny diagram zobrazuje konkrétne triedy, ktoré budú implementované v našej aplikácii. Obsahuje všetky potrebné premenné a metódy potrebné na funkcionality aplikácie. Niektoré triedne polia a metódy boli vynechané z diagramu, lebo nie sú predmetom návrhu.



Kapitola 8

Sekvenčný diagram



Sekvenčný diagram zobrazuje sekvenciu krokov pri prihlásení. Prihlasovacie údaje su poslané a následne overené. Používateľovi sa postupne pošle spracovaná odpoveď vo vizuálnej forme (webová stránka).

Kapitola 9

Diagram toku dát

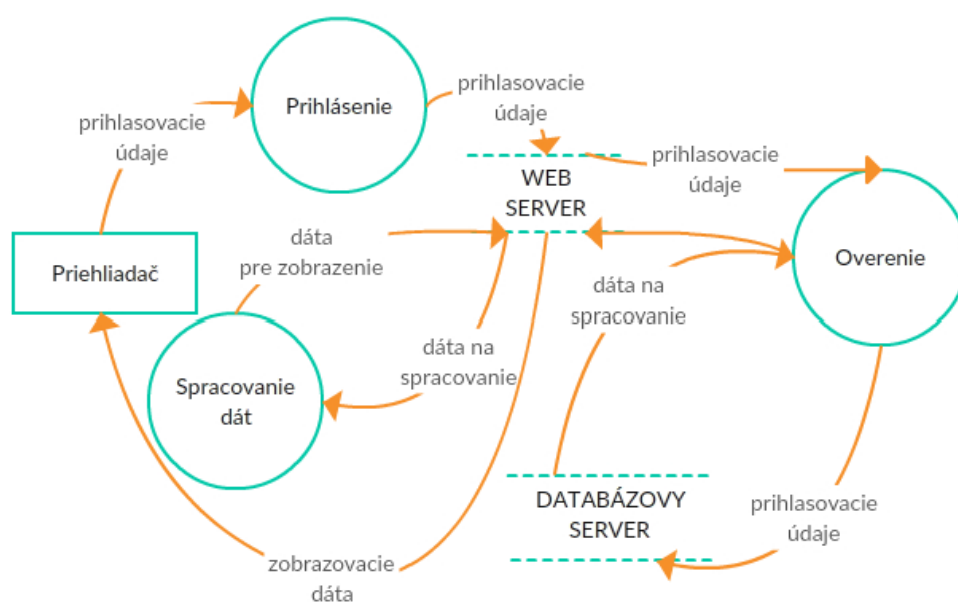
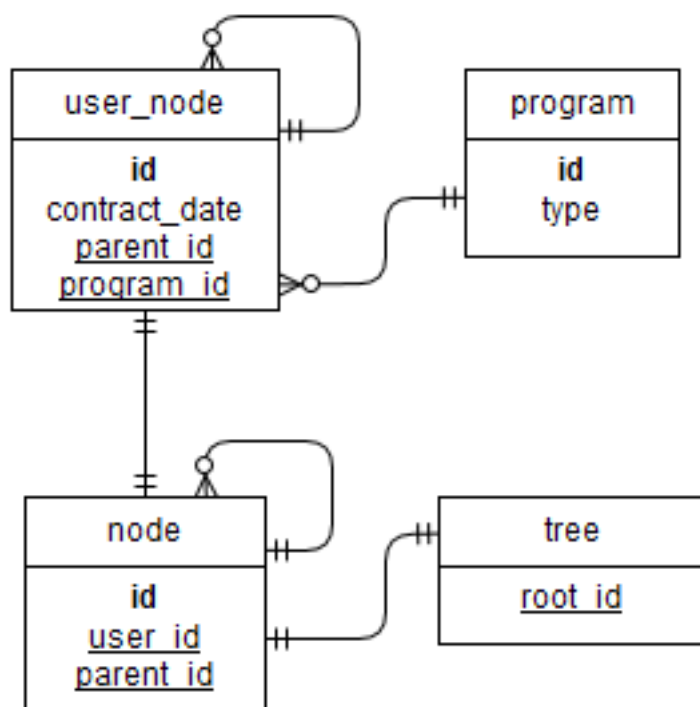


Diagram toku dát zobrazuje tok dát pri prihlásení a následnom vykreslení stromu. Používateľové prihlasovacie údaje spracováva proces prihlásenie. Tento proces reprezentuje funkcionalitu, ktorá na webový server zašle údaje, ktoré sa pošlu na databázový server, kde sa zistí či je používateľ registrovaný, teda či prihlasovacie údaje zodpovedajú konkrétnej osobe v databáze. Overenie naspäť posíla dáta na spracovanie. Tie sa rozpoznávajú, resp. spracovávajú procesom spracovanie. Spracovanie dáta spracuje na zobrazovacie dáta. Tie sa posielajú používateľovi vo vizuálnej forme.

Kapitola 10

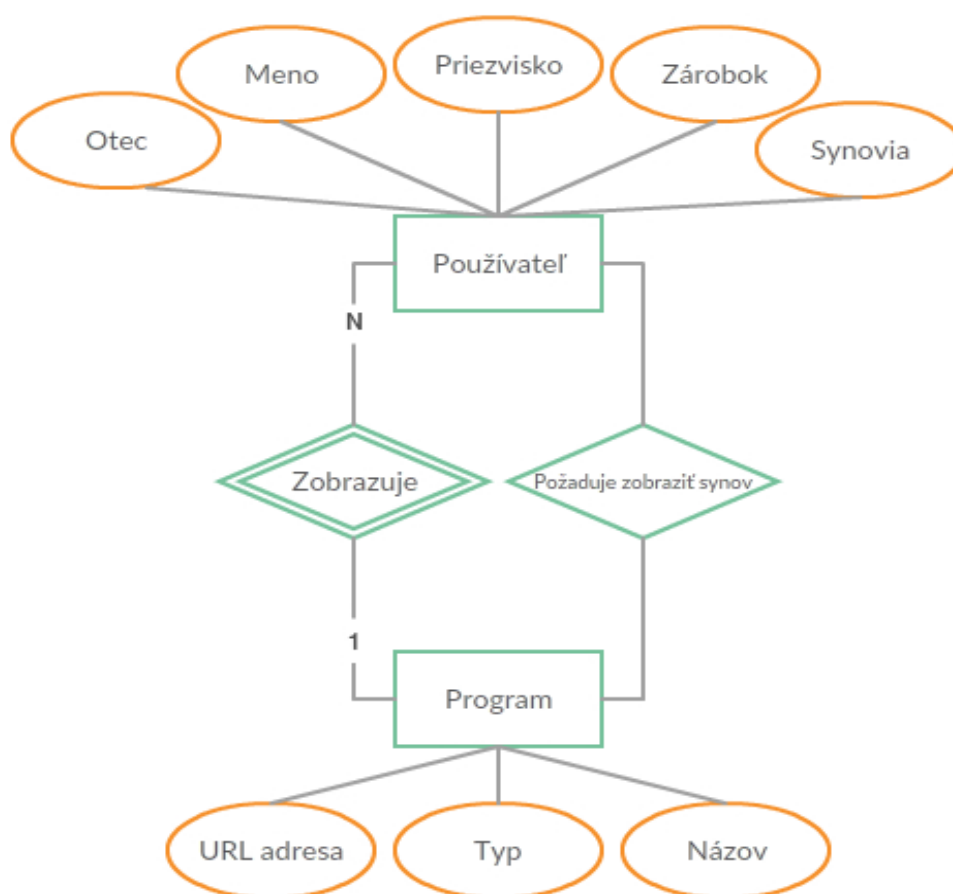
Data model

Dátový model popisuje stromovú štruktúru vrcholov s používateľmi s programami. Podčiarknutý text znamená kľúč, index, v tabuľke.



Kapitola 11

Entitno-relačný diagram



Entito - relačný diagram nášho projektu vyjadruje vzťah medzi dvoma entitami. Entita používateľ predstavuje reálnu osobu. Entita Program je naše riešenie problému zobrazovania stromovej štruktúry používateľov. Používateľ žiada zobrazit' iných používateľov, ktorými je vo vzťahu otec - syn. Program vracia výsledok vo forme zobrazenia.

Kapitola 12

Object design

Funkcie mimo tried:

- `loadFromDB(): Data`
 - Načíta údaje z databázy.
- `transform(): Array`
 - Transformuje načítané údaje do poľa.

Dávkovacia trieda `Dispenser`:

- `getCurrentTree(): Array`
 - Vrátí aktuálne obsiahnutú stromovú štruktúru ako pole vrcholov, vhodné na vygenerovanie stromu v triede `Tree`.
- `getCurrentTree(): Array`
 - Vrátí aktuálne obsiahnutú stromovú štruktúru ako pole vrcholov, vhodné na vygenerovanie stromu v triede `Tree`.
- `nextBatch(int parentId): void`
 - Aktuálnych synov vrcholu `parentId` nahradí novými, maximálne desiatimi synmi (novou dávkou synov) tohto vrcholu. Po minutí všetkých možných synov vrcholu, ďalšie volanie tejto funkcie znova generuje prvú dávku.
- `loadData(): array`
 - využije externé funkcie `loadFromDB()` a `transform()` pomocou ktorých načíta do triednej premennej zoznam tak, aby sa s ním dalo pracovať, teda do formátu pre `Treant`.

`Tree`: Trieda `Tree` je trieda knižnice `Treant.js`, ktorá bude doplnená o jedno triedne pole a jednu metódu, aby lepšie spĺňala požiadavky zadania:

- dispenser: Dispenser
 - Pole bude nastavené konštruktore triedy. Obsahuje dávkovací objekt, ktorý abstrahuje od vyberania používateľských dát z databázy a podáva synov ľubovoľného vrchola po dávkach maximálne desiatich synov naraz.
- onRightClick(): void
 - Callback funkcia eventu onRightClick". Synovia otca, na ktorého bolo kliknuté, sa nahradia za novú dávku, maximálne desiatich synov, a následne sa strom prekreslí.

Kapitola 13

Komponenty aplikácie

13.1 Diagram rozmiestnenia komponentov

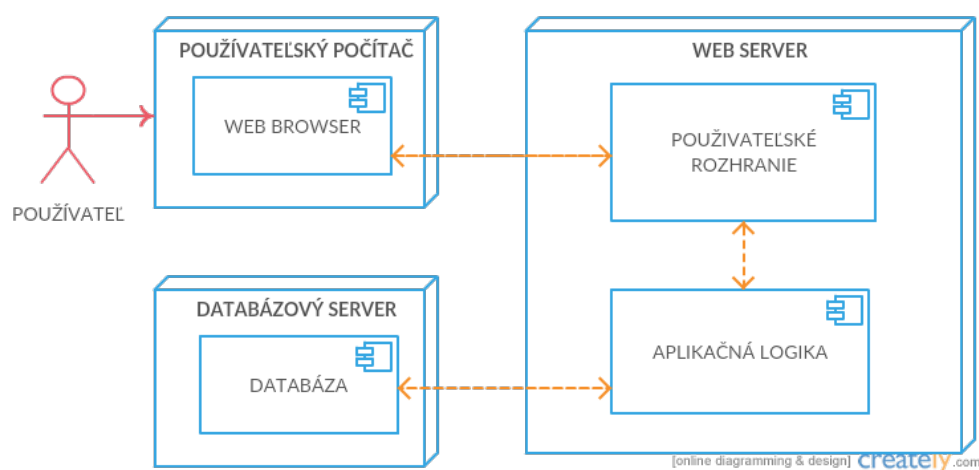


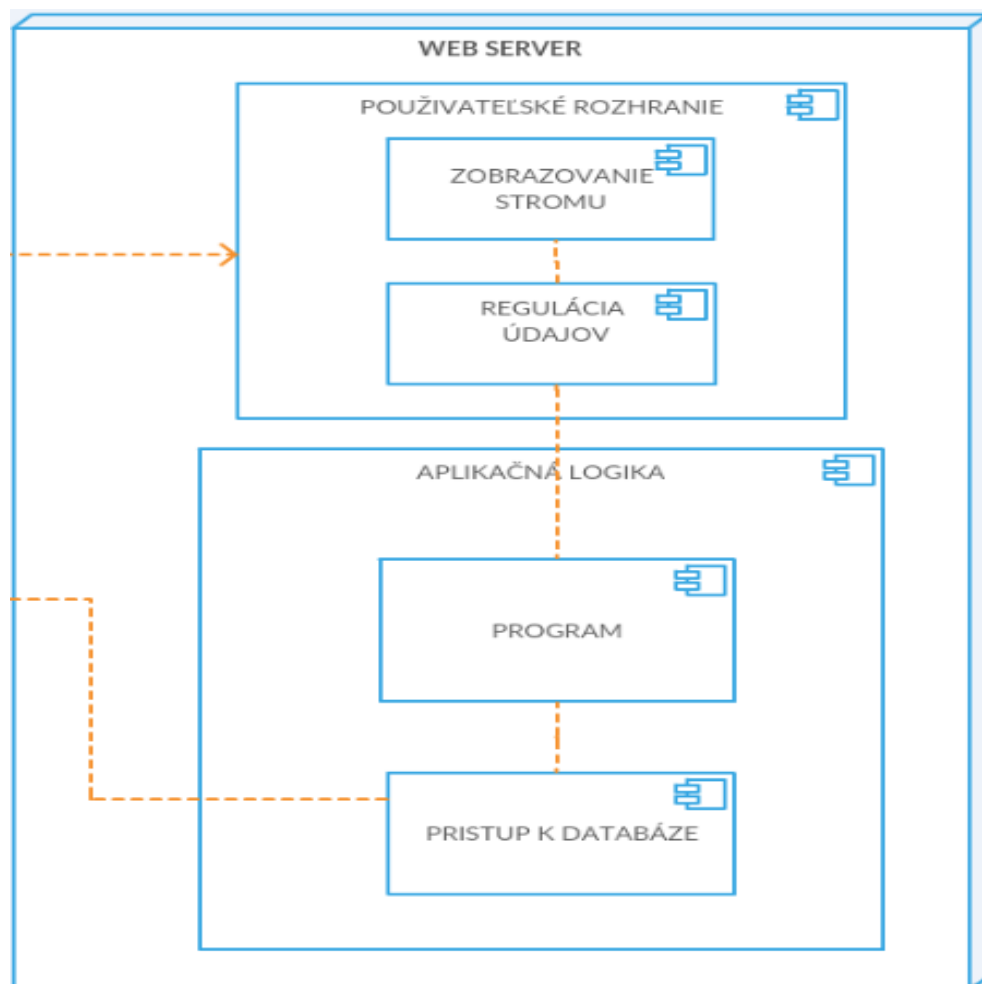
Diagram rozmiestnenia komponentov zobrazuje rozloženie komponentov na jednotlivých výpočtových uzloch počas reálnej prevádzky systému. V tomto riešení sa nachádzajú tri výpočtové uzly:

- Používateľský počítač,
- Web server,
- Databázový server

13.1.1 Používateľský počítač

Jeden z výpočtových uzol je používateľský počítač, používateľ má nainštalovaný webový prehliadač cez ktorý prístupuje k aplikácii na webovom servery.

13.1.2 Web server



Webový server predstavuje ďalší vypočtový uzol, obsahuje používateľské rozhranie (frontend), ktoré obsahuje zobrazovanie používateľských dát v stromovej štruktúre. Obsahuje aj nástroje na reguláciu dát, teda čo sa ma zobrazovať a v akom množstve.

Dalej obsahuje aplikačnú logiku (backend). To je časť programu ktorá obsahuje program, ktorý spracováva požiadavky používateľa a obsahuje aj program pre prístup k databáze, tento prístup k databáze my priamo neimplementujeme, bude nám dodaný zo strany zadavateľa.

13.1.3 Databázový server

Databázový server obsahuje schému databázy s používateľskými dátami, táto časť rovnako ako prístup k databáze neimplementujeme, bude dodaná zo strany zadavateľa.