

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Zátěžové testy webové aplikace**

# ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **David BARTA**  
Osobní číslo: **A16B0002P**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informatika**  
Téma práce: **Zátěžové testy webové aplikace**  
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

### Zásady pro vypracování

1. Seznamte se s projektem TbUIS (Testbed University Information System), zejména s jeho knihovnou support. Dále se seznamte s existujícími nástroji třetích stran pro zátěžové testování.
2. Navrhněte a realizujte zátěžový test, který bude modifikovanou metodou Monte Carlo simulovat průběh semestru v aplikaci UIS.
3. Na základě výsledků z předchozího bodu navrhněte kolekci zátěžových testů, kdy se budou postupně zvyšovat počty studentů i učitelů v aplikaci UIS. Tyto testy realizujte na jednom počítači a výsledky vyhodnoťte.
4. S využitím vhodného externího nástroje pro podporu zátěžových testů a poznatků z předchozích bodů navrhněte a realizujte zátěžové testy jedné aplikace UIS z několika souběžně pracujících počítačů.
5. Metodiku zátěžového testování popište a všechny získané výsledky vyhodnoťte.





Rozsah bakalářské práce: **doporuč. 30 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Doc. Ing. Pavel Herout, Ph.D.**  
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **7. října 2019**

Termín odevzdání bakalářské práce: **7. května 2020**

*Radová*

**Doc. Dr. Ing. Vlasta Radová**  
děkanka



*Brada*

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 11. května 2020

David Bárta

# Poděkování

Rád bych poděkoval panu Doc. Ing. Pavlu Heroutovi, Ph.D., za odborné vedení práce, připomínky a rady.

## **Abstract**

The aim of this thesis is to perform load tests in an already existing web application that would prove that the application can withstand a larger amount of data and can work without failure under this load for a longer period of time. An already existing external Support library was used for these tests. The testing program simulates the course of one semester at a university. It generates random actions which are run in a web application using the Support library and a tool called Selenium Webdriver. Different sets of data of different sizes are used to simulate the semester. When generating the random actions, an implementation of probabilities is tested. These should direct the course of the semester in the right direction.

## **Abstrakt**

Cílem této práce je provedení zátěžových testů již existující webové aplikace, které prokáží, že testovaná webová aplikace zvládne větší množství dat a dokáže bez selhání pracovat pod touto zátěží delší časový úsek. Pro tyto testy byla použita již existující externí knihovna Support. Testovací program simuluje průběh jednoho semestru na vysoké škole. Generuje náhodné akce, které jsou za pomoci knihovny Support a nástroje Selenium Webdriver spouštěny ve webové aplikaci. K simulaci semestru jsou použity různé sady dat různých velikostí. Při generování náhodných akcí je vyzkoušeno implementování pravděpodobností, které mají za cíl usměrnit průběh semestru správným směrem.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Webová aplikace</b>	<b>10</b>
2.1	TbUIS a její části . . . . .	10
2.1.1	UIS . . . . .	10
2.1.2	Poruchové klony . . . . .	12
2.2	Knihovna JUnit . . . . .	12
2.3	Nástroj Selenium . . . . .	13
2.3.1	Selenium Page Object model . . . . .	13
<b>3</b>	<b>Zátěžové testy</b>	<b>18</b>
3.1	Testovaná aplikace . . . . .	18
3.1.1	Knihovna Support . . . . .	18
3.2	Databáze . . . . .	19
3.2.1	Import databáze . . . . .	19
3.2.2	Implementace importu dat . . . . .	20
3.2.3	Modul <i>oracle</i> . . . . .	20
3.2.4	Struktura databázového souboru XML . . . . .	21
3.3	Seznam uživatelských akcí . . . . .	22
3.3.1	Omezení . . . . .	23
3.4	Modifikovaná metoda Monte Carlo . . . . .	24
3.4.1	Obecné informace . . . . .	24
3.4.2	Použití metody . . . . .	25
3.5	Externí testovací nástroj JMeter . . . . .	25
<b>4</b>	<b>Implementace</b>	<b>26</b>
4.1	Prerekvizity . . . . .	26
4.1.1	Použité závislosti . . . . .	26
4.1.2	Konfigurační soubor . . . . .	27

4.2	Struktura projektu . . . . .	28
4.2.1	Třída Main . . . . .	29
4.2.2	Balík json_configuration . . . . .	31
4.2.3	Třída DatabaseGenerator . . . . .	34
4.2.4	Třída OracleUtils . . . . .	34
4.2.5	Třídy UserActions a UserActionsProbabilities . . . . .	34
4.2.6	Třída Results . . . . .	37
4.2.7	Další soubory . . . . .	39
4.3	Uživatelské akce a jejich testy . . . . .	40
4.4	Problémy při vypracování . . . . .	40
4.4.1	Problém s pamětí v prohlížeči Firefox . . . . .	40
4.4.2	Problém s ovladačem Google Chrome . . . . .	41
4.5	Testování . . . . .	42
4.5.1	Dosažené výsledky . . . . .	43
<b>5</b>	<b>Zátěžové testy v nástroji JMeter</b>	<b>48</b>
<b>6</b>	<b>Závěr</b>	<b>52</b>
	<b>Literatura</b>	<b>54</b>
<b>A</b>	<b>Uživatelská příručka</b>	<b>55</b>
A.1	Prerekvizity . . . . .	55
A.2	Před spuštěním . . . . .	55
A.2.1	Příprava databáze . . . . .	55
A.2.2	Konfigurační soubor . . . . .	55
A.2.3	Nastavení konstant . . . . .	56
A.3	Překlad a spuštění . . . . .	56
A.3.1	JMeter . . . . .	56
<b>B</b>	<b>Obsah CD</b>	<b>57</b>



# 1 Úvod

Na Katedře informatiky a výpočetní techniky byla v rámci diplomové práce *Aplikace s možností injekce chyb pro ověřování kvality testů* vyvinuta webová aplikace University Information System (UIS). Tato aplikace simuluje reálný informační systém a je navržena tak, aby bylo možné do ní importovat defekty. Aplikace UIS s defekty bude sloužit jako testbed pro zkoumání nových testovacích metod. UIS patří společně s podpůrnými knihovnamy do projektu Testbed UIS.

Tato práce by měla pomoci zátěžových testů ověřit, že webová aplikace UIS odolá zvýšené zátěži. Zároveň by se tím měla ověřit funkčnost a pokrytí dosavadních testů.

Nejprve je třeba se seznámit s externí knihovnou Support, která obsahuje rozhraní umožňující komunikaci s webovou aplikací. Je třeba pochopit modul *oracle*, který vytváří paralelní byznys logiku zapisování a uchovávání dat v objektech v paměti.

Je třeba zvolit vhodnou reprezentaci velkého množství vygenerovaných náhodných dat, které by měly být nahrány do aplikace UIS. Pro vygenerování těchto dat je nutné vytvořit nástroj, který data uloží ve zvoleném formátu do souboru.

Pro zátěžový test je nutné vymezit akce, které budou prováděny. Tyto akce by měly být prováděny nejen náhodně s pomocí modifikované metody Monte Carlo, ale i za pomoci scénáře, který by měl co nejlépe simulovat reálný průběh jednoho semestru na vysoké škole.

Cílem této práce je provedení zátěžových testů, které ověří protestovanost a odolnost webové aplikace proti velkému množství dat a dlouhému běhu zátěžových testů.

## 2 Webová aplikace

### 2.1 TbUIS a její části

Testbed University Information System (TbUIS) je souhrn projektů, který slouží hlavně jako podpůrný nástroj pro výzkum nových testovacích metod. Součástí TbUIS je tzv. *ErrorSeeder*, pomocí kterého lze do webové aplikace uměle vložit defekty, které mají za cíl otestovat funkčnost a pokrytí testů v rámci aplikace.

Poznámka: Projekt TbUIS byl popsán již v několika kvalifikačních pracích [8] [9] [10] a z tohoto důvodu bude zde popis TbUIS velmi stručný a budou popisovány pouze relevantní části.

#### 2.1.1 UIS

Webová aplikace University Information System (UIS) simuluje reálný informační systém, v němž jsou tři hlavní entity – učitel, student a předmět. Přesto že je aplikace na první pohled jednoduchá, má překvapivě mnoho stavů, do kterých se lze dostat. Celkový přehled všech stavů [9] lze vidět na obrázku 2.1.

Systém simuluje reálnou situaci, kdy dva typy uživatele interagují s předměty a mezi sebou. Učitel si například zapisuje předměty, které bude učit, vypisuje zkuškové termíny, hodnotí studenta atp. Student si přihlašuje předmět, hlásí se na vypsaný zkuškový termín, může si předmět odepsat atp. Přehled případů užití je vidět na obrázku 2.2 a přehledně v tabulce:

Nepřihlášený uživatel		
ID	Název případu užití	Popis
UC.01	Přihlášení do aplikace	Registrovaný uživatel se přihlásí do aplikace a na základě uživatelského jména bude rozpoznán jako student či učitel.
Přihlášený uživatel (student či učitel)		
ID	Název případu užití	Popis
UC.02	Odhlášení z aplikace	Odhlášení přihlášeného uživatele z aplikace

UC.03	Změna osobních údajů uživatele	Přihlášený uživatel může změnit svoje jméno, příjmení a/nebo email.
<b>Přihlášený uživatel – student</b>		
ID	Název případu užití	Popis
UC.04	Odepsání zapsaného předmětu	Student si zruší zapsaný předmět.
UC.05	Zobrazení úspěšně absolvovaných předmětů	Přehled úspěšně absolvovaných předmětů a získaných známek
UC.06	Zapsání předmětu	Student si zapíše dosud nezapsaný předmět.
UC.07	Zrušení zapsaného zkouškového termínu	Student si zruší zapsaný zkouškový termín pro zvolený předmět.
UC.08	Zapsání zkouškového termínu	Student si zapíše nový zkouškový termín pro zvolený předmět.
UC.09	Zobrazení všech účastníků zkouškového termínu	Zobrazí seznam účastníků daného zkouškového termínu pro zvolený předmět.
<b>Přihlášený uživatel – učitel</b>		
ID	Název případu užití	Popis
UC.10	Zrušení zapsaného předmětu	Učitel zruší předmět, který učí.
UC.11	Zobrazení seznamu studentů předmětu	Zobrazí seznam studentů, kteří mají zapsaný daný předmět.
UC.12	Zrušení zkouškového termínu	Učitel zruší zkouškový termín svého předmětu.
UC.13	Zobrazení účastníků zkouškového termínu	Zobrazí všechny účastníky daného zkouškového termínu.
UC.14	Vypsání nového zkouškového termínu	Učitel vypíše nový zkouškový termín pro svůj předmět.
UC.15	Zapsání známky studentovi ve formuláři	Učitel zapíše známku prostřednictvím formuláře danému studentovi.
UC.16	Zapsání nebo změna známky studentovi v tabulce	Učitel zapíše nebo změní již zapsanou známku studentovi prostřednictvím tabulky.
UC.17	Zapsání předmětu	Učitel začne učit dosud nezapsaný předmět.

UC.18	Zobrazení seznamu učitelů a předmětů	Zobrazí seznam učitelů a předmětů, které učí.
<b>Nepřihlášený uživatel – pro servisní a testovací účely</b>		
ID	Název případu užití	Popis
UC.19	Obnovení databáze	Odhlášený uživatel obnoví stav databáze (většinou před začátkem provádění samotných testů)
UC.20	Získání informací o schopnostech systému	Odhlášený uživatel získá informaci o možnostech systému, o obsahu databáze a o restrikcích v systému UIS.
UC.21	Export databáze	Odhlášený uživatel exportuje obsah databáze do souboru XML/JSON.
UC.22	Import databáze	Odhlášený uživatel importuje a nastaví databázi ze souboru XML/JSON.

### 2.1.2 Poruchové klony

Poruchové klony jsou uměle vytvořeny. Defekt je do nich zanesen, přičemž uživatel ví, o jaký defekt jde a jakým způsobem ovlivní aplikaci. Ověřuje se funkčnost testů, které by daný defekt měly odhalit. Účelem UIS a poruchových klonů je poskytnout odborné veřejnosti ověřenou aplikaci, na které lze zkoušet vývoj nových testovacích metod.

## 2.2 Knihovna JUnit

Knihovna je napsaná pro programovací jazyk Java a slouží k psaní jednotkových a dalších testů [7]. Testuje se zpravidla jen jeden prvek, kterému jsou pomocí testů přiřazeny různé hodnoty. Prvek by měl na tyto hodnoty rozumně reagovat a uživatele nějakým způsobem varovat, je-li nějaká hodnota špatná či nesmyslná. Lze si představit formulář na webové stránce a testování validních hodnot jednoho prvku formuláře, například pole pro zadání emailové adresy.

Aplikace UIS je masivně otestovaná pomocí knihovny JUnit 5.

## 2.3 Nástroj Selenium

Selenium [4] je projekt zastřešující řadu nástrojů a knihoven, které mají za cíl automatizovat interakci s webovým prohlížečem. V jádru projektu je *Selenium WebDriver* – rozhraní pro psaní instrukcí, které mohou být spuštěny ve více různých prohlížečích (například Google Chrome, Mozilla Firefox, Opera, Internet Explorer a další).

Nástroj Selenium WebDriver může být použit ve více programovacích jazycích, například Java, Python, C#, JavaScript, Kotlin.

### 2.3.1 Selenium Page Object model

Při práci se Selenium WebDriver se ukázalo jako velmi výhodné použití návrhového vzoru Page Object, který má za cíl v nástroji Selenium zjednodušit údržbu testů a zredukovat duplikované části kódu.

Page Object je třída, která v sobě obsahuje metody, které přímo interagují s konkrétní webovou stránkou. Jednotlivé testy pak místo přímé interakce s webovou stránkou komunikují s Page Objekty. To s sebou přináší jednu velkou výhodu – při jakékoli změně struktury webové stránky stačí změnit kód daného Page Objektu, přičemž testy, které se stránkou komunikují a manipulují, můžou zůstat nezměněné.

Níže je vidět úryvek kódu, který nepoužívá Page Objekty:

```
/**
 * Tests login feature
 */
public class Login {

    @Test
    public void testLogin() {
        // fill login data on sign-in page
        driver.findElement(By.name("user_name"))
            .sendKeys("testUser");
        driver.findElement(By.name("password")).sendKeys("my
            supersecret password");
        driver.findElement(By.name("sign-in")).click();

        // verify h1 tag is "Hello userName" after login
        driver.findElement(By.tagName("h1")).isDisplayed();
        assertThat(driver.findElement(By.tagName("h1"))
            .getText(), is("Hello userName"));
    }
}
```

```
}
```

Výše zmíněný úryvek kódu testuje správné vyplnění uživatelského jména na webové stránce, úryvek je převzatý ze stránky s dokumentací k rozhraní Page Object na webu Selenium [3]. Při bližším prozkoumání kódu si lze všimnout, že programátor musí při každé interakci s webovou stránkou nejprve nalézt hledaný prvek podle určitého klíče, zde je použito `By.name()`. Teprve poté lze s prvkem manipulovat (vyplnit jeho hodnotu, kliknout na něj, atp.).

Při jakékoliv změně struktury webové stránky (například změny názvu pole s přihlašovacím jménem) se musí zasáhnout i do kódu samotného testu.

Následuje ukázka implementace Page Object v projektu *PageObjects*, který je součástí TbUIS. Ukázka bude obsahovat jednu třídu, která popisuje část stránky studenta po jeho přihlášení. Jedná se o menu v levé části stránky, jeho struktura je vidět na obrázku 2.3. Vybraný element *My Subjects* je popsán v kódu:

```
public class Stu_Menu_Page extends PageObject {
    // ID's on web page
    ...
    public static final String MYSUBJECTS_LINK =
        "stu.menu.mySubjects";
    ...
    // Texts on web page
    ...
    public static final String TXT_MYSUBJECTS_LINK = "My Subjects";
    ...
    @FindBy(id = MYSUBJECTS_LINK)
    private List<WebElement> mySubjects;
    ...
    public String getMySubjectsLinkText() {
        String text = getElement(mySubjects,
            MYSUBJECTS_LINK).getText().trim();
        return text;
    }
    ...
    public boolean isMySubjectsLinkEnabled() {
        return getElement(mySubjects, MYSUBJECTS_LINK).isEnabled();
    }
}
```

Tento kód vytváří rozhraní mezi samotnou webovou stránkou a testem. Toto rozhraní popisuje webovou stránku formou Java objektů, které lze

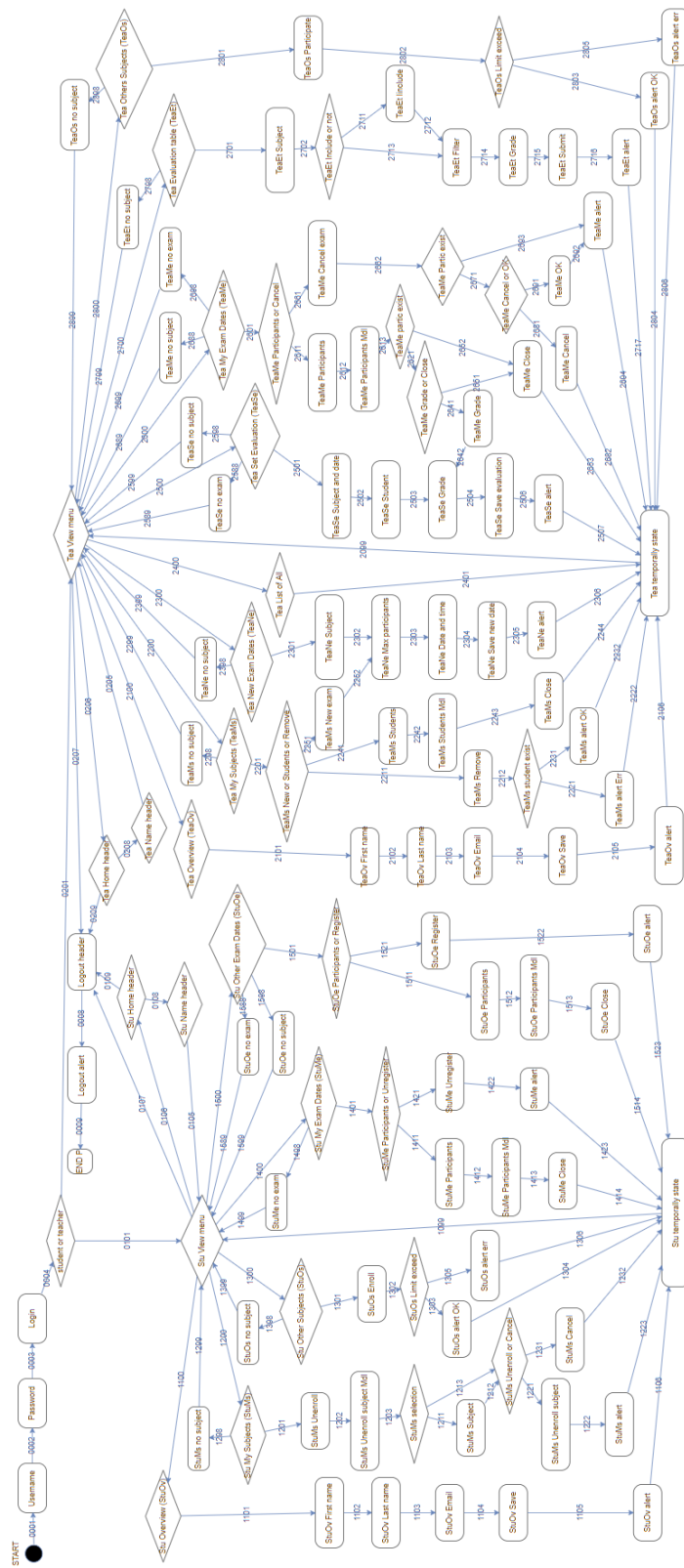


používat v testech. Při změně struktury stránky se pouze upraví její popis v Page Objektch odpovídajícím způsobem (například při změně názvu z *My Subjects* na *Your Subjects* by se pouze změnil text v konstantě `TXT_MYSUBJECTS_LINK`), testy zůstanou nezměněny.

Test poté může vypadat takto:

```
@Test
@Order(6)
@DisplayName("My Subjects enabled")
void MySubjects_enabled(Stu_Menu_Page menuPage) {
    Check.checkIsEnabled(menuPage.isMySubjectsLinkEnabled(),
        menuPage.MYSUBJECTS_LINK);
}
```

Page Objekty jako takové slouží pouze k popisu webové stránky, nikoliv k testování. Neměly by tedy obsahovat žádné testy ani porovnání pomocí `assert`.



Obrázek 2.1: Přehled stavů v aplikaci UIS [5]



Obrázek 2.2: Případy užití v aplikaci UIS [5]

## Student's View

[Overview](#)

**[My Subjects](#)**

[Other Subjects](#)

[My Exam Dates](#)

[Other Exam Dates](#)

Obrázek 2.3: Menu v levé části stránky přihlášeného studenta

## 3 Zátěžové testy

### 3.1 Testovaná aplikace

Testovanou aplikací je webová aplikace UIS. Cílem těchto testů je vyzkoušení chování aplikace při větším počtu dat. S použitím automatizačního nástroje Selenium WebDriver a zvýšením zátěže roste pravděpodobnost odhalení defektů v aplikaci již otestované jinými typy testů. Tyto případně objevené defekty jsou nad rámec této práce zaznamenány a případně opraveny.

V rámci analýzy je nutné se nejprve seznámit s knihovnou Support.

#### 3.1.1 Knihovna Support

Knihovna Support obsahuje důležité třídy, jejichž prostřednictvím se pomocí Page Objektů (které jsou také součástí knihovny) komunikuje s webovou aplikací UIS. Dále je v ní modul *oracle*, který je vysvětlen dále v samostatné sekci.

Pro využití knihovny v této bakalářské práci je nejdůležitější balík:

```
uis.support.services
```

(dále bude zmiňován jen jako *services*). Jeho struktura je popsána níže. Většina balíků v *services.student* a *services.teacher* obsahuje dvě třídy, jejichž názvy jsou pro větší přehlednost sjednoceny (například *Stu\_Overview\_Actions* a *Stu\_Overview\_Check* jsou sjednoceny na *Stu\_Overview\_Actions/Check*).

```
services
|
-- login -- Home_Actions
|         -- Login_Actions
|         -- Login_Check
|
-- student
|         -- m1_overview -- Stu_Overview_Actions/Check
|         -- m2_mysubjects -- Stu_MySubjects_Actions/Check
|         -- m3_othersubjects -- Stu_OtherSubjects_Actions/Check
|         -- m4_myexamdates -- Stu_MyExamDates_Actions/Check
|         -- m5_othereexamdates -- Stu_OtherExamDates_Actions/Check
|
```

```

-- teacher
    -- m1_overview -- Tea_Overview_Actions/Check
    -- m2_mysubjects -- Tea_MySubjects_Actions/Check
    -- m3_myexamdates -- Tea_MyExamDates_Actions/Check
    -- m4_newexamdates -- Tea_NewExamDates_Actions/Check
    -- m5_seteval -- Tea_SetEval_Actions/Check
    -- m6_evaltable -- Tea_EvalTable_Actions/Check
    -- m7_otherssubjects -- Tea_OthersSubjects_Actions/Check
    -- m8_listofall -- Tea_ListOfAll_Check

```

Třída `*_Actions` obsahuje metody, které provedou danou akci s použitím Page Objektů. Díky vysokoúrovňové interakci stačí jen vyvolat příslušnou metodu, není třeba se starat o přihlášení a odhlášení studenta či učitele apod. V těchto metodách je vždy použita i metoda ze třídy `*_Check`, která zkontroluje správné provedení akce na základě zobrazené potvrzovací zprávy (Alert) na webové stránce. V případě zobrazení chybové hlášky metoda nahlásí chybu a aplikace ukončí svoji činnost.

## 3.2 Databáze

Struktura databáze je následující:

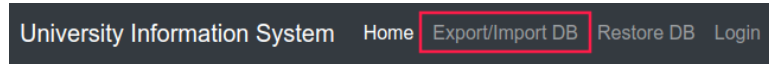
Tabulka	Řádků	Typ	Porovnávání
<code>examination_dates</code>	13	InnoDB	utf8_general_ci
<code>examination_dates_participants</code>	37	InnoDB	utf8_general_ci
<code>grades</code>	0	InnoDB	utf8_general_ci
<code>grade_types</code>	6	InnoDB	utf8_general_ci
<code>hibernate_sequence</code>	5	InnoDB	utf8_general_ci
<code>student_absolved_subjects</code>	0	InnoDB	utf8_general_ci
<code>student_learned_subjects</code>	46	InnoDB	utf8_general_ci
<code>subjects</code>	14	InnoDB	utf8_general_ci
<code>teacher_teaches_subjects</code>	15	InnoDB	utf8_general_ci
<code>users</code>	18	InnoDB	utf8_general_ci

Obrázek 3.1: Struktura databáze v aplikaci UIS

### 3.2.1 Import databáze

Databázi lze importovat tím, že se do aplikace UIS nahraje soubor XML či JSON obsahující data. Postup nahrání souboru je znázorněn na obrázku

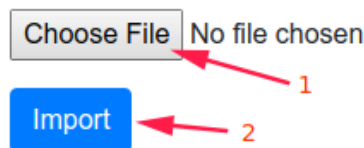
níže. Nejprve je nutné kliknout na stránku *Export/Import DB*. Poté v pravé části stránky lze zvolit v dialogovém okně soubor a tlačítkem *Import* zahájit import.



Obrázek 3.2: Kliknutí na stránku *Export/Import DB*

## Import data into DB

File to import



Obrázek 3.3: Samotné nahrání souboru

Tato služba bude nezbytná pro import vlastních dat pro zátěžové testy.

### 3.2.2 Implementace importu dat

V knihovně Support bylo třeba implementovat funkcionalitu na import a export dat za použití rozhraní PageObject.

Byla vytvořena třída `ExternalDB_Page`, která popisuje webovou stránku, ze které je možné importovat či exportovat data. Tato třída obsahuje dvě metody `isExportDbToFileTitleDisplayed()` a `isImportDataIntoDbTitleDisplayed()` testující, zda oba elementy pro import a export jsou zobrazeny. Dále jsou zde metody `exportDbToXML()`, `exportDbToJson()` a `importFileToDb()`, které se starají o samotný export a import dat. Metoda `importFileToDb()` byla v průběhu vypracování jednou změněna. Před touto změnou metoda pracovala tak, že od uživatele dostala cestu k souboru jako parametr a ten importovala. Na import měla metoda čas 5 vteřin. Po implementování změny může import trvat libovolně dlouhou dobu, to se hodí při importu velkého množství dat.

### 3.2.3 Modul *oracle*

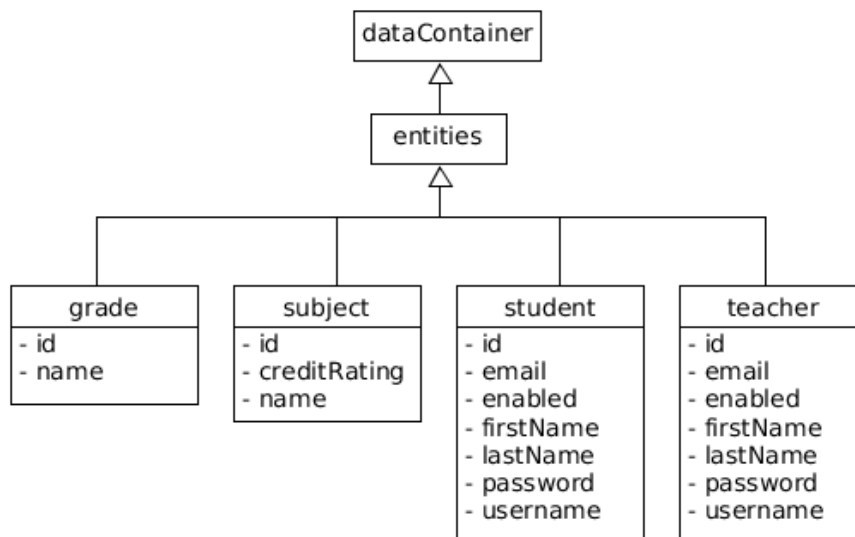
Modul *oracle* si lze představit jako soubor tříd, který představuje „paralelní“ byznys logiku aplikace UIS s úschovou dat v objektech v paměti. Tyto třídy



by měly obsahovat stejná data jako reálná databáze a slouží jako orákulum (tj. hodnota očekávaného stavu) pro vyhodnocování výsledků testů. Data jsou do *oracle* zapisována průběžně a dynamicky za běhu aplikace při každé provedené akci.

### 3.2.4 Struktura databázového souboru XML

V rámci aplikace je nutné zachovat již existující formát XML souboru pro manipulaci s daty. Níže v UML diagramu na obrázku 3.4, je popsána jeho struktura. Tato struktura je zachovaná při generování souboru s větším počtem náhodných dat.



Obrázek 3.4: Struktura XML souboru

Tento soubor lze poté naimportovat i do modulu *oracle* a to ve dvou krocích:

1. přepsání z XML do formátu, který modul *oracle* podporuje:

Oracle využívá k inicializaci své databáze čtyři textové soubory s příponou *.txt*. Jsou to následující soubory:

- *users\_list.txt* – seznam uživatelů v systému (učitelů i studentů); každý řádek označuje jeden záznam, řádek je ve formátu **jméno uživatele;příjmení uživatele**
- *students\_list.txt* – seznam studentů; každý záznam je uvozen řádkem **\$start**, následuje řádek s uživatelským jménem, poté

volitelně řádek ve formátu `zapsaný předmět;zapsaný termín zkoušky` a konečně řádek `$end`, který ukončuje záznam

- *teachers\_list.txt* – seznam učitelů; každý záznam je uvozen řádkem `$start`, následuje řádek s uživatelským jménem, poté volitelně řádek ve formátu `vyučovaný předmět;vypsany termín zkoušky` a konečně řádek `$end`, který ukončuje záznam
- *subjects\_list.txt* – seznam předmětů; každý řádek označuje jeden záznam, řádek je ve formátu `název předmětu;počet kreditů předmětu`

## 2. zavolání statické metody

`OracleInicIALIZATION.restoreOracleStorage()`, která obnoví modul *oracle* na základě výše zmíněných souborů

### 3.3 Seznam uživatelských akcí

V rámci zátěžových testů se budou provádět uživatelské akce, které budou manipulovat s aplikací UIS. Tyto akce se budou provádět náhodně s využitím modifikované metody Monte Carlo. Cílem těchto akcí je simulovat průběh jednoho semestru na vysoké škole. Provedené akce budou tedy vést ke změně dat v databázi, ta se budou kontrolovat oproti očekávanému výsledku, který bude předem daný ještě před zahájením samotných zátěžových testů. Očekávaný výsledek představuje strukturovanou informaci, že konkrétní učitelé vyučují konkrétní předměty. Na ty se zapsali konkrétní studenti a předmět úspěšně či neúspěšně zakončili. Vyhodnocení závěrečného stavu tedy není triviální operace.

Uživatelské akce úzce souvisí s akcemi zmíněnými v sekci 2.1.1. Protože se jedná o simulaci průběhu semestru, budou použity pouze akce, které:

1. provádí uživatel typu student nebo učitel
2. nějakým způsobem ovlivňují stav aplikace (přidávají, mění či mažou záznamy v databázi) – akce, které pouze získávají data za účelem jejich zobrazení, nebudou brány v potaz
3. nemění uživatelské údaje uživatele (tj. křestní jméno, příjmení a email)

Všechny validní uživatelské akce jsou včetně jejich popisu zmíněny níže. Jejich název koresponduje s jejich pojmenováním v programu.

Název akce	Popis
STU_ENROLL_SUBJECT	Student si zapíše nezapsaný předmět.
STU_UNENROLL_SUBJECT	Student si odepíše zapsaný předmět.
STU_REGISTER_EXAM_DATE	Student se přihlásí na vypsání zkouškového termínu.
STU_UNREGISTER_EXAM_DATE	Student se odhlásí ze zkouškového termínu, na který je přihlášený.
TEA_REMOVE_SUBJECT	Učitel si odepíše zapsaný předmět, který učí.
TEA_CANCEL_RANDOM_EXAM_DATE	Učitel zruší náhodný zkouškový termín, který vypsání.
TEA_NEW_EXAM_DATE	Učitel vypíše nový termín zkoušky.
TEA_SET_EVALUATION	Učitel napíše studentovi známku za zkoušku z předmětu.
TEA_SET_NEW_EVALUATION	Učitel napíše studentovi novou známku za nový pokus zkoušky.
TEA_UPDATE_EVALUATION	Učitel změní studentovi již existující známku.
TEA_PARTICIPATE_IN_SUBJECT	Učitel si zapíše předmět, který bude učit.

### 3.3.1 Omezení

S výše zmíněnými akcemi se pojí jistá omezení, akce může být provedena pouze za určitých podmínek. Omezení jsou dvojího typu:

- aplikační či globální – v rámci aplikace existují obecná omezení například o maximálním počtu předmětů, které může učitel učit nebo student studovat
- lokální – jedná se o kontrolu určitých podmínek, které jsou pro každou akci specifické; jako příklad může být třeba pokus učitele o zapsání známky studentovi za zkoušku z předmětu, který student vůbec nestuduje (nemá ho zapsaný)

Limitations regarding university		Limitations regarding users	
Name	Limit	Name	Limit
Maximal number of subjects for students	7	Minimal length of first name	2
Maximal number of subjects for teachers	5	Maximal length of first name	30
Maximal number of teachers for subject	2	Minimal length of last name	2
Maximal number of exam dates for subject	3	Maximal length of last name	30
Minimal number of participants for exam date	1	Minimal length of mail	8
Maximal number of participants for exam date	10	Maximal length of mail	40

Obrázek 3.5: Globální omezení aplikace

Na obrázku 3.5 lze vidět všechna globální omezení v aplikaci UIS.

Aplikace UIS má všechna tato omezení ošetřena, ale přesto je třeba v záťažových testech tato omezení důsledně respektovat. V opačném případě to totiž vede sice k vypsání přehledné chybové hlášky, ze které je ihned patrné, jaký problém nastal, ale zároveň aplikace ukončí svoji činnost, což je nežádoucí vedlejší efekt.

Tento vedlejší efekt je způsoben tím, že knihovna Support, jejíž statické metody se používají k ovládání webové aplikace, má v sobě implementované testy kontrolující správné vykonání příslušné akce. Z tohoto důvodu je třeba předem ošetřit, zda je akci vůbec možné provést a teprve poté metodu z knihovny Support zavolat.

## 3.4 Modifikovaná metoda Monte Carlo

### 3.4.1 Obecné informace

První zmínka o využití této metody je kolem roku 1940, kdy američtí vědci Stanisław Marcin Ulam a John von Neumann zkoumali v laboratoři v Národní laboratoři Los Alamos chování neutronů při průchodu různými materiály.

Metoda byla pojmenovaná po stejnojmenném casinu v Monacu, kde Ulmanovo strýc sázel. Pojmenování souvisí s tím, že metoda Monte Carlo patří mezi stochastické metody a tím, že ruleta v casinu je v podstatě mechanický generátor náhodných čísel.

### 3.4.2 Použití metody

Metoda Monte Carlo není jen jedna metoda. Je to termín, který zastřešuje více přístupů nebo simulací. Tyto přístupy však mají společný prvek, tím je postup, jakým se dá dostat k určitému výsledku:

1. jsou popsána vstupní data – výčet validních vstupů
2. tyto vstupy jsou náhodně vybírány a aplikovány
3. vstupy jsou deterministicky zpracovány a během toho jsou agregovány statistiky
4. výsledkem operace je statistický výsledek

Nevýhodou metody je, že je zapotřebí velké množství statistických dat pro získání výsledků. Z tohoto důvodu bude metoda modifikována a místo pseudonáhodných vstupů budou vstupy usměrňovány za pomoci implementace dynamických pravděpodobností. Tím se zajistí částečné upřednostnění těch vstupů, které v danou chvíli nejlépe usměrní průběh operace k cíli. Cílem je v tomto případě myšlen dokončený průběh jednoho semestru.

## 3.5 Externí testovací nástroj JMeter

Jako externí nástroj pro testování a tvorbu automatických testů byl zvolen JMeter od Apache. Nástroj je psaný v programovacím jazyce Java a je open-source. JMeter měří výkonnost webových stránek a dokáže uměle vytvářet zátěž na dané stránce tím, že simuluje posílání požadavků od více klientů v nějakém časovém intervalu. Kromě toho dokáže testovat zátěž na databázi SQL a podporuje protokoly jako HTTP, FTP, LDAP, SOAP a další.

Velkou výhodou nástroje JMeter je, že je multiplatformní, snadno se instaluje a není potřeba mít předchozí znalost konkrétního programovacího jazyka.

# 4 Implementace

## 4.1 Prerekvizity

Aplikace *TbUIS-LoadTests* je část projektu TbUIS, kde jsou implementovány zátěžové testy.

K sestavení projektu se používá nástroj Maven, který si za pomoci souboru `pom.xml` stáhne a doplní externí knihovny a závislosti.

### 4.1.1 Použité závislosti

Kromě knihovny *Log4j*, která se stará o logování do textových souborů, jsou v projektu použité tyto závislosti:

#### Knihovna Support

```
<dependency>
  <groupId>cz.zcu.kiv</groupId>
  <artifactId>uis-supportLibrary</artifactId>
  <version>${uis.supportLibrary.version}</version>
</dependency>
```

Knihovna Support obsahuje projekt `support`, který zprostředkovává důležité komponenty:

- zajišťuje komunikaci s webovou aplikací za použití rozhraní *PageObject*
- obsahuje testy, které kontrolují konzistenci databáze tím, že porovnávají data ve webové aplikaci a v orákulu
- loguje svoji činnost do textových souborů

#### Jackson (core, databind, annotations)

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-core</artifactId>
  <version>2.10.0</version>
</dependency>
<dependency>
```



```
<groupId>com.fasterxml.jackson.core</groupId>
<artifactId>jackson-databind</artifactId>
<version>2.10.0</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-annotations</artifactId>
  <version>2.10.0</version>
</dependency>
```

Tyto tři externí knihovny patří do projektu *Jackson Project* [2]. Mají na starosti konverzi z Java objektu na JSON soubor a zajišťují naopak jednoduché čtení a procházení JSON souborů a extrahování jejich dat do Java objektů.

Tyto knihovny jsou v projektu použity na čtení souboru, který obsahuje databázová data, jichž by aplikace měla dosáhnout.

## Java Faker

```
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>javafaker</artifactId>
  <version>1.0.2</version>
</dependency>
```

Knihovna *Java Faker* [1] slouží ke generování náhodných dat při generování obsahu databáze. Dokáže vytvářet data po vzoru *Lorem ipsum* generátoru [6], ale poskytuje možnost vytvářet data určitého typu, například jména, příjmení, adresy, názvy knížek a další. Knihovna je v projektu využita hlavně pro usnadnění čitelnosti vygenerovaných dat.

### 4.1.2 Konfigurační soubor

Pro správný chod aplikace *TbUIS-LoadTests* je třeba mít správně pojmenovaný a vyplněný konfigurační soubor `configurations.txt`. Tento soubor obsahuje tzv. *properties* (vlastnosti), které nastavují například typ webového prohlížeče použitého k testování, cestu k ovladačům prohlížečů (Google Chrome, Firefox či Opera), inicializační webovou stránku, která se otevře po spuštění aplikace a další.

## 4.2 Struktura projektu

Struktura projektu je následující:

```
uis
|
-- constants -- Constants
|
-- database_generator -- DatabaseGenerator
|
-- json_configuration -- JsonConfiguration
|
|               -- JsonLoader
|               -- JsonWriter
|               -- ResultChecker
|
-- main -- Main
|
-- oracle -- OracleUtils
|
-- results -- Results
|
-- user_action -- UserActions
|
|               -- UserActionsProbabilities
```

V kořenovém adresáři projektu se nacházejí soubory, které jsou důležité pro chod aplikace. Jsou zde všechny logovací soubory, jejichž název je zakončen písmeny *log.txt* (například *test-config-log.txt* nebo *test-results-log.txt*). Je zde konfigurační soubor *configurations.txt* a konfigurační soubor projektového loggeru *logger-config.xml*. Dále jsou používány dva typy důležitých souborů:

1. Soubory typu XML pojmenované podle vzoru `imported-file-A-B-C.xml`, kde *A* v názvu souboru udává počet předmětů, *B* udává počet učitelů a *C* udává počet studentů. Jsou to soubory s databázovými daty, přičemž při spuštění aplikace je ve třídě s konstantami zvolen právě jeden z nich. Tyto soubory lze vygenerovat pomocí nástroje `DatabaseGenerator` (více v sekci 4.2.3).
2. Soubory typu JSON pojmenované podle vzoru `expected_result-A-B-C.json`, kde *A* v názvu souboru udává počet předmětů, *B* udává počet učitelů a *C* udává počet studentů. Jedná se o manuálně vytvořený soubor s očekávaným výsledkem. Tento soubor musí být ručně vytvořený, vyplněný daty a správně pojmenovaný (podle generovaného

souboru) při každém generování souboru `imported-file-A-B-C.xml`. Údaje z tohoto souboru jsou při spuštění aplikace načteny a při jejím běhu průběžně porovnávány s aktuálními daty v modulu *oracle*.

### 4.2.1 Třída Main

Třída `uis.main.Main` je hlavní třídou programu. Stará se o chod zátěžového testu a o logování jednotlivých akcí. Metoda je rozdělena do dalších tří metod a tím do pomyslných částí – `setup()`, `mainLoop()` a `tearDown()`.

#### Metoda `setup()`

Stará se o přípravu prostředí k tomu, aby mohl test probíhat co nejrychleji. Také se tam nastavují proměnné využívané při logování akcí. Inicializuje se tam instance `mainLoopCounter`, která slouží k identifikaci jednotlivých iterací. Obnovují se data v modulu *oracle* na základě aktuálně zvoleného souboru s daty, který se bude importovat do databáze (více o tomto procesu v sekci 3.2.4). Zavolá se v knihovně Support metoda

`Configurations.getInstance().setConfiguration()` a v ní se na základě konfiguračního souboru `configurations.txt` nastaví kromě logovacích nástrojů i proměnné, které využívá Selenium pro komunikaci s webovým prohlížečem, následně se v ní obnoví data v modulu *oracle*, která se nacházejí v textových souborech. V poslední fázi příprav se do paměti načtou ze souboru `expected_result-A-B-C.json` data s očekávaným stavem obsahu databáze v modulu *oracle*.

#### Metoda `mainLoop()`

Má na starosti samotný zátěžový test. Nejdůležitější částí této metody je `switch`, který na základě vybrané akce zavolá metodu, která se o její obsluhu stará. Každá větev v konstrukci `switch` se stará o jednu z uživatelských akcí, celkem je jich 11 – 7 učitelských akcí a 4 studentské akce. Tyto akce jsou více popsány v samostatné sekci 4.3. Každá z akcí je logována, přičemž z logu je zřejmé, zda se akce povedla či nikoliv. Příklad úspěšně provedené akce vypadá takto:

```
INFO : 73/2000: TEA participating in subject
INFO : Teacher "Leroy Glover36" participating in subject
      "Odit".
```

První řádek má formát `číslo_akce/maximální_počet_iterací: Typ akce`. V druhém řádku je detailnější popis toho, co se v dané akci provádí.

Příklad neúspěšně provedené akce:

```
INFO : 62/2000: STU unenrolling subject
WARN : Student "Elouise Wiegand90" does not study the
       subject "Et nostrum". Subject cannot be removed.
```

Také jsou zde průběžně kontrolovány výsledky (tím je myšleno porovnání očekávaných dat s daty modulu *oracle*), o to se stará třída `ResultChecker`, jejíž metoda `checkAgainst()` je zavolána každou  $N$ -tou iterací, kde  $N$  je číslo specifikované ve třídě s aplikačními konstantami `constants.Constants`.

### Metoda `tearDown()`

Metoda zavře webový prohlížeč a postará se o vypsání výsledků. Výsledky jsou trojího typu:

1. logování základních statistik
2. podrobné výsledky v souboru typu CSV (vykonává třída `Results`)
3. průnik reálných a očekávaných dat v souboru typu JSON (vykonává třída `Results`)

Základní přehled výsledků operace lze najít na konci logovacího souboru. Přehled je rozdělen do dvou částí. V první části je vypsána statistika pokusů o provedení jednotlivých akcí. Akce se však nemusí z mnoha důvodů provést, a tak je dále vypsána statistika reálně provedených akcí. Pod ní je vypsán celkový počet pokusů o provedení akce a reálný počet provedených akcí. Posledním zobrazeným údajem je celkový uplynulý čas operace. Výpis vypadá následnově:

```
INFO : -----
INFO : ----- Statistics: attempts for actions -----
INFO : 1. Teacher's actions:
INFO : TEA_PARTICIPATE_IN_SUBJECT: 108 times
INFO : TEA_REMOVE_SUBJECT:         46 times
INFO : TEA_NEW_EXAM_DATE:          1250 times
INFO : TEA_CANCEL_RANDOM_EXAM_DATE: 70 times
INFO : TEA_SET_EVALUATION:         45 times
INFO : TEA_SET_NEW_EVALUATION:     53 times
INFO : TEA_UPDATE_EVALUATION:      68 times
INFO : 2. Student's actions:
INFO : STU_UNENROLL_SUBJECT:       71 times
INFO : STU_ENROLL_SUBJECT:        175 times
```

```

INFO : STU_REGISTER_EXAM_DATE: 57 times
INFO : STU_UNREGISTER_EXAM_DATE: 57 times
INFO : -----
INFO : ----- Statistics: successful actions -----
INFO : 1. Teacher's actions:
INFO : TEA_PARTICIPATE_IN_SUBJECT: 58 times
INFO : TEA_REMOVE_SUBJECT: 2 times
INFO : TEA_NEW_EXAM_DATE: 62 times
INFO : TEA_CANCEL_RANDOM_EXAM_DATE: 5 times
INFO : TEA_SET_EVALUATION: 0 times
INFO : TEA_SET_NEW_EVALUATION: 0 times
INFO : TEA_UPDATE_EVALUATION: 0 times
INFO : 2. Student's actions:
INFO : STU_UNENROLL_SUBJECT: 0 times
INFO : STU_ENROLL_SUBJECT: 97 times
INFO : STU_REGISTER_EXAM_DATE: 0 times
INFO : STU_UNREGISTER_EXAM_DATE: 0 times
INFO : -----
INFO : Actions summary: 2000 attempts, 224 successful actions
      performed
INFO : Elapsed time: 626.802 seconds, which is 10 minutes and
      26.802 seconds
INFO : -----

```

## 4.2.2 Balík `json_configuration`

Obsahuje tři třídy, které čtou či slouží ke čtení ze souboru s očekávaným stavem databáze `expected_result_A-B-C.json`. Balík k tomu využívá externí knihovnu *Jackson* [2]. Soubor s očekávaným stavem může vypadat následovně:

```

[ {
  "subject" : "Operating Systems",
  "teacher" : "strict",
  "stu_with_grades" : [ {
    "student" : "orange",
    "passed" : true
  }, {
    "student" : "purple",
    "passed" : false
  }, {
    "student" : "magenta",

```

```

        "passed" : true
    }]
}, {
    "subject" : "Computation Structures",
    "teacher" : "scatterbrained",
    "stu_with_grades" : [{
        "student" : "orange",
        "passed" : true
    }, {
        "student" : "purple",
        "passed" : false
    }]
}]
}]

```

Obsahuje výčet podmínek, které musí být splněny. Pokud nastane při kontrole dat situace, kdy jsou všechny podmínky splněny, aplikace ukončí svoji činnost a zobrazí výsledky. Jednotlivé podmínky představující jeden konkrétní předmět jsou ohraničeny složenými závorkami, které jsou ve výše uvedené ukázce zobrazeny červenou barvou.

### Třída `JsonConfiguration`

Tato třída slouží pouze jako přepravka, má vnitřně stejnou strukturu jako jedna podmínka v souboru s očekávanými výsledky. Jejími atributy jsou:

```

private String subject;
private String teacher;
@JsonProperty("stu_with_grades")
private List<StudentWithGrades> studentWithGradesList;

```

Musí být pojmenovány stejně jako v JSON souboru, aby mohly být ze souboru snadno načteny do přepravky. Atribut `subject` uchovává v textové podobě název předmětu, `teacher` uchovává v textové podobě přihlašovací jméno učitele a `studentWithGradesList` je seznam dvojic, ve kterých je student a jeho úspěšnost z daného předmětu. Anotace `@JsonProperty` umožňuje spárovat položku v souboru s atributem, který je jinak pojmenován. V této anotaci musí být uvedeno pojmenování použité v JSON souboru.

Třída `JsonConfiguration` obsahuje vnitřní třídu `StudentWithGrades`, která představuje jednoho studenta (atributem je textová reprezentace přihlašovacího jména studenta) a jeho úspěšnost z předmětu (atributem je booleovského typu a udává, zda student uspěl či neuspěl).

## Třída `JsonLoader`

Tato třída má pouze jednu statickou metodu `loadJsonFile()`, která načte data z JSON souboru do přepravky `JsonConfiguration`. Používá k tomu třídu `ObjectMapper` z knihovny *Jackson* [2], který je schopný namapovat data z JSON souboru na Java objekt.

## Třída `JsonWriter`

Má pouze jednu statickou metodu `writeJsonFile()`, která přečte data z Java objektu, který je namapovaný na přepravku `JsonConfiguration`, a uloží je do souboru JSON. Používá k tomu třídu `ObjectMapper` z knihovny *Jackson* [2].

## Třída `ResultChecker`

Ve třídě jsou dvě metody starající se o porovnání dat aktuálních a očekávaných. První metoda porovnávající shodu aktuálních a očekávaných dat má následující signaturu:

```
public boolean checkAgainst(  
    List<JsonConfiguration> expectedResult,  
    Map<String, Teacher> teacherMap,  
    Map<String, Student> studentMap,  
    Map<String, Subject> subjectMap)
```

Parametr `expectedResult` obsahuje data načtená z JSON souboru s očekávaným stavem databáze. Další tři parametry `teacherMap`, `studentMap` a `subjectMap` obsahují data o všech učitelích, studentech a předmětech. Z mapy lze získat učitele či studenta podle jeho přihlašovacího jména a předmět podle jeho názvu. Z těchto objektů lze získávat další informace – například zjistit seznam předmětů, které učitel učí, nebo získat zapsané zkouškové termíny studenta z nějakého předmětu. Tato data jsou získávána z modulu *oracle*, který by měl poskytovat aktuální data aplikace.

Metoda porovná tato data s obsahem souboru JSON a v případě splnění všech podmínek vrátí booleovskou hodnotu `true` a aplikace ukončí svoji činnost. V opačném případě vrátí `false`.

Druhá metoda `compareExpectedWithActual()` má stejné parametry jako metoda `checkAgainst()`, ale vrací seznam objektů `JsonConfiguration`. Metoda prochází soubor s očekávanými daty a porovná je s aktuálními daty. V případě shody zapíše tato data do objektů `JsonConfiguration`. Dále počítá počet kritérií, který je třeba splnit, a počet splněných kritérií.

### 4.2.3 Třída DatabaseGenerator

Tato třída je samostatný nástroj vyvinutý za účelem generování většího počtu dat v databázi. Výstupem je soubor typu XML mající stejnou strukturu jako již existující soubor s počátečními daty databáze. Struktura tohoto souboru byla popsána v sekci 3.2.4.

Nástroj využívá knihovnu *Java Faker* [1], která dokáže generovat různé typy textových řetězců. Uživatel může změnit velikost databáze a název výsledného souboru změnou těchto konstant:

```
/** Initial number of subjects */
public static final int INIT_DB_FILE_NUM_OF_SUBJECTS = 375;

/** Initial number of teachers */
public static final int INIT_DB_FILE_NUM_OF_TEACHERS = 75;

/** Initial number of students */
public static final int INIT_DB_FILE_NUM_OF_STUDENTS = 5625;

/** Name of the generated file */
public static final String IMPORTED_FILE_GENERATED_NAME =
    "imported-file";
```

Výsledný název souboru se skládá z první části jména souboru, zde *imported-file*, za kterou následují tři čísla ve formátu *-A-B-C*, kde *A* je počet předmětů, *B* je počet učitelů a *C* je počet studentů. Nakonec je připojena přípona souboru *.xml*.

### 4.2.4 Třída OracleUtils

Třída obsahuje pouze jednu metodu se signaturou `public static void refreshOracle(String importedFilePath)`. Její jediný parametr udává cestu k souboru s počátečními daty v databázi.

Metoda převede tento soubor do formátu, který využívá modul *oracle* pro obnovení svých dat. Výstupem jsou čtyři soubory typu *.txt*. Formát těchto čtyř souborů je popsán v sekci 3.2.4.

### 4.2.5 Třídy UserActions a UserActionsProbabilities

Tyto třídy se starají o uživatelské akce a ovlivňují program na základě toho, jaká uživatelská akce byla zvolena pro zpracování.



Zpočátku existovala pouze třída `UserActions`, která byla výčtovým typem obsahujícím jednotlivé uživatelské akce. Akce byly vybírány náhodně. V průběhu vypracování se však přišlo na to, že náhodný výběr nekonverguje a je třeba akcím přiřadit určitou váhu a na základě toho je vybírat. Hlubší popis této problematiky je popsán v sekci 4.3.

### Třída `UserAction`

Třída `UserAction` obsahuje vnitřní třídu `UserActionsEnum`, která je výčtovým typem a jsou v ní jednotlivé uživatelské akce.

Má dvě metody pro získávání uživatelských akcí: `getRandomAction()` a `getRandomActionBasedOnProbability()`. První zmíněná metoda vybere náhodně akci a tu se aplikace pokusí vykonat. Druhá zmíněná metoda vybere náhodně akci s tím, že je při výběru přihlédnuto k aktuální pravděpodobnosti (váze) akce. Tyto váhy jsou definovány v textovém souboru *probability-config.txt*, jehož struktura může vypadat takto:

```
TEA_PARTICIPATE_IN_SUBJECT, 80, 80
STU_ENROLL_SUBJECT, 80, 80
TEA_NEW_EXAM_DATE, 70, 80
STU_REGISTER_EXAM_DATE, 60, 80
TEA_SET_EVALUATION, 50, 80
```

Default

```
STU_ENROLL_SUBJECT, 9
STU_UNENROLL_SUBJECT, 9
STU_REGISTER_EXAM_DATE, 9
STU_UNREGISTER_EXAM_DATE, 9

TEA_REMOVE_SUBJECT, 9
TEA_CANCEL_RANDOM_EXAM_DATE, 9
TEA_NEW_EXAM_DATE, 9
TEA_SET_EVALUATION, 10
TEA_SET_NEW_EVALUATION, 9
TEA_UPDATE_EVALUATION, 9
TEA_PARTICIPATE_IN_SUBJECT, 9
```

Ze souboru jsou kvůli přehlednosti vyjmuty řádky s komentářem. Přehled všech uživatelských akcí a jejich popis je v sekci 3.3

První část souboru (až po řádek `Default`) se skládá z řádků, z nichž každý popisuje uživatelskou akci, za níž následuje číslo určující pravděpo-

dobnost, že bude tato akce vybrána (v procentech). Ostatním akcím je rovnoměrně rozdělena zbývající pravděpodobnost. Poslední číslo (také v procentech) určuje hranici, při níž se pravděpodobnosti upraví podle dalšího řádku v souboru. V této části souboru jsou uvedeny jen akce, které jsou stěžejní pro průběh simulace jednoho semestru. Ostatní méně důležité akce se vykonávat můžou, pravděpodobnost jejich výběru je ale malá.

Příklad uvedený výše tedy vykoná následující:

1. S 80% pravděpodobností je vybrána akce, kdy si učitel zapisuje předmět. Pokud je napříč všemi učiteli zapsáno 80 % předmětů, upraví se pravděpodobnosti podle dalšího řádku v souboru (tato akce bude poté mít pravděpodobnost pouze 2 %).
2. S 80% pravděpodobností si student zapisuje předmět. Pokud je napříč všemi studenty zapsáno 80 % předmětů, upraví se pravděpodobnosti.
3. S 70% pravděpodobností učitel vypisuje zkouškový termín. Pokud je napříč všemi učiteli vypsáno 80 % zkouškových termínů (počet možných termínů je omezený globálním omezením 3 termínů na předmět), upraví se pravděpodobnosti.
4. Studenti se s 60% pravděpodobností přihlašují ke zkouškovému termínu. Pokud je zaplněno 80 % z počtu vypsanych zkouškových termínů, upraví se pravděpodobnosti.
5. Učitelé s 50% pravděpodobností udělují známku studentům. Pokud je oznámkováno 80 % studentů přihlášených na zkouškový termín, pravděpodobnosti se dále upraví podle definovaného výchozího stavu v druhé části souboru *probability-config.txt*.

Druhá část souboru obsahuje na každém řádku jednu akci a číslo, které udává pravděpodobnost jejího výběrů (v procentech). V této části souboru jsou uvedeny všechny akce.

### **Třída `UserActionsProbabilities`**

Tato třída se stará o načtení pravděpodobností ze souboru *probability-config.txt* do dvou map:

```
private Map<String, Integer[]> probabilitiesMap;  
private Map<String, Integer> defaultProbabilitiesMap;
```

Tyto mapy jsou poté využívány k nastavení hodnot pravděpodobností ve třídě `UserActions` pro jednotlivé akce metodami `setProbabilitiesArray()` či `setDefaultProbabilitiesArray()`.

#### 4.2.6 Třída `Results`

Zajišťuje vypsání výsledků do souborů JSON a CSV.

V JSON souboru je vypsán průnik dat aktuálních a očekávaných. Formát dat v souboru je stejný, jako v souboru s očekávaným stavem databáze, který je do aplikace nahrán na začátku testu.

Před zapsáním výsledku do JSON souboru je zavolána metoda `compareExpectedWithActual()` ze třídy `ResultChecker` a postupným procházením souboru s očekávaným stavem databáze je vyhodnoceno, jak se data aktuální shodují s daty očekávanými. Metoda poté vrátí objekt obsahující průnik dat. Tento objekt je poté použit k vytvoření výsledného JSON souboru. Metoda kromě vytvoření objektu také spočítá, kolik částí ze souboru s očekávanými daty bylo splněno. Pro tuto situaci bylo zavedeno *kritérium*. Očekávaná data jsou rozdělena na jednotlivé podmínky (více v sekci 4.2.2) a tyto podmínky jsou rozdělené na kritéria. Takto může vypadat jedna podmínka ze souboru s očekávanými daty:

```
{
  "subject" : "Operating Systems",
  "teacher" : "strict",
  "stu_with_grades" : [{
    "student" : "orange",
    "passed" : true
  }, {
    "student" : "purple",
    "passed" : true
  }, {
    "student" : "magenta",
    "passed" : true
  }]
}
```

Ve výše zobrazené ukázce jedné podmínky jsou červenou barvou zobrazena jednotlivá kritéria, tato ukázka jich má 7. Zmíněná metoda `compareExpectedWithActual()` kromě porovnání dat spočítá očekávaná kritéria a splněná kritéria. Tato čísla se poté objeví ve výsledných statistikách.

Výsledné statistiky jsou zapsány do souboru CSV. Ten může vypadat následovně:

```
General info =====
File information,14-6-12n1
Number of iterations,500
Maximum number of iterations,500
Iterations between checking result,20
Probabilities activated (ON/OFF),false
Time seconds,87.208156997
Time minutes and seconds,1,27.208156997000003
Total actions,500
Successful actions,55
Successful actions rate [%],11.0

Expected result =====
Number of criteria,32
Number of successful criteria,3
Successful criteria rate [%],9.375

Actions (attempts) =====
TEA_PARTICIPATE_IN_SUBJECT,50
TEA_REMOVE_SUBJECT,53
TEA_NEW_EXAM_DATE,43
TEA_CANCEL_RANDOM_EXAM_DATE,46
TEA_SET_EVALUATION,41
TEA_SET_NEW_EVALUATION,53
TEA_UPDATE_EVALUATION,44
STU_ENROLL_SUBJECT,32
STU_UNENROLL_SUBJECT,45
STU_REGISTER_EXAM_DATE,51
STU_UNREGISTER_EXAM_DATE,42

Actions (success) =====
TEA_PARTICIPATE_IN_SUBJECT,25
TEA_REMOVE_SUBJECT,5
TEA_NEW_EXAM_DATE,7
TEA_CANCEL_RANDOM_EXAM_DATE,1
TEA_SET_EVALUATION,0
TEA_SET_NEW_EVALUATION,0
TEA_UPDATE_EVALUATION,0
STU_ENROLL_SUBJECT,16
STU_UNENROLL_SUBJECT,0
```

STU_REGISTER_EXAM_DATE,1 STU_UNREGISTER_EXAM_DATE,0
--

V první části se nachází obecné informace o souboru. Položka *File information* je odvozena z názvu souboru, který je v tomto případě *result-14-6-12n1.csv*. Stejným způsobem je pojmenován i JSON soubor s průnikem dat. První tři čísla v názvu určují stejně jako u souboru pro import dat a u souboru s očekávanými daty trojici čísel  $A$ ,  $B$  a  $C$ , kde  $A$  je počet předmětů,  $B$  je počet učitelů a  $C$  je počet studentů v databázi. Poslední dvě písmena *n1* jsou zkratkou z *number 1*, která říká, že to je první výsledný soubor v adresáři. Testy lze použít vícekrát za sebou, přičemž data nejsou v souborech CSV a JSON přepsána, ale je vytvořena nová dvojice souborů s pořadovým číslem o jedno vyšší. Dále zde vidíme provedení a maximální počet iterací v testu, počet iterací, po kterých se provede kontrola skutečného oproti očekávanému stavu, informaci o tom, zda byly použity pravděpodobnosti při generování náhodných akcí, uplynutý čas a počet úspěšně provedených akcí. Posledním údajem v této části je v procentech vyjádřený podíl úspěšných a neúspěšných pokusů o provedení akce.

V druhé části je vypsán počet kritérií, která byla očekávána, a počet splněných kritérií. Pod nimi je v procentech vypsán jejich podíl.

Ve třetí části je seznam akcí s počtem pokusů o jejich provedení následovaný stejným seznamem akcí s počtem úspěšných pokusů o jejich provedení.

## 4.2.7 Další soubory

Všechny níže vyjmenované soubory či složky se nachází v kořenovém adresáři projektu.

### Soubory *expected\_result* a *imported-file*

Soubory začínající písmeny *expected\_result* jsou JSON soubory s očekávaným stavem databáze. Soubory začínající písmeny *imported-file* jsou soubory, ze kterých se importují data do databáze před spuštěním testu.

### Logovací soubory

Tyto textové soubory vždy končí písmeny *-log.txt*. Důležité jsou zejména *test-config-log.txt* obsahující informace z načítání konfiguračního souboru a *test-results-log.txt* obsahující logy všech akcí (úspěšně i neúspěšně provedených).

## Databázové soubory modulu *oracle*

Jedná se o soubory *students\_list.txt*, *subjects\_list.txt*, *teachers\_list.txt* a *users\_list.txt*. Tyto čtyři soubory slouží modulu *oracle* k načítání dat o všech uživateliích a všech předmětech.

### Složka *results*

Obsahuje dvojice souborů CSV a JSON, ve kterých jsou vypsané výsledky jednoho testu.

## 4.3 Uživatelské akce a jejich testy

Uživatelských akcí je celkem 11. U každé z nich je před jejím provedením náhodně vybrán učitel, student i předmět. U této trojice je pak zkontrolováno, zda v dané situaci, vztahující se k provedení dané akce, dává smysl.

Níže je na obrázku 4.1 vidět tabulka s přehledem akcí a testů, které je třeba provést před tím, než se zavolá metoda z knihovny Support. Testy mají charakter filtru. V případě, že daná podmínka vrátí negativní odpověď, nemá smysl provádět další testy a daná akce se neprovede. Celkem je třeba testovat 42 podmínek, z nichž 25 je unikátních. U některých akcí je testování podobné či stejné, příkladem mohou být akce `TEA_SET_EVALUATION`, `TEA_SET_NEW_EVALUATION` a `TEA_UPDATE_EVALUATION`. Pomocí ID jednotlivého testu lze v obrázku snadno vyhledat opakující se testy. Modře zvýrazněný popis testu znamená, že se jedná o globální omezení (více o omezeních v sekci 3.3.1).

## 4.4 Problémy při vypracování

V průběhu vypracování bakalářské práce se narazilo na překážky, které omezovaly nebo znemožňovaly další práci. Některé z nich se podařilo odstranit.

### 4.4.1 Problém s pamětí v prohlížeči Firefox

Zpočátku se všechny testy pouštěly v prohlížeči Mozilla Firefox, přičemž se neobjevily žádné komplikace. Po zvětšení počtu provedených iterací se po více jak 5000 provedených iteracích objevila chyba, že došla paměť a aplikace ukončila svoji činnost. Chybu hlásil programovací jazyk JavaScript:

```
JavaScript error: , line 0: uncaught exception: out of memory
```

Název akce	ID	Popis
TEA_PARTICIPATE_IN_SUBJECT	1	Test max num of subject per teacher
	2	Teacher already teaches subject
	3	Test max num of teacher per subject
TEA_REMOVE_SUBJECT	4	Teacher does not teach
	5	Students are studying the subject
	6	There are exam dates
TEA_NEW_EXAM_DATE	4	Teacher does not teach
	7	Test max num of exam dates
	8	This exam date already exist
TEA_CANCEL_RANDOM_EXAM_DATE	9	There are no exam dates
	10	Students registered to exam date
	11	Teacher did not create exam date
TEA_SET_EVALUATION	12	Student does not study
	9	There are no exam dates
	13	Student not registered to exam date
	14	Student registered to exam date at other teacher
	15	Test max num of grades per student for subject
	16	Student already has a grade for this exam date
TEA_SET_NEW_EVALUATION	12	Student does not study
	9	There are no exam dates
	13	Student not registered to exam date
	14	Student registered to exam date at other teacher
	15	Test max num of grades per student for subject
	16	Student already has a grade for this exam date
TEA_UPDATE_EVALUATION	12	Student does not study
	9	There are no exam dates
	13	Student not registered to exam date
	14	Student registered to exam date at other teacher
	17	Student does not have a grade yet
STU_ENROLL_SUBJECT	18	There are no teachers teaching subject
	19	Test max num of subjects per student
	20	Student already studying the subject
STU_UNENROLL_SUBJECT	12	Student does not study
	21	Student has a grade for subject
STU_REGISTER_EXAM_DATE	12	Student does not study
	9	There are no exam dates
	22	Student already registered at this time at other subject
	23	Exam date full
	24	Student already registered for other exam date at same subject
STU_UNREGISTER_EXAM_DATE	12	Student does not study
	25	Student does not have any registered exam date for subject
	16	Student already has a grade for this exam date

Obrázek 4.1: Přehled uživatelských akcí a jejich testů

Problém se nepodařilo odstranit a bylo třeba přejít na jiný webový prohlížeč.

#### 4.4.2 Problém s ovladačem Google Chrome

V prohlížeči Google Chrome často nastává hned na začátku při importu dat situace, kdy se na webové stránce správně vyplní soubor s daty, který se má importovat, ale neprovede se kliknutí na tlačítko *Import* a aplikace čeká

nekonečně dlouhou dobu na provedení této akce.

Druhá situace vedoucí k ukončení činnosti aplikace nastává při přihlašování uživatele. Stránka s formulářem pro vyplnění přihlašovacích údajů se správně načte, ovladač webového prohlížeče však není schopen vyplnit údaje a aplikace po vypršení časové lhůty ukončí svoji činnost. Problém se opět nepodařilo odstranit, protože se jedná o problém ovladače.

## 4.5 Testování

Testy jsou pouštěny nad různým počtem dat. Data vycházejí z původního databázového souboru se 14 předměty, 6 učiteli a 12 studenty. Tento soubor je v testech také použit. Při generování většího množství dat byl zaveden poměr *předmět:učitel:student* 5:1:75, tj. průměrně 15 studentů na předmět. Testování probíhalo na těchto konfiguracích databáze:

- 14 předmětů, 6 učitelů, 12 studentů
- 60 předmětů, 12 učitelů, 900 studentů
- 125 předmětů, 25 učitelů, 1875 studentů
- 250 předmětů, 50 učitelů, 3750 studentů
- 375 předmětů, 75 učitelů, 5625 studentů
- 500 předmětů, 100 učitelů, 7500 studentů

Tento seznam byl pomyslně rozdělen na dvě části podle množství dat, první tři soubory byly přiřazeny do první části a další tři soubory do druhé. V každé z těchto částí se testovalo nastavení různého počtu maximálních iterací a zapnutí či vypnutí dynamických pravděpodobností při náhodném výběru uživatelské akce.

V první části se testovalo 500, 1000, 2000, 5000 a 10000 iterací. V druhé části se testovalo pouze 500, 1000 a 2000 iterací. Počet iterací byl otestován se zapnutými a vypnutými pravděpodobnostmi. Zde je třeba vzít v úvahu značnou časovou náročnost testu, kdy 1000 iterací při použití největšího databázového souboru a zapnutí pravděpodobností trvalo přibližně 116 minut a 43 sekund.

Všechny testy byly provedeny třikrát pro získání co nejobjektivnějších výsledků. Celkem bylo provedeno 144 testů.

V průběhu všech běhů všech testů nebyl zjištěn pád aplikace (viz též podrobně popis problémů v části 4.4). Takže hlavní účel tohoto specifického



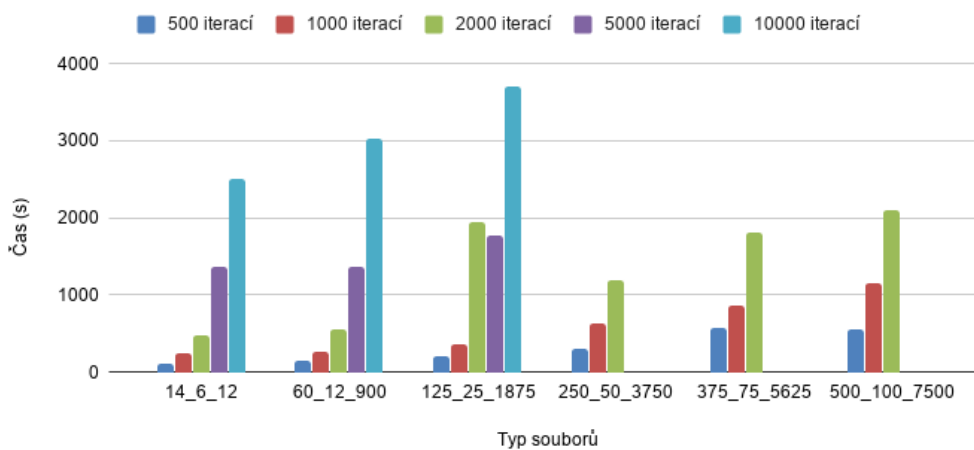
zátěžového testu byl splněn, neboť bylo prokázáno, že aplikace UIS odolá i zvýšené zátěži.

#### **4.5.1 Dosažené výsledky**

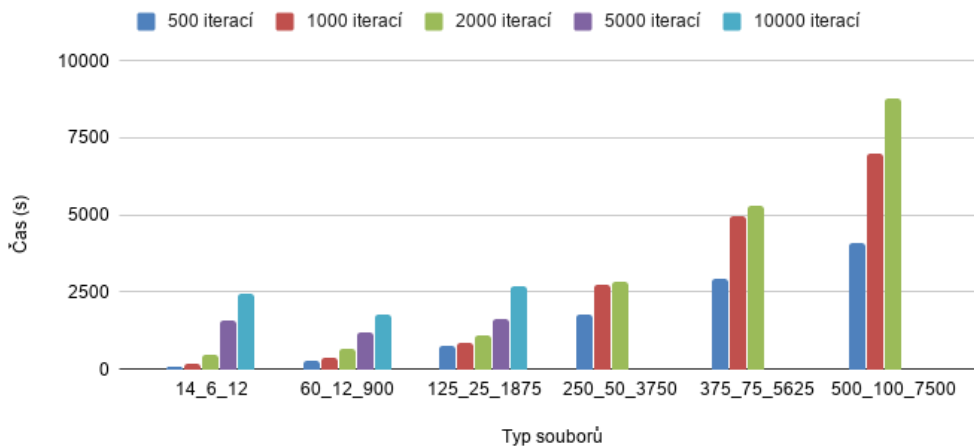
Všechny zátěžové testy byly provedeny na stroji Dell Inspiron 5558, Intel Core i7-5500U @ 4x 2,4GHz, 16GB RAM, operační systém Linux Manjaro 20.0 Lysia.

Bylo provedeno 48 zátěžových testů s tím, že každý z nich byl proveden třikrát pro získání co nejobektivnějších výsledků. Níže na obrázcích 4.2, 4.3 a 4.4 je zobrazeno několik grafů s přehlednými výsledky, tyto údaje však nejsou hlavním cílem této práce. Hlavním cílem bylo ověření, že aplikace tomuto objemu dat a delšímu času běhu odolá. Vzhledem k tomu, že všechny zátěžové testy proběhly v pořádku bez chyb, lze konstatovat, že byl cíl splněn – viz též dále shrnutí za grafy.

### Časová náročnost (pravděpodobnosti vypnuté)

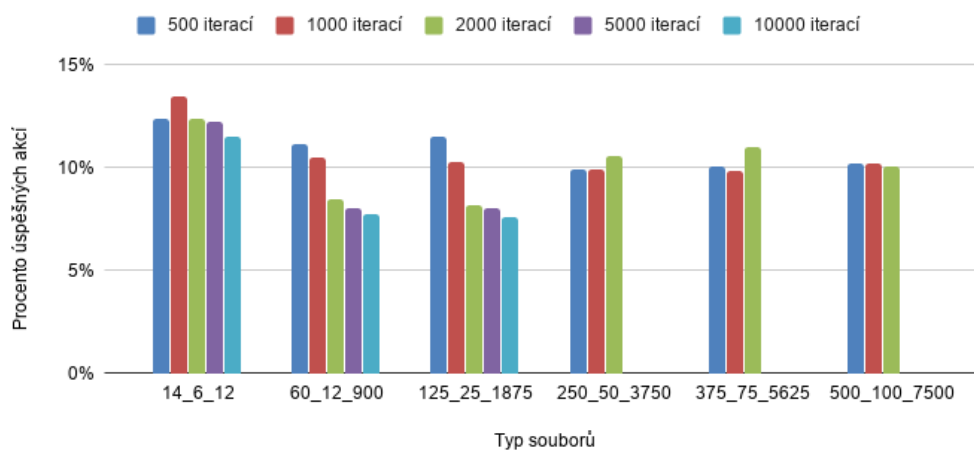


### Časová náročnost (pravděpodobnosti zapnuté)

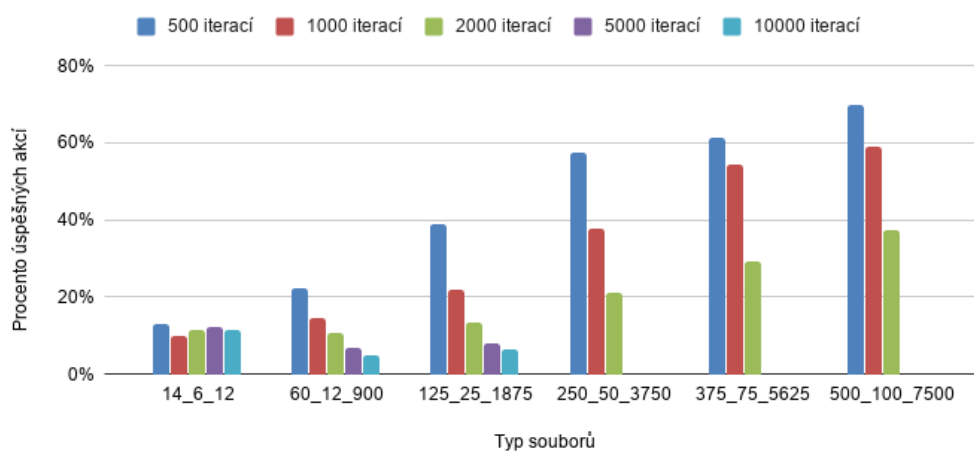


Obrázek 4.2: Časová náročnost testů

Poměr úspěšných akcí ku provedeným (pravděpodobnosti vypnuté)

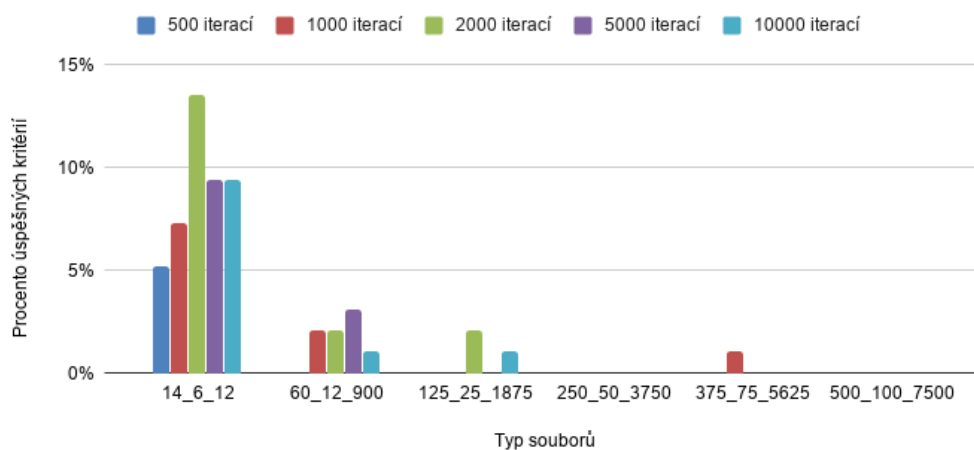


Poměr úspěšných akcí ku provedeným (pravděpodobnosti zapnuté)

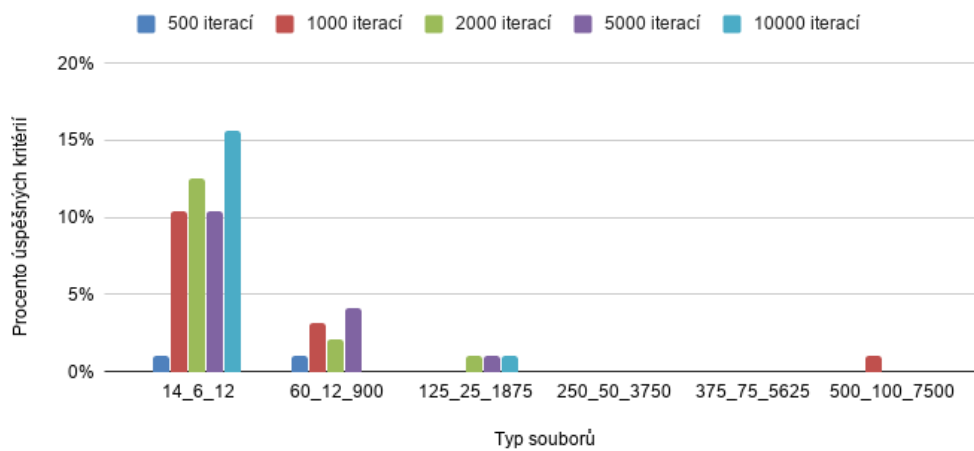


Obrázek 4.3: Procento úspěšných akcí

Procento splněných kritérií (pravděpodobnosti vypnuté)



Procento splněných kritérií (pravděpodobnosti zapnuté)



Obrázek 4.4: Procento úspěšných kritérií

Shrnutí: Ze dvou posledních grafů by se mohlo zdát, že výsledek je neuspokojivý, protože pro větší objemy dat program evidentně nekonverguje. Tím je míněno, že není splněn předpoklad, že náhodně volené akce povedou k úspěšnému složení zkoušek. Opět je nutné zdůraznit, že i přes tento nepříznivý výsledek potvrzuje testovací program hlavní předpoklad. A tím je fakt, že aplikace UIS zvládne dlouhodobě pracovat při zvýšené zátěži.

Se získanými poznatky a daty budou prováděny v rámci projektu TbUIS další experimenty, ovšem už mimo rozsah této práce.

# 5 Zátěžové testy v nástroji JMeter

JMeter je nástroj na vytváření zátěžových testů webové aplikace. Byl vytvořen scénář, který byl spuštěn v tomto nástroji s různými velikostmi databáze. Databáze byly importovány z již zmíněných a použitých souborů *imported-file-A-B-C.xml*.

V popisu scénáře jsou použita data z databáze ze souboru *imported-file-14-6-12.xml*. Scénář testu je následující (akce přihlášení a odhlášení uživatele jsou vynechány):

1. Je nahrán soubor s databází *imported-file-A-B-C.xml*
2. Učitel *easyrider* si zapíše předmět *Software Engineering*
3. Student *maroon* si zapíše tento předmět
4. Student *green* si zapíše tento předmět
5. Student *magenta* si zapíše tento předmět
6. Učitel *easyrider* vypíše termín zkoušky *2020-05-13 10:00*
7. Student *green* se přihlásí na tento termín zkoušky
8. Student *maroon* se přihlásí na tento termín zkoušky
9. Učitel *easyrider* udělí známku *B* studentovi *green*
10. Učitel *easyrider* udělí známku *F* studentovi *maroon*

Stejný scénář je použit s nahranými databázemi z ostatních souborů, jen jsou pozměněna všechna jména uživatelů a předmětu. JMeter umožňuje nahrávat akce, které uživatel provádí ve webovém prohlížeči. Tato možnost je použita při nastavování scénáře, který nástroj poté pouští.

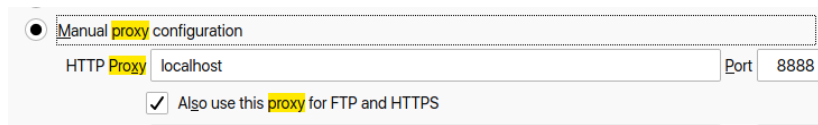
## Nastavení nástroje JMeter a prohlížeče

Nástroj JMeter je spustitelný `.jar` soubor a je umístěn ve složce `bin` v kořenovém adresáři nástroje JMeter.

Při spuštění nástroje lze pomocí nabídky *File – Templates...* vytvořit projekt podle šablony, která slouží pro nahrávání akcí provedených ve webovém prohlížeči a následně spuštění zátěžového testu. Po zvolení této nabídky je nutno vyplnit údaje následovně:

- `hostToRecord`: *localhost*
- `recordingOutputFile`: *recording.xml*
- `schemeToRecord`: *http*

Po nahrání projektu lze v levé části okna vidět jednotlivé položky projektu. Před zahájením samotného nahrání akcí podle scénáře je třeba nastavit proxy ve zvoleném prohlížeči Firefox. Proxy je nutné nastavit podle obrázku 5.1. Poté je třeba povolit proxy pro lokální připojení. V prohlížeči je třeba jít na adresu `about:config` a nastavit hodnotu položky `network.proxy.allow_hijacking_localhost` na `true`.



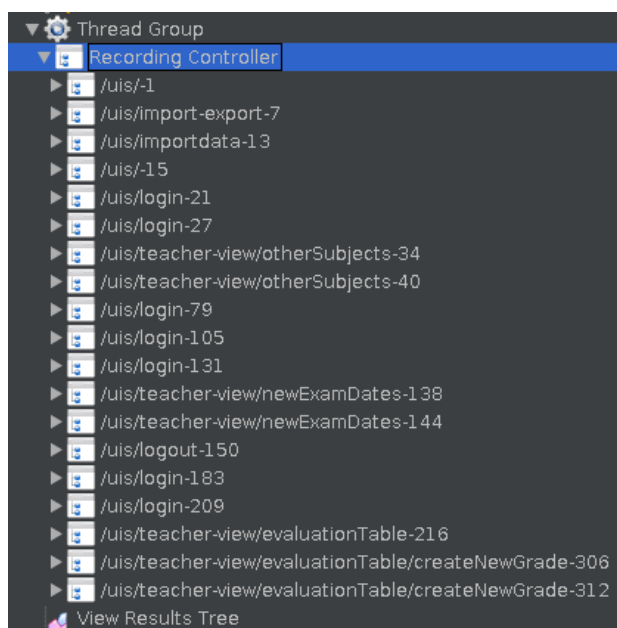
Obrázek 5.1: Konfigurace proxy v prohlížeči Firefox

## Nahrání a spuštění akcí

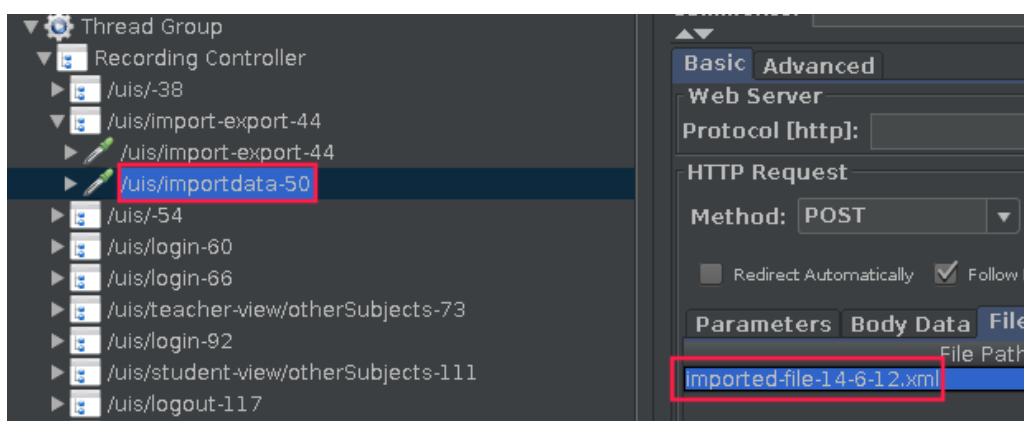
V nabídce v levé části okna nástroje JMeter se zvolí nabídka *HTTP(S) Test Script Recorder* a v pravé části okna se klikne na tlačítko *Start*. Tím se spustí nahrání akcí a veškeré provedené operace v prohlížeči se uloží do JMeteru. Nahraný scénář je na obrázku 5.2.

Před spuštěním zátěžového testu je nutné nastavit správnou cestu k nahrávanému souboru s databází. To se provede vybráním příslušné akce v levé části okna a poté v pravé části okna v záložce *Files Upload* lze editovat cestu k importovanému souboru, jak je naznačeno na obrázku 5.3. Tato cesta musí být absolutní.

Spuštění zátěžového testu se provede stisknutím tlačítka *Start* v horní části okna. Výsledky lze sledovat po kliknutí na položku *View Results Tree* v menu *Thread Group* v levé části okna. Všechny testy proběhly bez chyb, tj. pro každý databázový soubor byly akce úspěšně provedeny podle scénáře. Časy běhů jednotlivých testů pro jednotlivé soubory lze vidět v tabulce níže.



Obrázek 5.2: Nahrané akce z prohlížeče v nástroji JMeter



Obrázek 5.3: Upravení cesty k importovanému souboru

Soubor s daty	Čas běhu (sekundy)
14_6_12	1,269
60_12_900	7,312
125_25_1875	16,809
250_50_3750	45,615
375_75_5625	86,668
500_100_7500	157,732



Rozdílné časy jsou způsobeny dvěma hlavními důvody. První důvod je časová náročnost importu větší externí databáze a druhým důvodem je, že na zobrazené webové stránce bylo mnohem víc údajů, takže metodám z knihovny Support déle trvalo zpracování požadavku.

## 6 Závěr

V rámci bakalářské práce byla vyvinuta aplikace, která připraví a spustí zátěžové testy nad webovou aplikací. Nejprve proběhla analýza podpůrné knihovny Support, která je součástí projektu TbUIS. Bylo třeba se seznámit s rozhraním *PageObject*, skrze které knihovna Support komunikuje s webovou aplikací. Do zmíněné knihovny bylo s použitím rozhraní *PageObject* třeba implementovat chybějící funkcionalitu nahrávání souboru do databáze, která je využívána v zátěžových testech.

Bylo třeba porozumět fungování modulu *oracle*, který zajišťuje zapisování dat do objektů v paměti při paralelním zápisu dat do projektové databáze.

Pro uskutečnění testů s větším počtem dat bylo nutné nejprve prozkoumat již existující strukturu souboru obsahující databázová data a poté vytvořit nástroj, který data vygeneruje a poté je ukládá ve stejné podobě do souboru. Z tohoto souboru se poté data zkopírovala a konvertovala do podoby, kterou využívá modul *oracle*.

Pro zvýšení efektivity testu a zabránění pádu aplikace se musela realizovat řada omezení související s vykonáváním jednotlivých akcí. Ty mohly být vykonány teprve po průchodu filtrem podmínek, který kontroloval kontext volané akce a na základě aktuálních dat v databázi rozhodl o uskutečnění akce. Ty byly prováděny s pomocí modifikované metody Monte Carlo a poté s použitím pravděpodobností, které se snažily co nejlépe usměrnit průběh jednoho semestru na vysoké škole.

Program byl po odladění spuštěn s různě velkými datovými sadami v rozsáhlém experimentu. Jeho výsledky jsou podrobně zdokumentovány a budou sloužit jako podklady pro další výzkum.

Webová aplikace UIS byla poté otestována i externím nástrojem JMeter s různými konfiguracemi dat.

Práce splnila všechny body zadání. Ověřilo se, že aplikace UIS je schopna pracovat s větším množstvím dat po delší dobu běhu.

# Seznam zkratek

**CSV** – Comma separated values – formát souboru, kde jsou hodnoty oddělené čárkou, případně středníkem

**FTP** – File Transfer Protocol – protokol na přenos souborů

**JSON** – JavaScript Object Notation – formát zápisu dat

**LDAP** – Lightweight Directory Access Protocol – protokol pro adresářové služby

**SOAP** – Simple Object Access Protocol – protokol pro přenos strukturované informace ve formátu XML

**SQL** – Structured Query Language – jazyk používaný pro manipulaci s daty v databázi

**TbUIS** – Testbed University Information System – projekt pro výzkum testování

**UIS** – University Information System – webová aplikace simulující reálný informační systém

**UML** – Unified Modeling Language – grafický jazyk pro vizualizaci softwarového systému

**XML** – Extensible Markup Language – formát zápisu dat

# Literatura

- [1] *GitHub - DiUS/java-faker: Brings the popular ruby faker gem to Java* [online]. 2020. [cit. 2020/04/26]. Dostupné z: <https://github.com/DiUS/java-faker>.
- [2] *GitHub - FasterXML/jackson: Main Portal page for the Jackson project* [online]. 2020. [cit. 2020/04/26]. Dostupné z: <https://github.com/FasterXML/jackson>.
- [3] *Page object models :: Documentation for Selenium* [online]. 2020. [cit. 2020/04/14]. Selenium Documentation. Dostupné z: [https://www.selenium.dev/documentation/en/guidelines\\_and\\_recommendations/page\\_object\\_models/](https://www.selenium.dev/documentation/en/guidelines_and_recommendations/page_object_models/).
- [4] *The Selenium Browser Automation Project* [online]. 2020. [cit. 2020/04/20]. Selenium Documentation. Dostupné z: <https://www.selenium.dev/documentation/en/>.
- [5] *UIS – University Information System* [online]. 2020. [cit. 2020/05/05]. Dostupné z: <https://projects.kiv.zcu.cz/tbuis/web/page/uis>.
- [6] *Lorem Ipsum - All the facts - Lipsum generator* [online]. 2020. [cit. 2020/04/26]. Dostupné z: <https://www.lipsum.com/>.
- [7] HEROUT, P. *Testování pro programátory*. KOPP, 2016. ISBN 978-80-7232-481-1.
- [8] MATYÁŠ, J. *Aplikace s možností injekce chyb pro ověřování kvality testů*. Diplomová práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Plzeň, 2018, Vedoucí práce Pavel Herout.
- [9] POUBOVÁ, J. *Generování automatizovaných funkcionálních testů*. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Plzeň, 2019, Vedoucí práce Pavel Herout.
- [10] ČARNOGURSKÝ, J. *Mikroframework na bázi komponent Symphony pro publikování vědeckých projektů*. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Plzeň, 2019, Vedoucí práce Pavel Herout.

# A Uživatelská příručka

## A.1 Prerekvizity

Ke správnému přeložení a běhu aplikace je třeba mít nainstalovanou Javu minimálně verze 11, XAMPP (na OS Windows) či alternativu na OS Linux (na použitém stroji byl použit program *lampp*), Tomcat minimálně verze 7 a Maven pro překlad projektu (testováno na verzi 3.6.3).

## A.2 Před spuštěním

### A.2.1 Příprava databáze

Nejprve je nutné nastavit databázi. Po spuštění nástroje XAMPP a spuštění MySQL (v případě použití nástroje *lampp* stačí příkaz `lampp start`) je v nástroji *phpMyAdmin* třeba vytvořit databázi *uis-web-db* s kódováním *utf8\_general\_ci*. Poté se na ni klikne a v záložce *Oprávnění* se přidá uživatel s následujícími údaji:

- Jméno uživatele: `uis-web`
- Název počítače: `localhost`
- Heslo: `uis`

V sekci *Databáze pro uživatele* se uživateli povolí všechna oprávnění a v sekci *Globální oprávnění* se zaškrtně vše.

Poté se do složky `TOMCAT/webapps/`, kde `TOMCAT` je složka, kde je nainstalovaný nástroj Tomcat, přesune soubor `uis.war` a pustí se Tomcat.

Webová aplikace UIS se nachází na adrese `http://localhost:8080/uis/`.

### A.2.2 Konfigurační soubor

V kořenovém adresáři projektu se nachází ukázkový konfigurační soubor pojmenovaný `configurations.txt.default`. Ten je potřeba přejmenovat na `configurations.txt` a upravit tak, aby v něm uvedené cesty k souborům a další nastavení byly validní. Položka *URL\_Base* určuje, která stránka se otevře jako první při spuštění testů. *WebBrowserType* určuje používaný

prohlížeč a pod tímto nastavením lze najít jednotlivé cesty k ovladačům prohlížečů. Ty je třeba přenastavit na úplné cesty v systému. Prohlížeč Opera navíc potřebuje specifikovat cestu k jeho spustitelnému souboru.

Dále je v konfiguračním souboru možnost nastavit spuštění prohlížeče v *headless* módu, což znamená, že poběží na pozadí bez otevírání okna.

### A.2.3 Nastavení konstant

Ve třídě `uis.constants.Constants` je třeba nastavit tři konstanty. První konstantou je `MAIN_LOOP_LIMIT`, která určuje maximální počet provedených iterací neboli pokusů o provedení akce. Další hodnotou `ACTIONS_WITH_PROBABILITIES` se zapíná nebo vypíná použití pravděpodobností při výběru akcí. Poslední konstantou `DATABASE_SIZE` se zvolí soubor pro import dat do databáze a soubor s očekávanými výsledky.

## A.3 Překlad a spuštění

Nejprve je třeba nainstalovat knihovnu Support. Ve složce `TbUIS-support`, kde se nachází příslušný soubor `pom.xml`, se spustí v příkazovém řádku příkaz `mvn install`. Tím se nainstaluje externí knihovna do lokálního repozitáře nástroje Maven a lze ji využít v dalších programech.

V kořenovém adresáři projektu `TbUIS-LoadTests` (kde se nachází soubor `pom.xml`) se pomocí příkazu `mvn install` přeloží projekt a vznikne spustitelný JAR soubor `uis-loadTests-1.6.0-jar-with-dependencies.jar`, který se nachází ve složce `target`. Tento soubor je třeba přesunout do kořenového adresáře projektu, odkud je možné ho spustit.

### A.3.1 JMeter

Nástroj JMeter se nachází v samostatné složce `jmeter` mimo projekt `TbUIS-LoadTests`. Lze ho spustit pomocí souboru `ApacheJMeter.jar`, který je ve složce `bin`.

## B Obsah CD

- **TbUIS-LoadTests** – obsahuje vyvinutou aplikaci, která provádí zá-  
těžové testy, zdrojové kódy a přeložené soubory, vstupní data a spus-  
titelný `.jar` soubor
- **TbUIS-support** – obsahuje zdrojové kódy a soubor `pom.xml` externí  
knihovny Support, které jsou potřeba pro instalaci knihovny do lokál-  
ního repozitáře nástroje Maven
- **selenium-webdrivers** – obsahuje ovladače pro Selenium Webdriver  
(Firefox, Google Chrome, Opera)
- **UIS** – obsahuje `.war` soubor
- **jmeter** – obsahuje externí testovací nástroj JMeter
- **text** – obsahuje text práce ve formátu PDF a zdrojové soubory textu