

KVANTOVO-CHEMICKÉ VÝPOČTY

Návrh

Jaroslav Ištok

Katarína Fabianová

Dušan Suja

Jerguš Adamec

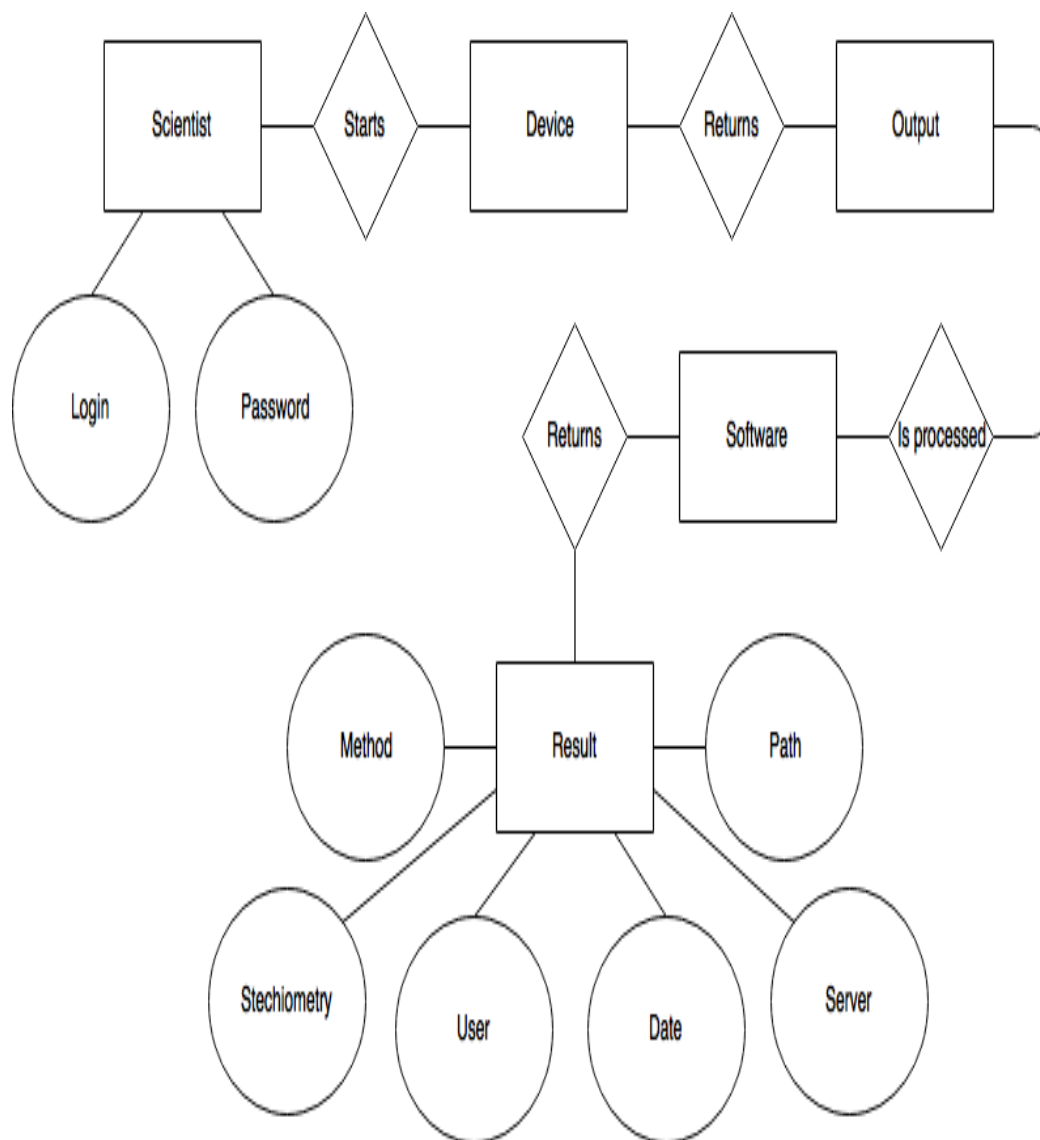
November 4, 2016

Contents

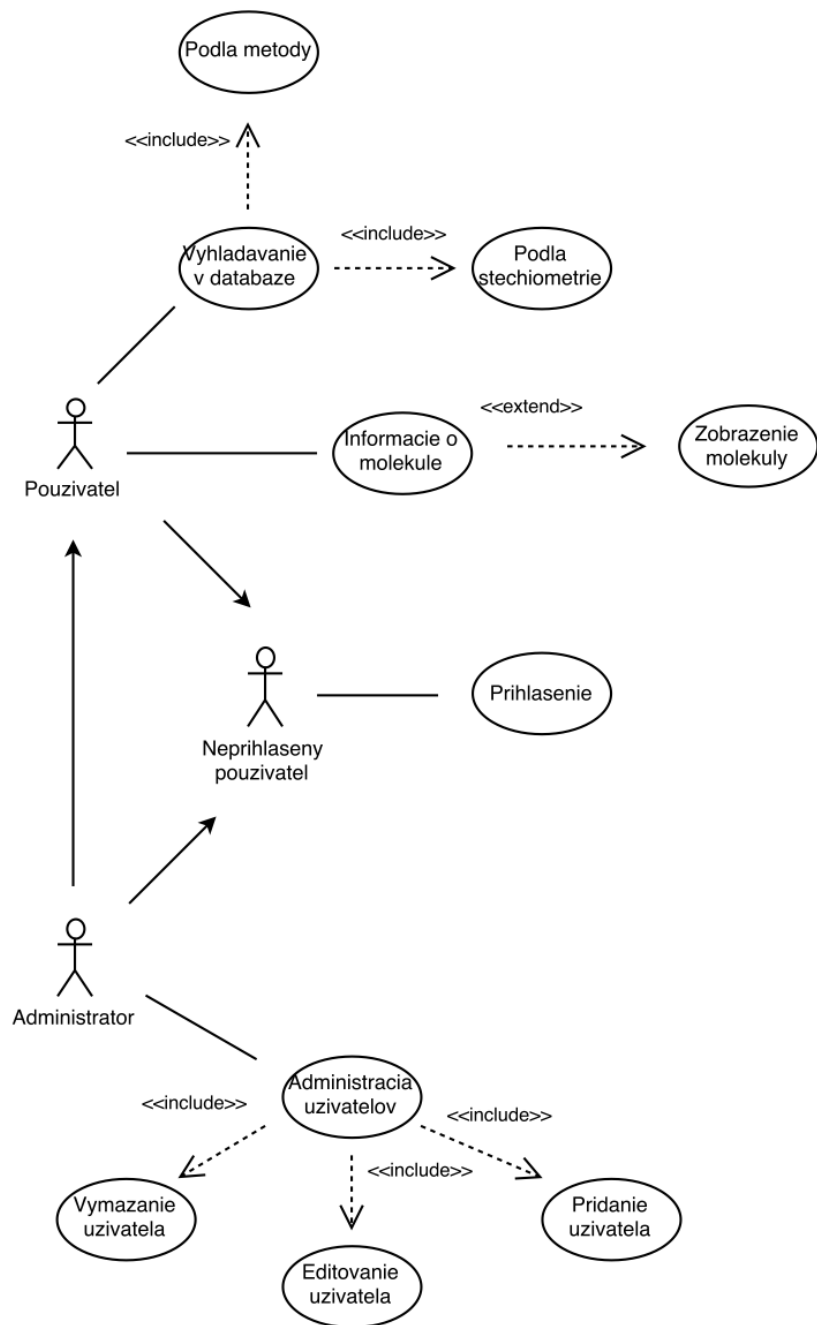
1	Diagramy	2
1.1	Entitno-relačný diagram	2
1.2	Use case diagram	3
1.3	Stavový diagramy	5
1.4	Sekvenčné diagramy	6
1.4.1	Spracovanie súboru	6
1.4.2	Pripojenie k databáze	7
1.5	Triedny diagram	8
1.6	Dátový model databázy	8
1.7	Komponentový diagram a dekompozícia	9
2	Návrh používateľského rozhrania rozhranie	9
3	Analýza technológii	11
3.1	Výber programovacieho jazyka	11
3.2	Výber databázového systému	12
4	Testovacie scenáre	12
4.1	Existencia súboru	12
4.2	Crawler	12
4.3	Validnosť súboru	12
4.4	Rozlišovanie linuxových a windowsových súborov	13
4.5	Databáza	13

1 Diagramy

1.1 Entitno-relačný diagram

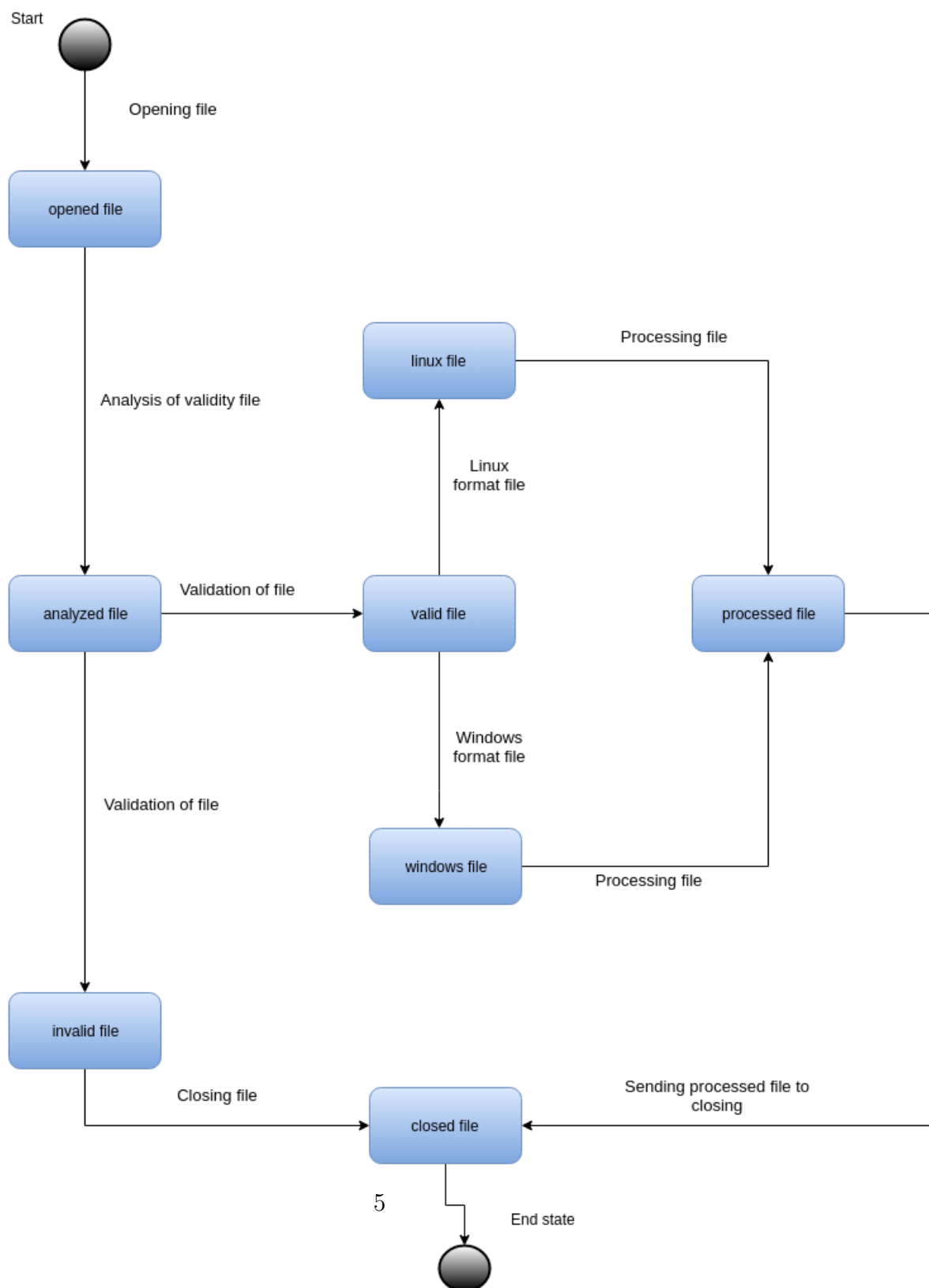


1.2 Use case diagram



1.3 Stavový diagramy

Stavový diagram



1.4 Sekvenčné diagramy

1.4.1 Spracovanie súboru

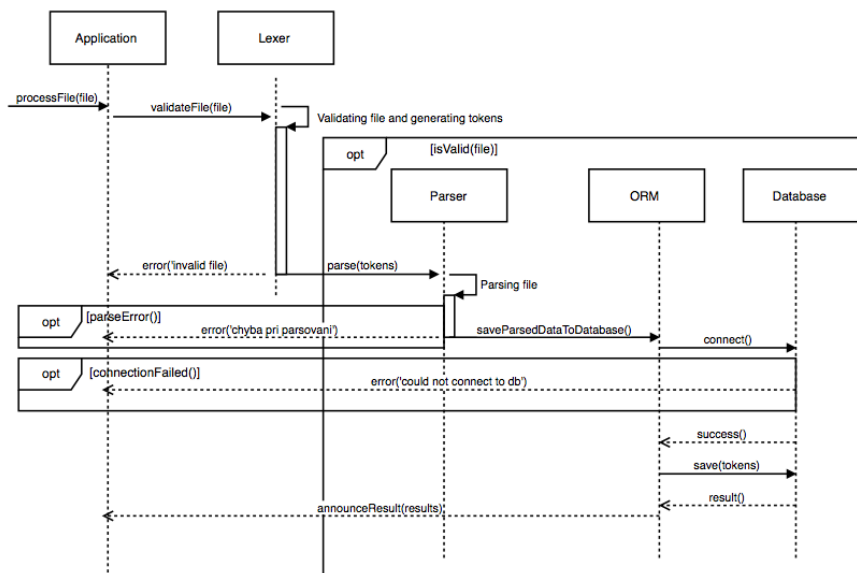


Diagram popisuje komunikáciu komponentov našej aplikácie počas spracovania súboru. Na začiatku aplikácia pošle lexeru požiadavku na validáciu súboru. Lexer súbor zvaliduje a rozparsuje na tokeny, v prípade chyby, pošle hlavnému programu správu s chybou. Tokeny pošle parseru, ktorý ich rozparsuje. Dáta 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.

1.4.2 Pripojenie k databáze

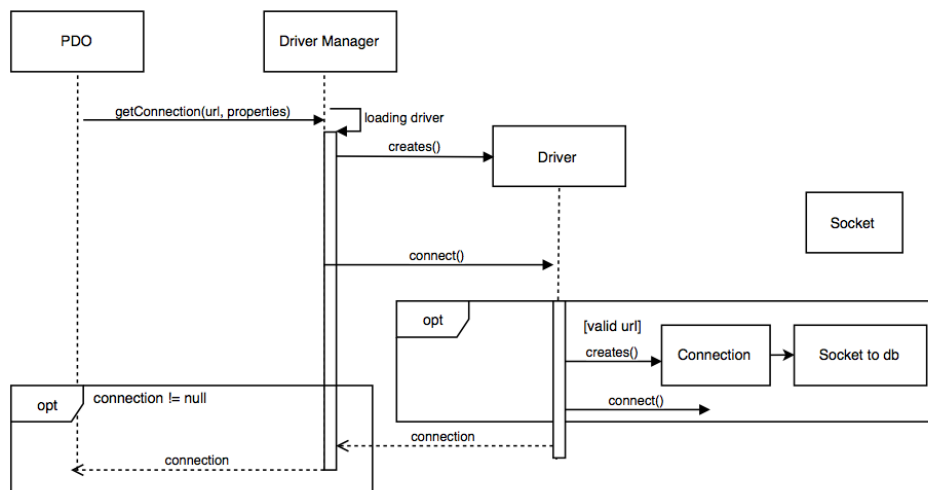
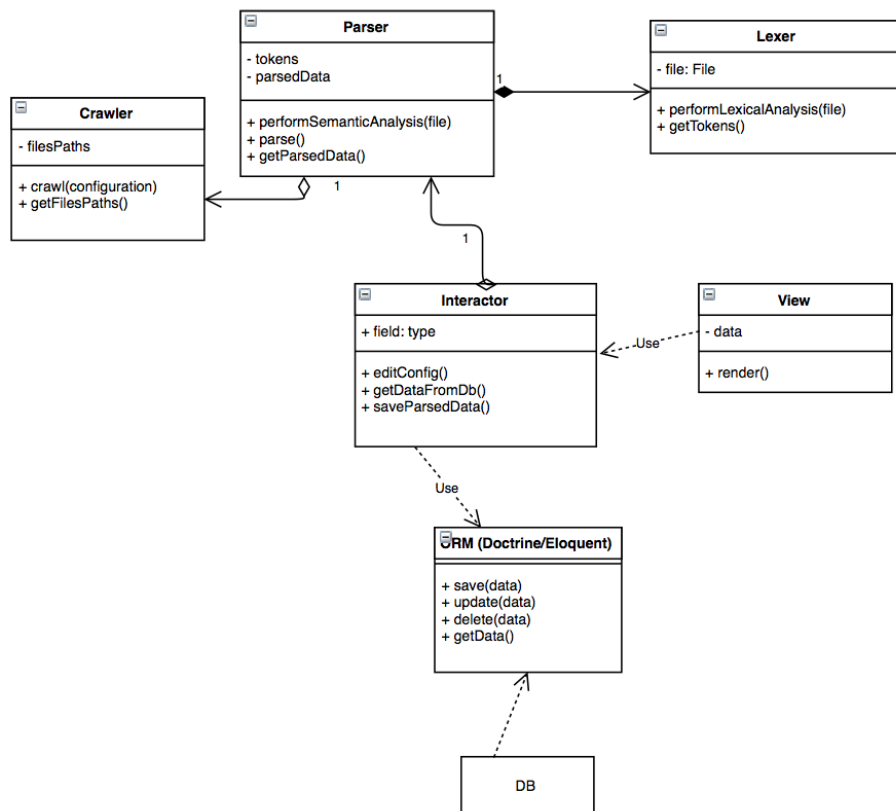
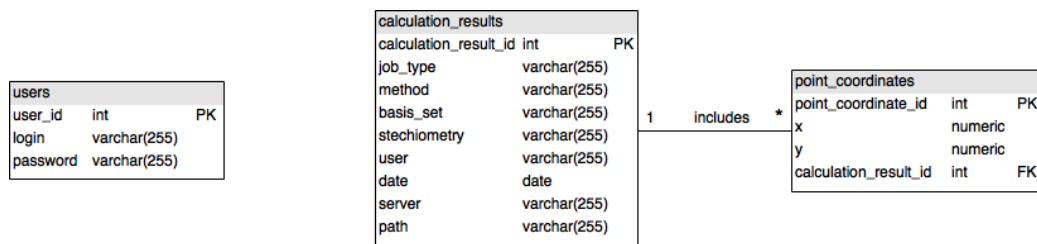


Diagram popisuje proces pripojenia k databáze. Knižnica PDO pošle požiadavku na získanie spojenia k databáze driver manageru. Ten načíta drivre, vyberie správny a vytvorí jeho inštanciu. Driver sa následne pokúsi vytvoriť spojenie k databáze, tým, že sa pokúsi pripojiť na socket. V prípade neúspechu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

1.5 Triedny diagram

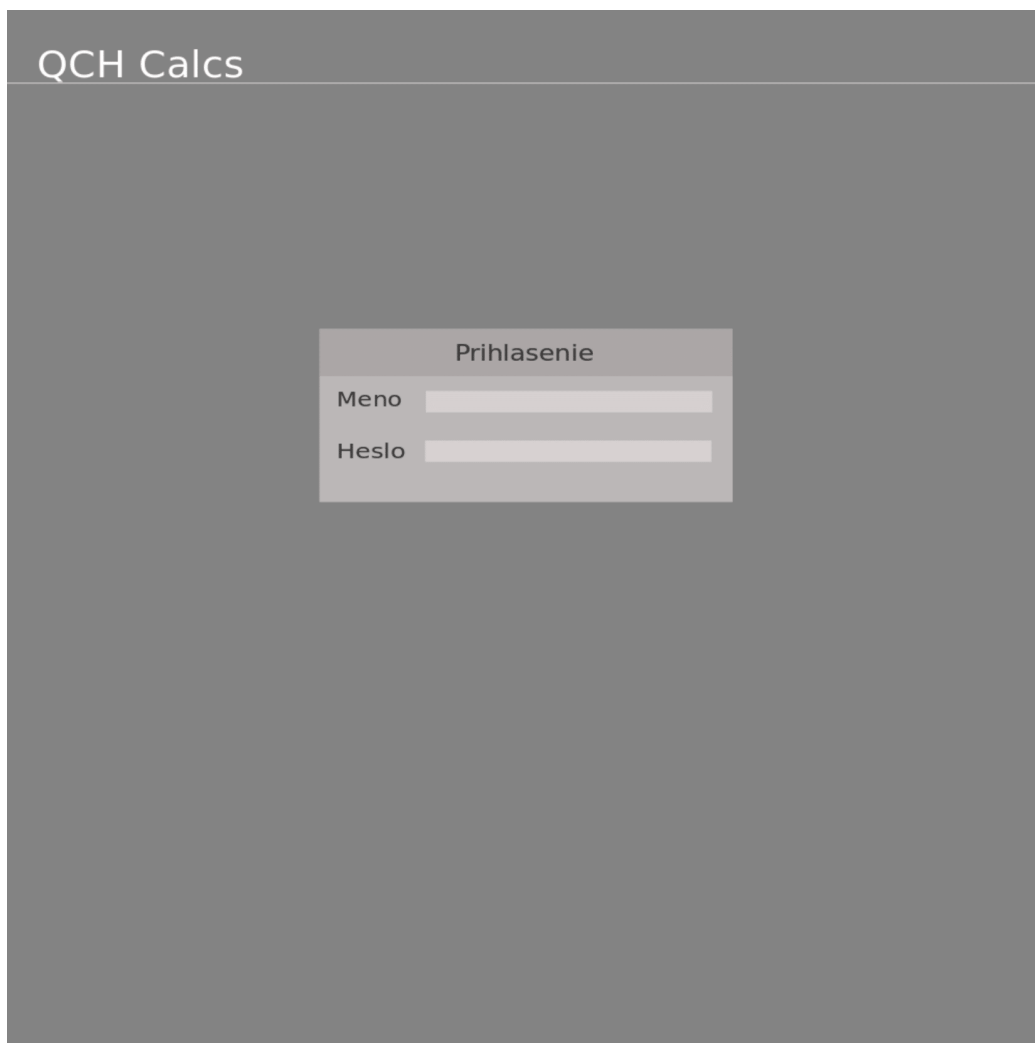


1.6 Dátový model databázy



1.7 Komponentový diagram a dekompozícia

2 Návrh používateľského rozhrania rozhranie



The image shows a user interface mockup for a login system. At the top, there is a dark gray header bar with the text "QCH Calcs" in white. Below this, the main area is a solid dark gray. Centered in this area is a light gray rectangular box representing the login form. The form has a title "Prihlasenie" at the top. Below the title are two input fields: the first is labeled "Meno" and the second is labeled "Heslo".

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



3 Analýza technológi

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 najpoužívanjšie 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 funkcionality 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 pridal aj do databázy. Pridá sa nový nekorektný súbor, následne treba skontrolovať, či sa údaje zo súboru nepridali do databázy.