

UNIXMOTION

# Kvantovo-chemické výpočty

*Jaroslav Ištók*

*Katarína Fabianová*

*Dušan Suja*

*Jerguš Adamec*



FMFI AIN

# Obsah

<b>1</b>	<b>Špecifikácia požiadaviek</b>	<b>3</b>
1.1	Úvod . . . . .	3
1.1.1	Účel . . . . .	3
1.1.2	Rozsah projektu . . . . .	3
1.1.3	Definície, akronymy a skratky . . . . .	3
1.1.4	Odkazy . . . . .	3
1.1.5	Prehľad zostávajúcej časti dokumentu . . . . .	4
1.2	Všeobecný popis . . . . .	4
1.2.1	Perspektívny pohľad na projekt . . . . .	4
1.2.2	Funkcionalita výslednej aplikácie . . . . .	4
1.2.3	Charakteristika používateľov . . . . .	5
1.2.4	Všeobecné obmedzenia . . . . .	5
1.2.5	Predpoklady a závislosti . . . . .	5
1.3	Zoznam špecifických požiadaviek na systém . . . . .	5
<b>2</b>	<b>Návrh</b>	<b>7</b>
2.1	Diagramy . . . . .	7
2.1.1	Entitno-relačný diagram . . . . .	7
2.1.2	Use-case diagram . . . . .	9
2.1.3	Stavový diagram . . . . .	11
2.1.4	Sekvenčný diagram - spracovanie súboru . . . . .	12
2.1.5	Sekvenčný diagram - pripojenie k databáze . . . . .	13
2.1.6	Dátový model . . . . .	14
2.1.7	Komponentný diagram . . . . .	15
2.1.8	Domain-level class diagram . . . . .	16
2.1.9	Data-flow diagram . . . . .	16
2.1.10	Triedny diagram . . . . .	16
2.2	Používateľské rozhranie . . . . .	17
2.3	Analýza technológií . . . . .	19
2.3.1	Výber programovacieho jazyka . . . . .	19
2.3.2	Výber databázového systému . . . . .	20
2.3.3	Výber ostatných technológií . . . . .	20
2.4	Testovacie scenáre . . . . .	21
2.4.1	Existencia súboru . . . . .	21
2.4.2	Crawler . . . . .	21
2.4.3	Validnosť súboru . . . . .	21
2.4.4	Rozlišovanie linuxových a windowsových súborov . . . . .	22
2.4.5	Databáza . . . . .	22

2.4.6	Prihlásenie . . . . .	22
2.4.7	Vyhľadávanie v databáze . . . . .	22
2.4.8	Pridávanie nových súborov . . . . .	22

## Zoznam obrázkov

1	Entitno-relačný diagram . . . . .	7
2	Use-case diagram . . . . .	9
3	Stavový diagram . . . . .	11
4	Sekvenčný diagram - spracovanie súboru . . . . .	12
5	Sekvenčný diagram - pripojenie k databáze . . . . .	13
6	Dátový model . . . . .	14
7	Triedny diagram . . . . .	16
8	Obrazovka s prihlásením . . . . .	17
9	Obrazovka s tabuľkou výpočtov . . . . .	18
10	Obrazovka s výpočtom . . . . .	19

# 1 Špecifikácia požiadaviek

## 1.1 Úvod

### 1.1.1 Účel

Špecifikácia obsahuje požiadavky, ktoré bude aplikácia implementovať. Je určená pre každého, kto sa bude podieľať na vývoji. Zadávateľ projektu na základe špecifikácie skontroluje, či výsledná aplikácia bude spĺňať všetky jeho požiadavky. Členovia vývojového tímu ju budú používať pri plánovaní a riadení procesu vývoja. Pomôže im pri ďalšom návrhu aplikácie a neskôr im ju umožní otestovať. Špecifikácia nie je odborným textom, je zrozumiteľná rovnako pre vývojárov aj zadávateľa projektu. Všetky použité odborné termíny sú vysvetlené v sekcii Definície, akronymi a skratky.

### 1.1.2 Rozsah projektu

Projekt sa radí medzi stredne veľké projekty. Jeho vývoj bude prebiehať v časovom horizonte približne pol roka, pričom jeho samotná implementácia bude prebiehať v rozsahu jedného mesiaca. Obsah aplikácie bude pozostávať z častí ako sú crawler, lexer, parser, ORM a jednoduchého, no súčasne intuitívneho grafického užívateľského rozhrania.

### 1.1.3 Definície, akronymi a skratky

- Crawler - nástroj, ktorý prelieza adresáre na serveroch a hľadá súbory
- Lexer - nástroj, ktorý analyzuje štruktúru súboru (dokumentu)
- Parser - nástroj na spracovanie údajov zo súborov
- ORM - nástroj na prácu s databázou z programovacieho jazyka
- mysql - relačná (sql) databáza na ukladanie údajov
- apache - webový server, na ktorom bude bežať grafické používateľské rozhranie aplikácie

### 1.1.4 Odkazy

- Príklad zobrazenia spracovaných dát: [http://neon.dpp.fmph.uniba.sk/qch\\_calcs/index.php](http://neon.dpp.fmph.uniba.sk/qch_calcs/index.php)

### 1.1.5 Prehľad zostávajúcej časti dokumentu

V nasledujúcich kapitolách nájdete rozširujúce informácie o projekte. Všeobecný popis projektu, perspektívu projektu, podrobný popis funkcionality, účel projektu, charakteristiku cieľových používateľov projektu a ďalšie.

## 1.2 Všeobecný popis

Aplikácia bude naplňať databázu zozbieranými údajmi. Bude obsahovať výsledky kvantovo-chemických výpočtov a bude pravidelne aktualizovaná o nové dáta. Aktualizácia dát bude prebiehať automaticky alebo manuálne na vyžiadanie používateľa. Aplikácia bude prístupná len konkrétnym používateľom, ktorí sa budú prihlasovať pomocou prihlasovacieho mena a hesla. Prihlásení používatelia budú môcť vo webovom rozhraní vyhľadávať a filtrovať údaje na základe bázy, metódy merania a ďalších parametrov.

### 1.2.1 Perspektívny pohľad na projekt

Projekt bude mať otvorený zdrojový kód. Bude bežať na linuxovom serveri a bude poskytovať webové rozhranie.

### 1.2.2 Funkcionalita výslednej aplikácie

- Aplikácia vyhľadá súbory s výsledkami výpočtov z meraciach prístrojov, ktoré sú uložené na konkrétnych serveroch. Dáta zo súborov najskôr analyzuje, potom spracuje a uloží ich do databázy.
- Aplikácia v pravidelných časových intervaloch, alebo na vyžiadanie používateľa, bude rozširovať databázu o dáta z novopridaných súborov.
- Aplikácia poskytne používateľovi jednoduché a intuitívne webové rozhranie, v ktorom bude možné vykonávať požadované operácie nad dátami z databázy, ako je napríklad pokročilé vyhľadávanie na základe rôznych kritérií. Webové rozhranie bude vedieť poskytnúť informácie o určitej molekule, či bola niekedy analyzovaná, akými metódami bola analyzovaná a podobne.
- Na prácu s aplikáciou postačí pripojenie na internet a webový prehliadač.
- Aplikácia bude chránená heslom a každý používateľ bude mať svoje prihlasovacie údaje, ktoré bude možné zmeniť.

### 1.2.3 Charakteristika používateľov

Aplikácia je určená pre chemikov, ktorí pracujú s veľkým množstvom dát z vypočítaných výpočtov a potrebujú v nich mať poriadok. Vzhľadom na povahu aplikácie, nebudú existovať špeciálne používateľské role.

### 1.2.4 Všeobecné obmedzenia

- Aplikácia je robená na mieru, takže nemôže si hocikto vytvoriť účet a používať ju.
- Na využívanie aplikácie je potrebné mať prístup k internetu a webový prehliadač.
- Webové rozhranie zobrazuje len molekuly, ktoré sú uložené v databáze.
- Ak je výpočet pre nejakú molekulu neúplný alebo chybný, potom danú molekulu nebude možné zobraziť. Objaví sa iba upozornenie, že dáta z výpočtu nie sú validné.
- Ak hľadaná molekula neexistuje v databáze, používateľ má možnosť nahráť súbor s výpočtom pre danú molekulu a v nasledujúcom vyhľadávaní, údaje o molekule budú zobrazené.

### 1.2.5 Predpoklady a závislosti

Aplikácia bude bežať na linuxovom serveri. Bude využívať relačnú databázu na ukladanie údajov. Predpokladáme, že nebude mať veľké nároky na procesor a operačnú pamäť. Vzhľadom na povahu dát, ktoré budú v databáze uložené, bude jej veľkosť maximálne niekoľko desiatok megabajtov. Používateľské rozhranie bude bežať na webovom serveri.

## 1.3 Zoznam špecifických požiadaviek na systém

### [1] Vyhľadávanie a spracovanie súborov s dátami

- (a) Aplikácie bude vyhľadávať súbory na serveroch ktoré obsahujú výsledky výpočtov z meracích zariadení. vyhľadávať bude na základe prípony v názve súboru. Konkrétne priečinky, ktoré sa budú prehľadávať budú špecifikované v konfiguračnom súbore, ktorý budú editovať používatelia.
- (b) Nad zozbieranými súbormi prebehne jednoduchá syntaktická analýza. Tým sa vylúčia súbory, ktoré sú chybné, resp. neobsahujú požadované informácie

- (c) Dáta z validných súborov sa rozparsujú na tokeny, čiže konkrétne údaje, ktoré sa uložia do databázy vo forme tabuliek.
- (d) Údaje v databáze budú aktualizované v pravidelných časových intervaloch alebo na vyžiadanie používateľa.

## [2] **Používateľské rozhranie aplikácie**

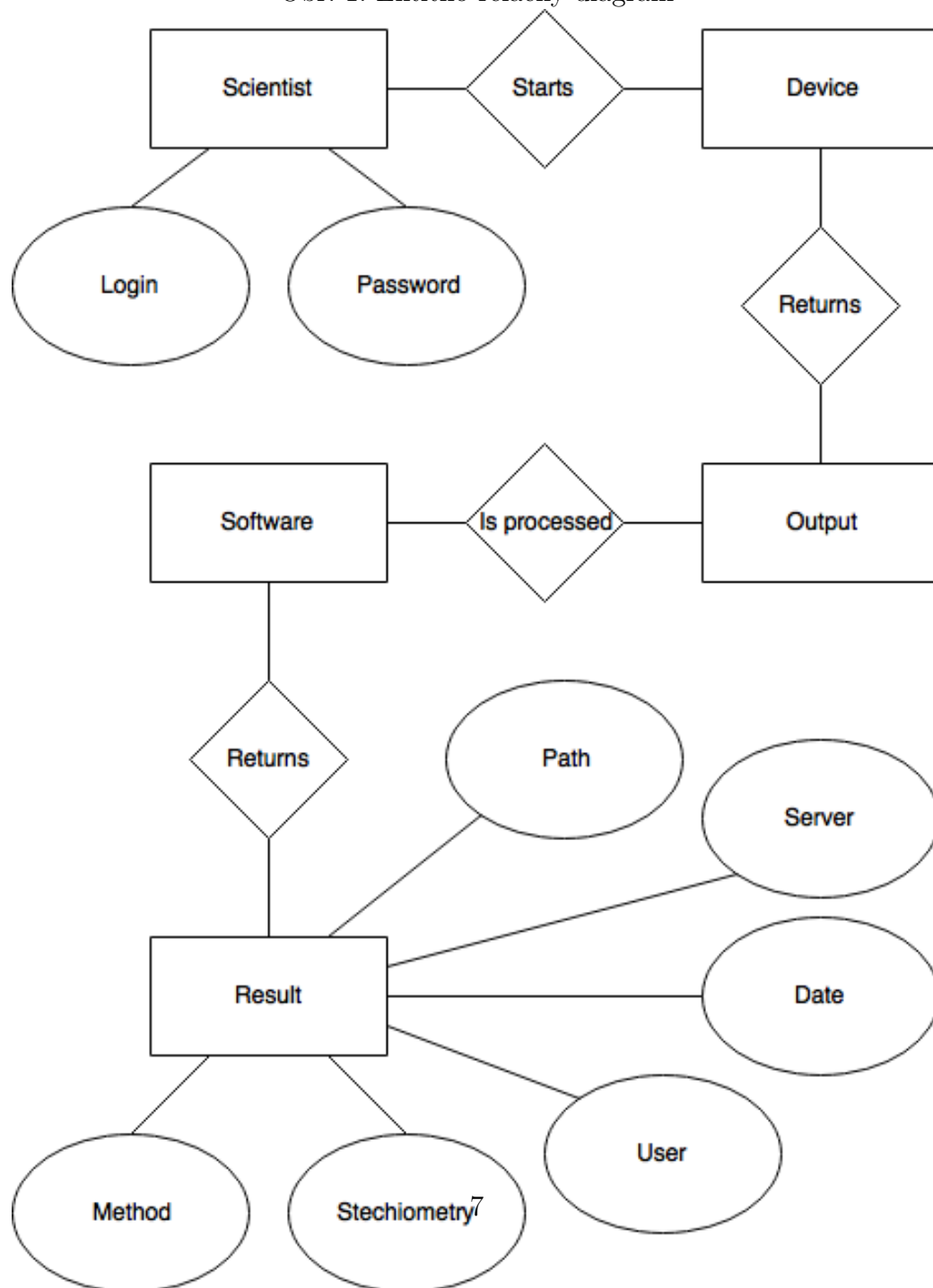
- (a) Po otvorení stránky sa zobrazí prihlasovacia obrazovka, kde je potrebné zadať prihlasovacie údaje.
- (b) V prípade zabudnutia hesla, bude možnosť obnoviť heslo, po zadaní emailu.
- (c) Po prihlásení sa zobrazí rozhranie aplikácie, ktoré bude generované dynamicky na základe zvolených kritérií filtrovania údajov, bude obsahovať ovládacie prvky, konkrétne tlačidlo na odhlásenie sa z aplikácie, tlačidlo na aktualizácie údajov v databáze, jednoduchý vyhľadávací box, ktorý bude prehľadávať databázu na základe zvolených kritérií, možnosť editácie konfiguračných súborov a možnosť vykreslenia náhľadu molekuly.
- (d) Kritéria, podľa ktorých bude možné vyhľadávať budú napríklad báza, metóda, stechiometria(vzorec) molekuly, dátum a ďalšie.
- (e) Údaje sa budú zobrazovať v prehľadnej tabuľke

## 2 Návrh

### 2.1 Diagramy

#### 2.1.1 Entitno-relačný diagram

Obr. 1: Entitno-relačný diagram

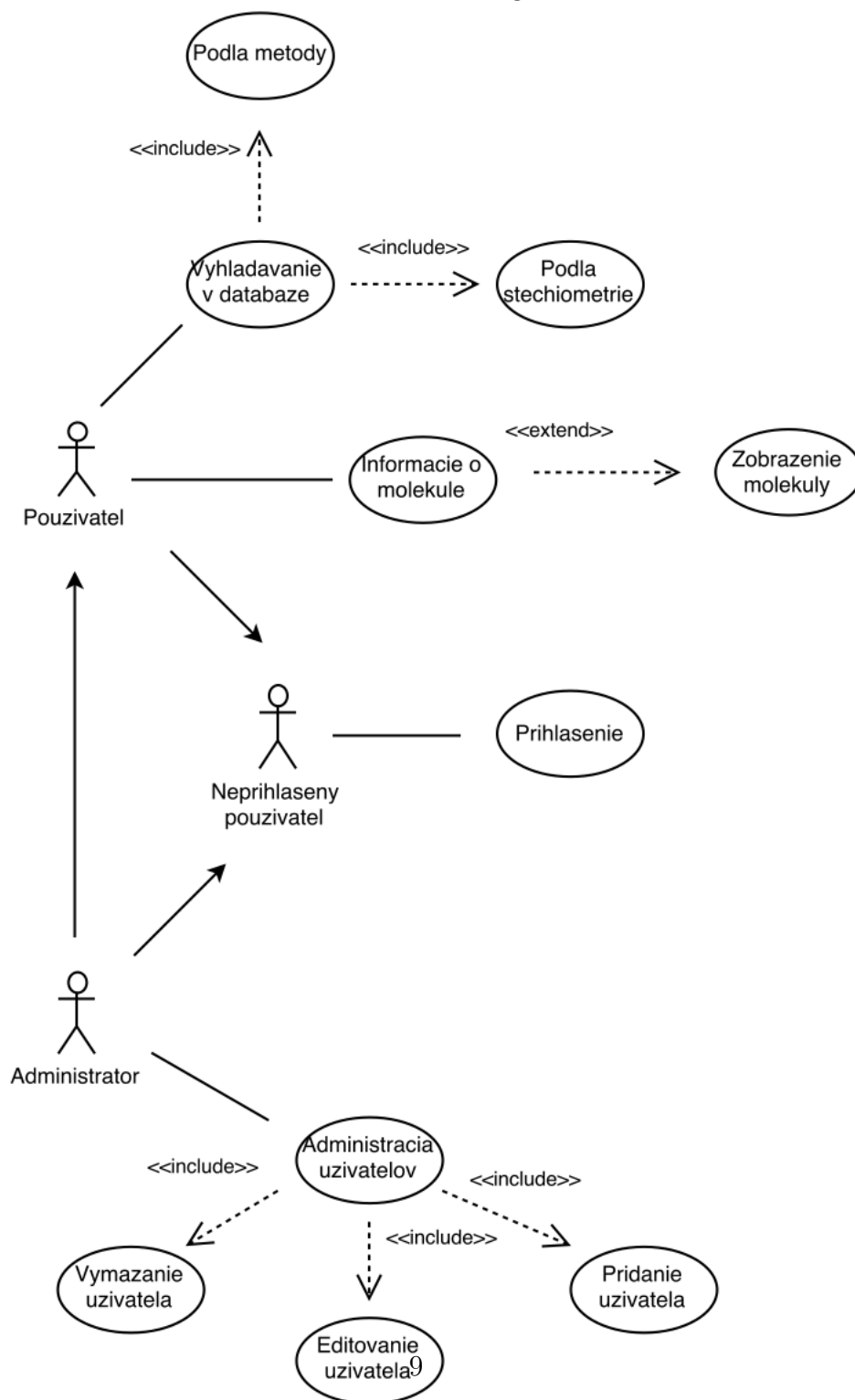




1 Entitno-relačný diagram znázorňuje vzťahy (relácie) medzi entitami. Diagram je použitý na modelovanie priestoru domény, pre ktorú sa informačný systém vyvíja (ústav experimentálnej fyziky). Entity sú zakreslené do obdĺžnikov. Vzťahy (relácie) medzi entitami sú v kosoštvorcoch, sú prepojené so všetkými 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.

### 2.1.2 Use-case diagram

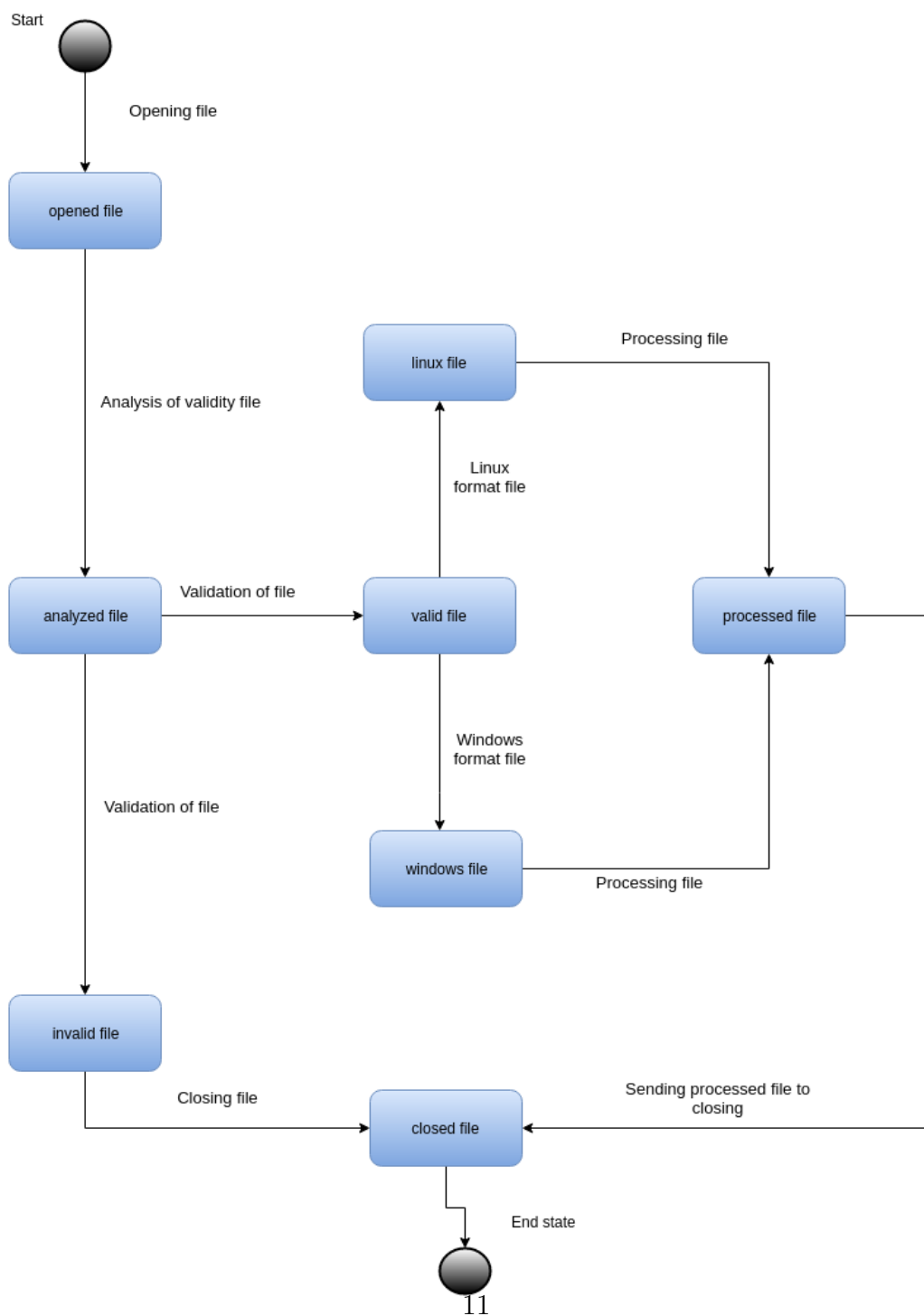
Obr. 2: Use-case diagram



2 Use-case diagram popisuje interakciu používateľov (aktorov) s aplikáciou. Neprihlásený používateľ sa môže prihlásiť. Prihlásený používateľ môže vyhľadávať v databáze výsledkov meraní podľa rôznych kritérií a zobrazit výpis podrobných informácií o danej molekule.

### 2.1.3 Stavový diagram

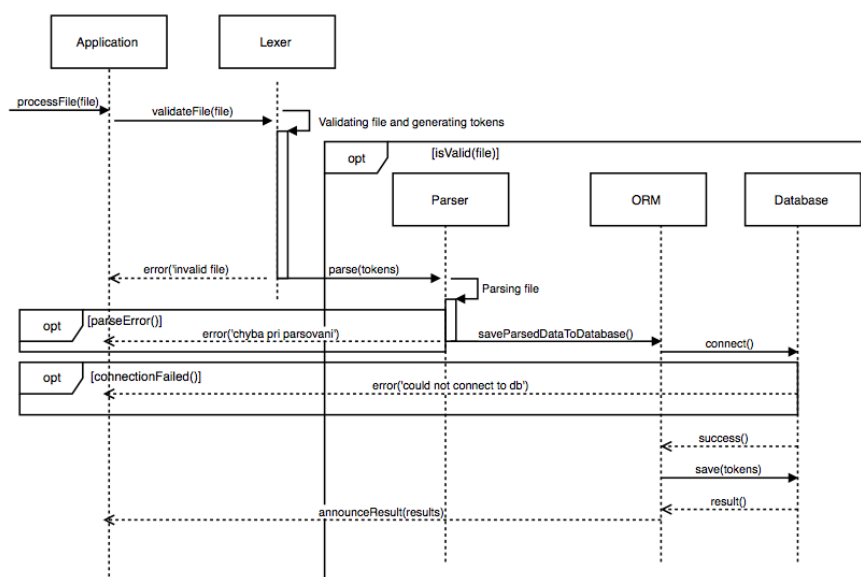
Obr. 3: Stavový diagram  
**Stavový diagram**



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 dva formáty sa líšia vnútornou štruktúrou a teda aj spôsobom spracovania. Zo súboru sa následne vyparsujú potrebné dáta a spracovanie súboru skončí jeho zatvorením.

#### 2.1.4 Sekvenčný diagram - spracovanie súboru

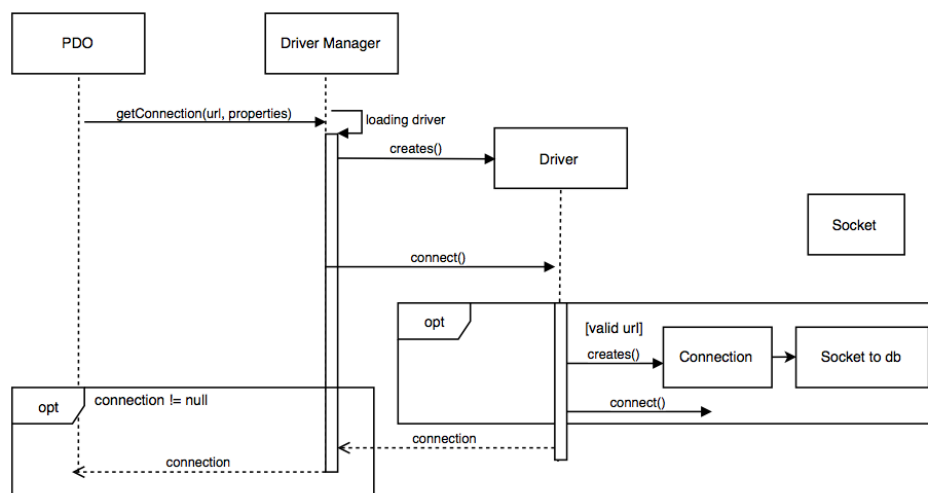
Obr. 4: Sekvenčný diagram - spracovanie súboru



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 naparsované dáta. Na konci pošle správu o úspechu, resp. neúspechu celej operácie do hlavného programu, ktorý ju spracuje.

### 2.1.5 Sekvenčný diagram - pripojenie k databáze

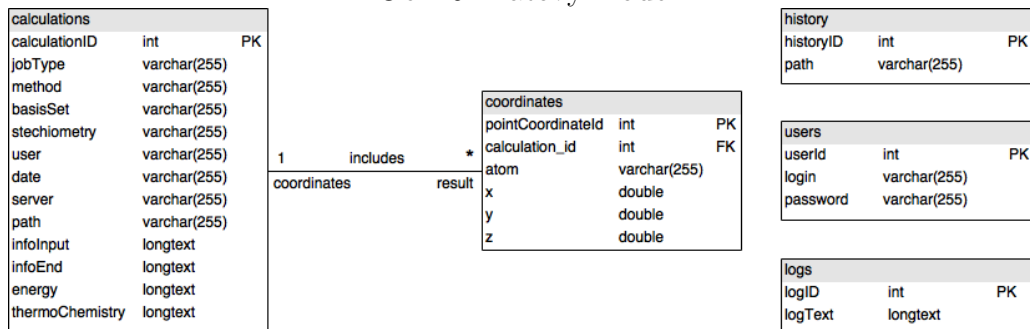
Obr. 5: Sekvenčný diagram - pripojenie k databáze



5 Diagram popisuje proces pripojenia k databáze. Knížnica PDO pošle požiadavku na získanie spojenia k databáze driver manageru. Ten načíta driver, 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.

### 2.1.6 Dátový model

Obr. 6: Dátový model



6 Dátový model popisuje štruktúru databázy.

- **Calculations**

Tabuľka calculations obsahuje zoznam výsledkov výpočtov. Výsledky výpočtov majú priradené id, plniace funkciu primárneho kľúča (calculationID), spôsob testovania vzorky (jobType), metódu testovania vzorky (method), iníciaľnu konfiguráciu (basisSet), zjednodušený chemický vzorec testovanej vzorky (stochiometry), používateľa, spúšťajúceho testovanie vzorky (user), dátum testovania vzorky (date), meno servera, ukladajúceho daný výsledok výpočtu (server), cestu k súboru daného výsledku výpočtu (path), bližšie nešpecifikované, zadávateľom požadované údaje (infoInput), (infoEnd), energiu (energy) a termochémiu (thermoChemistry).

- **Coordinates**

Tabuľka coordinates obsahuje zoznam súradníc jednotlivých atómov. Súradnice atómov majú priradené id, plniace funkciu primárneho kľúča (pointCoordinateId), atóm, ktorému prislúchajú (atom), súradnicu x (x), súradnicu y (y) a súradnicu z (z).

- **Users**

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

- **History**

Tabuľka history obsahuje zoznam spracovaných súborov. Spracované súbory majú priradené id, plniace funkciu primárneho kľúča (historyID) a cestu, ktorá popisuje ich umiestnenie (path).

- **Logs**

Tabuľka logs obsahuje zoznam chybových správ. Chybové správy majú priradené id, plniace funkciu primárneho kľúča (logID) a text, ktorý je ich obsahom (logText).

- **Popis vzťahov**

Tabuľky users, history a logs sa neviažu na žiadnu tabuľku. Tabuľka coordinates sa viaže na tabuľku calculations tak, že jedna súradnica atómu je obsiahnutá v práve jednom výsledku výpočtu, pričom jeden výsledok výpočtu môže obsahovať niekoľko súradníc atómov.

### 2.1.7 Komponentný diagram

Komponentný diagram popisuje dekompozíciu projektu na moduly.

- **Autorizácia**

Pomocou komponentu autorizácia sa budúci používateľ správnym vyplnením prihlasovacieho formulára prihlási do systému. Údaje z formulára sa porovnávajú s údajmi v databáze. Pri zhode sa používateľ prihlási do systému, kde má k dispozícii rôznu funkcionálnosť.

- **Crawler**

Komponent Crawler bude vyhľadávať súbory s výsledkami výpočtov z meracích prístrojov, ktoré sú uložené na konkrétnych serveroch.

- **Spracovanie súboru**

Komponent spracovanie súboru analyzuje (zistí či súbory majú validnú štruktúru) a následne spracuje vyhladané súbory do vhodného formátu.

- **Databáza**

Komponent databáza má na starosti pripojenie a prácu s databázou.

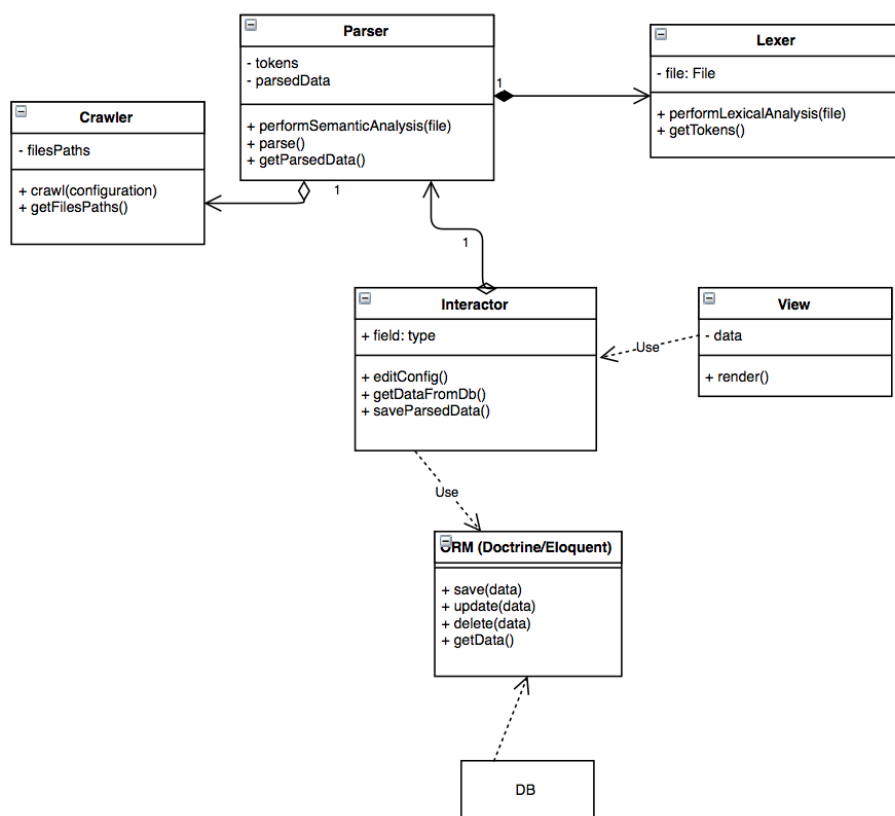


### 2.1.8 Domain-level class diagram

### 2.1.9 Data-flow diagram

### 2.1.10 Triedny diagram

Obr. 7: Triedny diagram

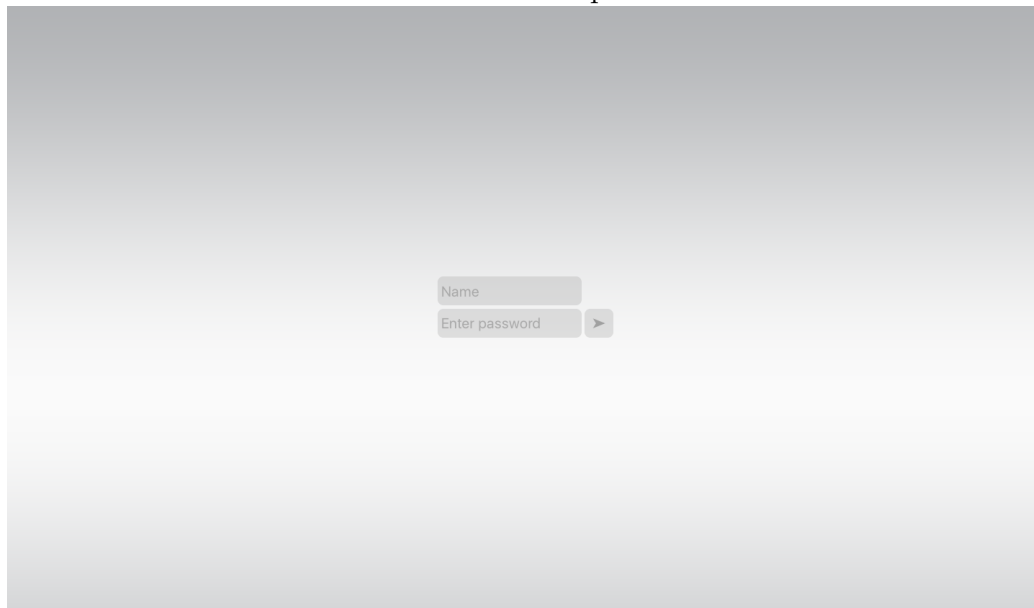


7 Triedny diagram modeluje jednotlivé triedy a vzťahy medzi nimi. Každá entita triedneho diagramu popisuje triedu, jej základné atribúty a metódy. Medzi triedami sú šípky, ktoré reprezentujú jednotlivé vzťahy. Parser bude používať Crawler na vyhľadanie súborov a rovnako bude používať aj Lexer, pomocou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

## 2.2 Používateľské rozhranie

Používateľské rozhranie aplikácie je navrhnuté jednoducho podľa požiadaviek zadávateľa projektu.

Obr. 8: Obrázovka s prihlásením



8 Úvodná obrazovka s prihlásením pozostáva z prihlasovacieho formuláru tvoreného textovými poľami, určenými pre zadanie používateľského mena a používateľského hesla, slúžiacimi na identifikáciu jednotlivých vybraných používateľov.

Obr. 9: Obrazovka s tabuľkou výpočtov

Select job type
Select method
Select basis set
Select stoichiometry

Show ReportRunLogout

Type to start searching ...

Id	Job Type	Method	Basis Set	Stoichiometry	User	Date	Server	Path	Show info
1	Mixed	G3MP2	G3MP2	C6H13N1O2	PAPP	30-Nov-2007	GINC-NODE-2	/home/tis/KChVypocty/files/iso-2a1-G3MP2.log	Show details
2	Mixed	G3MP2	G3MP2	C6H13N1O2(1+,2)	PAPP	28-Jun-2010	GINC-NOD2	/home/tis/KChVypocty/files/leucine-1b1-G3MP2.log	Show details
3	FOpt	RHF	6-311G(2df,2pd)	C6H6	MATEJKA	27-Feb-2016	GINC-HAL	/home/tis/KChVypocty/files/C6H6-OVGF.log	Show details
4	Freq	RHF	6-311G(2df,2pd)	C6H6	MATEJKA	27-Feb-2016	GINC-HAL	/home/tis/KChVypocty/files/C6H6-OVGF.log	Show details
5	SP	ROVGF-FC	6-311G(2df,2pd)	C6H6	MATEJKA	27-Feb-2016	GINC-HAL	/home/tis/KChVypocty/files/C6H6-OVGF.log	Show details
6	Scan	RCIS-FC	Aug-CC-pVQZ	H2	PAPO	08-Apr-2013	UNPC-HPBOOK	/home/tis/KChVypocty/files/h2.out	Show details
7	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	31-Jul-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP3.log	Show details
8	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	30-Jul-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP24.log	Show details
9	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	01-Aug-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP8c.log	Show details
10	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	28-Jul-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP13.log	Show details
11	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	01-Aug-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP8.log	Show details
12	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	30-Jul-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP27.log	Show details
13	SP	RB97D	Aug-CC-pVDZ	C20H26O4	PAPP	30-Jul-2016	GINC-NEON	/home/papp/calc/DCHP/DCHP25.log	Show details

9 Po prihlásení sa zobrazí hlavná stránka. Nachádza sa na nej tabuľka všetkých údajov, ktoré sú uložené v databáze. Medzi údajmi je možné vyhľadávať pomocou interaktívneho vyhľadávacieho pola alebo ich filtrovať pomocou filtrov. Nachádza sa tam aj tlačidlo na zobrazenie logu, spustenie novej analýzy súborov a odhlásenie používateľa.

Obr. 10: Obrazovka s výpočtom

Informácie	
ID	3
Job type	FOpt
Method	RHF
Basis set	6-311G(2df,2pd)
Stechiometry	C6H6
User	MATEJKA
Date	27-Feb-2016
Server	GINC-HAL
Path	/home/tis/KChVypocty/files/C6H6-OVGF.log
InfoInput	P HF/6-311G(2df,2pd) gfinput pop=full opt freq # SCF(MaxConventionalCycles=200,XQC) symmetry(pg=d06h) benzene ovgt 0,1
Coordinates	C 0 0 -0.0053424137 H 0 0 1.0679866598 C 1.1965505691 0 -0.6961712068 H 2.1260808133 0 -0.1595066701 C 1.1965505691 0 -2.0778287932 H 2.1260808133 0 -2.6144933299 C 0 0 -2.7686575863 H 0 0 -3.8419866598 C -1.1965505691 0 -2.0778287932 H -2.1260808133 0 -2.6144933299 C -1.1965505691 0 -0.6961712068 H -2.1260808133 0 -0.1595066701
Energy	HF=-230.7726743

10 Pri každom výpočte zobrazenom v tabuľke sa nachádza tlačidlo “Show details”. Kliknutím na toto tlačidlo sa zobrazí detailný výpis všetkých informácií o výpočte v prehľadnej tabuľke.

## 2.3 Analýza technológií

### 2.3.1 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.

### 2.3.2 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.

### 2.3.3 Výber ostatných technológií

- **HTML**

Hypertextový značkový jazyk (HyperText Markup Language; HTML) je značkový jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom prehliadači. HTML kladie dôraz skôr na prezentáciu informácií (odseky, fonty, váha písma, tabuľky atď.) ako na sémantiku (význam slov) a umožňuje vytvárať dokumenty obsahujúce text, hypertextové odkazy, multimediálny a iný obsah, formuláre, skripty a metainformácie prehliadateľné v tzv. webovom prehliadači. Jazyk HTML je textový, teda umožňuje čítanie a upravovanie priamo v textovom editore. V projekte bude použitý pri tvorbe webových dokumentov z hľadiska ich obsahu, štruktúry.

- **CSS**

Kaskádové štýly (Cascading Style Sheets; CSS) je všeobecné rozšírenie HTML. CSS je jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov. Pomocou kaskádových štýlov sa vytvárajú štruktúrované dokumenty, teda oddeľuje sa obsah dokumentu (HTML) od jeho vzhľadu (CSS). Získa sa tým prehľadný a jednoduchý kód. CSS je možné presunúť do externých súborov, zmenší sa tým dátová veľkosť a dá sa jedným súborom zmeniť celý štýl stránky, pričom sa nezaručuje rovnaké vykresľovanie vo všetkých prehliadačoch, vzhľadom k rôznym interpretáciám CSS rôznymi prehliadačmi. V projekte budú použité pri tvorbe webových dokumentov z hľadiska ich vzhľadu.

- **JavaScript**

JavaScript je skriptovací programovací jazyk využívaný najmä na vytváranie dynamického obsahu webových stránok. V projekte bude použitý pri tvorbe webových dokumentov z hľadiska ich dynamického obsahu a reagovania na vstup používateľa.

- **AJAX**

AJAX (Asynchronous JavaScript + XML) je súhrnné označenie pre technológie vývoja interaktívnych webových aplikácií, ktoré umožňujú meniť obsah stránok bez potreby ich kompletného znovunačítania zo servera. V porovnaní s klasickými webovými aplikáciami môžu AJAX-ové aplikácie pri vhodnom návrhu poskytovať používateľsky komfortnejšie prostredie, vyžadujú však použitie moderných webových prehliadačov. AJAX nie je samostatný programovací jazyk ani technológia sama o sebe. Je to kombinácia HTML a CSS pre značkovanie a štýlovanie informácií pri zobrazení, DOM spojeného s JavaScriptom pre dynamické zobrazenie a interakciu s prezentovanou informáciou, metódy pre výmenu dát medzi prehliadačom a serverom, bez nutnosti obnovovať zobrazovanú stránku a formátu pre dáta poslané prehliadaču, ktoré môžu byť dynamicky vytvorené skriptom na strane serveru (bežné formáty zahŕňajú XML, predformátované HTML, plain text a JavaScript Object Notation, JSON). V projekte budú použité pri zmene obsahu webových dokumentov bez potreby ich kompletného znovunačítania zo servera.

## **2.4 Testovacie scenáre**

### **2.4.1 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.

### **2.4.2 Crawler**

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

### **2.4.3 Validnosť súboru**

Otestovať validnosť 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.

#### **2.4.4 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.

#### **2.4.5 Databáza**

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

#### **2.4.6 Prihlásenie**

Otestovať správne prihlásenie používateľa. V používateľskom rozhraní sa zadajú správne prihlasovacie údaje (meno a heslo), očakávaným výsledkom je úspešné prihlásenie sa na stránku. Zadajú sa nesprávne prihlasovacie údaje (nesprávne meno alebo nesprávne heslo alebo oboje), aplikácia zobrazí chybovú hlášku zlého prihlásenia a stránka nie je dostupná.

#### **2.4.7 Vyhľadávanie v databáze**

Otestovať správnosť vyhľadávania v databáze. Zadajú sa parametre, podľa ktorých sa v databáze nachádzajú nejaké molekuly, aplikácia zobrazí všetky molekuly korektne.

#### **2.4.8 Pridávanie nových súborov**

Otestovať aktualizovanie databázy. Po pridaní nového súboru cez používateľské rozhranie, sa automaticky aktualizuje databáza a pri vyhľadávaní molekúl sa zobrazia aj najnovšie pridané údaje.