

# KVANTOVO-CHEMICKÉ VÝPOČTY

## Návrh

*Jaroslav Ištok*

*Katarína Fabianová*

*Dušan Suja*

*Jerguš Adamec*

November 13, 2016

# Contents

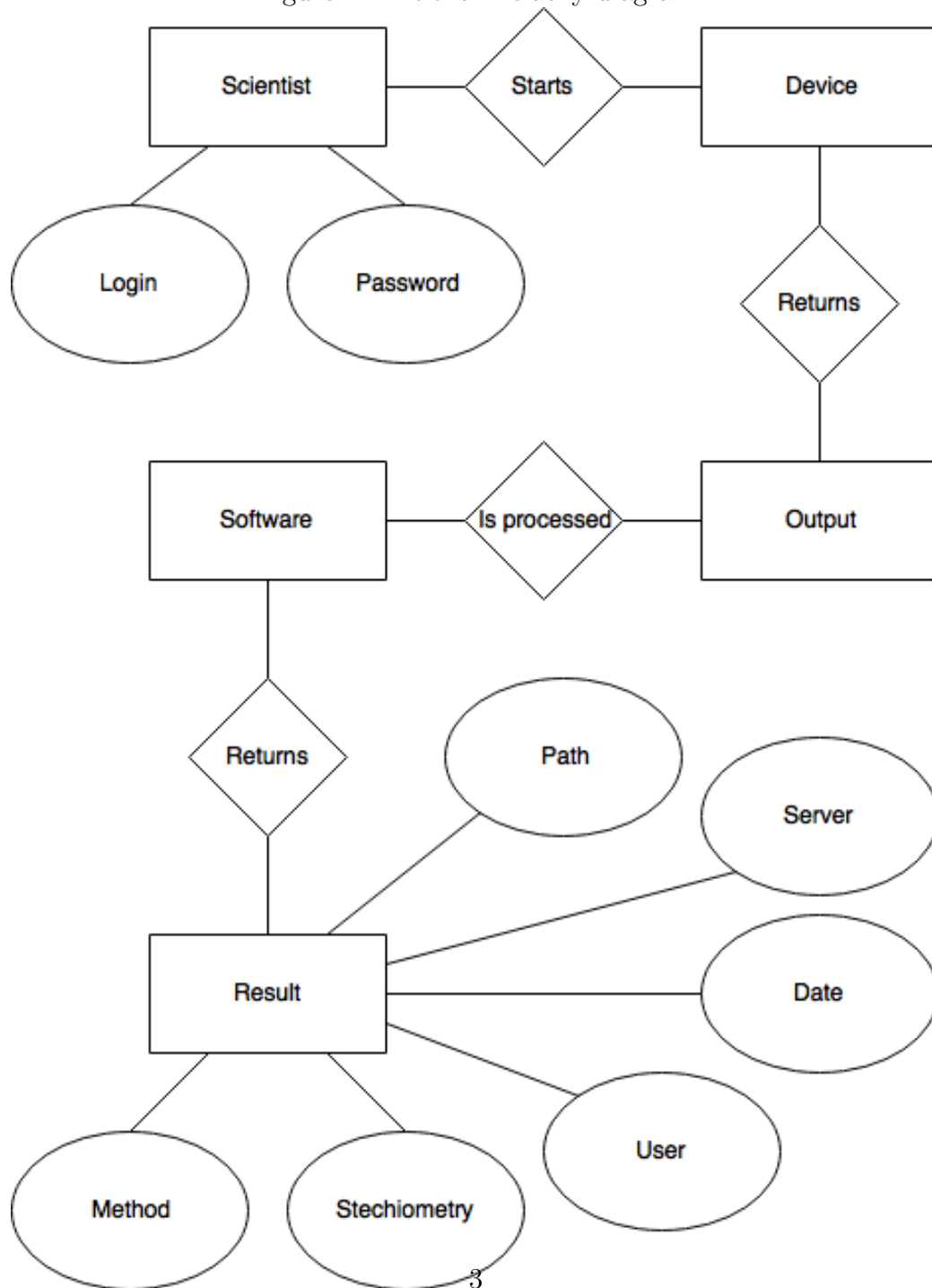
<b>1</b>	<b>Diagramy</b>	<b>3</b>
1.1	Entitno-relačný diagram . . . . .	3
1.2	Use case diagram . . . . .	6
1.3	Stavový diagramy . . . . .	9
1.4	Sekvenčný diagram - spracovanie súboru . . . . .	10
1.5	Sekvenčný diagram - pripojenie k databáze . . . . .	11
1.6	Triedny diagram . . . . .	12
1.7	Dátový model databázy . . . . .	13
1.8	Dekompozícia . . . . .	14
<b>2</b>	<b>Návrh používateľského rozhrania rozhranie</b>	<b>15</b>
<b>3</b>	<b>Analýza technológii</b>	<b>17</b>
3.1	Výber programovacieho jazyka . . . . .	17
3.2	Výber databázového systému . . . . .	18
<b>4</b>	<b>Testovacie scenáre</b>	<b>18</b>
4.1	Existencia súboru . . . . .	18
4.2	Crawler . . . . .	18
4.3	Validnosť súboru . . . . .	18
4.4	Rozlišovanie linuxových a windowsových súborov . . . . .	19
4.5	Databáza . . . . .	19



# 1 Diagramy

## 1.1 Entitno-relačný diagram

Figure 1: Entitno - relačný diagram

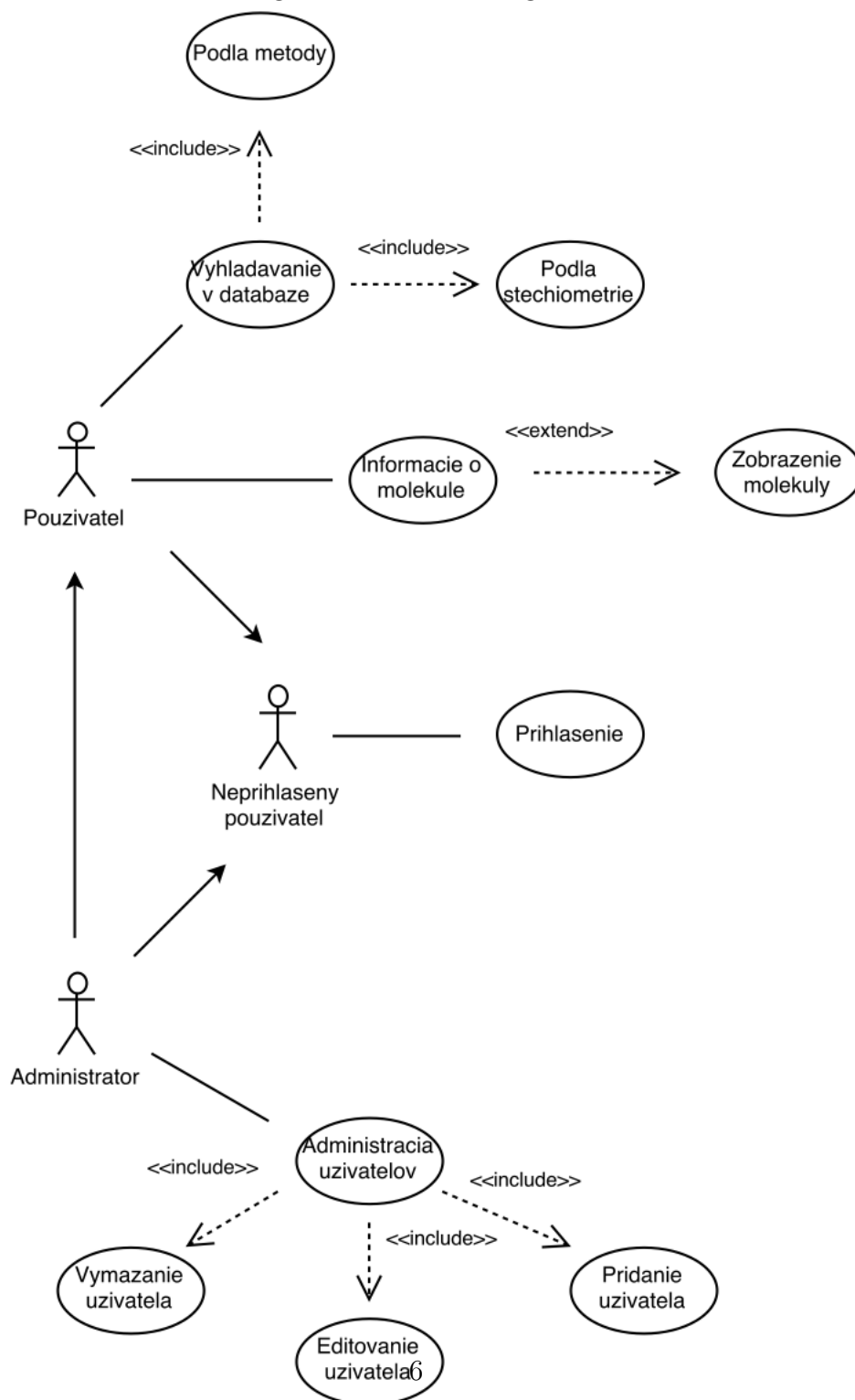


1 Entitno-relačný diagram znázorňuje vzťahy (relácie) medzi entitami. Diagram je použitý na modelovanie priestoru domény, pre ktorú sa informačný systém vyvíja (ústav experimentálnej fyziky). Entity sú zakreslené do obdĺžnikov. Vzťahy (relácie) medzi entitami sú v kosoštvorcoch, sú prepojené so všetkými entitami, ktoré do daného vzťahu vstupujú a sú pomenované. Entity majú svoje atribúty, ktoré sú do diagramu zakreslené ako ovály spojené so svojou entitou úsečkou.



## 1.2 Use case diagram

Figure 2: Use case diagram



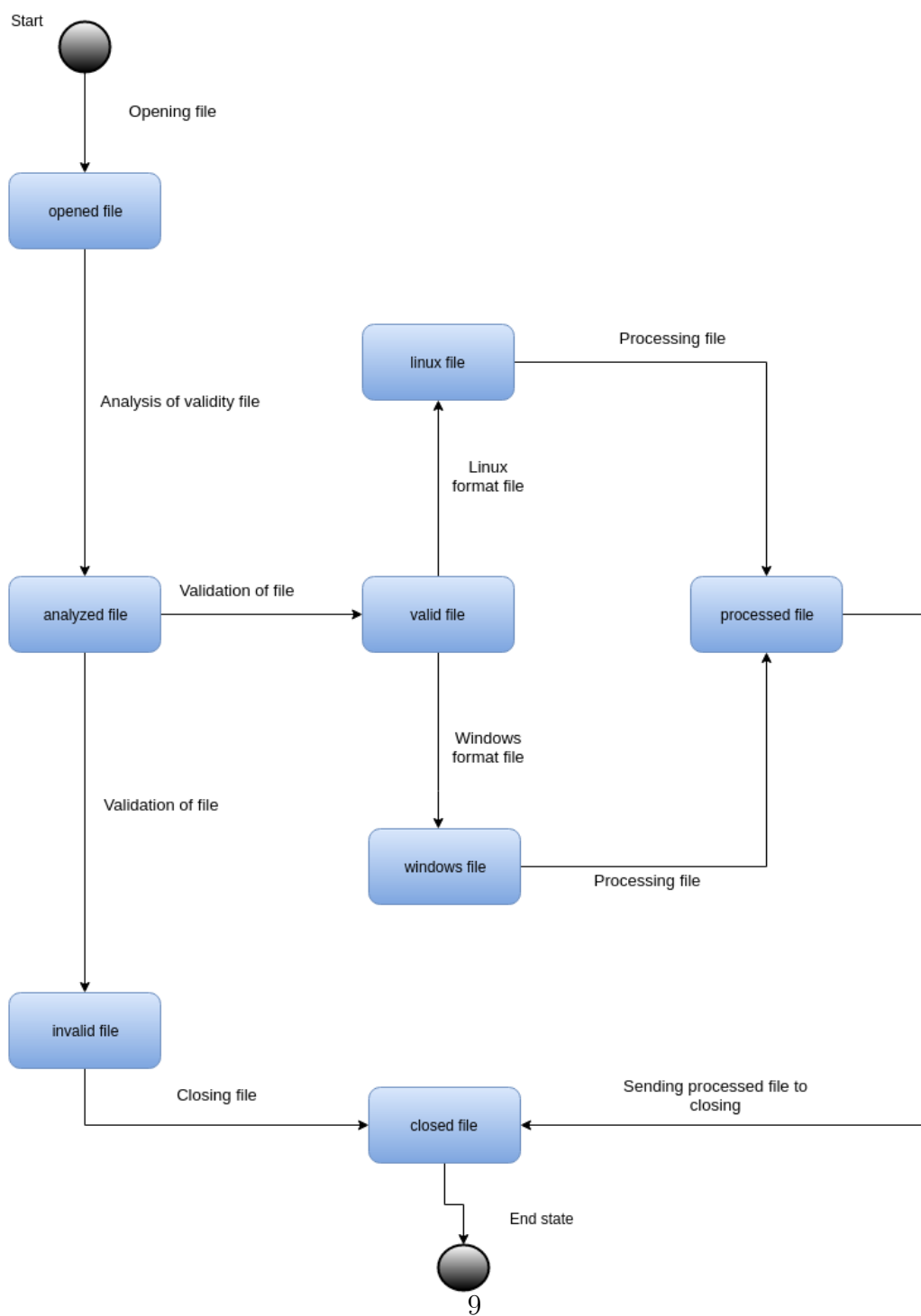
2 Use case diagram popisuje interakciu používateľov (aktorov) s aplikáciou. Neprihlásený používateľ sa môže prihlásiť. Prihlásený používateľ môže robiť administratívne úkony ako napríklad: vymazávať, editovať alebo pridávať nových používateľov. Ďalej môže vyhľadávať v databáze výsledkov meraní podľa rôznych kritérií, či zobrazíť výpis podrobných informácií o danej molekule a tiež zobrazíť jej náhľad.





### 1.3 Stavový diagramy

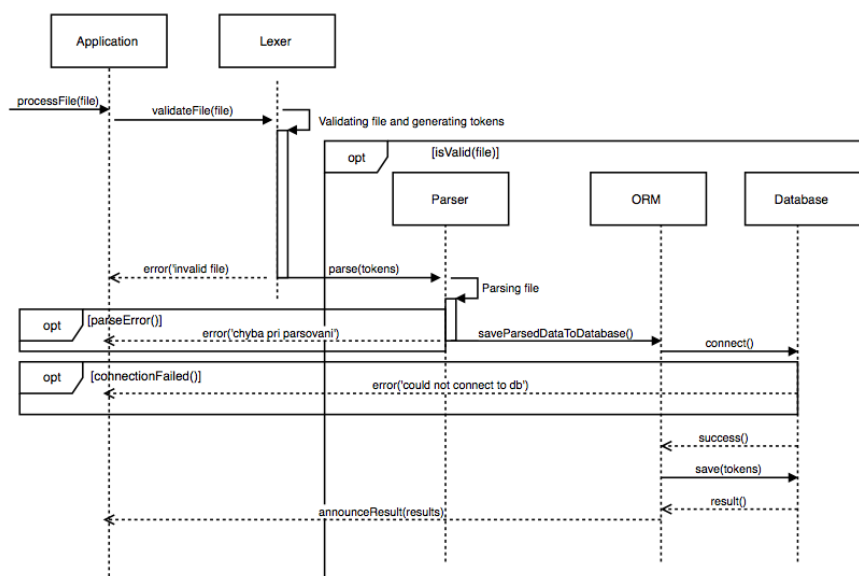
Figure 3: Stavový diagram  
**Stavový diagram**



3 Stavový diagram popisuje proces spracovania súboru aplikáciou. Spracovanie sa začína otvorením súboru a jeho následnou prvotnou analýzou (validáciou). V prípade, že súbor nie je validný, tak sa nepokračuje ďalej v jeho spracovaní. V prípade, že súbor je validný sa zistí, či má súbor windowsový alebo linuxový formát. Tieto 2 formáty sa líšia vnútornou štruktúrou a teda aj spôsobom spracovania. Zo súboru sa následne vyparsujú potrebné dáta a spracovanie súboru skončí jeho zatvorením.

## 1.4 Sekvenčný diagram - spracovanie súboru

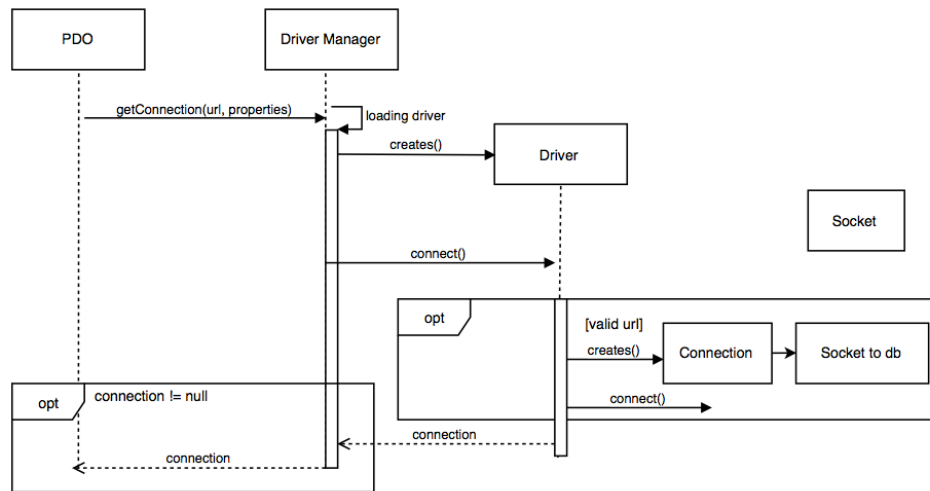
Figure 4: Sekvenčný diagram - spracovanie súboru



4 Diagram popisuje komunikáciu komponentov našej aplikácie počas spracovania súboru. Na začiatku aplikácia pošle lexeru požiadavku na validáciu súboru. Lexer súbor zvaliduje a rozparsuje na tokeny, v prípade chyby, pošle hlavnému programu správu s chybou. Tokeny pošle parseru, ktorý ich rozparsuje. Dáta z parseru sa pošlú ORM-ku, ktoré sa pripojí k databáze a uloží do nej naparované dáta. Na konci pošle správu o úspechu, resp. neúspechu celej operácie do hlavného programu, ktorý ju spracuje.

## 1.5 Sekvenčný diagram - pripojenie k databáze

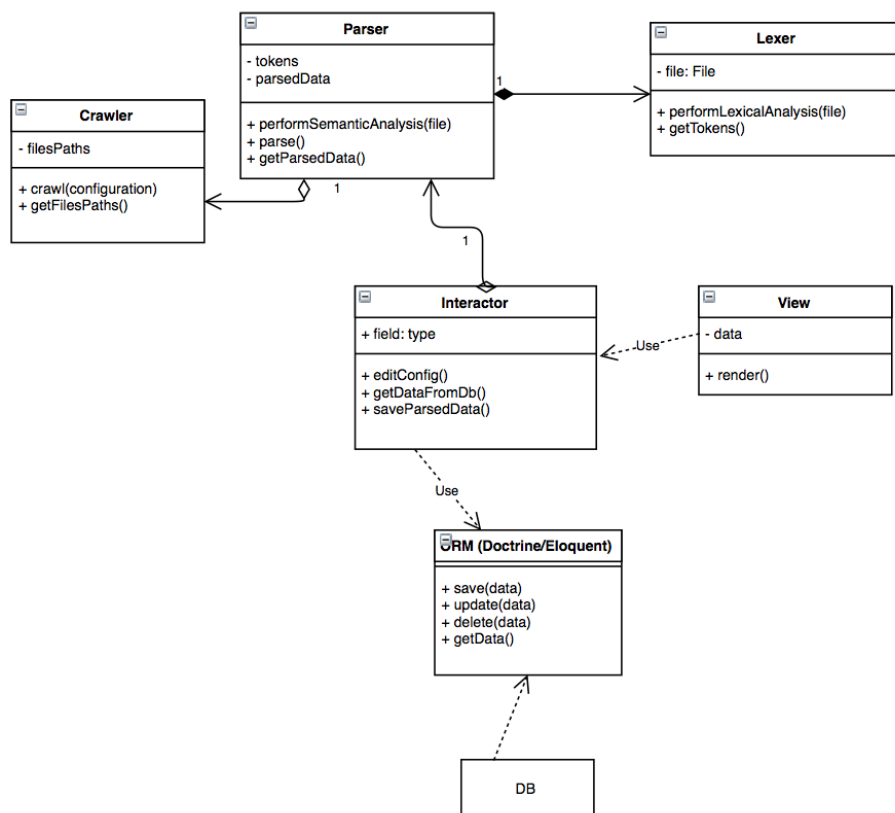
Figure 5: Sekvenčný diagram - pripojenie k databáze



5 Diagram popisuje proces pripojenia k databáze. Knížnica PDO pošle požiadavku na získanie spojenia k databáze driver manageru. Ten načíta drivre, vyberie správny a vytvorí jeho inštanciu. Driver sa následne pokúsi vytvoriť spojenie k databáze, tým, že sa pokúsi pripojiť na socket. V prípade neúspechu sa pošle správa do PDO o neúspechu. V prípade úspešného pripojenia na socket sa vráti spojenie k databáze do PDO.

## 1.6 Triedny diagram

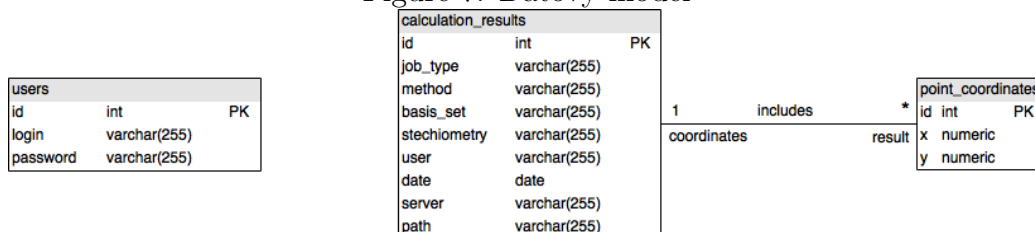
Figure 6: Triedny diagram



6 Triedny diagram modeluje jednotlivé triedy a vzťahy medzi nimi. Každá entita triedneho diagramu popisuje triedu a jej základné atribúty a metódy. Medzi triedami sú šípky, ktoré reprezenutujú jednotlivé vzťahy. Parser bude používať Crawler na vyhľadanie súborov a rovnako bude pouívať aj Lexer, pomou ktorého spraví lexikálnu analýzu súboru. Interactor bude akési spojítka jednotlivých hlavných častí aplikácie. View-u bude poskytovať dáta na zobrazenie a Parseru poskytne prístup k databáze.

## 1.7 Dátový model databázy

Figure 7: Dátový model



7 Dátový model popisuje štruktúru databázy

### Users

Tabuľka users obsahuje zoznam používateľov. Používatelia majú priradené id, plniace funkciu primárneho kľúča (user id), meno, pod ktorým sa prihlasujú (login) a heslo, s ktorým sa prihlasujú (password).

### Calculation results

Tabuľka calculation results obsahuje zoznam výsledkov výpočtov. Výsledky výpočtov majú priradené id, plniace funkciu primárneho kľúča (calculation result id), spôsob testovania vzorky (job type), metódu testovania vzorky (method), iniciálnu konfiguráciu (basis set), zjednodušený chemický vzorec testovanej vzorky (stochiometry), používateľa, spúšťajúceho testovanie vzorky (user), dátum testovania vzorky (date), meno servera, ukládajúceho daný výsledok výpočtu (server) a cestu k súboru daného výsledku výpočtu (path).

### Point coordinates

Tabuľka point coordinates obsahuje zoznam súradníc bodov. Súradnice bodov majú priradené id, plniace funkciu primárneho kľúča (point coordinate id), súradnicu x bodu (x), súradnicu y bodu (y) a id, plniace funkciu cudzieho kľúča (calculation result id).

### Popis vzťahov

Tabuľka users sa neviaže na žiadnu tabuľku. Tabuľka point coordinates sa viaže na tabuľku calculation results tak, že jedna súradnica bodu je ob-

siahnutá v práve jednom výsledku výpočtu, pričom jeden výsledok výpočtu môže obsahovať niekoľko súradníc bodov.

## 1.8 Dekompozícia

Komponent Autorizacia Pomocou tohto komponentu sa budú používatelia sprvným vyplnením prihlasovacieho formulára prihlásiť do systému. Komponent budú využívať bežní používatelia aj používatelia s administrátorskými právami. Údaje z formulára sa porovnávajú s údajmi v databáze. Pri zhode sa používatelia prihlásiť do systému a podľa typu majú k dispozícii rôzne funkcionality systému. Komponent Crawler Komponent bude vyhľadávať súbory s výsledkami výpočtov z meracích prístrojov, ktoré sú uložené na konkrétnych serveroch. Komponent Spracovanie súboru Tento komponent zanalyzuje (zistí, či súbory majú validnú štruktúru) a následne spracuje vyhľadane súbory do vhodného formátu. Komponent Databáza Komponent, ktorý má na starosti pripojenie a prácu s databázou.

## 2 Návrh používateľského rozhrania rozhranie

Figure 8: Používateľské rozhranie 1

The image shows a user interface design for a application named "QCH Calcs". The interface has a dark gray background. At the top left, the text "QCH Calcs" is displayed in a light gray font. In the center of the screen, there is a light gray rectangular box titled "Prihlasenie" (Login). Inside this box, there are two input fields: one labeled "Meno" (Name) and another labeled "Heslo" (Password). Both input fields are empty and have a light gray border.



Figure 9: Používateľské rozhranie 2

QCH Calcs								
ID	Job Type	Method	Basis set	Stechiometry	User	Date	Server	Path
1	Scan	RM062X	6-311++G(2df,2pd)	C3H7N1O2	MATEJKA	30-Jul-14	GINC-HAL	calc/Alanine/s-ala-M062x.log: 1

[H]

Figure 10: Používateľské rozhranie 3

The screenshot displays the 'QCH Calcs' application interface. At the top, there is a header bar with the application name 'QCH Calcs' on the left and a user profile icon on the right. Below the header, there are two navigation links: 'Pridanie uzivatela' and 'Administracia uzivatelov'. The main content area is titled 'Vytvorenie noveho uzivatela' and contains a form with three input fields: 'Meno', 'Heslo', and 'Zopakovat heslo'. Each field is represented by a text label followed by a horizontal input bar.

8 Používateľské rozhranie aplikácie bolo navrhnuté jednoducho podľa požiadaviek zadávateľa projektu.

## 3 Analýza technológií

### 3.1 Výber programovacieho jazyka

Pri výbere programovacieho jazyka sme sa rozhodovali medzi Python-om a PHP. Vybrali sme si PHP, pretože je pre tento projekt najvhodnejší, medzi výhody, ktoré nám ponúka pri vývoji patria napríklad:

- Patrí medzi najpoužívanejšie jazyky vo webových aplikáciach
- Je predinštalovaný na serveri, na ktorom bude bežať aj naša aplikácia
- Plne podporuje objektovo orientované programovanie
- V PHP je na výber veľa kvalitných frameworkov na prácu s databázou
- Vieme s ním efektívne pracovať

Python je náročnejšie nakonfigurovať na webovom serveri, kde pravdepodobne nebudeme mať možnosť inštalácie nového softvéru a oproti PHP ponúka len málo výhod pre náš projekt. Žiadnu podstatnú výhodu nám Python neponúka.

## 3.2 Výber databázového systému

Pri výbere databázového systému sme sa rozhodovali medzi Mysql a PostgreSQL. PostgreSQL ponúka veľa pokročilých funkcií, ako napríklad rekurzívne dopyty, pohľady. Naša aplikácia bude obsahovať jednoduchú databázu s malým počtom tabuliek a tieto pokročilé funkcie nevyužijeme. Preto sme si vybrali mysql databázu, ktorá je predinštalovaná na serveri a jej databázové enginy sú optimalizované pre webové aplikácie.

## 4 Testovacie scenáre

### 4.1 Existencia súboru

Otestovať existenciu súboru a jeho korektné otvorenie. Ak funkcia dostane cestu korektného súboru, so súborom sa dá ďalej pokračovať, v opačnom prípade funkcia súbor zahodí a nepokračuje sa v ďalšom spracovávaní.

### 4.2 Crawler

Otestovať funkcionálnosť Crawlera. Používateľ pridá nové súbory v používateľskom rozhraní. Crawler má nájsť novopridané súbory a aktualizovať databázu.

### 4.3 Validnosť súboru

Otestovať validitu súboru. Ak funkcia dostane validný súbor (je v požadovanom formáte), pokračuje sa ďalej v procese. V opačnom prípade ak funkcia

dostane nevalidný vstup (súbor je v zlom formáte), ďalej sa nepokračuje a funkcia súbor zahodí.

#### **4.4 Rozlišovanie linuxových a windowsových súborov**

Otestovať rozlišovanie medzi linuxovým a windowsovým súborom (rozdiel je v type súboru a v pár znakoch). Funkcia rozlíši, či dostala na vstup linuxový alebo windowsový súbor a následne sa súbor parsuje podľa linuxového alebo windowsového formátu.

#### **4.5 Databáza**

Otestovať pridávanie nových prvkov do databázy. Pridá sa nový korektný súbor a následne treba zistiť, či sa pridá aj do databázy. Pridá sa nový nekorektný súbor, následne treba skontrolovať, či sa údaje zo súboru nepridali do databázy.